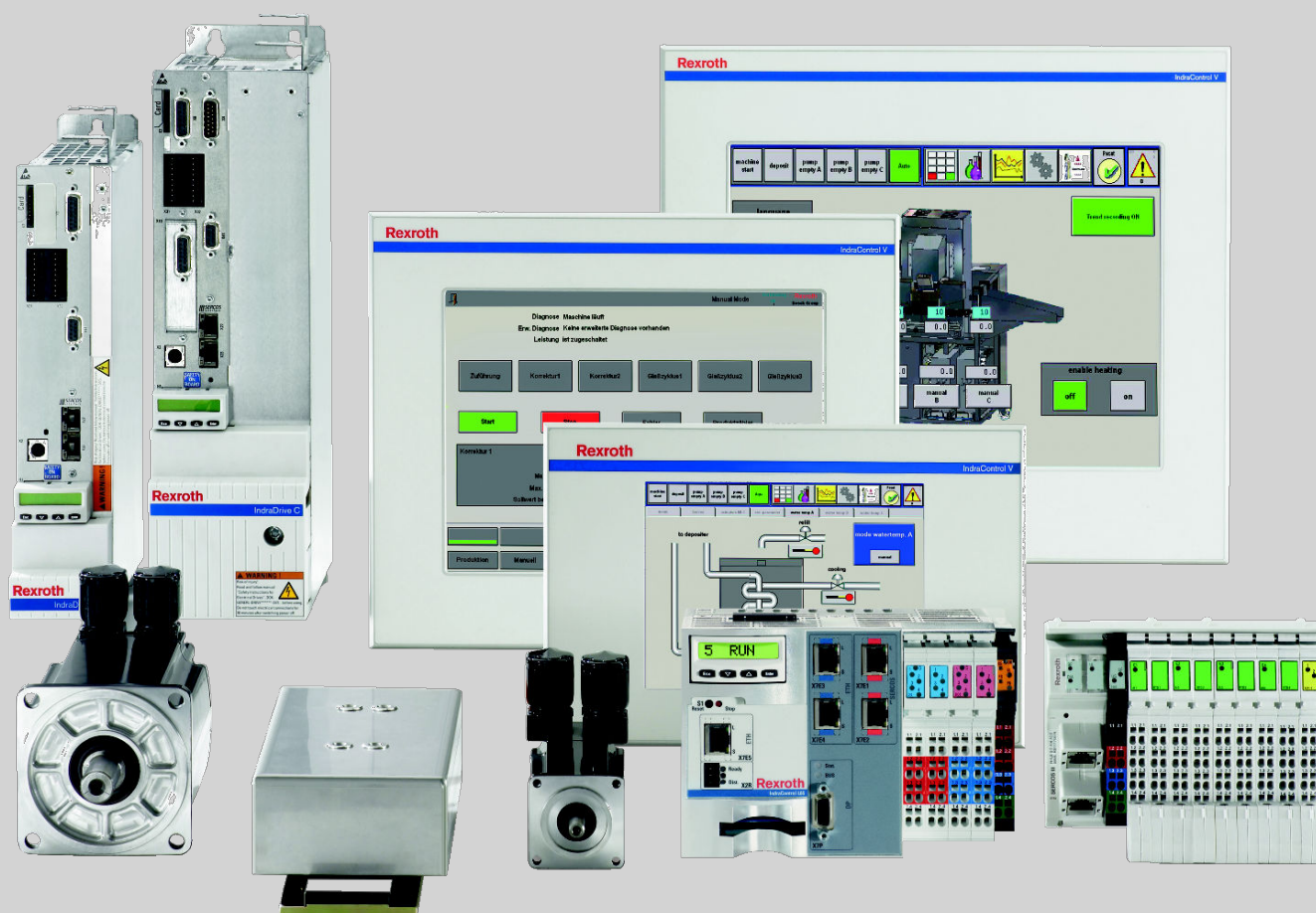


MLC 14VRS

Technologie-Bibliotheken

**Bibliothek
R911341510**

Ausgabe 08



Titel	MLC 14VRS Technologie-Bibliotheken
Art der Dokumentation	Bibliothek
Dokumentations-Type	DOK-MLC***-TF*LIB**V14-LI08-DE-P
Interner Ablagevermerk	RS-d83f0c22df08c8b60a6846a5005708f8-8-de-DE-12
Änderungsverlauf	Ausgabe 08, 2020-01 Siehe "Ausgaben dieser Dokumentation" auf Seite 1
Schutzvermerk	© Bosch Rexroth AG 2020 Alle Rechte vorbehalten, auch bzgl. jeder Verfügung, Verwertung, Reproduktion, Bearbeitung, Weitergabe sowie für den Fall von Schutzrechtsanmeldungen.
Verbindlichkeit	Die angegebenen Daten dienen allein der Produktbeschreibung und sind nicht als zugesicherte Eigenschaften im Rechtssinne zu verstehen. Änderungen im Inhalt der Dokumentation und Liefermöglichkeiten der Produkte sind vorbehalten.
Redaktion	Systementwicklung Automation Motion Logic Control, VK (TaDo)

Inhaltsverzeichnis

	Seite
1	Zu dieser Dokumentation..... 1
1.1	Gültigkeit der Dokumentation..... 1
1.2	Bibliotheken und Zielsysteme..... 2
1.3	Besonderheiten von MLD..... 3
1.4	Gebrauch der Sicherheitshinweise..... 7
1.4.1	Aufbau der Sicherheitshinweise..... 7
1.4.2	Erläuterung der Signalwörter und der Signalgrafik..... 7
1.4.3	Verwendete Symbole..... 9
1.4.4	Erläuterung der Signalgrafik auf dem Gerät..... 9
1.5	Erforderliche und ergänzende Dokumentationen XLC/MLC..... 9
1.6	Bezeichnungen und Abkürzungen..... 15
1.7	Kundenfeedback..... 16
2	ML_TechBase.library // MX_TechBase.library..... 17
2.1	Einführung und Übersicht..... 17
2.2	IMC-Interface..... 19
2.2.1	Einführung und Übersicht..... 19
2.2.2	MB_IMC_CONTROL_TYPE01..... 20
2.2.3	MB_IMC_ADMINISTRATION..... 20
2.2.4	MB_IMC_MODE_CONTROL..... 21
2.2.5	MB_IMC_STATUS_TYPE01..... 21
2.2.6	MB_IMC_ADMIN_STATUS..... 21
2.2.7	MB_IMC_MODE_STATUS..... 22
2.2.8	MB_IMC_SERCOS_PHASE..... 22
2.2.9	MB_IMC_CONTROL_MODE_STATUS..... 23
2.2.10	MB_IMC_DIAGNOSIS..... 23
2.2.11	MB_IMC_ERROR_SEVERITY..... 23
2.3	Trigonometrische Funktionen..... 24
2.3.1	Einführung und Übersicht..... 24
2.3.2	ML_DegToInc..... 24
2.3.3	ML_DegToRad..... 25
2.3.4	ML_IncToDeg..... 25
2.3.5	ML_IncToRad..... 26
2.3.6	ML_RadToDeg..... 27
2.3.7	ML_RadToInc..... 27
2.4	Systeminformationen..... 28
2.4.1	MB_GetSystemInfo..... 28
2.4.2	MB_SYSTEM_TYPE..... 31
3	ML_TechMotionBase.library // MX_TechMotionBase.library..... 33
3.1	Einführung und Übersicht..... 33
3.2	Sichern und Wiederherstellen von Antriebsparametern..... 35
3.2.1	Einführung und Übersicht..... 35

	Seite
3.2.2 ML_AxisBackup.....	35
3.2.3 ML_AxisRestore.....	38
3.2.4 ML_AxisBackup_Type02.....	41
3.2.5 ML_AxisRestore_Type02.....	44
3.2.6 ML_BACKUP_OPTIONS.....	49
3.2.7 ML_BACKUP_PARAMS.....	49
3.2.8 ML_BACKUP_ERRORS.....	49
3.3 Messwertgeber-Unterstützung.....	50
3.3.1 Einführung und Übersicht.....	50
3.3.2 MB_DistanceAccumulator.....	50
3.3.3 MB_AxisDistanceAccumulator.....	51
3.4 Nockenschaltwerk, Anwendung.....	52
3.4.1 Einführung und Übersicht.....	52
3.4.2 MC_DigitalCamSwitch.....	53
3.4.3 MC_CAMSWITCH_REF.....	69
3.4.4 ML_BASIC_CAMSWITCH_REF.....	69
3.4.5 MC_TRACK_REF.....	69
3.4.6 ML_BASIC_TRACK_REF.....	70
3.4.7 MC_OUTPUT_REF.....	70
3.4.8 MB_DigitalCamSwitchOvs.....	71
3.4.9 MB_CAMSWITCH_REF_OVS.....	97
3.4.10 ML_BASIC_CAMSWITCH_REF_OVS.....	98
3.4.11 MB_TRACK_REF_OVS.....	98
3.4.12 ML_BASIC_TRACK_REF_OVS.....	98
3.5 Download von Kurvenscheiben.....	99
3.5.1 Einführung und Übersicht.....	99
3.5.2 MB_LoadCamData.....	99
3.5.3 MB_CAMPROF.....	102
3.5.4 MB_CAMTABLE.....	102
3.6 PositionFiltering (Filtern inkrementeller Positionen).....	103
3.6.1 Einführung und Übersicht.....	103
3.6.2 MB_PositionFilterType01.....	103
3.7 Positionsüberwachung.....	106
3.7.1 Einführung und Übersicht.....	106
3.7.2 MB_PositionMonitoring_GantryCant.....	106
3.7.3 MB_POS_MON_GANTRY_CANT.....	108
3.7.4 MB_PositionMonitoring_PosCollision.....	109
3.7.5 MB_POS_MON_POS_COLLISION.....	111
3.8 S-Kurve, Programmierung von Bewegungsabläufen.....	112
3.8.1 Einführung und Übersicht.....	112
3.8.2 MB_ReachVelocityType02.....	112
3.9 Messtasterfunktionsbausteine.....	116
3.9.1 Einführung und Übersicht.....	116
3.9.2 Messtasterfunktionsbausteine – Komponenten und Parametrierung.....	118
Hardware.....	118
Firmware.....	118

	Seite
Parametrierung der Messtasterfunktionsbausteine für MLC/XLC und MLD.....	118
3.9.3 MB_InitTouchProbe.....	127
3.9.4 MB_PROBE_EXECUTION_MODE.....	135
3.9.5 MB_TouchProbe.....	135
3.9.6 MB_TouchProbeContinuous.....	138
3.9.7 MB_WriteExpectWindow.....	142
3.9.8 MB_AbortTrigger.....	145
3.10 Zyklischer Datenkanäle, Administration	147
3.10.1 Einführung und Übersicht.....	147
3.10.2 MB_AllocCyclicParameter.....	148
3.10.3 MB_FreeCyclicParameter.....	150
3.10.4 MB_GetCyclicChannelConfig.....	152
3.10.5 MB_GetCyclicParameterHandle.....	154
3.10.6 MB_ReadCyclicParameter.....	155
3.10.7 MB_ReadCyclicRealParameter.....	156
3.10.8 MB_WriteCyclicParameter.....	157
3.10.9 MB_WriteCyclicRealParameter.....	157
3.10.10 MB_GetLastCyclicParameterError.....	158
3.10.11 MB_CYCLIC_PARAM_REF.....	159
3.10.12 MB_CYCLIC_TYPES.....	159
3.10.13 MB_CYCLIC_CHANNELS.....	160
3.10.14 MB_CYCLIC_CHANNEL_CONFIG.....	161
4 ML_TechMotion.library // MX_TechMotion.....	163
4.1 Übersicht über die Funktionsbausteine.....	163
4.2 CamLock - Taktsynchronen Ein- und Auskoppeln.....	164
4.2.1 Einführung und Übersicht	164
4.2.2 CamLock - Applikationsbeispiel.....	165
4.2.3 MB_PrepareCams.....	168
4.2.4 MB_CamLock.....	170
4.2.5 MB_CamLock Parametrierung.....	177
4.3 Fliegende Säge.....	177
4.3.1 Einführung und Übersicht.....	177
4.3.2 ML_FlyingShear / MX_FlyingShear.....	178
4.3.3 MB_FS_RETAIN_DATA.....	186
4.3.4 ML_FlyingShear - Besonderheiten für MLC.....	186
4.3.5 MX_FlyingShear - Besonderheiten für MLD.....	187
4.4 MagicBelt.....	191
4.4.1 Einführung und Überblick.....	191
4.4.2 MB_MagicBeltType01.....	192
4.4.3 Datentypen, MagicBelt.....	200
MB_MBELT_MODES.....	200
MB_MBELT_GENERAL.....	200
MB_MBELT_MODEJOG.....	202
MB_MBELT_MODETUNE.....	202
MB_MBELT_MODEAUTO.....	203

	Seite
MB_MBELT_STARTTYPE.....	204
MB_MBELT_STOPTYPE.....	204
MB_MBELT_INFEED.....	204
MB_MBELT_FEEDTYPE.....	207
MB_MBELT_INFEEDSTEPS.....	207
MB_MBELT_OUTFEED.....	209
MB_MBELT_MOVE.....	210
MB_MBELT_STATUS.....	211
4.4.4 Betriebsarten, MagicBelt.....	213
Betriebsarten, MagicBelt, Einführung und Überblick.....	213
Jog-Mode, MagicBelt.....	214
Tune-Mode, MagicBelt.....	214
Automatik-Mode, MagicBelt.....	215
Automatik-Mode, MagicBelt, Allgemeines.....	215
Start, Automatik-Mode, MagicBelt.....	215
Beladen, Automatik-Mode, MagicBelt.....	217
Getaktetes Beladen, Automatik-Mode, MagicBelt.....	217
Getaktetes Beladen mit Eintaktung in fahrenden Zug, Automatik-Mode, MagicBelt.....	220
Infeed mit konstanter Geschwindigkeit, Automatik-Mode, MagicBelt.....	221
Synchrones Beladen, Automatik-Mode, MagicBelt.....	222
Fahrten zwischen Be- und Entladestation, Automatik-Mode, MagicBelt.....	224
Entladen, Automatik-Mode, MagicBelt.....	224
Stopp, Automatik-Mode, MagicBelt.....	226
Überwachungen, Automatik-Mode, MagicBelt.....	227
Wiederanlauf, Automatik-Mode, MagicBelt.....	229
4.4.5 Inbetriebnahmehinweise, MagicBelt.....	229
4.5 Schlaufenregelung.....	230
4.5.1 Einführung und Übersicht.....	230
4.5.2 ML_SagControlC.....	231
4.5.3 ML_SagControlA.....	234
4.6 SmartBelt.....	238
4.6.1 MB_SmartBeltType01.....	238
4.6.2 MB_SMARTBELT_STATISTICS.....	245
4.6.3 Einführung in die SmartBelt-Verwendung.....	245
SmartBelt-Verwendung.....	245
Stauförderer.....	246
Abfolgegrad.....	247
Reibung.....	248
4.6.4 Projektplanung.....	249
Allgemeines.....	249
Mechanische Anforderungen an Band und Maschine.....	249
Antriebsanforderungen.....	250
Antriebsparametereinstellungen.....	251
SPS-Programmierung.....	253
Angabe des "SmartBeltPitch".....	255
Berechnung des "ProbeSetpoint".....	257

	Seite
4.6.5 SmartBelt-Beschränkungen und -Verhalten.....	258
Beschränkungen.....	258
Zwei oder mehr Produkte auf einem Band.....	258
Mehrere Produkte pro Leitachszklus.....	259
4.6.6 Tipps und Tricks.....	259
Korrekturabstand.....	259
Schlupf.....	260
4.6.7 Verbreitete Fehler und Missverständnisse.....	261
4.7 Kurvenscheibentabellen für Technologiefunktionen.....	261
4.7.1 Einführung und Übersicht.....	261
4.7.2 MB_PilgrimStepCalcType01.....	262
4.7.3 MB_PrintLengthCorrCalcType01.....	272
4.8 Zugkraftregler.....	277
4.8.1 Einführung und Übersicht.....	277
4.8.2 MB_TensionControlLoadCellType01.....	278
4.8.3 MB_TensionControlLoadCellMatParType01.....	292
4.8.4 MB_TENSION_CONTROL_MATERIAL.....	305
4.8.5 MB_TENSION_CONTROL_MATERIAL_TYPE.....	305
4.8.6 MB_TENSION_CONTROL_MODE.....	306
5 ML_TechRegi.library // MX_TechRegi.lib.....	307
5.1 Einführung und Übersicht.....	307
5.2 Messfunktionsbausteine für Registerregler.....	308
5.2.1 Einführung und Übersicht.....	308
5.2.2 MB_RegiMeasuringType01.....	308
5.2.3 MB_RegiMeasuringType02.....	310
5.2.4 MB_RegiMeasuringType03.....	314
5.3 Registerqualitäts-FB.....	318
5.3.1 Einführung und Übersicht.....	318
5.3.2 IL_RegiQualityType01.....	318
5.3.3 IL_REGI_QUALITY.....	321
5.4 Stellfunktionsbausteine für Registerregler.....	321
5.4.1 Einführung und Übersicht.....	321
5.4.2 MB_RegiRegulateType01.....	321
5.4.3 MB_RegiRegulateType02.....	326
5.4.4 MB_RegiRegulateType03.....	334
5.4.5 MB_RegiRegulateType04.....	341
5.4.6 MB_CORRECTION_MODE.....	349
5.4.7 MB_RAMPING_MODE.....	350
5.5 Registersollwertübernahme-FB.....	350
5.5.1 Einführung und Übersicht.....	350
5.5.2 MB_SetpointLockType01.....	350
5.6 Registerregler-FB.....	352
5.6.1 Einführung und Übersicht.....	352
5.6.2 MB_RegisterControllerType02.....	352
5.6.3 MB_RegisterControllerType04.....	362

	Seite
5.6.4	Konfiguration des Registerreglers..... 374
	Überblick..... 374
	Einsatz der Registerregelung..... 375
	Überblick..... 375
	Konfiguration des Messtaster-FB..... 375
	Konfiguration des Messtaster-FB und des RegisterControllerType04-FB..... 376
	Parametrierung des Registerreglers..... 376
5.6.5	Registerregler - Anwendungsbeispiele..... 378
	Stanzen beim Etikettendruck..... 378
	Insetter-Regelung..... 379
	Schlauchbeutelmaschine..... 380
5.7	Seitenregisterregler-FB..... 381
5.7.1	Einführung und Übersicht..... 381
5.7.2	MB_RegisterControllerSideType01..... 381
5.7.3	Konfiguration des Seitenregisters..... 389
6	ML_TechCrank.library..... 395
6.1	Einführung und Übersicht..... 395
6.2	Kniehebelkinematik (Bell Crank Kinematik)..... 396
6.2.1	Einführung und Übersicht..... 396
6.2.2	Allgemeine Definitionen..... 396
6.2.3	MB_BellCrankData..... 400
6.2.4	MB_YvirtToYmech..... 402
6.2.5	MB_CamTableBellCrank..... 404
6.2.6	MB_CamTableBellCrankSuperimposed..... 408
6.2.7	MB_PhiToYvirt..... 413
6.2.8	MB_YvirtToPhi..... 415
6.2.9	MB_YvirtToPhiPoly5..... 417
6.2.10	MB_YVirtToWorkRange..... 419
6.3	Kurbelkinematik..... 420
6.3.1	Einführung und Übersicht..... 420
6.3.2	Allgemeine Definitionen..... 422
	Definition der grundlegenden Variablen bei der Kurbelkinematik..... 422
	Mechanische (Xmech) und virtuelle (Xvirt) translatorische Position..... 424
6.3.3	MB_CamTableCrank..... 425
6.3.4	MB_CRANK..... 429
6.3.5	MB_CamTableCrankSuperimposed..... 430
6.3.6	MB_PhiToXvirt..... 435
6.3.7	MB_MasterToPhi..... 436
6.3.8	MB_XvirtToXmech..... 437
7	ML_TechTensionAdvanced.library..... 439
7.1	Einführung und Übersicht..... 439
7.2	MB_TensionIdentModelParamType01..... 439
7.3	MB_TENSION_IDENT_EXTENDED_SETTINGS..... 447

	Seite
7.4 MB_TensionControlLoadCellType02.....	447
7.5 MB_TensionControlLoadCellMatParType02.....	463
7.6 MB_TENSION_CONTROL_CONFIG.....	478
7.7 MB_TENSION_CONTROL_ADJUST.....	478
7.8 MB_TENSION_PRE_CTRL.....	479
7.9 Entkopplungsnetzwerk.....	479
7.9.1 MB_TensionDecouplingNetworkType01.....	479
7.9.2 MB_TENSION_DECOUPLING_CONTROLLER_VALUE.....	487
7.9.3 MB_TENSION_DECOUPLING_AXIS_VALUE.....	488
8 RMB_TechWinder.library // MX_TechWinder.....	491
8.1 Einführung und Übersicht.....	491
8.2 Tänzerlageregelung.....	493
8.2.1 Einführung und Übersicht.....	493
8.2.2 MB_DancerControlType03.....	495
8.2.3 MB_DancerControlType04.....	501
8.2.4 Inbetriebnahme Tänzerlageregelung.....	508
8.3 Durchmesserberechnung im Wickelrechner.....	509
8.3.1 Einführung und Übersicht.....	509
8.3.2 MB_DiameterCalculatorType03.....	509
8.3.3 Inbetriebnahme des Durchmesserrechners	518
8.4 Erfassung eines extern gemessenen Durchmessers.....	519
8.4.1 Einführung und Übersicht.....	519
8.4.2 MB_DiameterMeasurementType01.....	519
8.5 Wickelrechner mit Tänzer	524
8.5.1 Einführung und Übersicht.....	524
8.5.2 MB_WinderDancerCtrlType01.....	525
8.5.3 MB_WinderDancerCtrlType02.....	532
8.5.4 Inbetriebnahme des Wickelrechners mit Tänzer	539
8.6 Wickelrechner mit Zugkraftsteuerung bzw. Zugkraftregelung.....	541
8.6.1 Einführung und Übersicht.....	541
8.6.2 MB_WinderTensionCtrlType02.....	542
8.6.3 Inbetriebnahme des Wickelrechners mit Zugkraftsteuerung bzw. Zugkraftregelung.....	556
8.7 Wickelrechner mit Zugkraftregelung und Drehzahlkorrektur.....	560
8.7.1 Einführung und Übersicht.....	560
8.7.2 MB_WinderTensionCtrlSpeedType01.....	560
8.7.3 Inbetriebnahme des Wicklers mit Zugkraftregelung und Drehzahlkorrektur.....	573
8.8 Ergänzende Funktionsbausteine.....	576
8.8.1 Einführung und Übersicht.....	576
8.8.2 MB_CalcInertiaLimitType02.....	577
8.8.3 MB_UnwindMaterialType02.....	579
8.8.4 MB_WindTaperProfileType01.....	585
8.8.5 MB_WindSpeedControlAdaptionType01.....	588
8.8.6 MB_TraverseControlType01.....	592
8.8.7 MB_DancerCalcControlParamType01.....	608
8.9 Datentypen.....	612

	Seite
8.9.1 Einführung und Übersicht.....	612
8.9.2 MB_TRAVERSE_ADAPTION.....	612
8.9.3 MB_TRAVERSE_DIRECTION.....	613
8.9.4 MB_TRAVERSE_MODE_LAYER_COUNTER.....	613
8.9.5 MB_TRAVERSE_UNWIND.....	613
8.9.6 MB_TRAVERSE_UNWIND_CONTROL.....	614
8.9.7 MB_TRAVERSE_UNWIND_MODE.....	615
8.9.8 MB_WIND_CTRL_TYPE	615
8.9.9 MB_WIND_DANCER_LIMITS.....	615
8.9.10 MB_WIND_DIAM_CALC_MODE	616
8.9.11 MB_WIND_DIAMETER	616
8.9.12 MB_WIND_DIRECTION	618
8.9.13 MB_WIND_FRICTION	618
8.9.14 MB_WIND_TAPER_PROFILE.....	618
8.9.15 MB_WIND_TENS_DIAG	618
8.9.16 MB_WIND_TENSION	619
8.9.17 MB_WIND_TENSION_MODE	619
8.9.18 MB_WIND_TORQUE_LIMITS.....	619
8.9.19 MB_DANCER_CONTROL_MODE.....	620
8.10 Parameterübersicht.....	620
 9 RMB_TechCrossCutCrossSeal.library // MX_TechCrossCutCrossSeal.....	 623
9.1 Einführung und Übersicht.....	623
9.2 MB_CrossCutSealType01.....	623
9.3 MB_CrossCutterCalcType04.....	638
9.4 MB_CrossCutterCalcType05.....	647
9.5 MB_CAM_TABLE_DATA.....	653
9.6 MB_CORR_PROFILE.....	655
9.7 MB_PUSH_OUT_CONFIG.....	655
9.8 MB_RESOLUTION.....	657
 10 RMB_TechCam.library.....	 659
10.1 Einführung und Übersicht.....	659
10.2 MB_CoordinatePairCamType01.....	659
10.3 MB_COORDINATE_PAIR_CAM_SELECT.....	666
10.4 MB_COORDINATE_PAIR_TYPE01.....	666
10.5 MB_FeedMovementCalcType01.....	667
10.6 MB_FeedMovementCalcType02.....	673
10.7 MB_CURVETYPE.....	679
10.8 Slotter, Einführung und Definitionen.....	680
10.9 MB_SlotterCalcType01.....	685
10.10 MB_SlotterCalcType02.....	697
10.11 MB_CAMPROF_SLOTTER.....	708

	Seite
11 RIL_ParameterChannel.library	709
11.1 Einführung und Übersicht	709
11.2 IL_ParameterChannel	709
11.3 IL_ParameterChannel – Protokoll, Fehlerbehandlung und Inbetriebnahme	716
11.3.1 Beschreibung des unterlagerten Protokolls	716
Parameterkanal im zyklischen Datenkanal	716
Steuer-/Statuswort	716
Index und Subindex	718
Initialisierung der Kommunikation	720
Fehlertelegramme	721
11.3.2 Inbetriebnahme des Bausteins	723
11.3.3 Unterschiede zu Altsystemen	727
Allgemeines	727
Auf dem Feldbus-Slave	727
Auf dem Feldbus-Master	728
12 ML_JobComm.library	731
12.1 Einführung und Übersicht	731
12.2 Architektur der JobComm-Bibliothek	731
12.3 Funktionsbausteine	733
12.3.1 Einführung und Übersicht	733
12.3.2 ML_SendRecvJobAss	734
12.3.3 ML_SendJobTlg	737
12.3.4 ML_RecvJobTlg	739
12.3.5 ML_CtrlJobConnect	742
12.3.6 ML_ShowJobConnect	745
12.3.7 ML_DebugJobConnect	747
12.3.8 ML_DebugJobConnect_Cyclic	750
12.4 Datentypen	753
12.4.1 ML_CFG_CONNECT_JOB	753
12.4.2 ML_CFG_JUNK_JOB	753
12.4.3 ML_CFG_SEGMENT_JOB	754
12.4.4 ML_CFG_SEND_RECV_JOB	754
12.4.5 ML_CONNECT_CMD_JOB	754
12.4.6 ML_CONNECT_STATUS_JOB	754
12.4.7 ML_JOB_ASS_BUFF	755
12.4.8 ML_JOB_ASS_BUFF_ADDR	755
12.4.9 ML_JOB_ASS_BUFF_DATA	755
12.4.10 ML_JOB_ASS_BUFF_SCL	756
12.4.11 ML_JOB_BUFF	756
12.4.12 ML_JOB_BUFF_DATA	756
12.4.13 ML_JOB_BUFF_MANAGEMENT	757
12.4.14 ML_JOB_DBG_JOB_ASS_BUFF	757
12.4.15 ML_JOB_DBG_JOB_BUFF	757
12.4.16 ML_JOB_DBG_TBL_CONNECT	758
12.4.17 ML_JOB_DBG_UDP_BUFF	758

	Seite
12.4.18 ML_LINK_SEND_RECV_JOB.....	758
12.4.19 ML_TBL_CONNECT_JOB.....	758
12.5 SCL-Protokoll.....	759
12.5.1 Einführung und Übersicht.....	759
12.5.2 Ablauf der Kommunikation.....	759
12.5.3 Aufbau des UDP-Datagramms.....	761
12.5.4 Fehlercodes des SCL-Protokolls.....	762
 13 Service und Support.....	 765
 Index.....	 767

1 Zu dieser Dokumentation

Ausgaben dieser Dokumentation

Ausgabe	Stand	Bemerkung
01	2014-05	Erstausgabe MLC/XLC 14VRS
02	2014-08	MLD (IL 2G): MX_TechBase, MX_TechMotionBase MB_PilgrimStepCalcType01, MB_PositionFilterType01, MB_CrossCutterCalcType05
03	2015-07	Hinweis: Anwendung der Punktetabelle mit dem FlexProfile Update: ML_TechMotion_MagicBelt, Neu: MB_DigitalCamSwitchOvs01, ML_JobComm.library
04	2015-09	MLD (IL 2G): RMB_TechWinder ML_TechMotionBase: ML_AxisBackup_Type02, ML_AxisRestore_Type02; Update MB_DigitalCamSwitchOvs
05	2016-05	ML_TechBase: ImcInterface ML_TechCrank: Update Beschreibung Kniehebelkinematik (svg) ML_TechMotion: ML_TechMotion_CamLock, ML_TechMotion_SmartBelt_MLC_MLD (svg) ML_TechRegi (svg) RMB_TechWinder (svg) RMB_TechCrosscutCrossSeal: MB_CrossCutterCalcType05 ML_TechMotionBase: MB_DigitalCamSwitchOvs für MLC und MLD
06	2017-08	Update: ML_TechMotionBase, RMB_TechCam, RMB_TechWinder, Übersicht "Bibliotheken und Zielsysteme"
07	2018-03	MB_TouchProbe_14, MB_TouchProbeContinuous_14, MB_RegiRegulateType02, MB_RegiMeasuringType03, MB_SetpointLockType01, MB_RegisterControllerType04, MB_DigitalCamSwitchOvs, MB_CAM_TABLE_DATA, MB_CORR_PROFILE, MB_PUSH_OUT_CONFIG ergänzt bzw. geändert MB_DancerControlType03, MB_DancerControlType04, MB_DiameterCalculatorType03, MB_WinderDancerCtrlType01, MB_WinderDancerCtrlType02, MB_WinderTensionCtrlSpeedType01, MB_WinderTensionCtrlType02 um Marginalienblock Inbetriebnahme ergänzt MB_WinderTensionCtrlSpeedType01 um "Beeinflussung der Einrastbedingung" ergänzt
08	2020-01	Neu: Kap. 8.8.7 "MB_DancerCalcControlParamType01" auf Seite 608

Tab. 1-1: Änderungsverlauf

1.1 Gültigkeit der Dokumentation

Zielgruppe Diese Dokumentation richtet sich an Anwender, die für Ihre Applikationen Funktionen und Funktionsbausteine aus den Technologie-Bibliotheken einsetzen.

Diese Dokumentation beschreibt die Bibliotheken zur Wicklerfunktionalität, Registerreglerfunktionalität und Querschneiderfunktionalität und deren Grundlagen.

Die Dokumentation enthält Funktionsbausteine, Funktionen, Schnittstellen und Datentypen der folgenden Bibliotheken:

- ML_TechBase // MX_TechBase
- ML_TechMotionBase // MX_TechMotionBase
- ML_TechMotion // MX_TechMotion
- ML_TechRegi // MX_TechRegi.lib

Zu dieser Dokumentation

- ML_TechCrank
- ML_TechTensionAdvanced
- ML_JobComm
- RMB_TechWinder // MX_TechWinder.lib
- RMB_TechCrossCutCrossSeal // MX_TechCrossCutCrossSeal
- RMB_TechCam

Übersicht über Zielgruppen und Produktphasen

In der folgenden Grafik beziehen sich die umrandeten Aktivitäten, Produktphasen und Zielgruppen auf die vorliegende Dokumentation.

Beispiel: In der Produktphase "Inbetriebnahme" kann die Zielgruppe "Technologie" mit Hilfe dieser Dokumentation die Aktivitäten "optimieren" und "erproben" ausführen.

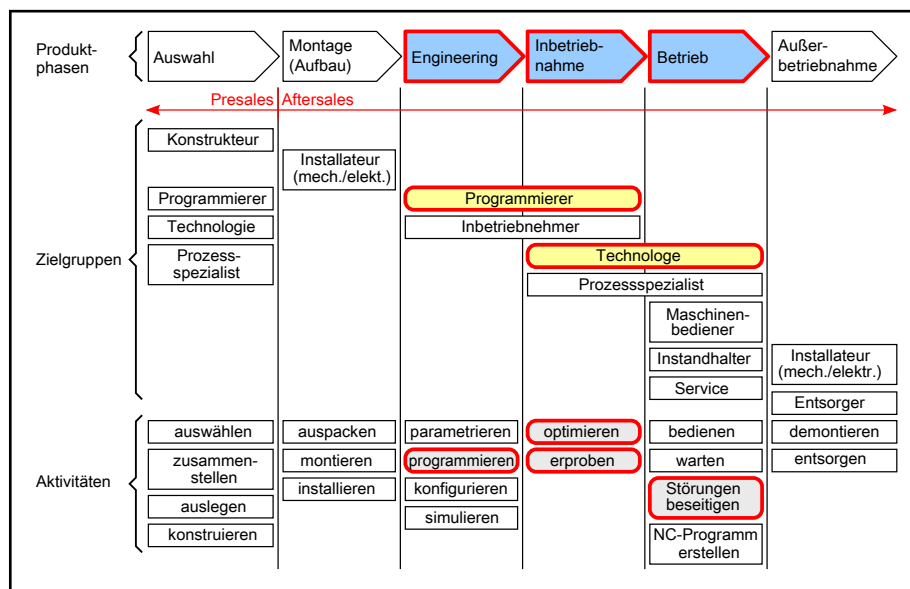


Abb. 1-1: Zuordnung der vorliegenden Dokumentation zu den Zielgruppen, Produktphasen und den Aktivitäten der Zielgruppe

1.2 Bibliotheken und Zielsysteme

Die nachfolgende beiden Tabellen zeigen die unterstützten Zielsysteme für die Technologie-Bibliotheken nebst den dazu erforderlichen Funktionspaketen.

Bibliothek	MLC mit Funktionspaket
ML_TechBase.library	PLC
ML_TechMotionBase.library	MOT
ML_TechMotion.library	MOT
ML_TechRegi.library	TEC
ML_TechCrank.library, Seite 395	TEC
ML_TechTensionAdvanced.library, Seite 439	Lizenz
ML_JobComm.library, Seite 731	PLC
RMB_TechWinder.library	TEC
RMB_TechCrosscutCrossSeal.library	TEC

Bibliothek	MLC mit Funktionspaket
RMB_TechCam.library, Seite 659	TEC
RIL_ParameterChannel.library, Seite 709	PLC

PLC Basispaket
MOT Motion
ROCO Robotik
TEC Technologie
HYD Hydraulik

Tab. 1-2: Übersicht Bibliotheken und Zielsystem MLC

Bibliothek	MLD mit Funktionspaket	Hinweis
MX_TechBase.library, Seite 17	Kein Funktionspaket erforderlich	
MX_TechMotionBase.library, Seite 33	Kein Funktionspaket erforderlich	
MX_TechMotion, Seite 163	MA	Für IndraLogic 1.x; ab MPx20 mit angepasstem Funktionsumfang
MX_TechRegi.library, Seite 307	MA	Nur für IndraLogic 1.x und Firmware < MPx18
MX_TechWinder.lib, Seite 491	MA	Für IndraLogic 1.x; ab MPx20 mit angepasstem Funktionsumfang
MX_TechCrossCutCrossSeal, Seite 623	MA	Für IndraLogic 1.x; ab MPx20 mit angepasstem Funktionsumfang

MA MLD Advanced

Tab. 1-3: Übersicht Bibliotheken und Zielsystem MLD



Für SercosDrive werden die Technologie-Funktionsbausteine (Wickler, Registerregler, Zugspannungsregler...) nicht unterstützt.

Hierzu ist ein IndraDrive mit Interpolation in der Steuerung oder im Antrieb erforderlich.

1.3 Besonderheiten von MLD

Allgemeines

MLD ist eine "Onboard"-Steuerung, die in der Hardware des IndraDrive-Steuerteils integriert ist. Die Firmware der SPS ist damit ein Teil der Antriebsfirmware. Die SPS-Entwicklungsumgebung ist ab der Antriebsfirmware MPx18 identisch zur MLC-Entwicklungsumgebung IndraLogic 2G.

Für eine Antriebsfirmware < MPx18 wird im Unterschied zur MLC die SPS-Entwicklungsumgebung IndraLogic 1.x. verwendet.



Bei MLD-Steuerungssystemen bis Firmwareversion MPx08 ist der nachfolgend verwendete E/A-Typ "VAR_IN_OUT CONSTANT" durch "VAR_IN_OUT" zu ersetzen.



Nicht alle Technologiebibliotheken sind in identischem Funktionsumfang für die MLD vorhanden. Mit Hilfe der Zielsystemtabellen für die Technologiebibliotheken und der Dokumentation der jeweiligen Funktionsbausteine kann dies überprüft werden.

Antriebsfirmware, Funktionspakete

Die Antriebsfirmware gibt es ab 07VRS (MPx 0x-Hardware) in folgenden Varianten (mit IndraLogic 1.x):

- MPC-Firmware: Mehrachs-MotionControl MLD-M für bis zu 8 Achsen
- MPH-Firmware: Einachssteuerung MLD-S für 1 Achse
- MPD-Firmware: Keine MLD-Funktionalität möglich
- MPB-Firmware: Einachssteuerung MLD-S für 1 Achse

Die Antriebsfirmware gibt es ab 18VRS (MPx 1x-Hardware) in folgenden Varianten (mit IndraLogic 2G):

- MPC-Firmware: Mehrachs-MotionControl MLD-M für bis zu 11 Achsen
- MPM-Firmware: Keine MLD-Funktionalität möglich
- MPB-Firmware: Einachssteuerung MLD-S für 1 Achse
- MPE-Firmware: Keine MLD-Funktionalität möglich

Funktionspakete

Die Funktionalität der Antriebsfirmware ist in mehrere Funktionspakete unterteilt. Die Funktionspakete müssen der Applikation entsprechend bestellt und lizenziert werden. Die benötigten Funktionspakete sind dann freigeschaltet.



Für MLD muss das Funktionspaket "MLD" freigeschaltet sein.

Für erweiterte Technologiebausteine muss das Zusatzpaket "MLD Advanced" (abgekürzt "MA") freigeschaltet sein.



Das Zusatzpaket "MLD Advanced" ist

- für die MPx 0x-Hardware: nur für MPC-Firmware und MPH-Firmware verfügbar (nicht für MPB-Firmware)
- für die MPx 1x-Hardware: für MPC-Firmware und MPB-Firmware verfügbar

Technologiebibliotheken, die das Zusatzpaket "MLD Advanced" voraussetzen, sind z. B.:

- MX_TechWinder
- MX_TechRegi

Mehrachssteuerung MLD-M

Für die Mehrachssteuerung MLD-M wird die Antriebsfirmware MPC benötigt. Für diese Firmware wird ein gesondertes Steuerteil benötigt (siehe Beschreibung Antriebsfirmware).

Zusätzlich muss in der SPS-Konfiguration eingestellt werden, dass die SPS die "permanente Kontrolle" über den Antrieb hat (P-0-1367 "SPS-Konfiguration" Bit 4 gesetzt bzw. Dialogeinstellung für SPS-Konfiguration).



Für die MPC-Firmware wird ein gesondertes Steuerteil benötigt.

Software

IndraWorks 14VRS für MLD

Besondere Parametrierung für die Steuerung MLD

Die "AxisData-Struktur" des Antriebes muss parametrierung sein ("P-0-1367 SPS Konfiguration" Bit 6 ist gesetzt bzw. Dialog SPS-Konfiguration).

Für die Mehrachssteuerung MLD-M muss zusätzlich in der SPS-Konfiguration eingestellt werden, dass die SPS die "permanente Kontrolle" über den Antrieb hat ("P-0-1367 SPS Konfiguration" Bit 4 gesetzt bzw. Dialog für SPS-Konfiguration).

Bei der Steuerung MLD wird die virtuelle Achse mit Hilfe eines Leitachsgenerators realisiert.

Konfiguration des virtuellen Leitachsgenerators:

Die Konfiguration des virtuellem Leitachsgenerators umfasst 3 Parametrierungen:

- Betriebsarten der virtuellen Leitachse
Positionierbetrieb einstellen

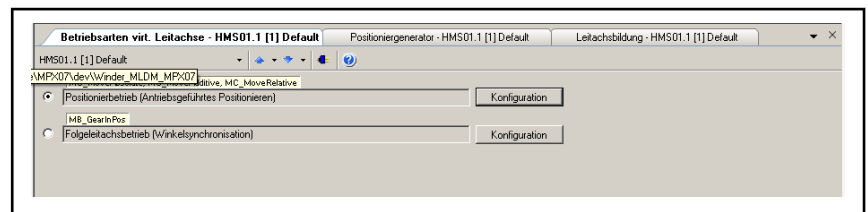


Abb. 1-2: Betriebsarten virtuelle Leitachse

- Leitachsbildung
Parameter "P-0-0758 Virtuelle Leitachse, Lageistwert" eintragen

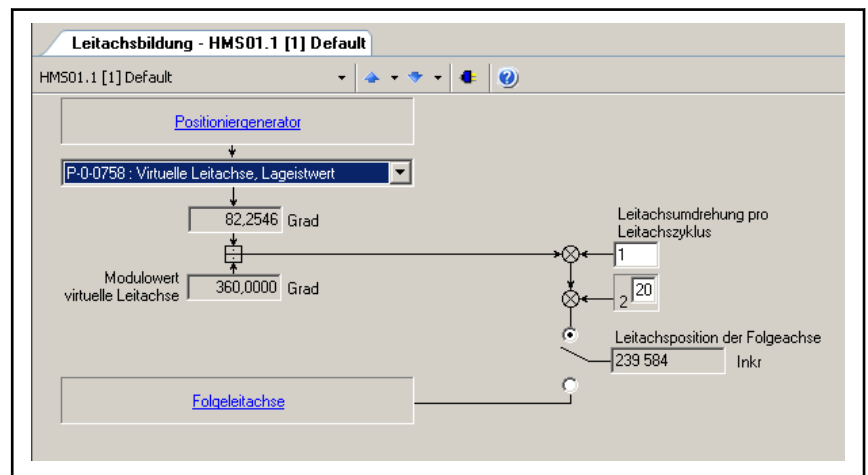


Abb. 1-3: Leitachsbildung

- Positioniergenerator
Positioniersollwert aktivieren

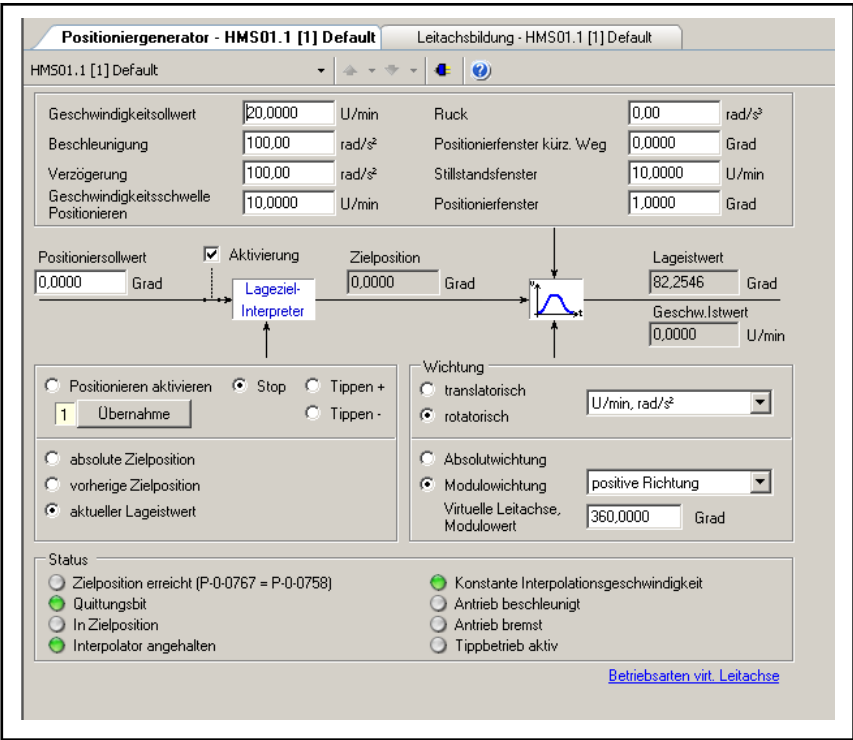


Abb. 1-4: Positioniergenerator

Für eine kontinuierliche Geschwindigkeitsvorgabe "Tippen+" und "Tippen-" einstellen und den Leitachsgenerator mittels des Button "Übernahme" starten.

Achsreferenzen bei der Steuerung MLD:

Die Achsreferenzen für die Steuerung MLD sind in der Bibliothek MX_Base in der globalen Variablenliste GVL_Base eingetragen.

Ab MPx18 (mit IndraLogic 2G) gelten folgende Achsreferenzen:

VAR_GLOBAL CONSTANT GVL_Base						
Name	Datentyp	Geerbt von	Adresse	Initialwert	Kommentar	
lvcAxis	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := NO_OBJECT)	definition for no axis	
Axis1	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_1)	local axis 1 use as 'Axis' or 'Slave' or as 'Master' (MLD-M)	
Axis2	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_2)	remote axis 2 use as 'Axis' or 'Slave' or as 'Master' (MLD-M)	
Axis3	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_3)	remote axis 3 use as 'Axis' or 'Slave' or as 'Master' (MLD-M)	
Axis4	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_4)	remote axis 4 use as 'Axis' or 'Slave' or as 'Master' (MLD-M)	
Axis5	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_5)	remote axis 5 use as 'Axis' or 'Slave' or as 'Master' (MLD-M)	
Axis6	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_6)	remote axis 6 use as 'Axis' or 'Slave' or as 'Master' (MLD-M)	
Axis7	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_7)	remote axis 7 use as 'Axis' or 'Slave' or as 'Master' (MLD-M)	
Axis8	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_8)	remote axis 8 use as 'Axis' or 'Slave' or as 'Master' (MLD-M)	
Axis9	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_9)	remote axis 9 use as 'Axis' or 'Slave' or as 'Master' (MLD-M)	
Axis10	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := AXIS_10)	remote axis 10 use as 'Axis' or 'Slave' or as 'Master' (MLD-M)	
VmAxis1	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := VMA_1)	virtual axis 1 use as 'Axis' or 'Slave' (internal) or as 'Master' (MLD-M)	
RmAxis1	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := RMA_1)	real master axis P-0-0761 use as 'Master'	
RmAxisInt	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := RMA_INT)	real master axis P-0-0052 use as 'Master'	
LinkAxis1	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := MA_LINK_1)	external master axis 1 use as 'Master' (MLD-M)	
LinkAxisInt	AXIS_REF			STRUCT(CntrlNo := LOCAL_CNTRL, AxisNo := MA_LINK_INT)	external master axis P-0-0053 or P-0-0767 use as 'Master'	
DV_Axis	ARRAY [AXIS_1..AXIS_10] OF MX_DirectVarAxis				Direct variable structure for 'all' local axis	

Abb. 1-5: Achsreferenzen bei der Steuerung MLD (IL 2G)



Die AxisData-Struktur ("AxisData[AxisNumber]") wird bis zur Referenz "VmAxis1" (Nummer 20) unterstützt!

Für höhere Nummern bzw. Referenzen (ab "VmAxisInt", Nummer 21) kann nicht mit der AxisData-Struktur gearbeitet werden. In diesem Fall müssen die benötigten Größen über Einzelparameter gelesen werden!

Auch die Verwendung des AchsInterface ist dann nicht möglich!



Für die Technologiebausteine wird als Achsreferenz für die Masterachse empfohlen:

- Für eine virtuelle Leitachse die Achsreferenz "**VmAxis1**" verwenden.

Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 20 (VMA_1) eingestellt werden.

- Für eine reale Leitachse die Achsreferenz "**RmAxis1**" verwenden.

Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 23 (RMA_1) eingestellt werden.

Zusätzlich muss in diesem Fall noch die logische Adresse des Messgebers im Parameter "P-0-1820.0.2 CCD: Quelle des Messgeber-Lageistwertes" eingetragen werden.

Detaillierte Informationen können der Funktionsbeschreibung MLD entnommen werden.

1.4 Gebrauch der Sicherheitshinweise

1.4.1 Aufbau der Sicherheitshinweise

Die Sicherheitshinweise sind wie folgt aufgebaut:

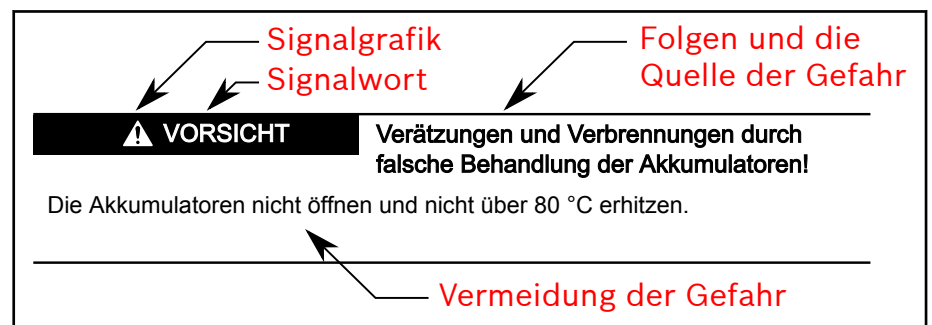


Abb. 1-6: Aufbau der Sicherheitshinweise

1.4.2 Erläuterung der Signalwörter und der Signalgrafik

Die Sicherheitshinweise in der vorliegenden Dokumentation beinhalten bestimmte Signalwörter (Gefahr, Warnung, Vorsicht, Hinweis) und gegebenenfalls eine Signalgrafik (nach ANSI Z535.6-2006).

Das Signalwort soll die Aufmerksamkeit auf den Sicherheitshinweis lenken und bezeichnet die Schwere der Gefährdung.

Die Signalgrafik (Warndreieck mit Ausrufezeichen), welche den Signalwörtern Gefahr, Warnung und Vorsicht vorangestellt wird, weist auf Gefährdungen für Personen hin.

⚠ GEFAHR

Bei Nichtbeachtung dieses Sicherheitshinweises **werden** Tod oder schwere Körpervverletzung eintreten.

⚠ WARNUNG

Bei Nichtbeachtung dieses Sicherheitshinweises **können** Tod oder schwere Körpervverletzung eintreten.

⚠ VORSICHT

Bei Nichtbeachtung dieses Sicherheitshinweises können mittelschwere oder leichte Körpervverletzung eintreten.

HINWEIS

Bei Nichtbeachtung dieses Sicherheitshinweises können Sachschäden eintreten.

1.4.3 Verwendete Symbole

Fingerzeige werden wie folgt dargestellt:



Dies ist ein Hinweis.

Tipps werden wie folgt dargestellt:



Dies ist ein Tipp.

1.4.4 Erläuterung der Signalgrafik auf dem Gerät



Wenn dieses Symbol am Gerät angebracht ist, beachten Sie unbedingt die Dokumentation zu dem Gerät. In der jeweiligen Dokumentation finden Sie die Art der Gefährdung sowie die notwendigen Schritte zur Vermeidung der Gefährdung.

1.5 Erforderliche und ergänzende Dokumentationen XLC/MLC

Dokumentationstitel mit Typenschlüssel und Materialnummer

Systemübersicht
IndraLogic XLC IndraMotion MLC 14VRS Systemübersicht DOK-XLCMLC-SYSTEM**V14-PRRS-DE-P, R911341503 Diese Dokumentation stellt eine Übersicht der möglichen Hardware-/Softwarekomponenten der Automationssysteme IndraLogic XLC und IndraMotion MLC in der genannten Version dar. Die Dokumentation dient als Auswahlhilfe zur Zusammenstellung eines Systems.
<i>Tab. 1-4: XLC/MLC-Dokumentationsübersicht - Systemübersicht</i>
Erste Schritte
IndraMotion MLC 14VRS Erste Schritte DOK-MLC***-F*STEP**V14-CORS-DE-P, R911341516 Diese Dokumentation beschreibt die "ersten Schritte" mit der IndraMotion MLC und RobotControl. Beschrieben werden die Hard- und Softwarevoraussetzungen und das Erstellen eines Projekts.
IndraLogic XLC 14VRS Erste Schritte DOK-XLC***-F*STEP**V14-CORS-DE-P, R911341488 Diese Dokumentation beschreibt die "ersten Schritte" mit der IndraLogic XLC. Beschrieben werden die Hard- und Softwarevoraussetzungen und das Erstellen eines Projekts.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Projektkonvertierung DOK-XLCMLC-PROCONV*V14-APRS-DE-P, R911341493 Diese Dokumentation beschreibt die Projektkonvertierung der IndraMotion MLC.

Tab. 1-5: XLC/MLC-Dokumentationsübersicht - Erste Schritte

Engineering
IndraWorks 14VRS Software-Installation DOK-IWORKS-SOFTINS*V14-CORS-DE-P, R911344285 Die Dokumentation beschreibt den Installationsprozess von IndraWorks.
IndraWorks 14VRS Engineering DOK-IWORKS-ENGINEE*V14-APRS-DE-P, R911343564 Diese Dokumentation beschreibt die Verwendung von IndraWorks, in das sich die Rexroth Engineering-Tools integrieren. Beschrieben werden das Arbeiten mit IndraWorks und die Bedienung der Oszilloskop-Funktion.
IndraLogic XLC IndraMotion MLC 14VRS Funktionsbeschreibung DOK-XLCMLC-FUNC****V14-APRS-DE-P, R911341699 Die Dokumentation dient der Beschreibung der Wizards, Kontextmenüs, Dialoge, Steuerungsinbetriebnahme, Gerätekonfiguration, Funktionalitäten der IndraMotion MLC.

Tab. 1-6: XLC/MLC-Dokumentationsübersicht - Engineering

Diagnose und Service
IndraLogic XLC IndraMotion MLC 14VRS Diagnosen DOK-XLCMLC-DIAG****V14-RERS-DE-P, R911341480 Diese Dokumentation enthält alle in den Steuerungssystemen IndraLogic XLC und IndraMotion MLC implementierten Steuerungsdiagnosen.
IndraLogic XLC IndraMotion MLC 14VRS Parameter DOK-XLCMLC-PARAM***V14-RERS-DE-P, R911341478 Diese Dokumentation beschreibt die Parameter der Systeme XLC/MLC, sowie das Wechselspiel zwischen Parametrieren und Programmieren. Beschrieben werden die Achs-Parameter, die Steuerungs-Parameter, die Kinematik-Parameter, die Messtaster- und die Nockenschaltwerks-Parameter.
IndraLogic XLC IndraMotion MLC 14VRS Inbetriebnahme DOK-XLCMLC-STARTUP*V14-CORS-DE-P, R911341500 Das vorliegende Handbuch beschreibt die Inbetriebnahme- und Servicearbeitsschritte für die Systeme IndraLogic XLC und IndraMotion MLC. Es beinhaltet Checklisten für häufig durchzuführende Arbeiten und eine ausführliche Beschreibung der Arbeitsschritte.

Diagnose und Service
WebAssistant DOK-GENERL-WEB*ASSIST*-APRS-DE-P, R911381468 Diese Dokumentation beschreibt den WebAssistant. Der WebAssistant ist ein webbasiertes Diagnosewerkzeug, das zum Zugriff auf ein Steuerungssystem über eine Ethernet-Hochgeschwindigkeitsverbindung dient. Der WebAssistant ermöglicht OEMs, Endanwendern und Kundenbetreuern von überall auf ein System zuzugreifen und Ferndiagnosen zu stellen. Voraussetzung ist ein PC mit Internet Explorer Version 11 oder neuer, Firefox Version 45 oder neuer oder Google Chrome Version 54 oder neuer. Die folgenden Steuerungsvarianten werden unterstützt: • IndraMotion MLC L45/L65/L75/XM21/XM22/XM42/VPx • IndraLogic XLC L45/L65/L75/VEP
IndraWorks IndraMotion Service Tool DOK-IWORKS-IMST*****-APRS-DE-P, R911341382 Diese Dokumentation beschreibt das IndraMotion Service Tool (IMST). Das IMST ist ein webbasiertes Diagnosewerkzeug, das zum Zugriff auf ein Steuerungssystem über eine Ethernet-Hochgeschwindigkeitsverbindung dient. Das IMST ermöglicht OEMs, Endanwendern und Kundenbetreuern von überall auf ein System zuzugreifen und Ferndiagnosen zu stellen. Voraussetzung ist ein PC mit Internet Explorer Version 8 oder Firefox 3.5 oder eine höhere Version. Die folgenden Steuerungs-Varianten werden unterstützt: • IndraMotion MLC L45/L65/L75/XM21/XM22/XM42/VPx • IndraLogic XLC L45/L65/L75

Tab. 1-7: XLC/MLC-Dokumentationsübersicht - Diagnose und Service

Firmware
Rexroth IndraDrive MPx-20, Funktionen ¹⁾ DOK-INDRV*-MP*-20VRS**-AP01-DE-P, R911345607 Diese Dokumentation beschreibt alle funktionellen Eigenschaften der IndraDrive-Firmware in den Varianten MPB-20, MPM-20, MPC-20 und MPE-20.
Rexroth IndraDrive MPx-16 bis MPx-20 und PSB, Parameter DOK-INDRV*-GEN1-PARA**-RERS-DE-P, R911328650 Diese Dokumentation enthält die Beschreibung aller, in der Firmware für Antriebsregelgeräte der IndraDrive-Familie implementierten Parameter. Sie unterstützt die Parametrierung der Antriebsregelgeräte.
Rexroth IndraDrive MPx-16 bis MPx-20 und PSB, Diagnosen DOK-INDRV*-GEN1-DIAG**-RERS-DE-P, R911326539 Diese Dokumentation enthält die Beschreibung aller in den folgenden Firmwares implementierten Diagnosen: • Antriebsregelgerät-Firmwares MPx-16 bis MPx-20 • Firmwares PSB der Versorgungsgeräte des Typs "HMU05" und "KMOV3" • Firmwares der Versorgungsgeräte des Typs "HMOV1". Sie unterstützt den Maschinenbediener und den Anlagenprogrammierer bei der Fehlerbehebung.
Hardware
Rexroth IndraDrive Steuerteile CSB02, CSE02, CSH02, CDB02 DOK-INDRV*-Cxx02*****-PRRS-DE-P, R911338961
Rexroth IndraDrive Cs Antriebssysteme mit HCS01 DOK-INDRV*-HCS01*****-PRRS-DE-P, R911322209

Zu dieser Dokumentation

Rexroth IndraDrive Mi Antriebssysteme mit KCU02, KSM02, KMS02/03, KMV03

DOK-INDRV*-KCU02+KSM02-PRxx-DE-P, R911335702

Rexroth IndraDrive ML Antriebssysteme mit HMU05

DOK-INDRV*-Hxx05*****-PRxx-DE-P, R911344278

1) in Vorbereitung
 Tab. 1-8: Übersicht der Dokumentationen für Antriebsregelgeräte

SPS**IndraWorks 14VRS SPS-Programmiersystem IndraLogic 2G**

DOK-IWORKS-IL2GPRO*V14-APRS-DE-P, R911343570

Diese Dokumentation beschreibt das SPS-Programmierzug IndraLogic 2G und seine Verwendung. Beschrieben werden die Basisverwendung, Erste Schritte, die Visualisierung, Menüpunkte und die Editoren.

IndraWorks 14VRS Basis-Bibliotheken IndraLogic 2G

DOK-IL*2G*-BASLIB**V14-LIRS-DE-P, R911343918

Das vorliegende Handbuch beschreibt die systemübergreifenden SPS-Bibliotheken.

IndraLogic XLC IndraMotion MLC 14VRS PLCopen-Bibliotheken

DOK-XLCMLC-FUNLIB**V14-LIRS-DE-P, R911341490

Diese Dokumentation beschreibt die Funktionsbausteine, Funktionen und Datentypen der Bibliotheken RIL_CommonTypes, ML_Base und ML_PLCopen für die IndraLogic XLC/IndraMotion MLC. Die Dokumentation enthält ferner die Fehlerreaktion der Bausteine.

Tab. 1-9: XLC/MLC-Dokumentationsübersicht - SPS

Feldbusse**IndraWorks 14VRS Feldbusse**

DOK-IWORKS-FB*****V14-APRS-DE-P, R911341484

Die Dokumentation dient der Beschreibung der von den Systemen IndraLogic XLC, IndraMotion MLC und IndraMotion MTX unterstützten Feldbus- und lokalen Peripherieanschlüssen. Der Schwerpunkt dieser Dokumentation liegt in der Konfiguration, Parametrierung, Inbetriebnahme und Diagnose der verschiedenen Peripherieanschlüssen. Sie ist die Grundlage für die Online-Hilfe.

IndraWorks 14VRS Feldbus-Bibliotheken

DOK-IWORKS-FB*LIB**V14-LIRS-DE-P, R911343574

Das vorliegende Handbuch beschreibt die Bibliotheken der Feldbusse: RIL_ProfibusDP_02, RIL_ProfibusDP Slave, RIL_ProfinetIO, RIL_ProfinetIODevice, RIL_EtherNetIPAdapter, RIL_MappingList, RIL_SERCOSIII, RIL_Inline, einschließlich deren Diagnosen, sowie die Fehlerreaktion der Bausteine.

Rexroth IndraWorks 12VRS FDT-Container

DOK-IWORKS-FDT*CON*V12-APRS-DE-P, R911334397

Diese Dokumentation beschreibt die FDT-Container-Funktionalität von IndraWorks. Beschrieben wird das Aktivieren der Funktionalität im Projekt und das Arbeiten mit DTMs.

Sercos Systemhandbuch für I/O-Geräte

DOK-CONTRL-ILS3*****-APxx-DE-P, R911333511

Diese Dokumentation beschreibt die Projektierung, Parametrierung, Inbetriebnahme und Diagnose von I/O-Geräten mit Sercos Schnittstelle.

xx entsprechender Ausgabestand
 Tab. 1-10: XLC/MLC-Dokumentationsübersicht - Feldbusse

HMI
IndraWorks 14VRS HMI DOK-IWORKS-HMI*****V14-APRS-DE-P, R911343567 Diese Dokumentation beschreibt die Funktionen, die Konfiguration und die Bedienung der Bedienoberflächen IndraWorks HMI Engineering und IndraWorks HMI Operation.
IndraWorks 14VRS WinStudio 7.4 DOK-IWORKS-WINSTUD*V14-APRS-DE-P, R911341584 Dieses "Benutzerhandbuch und Technische Referenz" soll Sie dabei unterstützen, mit Ihrer Software "Rexroth WinStudio" optimale Ergebnisse zu erzielen. Das Dokument bietet technische Informationen und Schritt-für-Schritt-Anleitungen zum Anlegen von webfähigen HMI/SCADA-Programmen.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS HMI-Anbindung DOK-XLCMLC-HMI*****V14-APRS-DE-P, R911341496 Dieses Dokument beschreibt die von der IndraLogic XLC und IndraMotion MLC unterstützten Visualisierungssysteme sowie deren Anbindung.

Tab. 1-11: XLC/MLC-Dokumentationsübersicht - HMI

Technologie
IndraLogic XLC IndraMotion MLC 14VRS Generic Application Template DOK-XLCMLC-TF*GAT**V14-APRS-DE-P, R911341486 Dieses Handbuch stellt dem IndraLogic SPS-Programmierer ein strukturiertes Template, welches zum Hinzufügen und Bearbeiten des SPS-Programmierscodes verwendet werden kann, zur Verfügung. Beschrieben werden das Template selbst, der Template Wizard und Beispielapplikationen.
IndraMotion MLC 14VRS Technologie-Bibliotheken DOK-MLC***-TF*LIB**V14-LIRS-DE-P, R911341510 Diese Dokumentation beschreibt die Funktionsbausteine, Funktionen und Datentypen der Bibliotheken "ML_TechInterface.library", "ML_TechMotion.library", "RMB_TechCam.library" und "ML_TechBase.library" sowie Bibliotheken zur Wicklerfunktionalität, Registerreglerfunktionalität und Querschneiderfunktionalität.
IndraWorks 14VRS Energy Efficiency Management DOK-IWORKS-4EE*****V14-APRS-DE-P, R911339228 Diese Dokumentation beschreibt die HMI-Anbindung des Energiemanagement-Systems und die Funktionsbausteine, Funktionen und Datentypen der Bibliothek "RIL_4EE ". Die Dokumentation enthält ferner die Fehlerreaktion der Bausteine.
Rexroth IndraMotion MLC 14VRS RegisterControl (Bibliothek) DOK-MLC***-REGI*CO*V14-LIRS-DE-P, R911341508 Diese Dokumentation beschreibt die Ein- und Ausgänge der einzelnen Funktionsbausteine und gibt Hinweise für deren Verwendung.
IndraMotion MLC 14VRS RegisterControl (Anwendungsbeschreibung) DOK-MLC***-REGI*CO*V14-APRS-DE-P, R911341506 Dieses Dokument beschreibt die Anwendung der integrierten Registerregelung für eine Tiefdruckmaschine. Beschrieben werden die Komponenten des Markenstromsensors, die HMI-Anwendung und die Fehlerbehebung. Diese Anleitung dient dazu, die Registerregelung zu bedienen und auf Fehler zu reagieren sowie Diagnosen abzufragen. Die Zielgruppe für diese Dokumentation sind Maschineneinrichter und Maschinenbediener.
Rexroth IndraMotion MLC 14VRS Anwendung Wicklerfunktionen DOK-MLC***-TF*WIND*V14-APRS-DE-P, R911341512 Diese anwendungsbezogene Systemdokumentation beschreibt die Anwendung der Wickler-Technologiefunktionen.

Zu dieser Dokumentation

Technologie
Rexroth IndraWorks 13VRS CamBuilder DOK-IWORKS-CAMBUIL*V13-APRS-DE-P, R911336290 Diese Dokumentation dient zur Beschreibung der Grundlagen und der Bedienung des CamBuilder, des Werkzeugs zur Kurvenscheibenbearbeitung.
IndraMotion MLC 14VRS Robot-Control V2 DOK-MLC***-ROCO****V14-RERS-DE-P, R911341586 Die Dokumentation informiert über die Robot-Control V2. Die Programmierung über PLCopen steht hierbei im Vordergrund. Beschrieben werden die Programmstruktur, Variablen, Funktionen, Bewegungsanweisungen sowie die notwendigen Parameter des Systems. Die Dokumentation enthält ferner die Beschreibung der Bibliothek "ML_Robot". Robot-Control V2 wird unterstützt durch ein Kinematik-Interface. Die Komponenten dieses Interfaces werden in der Bibliothek "ML_KinTech" beschrieben.

Tab. 1-12: XLC/MLC-Dokumentationsübersicht - Technologie

Open Core Engineering - OCE
IndraLogic XLC IndraMotion MLC 14VRS Erste Schritte MLPI (Onlinehilfe) DOK-XLCMLC-MLPI****V14-CORS-EN-E, R911341033 Dieses Dokument beschreibt die Installation und Inbetriebnahme der MLPI-Schnittstelle.
IndraLogic XLC IndraMotion MLC 14VRS Automation Interface (Onlinehilfe) DOK-XLCMLC-AUT*INT*V14-APRS-EN-E, R911341499 Diese Dokumentation beschreibt den script-basierten Zugriff auf IndraWorks-Projektdateien über die Schnittstelle des Automation Interfaces.

Tab. 1-13: XLC/MLC-Dokumentationsübersicht - Open Core Engineering - OCE

SafeLogic
IndraWorks SafeLogic 14VRS Die ersten Schritte DOK-IWORKS-SL*STEP*V14-CORS-DE-P, R911341519 Dieses Dokument beschreibt die Erstinbetriebnahme des Safety-Funktionsmoduls und dessen externer Safety-Peripherie. Am Beispiel eines Projektes werden alle notwendigen Schritte zur Inbetriebnahme einer einfachen Safety-Anwendung durchgeführt.
IndraLogic XLC IndraMotion MLC 14VRS SafeLogic Systemübersicht DOK-XLCMLC-SL**SYS*V14-PRRS-DE-P, R911341695 Dieses Dokument beschreibt die Projektierung, die Erstellung und die Inbetriebnahme von sicherheitstechnischen Vorrichtungen.
IndraWorks 14VRS SafeLogic - Projektkonfiguration, Stand 1.2.0.0. DOK-IWORKS-SL**PRJ*V14-APRS-DE-P, R911341693 Dieses Dokument beschreibt die Projektierung von Safety-Anwendungen.
IndraControl CFL01.1-F1 Safety-Funktionsmodul DOK-CONTRL-SL**FM*****-ITRS-DE-P, R911336575 Dieses Dokument liegt der Hardware bei und beschreibt die Inbetriebnahme, die Installation, die Außerbetriebnahme und Entsorgung der Hardware.

Tab. 1-14: XLC/MLC-Dokumentationsübersicht - SafeLogic

Hardware
Rexroth IndraControl L45/L65/L85 Steuerung DOK-CTRL-ICL45L65L85-PRxx-DE-P, R911332115 Diese Dokumentation dient der Beschreibung der Steuerungen IndraControl L45/L65/L85.
Rexroth IndraControl Lxx 14VRS Funktionsmodule DOK-CTRL-FM*LXX**V14-APRS-DE-P, R911341582 Diese Dokumentation dient der Beschreibung aller Funktionsmodule der Lxx Steuerungen inkl. Engineering und Diagnose.
IndraControl XM21, XM22 Steuerungen DOK-CTRL-IC*XM2****-ITRS-DE-P, R911340666 Diese Dokumentation dient der Beschreibung der Steuerungen IndraControl XM21/XM22.
IndraControl XM42 Steuerungen DOK-CTRL-XM4X*CTRL**-ITRS-DE-P, R911345565 Diese Dokumentation dient der Beschreibung der Steuerungen IndraControl XM42.
IndraControl XFE 01.1 Erweiterungsmodule Profibus, RT-Ethernet, Sercos, CAN DOK-CTRL-XFE**EXTMOD-ITRS-DE-P, R911345569 Diese Dokumentation dient der Beschreibung aller Erweiterungsmodule der XM-Steuerungen.

xx entsprechender Ausgabestand
Tab. 1-15: XLC/MLC-Dokumentationsübersicht - Hardware

Hydraulik
IndraMotion MLC 14VRS Hydraulische Funktionen DOK-MLC***-TF*HMOT*V14-LIRS-DE-P, R911341905 Diese Dokumentation beschreibt die Funktionsbausteine, Funktionen und Datentypen der Bibliotheken MH_TechHydrBase und MH_TechHydrMotion. Die Dokumentation enthält ferner die Fehlerreaktion der Bausteine.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Sequentielle Programmierung DOK-XLCMLC-SEQPROG*V14-LIRS-DE-P, R911341908 Diese Dokumentation beschreibt die Möglichkeiten der Sequentiellen Programmierung, auch in Verbindung mit GAT, und die Bibliothek MH_HydrControlProg.

Tab. 1-16: XLC/MLC-Dokumentationsübersicht - Hydraulik

1.6 Bezeichnungen und Abkürzungen

Begriff	Erklärung
FUP	Funktionsplan, Programmiersprache nach IEC 61131-3
IndraDrive	Antriebssteuergerät
XLC	kompaktes SPS-System mit integrierter Motion-Funktionalität, Basis-Funktionspaket PLC
MLC	kompaktes Motion-Logic-System mit Motion-, Robot- und Logic-Control-Funktionalität, gestaffelt nach Funktionspaketen
MLD	Antriebsbasiertes Motion-Logic-System
IndraWorks	Projektierungs- und Inbetriebnahmewerkzeug von Bosch Rexroth
KOP	Kontaktplan, Programmiersprache nach IEC 61131-3
MA	MLD Advanced
MLD-M	Motion Logic Drive (multi drive)

MLD-S	Motion Logic Drive (single drive)
PLCOpen	ist eine 1992 gegründete, internationale hersteller- und produktunabhängige Interessensgemeinschaft von Steuerungsherstellern, Softwarehäusern und Instituten. Dem SPS-Standard IEC 61131-3 verpflichtend werden in verschiedenen Arbeitsgruppen (Technical Committes) Standards definiert, die eine erhebliche Effizienzsteigerung in der Entwicklung von Applikationssoftware ermöglichen
PLS	Programmable Limit Switch (Nockenschaltwerk)
PWM	Pulsweitenmodulation
ST	Strukturierter Text, Programmiersprache nach IEC 61131-3
SYNAX	Automatisierungssystem für Druck- und Verarbeitungsmaschinen
Tech-FB	Technologie-Funktionsbaustein
VisualMotion	programmierbares Mehrachsen-Bewegungssteuerungssystem
Zielsystem	Steuerung (z. B. MLC)

Tab. 1-17: Verwendete Bezeichnungen und Abkürzungen

1.7 Kundenfeedback

Anregungen, Wünsche oder Verbesserungen von unseren Kunden haben bei uns einen hohen Stellenwert. Senden Sie uns Ihre Anmerkungen zu den Dokumentationen per E-Mail an Feedback.Documentation@boschrexroth.de. Sie können direkt im elektronischen PDF-Dokument Kommentare einfügen und uns die PDF-Datei zusenden.

2 ML_TechBase.library // MX_TechBase.library

2.1 Einführung und Übersicht

Allgemeines Technologie-Funktionsbausteine (Tech-FB) erweitern die Grundfunktionalität der Zielsysteme MLC/XLC und MLD und stellen applikationsspezifische Funktionalitäten zur Verfügung. Die hier beschriebene Bibliothek beinhaltet Basis-Funktionalitäten die für viele verschiedene Anwendungen zum Einsatz kommen, aber auch einzeln verwendet werden können.

Die hier beschriebenen Funktionsbausteine werden über die Bibliothek "ML_TechBase" für MLC/XLC zur Verfügung gestellt.

Diese Dokumentation beschreibt die Funktionalität, sowie die Ein- und Ausgänge der Technologie-Funktionsbausteine.



Die Bibliothek benötigt das Grundpaket PLC der Steuerung.

Voraussetzungen Technologie-Funktionsbausteine der Bibliothek "ML_TechBase" benötigen die Firmware-Unterstützung des Zielsystems. Spezifische Voraussetzungen der Technologie-Funktionsbausteine sind im Kapitel der entsprechenden Funktionsbausteine dokumentiert.

Alias In der Bibliothek befindet sich im Verzeichnis "Global variables" auch ein Eintrag "FP_Aliases". Dabei handelt es sich um Bezeichner, hinter denen Parameter stehen. Die Parameter sind dabei systemabhängig. Der Anwender sollte bevorzugt diese Bezeichner verwenden, da er damit unabhängig vom gewählten Zielsystem ist.

So wird beispielsweise der Bezeichner "FP_MODULO"

auf einer MLC/XLC auf den Achsparameter A-0-0045, Modulowert gesetzt,

auf einem MLD-System auf den Sercos-Parameter S-0-0103, Modulowert.

Auf Applikationsebene kann man damit das verwendete System besser abstrahieren und allgemein gültigen Code schreiben.

Anwendungsgebiete und Funktionsbausteine von ML_TechBase

Anwendungsgebiet (Ordner in der Bibliothek)	
Funktionsbausteine	Beschreibung
IMC-Interface (ImcInterface)	
IMC-Interface, Seite 19	Kommando- und Diagnoseschnittstelle
Trigonometrische Funktionen (Trigonometric Functions)	
ML_DegToInc, Seite 24	Wandelt Winkel vom Gradmaß in entsprechenden Inkrementwert
ML_DegToRad, Seite 25	Konvertiert Winkel im Gradmaß in Winkel in Bogenmaß inkl. Modulo
ML_IncToDeg, Seite 25	Wandelt einen in Inkrementen vorgegebenen Winkel ins Gradmaß
ML_IncToRad, Seite 26	Konvertiert einen in Inkrementen vorgegebenen Winkel in das entsprechende Bogenmaß
ML_RadToDeg, Seite 27	Konvertiert einen Winkel im Bogenmaß in den entsprechenden Winkel im Gradmaß inkl. Modulo

ML_RadToInc, Seite 27	Wandelt einen im Bogenmaß gegebenen Winkel in den entsprechenden Inkrementwert
Hilfsfunktionen (Utilities/ Systeminfo)	
MB_GetSystemInfo, Seite 28	Überprüft Funktionspakete und Systemoptionen

Tab. 2-1: Funktionsbausteine in Bibliothek ML_TechBase

_PhaseOut Die nachstehend genannten Funktionsbausteine (nebst zugehöriger Datentypen) wurden aus organisatorischen Gründen in andere Bibliotheken verschoben.

Dies geschah teilweise unter Anpassung des Namens.

Aus Kompatibilitätsgründen steht der Baustein jedoch unter dem alten Namen noch zur Verfügung.

Neue Bibliothek: RIL_LoopControl ab 14.3.0.0

Ordner (neu)	Name (neu)	POU-Typ	Name (alt)
TemperatureController	IL_TempControlType01	FB	
	IL_TempControlType02	FB	
TwoPosCtrl	IL_TwoPosCtrlType01	FB	ML_TwoPosCtrlType01

Tab. 2-2: POU's, die von ML_TechBase nach RIL_LoopControl verschoben wurden

Neue Bibliothek: RIL_Uutilities ab 14.3.0.0

Ordner (neu)	Name (neu)	POU-Typ	Name (alt)
Adjustment	IL_ContinuousAdjustType01	FB	MB_ContinuousAdjustType01
	IL_ContinuousAdjustType02	FB	MB_ContinuousAdjustType02
	IL_IncrementalAdjustType01	FB	MB_IncrementalAdjustType01
Mathematical Functions	IL_InterpolationLinear	FB	ML_InterpolationLinear
	IL_MaxValue	FB	ML_MaxValue
Utilities			
Logic	IL_LinkedList	FB	
Modulo	IL_GetModuloType01	FB	MB_GetModuloType01
	IL_GetModuloType02	FB	MB_GetModuloType02
	IL_GetModuloType03	FB	MB_GetModuloType03
	IL_GetModuloType04	FB	MB_GetModuloType04

Tab. 2-3: POU's, die von ML_TechBase nach RIL_Uutilities verschoben wurden

Neue Bibliothek: ML_TechMotionBase ab 14.3.0.0

Ordner (neu)	Name (neu)	POU-Typ	Name (alt)
AxisBackupRestore	ML_AxisBackup	FB	
	ML_AxisRestore	FB	
PhaseLimitSwitch	MC_DigitalCamSwitch	FB	
TouchProbe	MB_AbortTrigger	FB	
	MB_InitTouchProbe	FB	
	MB_TouchProbe	FB	
	MB_TouchProbeContinuous	FB	
	MB_WriteExpectWindow	FB	
LoadCamData	MB_LoadCamData	FB	
Utilities/CyclicData	MB_AllocCyclicParameter	FB	
	MB_FreeCyclicParameter	FB	
	MB_GetCyclicChannelConfig	FB	
	MB_GetCyclicParameterHandle	FB	
	MB_ReadCyclicParameter	FB	
	MB_ReadCyclicRealParameter	FB	
	MB_WriteCyclicParameter	FB	
	MB_WriteCyclicRealParameter	FB	
	MB_GetLastCyclicParameterError	FB	

Tab. 2-4: POU's, die von ML_TechBase nach ML_TechMotionBase verschoben wurden

2.2 IMC-Interface

2.2.1 Einführung und Übersicht

Die Bibliotheken ML_TechBase für MLC-Steuerungen und das MX_ImcInterface für MLD-M-Antriebe bieten ein einfach zu verwendendes Interface für das System an.

Das Interface wird als IMC-Interface bezeichnet - (I)ndra (M)otion (C)ontrol Interface.

Funktionen des IMC-Interfaces

- wählt MLC und für MLD-M Betriebsmodus, P0 (Phase 0), P2 (Parametermodus), BB (Betriebsmodus)
- führt den Befehl "Clear All Errors" (alle Fehler löschen) aus
- führt "RetriggerOpMode" aus, um es den Benutzern zu ermöglichen, den gewählten Betriebsmodus erneut auszuführen. Der Benutzer kann

die Steuerung für den Betriebsmodus Imc-Interface zurücksetzen, wenn der Sercos-Ring getrennt wird

- Bei einer Phasenumschaltung über eine externe Instanz (z.B. IndraWorks) setzt das Imc-Interface den Ausgang "PassiveMode" und der angewählte Modus wird nicht angesteuert.

Mit "RetriggerOpMode" oder durch Zurückschalten in den angewählten Modus wird der "PassiveMode" wieder verlassen



Die nachstehend verwendeten Bibliotheksnamen sind systemabhängig:

- MLC-Steuerungssysteme verwenden die Bibliothek ML_TechBase, Grundpaket PLC
- MLD-M-Antriebssysteme verwenden die Bibliothek MX_ImcInterface.lib

Die folgende Tabelle listet das Benutzerinterface mit der Datenstruktur des Steuerungsinterface auf:

Benutzer-Interface	Typ	Beschreibung
ImcCtrl[]	MB_IMC_CONTROL_TYPE01, Seite 20	Steuerungsstruktur, die die Eingaben für Betriebsmodus, Fehler löschen und erneutes "Auslösen des Betriebsmodus" bereitstellt
ImcStatus[]	MB_IMC_STATUS_TYPE01, Seite 21	Statusstruktur, die Bestätigungen und Diagnoseinformationen für den Betriebsmodus umfasst

Tab. 2-5: Steuerungsinterface - Datenstrukturen Benutzerinterface

Um das Imc-Interface benutzen zu können, muss nur das Programm "MB_ImcInterface" zyklisch aufgerufen werden, z.B. in der PlcTask. Bei der Verwendung von GAT bzw. GAT ^{compact} ist der Programmaufruf bereits integriert.

Bibliothek ML_TechBase für MLC, Grundpaket PLC

MX_ImcInterface.lib für MLD-M

Abhängigkeit MX_AxisInterface.lib für MLD-M

2.2.2 MB_IMC_CONTROL_TYPE01

Kurzbeschreibung Auf die Datenstruktur "MB_IMC_CONTROL_TYPE01" wird über die Instanz **ImcCtrl[]** zugegriffen.

Unter Nutzung der Struktur [MB_IMC_ADMINISTRATION, Seite 20](#) wird ein Auftrag an die MotionControl-Komponente erteilt.

Name	Typ	Beschreibung
Admin	MB_IMC_ADMINISTRATION	Auftragserteilung

Tab. 2-6: STRUCT MB_IMC_STATUS_TYPE01

2.2.3 MB_IMC_ADMINISTRATION

Die Struktur "MB_IMC_ADMINISTRATION" unterstützt die Befehlsübergabe an die Steuerung (LogicControl ⇒ MotionControl).

Name	Typ	Beschreibung
_OpMode	MB_IMC_MODE_CONTROL, Seite 21	Auswahl der gewünschten Betriebsart (P0, P2, P4)
ClearError	BOOL	(Flankengesteuertes) Löschen aller Steuerungsfehler (C-0-1030)
RetriggerOpMode	BOOL	Erneutes "Auslösen des Betriebsmodus"
arCheckSercos	ARRAY [ML_MIN_SERCOS_IFC..ML_MAX_SERCOS_IFC] OF BOOL	Die Sercos-Phase der Onboard-Sercos-Schnittstelle (ML_LOCAL_SERCOS_IFC) soll in die Berechnung von "ImcStatus.Admin._OpModeAck " mit eingehen. Default = TRUE Array für zukünftige Erweiterungen mit mehreren Schnittstellen. Nur bei MLC vorhanden

Tab. 2-7: Elemente der Struktur "MB_IMC_ADMINISTRATION"

2.2.4 MB_IMC_MODE_CONTROL

Der Aufzählungstyp "MB_IMC_MODE_CONTROL" gibt die gewünschte Betriebsart vor.

Name	Wert	Beschreibung
IMC_MODE_CTRL_P0	0	Auswahl der Phase P0
IMC_MODE_CTRL_P2	2	Auswahl der Phase P2
IMC_MODE_CTRL_P4	4	Auswahl der Phase P4

Tab. 2-8: Elemente der Struktur "MB_IMC_ADMINISTRATION"

2.2.5 MB_IMC_STATUS_TYPE01

Kurzbeschreibung Auf die Datenstruktur "MB_IMC_STATUS_TYPE01" wird über die Instanz **ImcStatus[]** zugegriffen.

Sie enthält zum einen Diagnoseinformationen zum Auftrag, der an die MotionControl-Komponente erteilt wurde ([MB_IMC_ADMIN_STATUS, Seite 21](#)) und zum anderen allgemeine Diagnoseinformationen der Steuerung selbst ([MB_IMC_DIAGNOSIS, Seite 23](#)).

Name	Typ	Beschreibung
InOperation	BOOL	TRUE: ImcInterface arbeitet
Admin	MB_IMC_ADMIN_STATUS	Diagnose zum erteilten Auftrag
Diag	MB_IMC_DIAGNOSIS	Allgemeine Diagnose zur Steuerung
PassiveMode	BOOL	TRUE: externe Quelle steuert die Betriebsart

Tab. 2-9: STRUCT MB_IMC_STATUS_TYPE01

2.2.6 MB_IMC_ADMIN_STATUS

Die Struktur "MB_IMC_ADMIN_STATUS" enthält Status-Informationen zum aktuell laufenden Auftrag (MotionControl ⇒ LogicControl).

Name	Typ	Beschreibung
_OpModeAck	MB_IMC_MODE_STATUS	Bestätigung der gewählten Betriebsart P0..P2..BB. Nur bei MLC: Berechnung aus Steuerungsmodus und Sercos-Phase (wenn nicht in ImcCtrl.Admin.arCheckSercos[] deaktiviert.) MB_IMC_MODE_STATUS, Seite 22
ModeStatus_P0	BOOL	Bestätigung Imc-Modus 0
ModeStatus_P1	BOOL	Bestätigung Imc-Modus 1
ModeStatus_P2	BOOL	Bestätigung Imc-Modus 2
ModeStatus_P3	BOOL	Bestätigung Imc-Modus 3
ModeStatus_BB	BOOL	Bestätigung Imc-Modus BB
PwrUpModeAck	BOOL	Steuerung erreicht PowerUpTargetMode (C-0-0450)
PhaseSwitchActive	BOOL	Sercos-Phasen-Schaltung ist aktiv
arSercosPhase	ARRAY [ML_MIN_SERCOS_IFC..ML_MAX_SERCOS_IFC] OF MB_IMC_SERCOS_PHASE	Die Sercos-Phase der Onboard-Sercos-Schnittstelle (ML_LOCAL_SERCOS_IFC). Array für zukünftige Erweiterungen mit mehreren Sercos-Schnittstellen. Nur bei MLC vorhanden MB_IMC_SERCOS_PHASE, Seite 22
ControlMode	MB_IMC_CONTROL_MODE_STATUS	Steuerungsmodus (C-0-0023, Bits 27..31). Nur bei MLC vorhanden MB_IMC_CONTROL_MODE_STATUS, Seite 23

Tab. 2-10: Elemente der Struktur "MB_IMC_ADMIN_STATUS"

2.2.7 MB_IMC_MODE_STATUS

Der Aufzählungstyp "MB_IMC_MODE_STATUS" enthält die Bestätigung der gewählten Betriebsart.

Name	Wert	Beschreibung
IMC_MODE_STATUS_P0	0	Bestätigung des Imc-Modus 0
IMC_MODE_STATUS_P1	1	Bestätigung des Imc-Modus 1
IMC_MODE_STATUS_P2	2	Bestätigung des Imc-Modus 2
IMC_MODE_STATUS_P3	3	Bestätigung des Imc-Modus 3
IMC_MODE_STATUS_BB	4	Bestätigung des Imc-Modus BB
IMC_MODE_UNDEFINED	255	Imc-Modus undefiniert

Tab. 2-11: Elemente der Struktur "MB_IMC_ADMINISTRATION"

2.2.8 MB_IMC_SERCOS_PHASE

Der Aufzählungstyp "MB_IMC_SERCOS_PHASE" enthält die Bestätigung der gewählten Sercos-Phase (nur bei MLC vorhanden).

Name	Wert	Beschreibung
IMC_SERCOS_PHASE_P0	0	Bestätigung der Sercos-Phase P0
IMC_SERCOS_PHASE_P1	1	Bestätigung der Sercos-Phase P1

Name	Wert	Beschreibung
IMC_SERCOS_PHASE_P2	2	Bestätigung der Sercos-Phase P2
IMC_SERCOS_PHASE_P3	3	Bestätigung der Sercos-Phase P3
IMC_SERCOS_PHASE_BB	4	Bestätigung der Sercos-Phase P4
IMC_SERCOS_PHASE_UNDEFINED	255	Sercos-Phase undefiniert

Tab. 2-12: Elemente der Struktur "MB_IMC_SERCOS_PHASE"

2.2.9 MB_IMC_CONTROL_MODE_STATUS

Der Aufzählungstyp "MB_IMC_CONTROL_MODE_STATUS" enthält die Bestätigung des gewählten Steuerungsmodus (nur bei MLC vorhanden).

Name	Wert	Beschreibung
IMC_CONTROL_MODE_STATUS_P0	0	Bestätigung Steuerungsmodus 0
IMC_CONTROL_MODE_STATUS_P1	1	Bestätigung Steuerungsmodus 1
IMC_CONTROL_MODE_STATUS_P2	2	Bestätigung Steuerungsmodus 2
IMC_CONTROL_MODE_STATUS_P3	3	Bestätigung Steuerungsmodus 3
IMC_CONTROL_MODE_STATUS_BB	4	Bestätigung Steuerungsmodus 4
IMC_CONTROL_MODE_UNDEFINED	255	Steuerungsmodus undefiniert

Tab. 2-13: Elemente der Struktur "MB_IMC_CONTROL_MODE_STATUS"

2.2.10 MB_IMC_DIAGNOSIS

Die Struktur enthält die Diagnoseinformationen der Steuerung (MotionControl → LogicControl).

Name	Typ	Beschreibung
Error	BOOL	Steuerungsfehler (C-0-0023, Bit 26: Systemstatus, Fehler)
Warning	BOOL	Steuerungswarning (C-0-0023, Bit 25: Systemstatus, Warnung)
ErrorID	ERROR_CODE	Fehlercode, nur gültig, wenn Error= TRUE
ErrorIdent	ERROR_STRUCT	Fehler-Detailinformation, nur gültig, wenn Error= TRUE
ClearErrorAck	BOOL	Bestätigung der Ausführung des Fehlerlöscher-Kommandos
ErrSeverity	MB_IMC_ERROR_SEVERITY, Seite 23	Fehlerkategorie
Number	DWORD	Anzeige der Fehlernummer (C-0-0627, Diagnosenummer)
Message	STRING	Anzeige des Fehlertextes (C-0-0626, Diagnose)

Tab. 2-14: Elemente der Struktur "MB_IMC_DIAGNOSIS"

2.2.11 MB_IMC_ERROR_SEVERITY

Der Aufzählungstyp "MB_IMC_ERROR_SEVERITY" vom TYP WORD zeigt die Schwere des Fehlers / der Warnung an. Er unterstützt den Aufbau sprachabhängiger Texte.

Name	Wert	Beschreibung
IMC_ERROR_UNDEFINED	16#00	Fehler undefiniert
IMC_ERROR_F0	16#F0	F0-Fehler (Nicht fataler Fehler)
...	...	...
IMC_ERROR_F9	16#F9	F9-Fehler (Fataler Systemfehler)
IMC_WARNING_E0	16#E0	E0-Warnung
...	...	...
IMC_WARNING_E9	16#E9	E9-Warnung

Tab. 2-15: Elemente der Aufzählung "MB_IMC_ERROR_SEVERITY"

2.3 Trigonometrische Funktionen

2.3.1 Einführung und Übersicht

Die nachfolgenden Funktionen erlauben dem Anwender Umrechnungen zwischen unterschiedlichen Einheiten vorzunehmen.

Da es sich um Funktionen handelt, können keine Fehler ausgegeben werden. Werden ungültige Werte verwendet, so wird der Wert 0 zurückgeliefert.

2.3.2 ML_DegToInc

Kurzbeschreibung

Die Funktion ML_DegToInc wandelt den in Gradmaß vorgegebenen Winkel ($0^\circ \leq x < 360^\circ$) in den entsprechenden Inkrementwert ($0 \leq x < 2^{20}$).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Grundpaket PLC	ML_TechBase
MLD (IL 2G)	MX_TechBase

Tab. 2-16: Referenztablelle

Schnittstellenbeschreibung

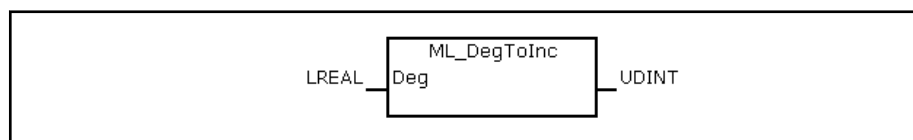


Abb. 2-1: Funktion ML_DegToInc

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Deg	LREAL	Winkel in Gradmaß [DEG]
Rückgabewert		UDINT	Winkel in Inkrementen

Tab. 2-17: Schnittstellenvariablen Funktion ML_DegToInc

Min.- Max.- und Default-Wert

In der folgenden Tabelle sind der Min-/Max- und Default-Wert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
Deg	LREAL	$-2^{32} * 360.0$	$2^{32} * 360.0$	0.0

Tab. 2-18: Min-/Max- und Default-Wert für Funktion ML_DegToInc

Funktionsbeschreibung

Die Funktion ML_DegToInc wandelt den in Gradmaß vorgegebenen Winkel ($0^\circ \leq x < 360^\circ$) in den entsprechenden Inkrementwert ($0 \leq x < 2^{20}$). Wird ein

Winkel größer gleich 360° vorgegeben, wird er auf den entsprechenden Winkel innerhalb des gültigen Wertebereichs umgerechnet (Modulo 360).

Wird ein Wert außerhalb des gültigen Wertebereiches vorgegeben, wird der Wert 0 zurückgeliefert.

Fehlercodes Die Funktion ML_DegToRad generiert keinen Fehler.

2.3.3 ML_DegToRad

Kurzbeschreibung Die Funktion ML_DegToRad konvertiert einen Winkelwert im Gradmaß in das entsprechende Bogenmaß inkl. Modulo.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Grundpaket PLC	ML_TechBase
MLD (IL 2G)	MX_TechBase

Tab. 2-19: Referenztablelle

Schnittstellenbeschreibung

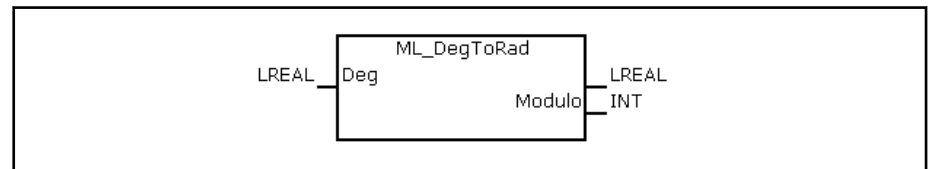


Abb. 2-2: Funktion ML_DegToRad

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Deg	LREAL	Winkel [DEG]
Rückgabewert	Rad	LREAL	Winkel [RAD]
VAR_OUTPUT	Modulo	INT	Modulo

Tab. 2-20: Schnittstellenvariablen Funktion ML_DegToRad

Min.- Max.- und Default-Wert In der folgenden Tabelle sind der Min-/Max- und Default-Wert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
Deg	LREAL	$-2^{15} * 360.0$	$2^{15} * 360.0$	0.0

Tab. 2-21: Min-/Max- und Default-Wert für Funktion ML_DegToRad

Funktionsbeschreibung Die Funktion ML_DegToRad konvertiert bei Aufruf der Instanz zyklisch den angelegten Winkel in Gradmaß in das entsprechende Bogenmaß inkl. Modulo.

Wird ein Wert außerhalb des gültigen Wertebereiches vorgegeben, wird der Wert 0 zurückgeliefert.

Fehlercodes Die Funktion ML_DegToRad generiert keinen Fehler.

2.3.4 ML_IncToDeg

Kurzbeschreibung Die Funktion ML_IncToDeg wandelt den in Inkrementen vorgegebenen Winkel ($0 \leq x < 2^{20}$) in den entsprechenden Winkel in Gradmaß ($0^\circ \leq x < 360^\circ$).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Grundpaket PLC	ML_TechBase
MLD (IL 2G)	MX_TechBase

Tab. 2-22: Referenztablelle

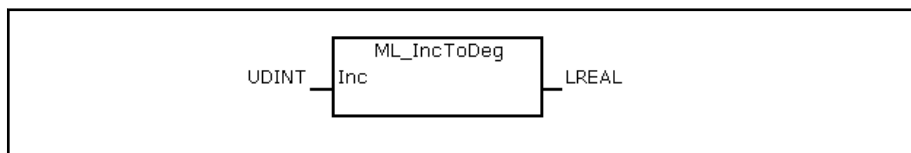
Schnittstellenbeschreibung

Abb. 2-3: Funktion ML_IncToDeg

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Inc	UDINT	Winkel in Inkrementen
Rückgabewert		LREAL	Winkel in Gradmaß [DEG]

Tab. 2-23: Schnittstellenvariablen Funktion ML_IncToDeg

Funktionsbeschreibung

Die Funktion ML_IncToDeg wandelt den in Inkrementen vorgegebenen Winkel ($0 \leq x < 2^{20}$) in den entsprechenden Winkel in Gradmaß ($0^\circ \leq x < 360^\circ$). Wird ein Winkel größer gleich 2^{20} Inkrementen vorgegeben, so wird er auf den entsprechenden Winkel innerhalb des gültigen Wertebereichs umgerechnet (Modulo 2^{20}).

Fehlercodes

Die Funktion ML_IncToDeg generiert keinen Fehler.

2.3.5 ML_IncToRad

Kurzbeschreibung

Die Funktion ML_IncToRad wandelt den in Inkrementen vorgegebenen Winkel ($0 \leq x < 2^{20}$) in den entsprechenden Winkel in Bogenmaß ($0 \leq x < 2\pi$).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Grundpaket PLC	ML_TechBase
MLD (IL 2G)	MX_TechBase

Tab. 2-24: Referenztablelle

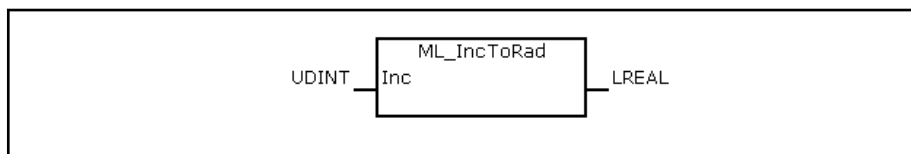
Schnittstellenbeschreibung

Abb. 2-4: Funktion ML_IncToRad

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Inc	UDINT	Winkel in Inkrementen
Rückgabewert		LREAL	Winkel in Bogenmaß [RAD]

Tab. 2-25: Schnittstellenvariablen Funktion ML_IncToRad

Funktionsbeschreibung

Die Funktion ML_IncToRad wandelt den in Inkrementen vorgegebenen Winkel ($0 \leq x < 2^{20}$) in den entsprechenden Winkel in Bogenmaß ($0 \leq x < 2\pi$). Wird ein Winkel größer gleich 2^{20} Inkrementen vorgegeben, so wird er auf den entsprechenden Winkel innerhalb des gültigen Wertebereichs umgerechnet (Modulo 2^{20}).

Fehlercodes Die Funktion ML_IncToRad generiert keinen Fehler.

2.3.6 ML_RadToDeg

Kurzbeschreibung Die Funktion ML_RadToDeg konvertiert einen Winkelwert im Bogenmaß in das entsprechende Gradmaß inkl. Modulo.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Grundpaket PLC	ML_TechBase
MLD (IL 2G)	MX_TechBase

Tab. 2-26: Referenztable

Schnittstellenbeschreibung

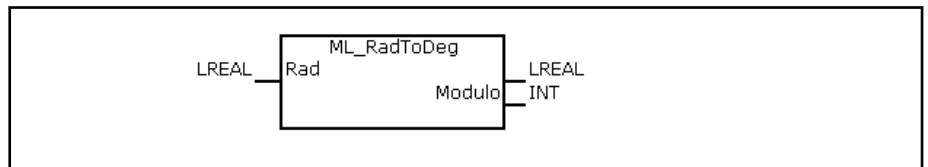


Abb. 2-5: Funktion ML_RadToDeg

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Rad	LREAL	Winkel [RAD]
Rückgabewert		LREAL	Winkel [DEG]
VAR_OUTPUT	Modulo	INT	Modulo

Tab. 2-27: Schnittstellenvariablen Funktion ML_RadToDeg

Min.- Max.- und Default-Wert In der folgenden Tabelle sind der Min-/Max- und Default-Wert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
Rad	LREAL	$-2 * \pi * 2^{15}$	$2 * \pi * 2^{15}$	0.0

Tab. 2-28: Min-/Max- und Default-Wert für Funktion ML_RadToDeg

Funktionsbeschreibung Die Funktion ML_RadToDeg konvertiert bei Aufruf der Instanz zyklisch den angelegten Winkel in Bogenmaß in das entsprechende Gradmaß inkl. Modulo.

Wird ein Wert außerhalb des gültigen Wertebereiches vorgegeben, wird der Wert 0 sowohl für "Deg", als auch für "Modulo" zurückgeliefert.

Fehlercodes Die Funktion ML_RadToDeg generiert keinen Fehler.

2.3.7 ML_RadToInc

Kurzbeschreibung Die Funktion ML_RadToInc wandelt den in Bogenmaß vorgegebenen Winkel ($0 \leq x < 2\pi$) in den entsprechenden Inkrementwert ($0 \leq x < 2^{20}$).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Grundpaket PLC	ML_TechBase
MLD (IL 2G)	MX_TechBase

Tab. 2-29: Referenztable

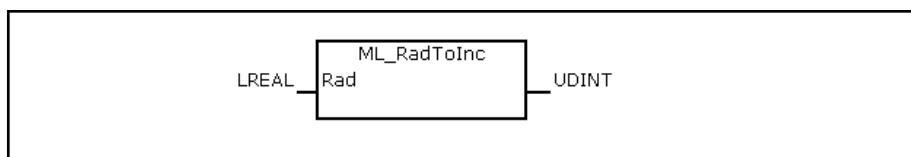
Schnittstellenbeschreibung

Abb. 2-6: Funktion ML_RadToInc

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Rad	LREAL	Winkel in Bogenmaß [RAD]
Rückgabewert		UDINT	Winkel in Inkrementen

Tab. 2-30: Schnittstellenvariablen Funktion ML_RadToInc

Min.- Max.- und Default-Wert

In der folgenden Tabelle sind der Min-/Max- und der Defaultwert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
Rad	LREAL	$-2 * \pi * 2^{32}$	$2 * \pi * 2^{32}$	0.0

Tab. 2-31: Min-/Max- und Default-Wert für Funktion ML_RadToInc

Funktionsbeschreibung

Die Funktion ML_RadToInc wandelt den in Bogenmaß vorgegebenen Winkel ($0 \leq x < 2\pi$) in den entsprechenden Inkrementwert ($0 \leq x < 2^{20}$). Wird ein Winkel größer gleich 2π vorgegeben, wird er auf den entsprechenden Winkel innerhalb des gültigen Wertebereichs umgerechnet (Modulo 2π).

Wird ein Wert außerhalb des gültigen Wertebereiches vorgegeben, wird der Wert 0 zurückgeliefert.

Fehlercodes

Die Funktion ML_RadToInc generiert keinen Fehler.

2.4 Systeminformationen

2.4.1 MB_GetSystemInfo

Kurzbeschreibung

Dieser Funktionsbaustein überprüft Systemoptionen und gibt alle aktivierten Systemoptionen zusammen mit dem Systemtyp aus.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Grundpaket PLC	ML_TechBase
MLD (IL2G)	MX_TechBase

Tab. 2-32: Referenztable MB_GetSystemInfo

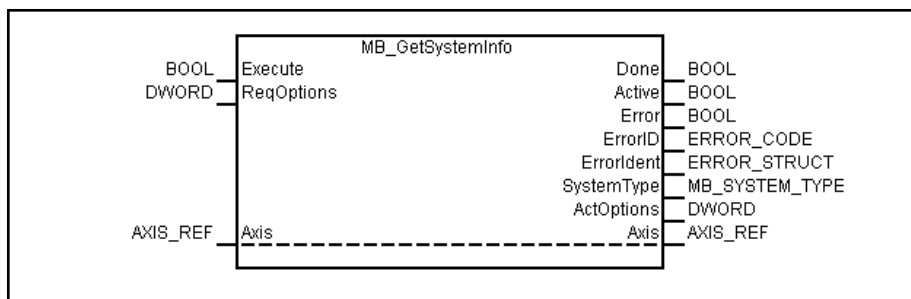
Schnittstellenbeschreibung

Abb. 2-7: Funktionsbaustein MB_GetSystemInfo

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Startet die Überprüfung
	ReqOptions	DWORD	Benötigte Systemoptionen
VAR_OUTPUT	Done	BOOL	Die Überprüfung verlief fehlerfrei
	Active	BOOL	Der Funktionsbaustein ist aktiv
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Grundsätzliche Fehlerursache
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler Fehlercodes, Seite 30
	SystemType	MB_SYSTEM_TYPE	Systemtyp MB_SYSTEM_TYPE, Seite 31
	ActOptions	DWORD	tatsächlich aktivierte Systemoptionen
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Zu prüfende Achse

Tab. 2-33: Schnittstellenvariablen Funktionsbaustein MB_GetSystemInfo

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
ReqOptions	DWORD	0	n.def.	0	steigende Flanke an "Execute"

Tab. 2-34: Eingangsverhalten MB_GetSystemInfo

Funktionsbeschreibung

System MLC/XLC: Dieser Funktionsbaustein arbeitet ab 14VRS unabhängig von der Sercos-Phase.

System MLD: Dieser Funktionsbaustein arbeitet in Sercos Phase 2 oder 4.

Bei steigender Flanke an "Execute" wird der Eingang "ReqOptions" übernommen. Danach wird das System überprüft.

Sind in "ReqOptions" Optionen oder Funktionspakete gesetzt, die nicht verfügbar sind, wird ein entsprechender Fehler (siehe [Fehlercodes, Seite 30](#)) generiert.

Nach erfolgreichem Abschluss wird der Systemtyp in "SystemType" und alle aktivierten Optionen in "ActOptions" ausgegeben und "Done" auf TRUE gesetzt.

Option	MLC/XLC	MLD	Wert	Beschreibung
MB_OPEN_LOOP	Axis	Axis	16#00000001	Grundpaket Open Loop
MB_CLOSED_LOOP	Axis	Axis	16#00000002	Grundpaket ClosedLoop
MB_SERVO	Axis	Axis	16#00000010	Funktionspaket Servo
MB_SYNCHRONIZATION	Axis	Axis	16#00000020	Funktionspaket Synchronisation bei Interpolation im Antrieb oder virtuelle Achse
MB_MAIN_SPINDLE	Axis	Axis	16#00000040	Funktionspaket Hauptspindel
MB_TF	immer gesetzt	erste Achse	16#00000200	Option erweiterte Technologiefunktion

Option	MLC/XLC	MLD	Wert	Beschreibung
MB_ML	immer gesetzt	erste Achse	16#00000200	Option Technologiefunktion
MB_MA	immer gesetzt	erste Achse	16#00000300	
MB_SAFETY			16#00010000	Sicherheitstechnik in der Achse aktiv
MB_AXISDATA	lokale Achsen	erste Achse	16#00100000	AxisData aktiviert
MB_REALAXIS	Reale Achse	immer gesetzt	16#01000000	
MB_VIRTUALAXIS	Virtuelle Achse		16#02000000	
MB_LINKAXIS	Achsen über Querverbund		16#04000000	
MB_LOCAL_INTERPOL	Interpolation in der Steuerung		16#10000000	
MB_INDRADRIIVE	IndraDrive	immer gesetzt	16#20000000	
MB_HYDRAULICDRIVE	HydraulicDrive		16#40000000	

Tab. 2-35: Mögliche Grundpakete, Funktionspakete und Optionen in ReqOptions und ActOptions



Der aktuelle Stand möglicher Optionen ist in der GVL "SystemOptions" enthalten.

Benötigte Optionen sollten in "ReqOptions" angegeben werden.

Ist mindestens eine Option auf der angegebenen Achse nicht verfügbar, wird die entsprechende Fehlermeldung generiert.

Sind alle in "ReqOptions" angeforderten Optionen vorhanden, wird "Done" aktiv.

Es ist auch möglich keine benötigten Optionen anzugeben und die Auswertung selbst vorzunehmen.

Die Überprüfung der Optionen benötigt mehrere Funktionsbausteinaufrufe. Während dieser Zeit ist der Ausgang "Active" gesetzt.

Fehlerverhalten Ist eine in "ReqOptions" angegebene Option nicht verfügbar oder ist ein anderer Fehler aufgetreten, wird "Error" TRUE.

Fehlercodes Der Funktionsblock generiert die folgenden Fehlermeldungen der "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR	16#1503	16#0000	AxisData-Struktur nicht aktiviert
RESOURCE_ERROR	16#000F	16#0001	Notwendige Option TF nicht freigeschaltet
RESOURCE_ERROR	16#000F	16#0002	Notwendige Option ML nicht freigeschaltet
RESOURCE_ERROR	16#000F	16#0003	Notwendige Option MA nicht freigeschaltet
RESOURCE_ERROR	16#000F	16#0010	Notwendiges Grundpaket OpenLoop nicht aktiv
RESOURCE_ERROR	16#000F	16#0011	Notwendiges Grundpaket ClosedLoop nicht aktiv
RESOURCE_ERROR	16#000F	16#0020	Notwendiges Funktionspaket Synchronisation nicht aktiv
RESOURCE_ERROR	16#000F	16#0021	Notwendiges Funktionspaket Servo nicht aktiv
RESOURCE_ERROR	16#000F	16#0022	Notwendiges Funktionspaket Hauptspindel nicht aktiv
RESOURCE_ERROR	16#000F	16#0023	Notwendiger Achstyp (REAL,VIRTUAL, LINK) nicht korrekt

Tab. 2-36: Fehlercodes Funktionsbaustein MB_GetSystemInfo

2.4.2 MB_SYSTEM_TYPE

Der Aufzählungstyp enthält die aktuell verfügbaren Steuerungssysteme.

Systemtyp		Beschreibung
MB_SYSTEM_UNKNOWN	0	Unbekanntes System
MB_SYSTEM_MLC	1	MLC
MB_SYSTEM_MLD	2	MLD
MB_SYSTEM_MTX	3	MTX
MB_SYSTEM_MLP	4	MLP
MB_SYSTEM_XLC	5	XLC

Tab. 2-37: Mögliche Systemtypen

3 ML_TechMotionBase.library // MX_TechMotionBase.library

3.1 Einführung und Übersicht

Allgemeines Technologie-Funktionsbausteine (Tech-FB) erweitern die Grundfunktionalität der Zielsysteme MLC/XLC und MLD und stellen applikationsspezifische Funktionalitäten zur Verfügung. Die hier beschriebene Bibliothek beinhaltet Basis-Funktionalitäten, die für viele verschiedene Anwendungen zum Einsatz kommen, aber auch einzeln verwendet werden können.

Die hier beschriebenen Funktionsbausteine werden über die Bibliothek "ML_TechMotionBase" für MLC/XLC zur Verfügung gestellt.

Diese Dokumentation beschreibt die Funktionalität, sowie die Ein- und Ausgänge der Technologie-Funktionsbausteine.

Voraussetzungen Technologie-Funktionsbausteine der Bibliothek "ML_TechMotionBase" benötigen die Firmware-Unterstützung des Zielsystems. Spezifische Voraussetzungen der Technologie-Funktionsbausteine sind bei den entsprechenden Funktionsbausteinen dokumentiert.



Die Bibliothek benötigt das **MOT (Motion) Funktionspaket** der Steuerung.

Anwendungsgebiete und Funktionsbausteine von ML_TechMotionBase

Anwendungsgebiet (Ordner in der Bibliothek)	
Funktionsbausteine	Beschreibung
Achsen, Sichern und Wiederherstellen (AxisBackupRestore)	
ML_AxisBackup, Seite 35	Parametersicherung einer Achse auf der Kompakt-Flash-Karte
ML_AxisRestore, Seite 38	Wiederherstellung von Parametern einer Achse von der Kompakt-Flash-Karte
Messgeber, Ermittlung gefahrener Wege (DistanceAccumulator)	
MB_DistanceAccumulator, Seite 50	Summiert den relativ gefahrenen Weg mittels eines Gebers auf und hält an, sobald der aufsummierte Weg größer als oder gleich des TotalDistance Eingangswerts ist. Die aktuelle Geberposition wird mittels des CurrentPosition Eingangs eingegeben
MB_AxisDistanceAccumulator, Seite 51	Summiert den relativ gefahrenen Weg mittels eines Gebers auf und hält an, sobald der aufsummierte Weg größer als oder gleich des TotalDistance Eingangswerts ist. Dieser Funktionsbaustein verwendet den Achsreferenzeingang (AXIS_REF) um die aktuelle Geberposition automatisch abzufragen
Kurvenscheibentabellen, Download (LoadCamData)	
MB_LoadCamData, Seite 99	übernimmt den Download von Kurvenscheiben auf den Antrieb und die Steuerung, um die vorhandenen Detailunterschiede zu kapseln
Nockenschaltwerk (PhaseLimitSwitch)	
MC_DigitalCamSwitch, Seite 53	implementiert ein frei programmierbares Nockenschaltwerk mit bis zu 32 Nocken auf 8 Ausgängen

MB_DigitalCamSwitchOvs, Seite 71	implementiert ein frei programmierbares Nockenschaltwerk mit bis zu 64 Nocken auf 32 Nockenscheiben unterstützt Oversampling, dadurch kann eine entsprechend höhere zeitliche Auflösung der Nockenschaltwerkssignale erreicht werden setzt EA-Module, die Oversampling unterstützen, voraus
Filtern einer Position (PositionFiltering)	
MB_PositionFilterType01, Seite 103	filtert eine inkrementelle Position und berechnet Position und Geschwindigkeit
Positionsüberwachung (PositionMonitoring)	
MB_PositionMonitoring_GantryCant, Seite 106	Überwacht die relative Position zweier Achsen bezüglich Verkanten zueinander. Mittels der Toleranzband-Funktionalität ist die Rückzugsbewegung möglich, wobei die Bewegungsrichtung aus der Verkanten weiter überwacht wird
MB_PositionMonitoring_PosCollision, Seite 109	Überwacht die absolute und relative Position zweier Achsen bezüglich Auszahnung und Kollision
S-Kurve (SCurve)	
MB_ReachVelocityType02, Seite 112	Berechnet den entsprechenden Ruck, die Beschleunigung und die Zielgeschwindigkeit zum Abfahren bestimmter ruckbegrenzter Geschwindigkeitsverläufe (z.B. S-Kurve) und lässt hierbei bis zu 3 Änderungen der Zielgeschwindigkeiten zu
Messtasterfunktion (TouchProbe)	
MB_InitTouchProbe, Seite 127	Initialisierung der Messtasterfunktionalität
MB_TouchProbe, Seite 135	Erfassen einer Achsposition, Leitachsposition oder Zeiterfassung, Einzelmessung
MB_TouchProbeContinuous, Seite 138	Erfassen einer Achsposition, Leitachsposition oder Zeiterfassung bei fortlaufender Messung
MB_WriteExpectWindow, Seite 142	konfiguriert das Erwartungsfenster für die Messtasterfunktionalität
MB_AbortTrigger, Seite 145	Abbruch des Funktionsbausteins MB_TouchProbe
Zyklischer Datenkanal, Verwaltung (Utilities/ CyclicData)	
MB_AllocCyclicParameter, Seite 148	Überprüfung und Überarbeitung der Konfiguration eines zyklisch zu lesenden oder beschreibenden Parameters im zyklischen Datenkanal
MB_FreeCyclicParameter, Seite 150	Freigabe eines bereits konfigurierten zyklisch zu lesenden oder beschreibenden Parameters im zyklischen Datenkanal
MB_GetCyclicChannelConfig, Seite 152	liest die Konfiguration zyklischer Lese- und Schreibdatenkanäle aus und speichert sie in einer Datenstruktur
MB_GetCyclicParameterHandle, Seite 154	sucht im zyklischen Datenkanal nach einem zyklisch zu lesenden oder beschreibenden Parameter und legt alle benötigten Informationen in einem Handle ab
MB_GetLastCyclicParameterError, Seite 158	gibt den letzten aufgetretenen Fehler beim Lesen oder Schreiben eines zyklischen Werts für einen Handle zurück und löscht den Fehler
MB_ReadCyclicParameter, Seite 155	liest einen zyklischen Parameter über den Handle und gibt den aktuellen Inhalt als DINT zurück
MB_ReadCyclicRealParameter, Seite 156	liest einen zyklischen Parameter über den Handle und gibt den aktuellen Inhalt als REAL zurück

MB_WriteCyclicParameter, Seite 157	schreibt einen zyklischen Parameter über den Handle, Nachkommastellen werden gemäß sercos Vereinbarung gewandelt
MB_WriteCyclicRealParameter, Seite 157	schreibt einen zyklischen Parameter über den Handle, Nachkommastellen werden gerundet

Tab. 3-1: *ML_TechMotionBase-Bibliothek*

3.2 Sichern und Wiederherstellen von Antriebsparametern

3.2.1 Einführung und Übersicht

Antriebe werden normalerweise über IndraWorks parametrieren. An laufenden Anlagen besteht jedoch häufig die Notwendigkeit, Antriebe ohne die Verwendung von IndraWorks zu tauschen und danach neu parametrieren zu können.

Dieser Anwendungsfall wird durch nachstehende Funktionsbausteine abgedeckt.

Die Wiederherstellung von Parametern erfordert, dass sich der Antrieb im Parametriermodus (P2) befindet. Dies wird auch beim Backup von Parametern empfohlen.

- [ML_AxisBackup, Seite 35](#), erstellt eine Parametersicherung einer Achse und legt sie als textbasierte Datei auf der Kompakt-Flash-Karte der Steuerung ab.
- [ML_AxisRestore, Seite 38](#), spielt diese Parametersicherung wieder auf der Achse ein
- [ML_AxisBackup_Type02, Seite 41](#), sichert die Parameter einer Achse in einer textbasierten Datei auf der Kompakt-Flash-Karte der Steuerung
- [ML_AxisRestore_Type02, Seite 44](#), lädt die Parameter aus einer textbasierten Datei auf der Kompakt-Flash-Karte der Steuerung auf eine Achse

3.2.2 ML_AxisBackup

Kurzbeschreibung Der Funktionsbaustein ML_AxisBackup sichert die Parameter einer Achse in einer textbasierten Datei auf der Kompakt-Flash-Karte der Steuerung.



Der Funktionsbaustein "ML_AxisBackup (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase

Tab. 3-2: *Referenztable ML_AxisBackup*

ML_TechMotionBase.library // MX_TechMotionBase.library

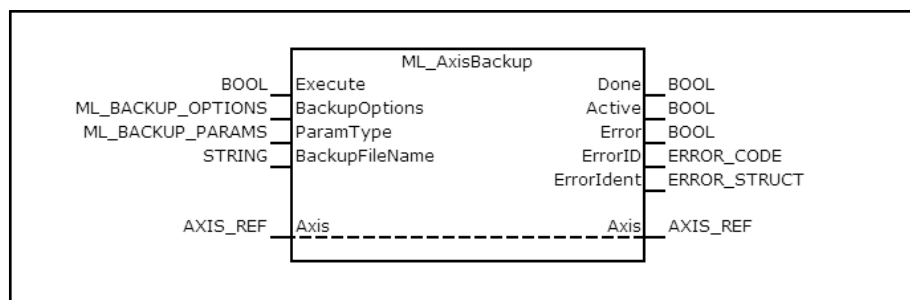
Schnittstellenbeschreibung

Abb. 3-1: Funktionsbaustein ML_AxisBackup

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (flanken-gesteuert)
	BackupOptions	ML_BACKUP_OPTI- ONS	Umfang der Parametersicherung ML_BACKUP_OPTIONS, Seite 49
	ParamType	ML_BACKUP_PA- RAMS	Auswahl der zu sichernden Parameter: alle A, S und P ML_BACKUP_PARAMS, Seite 49
	BackupFileName	STRING	Dateiname auf der Kompakt-Flash-Karte der Steuerung
VAR_OUTPUT	Done	BOOL	Bearbeitung abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten gültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Fehlerdiagnose (siehe Fehlercodes, Seite 37)
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Referenz auf die Achse

Tab. 3-3: Schnittstellenvariablen Funktionsbaustein ML_AxisBackup

Min.- Max.- und Default-Werte der Eingänge

Die nachfolgenden Wertebereiche und Defaultwerte können an den Baustein angelegt werden:

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
BackupOptions	ML_BACK- UP_OPTIONS			TO_SAVE_PA- RAMS	Steigende Flanke an "Execute"
ParamType	ML_BACK- UP_PARAMS			ASP_PARAMS	Steigende Flanke an "Execute"
BackupFileName	STRING			"	Steigende Flanke an "Execute"

Tab. 3-4: Min-/Max- und Defaultwerte Funktionsbaustein ML_AxisBackup

Funktionsbeschreibung

Der Funktionsbaustein ML_AxisBackup sichert die Parameter einer Achse in einer textbasierten Datei auf einer Partition der Kompakt-Flash-Karte der Steuerung. Ist die Datei bereits vorhanden, so wird sie überschrieben. Die Datei kann vom Funktionsbaustein ML_AxisRestore und von IndraWorks wieder gelesen werden.

BackupOptions

Über den Eingang "BackupOptions" werden die zu sichernden Parameter ausgewählt. Die folgenden Auswahlmöglichkeiten sind vorhanden:

- ALL_PARAM: Es werden alle Parameter gesichert. Diese Option wird vor allem zum Überprüfen der Konfiguration benötigt
- MODIFIED_PARAM: Es werden die von den Default-Werten abweichenden Parameter gesichert
- TO_SAVE_PARAM: Es werden die für eine Wiederherstellung notwendigen Parameter gesichert. Dies ist die Voreinstellung



Der Eingang "BackupOptions" wird derzeit nur für S- und P-Parameter ausgewertet. Beim Export von A-Parametern werden immer alle Parameter (ALL_PARAM), oder die für die Wiederherstellung notwendigen Parameter (MODIFIED_PARAM und TO_SAVE_PARAM) exportiert.

ParamType

Über den Eingang "ParamType" werden die zu sichernden Parametertypen ausgewählt. Die folgenden Optionen stehen zur Wahl:

- ASP_PARAMS: Parameter im Antrieb und steuerungsinterne Achsparameter
- A_PARAMS: steuerungsinterne Achsparameter
- SP_PARAMS: Parameter im Antrieb

BackupFileName

Über "BackupFileName" kann der Name der Datei auf der Kompakt-Flash-Karte der Steuerung angegeben werden. Es ist möglich, eine Pfadangabe zu verwenden. Der Pfad muss dann aber auf der Kompakt-Flash-Karte ansprechbar sein.

Beispiel:

- "Antrieb1.par" (legt die Datei "Antrieb1.par" auf der User-Partition der Kompakt-Flash-Karte ab)
- "Params/Antrieb1.par" (legt die Datei "Antrieb1.par" im Verzeichnis "Params" auf der User-Partition der Kompakt-Flash-Karte ab)
- "/ata0a/Antrieb2.par" (legt die Datei "Antrieb2.par" auf der OEM-Partition der Steuerung ab)

Zeitverhalten

Der Funktionsbaustein ML_AxisBackup arbeitet asynchron. Nach dem Start durch eine steigende Flanke am Eingang "Execute" liest er die Eingänge "BackupOptions", "ParamType" und "BackupFileName". Solange der Funktionsbaustein arbeitet, ist der Ausgang "Active" gesetzt. Nach erfolgreichem Abschluss der Operation ist der Ausgang "Done" gesetzt. Tritt ein Fehler auf, wird der Ausgang "Error" gesetzt.

Fehlercodes

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#0000000D	16#00000000	AxisRef liegt außerhalb des gültigen Bereiches
INPUT_RANGE_ERROR	16#00001E00	16#00000000	Ungültiger Eingang "BackupOptions"
INPUT_RANGE_ERROR	16#00001E01	16#00000000	Ungültiger Eingang "ParamType"

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#00001E02	16#00000000	Die in "ParamType" ausgewählte Parameterart wird von dieser Achse nicht unterstützt
RESOURCE_ERROR	16#00001E03	16#xxxxxxx	Fehler beim Lesen eines Parameters
ACCESS_ERROR	16#00001E10	16#00000000	Fehler beim Anlegen der Backup-Datei
ACCESS_ERROR	16#00001E12	16#00000000	Fehler beim Schreiben der Backup-Datei
ACCESS_ERROR	16#00001E14	16#00000000	Fehler beim Schließen der Backup-Datei
STATE_MACHINE_ERROR	16#00000006	16#0000FFFF	Unerlaubter Zustand des Funktionsbausteins

Tab. 3-5: Fehlercodes Funktionsbaustein ML_AxisBackup

3.2.3 ML_AxisRestore

Kurzbeschreibung

Der Funktionsbaustein ML_AxisRestore lädt die Parameter aus einer textbasierten Datei auf der Kompakt-Flash-Karte der Steuerung auf eine Achse.



Der Funktionsbaustein "ML_AxisRestore (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase

Tab. 3-6: Referenztable ML_AxisRestore

Schnittstellenbeschreibung

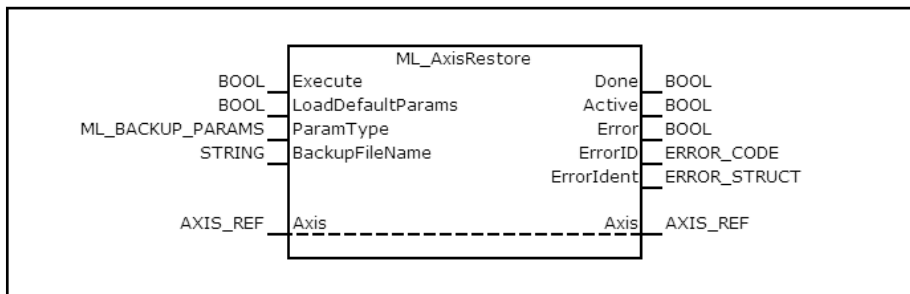


Abb. 3-2: Funktionsbaustein ML_AxisRestore

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (flankengesteuert)
	LoadDefaultParams	BOOL	Default-Parameter laden, bevor die Parameter aus der Datei geladen werden
	ParamType	ML_BACKUP_PARAMS	Auswahl der wiederherzustellenden Parameter: alle A, S und P ML_BACKUP_PARAMS, Seite 49
	BackupFileName	STRING	Dateiname auf der Kompakt-Flash-Karte der Steuerung

E/A-Typ	Name	Datentyp	Beschreibung
VAR_OUTPUT	Done	BOOL	Bearbeitung abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten gültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Fehlerdiagnose (siehe Fehlercodes , Seite 41)
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Referenz auf die Achse

Tab. 3-7: Schnittstellenvariablen Funktionsbaustein ML_AxisRestore

Min.- Max.- und Default-Werte der Eingänge

Die nachfolgenden Wertebereiche und Defaultwerte können an den Baustein angelegt werden:

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
LoadDefaultParams	BOOL			FALSE	Steigende Flanke an "Execute"
ParamType	ML_BACK- UP_PARAMS			ASP_PARAMS	Steigende Flanke an "Execute"
BackupFileName	STRING			"	Steigende Flanke an "Execute"

Tab. 3-8: Min-/Max- und Defaultwerte Funktionsbaustein ML_AxisRestore

Funktionsbeschreibung

Der Funktionsbaustein ML_AxisRestore lädt die Parameter aus einer textbasierten Datei auf der Kompakt-Flash-Karte der Steuerung auf eine Achse. Nach erfolgreichem Laden der Parameterdatei wird die Konfiguration auf dem Antrieb oder der Achse gespeichert. Sie steht somit auch nach dem Ausschalten wieder zur Verfügung. Die Parameter-Datei kann durch den Funktionsbaustein ML_AxisBackup oder durch IndraWorks erzeugt werden.

Der Baustein sichert folgende Parameter **nicht** zurück, auch wenn diese in der Parameterdatei enthalten sind:

- alle Parameter die schreibgeschützt sind
- alle Parameter die Kommandos darstellen
- alle Parameter die zur Zeit schreibgeschützt sind (z.B. durch Passwort geschützt, Schreibschutz da SPS in RUN, etc.)
- Parameter A-0-0002, A-0-0005, A-0-0016, A-0-0024, A-0-0501, A-0-0503, A-0-0505, A-0-0507, A-0-0509, A-0-0511, A-0-2220, S-0-0265, S-0-0267, S-0-1040, S-0-4025, P-0-4025

LoadDefaultParams

Ist der Eingang "LoadDefaultParams" gesetzt, so wird vor dem Laden der Parameter ein Laden der Default-Parameter des Antriebs bzw. der Achse durchgeführt. Dies ist vor allem dann sinnvoll, wenn die Parameter-Datei mit den Optionen "MODIFIED_PARAM" oder "TO_SAVE_PARAM" erzeugt wurde.



Das Laden der Default-Parameter dauert pro Achse mehrere Sekunden.



Beim Laden der Default-Parameter für ein IndraDrive wird vor dem Auslösen des Kommandos noch der Parameter P-0-4090 mit dem Wert 16#51 geschrieben. Damit werden die Parameter der Führungskommunikation und der Engineering Schnittstelle nicht mit Defaultwerten belegt. Sollen weitere und/oder andere Parametergruppen nicht initialisiert werden, so ist das Laden der Default-Parameter selbst durchzuführen.

ParamType

Über den Eingang "ParamType" werden die wiederherzustellenden Parametertypen ausgewählt. Die folgenden Optionen stehen zur Wahl:

- ASP_PARAMS: Parameter im Antrieb und steuerungsinterne Achsparameter
- A_PARAMS: steuerungsinterne Achsparameter
- SP_PARAMS: Parameter im Antrieb

BackupFileName

Über "BackupFileName" kann der Name der Datei auf der Kompakt-Flash-Karte der Steuerung angegeben werden. Es ist möglich, eine Pfadangabe zu verwenden. Der Pfad muss dann aber auf der Kompakt-Flash-Karte ansprechbar sein.

Beispiel:

- "Antrieb1.par" (liest die Datei "Antrieb1.par" von der User-Partition der Kompakt-Flash-Karte)
- "Params/Antrieb1.par" (liest die Datei "Antrieb1.par" vom Verzeichnis "Params" auf der User-Partition der Kompakt-Flash-Karte)
- "/ata0a/Antrieb2.par" (liest die Datei "Antrieb2.par" von der OEM-Partition der Steuerung)

Zeitverhalten

Der Funktionsbaustein ML_AxisRestore arbeitet asynchron. Nach dem Start durch eine steigende Flanke am Eingang "Execute" liest er die Eingänge "LoadDefaultParams", "ParamType" und "BackupFileName". Solange der Funktionsbaustein arbeitet, ist der Ausgang "Active" gesetzt. Nach erfolgreichem Abschluss der Operation ist der Ausgang "Done" gesetzt. Tritt ein Fehler auf, wird der Ausgang "Error" gesetzt.

Tritt ein Fehler beim Speichern eines Parameters auf, so fährt der Baustein mit dem nächsten Parameter fort, bis alle Parameter abgearbeitet sind. Am Ausgang wird dann der erste aufgetretene Fehler angezeigt.

Properties "plgnoreSourceAxis-Number" und "pUsedSourceAxis-Number"

Der Baustein importiert nur Dateien der gleichen Achsinstanz. Wird eine Parameterdatei verwendet, die von einer anderen Achse exportiert wurde, so werden bei einem Import keine Parameter importiert, da die Achsinstanzen nicht gleich sind. Dies dient dazu, um bei einer Parameterdatei mit vielen Achsinstanzen die richtige Instanz zu importieren.

Es kann aber auch sinnvoll sein, diese Achsinstanz zu ignorieren, um beispielsweise bei einer Serieninbetriebnahme nur eine Parameterdatei mit einer einzigen Instanz zu besitzen, diese aber auf viele Achsen zu importieren. Für diesen Fall besitzt der Baustein noch eine Property "plgnoreSourceAxis-Number". Setzt man diese vor Verwendung des Bausteins auf TRUE, so ignoriert der Baustein die Achsinstanz und importiert die Parameter. Die Defaulteinstellung für die Property ist FALSE, d. h. es wird nur die jeweilige Achsinstanz importiert.



Verwendet man eine Parameterdatei mit vielen Instanzen und hat die Property auf TRUE gesetzt, so importiert der Baustein alle Instanzen nacheinander auf die gleiche Achse und diese Achse besitzt anschließend die Parameter des letzten Datensatzes in der Parameterdatei.

In der Property "pUsedSourceAxisNumber" kann nach der Ausführung des Bausteins zurückgelesen werden, welche Adresse der Baustein in der Parameterdatei gefunden und verwendet hat.

Fehlercodes

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#0000000D	16#00000000	AxisRef liegt außerhalb des gültigen Bereiches
INPUT_RANGE_ERROR	16#00001E01	16#00000000	Ungültiger Eingang "ParamType"
RESOURCE_ERROR	16#00001E02	16#00000000	Die in "ParamType" ausgewählte Parameterart wird von dieser Achse nicht unterstützt
RESOURCE_ERROR	16#00001E04	16#xxxxxxx	Fehler beim Schreiben eines Parameters. Additional2 gibt die fehlerhafte Parameternummer an.
ACCESS_ERROR	16#00001E11	16#00000000	Fehler beim Öffnen der Backup-Datei
ACCESS_ERROR	16#00001E13	16#00000000	Fehler beim Lesen der Backup-Datei
ACCESS_ERROR	16#00001E14	16#00000000	Fehler beim Schließen der Backup-Datei
OTHER_ERROR	16#00001E15	16#xxxxxxx	Fehler beim Parsen der Backup-Datei
STATE_MACHINE_ERROR	16#00000006	16#0000FFx	Unerlaubter Zustand des Funktionsbausteins

Tab. 3-9: Fehlercodes Funktionsbaustein ML_AxisRestore

3.2.4 ML_AxisBackup_Type02

Kurzbeschreibung

Der Funktionsbaustein ML_AxisBackup_Type02 sichert die Parameter einer Achse in einer textbasierten Datei auf der Kompakt-Flash-Karte der Steuerung.

Der Baustein ist eine Weiterentwicklung des ML_AxisBackup, bei dem die Datensicherung, je nach Parametertyp wesentlich schneller erfolgt ([ML_AxisBackup](#), Seite 35).

Der Baustein ist weitestgehend kompatibel geblieben.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase

Tab. 3-10: Referenztable ML_AxisBackup_Type02

ML_TechMotionBase.library // MX_TechMotionBase.library

Schnittstellenbeschreibung

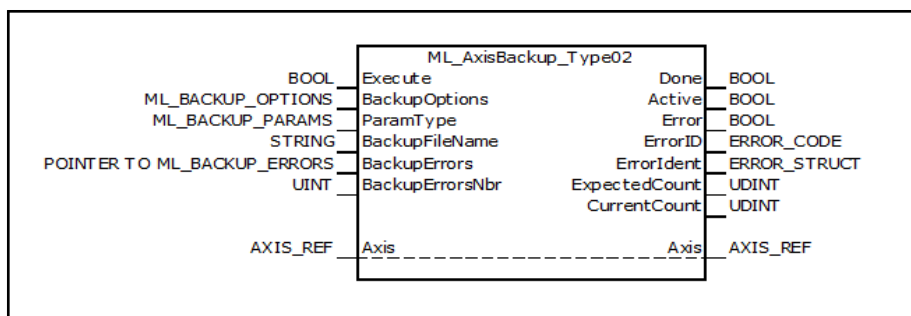


Abb. 3-3: Funktionsbaustein ML_AxisBackup_Type02

EA-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (flankengesteuert)
	BackupOptions	ML_BACKUP_OPTIONS	Umfang der Parametersicherung ML_BACKUP_OPTIONS, Seite 49
	ParamType	ML_BACKUP_PARAMS	Auswahl der zu sichernden Parameter: A, S, P ML_BACKUP_PARAMS, Seite 49
	BackupFileName	STRING	Dateiname auf der Kompakt-Flash-Karte der Steuerung
	BackupErrors	POINTER TO ML_BACKUP_ERRORS	Pointer auf ein Array von ML_BACKUP_ERRORS Strukturen ML_BACKUP_ERRORS, Seite 49
	BackupErrorsNbr	UINT	Anzahl der Felder im Array BackupErrors
VAR_OUTPUT	Done	BOOL	Bearbeitung abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten gültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Fehlerdiagnose (siehe Fehlercodes, Seite 44)
	ExpectedCount	UDINT	Anzahl an erwarteten Parametern für den Export
	CurrentCount	UDINT	Anzahl bereits exportierter Parameter
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Referenz auf die Achse

Tab. 3-11: Schnittstellenvariablen Funktionsbaustein ML_AxisBackup_Type02

Min.- Max.- und Default-Werte der Eingänge

Die nachfolgenden Wertebereiche und Defaultwerte können an den Baustein angelegt werden:

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL	FALSE	TRUE	FALSE	Kontinuierlich
BackupOptions	ML_BACKUP_OPTIONS			TO_SAVE_PARAMS	Steigende Flanke an "Execute"
ParamType	ML_BACKUP_PARAMS			ASP_PARAMS	Steigende Flanke an "Execute"

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
BackupFileName	STRING			"	Steigende Flanke an "Execute"
BackupErrors	POINTER TO ML_BACKUP_ERRORS	> 0			Kontinuierlich
BackupErrorsNbr	UINT	0	65535	0	Kontinuierlich

Tab. 3-12: Min-/Max- und Defaultwerte Funktionsbaustein ML_AxisBackup_Type02

Funktionsbeschreibung

Der Funktionsbaustein ML_AxisBackup_Type02 sichert die Parameter einer Achse in einer textbasierten Datei auf der Kompakt-Flash-Karte der Steuerung. Ist die Datei bereits vorhanden, so wird sie überschrieben. Die Datei kann vom Funktionsbaustein IL_AxisRestore_Type02 und von IndraWorks wieder gelesen werden.

BackupOptions

Über den Eingang "BackupOptions" werden die zu sichernden Parameter ausgewählt. Die folgenden Auswahlmöglichkeiten sind vorhanden:

- ALL_PARAM: Es werden alle Parameter gesichert. Diese Option wird vor allem zum überprüfen der Konfiguration benötigt
- MODIFIED_PARAM: Es werden die von den Default-Werten abweichenden Parameter gesichert
- TO_SAVE_PARAM: Es werden die für eine Wiederherstellung notwendigen Parameter gesichert. Dies ist die Voreinstellung

ParamType

Über den Eingang "ParamType" werden die zu sichernden Parametertypen ausgewählt. Die folgenden Optionen stehen zur Wahl:

- ASP_PARAMS: Parameter im Antrieb und steuerungsinterne Achsparameter
- A_PARAMS: steuerungsinterne Achsparameter
- SP_PARAMS: Parameter im Antrieb

BackupFileName

Über "BackupFileName" kann der Name der Datei auf der Kompakt-Flash-Karte der Steuerung angegeben werden. Es sollte immer der gesamte Pfad angegeben werden. Um Hardware unabhängig zu sein, sollte daher die Funktion "ML_GetSystemSpecialPath" aus der Bibliothek "ML_Base" verwendet werden, um den gewünschten Pfad zu ermitteln. Unterlagerte Pfade können angegeben werden, müssen dann aber auch vorhanden sein.

Programm:

```
// Variable definition
strPathName: STRING(255);
strPathFile: STRING(255);

// Code
// Get path to USER partition
strPathName := ML_GetSystemSpecialPath(PATH_USER);
strPathFile := concat(strPathName, 'MyFile.par');
// Final string will be (depending on hardware)
// '/USER//MyFile.par'

// Get path to OEM partition
strPathName := ML_GetSystemSpecialPath(PATH_OEM);
// Add own path within OEM partition (use '/' as folder delimiter)
```

ML_TechMotionBase.library // MX_TechMotionBase.library

```
// Own path must already exist in the directory
strPathName := concat(strPathName, '/myPath/');
// Add filename to path
strPathFile := concat(strPathName, 'MyFile.par');
// Final string will be (depending on hardware)
//      '/OEM/myPath/MyFile.par'
```

Zeitverhalten Der Funktionsbaustein ML_AxisBackup_Type02 arbeitet asynchron. Nach dem Start durch eine steigende Flanke am Eingang "Execute" liest er die Eingänge "BackupOptions", "ParamType" und "BackupFileName" einmalig ein, die Eingänge "BackupErrors" und "BackupErrorsNbr" werden kontinuierlich gelesen. Solange der Funktionsbaustein arbeitet, ist der Ausgang "Active" gesetzt. Nach erfolgreichem Abschluss der Operation ist der Ausgang "Done" gesetzt. Tritt ein Fehler auf, wird der Ausgang "Error" gesetzt.

Der Baustein sollte nicht in der Motion-Task verwendet werden, da er zeitweise längere Ausführungszeiten benötigt.

Fehlercodes Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**.

Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#0000000D	16#00000000	AxisRef liegt außerhalb des gültigen Bereiches
INPUT_RANGE_ERROR	16#00001E00	16#00000000	Ungültiger Eingang "BackupOptions"
INPUT_RANGE_ERROR	16#00001E01	16#00000000	Ungültiger Eingang "ParamType"
RESOURCE_ERROR	16#00001E02	16#00000000	Die in "ParamType" ausgewählte Parameterart wird von dieser Achse nicht unterstützt
RESOURCE_ERROR	16#00001E03	16#xxxxxxx	Fehler beim Lesen eines Parameters
ACCESS_ERROR	16#00001E05	16#00000000	"BackupErrors" ist 0 oder "BackupErrorsNbr" ist 0
ACCESS_ERROR	16#00001E10	16#00000000	Fehler beim Erstellen der Parameterdatei
STATE_MACHINE_ERROR	16#00000006	16#0000FFFF	Unerlaubter Zustand des Funktionsbausteins

Tab. 3-13: Fehlercodes FB ML_AxisBackup_Type02

3.2.5 ML_AxisRestore_Type02

Kurzbeschreibung Der Funktionsbaustein ML_AxisRestore_Type02 lädt die Parameter aus einer textbasierten Datei auf der Kompakt-Flash-Karte der Steuerung auf eine Achse.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase

Tab. 3-14: Referenztable ML_AxisRestore_Type02

Schnittstellenbeschreibung

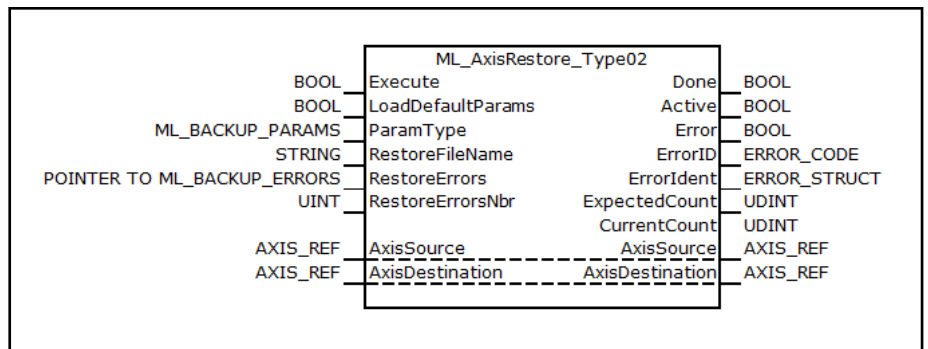


Abb. 3-4: Funktionsbaustein ML_AxisRestore_Type02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (flankengesteuert)
	LoadDefaultParams	BOOL	Default-Parameter laden, bevor die Parameter aus der Datei geladen werden
	ParamType	ML_BACKUP_PARAMS	Auswahl der wiederherzustellenden Parameter: alle A, S und P ML_BACKUP_PARAMS, Seite 49
	RestoreFileName	STRING	Dateiname auf der Kompakt-Flash-Karte der Steuerung
	BackupErrors	POINTER TO ML_BACKUP_ERRORS	Pointer auf ein Array von ML_BACKUP_ERRORS Strukturen ML_BACKUP_ERRORS, Seite 49
	BackupErrorsNbr	UINT	Anzahl der Felder im Array BackupErrors
VAR_OUTPUT	Done	BOOL	Bearbeitung abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten gültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Fehlerdiagnose (siehe Fehlercodes, Seite 48)
VAR_IN_OUT CONSTANT	AxisSource	AXIS_REF	Referenz auf die Achse, von der Parameter importiert werden sollen
	AxisDestination	AXIS_REF	Referenz auf die Achse, zu der Parameter importiert werden sollen

Tab. 3-15: Schnittstellenvariablen Funktionsbaustein ML_AxisRestore_Type02

Min.- Max.- und Default-Werte der Eingänge

Die nachfolgenden Wertebereiche und Defaultwerte können an den Baustein angelegt werden:

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL	FALSE	TRUE	FALSE	Kontinuierlich
LoadDefaultParams	BOOL	FALSE	TRUE	FALSE	Steigende Flanke an "Execute"

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ParamType	ML_BACKUP_PARAMS			ASP_PARAMS	Steigende Flanke an "Execute"
RestoreFilename	STRING			"	Steigende Flanke an "Execute"
BackupErrors	POINTER TO ML_BACKUP_ERRORS	> 0			Kontinuierlich
BackupErrorsNbr	UINT	0	65535	0	Kontinuierlich

Tab. 3-16: Min-/Max- und Defaultwerte Funktionsbaustein ML_AxisRestore_Type02

Funktionsbeschreibung

Der Funktionsbaustein ML_AxisRestore_Type02 lädt die Parameter aus einer textbasierten Datei auf der Kompakt-Flash-Karte der Steuerung auf eine Achse. Nach erfolgreichem Laden der Parameterdatei wird die Konfiguration auf dem Antrieb oder der Achse gespeichert. Sie steht somit auch nach dem Ausschalten wieder zur Verfügung. Die Parameter-Datei kann durch den Funktionsbaustein [ML_AxisBackup_Type02, Seite 41](#), oder durch IndraWorks erzeugt werden.

Der Baustein sichert folgende Parameter **nicht** zurück, auch wenn diese in der Parameterdatei enthalten sind:

- alle Parameter, die schreibgeschützt sind
- alle Parameter, die Kommandos darstellen
- alle Parameter, die zur Zeit schreibgeschützt sind (z.B. durch Passwort geschützt, Schreibschutz da SPS in RUN, etc.)
- und weitere Parameter wie z.B. A-0-0005, A-0-0016 und weitere



Da in Sercos-Phase 4 nicht alle Parameter schreibbar sind und auch das Laden der Defaultwerte nicht möglich ist, sollte der Baustein nur in Sercos-Phase 2 oder im Parametriemodus (PM) ausgeführt werden.

LoadDefaultParams

Ist der Eingang "LoadDefaultParams" gesetzt, so wird vor dem Laden der Parameter ein Laden der Default-Parameter des Antriebs bzw. der Achse durchgeführt. Dies ist vor allem dann sinnvoll, wenn die Parameter-Datei mit den Optionen "MODIFIED_PARAM" oder "TO_SAVE_PARAM" erzeugt wurde.



Das Laden der Default-Parameter dauert pro Achse mehrere Sekunden.



Beim Laden der Default-Parameter für ein IndraDrive wird vor dem Auslösen des Kommandos noch der Parameter P-0-4090 mit dem Wert 16#51 geschrieben. Damit werden die Parameter der Führungskommunikation und der Engineering-Schnittstelle nicht mit Defaultwerten belegt. Sollen weitere und/oder andere Parametergruppen nicht initialisiert werden, so ist das Laden der Default-Parameter selbst durchzuführen.

ParamType

Über den Eingang "ParamType" werden die wiederherzustellenden Parametertypen ausgewählt. Die folgenden Optionen stehen zur Wahl:

- ASP_PARAMS: Parameter im Antrieb und steuerungsinterne Achsparameter
- A_PARAMS: steuerungsinterne Achsparameter
- SP_PARAMS: Parameter im Antrieb

RestoreFilename

Über "RestoreFilename" kann der Name der Datei auf der Kompakt-Flash-Karte der Steuerung angegeben werden. Es sollte immer der gesamte Pfad angegeben werden. Um Hardware unabhängig zu sein, sollte daher die Funktion ML_GetSystemSpecialPath aus der Bibliothek "ML_Base" verwendet werden, um den gewünschten Pfad zu ermitteln.

Unterlagerte Pfade können angegeben werden, müssen dann aber auch vorhanden sein.

Programm:

```
// Variable definition
strPathName: STRING(255);
strPathFile: STRING(255);

// Code
// Get path to USER partition
strPathName := ML_GetSystemSpecialPath(PATH_USER);
strPathFile := concat(strPathName, 'MyFile.par');
// Final string will be (depending on hardware)
//      '/USER//MyFile.par'

// Get path to OEM partition
strPathName := ML_GetSystemSpecialPath(PATH_OEM);
// Add own path within OEM partition (use '/' as folder delimiter)
// Own path must already exist in the directory
strPathName := concat(strPathName, '/myPath/');
// Add filename to path
strPathFile := concat(strPathName, 'MyFile.par');
// Final string will be (depending on hardware)
//      '/OEM/myPath/MyFile.par'
```

"AxisSource", "AxisDestination"

Beim Baustein ML_AxisRestore, der als Grundlage für diesen Baustein diente, gibt es nur einen Eingang "Axis". Dabei werden implizit immer nur die Parameter dieser Achse importiert. Dies ist ausreichend für Parameterdateien, die ohnehin nur die Daten einer Achse enthalten. Für den Baustein ML_AxisRestore_Type02 wird diese Abhängigkeit vermieden. Dadurch ist es möglich, die Parameter einer Achse auf eine andere Achse zu importieren, oder die richtigen Parameter in einer Parameterdatei mit mehreren Datensätzen zu finden. Das ist sinnvoll, wenn z.B. mehrere Geräte in einer Parameterdatei enthalten sind und nur eine bestimmte Achse importiert werden soll. Ein anderer Anwendungsfall ist, nur eine Parameterdatei mit einer Achse zu erstellen, diese aber auf mehrere Achsen zu importieren. "AxisSource", "AxisDestination" definieren die Quelle und das Ziel der Daten. "AxisSource" gibt dabei die logische Adresse der Achse an, von der die Parameter aus der Parameterdatei verwendet werden sollen. "AxisDestination" ist die Achse in der Anlage, auf die die Parameter geladen werden sollen. Um das gleiche Verhalten wie beim Baustein ML_AxisRestore zu erreichen kann man einfach den Wert für "AxisSource" und "AxisDestination" gleich setzen.

Properties "plgnoreSourceAxisNumber" und "pUsedSourceAxisNumber"

Es kann Fälle geben, in denen die "AxisSource" nicht angegeben werden kann. Das ist z.B. dann der Fall, wenn eine Parameterdatei von einer Achse erstellt wurde, die logische Adresse dieser Achse aber nicht mehr bekannt

ML_TechMotionBase.library // MX_TechMotionBase.library

ist. Diese Information ist aber auch in der Parameterdatei enthalten. In diesem Fall kann die Property "plgnoreSourceAxisNumber" vor der Ausführung des Bausteins mit dem Wert "True" geschrieben werden. Der Baustein verwendet dann intern nicht mehr den Eingang "AxisSource", sondern sucht in der Parameterdatei nach der logischen Adresse. Mit anderen Worten, die "AxisSource" muss nicht bekannt sein. In der Property "pUsedSourceAxisNumber" kann nach der Ausführung des Bausteins zurückgelesen werden, welche Adresse der Baustein in der Parameterdatei gefunden und verwendet hat.

Das Verfahren funktioniert nur, sofern in der Parameterdatei nur die Daten einer Achse enthalten sind (bei mehreren enthaltenen Datensätzen wird immer der letzte Datensatz verwendet). Der Default Zustand für die Property "plgnoreSourceAxisNumber" ist "FALSE", damit muss "AxisSource" angegeben werden.

Zeitverhalten Der Funktionsbaustein ML_AxisRestore_Type02 arbeitet asynchron. Nach dem Start durch eine steigende Flanke am Eingang "Execute" liest er die Eingänge "LoadDefaultParams", "ParamType" und "RestoreFilename". Solange der Funktionsbaustein arbeitet, ist der Ausgang "Active" gesetzt. Nach erfolgreichem Abschluss der Operation ist der Ausgang "Done" gesetzt. Tritt ein Fehler auf, wird der Ausgang "Error" gesetzt.

Tritt ein Fehler beim Speichern eines Parameters auf, so fährt der Baustein mit dem nächsten Parameter fort, bis alle Parameter abgearbeitet sind. Am Ausgang werden dann unter dem Pointer "BackupErrors" alle aufgetretenen Fehler angezeigt.

Fehlercodes Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#0000000D	16#00000000	AxisRef liegt außerhalb des gültigen Bereiches
INPUT_RANGE_ERROR	16#00001E01	16#00000000	Ungültiger Eingang "ParamType"
RESOURCE_ERROR	16#00001E02	16#00000000	Die in "ParamType" ausgewählte Parameterart wird von dieser Achse nicht unterstützt
ACCESS_ERROR	16#00001E05	16#00000000	"BackupErrors" ist 0 oder "BackupErrorsNbr" ist 0
OTHER_ERROR	16#00001E16	16#xxxxxxx	Die unterlagerte Firmwarefunktionalität lieferte einen Fehler zurück. In Additional2 wird der Fehlercode der Firmwarefunktion geliefert. Ist Additional2 = 16#F00A0008, so sagt dies nur aus, dass beim Import Fehler aufgetreten sind. Die einzelnen Fehler und Parameter, bei denen dies der Fall war, können dann in der angelegten Fehlerstruktur ""BackupErrors"" eingesehen werden.
STATE_MACHINE_ERROR	16#00000006	16#0000FFFx	Unerlaubter Zustand des Funktionsbausteins

Tab. 3-17: Fehlercodes Funktionsbaustein ML_AxisRestore_Type02

3.2.6 ML_BACKUP_OPTIONS

Typdefinitionen Die folgende Definition beschreibt den Aufzählungstyp ML_BACKUP_OPTIONS, der in nachstehenden Funktionsbausteinen verwendet wird.

- [ML_AxisBackup, Seite 35](#)
- [ML_AxisBackup_Type02, Seite 41](#)

Die Typ-Definitionen sind in der Bibliothek enthalten.

Der Datentyp "ML_BACKUP_OPTIONS" ist wie folgt definiert:

Aufzählungselement	Wert	Beschreibung
ALL_PARAM	1	Alle Parameter
MODIFIED_PARAM	2	Vom Default-Wert abweichende Parameter
TO_SAVE_PARAM	3	Zur Wiederherstellung notwendige Parameter

Tab. 3-18: Definition der Aufzählungselemente des Datentyps ML_BACKUP_OPTIONS

3.2.7 ML_BACKUP_PARAMS

Typdefinitionen Die folgende Definition beschreibt den Aufzählungstyp ML_BACKUP_PARAMS, der in nachstehenden Funktionsbausteinen verwendet wird.

- [ML_AxisBackup, Seite 35](#)
- [ML_AxisRestore, Seite 38](#)
- [ML_AxisBackup_Type02, Seite 41](#)
- [ML_AxisRestore_Type02, Seite 44](#)

Die Typ-Definitionen sind in der Bibliothek enthalten.

Der Datentyp "ML_BACKUP_PARAMS" ist wie folgt definiert:

Aufzählungselement	Wert	Beschreibung
ASP_PARAMS	3	Parameter im Antrieb und steuerungsinterne Achsparameter
A_PARAMS	2	steuerungsinterne Achsparameter
SP_PARAMS	1	Parameter im Antrieb

Tab. 3-19: Definition der Aufzählungselemente des Datentyps ML_BACKUP_PARAMS

3.2.8 ML_BACKUP_ERRORS

Typdefinitionen Die folgende Definition beschreibt den Datentyp ML_BACKUP_ERRORS, der in nachstehenden Funktionsbausteinen verwendet wird.

- [ML_AxisBackup_Type02, Seite 41](#)
- [ML_AxisRestore_Type02, Seite 44](#)

Die Typ-Definitionen sind in der Bibliothek enthalten.

Der Datentyp "ML_BACKUP_ERRORS" ist wie folgt definiert:

Strukturelement	Datentyp	Beschreibung
ParameterNumber	MB_IDN	Parameternummer im IDN Format
ErrorCode	DWORD	Fehlercode

Tab. 3-20: Definition des Datentyps ML_BACKUP_ERRORS

3.3 Messwertgeber-Unterstützung

3.3.1 Einführung und Übersicht

Der Funktionsbaustein [MB_DistanceAccumulator](#), [Seite 50](#), summiert einen **beliebigen** relativ gefahrenen Weg mittels eines Gebers auf.

Der Funktionsbaustein [MB_AxisDistanceAccumulator](#), [Seite 51](#), summiert den relativ gefahrenen Weg **einer Achse** mittels eines Gebers auf.

3.3.2 MB_DistanceAccumulator

Kurzbeschreibung Der MB_DistanceAccumulator Funktionsbaustein summiert einen **beliebigen** relativ gefahrenen Weg mittels eines Gebers auf.

Mit der steigenden Flanke des **Enable**-Eingangs wird der gefahrene Weg durch den FB aufsummiert bis der aufsummierte Weg größer als oder gleich des **TotalDistance**-Eingangswerts ist. Dabei wird der **DistanceReached**-Ausgang auf TRUE gesetzt und die Bearbeitung des Funktionsbausteines ist abgeschlossen.

Die aktuelle Geberposition wird mittels des **CurrentPosition**-Eingangs vorgegeben.

 Der Funktionsbaustein "MB_DistanceAccumulator (ML_TechMotion-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-21: Referenztabelle MB_DistanceAccumulator

Interface-Beschreibung

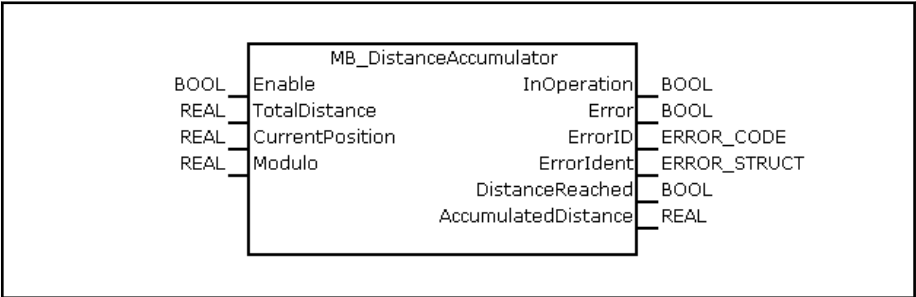


Abb. 3-5: Funktionsbaustein MB_DistanceAccumulator

E/A-Typ	Name	Datentyp	Kommentar
VAR_INPUT	Enable	BOOL	Aktiviert Funktionsbaustein mit steigender Flanke
	TotalDistance	REAL	Zu fahrender Gesamtweg bevor Prozess abgeschlossen ist
	CurrentPosition	REAL	Aktuelle Geberposition
	Modulo	REAL	Modulo Wert: (360: rotatorisch 0: linear) Im Fall einer Servo-Achse kann dieser Wert aus A-0-0045 bzw. S-0-0103 ausgelesen werden

E/A-Typ	Name	Datentyp	Kommentar
VAR_OUTPUT	InOperation	BOOL	Der Funktionsbaustein ist zurzeit aktiv
	Error	BOOL	Prozess mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Kurze Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 51
	DistanceReached	BOOL	TotalDistance wurde erreicht und die Bearbeitung des Funktionsbausteins ist abgeschlossen.
	AccumulatedDistance	REAL	Zeigt den aktuell gefahrenen Weg an

Tab. 3-22: Schnittstellenvariablen Funktionsbaustein MB_DistanceAccumulator

Fehlerbehandlung

Der Funktionsbaustein generiert folgende Fehlermeldungen in Additional1/ Additional2 für **F_RELATED_TABLE**.

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#0003	16#0000	Der Funktionsbaustein wurde durch einen anderen Funktionsbaustein abgebrochen
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Status der Zustandmaschine

Tab. 3-23: Fehlercodes Funktionsbaustein MB_DistanceAccumulator

3.3.3 MB_AxisDistanceAccumulator

Der MB_AxisDistanceAccumulator Funktionsbaustein summiert den relativ gefahrenen Weg einer Achse mittels eines Gebers auf.

Mit der steigenden Flanke des **Enable** Eingangs wird der gefahrene Weg durch den Geber aufsummiert, bis der aufsummierte Weg größer als oder gleich des **TotalDistance** Eingangswerts ist. Dabei wird der **DistanceReached** Ausgang auf TRUE gesetzt und der Funktionsbaustein hält an. Dieser Funktionsbaustein verwendet den Achsreferenzeingang (AXIS_REF), um die aktuelle Geberposition automatisch abzufragen.



Der Funktionsbaustein "MB_AxisDistanceAccumulator (ML_TechMotion-Bibliothek)" wurde mit Version 14VRS in die "TechMotion-Base-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-24: Referenztable MB_AxisDistanceAccumulator

Schnittstellenbeschreibung

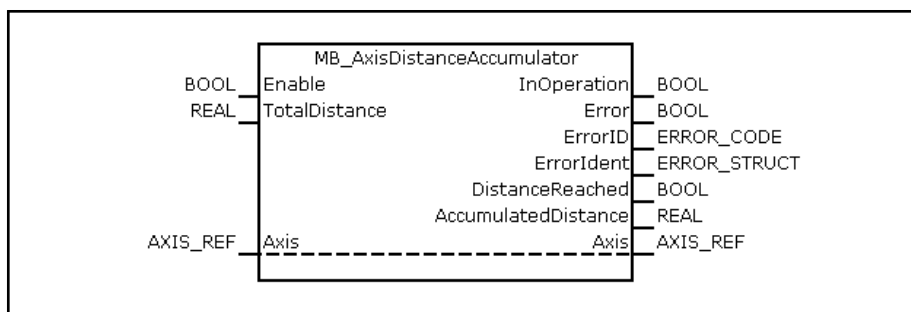


Abb. 3-6: Funktionsbaustein MB_AxisDistanceAccumulator

E/A-Typ	Name	Datentyp	Kommentar
VAR_INPUT	Enable	BOOL	Aktiviert Funktionsbaustein mit steigender Flanke
	TotalDistance	REAL	Zu fahrender Gesamtweg bevor Prozess abgeschlossen ist
VAR_OUTPUT	InOperation	BOOL	Der Funktionsbaustein ist zurzeit aktiv
	Error	BOOL	Prozess mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Kurze Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 52
	DistanceReached	BOOL	TotalDistance wurde erreicht und die Bearbeitung des Funktionsbausteins ist abgeschlossen
	AccumulatedDistance	REAL	Zeigt den aktuell gefahrenen Weg an
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Achsreferenz

Tab. 3-25: Schnittstellenvariablen Funktionsbaustein MB_AxisDistanceAccumulator

Fehlerbehandlung

Der Funktionsbaustein generiert folgende Fehlermeldungen in Additional1/Additional2 für "F_RELATED_TABLE":

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#0003	16#0000	Der Funktionsbaustein wurde durch einen anderen Funktionsbaustein abgebrochen
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Status der Zustandmaschine

Tab. 3-26: Fehlercodes Funktionsbaustein MB_AxisDistanceAccumulator

3.4 Nockenschaltwerk, Anwendung

3.4.1 Einführung und Übersicht

Der Funktionsbaustein zur Anwendung "Nockenschaltwerk" ist ein PLCopen-konformer Baustein ([MC_DigitalCamSwitch, Seite 53](#)).

Mit diesem Funktionsbaustein können bis zu 8 Ausgänge mit bis zu 32 Nocken frei programmiert werden.

Der Funktionsbaustein [MB_DigitalCamSwitchOvs, Seite 71](#) implementiert ein frei programmierbares Nockenschaltwerk mit bis zu 64 Nocken auf 32 Nockenscheiben. Im Unterschied zum Funktionsbaustein MC_DigitalCamS-

witch unterstützt er Oversampling. Dadurch kann eine entsprechend höhere zeitliche Auflösung der Nockenschaltwerkssignale erreicht werden. Die Bausteine sind mit einem EA-Modul zu verwenden, das Oversampling unterstützt, z.B. dem Sercos-Block-EA-Modul R-ILB S3 24 DI16 DIO16/F.

Datenstrukturen

- [MC_OUTPUT_REF](#), Seite 70
- [MC_CAMSWITCH_REF](#), Seite 69
- [ML_BASIC_CAMSWITCH_REF](#), Seite 69
- [MB_CAMSWITCH_REF_OVS](#), Seite 97
- [ML_BASIC_CAMSWITCH_REF_OVS](#), Seite 98
- [MC_TRACK_REF](#), Seite 69
- [ML_BASIC_TRACK_REF](#), Seite 70
- [MB_TRACK_REF_OVS](#), Seite 98
- [ML_BASIC_TRACK_REF_OVS](#), Seite 98

3.4.2 MC_DigitalCamSwitch

Kurzbeschreibung

Der Funktionsbaustein MC_DigitalCamSwitch implementiert ein frei programmierbares Nockenschaltwerk mit bis zu 32 Nocken auf 8 Ausgängen.



Der Funktionsbaustein "MC_DigitalCamSwitch (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS	ML_TechMotionBase
Funktionspaket MOT	
MLD (IL2G)	MX_TechMotionBase

Tab. 3-27: Referenztable Funktionsbaustein MC_DigitalCamSwitch

Schnittstellenbeschreibung

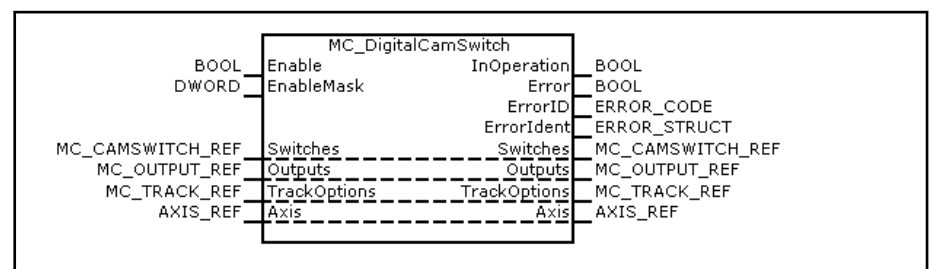


Abb. 3-7: Funktionsbaustein MC_DigitalCamSwitch

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	aktiviert die Schaltausgänge
	EnableMask	DWORD	32 Bits vom Typ BOOL. Aktiviert die verschiedenen Nockenscheiben. Das niederwertigste Bit ist der niedrigsten "TrackNumber" zugeordnet. Wird das Datum auf 1, bzw. auf TRUE gesetzt, ist die zugeordnete "TrackNumber" aktiviert. Da nur die 8 niederwertigsten Bits zur Verfügung stehen, ist die Anzahl auf 8 Nockenscheiben pro Funktionsbaustein begrenzt. Ist das Bit nicht gesetzt, wird der entsprechende Ausgang auf 0 gesetzt
VAR_OUTPUT	InOperation	BOOL	Die aktivierten Nockenscheiben werden berechnet und die Ausgänge aktualisiert
	Error	BOOL	Zeigt an, dass ein Fehler innerhalb des Funktionsbausteins aufgetreten ist
	ErrorID	ERROR_CODE	Fehlerkennung
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler. Fehlercodes, Seite 68
VAR_IN_OUT	Switches	MC_CAMSWITCH_REF	Referenz zur Nockenstruktur, die alle Definitionen für alle Nocken enthält. MC_CAMSWITCH_REF, Seite 69
	Outputs	MC_OUTPUT_REF	Referenz zu den Signalausgängen direkt bezogen auf die zugehörigen Nockenscheiben (max. 8 pro Funktionsbaustein, erster Ausgang = erste "TrackNumber"). MC_OUTPUT_REF, Seite 70
	TrackOptions	MC_TRACK_REF	Referenz zur Struktur, die die zugehörigen Nockenscheibeneigenschaften enthält, z.B. die Ein/Aus-Vorhaltezeit pro Ausgang/Nockenscheibe. MC_TRACK_REF, Seite 69
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Referenz zur Achse, der das Nockenschaltwerk zugeordnet ist

Tab. 3-28: Schnittstellenvariablen Funktionsbaustein MC_DigitalCamSwitch

Funktionalität

Der folgende Funktionsbaustein MC_DigitalCamSwitch bietet eine PLCopen-spezifische (Teil 2) Schnittstelle für ein programmierbares Nockenschaltwerk. Der FB berechnet 8 binäre Nockenscheibenausgänge entsprechend der unten gezeigten Spezifikation.

Jeder binäre Ausgang ist einer Nockenscheibe zugeordnet. Eine Nockenscheibe kann mehrere Ein/Aus-Positionen entsprechend der folgenden Spezifikation enthalten. Es sind maximal 32 Nocken verfügbar die frei auf die 8 Nockenscheiben verteilt werden können.

Terminologie

Bezeichnung	Beschreibung
MLC/XLC	BRC Rack-basierende MotionLogic Control
Digital cam switch	Programmierbares Nockenschaltwerk (Nockenschaltwerk)

*Tab. 3-29: Terminologie des Nockenschaltwerkes***Einstellungen**

Die Einheiten für Position und Geschwindigkeit müssen für die Achse, die dem Nockenschaltwerk zugeordnet sind, entsprechend eingestellt sein. Wenn z.B. die Positionen der Achse in Grad definiert sind, werden die Nockenschaltwerk-Positionen ebenfalls in Grad interpretiert.

Datentypen

Der Funktionsbaustein "MC_DigitalCamSwitch" nutzt folgende Datentypen:

Eine Nockenscheibe ist ein Element, das die Eigenschaften des Nockenschaltwerkes bezogen auf einen Ausgang beschreibt.

- [MC_TRACK_REF, Seite 69](#), enthält ein Array mit 8 Basis-Strukturen (d.h. 8 Nockenscheiben, die 8 Ausgängen zugeordnet sind). Jeder Nockenscheibe können mehrere Nocken zugeordnet sein
- [ML_BASIC_TRACK_REF, Seite 70](#)
- [MC_CAMSWITCH_REF, Seite 69](#), enthält ein Array mit 32 Basis-Strukturen, die jede eine einzelne Nocke beschreibt
- [ML_BASIC_CAMSWITCH_REF, Seite 69](#)
- [MC_OUTPUT_REF, Seite 70](#), enthält ein Array mit 8 booleschen Variablen, die den Ausgängen des Nockenschaltwerkes zugeordnet sind

Beispielprogramm

Ein kurzes Beispiel soll das Konzept erklären, wie der Funktionsbaustein programmiert wird. Die verschiedenen Strukturelemente werden später im Detail erklärt.

Angenommen, eine einzelne Nocke soll angelegt werden:

- mit Nummer 4 auf Nockenscheibe 3
- mit einer "FirstOnPosition" von 35 und
- einer "LastOnPosition" von 55
- die Nocke soll 10ms früher auslösen und die Nockenscheibe sollte eine Hysterese von 3 haben
- die Nocke soll positionsbezogen, nicht zeitbezogen, ein/ausschalten und nur in positiver Richtung wirksam sein

Die Variablendeklaration und der Funktionsbaustein-Aufruf sehen dann folgendermaßen aus:

ML_TechMotionBase.library // MX_TechMotionBase.library

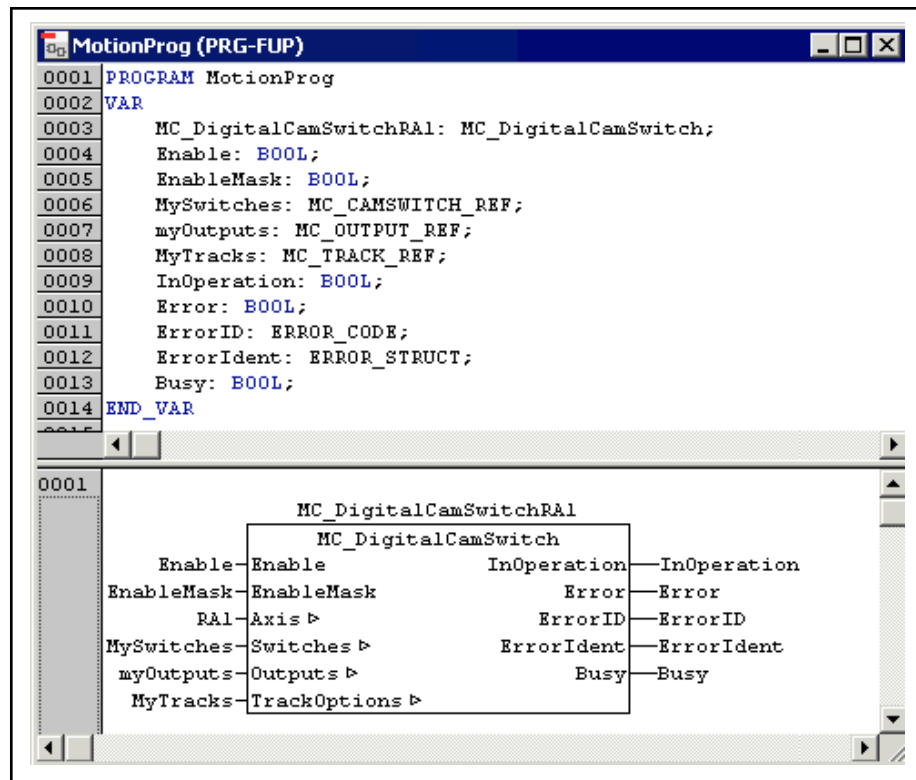


Abb. 3-8: Deklaration der Variablen und Funktionsbaustein-Aufruf

Beachten Sie die drei Eingangs-Strukturen "MySwitches", "MyOutputs" und "MyTracks", die alle Eigenschaften der Nocken, Nockenscheiben und Ausgänge definieren. Um diese Strukturen besser verstehen zu können, ist eine nähere Betrachtung der Daten-Inhalte der Strukturen, welche die gewünschten Ergebnisse des obigen Beispiels liefern, notwendig:

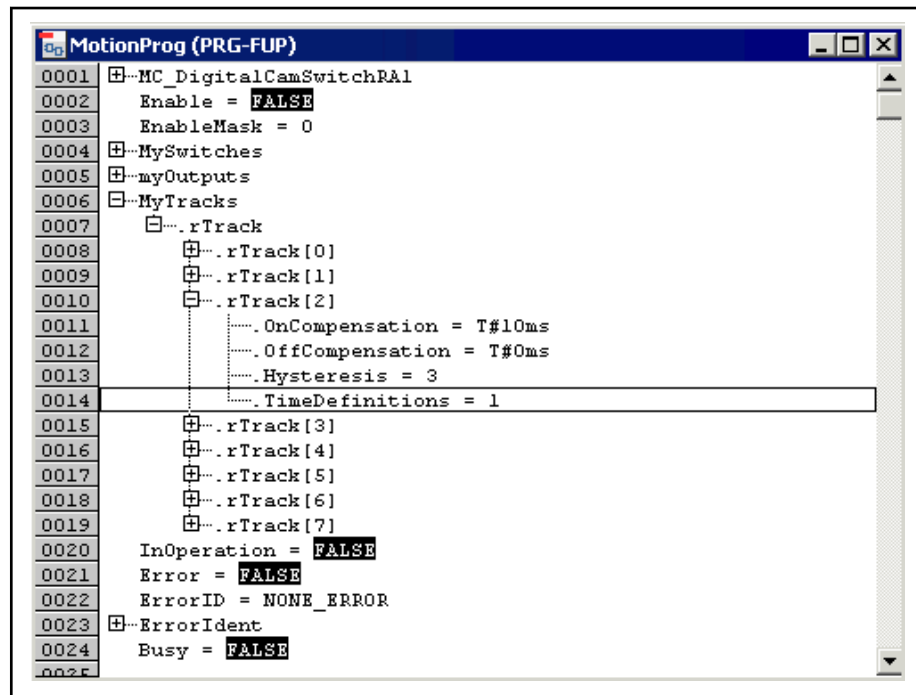


Abb. 3-9: Inhalt der MyTracks-Struktur

Zuerst wird die Nockenscheibe definiert (Bild oben).

1. Da Nockenscheibe 3 benutzt werden soll, muss **rTrack[2]** initialisiert werden, da das Feld mit 0 beginnt.
2. Die OnCompensation-Zeit wird auf die gewünschten 10ms gesetzt. Damit die Nocke früher und nicht später schaltet, muss Bit 0 in "TimeDefinitions" gesetzt werden, welches die "OnCompensation"-Zeit als negativ definiert.
3. Zum Schluss muss noch die Hysterese auf den gewünschten Wert 3 gesetzt werden.

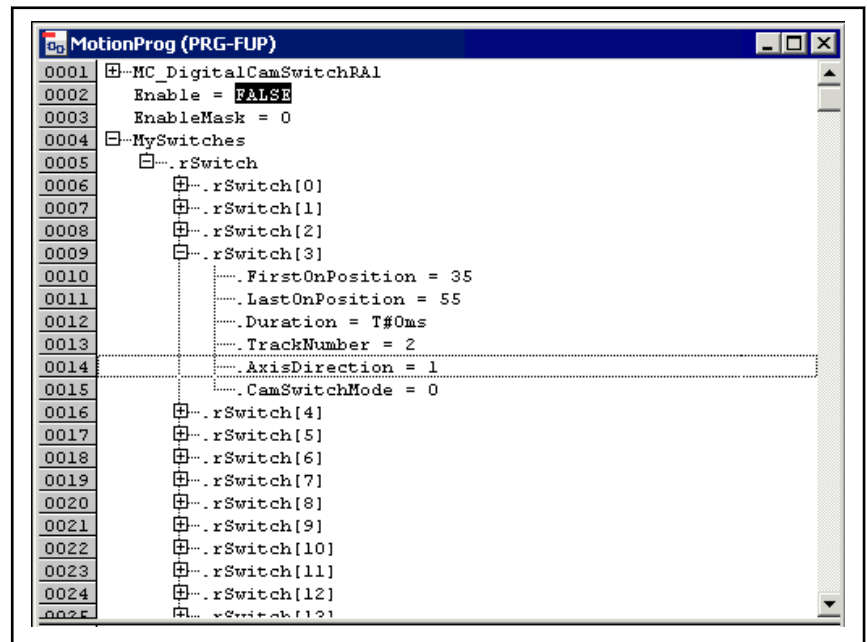


Abb. 3-10: Inhalt der MySwitches-Struktur

Nun wird die Nocke definiert (Bild oben).

1. Die Initialisierung der Nocken-Struktur muss mit Index 3 erfolgen, um Nocke 4 zu initialisieren, da das Feld mit Index 0 beginnt.
2. "FirstOnPosition" wird auf den gewünschten Wert 35 gesetzt, "LastOnPosition" auf 55 und die "AxisDirection" wird auf 1 gesetzt, was bedeutet, dass die Nocke nur in positiver Richtung wirksam ist.
3. Um die Nocke einer Nockenscheibe zuzuordnen, wird TrackNumber auf 2 gesetzt; damit wird festgelegt, dass diese Nocke der Nockenscheibe mit Index 2 in der Nockenscheiben-Struktur zugeordnet ist.
4. Zum Schluss muss noch der "CamSwitchMode" auf 0 gesetzt werden, um den positionsbezogenen Mode einzuschalten.



Der Zugriff auf die einzelnen Elemente der REF-Strukturen erfolgt nach nachstehendem Muster:

- MyTracks.rTrack[2].TimeDefinitions
- MySwitches.rSwitches[3].AxisDirection
- MyOutputs.Output[4]

Der Start des Funktionsbausteines wird vorbereitet:

Damit der Funktionsbaustein gestartet werden kann, müssen die restlichen Variablen zum Funktionsbaustein hinzugefügt und ihre Werte entsprechend

ML_TechMotionBase.library // MX_TechMotionBase.library

gesetzt werden. Speziell der "Enable"-Eingang muss gesetzt werden und "EnableMask" muss das entsprechende Bit gesetzt haben:

2#0000100, Bit 2 für dritte Nockenscheibe rTrack[2].

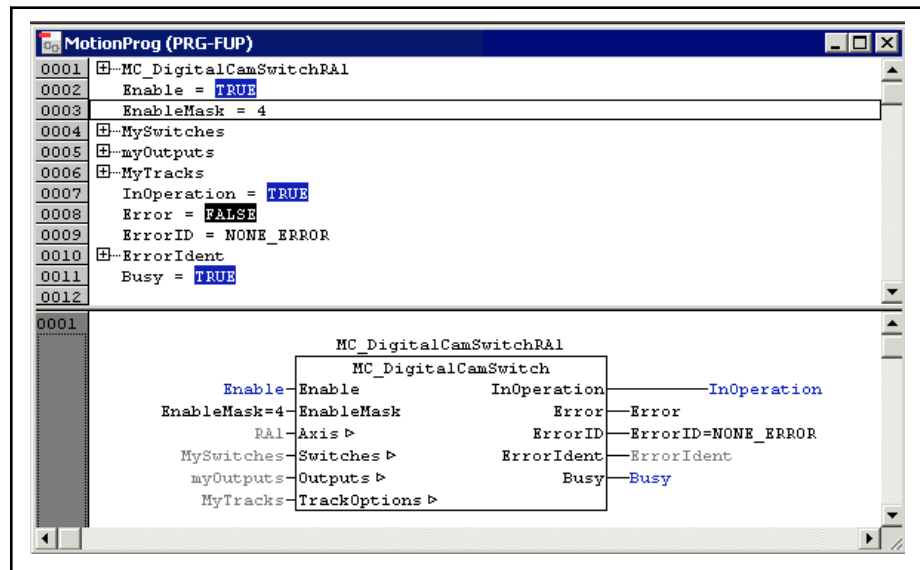


Abb. 3-11: Laufender Funktionsbaustein

Funktionsbeschreibung

Der **Axis-Eingang** definiert die Position und Geschwindigkeit die für die Berechnungen benötigt wird. Wechselt der "Axis"-Eingang während der "Enable"-Eingang noch gesetzt ist, so wird ein Fehler erzeugt.

Wird der **Enable**-Eingang das erste Mal gesetzt, wird auch der "Busy"-Ausgang gesetzt, der "InOperation"-Ausgang bleibt jedoch inaktiv. Dies geschieht ebenfalls aufgrund der Tatsache, dass einige interne Werte zuerst aktualisiert werden müssen, bevor die Berechnung fortgeführt wird. Dies benötigt einige SPS-Takte. Nach dieser Initialisierungssequenz wird der "InOperation"-Ausgang wieder aktiv.

Mit dem **EnableMask**-Eingang können die Ausgänge schnell freigegeben oder gesperrt werden. Das niederwertigste Bit ist der niedrigsten "TrackNumber" zugeordnet. Wird der Wert auf 1, bzw. TRUE gesetzt, dann ist die entsprechende "TrackNumber" freigegeben. Durch die Beschränkung auf die 8 niederwertigsten Bits stehen nur 8 Nockenscheiben pro Funktionsbaustein zur Verfügung. Ist das Bit nicht gesetzt, wird der entsprechende Ausgang auf 0 gesetzt, ungeachtet dessen, ob die aktuelle Position innerhalb des Bereiches einer Nocke liegt. Wenn "EnableMask" 0 ist, wird "InOperation" inaktiv, da keiner der Ausgänge freigegeben sind.

Jede Nocke hat eine Start- und Endposition. Dadurch kann der Benutzer für jede Nocke eine **FirstOnPosition** und eine **LastOnPosition** (oder Zeit) festlegen. Zusätzlich zu einer mechanischen Nocke besteht die Möglichkeit, die Nocke für eine bestimmte Zeit zu aktivieren und ihr für eine bestimmte Zeit eine Vorhaltezeit und Hysterese zuzuordnen. Wenn ("FirstOnPosition" > "LastOnPosition") ist, verhält sich die Nocke invers und ist nur ausgeschaltet innerhalb dieser Positionen.

Bereich/Modulo-Berechnungen

Das Nockenschaltwerk wird höchstwahrscheinlich nur für rotatorische Abläufe eingesetzt. Dennoch könnte man das Nockenschaltwerk auch für eine Linienachse, die vorwärts und rückwärts läuft, einsetzen. In beiden Fällen basieren die Nockenschaltwerk-Einheiten auf die für die Achse definierten Einheiten.

AxisDirection gibt an, in welcher Bewegungsrichtung die Nocke aktiv wird. Ist "AxisDirection" 0 (default), wird die Nocke in beiden Bewegungsrichtungen aktiv. Ist "AxisDirection" 1, wird der Ausgang nur gesetzt, wenn in positiver Richtung verfahren wird; ist "AxisDirection" auf 2 gesetzt, wird der Ausgang nur gesetzt, wenn in negative Richtung verfahren wird.

CamSwitchMode wechselt zwischen Positions- oder Zeitangabe. "CamSwitchMode" = 0 bedeutet Positionsmodus, dabei schaltet der Ausgang an der "FirstOnPosition" ein und an der "LastOnPosition" aus. Ist "CamSwitchMode" = 1, wird der Ausgang an der "FirstOnPosition" eingeschaltet, und nach der angegebenen "Duration" (Dauer) wieder ausgeschaltet.

Duration: Zeitdauer, die eine zeitgesteuerte Nocke eingeschaltet ist. Wenn die Position nach Ablauf dieser Zeit noch innerhalb des Nockenbereichs ist, wird der Ausgang nicht neu gesetzt, bis die Position die Nocke ausgeschaltet hat.

Die Vorhaltezeit (**OnCompensation** und **OffCompensation**) kann positiv oder negativ sein. Negativ bedeutet, dass der Ausgang schaltet bevor die Schaltposition erreicht wird. Die Vorhaltezeit wird in ms angegeben (siehe auch "TimeDef").

Hysteresis: Dieser Parameter verhindert, dass der Ausgang zu oft schaltet, wenn die Achse nahe dem Schaltpunkt ist und die aktuelle Position um den Schaltpunkt schwankt. Die Hysteresis ist ein Teil von "MC_TRACK_REF", was bedeutet, dass für jede Nockenscheibe eine eigene Hysteresis einstellbar ist.

Sowohl in Vorwärts- als auch Rückwärtsrichtung muss zunächst die Hysteresis durchlaufen werden, bevor die Nocke schaltet.

Im folgenden Bild sind zwei Nocken jeweils von 150° bis 180° definiert.

Die obere Kurve zeigt den Ausgang der Nocke ohne Hysteresis, die mittlere Kurve zeigt den Ausgang der Nocke mit einer Hysteresis von 10°. Die untere Kurve stellt die Masterposition dar.

Der Master fährt zunächst in positiver Richtung über die Nocken, anschließend in negativer und schließlich wieder in positiver Richtung. Dabei ist zu erkennen, dass die Nocke mit Hysteresis jeweils um den Betrag der Hysteresis verspätet eingeschaltet wird, unabhängig davon aus welcher Richtung die Nocke durchfahren wird.

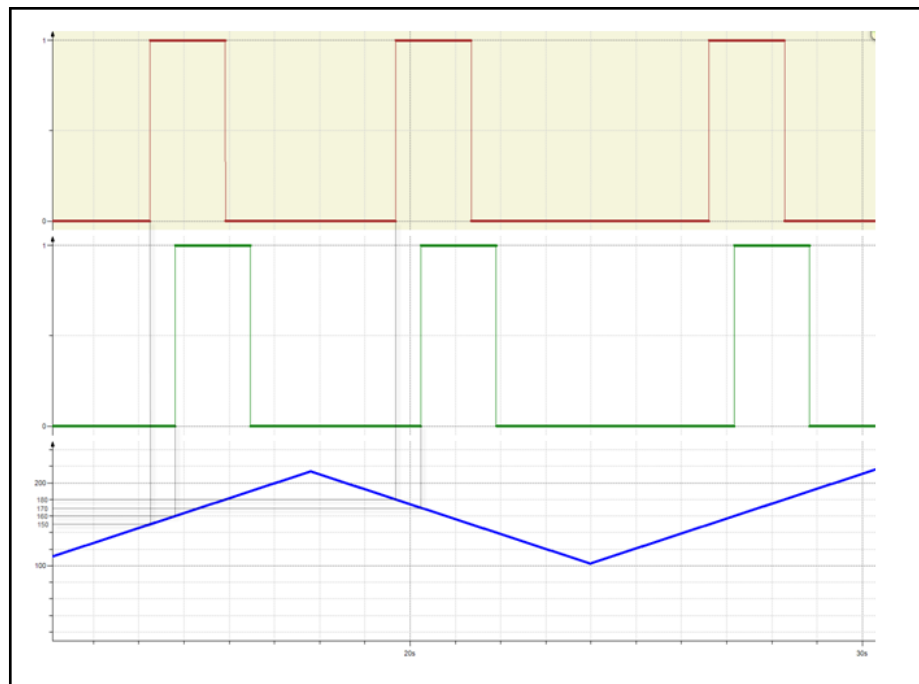


Abb. 3-12: Auswirkung der Hysterese auf die Schaltpunkte einer Nocke

TimeDef: Dieser Parameter wurde hinzugefügt, um zwischen positiven und negativen Zeiten für die "On/OffCompensation" zu unterscheiden. Wenn Bit 0 gesetzt ist, dann ist die "OnCompensation"-Zeit negativ, ist Bit 1 gesetzt, dann ist die "OffCompensation"-Zeit negativ.

Beispiel:

"OffCompensation" ist auf 300ms gesetzt, "TimeDef" auf 2.

Das bedeutet, dass die Vorhaltezeit "-300ms" ist.

Die folgende Abbildung zeigt die verschiedenen Variablen und Strukturen und ihre entsprechende Benutzung:

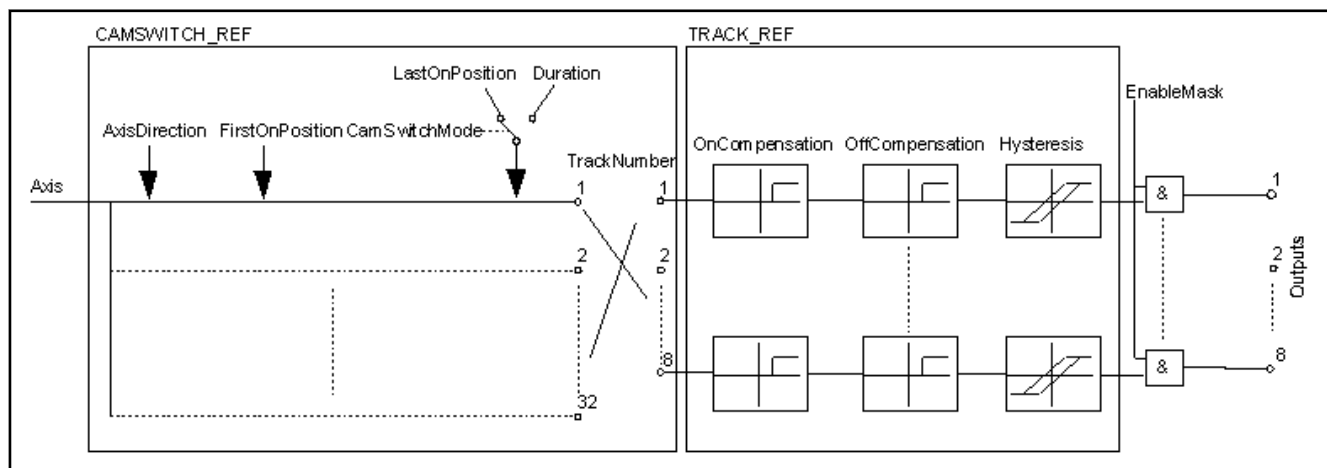


Abb. 3-13: Beziehung der Strukturen und entsprechende Funktionalität des Bausteins MC_DigitalCamSwitch

Das folgende Bild zeigt das mechanische Äquivalent des digitalen Nockenschaltwerks.

Beachten Sie bitte, dass nicht alle Funktionen dargestellt wurden, da sie entweder nicht möglich, bzw. schwer darstellbar sind.

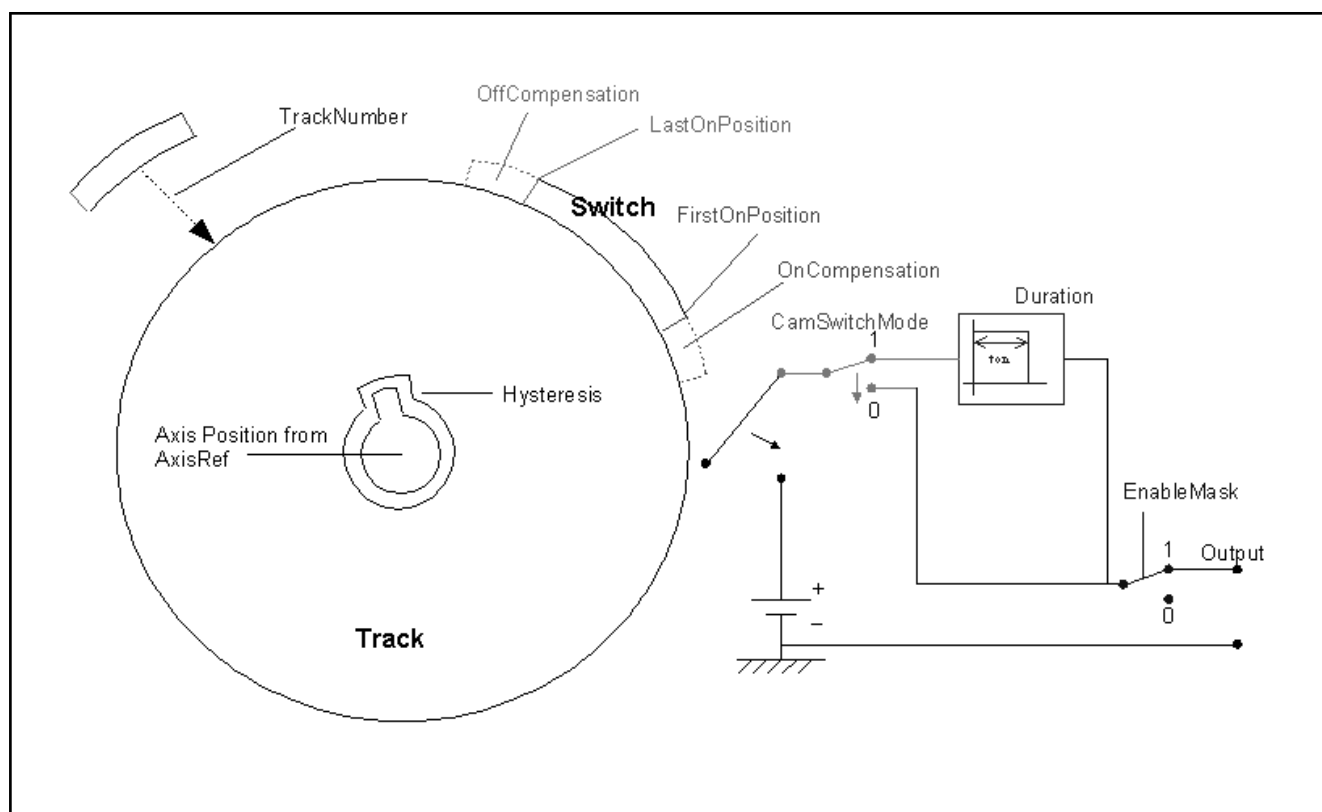


Abb. 3-14: Mechanisches Äquivalent des digitalen Nockenschaltwerks

Zeitdiagramm

Das folgende Beispiel soll die Benutzung der "MC_CAMSWITCH_REF"-Struktur und der zeitbasierenden Nocken zeigen. In der folgenden Liste finden Sie eine vorgegebene Konfiguration von Werten:

Parameter	Typ	Switch01	Switch02	Switch03	Switch04	...	SwitchN
FirstOnPosition [u]	REAL	2000	2500	4000	3000		
LastOnPosition [u]	REAL	3000	3000	1000	-		
Duration	TIME	-	-	-	t#1350ms		
TrackNumber	INT	1	1	1	2		
AxisDirection	INT	1=Pos	2=Neg	0=Beide	0=Beide		
CamSwitchMode	INT	0=Position	0=Position	0=Position	1=Zeit		

Tab. 3-30: Beispiel MC_CAMSWITCH_REF-Struktur



Werte werden in Einheiten [u] angegeben. Diese beziehen sich auf die zugehörige Achse.

Ist z. B. eine Einschaltposition bei 20u und die zugehörige Achse wird in Grad gewichtet, ist die Einschaltposition entsprechend 20°.

Dieses Beispiel benutzt die Werte aus dem obigen Beispiel "MC_CAMSWITCH_REF". Es benutzt weder "On/OffCompensation" noch "Hysteresis".

Dies ist das Verhalten der Ausgänge, wenn die Achse sich kontinuierlich in positive Richtung bewegt. Die Achse ist eine Modulo-Achse mit einer Modulolänge von 5000u (Einheiten).

ML_TechMotionBase.library // MX_TechMotionBase.library

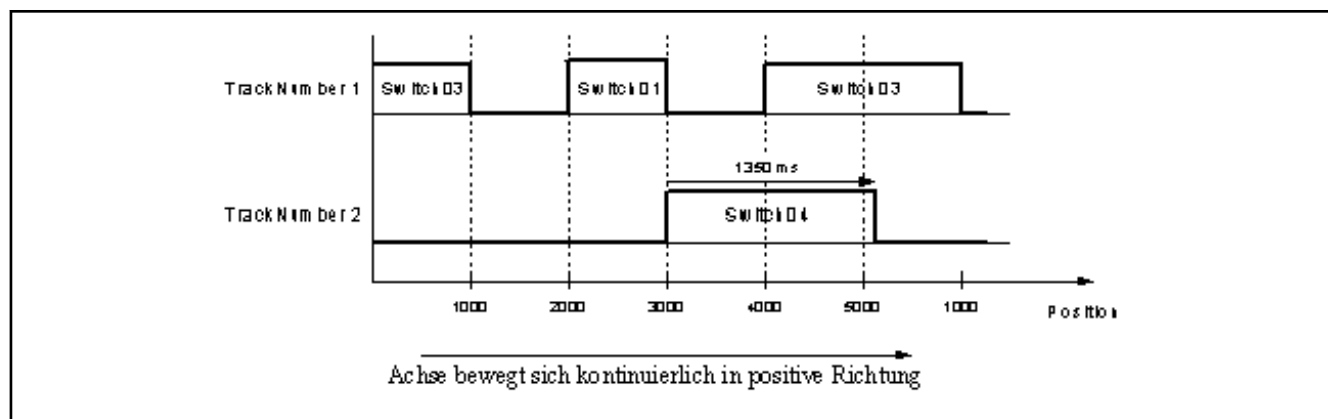


Abb. 3-15: Verhalten der Ausgänge im Beispiel

Detaillierte Beschreibung der Nocke "Switch01"

Dieses Beispiel benutzt zusätzlich "OnCompensation" = -125ms und "OffCompensation" = +250ms.

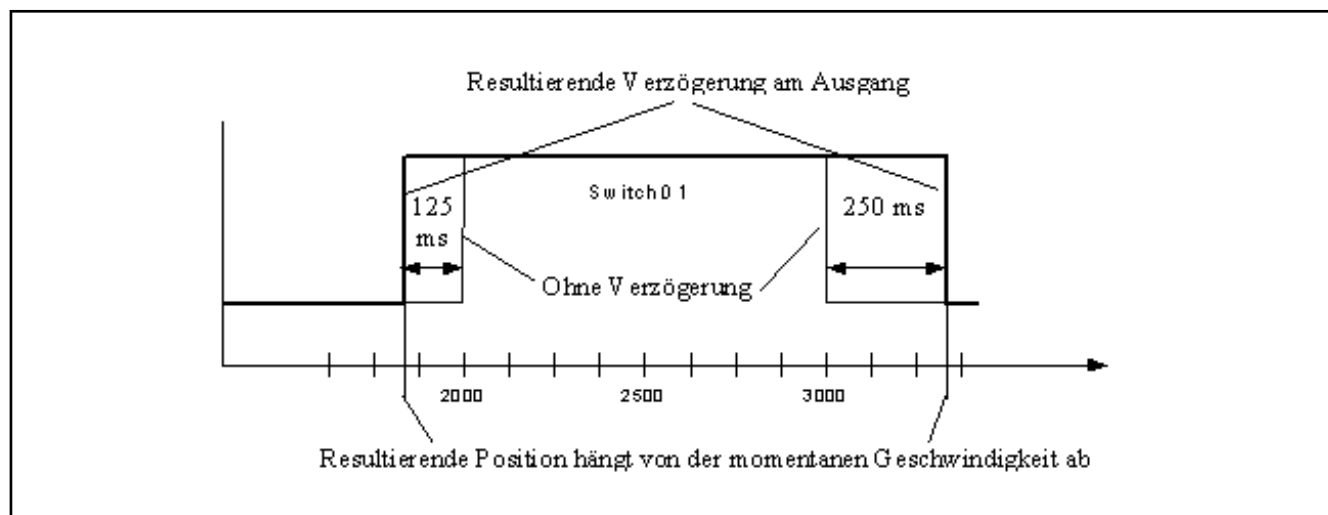


Abb. 3-16: Detailliertes Verhalten von Switch01 beim Bewegen in positive Richtung

Dies ist das Verhalten der Ausgänge, wenn sich die Achse ohne "On-" oder "OffCompensation" und ohne "Hysteresis" kontinuierlich in negative Richtung bewegt:

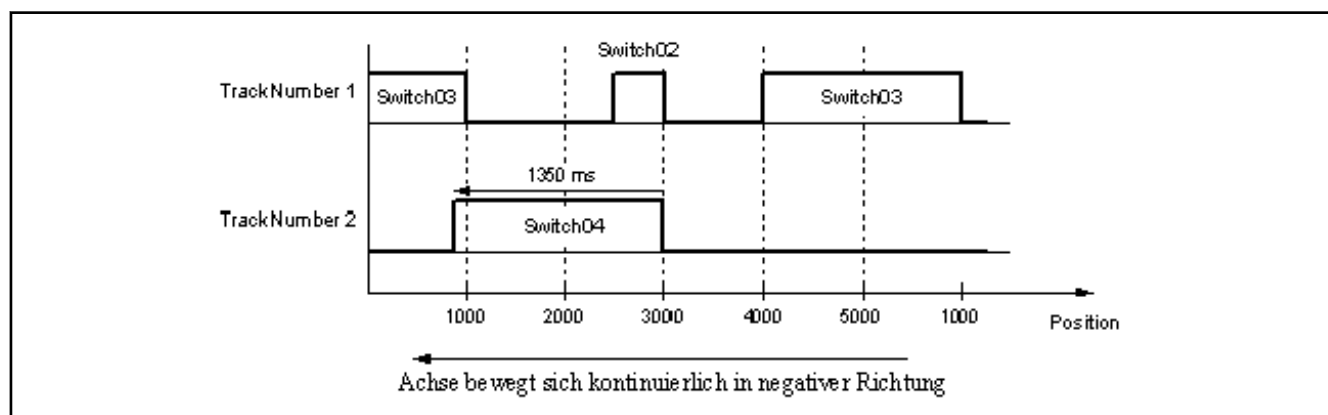


Abb. 3-17: Detailliertes Verhalten der Nocken beim Bewegen in negative Richtung

Realisierungsdetails

Die PLCOpen-Erweiterung spezifiziert das Interface der Nockenschaltwerk-Funktionalität. Jedoch sind einige Anwendungsdetails nicht spezifiziert. Deshalb wurden diese für eine Durchführung mit besserem Laufzeitverhalten definiert.

OnCompensation/OffCompensation

PLCOpen definiert die "On-/OffCompensation" nur für Bewegungen in positive Richtung. In diesem Fall bezieht sich die "OnCompensation" auf die "FirstOnPosition", während die "OffCompensation" sich auf die "LastOnPosition" bezieht. Wie ist dies zu definieren für Bewegungen in negative Richtung?

Basierend auf Applikationen bei denen die Vorhaltezeiten benutzt werden, um bei Aktuatoren die An- und Ausschaltverzögerungen zu kompensieren bedeutet dies, dass die "OnCompensation" für die steigende Flanke des Signals zuständig ist. Mit anderen Worten: fährt man in positive Richtung, dann bezieht sich die "OnCompensation" auf die "**FirstOnPosition**", fährt man in negative Richtung, dann bezieht sich die "OnCompensation" auf die "**LastOnPosition**". Die "OffCompensation" ist entsprechend definiert

Die folgende Abbildung zeigt das Verhalten für "OnCompensation"= -0.1s und "OffCompensation"= +0.2s:

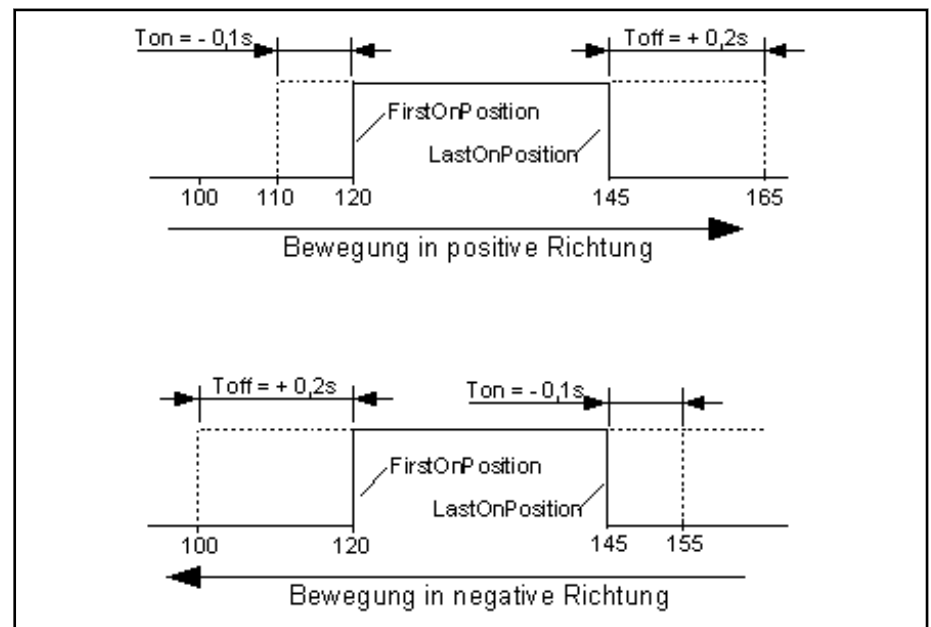


Abb. 3-18: Definition der OnCompensation- und OffCompensation-Zeit

Duration

Die "Duration" (Dauer) ist definiert als die Zeit, die die Nocke eingeschaltet bleibt, nachdem sie eingeschaltet wurde. Die Frage stellt sich was passiert, wenn die Zeitdauer abgelaufen ist, aber die momentane Position noch innerhalb der "FirstOnPosition" und "LastOnPosition" ist. Soll der Ausgang sofort eingeschaltet, oder soll gewartet werden, bis die aktuelle Position sich aus der "FirstOnPosition" und "LastOnPosition" heraus bewegt hat?

Es wurde entschieden auf eine steigende Flanke des Nockenausgangs zu warten. Mit anderen Worten: die aktuelle Position muss außerhalb der "FirstOnPosition" und "LastOnPosition" sein, bevor die "Duration" wieder neu angestoßen werden kann.

Die gleiche Überlegung gilt für die "OnCompensation"/"OffCompensation", wobei die "Duration" mit der steigenden Flanke einer Nocke beginnt, d.h. an der "LastOnPosition" beim Bewegen in negativer Richtung.

Der Sachverhalt wird in der folgenden Abbildung erläutert:

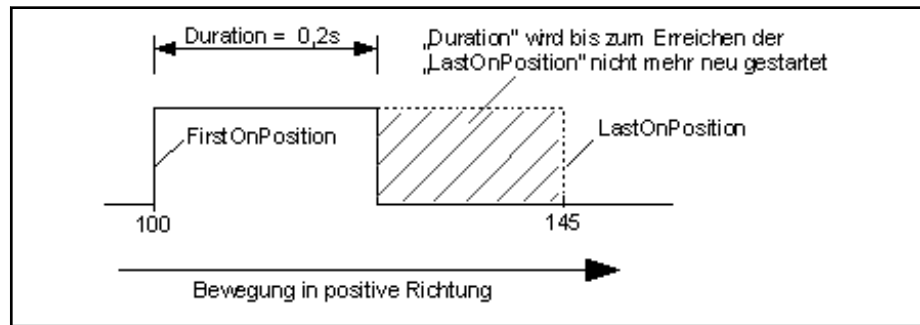


Abb. 3-19: Definition der Zeitdauer

Kombination von OnCompensation/OffCompensation und Duration

Sollen bei der Benutzung einer zeit-basierenden Nocke mit "On-/OffCompensation"-Zeiten diese dann berücksichtigt werden oder nicht? Die "OnCompensation" wird berücksichtigt. Daraus resultiert eine frühere/spätere Aktivierung des Ausgangs. Die "OffCompensation"-Zeit ist allerdings schwieriger, da der Ausgang nach der Vorhaltezeit ("Duration") ausgeschaltet wird. Die "OffCompensation"-Zeit ist nur relevant, wenn man bestimmen muss, dass der Ausgang regetriggert werden soll (schraffierter Bereich im oberen Bild).

In dieser Anwendung wurden beide, "OnCompensation" und "OffCompensation", in Betracht gezogen.

Wie eine Nocke nicht verwendet wird

Das Nockenschaltwerk hat bis zu 32 Nocken. Um festzustellen, ob ein Nocken benutzt wird oder nicht gibt es das Element "TrackNumber" in der MC_CAMSWITCH_REF-Struktur.

Diese Variable beinhaltet die Nockenscheiben-Nummer. Gültige Werte dafür sind 0 bis 7, da es maximal 8 Nockenscheiben gibt. Die Default-Einstellung dieser Variable ist 8, sie wird als nicht benutzt markiert. Jede ungültige Nummer (8 bis 65535) wird interpretiert als eine nicht benutzte Nocke und wird deshalb auch nicht von der Firmware ausgewertet.

Einschränkungen des Nockenschaltwerkes

Die Einschränkungen des Nockenschaltwerkes werden nachstehend diskutiert. Die Einschränkungen sind lediglich Definitionen, die jedoch zu keinem Fehler führen.

Das Systemverhalten wird über diese Definitionen festgelegt, der Anwender muss deshalb mit unterschiedlichem Verhalten rechnen.

Verlorene Schaltvorgänge

Das Abtast-Theorem gibt an, dass die Abtast-Rate doppelt so hoch ist, wie die höchste Signalfrequenz die man braucht, um das Signal erkennen zu können. Dies gilt auch für das Nockenschaltwerk. Das Nockenschaltwerk tastet die Positionsdaten jedes Mal ab, wenn es von der SPS aufgerufen wird. Das ist normalerweise die Motion-synchrone Task, oder aber jede andere zyklische Task. Jede Nocke die kürzer eingeschaltet ist als die Abtast-Rate, wird nicht richtig erkannt. Die folgenden Gleichungen zeigen, wie man die Geschwindigkeit oder die Größe der Nocke berechnet, bei denen Schaltvorgänge verloren gehen.

Rotatorische Systeme:

$$\omega = 2\pi n \quad (\text{Gleichung 1})$$

n ist die Umdrehung pro Sekunde [1/s].

$$\varphi = \rho / t \quad (\text{Gleichung 2})$$

ρ ist der Winkel in rad und t die Zeit in Sekunden [s].

Konvertierungsfaktor von Grad to rad: $\pi / 180$

Nimmt man Gleichungen 1 und 2 erhält man

$$\rho [\text{rad}] = 2\pi n t$$

für den Winkel, oder

$$n \left[\frac{\text{Umdrehungen}}{\text{Sekunde}} \right] = \frac{\rho}{2\pi t}$$

Passt man die Faktoren für gemeinsame Einheiten an, so erhält man:

$$\rho [^\circ] = n [1/\text{min}] * t [\text{ms}] * 0,006 \quad (\text{Gleichung A1})$$

oder für die Geschwindigkeit:

$$n [1/\text{min}] = \frac{\rho [^\circ]}{t [\text{ms}]} * 166,6 \quad (\text{Gleichung B1})$$

Die Gleichungen A1 und B1 liefern dem Anwender die beiden notwendigen Gleichungen, um festzustellen, ob Schaltvorgänge verloren gehen können. Zum Beispiel, wenn die maximale Drehzahl von 1000 min^{-1} bekannt ist, und das Nockenschaltwerk in einer zyklischen Task mit einer Zykluszeit von 2 ms aufgerufen wird. Was ist dann die minimale Größe einer Nocke, um sicherzustellen, dass sie von dem Nockenschaltwerk erkannt wird?

Nimmt man Gleichung A1 ergibt das:

$$\rho [^\circ] = 1000 \text{ min}^{-1} * 2 \text{ ms} * 0,006 = 12$$

D.h., eine Nocke muss mindestens 12° groß sein, um bei einer Drehzahl von max. 1000 min^{-1} noch sicher erkannt zu werden.

Lineare Systeme:

$$v = s / t$$

s ist die Distanz in Millimeter [mm] und t die Zeit in Sekunden [s].

Nimmt man die Zykluszeit der SPS als Abtastzeit für das Nockenschaltwerk und addiert die entsprechenden Faktoren um die Einheiten anzugleichen, so erhält man die folgenden Gleichungen:

$$s [\text{mm}] = v \left[\frac{\text{mm}}{\text{min}} \right] * t [\text{ms}] * 1,66 * 10^{-5} \quad (\text{Gleichung A2})$$

oder für die Geschwindigkeit:

$$v \left[\frac{mm}{min} \right] = \frac{s [mm]}{t [ms]} * 60000 \quad (\text{Gleichung B2}) .$$

Diese Gleichungen gelten auch für Systeme die Inch [in] als Einheit für den Weg:

$$s [in] = v \left[\frac{in}{min} \right] * t [ms] * 1,66 * 10^{-5} \quad (\text{Gleichung A2})$$

oder für die Geschwindigkeit haben:

$$v \left[\frac{in}{min} \right] = \frac{s [in]}{t [ms]} * 60000 \quad (\text{Gleichung B2})$$

Beispiel: Eine Nocke in einem linearen System ist 0,02 in lang. Das Nockenschaltwerk ist in einer SPS-Task mit 2ms Zykluszeit. Welche maximale Geschwindigkeit ist möglich, damit die Nocke noch sicher erkannt wird?

Nimmt man Gleichung B2 erhält man das Ergebnis:

$$v \left[\frac{in}{min} \right] = \frac{0,02in}{2ms} * 60000 \frac{1}{min} = 600$$

Nocken haben scheinbar die falsche Größe

Voranstehend wurden die Randbedingungen für "verlorene Schaltvorgänge" beschrieben.

Auch wenn die Bedingungen für verlorene Nocken nicht erfüllt sind, können die Schaltzeitpunkte ungenau sein. Das Nockenschaltwerk kann nur dann den Zustand seiner Ausgänge wechseln, wenn es von der SPS aufgerufen wird. Wenn z.B. eine Nocke für 3,7 ms bei einer vorgegebenen Geschwindigkeit, geschlossen sein soll, das Nockenschaltwerk aber nur alle 2 ms aufgerufen wird, wird der Ausgang entweder für 2 ms (Fall B) oder 4 ms (Fall A) auf "On" gesetzt, abhängig von dem Abtastzeitpunkt des Nockenschaltwerkes. Eine nähere Erläuterung hierzu sehen Sie in der folgende Abbildung

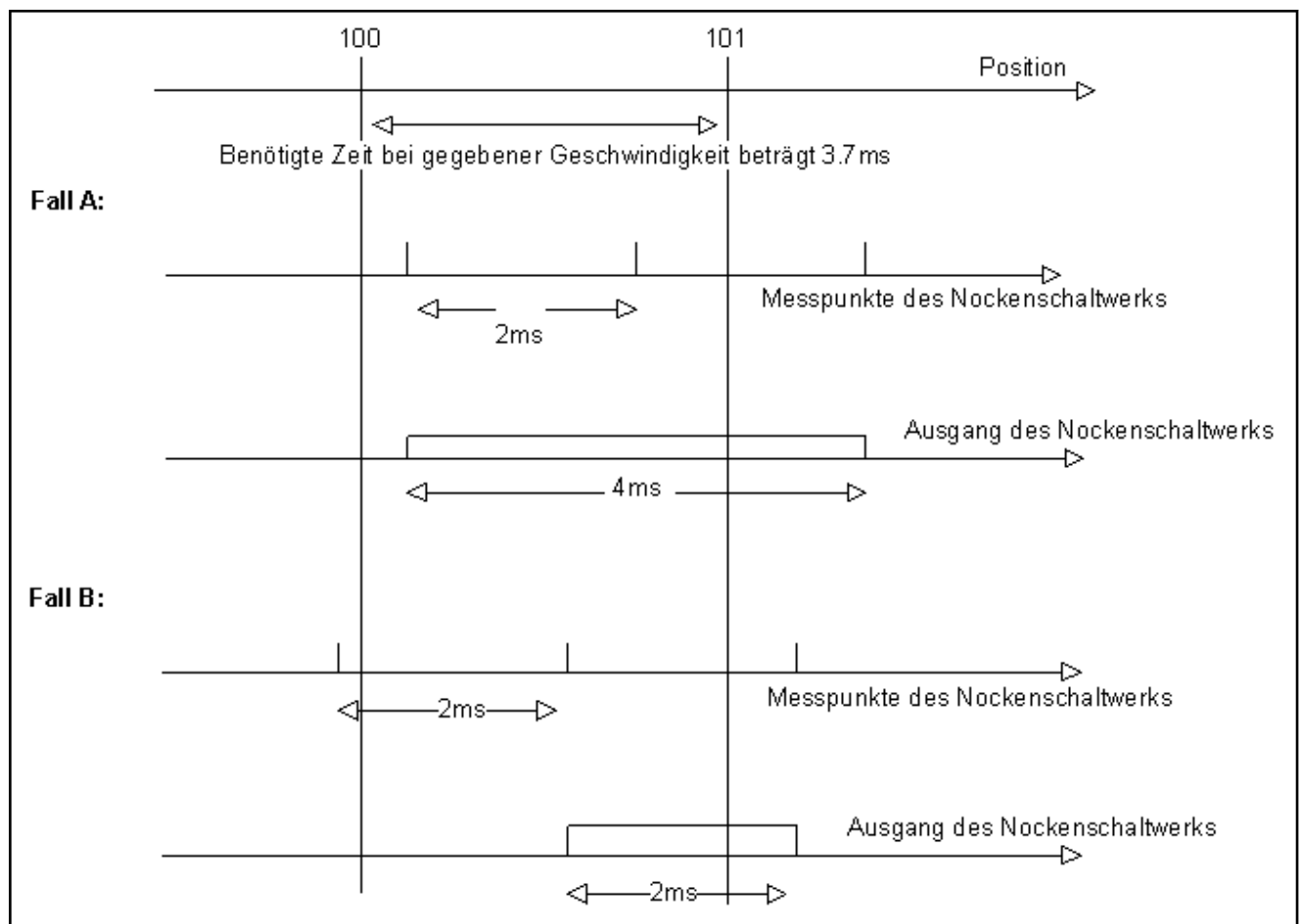


Abb. 3-20: Ungenaue Größe der Nocken

Schaltvorgang erfolgt zweimal
beim Beschleunigen/Verzögern

Werden "OnCompensation" und "OffCompensation" für eine Nocke benutzt, ist es möglich, dass sie mehrfach ein- und ausgeschaltet wird, wenn die Nockenschaltwerk Masterposition beschleunigt oder abbremst. Dies passiert aufgrund der Tatsache, dass das Nockenschaltwerk eine konstante Geschwindigkeit voraussetzt, was unter bestimmten Umständen jedoch nicht gegeben ist. Das größere Problem taucht beim Abbremsen auf. Einige Betrachtungen, wann eine Nocke zu mehreren Zeiten aktiviert werden kann, folgt.

Man kann die folgenden Gleichungen aufstellen:

$$s_v = v_0 * t_{lead} \quad (\text{Gleichung 3})$$

dabei ist s_v der Weg, der bei der vorgegebenen Geschwindigkeit v_0 in der "OnCompensation"-Zeit t_{lead} zurückgelegt wird.

$$s_b = v_0 * t - \frac{1}{2} a t^2 \quad (\text{Gleichung 4})$$

s_b ist der Weg, der nötig ist zum Abbremsen bei einer vorgegebenen Startgeschwindigkeit v_0 und Verzögerung a in der Zeit t .

Außerdem die allgemeinen Gleichungen:

$$v = a * t \text{ oder } t = v / a$$

Die "OnCompensation"-Zeit eines programmierbaren Nockenschaltwerks "bewegt" die Nocke in Richtung niedriger Positionen entsprechend Gleichung 3.

Der Weg, der benötigt wird, um bei einer vorgegebenen Geschwindigkeit zum Stillstand zu kommen, kann mit Gleichung 4 ermittelt werden.

Die allgemeine Regel lautet deshalb:

Wenn die Position mit $v = v_0$ erreicht ist, damit der Ausgang eines programmierbaren Nockenschaltwerks mit "OnCompensation"-Zeit eingeschaltet werden soll – und die Eingangsposition in einer Weise zum Stoppen abgebremst wird, dass die Stopposition gleichzeitig auch die Einschaltposition des programmierbaren Nockenschaltwerks ohne "OnCompensation"-Zeit ($v = 0$) ist, dann wird der Ausgang dazwischen nicht ausgeschaltet.

Wenn wir Gleichung 3 mit Gleichung 4 gleichsetzen, erhalten wir die Gleichung, mit der die Nocke nur einmal eingeschaltet wird. Wenn wir etwas schneller abbremsen, dann wird die Nocke mehrfach eingeschaltet.

$$s_b = s_v \Rightarrow v_0 t - \frac{1}{2} a t^2 = v_0 t_{\text{lead}} \Rightarrow v_0 \frac{v_0}{a} - \frac{1}{2} a \frac{v_0^2}{a^2} = v_0 t_{\text{lead}}$$

$$\Rightarrow \frac{v_0^2}{2a} = v_0 t_{\text{lead}}$$

$$\Rightarrow v_0 = 2 t_{\text{lead}} a \quad (\text{Gleichung 5})$$

In Gleichung 5 kann man sehen, dass die Startgeschwindigkeit v_0 bei der es möglich ist, dass ein Nockenschalter mehrfach eingeschaltet wird, größer wird durch Erhöhung der "OnCompensation"-Zeit t_{lead} oder Verzögerung a .

Es gibt keine Lösung für dieses Problem. Man kann lediglich die "OnCompensation"-Zeit und Verzögerung für die gewünschte Applikation so klein wie möglich halten.

Eine andere Möglichkeit ist es, die "Hysterese" zu benutzen, damit die Schwankungen aufgrund der Beschleunigung/Verzögerung darin berücksichtigt werden.

Die gleichen Überlegungen können angewandt werden für das Beschleunigen, und/oder negative Geschwindigkeiten.

Fehlercodes Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in Additional1/ Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#0006	16#00xx	Ungültiger Zustand des Funktionsbausteins xx bezeichnet den ungültigen Zustand
RESSOURCE_ERROR	16#0009	16#0000	Achsreferenznummer hat sich während der Bausteinausführung geändert

ErrorID	Additional1	Additional2	Beschreibung
RESSOURCE_ERROR	16#0012	16#00xx	Steuerungsnummer ist ungültig xx bezeichnet die Steuerungsnummer
INPUT_INVALIDID_ERROR	16#2B00	16#0000	Ungültige Kombination von Einheiten für Geschwindigkeit, Zeit und Position

Tab. 3-31: Fehlercodes Funktionsbaustein MC_DigitalCamSwitch

3.4.3 MC_CAMSWITCH_REF

Typdefinitionen Die folgende Definition beschreibt Datentypen, die im Funktionsbaustein [MC_DigitalCamSwitch, Seite 53](#), benutzt werden.

Der Datentyp "MC_CAMSWITCH_REF" enthält ein Array von 32 Basis-Strukturen, die jede eine einzelne Nocke beschreibt ([ML_BASIC_CAMSWITCH_REF, Seite 69](#)).

Typdefinition

```

TYPE MC_CAMSWITCH_REF :
STRUCT
// Dies ist ein ARRAY von Basisstrukturen, eine für jede Nocke.
// Die Zählung beginnt mit 0!
// D.h. die erste Nocke wird adressiert durch rSwitch[0].xxxx
rSwitch : ARRAY [0..31] OF ML_BASIC_CAMSWITCH_REF;
END_STRUCT
END_TYPE

```

3.4.4 ML_BASIC_CAMSWITCH_REF

Typdefinitionen Die folgende Definition beschreibt Datentypen, die im Funktionsbaustein [MC_DigitalCamSwitch, Seite 53](#), benutzt werden.

Der Datentyp [MC_CAMSWITCH_REF, Seite 69](#), enthält ein Array von 32 Basis-Strukturen, die jede eine einzelne Nocke beschreibt.

Werte werden in Einheiten [u] angegeben. Diese beziehen sich auf die zugehörige Achse.

Ist z. B. eine Einschaltposition bei 20u und die zugehörige Achse wird in Grad gewichtet, ist die Einschaltposition entsprechend 20°.

Die Basis-Struktur "ML_BASIC_CAMSWITCH_REF" sieht wie folgt aus:

Strukturelement	Typ	Beschreibung
FirstOnPosition [u]	REAL	Untere Grenze bei der der Ausgang AN ist (Einschaltposition)
LastOnPosition [u]	REAL	Obere Grenze bei der der Ausgang AN ist (Ausschaltposition)
Duration	TIME	Gekoppelt mit zeitbezogenem "CamSwitchMode". D.h., der Ausgang bleibt für die angegebene Zeit an
TrackNumber	INT	"TrackNumber" ist die Referenz zur Nockenscheibe
AxisDirection	INT	Richtung in der das Nockenschaltwerk wirksam ist: beide (=0; Default); Positiv (1); Negativ (2)
CamSwitchMode	INT	Positionsbezogen (=0; Default); zeitbezogen (=1)

Tab. 3-32: Definition der Basiselemente der ML_BASIC_CAMSWITCH_REF-Struktur

3.4.5 MC_TRACK_REF

Typdefinitionen Die folgende Definition beschreibt Datentypen, die im Funktionsbaustein [MC_DigitalCamSwitch, Seite 53](#), verwendet werden.

ML_TechMotionBase.library // MX_TechMotionBase.library

Eine Nockenscheibe ist ein Element, das die Eigenschaften des Nockenschaltwerkes bezogen auf einen Ausgang beschreibt.

Der Datentyp "MC_TRACK_REF" enthält ein Array mit 8 Basis-Strukturen, d.h., 8 Nockenscheiben, die 8 Ausgängen zugeordnet sind ([ML_BASIC_TRACK_REF, Seite 70](#)).

Typdefinition

```
TYPE MC_TRACK_REF :
STRUCT
// Dies ist ein ARRAY von Basisstrukturen, eine für jede Nockenscheibe.
// Die Zählung beginnt mit 0!
// D.h. die erste Nocke wird adressiert durch rTrack[0].xxxx
rTrack: ARRAY [0..7] OF ML_BASIC_TRACK_REF;
END_STRUCT
END_TYPE
```

Jeder Nockenscheibe können mehrere Nocken zugeordnet sein.

3.4.6 ML_BASIC_TRACK_REF

Typdefinitionen Die folgende Definition beschreibt Datentypen, die im Funktionsbaustein [MC_DigitalCamSwitch, Seite 53](#), verwendet werden.

Eine Nockenscheibe ist ein Element, das die Eigenschaften des Nockenschaltwerkes bezogen auf einen Ausgang beschreibt.

Der Datentyp [MC_TRACK_REF, Seite 69](#), enthält ein Array mit 8 Basis-Strukturen (d.h., 8 Nockenscheiben, die 8 Ausgängen zugeordnet sind).

Die Basis-Struktur ("ML_BASIC_TRACK_REF") einer Nockenscheibe sieht wie folgt aus:

Strukturelement	Typ	Beschreibung
OnCompensation	TIME	Vorhaltezeit der Nockenscheibe bezogen auf den Einschaltzeitpunkt.
OffCompensation	TIME	Vorhaltezeit der Nockenscheibe bezogen auf den Ausschaltzeitpunkt.
Hysteresis [u]	REAL	Einstellbare Hysterese um mit Rauschen behaftete Eingangssignale zu filtern.
TimeDef	WORD	Zeitdefinition: ¹⁾ Bit 0 gesetzt: "OnCompensation"-Zeit ist negativ; Bit 1 gesetzt: "OffCompensation"-Zeit ist negativ

Tab. 3-33: Definition der Basiselemente der MC_TRACK_REF-Struktur

Jeder Nockenscheibe können mehrere Nocken zugeordnet sein.

3.4.7 MC_OUTPUT_REF

Typdefinitionen Die folgende Definition beschreibt Datentypen, die im Funktionsbaustein [MC_DigitalCamSwitch, Seite 53](#), verwendet werden. Die Definitionen der Datentypen sind in der Bibliothek enthalten.

Der Datentyp beschreibt die 8 Ausgänge des Nockenschaltwerkes.

"MC_OUTPUT" enthält ein ARRAY mit 8 Booleschen Variablen, die den Ausgängen zugeordnet sind und sieht wie folgt aus:

Strukturelement	Typ	Beschreibung
OUTPUT	ARRAY [0..7] OF BOOL	Ausgänge des Nockenschaltwerkes.

Tab. 3-34: Definition der Basiselemente der MC_OUTPUT_REF-Struktur

¹⁾ Eine negative Zeit bedeutet, dass der Ausgang eher schaltet, eine positive Zeit, dass er später gemessen am Schaltzeitpunkt, schaltet.

3.4.8 MB_DigitalCamSwitchOvs

Kurzbeschreibung Der Funktionsbaustein MB_DigitalCamSwitchOvs implementiert ein frei programmierbares Nockenschaltwerk mit bis zu 64 Nocken auf 64 Nockenscheiben.

Im Unterschied zum Funktionsbaustein MC_DigitalCamSwitch unterstützt der Baustein MB_DigitalCamSwitchOvs Oversampling ([MC_DigitalCamSwitch](#), Seite 53). Dadurch kann eine entsprechend höhere zeitliche Auflösung der Nockenschaltwerkssignale erreicht werden.



Die Bausteine sind mit einem EA-Modul zu verwenden, das Oversampling unterstützt, z.B. mit dem Sercos-Block-EA-Modul R-ILB S3 24 DI16 DIO16/F.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-35: Referenztablette Funktionsbaustein MB_DigitalCamSwitchOvs

Schnittstellenbeschreibung

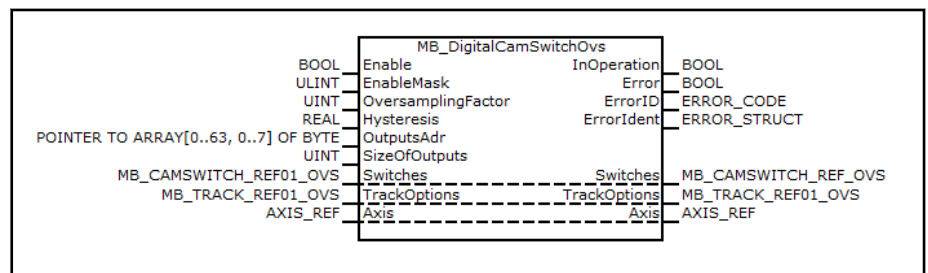


Abb. 3-21: Funktionsbaustein MB_DigitalCamSwitchOvs

EA-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert die Schaltausgänge
	EnableMask	ULINT	64 Bits vom Typ BOOL. Aktiviert die verschiedenen Nockenscheiben. Das niederwertigste Bit ist der niedrigsten "TrackNumber" zugeordnet. Wird das Datum auf 1, bzw. auf TRUE gesetzt, ist die zugeordnete "TrackNumber" aktiviert. Ist das Bit nicht gesetzt, wird der entsprechende Bitstrom der Nockenscheibe permanent auf 0 gesetzt.
	OversamplingFactor	UINT	Faktor für das Oversampling: Gibt an, wie hoch die Anzahl der Abtastungen in einem Sercos-Zyklus ist und bestimmt damit in Abhängigkeit von der Achsgeschwindigkeit und der Sercos-Zykluszeit die bestmögliche Positionsauflösung bei positionsbezogenen Nocken. Der Oversampling-Faktor bestimmt somit auch, wie oft ein Ausgang (eine Nockenscheibe) seinen Zustand in einem Sercos-Takt bis zum nächsten Funktionsbausteinaufruf maximal ändern kann.
	Hysteresis	REAL	Einstellbare Hysterese, um mit Rauschen behaftete Eingangssignale zu filtern.

EA-Typ	Name	Datentyp	Beschreibung
	OutputsAdr	POINTER TO ARRAY [0..63,0..7] OF BYTE	Pointer auf das Ausgangsdatenfeld: Referenz zu den Signalausgängen, direkt bezogen auf die zugehörigen Nockenscheiben. Erster Feldindex: Gibt die Nockenscheibe an. Zweiter Feldindex: Für jede Nockenscheibe gibt es maximal 64 Zustände in einem Sercos-Zyklus: 8 Bytes mit jeweils 8 Bit. Das niederwertigste Bit in Byte 0 entspricht dem ersten Zustand der Nockenscheibe während des SPS-Zyklus.
	SizeOfOutputs	UINT	Größe des Ausgangsdatenfeldes: Der Oversampling-Faktor bestimmt die Mindestgröße des Ausgabedatenfeldes. Das Ausgabedatenfeld benötigt maximal 512 Byte: Bei einem Oversampling-Faktor von 64, werden je Nockenscheibe 8 Byte = 64 Bit benötigt. Damit ergibt sich die maximale Ausgangsdatenlänge von 64x8 Byte = 512 Byte.
VAR_OUTPUT	InOperation	BOOL	Die aktivierten Nockenscheiben werden berechnet und die Ausgänge aktualisiert
	Error	BOOL	Zeigt an, dass ein Fehler innerhalb des Funktionsbausteins aufgetreten ist
	ErrorID	ERROR_CODE	Fehlerkennung
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler. Fehlercodes, Seite 95
VAR_IN_OUT	Switches	MB_CAMS- WITCH_REF_OVS	Referenz zur Nockenstruktur, die alle Definitionen für alle Nocken enthält. MB_CAMSWITCH_REF_OVS, Seite 97
	TrackOptions	MB_TRACK_REF_OVS	Referenz zur Struktur, die die zugehörigen Nockeneigenschaften enthält, z.B. die Ein/Aus-Vorhaltezeit pro Ausgang/Nockenscheibe. MB_TRACK_REF_OVS, Seite 98
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Referenz zur Achse, der das Nockenschaltwerk zugeordnet ist

Tab. 3-36: Schnittstellenvariablen Funktionsbaustein MB_DigitalCamSwitchOvs

Hinweis zu Eingang "OutputsAdr"

Je nach Oversamplingfaktor kann das Ausgangsdatenfeld, dessen Adresse am Eingang "OutputsAdr" übergeben wird, eine variable Größe haben. Mit dem Feldindex wird die Nockenscheibe referenziert. Da die Anzahl der Nockenscheiben fest ist (64), muss die Dimension des Ausgangsdatenfelds stets von 0 bis 63 deklariert sein. Die Größe des einzelnen Index des Ausgangsdatenfelds kann je nach Oversamplingfaktor variieren. Beispielsweise gibt es bei 16-fach Oversampling für jede Nockenscheibe 16 Zustände in einem Sercos-Zyklus. Die 16 Zustände können mit einem Word (= 16 Bits) abgebildet werden.

Nachfolgend eine mögliche Deklaration des Ausgangsdatenfelds "MyOutputs" bei einem Oversampling Faktor von 16.

Deklaration des Ausgangsdatenfelds

```

MyOutputs: ARRAY [0..63] OF WORD;           // Output data field
// First index: Specifies the track number,
// second index specifies the number of bits necessary for given oversampling factor
// Example here is oversampling factor of 16, hence,
// 16 bits are necessary for an output -> one word
// For oversampling factor of 2 until 8: MyOutputs: ARRAY [0..63] OF BYTE;
// For oversampling factor of 16:       MyOutputs: ARRAY [0..63] OF WORD;
// For oversampling factor of 32:       MyOutputs: ARRAY [0..63] OF DWORD;
// For oversampling factor of 64:       MyOutputs: ARRAY [0..63] OF LWORD;

```

Falls Sie den Oversamplingfaktor erhöhen wollen, müssen Sie den Datentyp des Ausgangsdatenfelds entsprechend verändern.

Datentypen in Abhängigkeit des Oversampling Faktors

Oversampling Faktor	Basis Datentyp	Deklaration für "MyOutputs"
1, 4, 8	BYTE	MyOutputs: ARRAY [0..63] OF BYTE
16	WORD	MyOutputs: ARRAY [0..63] OF WORD
32	DWORD	MyOutputs: ARRAY [0..63] OF DWORD
64	LWORD	MyOutputs: ARRAY [0..63] OF LWORD

Tab. 3-37: Oversampling Faktoren, Datentypen und Deklaration

Empfehlung

Wenn Sie sich nicht auf einen bestimmten Oversamplingfaktor festlegen wollen, dann verwenden Sie für den Datentyp des Ausgangsdatenfelds stets ein LWORD (64 Bit). Bei dieser Deklaration können Sie den Oversamplingfaktor frei wählen. Falls Sie einen Oversamplingfaktor kleiner 64 verwenden, werden für den jeweiligen Track nur die niederwertigsten Bits (je nach Oversamplingfaktor) beschrieben.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL	FALSE	TRUE	FALSE	Kontinuierlich
EnableMask	ULINT	0	2 ⁶⁴ -1	0	Kontinuierlich
OversamplingFactor	UINT	1	64	1	Positive Flanke an "Enable"
Hysteresis	REAL			0.0	Positive Flanke an "Enable"
OutputsAdr	POINTER TO ARRAY[0..63,0..7] OF BYTE				Kontinuierlich
SizeOfOutputs	UINT	64	512	0	Positive Flanke an "Enable"
Switches	MB_CAMSWITCH_REF_OVS	Siehe Datentypbeschreibung	Siehe Datentypbeschreibung	Siehe Datentypbeschreibung	Kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
TrackOptions	MB_TRACK_REF_OVS	Siehe Datentypbeschreibung	Siehe Datentypbeschreibung	Siehe Datentypbeschreibung	Positive Flanke an "Enable"
Axis	AXIS_REF	CntrlNo: 0 AxisNo: 1	CntrlNo: 0 AxisNo: 512	CntrlNo: 0 AxisNo: 0	Positive Flanke an "Enable"

Tab. 3-38: Min-/Max- und Defaultwerte Funktionsbaustein MB_DigitalCamSwitchOvs

Funktionalität Der Funktionsbaustein MB_DigitalCamSwitchOvs implementiert ein frei programmierbares Nockenschaltwerk mit bis zu 64 Nocken auf 64 Nockenscheiben. Die Nocken können frei auf die 64 Nockenscheiben verteilt werden.

Die Nocken werden nicht als einzelne Bits, sondern als "Bitstrom" ausgegeben. Diesen kann man sich so vorstellen, als würde der Bitstrom mittels eines zeitlich getakteten Schieberegisters an einem Binärausgang mittels Oversampling ausgegeben.

Das Oversampling wird mit Hilfe des "Oversampling-Faktors" (Eingang "OversamplingFactor") vorgegeben. Dieser bezieht sich auf die Sercos-Zykluszeit.

Es werden Oversampling-Faktoren von 2^n mit $n = [0..6]$ zugelassen. Das ergibt die Faktoren 1, 4, 8, 16, 32 und 64. Bei einem Oversampling-Faktor von 1 findet kein Oversampling statt.



Da das Sercos-Block E/A-Modul R-ILB S3 24 DI16 DIO16/F den Oversampling-Faktor 2 nicht unterstützt, wird der Faktor 2 auch vom Funktionsbaustein nicht zugelassen.

Einstellungen Die Einheiten für Position und Geschwindigkeit müssen für die Achse, die dem Nockenschaltwerk zugeordnet sind, entsprechend eingestellt sein. Wenn z. B. die Positionen der Achse in Grad definiert sind, werden die Nockenschaltwerk-Positionen ebenfalls in Grad interpretiert.

Für die korrekte Unterstützung der Oversampling-Funktionalität mit einem oversamplingfähigen EA-Modul müssen die Funktionsbausteine zwingend in einer Sercos-synchronen Task aufgerufen werden.

Für eine **MLC/XLC** sind dazu in der Taskkonfiguration der Typ "Extern" und als externes Ereignis die Einstellung "EVENT_OP_MODE_SERCOS_CYCLIC" zu wählen.

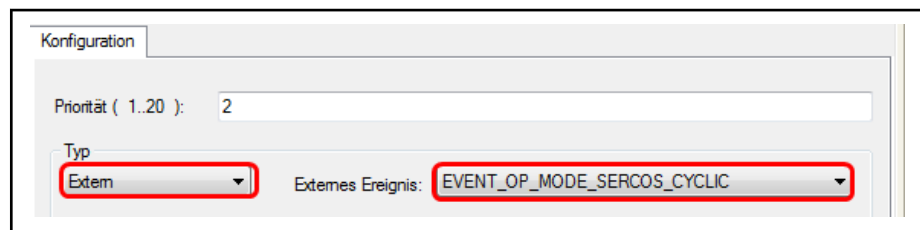


Abb. 3-22: Voreinstellung Sercos-synchroner Task bei MLC/XLC

Bei einer **MLD** sind die Einstellungen in der Taskkonfiguration entsprechend: Typ "Extern" und als externes Ereignis die Einstellung "CCD_SYNCHRONIZED_TASK".

Wird der Baustein nicht in einer Sercos-synchronen Task aufgerufen, so wird der Fehler 16#2B03 an den Fehlerausgängen des Bausteins erzeugt.

Bei einer MLC/XLC müssen die Motion-Zykluszeit (C-0-0400) und die Sercos-Zykluszeit (C-0-0503) den gleichen Wert haben. Den Dialog zur Einstellung der Sercos-Zykluszeit bzw. der Motion-Zykluszeit erreichen Sie über das Kontextmenü des Motion-Knotens im Projektexplorer.

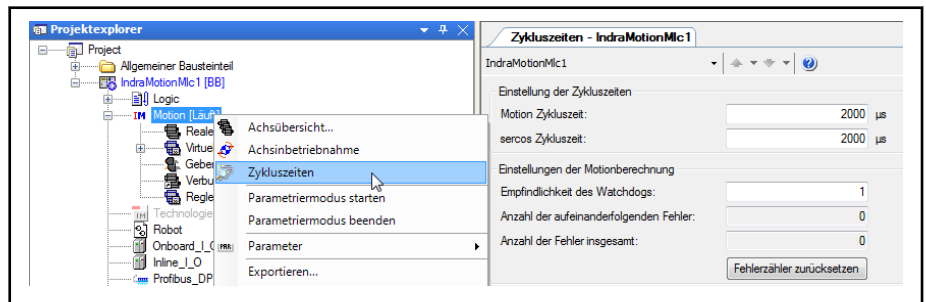


Abb. 3-23: Dialog zur Einstellung der Sercos-Zykluszeit bzw. der Motion Zykluszeit bei einer MLC/XLC

Haben Motion Zykluszeit und Sercos-Zykluszeit unterschiedliche Werte, so wird der Fehler 16#2B09 an den Fehlerausgängen des Bausteins erzeugt.

Datentypen

Der Funktionsbaustein MB_DigitalCamSwitchOvs nutzt folgende Datentypen:

- [MB_TRACK_REF_OVS, Seite 98](#)

Der Datentyp MB_TRACK_REF_OVS enthält ein Array mit 64 Basis-Strukturen, d. h., 64 Nockenscheiben, die 64 oversamplingfähigen Ausgängen zugeordnet werden können.

Jeder Nockenscheibe können mehrere Nocken zugeordnet sein.

- [ML_BASIC_TRACK_REF_OVS, Seite 98](#)

Eine Nockenscheibe ist ein Element, das die Eigenschaften des Nockenschaltwerkes bezogen auf einen oversamplingfähigen Ausgang beschreibt.

Die Basis-Struktur ML_BASIC_TRACK_REF_OVS enthält Elemente, mit der das Verhalten für eine Nockenscheibe eingestellt werden kann.

- [MB_CAMSWITCH_REF_OVS, Seite 97](#)

Der Datentyp MB_CAMSWITCH_REF_OVS enthält ein Array mit 64 Basis-Strukturen, die jede eine einzelne Nocke beschreibt.

- [ML_BASIC_CAMSWITCH_REF_OVS, Seite 98](#)

Die Basis-Struktur ML_BASIC_TRACK_REF_OVS enthält Elemente, mit denen das Verhalten für eine einzelne Nocke eingestellt werden kann.

Beispiel:

Funktionsbeschreibung

Mit Hilfe des "Axis"-Eingang besteht Zugriff auf die aktuelle Achsposition und -geschwindigkeit, die für die Berechnungen benötigt werden.

Nach einer steigenden Flanke am "Enable"-Eingang werden alle Werte an den Bausteineingängen übernommen und die bausteininterne Initialisierungsphase gestartet. Während der Initialisierungsphase wird der Ausgang "InOperation" auf FALSE gesetzt. Die Initialisierungsphase benötigt einige SPS-Zyklen. Nach Durchlaufen der Initialisierungsphase wird der Ausgang "InOperation" gesetzt, um anzuzeigen, dass der Funktionsbaustein nun seine zyklische Aufgabe durchführt.



Während der Initialisierungsphase werden die Motion- und die Sercos-Zykluszeiten einmalig ausgelesen und in der weiteren Bausteinbearbeitung nicht mehr auf Änderung geprüft. Die Sercos-Zykluszeit ist bei gegebenem Oversampling-Faktor maßgeblich für die Schaltgenauigkeit einer Nocke.

Falls Sie die Sercos-Zykluszeit im Parameter C-0-0503 und/oder die Motion-Zykluszeit im C-0-0400 (**MLC/XLC**), bzw den S-0-0001 (**MLD**) ändern, müssen Sie mit einer steigenden Flanke am Bausteineingang "Enable" eine neue Baustein-Initialisierung starten.

Mit dem "EnableMask"-Eingang können die Ausgänge freigegeben oder gesperrt werden.

Das niederwertigste Bit ist der niedrigsten "TrackNumber" zugeordnet. Wird der Wert auf 1, bzw. TRUE gesetzt, dann ist die entsprechende "TrackNumber" freigegeben. Ist das Bit nicht gesetzt, wird der entsprechende Ausgang auf 0 gesetzt, ungeachtet dessen, ob die für den aktuellen Sercos-Takt mit Oversampling interpolierten Positionen innerhalb des Bereiches einer Nocke auf der Nockenscheibe liegen.

Wenn "EnableMask" 0 ist, wird "InOperation" inaktiv, da keiner der Ausgänge freigegeben ist.



Zur Erreichung möglichst niedriger Rechenzeiten sollten Sie stets alle nicht benötigten Nockenscheiben mit Hilfe des Eingangs "EnableMask" deaktivieren!

Jede Nocke hat eine Start- und eine Endposition. Dadurch können Sie für jede Nocke eine "FirstOnPosition" und eine "LastOnPosition" (oder Zeit) festlegen. Zusätzlich zu einer mechanischen Nocke besteht die Möglichkeit, die Nocke für eine bestimmte Zeit zu aktivieren und für jede Nockenscheibe eine Vorhaltezeit zu definieren.

Wenn `FirstOnPosition > LastOnPosition` ist, verhält sich die Nocke invers und ist nur ausgeschaltet innerhalb dieser Positionen.

Bereich/Modulo-Berechnungen

Das Nockenschaltwerk kann sowohl für rotatorische als auch für translatorische Achswichtungen eingesetzt werden. Die Lagedaten können sowohl im Moduloformat als auch im Absolutformat (Achse läuft vorwärts und rückwärts) gerechnet werden.

In beiden Fällen basieren die Nockenschaltwerk-Einheiten auf den für die Achse definierten Einheiten.

AxisDirection gibt an, in welcher Bewegungsrichtung die Nocke aktiv wird. Ist "AxisDirection" 0 (default), wird die Nocke in beiden Bewegungsrichtungen aktiv.

Ist "AxisDirection" 1, wird der Ausgang nur gesetzt, wenn in positiver Richtung verfahren wird.

Ist "AxisDirection" auf 2 gesetzt, wird der Ausgang nur gesetzt, wenn in negative Richtung verfahren wird.

CamSwitchMode wechselt zwischen Positions- oder Zeitangabe.

"CamSwitchMode" = 0 bedeutet Positionsmodus, dabei schaltet der Ausgang an der "FirstOnPosition" ein und an der "LastOnPosition" aus.

Ist "CamSwitchMode" = 1, wird der Ausgang an der "FirstOnPosition" eingeschaltet, und nach der angegebenen "Duration" (Dauer) wieder ausgeschaltet.

Duration: Zeitdauer, die eine zeitgesteuerte Nocke eingeschaltet ist. Nach Ablauf der Zeitdauer wird der Ausgang ausgeschaltet, auch wenn die Position nach Ablauf dieser Zeit noch innerhalb des Nockenbereichs ist. Der Ausgang wird aufgrund dieser zeitgesteuerten Nocke erst wieder eingeschaltet, wenn die Achsposition den Nockenbereich verlassen hat und erneut in den Nockenbereich positioniert wird.



Die Einschaltdauer der Nocke (Duration) kann in Nanosekunden angegeben werden.

Die Auflösung für die Schaltgenauigkeit (Ausschalten der Nocke zum definierten Zeitpunkt) ergibt sich jedoch aus den Einstellungen von Sercos-Zykluszeit (MLC/XLC: C-0-0503, MLD: S-0-0001) und dem eingestellten Oversamplingfaktor.

Ist die eingestellte "Duration" kleiner als die Dauer eines Samples, so wird der Ausgang mindestens für ein Sample eingeschaltet.

Beispiel:

Bei einer Sercos-Zykluszeit von 2ms und einem Oversamplingfaktor von 16 ergibt sich eine Zeitauflösung von $2\text{ms}/16 = 125\mu\text{s}$.

Bei einer Duration von 50ns wird der Ausgang bei diesen Einstellungen dennoch für 125 μs eingeschaltet.

Die Vorhaltezeit (**OnCompensation** und **OffCompensation**) kann positiv oder negativ sein. Negativ bedeutet, dass der Ausgang schaltet bevor die Schaltposition erreicht wird. Die Vorhaltezeit wird in Nanosekunden angegeben (siehe auch "TimeDef"). Die Vorhaltezeiten können für jede Nockenscheibe separat eingestellt werden.

Hysteresis: Dieser Parameter verhindert, dass der Ausgang zu oft schaltet, wenn die Achse nahe dem Schaltpunkt ist und die aktuelle Position um den Schaltpunkt schwankt. Der am Eingang „Hysteresis“ eingestellte Wert ist bei allen Nocken auf allen Nockenscheiben wirksam. Wird ein negativer Wert eingegeben, so wird dieser automatisch als positiver Wert übernommen.



Die Hysterese wirkt bei diesem Baustein anders als bei dem Baustein MC_DigitalCamSwitch!

Die Hysterese wirkt nur bei einer Richtungsänderung. Sowohl in Vorwärts- als auch Rückwärtsrichtung muss nach einer Richtungsänderung zunächst die Hysterese durchlaufen werden, bevor wieder die Masterposition übernommen wird. Dadurch kann eine Nocke zu anderen Schaltpunkten schalten als diese parametrisiert ist.

Im folgenden Bild sind zwei Nocken jeweils von 100° bis 150° definiert.

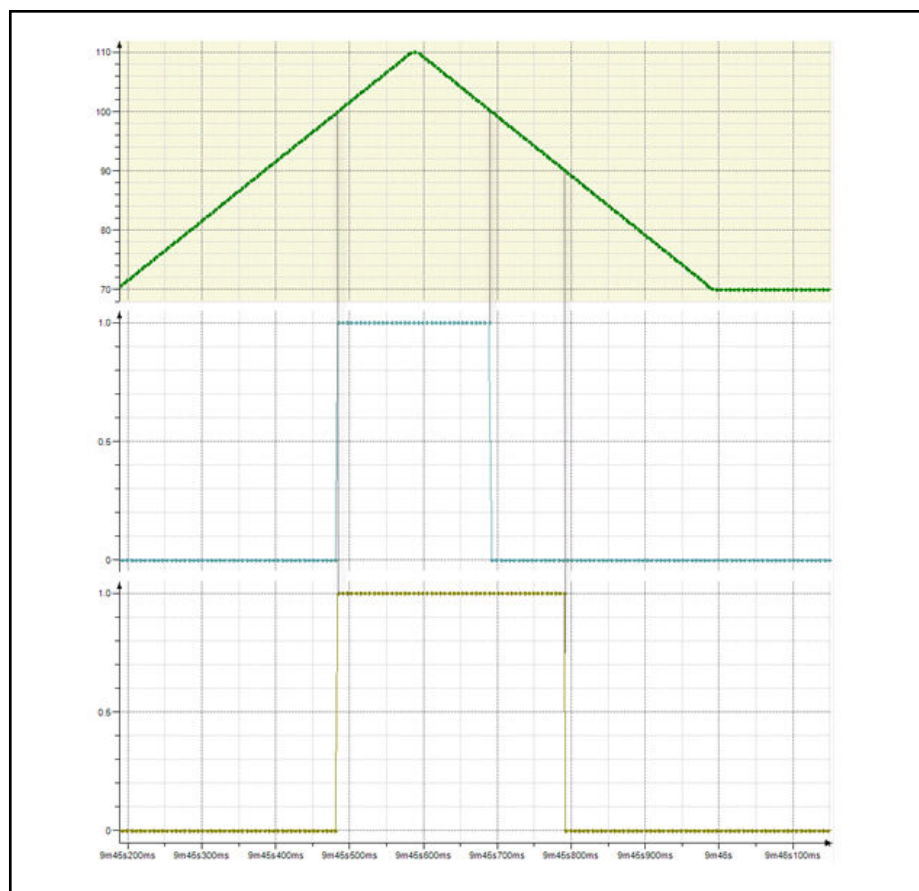


Abb. 3-24: Auswirkungen der Hysterese auf die Schaltpunkte einer Nocke

Die mittlere Kurve zeigt den Ausgang der Nocke ohne Hysterese, die untere Kurve zeigt den Ausgang der Nocke mit einer Hysterese von 20°. Die obere Kurve stellt die Masterposition dar.

Der Master fährt zunächst in positiver Richtung über den Beginn der Nocken, anschließend in negativer Richtung. Dabei ist zu erkennen, dass die Nocke mit und ohne Hysterese jeweils gleichzeitig bei 100° einschalten. Bei 110° wechselt der Master seine Richtung und fährt nun in negativer Richtung. Die Nocke ohne Hysterese schaltet dabei wieder bei 100° aus. Die Nocke mit Hysterese dagegen erst bei 90°. Das kommt dadurch zu Stande, da bei der Nocke mit Hysterese nach einer Richtungsumkehr erst die Hysterese durchfahren werden muss, bevor die Masterposition wieder übernommen wird. Im Beispiel hier erfolgt die Richtungsumkehr bei 110°, die Hysterese ist 20°, also wird die Masterposition erst wieder bei $110^\circ - 20^\circ = 90^\circ$ übernommen. Bei 90° ist die Nocke dann ausgeschaltet, weshalb dann auch der Ausgang gelöscht wird.

TimeDef: Mit Hilfe dieses Parameters ist es möglich, zwischen positiven und negativen Zeiten für die "On/OffCompensation" zu unterscheiden.

Wenn Bit 0 gesetzt ist, dann ist die "OnCompensation"-Zeit negativ, ist Bit 1 gesetzt, dann ist die "OffCompensation"-Zeit negativ.

Beispiel:

"OffCompensation" ist auf 300ms gesetzt, "TimeDef" auf 2.

Das bedeutet, dass die Vorhaltezeit "-300ms" ist.

Das folgende Bild zeigt das mechanische Äquivalent des digitalen Nockenschaltwerks.

Beachten Sie bitte, dass nicht alle Funktionen dargestellt wurden, da sie entweder nicht möglich, bzw. schwer darstellbar sind.

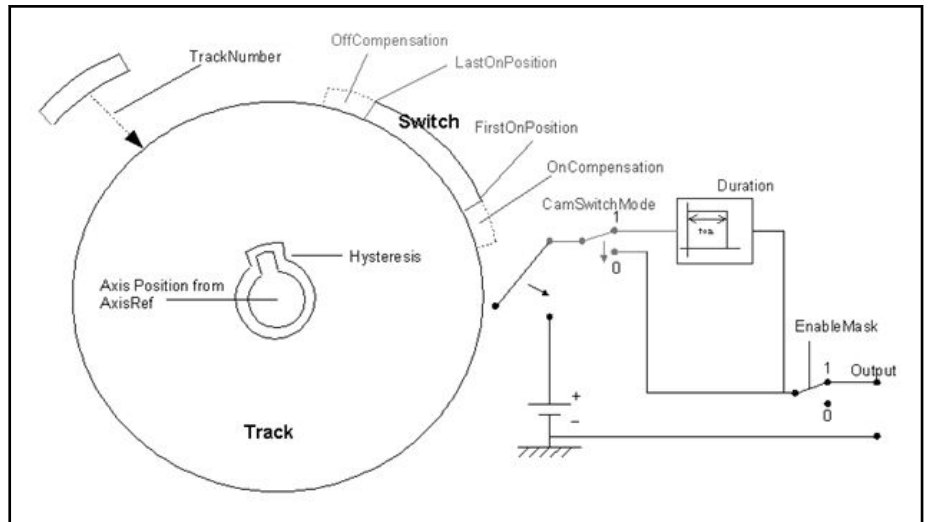


Abb. 3-25: Mechanisches Äquivalent des digitalen Nockenschaltwerks

Beispielprogramm

Ein kurzes Beispiel soll das Konzept erklären, wie der Funktionsbaustein programmiert wird. Die verschiedenen Strukturelemente werden später im Detail erklärt.

Angenommen, eine einzelne Nocke soll angelegt werden:

- mit Nummer 4 auf Nockenscheibe 3
- mit einer "FirstOnPosition" von 35 und
- einer "LastOnPosition" von 36
- die Nocke soll 125µs früher auslösen, und die Nockenscheibe sollte eine Hysterese von 3 haben
- die Nocke soll positionsbezogen, nicht zeitbezogen, ein/ausschalten und nur in positiver Richtung wirksam sein
- Die Achse soll sich mit einer Geschwindigkeit von 1000 U/min bewegen, der Oversamplingfaktor ist so auszulegen, dass die Nocke sicher erkannt wird.

Die Variablendeklaration und der Funktionsbaustein-Aufruf sehen dann folgendermaßen aus:

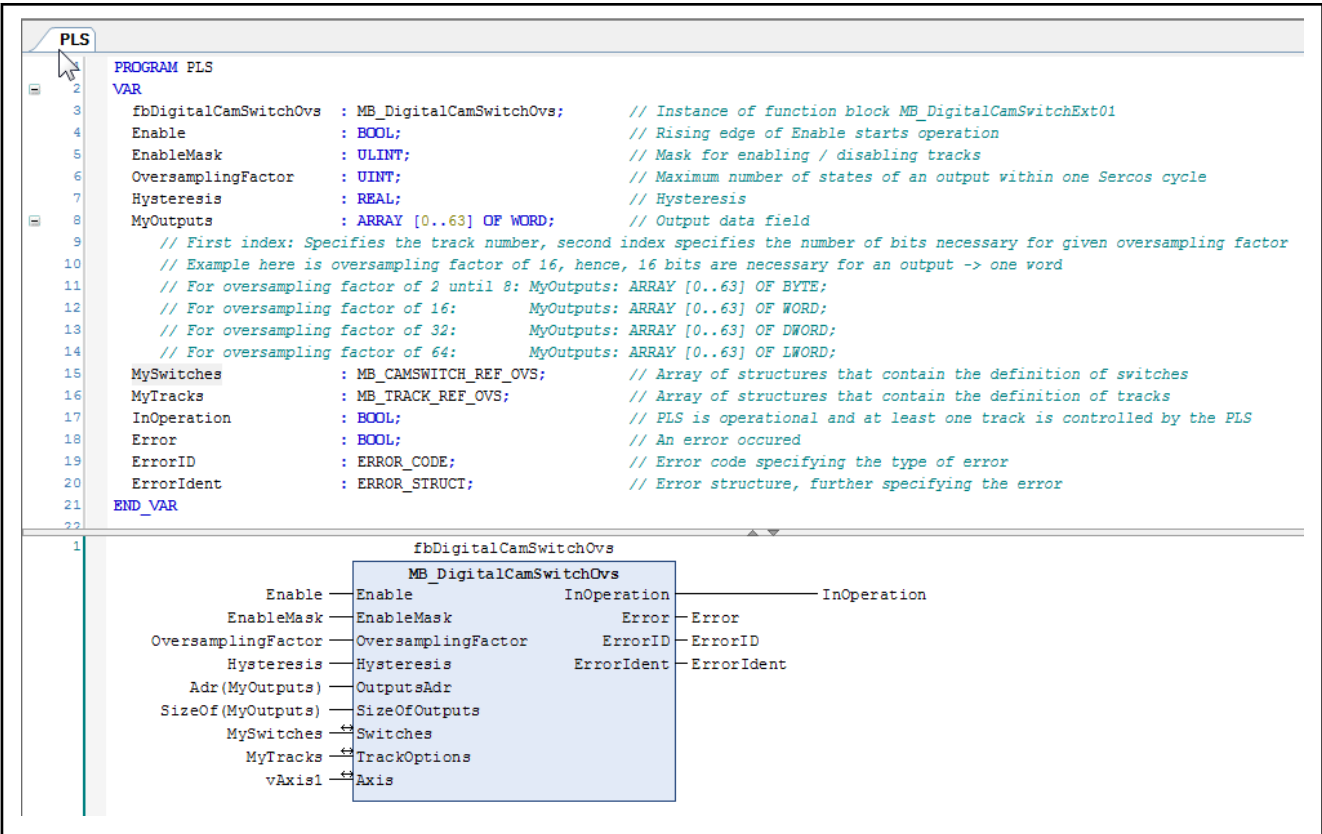


Abb. 3-26: Beispielprogramm: Deklaration der Variablen und Funktionsbaustein-Aufruf

Beachten Sie die beiden Eingangs-Strukturen "MySwitches" und "MyTracks", die alle Eigenschaften der Nocken, Nockenscheiben und Ausgänge definieren. Um diese Strukturen besser verstehen zu können, ist eine nähere Betrachtung der Daten-Inhalte der Strukturen, welche die gewünschten Ergebnisse des obigen Beispiels liefern, notwendig:

XM12.Application.PLS					
Ausdruck	Datentyp	Wert	Vorbereiteter Wert	Adresse	Kommentar
* FbDigitalCamSwitchOvs	MB_DigitalCamSwitchOvs				Instance of function block MB_DigitalCamSwitchExt01
* Enable	BOOL	FALSE			Rising edge of Enable starts operation
* EnableMask	ULINT	0			Mask for enabling / disabling tracks
* OversamplingFactor	UINT	0			Maximum number of states of an output within one Sercos cycle
* Hysteresis	REAL	0			Hysteresis
* MyOutputs	ARRAY [0..63] OF WORD				
* MySwitches	MB_CAMSWITCH_REF_OVS				Array of structures that contain the definition of switches
* MyTracks	MB_TRACK_REF_OVS				Array of structures that contain the definition of tracks
* rTrack	ARRAY [0..63] OF ML_BASIC_TRACK_REF_OVS				
* rTrack[0]	ML_BASIC_TRACK_REF_OVS				
* rTrack[1]	ML_BASIC_TRACK_REF_OVS				
* rTrack[2]	ML_BASIC_TRACK_REF_OVS				
* OnCompensation	LTIME	LTIME#125us			On compenstaion time. Positive values are defined as delay. See also TimeDefinit
* OffCompensation	LTIME	LTIME#0ns			Off compenstaion time. Positive values are defined as delay. See also TimeDefini
* TimeDefinitions	WORD	1			Defines positive or negative sign for the On-/OffCompensation. If bit 0 is set,
* rTrack[3]	ML_BASIC_TRACK_REF_OVS				
* rTrack[4]	ML_BASIC_TRACK_REF_OVS				
* rTrack[5]	ML_BASIC_TRACK_REF_OVS				

Abb. 3-27: Definition der Nockenscheibe

Zuerst wird die Nockenscheibe definiert (Bild oben).

- Da Nockenscheibe 3 benutzt werden soll, muss **rTrack[2]** initialisiert werden, da das Feld mit Index 0 beginnt.

- Die OnCompensation-Zeit wird auf die gewünschten 125µs gesetzt. Damit die Nocke früher und nicht später schaltet, muss Bit 0 in "TimeDefinitions" gesetzt werden, welches die "OnCompensation"-Zeit als negativ definiert.

XM2.Application.PLS					
Ausdruck	Datentyp	Wert	Vorbereiteter Wert	Adresse	Kommentar
fbDigitalCamSwitchOvs	MB_DigitalCamSwitchOvs				Instance of function block MB_DigitalCamSwitchExt01
Enable	BOOL	FALSE			Rising edge of Enable starts operation
EnableMask	ULINT	0			Mask for enabling / disabling tracks
OversamplingFactor	UINT	0			Maximum number of states of an output within one Sercos cycle
Hysteresis	REAL	0			Hysteresis
MyOutputs	ARRAY [0..63] OF WORD				
MySwitches	MB_CAMSWITCH_REF_OVS				Array of structures that contain the definition of switches
rSwitch	ARRAY [0..63] OF ML_BASIC_CAMSWITCH_R...				
rSwitch[0]	ML_BASIC_CAMSWITCH_REF_OVS				
rSwitch[1]	ML_BASIC_CAMSWITCH_REF_OVS				
rSwitch[2]	ML_BASIC_CAMSWITCH_REF_OVS				
rSwitch[3]	ML_BASIC_CAMSWITCH_REF_OVS				
FirstOnPosition	REAL	35			This is the position where the output will come on.
LastOnPosition	REAL	36			This is the position where the output will come off. NOTE: When FirstOnPosition
Duration	LTIME	LTIME#0ns			This is the time given in ns that the output will remain on after it is turned o
TrackNumber	INT	2			The track number which the switch is assigned to. Multiple switches can be assign
AxisDirection	BYTE	1			Writing a 0 to this variable will turn the output for this switch ON in both dir
CamSwitchMode	BYTE	0			If set to 0, the output will be determined by the FirstOnPosition and LastOnPosi
rSwitch[4]	ML_BASIC_CAMSWITCH_REF_OVS				

Abb. 3-28: Definition einer Nocke

Nun wird die Nocke definiert (Bild oben).

- Die Initialisierung der Nocken-Struktur muss mit Index 3 erfolgen, um Nocke 4 zu initialisieren, da das Feld mit Index 0 beginnt.
- "FirstOnPosition" wird auf den gewünschten Wert 35 gesetzt, "LastOnPosition" auf 36 und die "AxisDirection" wird auf 1 gesetzt, was bedeutet, dass die Nocke nur in positiver Richtung wirksam ist.
- Um die Nocke einer Nockenscheibe zuzuordnen, wird "TrackNumber" auf 2 gesetzt; damit wird festgelegt, dass diese Nocke der Nockenscheibe mit Index 2 in der Nockenscheiben-Struktur zugeordnet ist.
- Zum Schluss muss noch der "CamSwitchMode" auf 0 gesetzt werden, um den positionsbezogenen Mode einzuschalten.



Der Zugriff auf die einzelnen Elemente der REF-Strukturen erfolgt nach nachstehendem Muster:

- MyTracks.rTrack[2].TimeDefinitions
- MySwitches.rSwitches[3].AxisDirection

Oversampling

- Der Baustein wird in einer Sercos-synchronen Task aufgerufen: Die Sercos-Zykluszeit soll 2ms betragen.
- Die Achse soll sich mit einer Geschwindigkeit von 1000 U/min bewegen.

Das Abtast-Theorem gibt an, dass die Abtast-Rate doppelt so hoch ist wie die höchste Signalfrequenz, die man braucht, um das Signal erkennen zu können. Dies gilt auch für das Nockenschaltwerk. Das Nockenschaltwerk tastet die Positionsdaten jedes Mal ab, wenn es von der SPS aufgerufen wird und interpoliert in Abhängigkeit vom Oversampling-Faktor für den aktuellen Sercos-Zyklus eine entsprechende Anzahl von Positionswerten. Dadurch wird die Abtastrate um den Oversampling-Faktor erhöht.

Jede Nocke, die kürzer eingeschaltet ist als die aus Sercos-Zykluszeit und Oversampling-Faktor resultierende Abtast-Rate, wird nicht richtig erkannt. Die folgenden Gleichungen zeigen, wie man die Geschwindigkeit oder die

ML_TechMotionBase.library // MX_TechMotionBase.library

Größe der Nocke berechnet, bei denen Schaltvorgänge verloren gehen können.

Rotatorische Systeme:

$$\omega = 2\pi n \quad (\text{Gleichung 1})$$

n ist die Umdrehung pro Sekunde [1/s].

$$\omega = \rho / t \quad (\text{Gleichung 2})$$

ρ ist der Winkel in rad und t die Zeit in Sekunden [s].

Konvertierungsfaktor von Grad to rad: $\pi / 180$

Nimmt man Gleichungen A1 und A2 erhält man

$$\rho[\text{rad}] = 2\pi n t$$

für den Winkel, oder

$$n \left[\frac{\text{Umdrehungen}}{\text{Sekunde}} \right] = \frac{\rho}{2\pi t}$$

Passt man die Faktoren für gemeinsame Einheiten an, so erhält man:

$$\rho[^\circ] = n[1/\text{min}] * t[\text{ms}] * 0,006 \quad (\text{Gleichung 3})$$

oder für die Geschwindigkeit:

$$n[1/\text{min}] = \frac{\rho[^\circ]}{t[\text{ms}]} * 166,6 \quad (\text{Gleichung 4})$$

Die Gleichungen 3 und 4 liefern dem Anwender die beiden notwendigen Gleichungen, um festzustellen, ob Schaltvorgänge verloren gehen können.

Zum Beispiel, wenn die maximale Drehzahl von 1000 min^{-1} bekannt ist, und das PLS in einer zyklischen Task mit einer Zykluszeit von 2 ms aufgerufen wird. Was ist dann die minimale Größe einer Nocke, um sicherzustellen, dass sie von dem PLS erkannt wird?

Nimmt man Gleichung 3 ergibt das:

$$\rho[^\circ] = 1000 * 2 * 0,006 = 12$$

D.h., eine Nocke muss mindestens 12° groß sein, um bei einer Drehzahl von max. 1000 min^{-1} noch sicher erkannt zu werden.

Durch Verwendung von Oversampling erhöht sich die Abtastrate um den Oversamplingfaktor.

Da die Nocke im Beispiel eine Größe von 1° hat und sich die Achse mit einer Geschwindigkeit von 1000 U/min bewegt, muss der Oversamplingfaktor mindestens 12 betragen, damit die Nocke noch sicher erkannt wird.

Da für den Oversamplingfaktor nur 2er Potenzen zulässig sind, muss der Oversamplingfaktor mindestens auf 16 eingestellt werden, damit die Nocke noch sicher erkannt wird.

Deklaration des Ausgangsdatenfelds

```
MyOutputs: ARRAY [0..63] OF WORD;           // Output data field
// First index: Specifies the track number,
// second index specifies the number of bits necessary for given oversampling factor
// Example here is oversampling factor of 16, hence,
// 16 bits are necessary for an output -> one word
// For oversampling factor of 2 until 8: MyOutputs: ARRAY [0..63] OF BYTE;
// For oversampling factor of 16:      MyOutputs: ARRAY [0..63] OF WORD;
// For oversampling factor of 32:      MyOutputs: ARRAY [0..63] OF DWORD;
// For oversampling factor of 64:      MyOutputs: ARRAY [0..63] OF QWORD;
```

Für das Ausgangsdatenfeld wird im Beispiel das Array "MyOutputs" verwendet.

Mit dem Feldindex wird die Nockenscheibe referenziert. Bei 16-fach Oversampling gibt es für jede Nockenscheibe 16 Zustände in einem Sercos-Zyklus. Die 16 Zustände können mit einem WORD (16 Bit pro WORD) abgebildet werden. Das niederwertigste Bit im Word entspricht dem ersten Zustand der Nockenscheibe während des SPS-Zyklus.

Je nach Oversamplingrate kann das Ausgangsdatenfeld somit also eine variable Größe haben. Aufgrund der variablen Größe wird am Bausteineingang "OutputsAdr" die Adresse (Startadresse) des Ausgangsdatenfelds übergeben. Benutzen Sie dazu den Adressoperator "ADR".

Damit im Baustein die Größe des Ausgangsdatenfelds bekannt ist, muss diese am Eingang "SizeOfOutputs" übergeben werden. Benutzen Sie dazu den Operator "SIZEOF", der die Anzahl der Bytes des Ausgangsdatenfelds ermittelt.



In der Schnittstelle des Funktionsbausteins ist das Ausgangsdatenfelds als POINTER TO ARRAY [0..63,0..7] OF BYTE deklariert. Die zweite Dimension ist also ein Array of BYTE. In den Hinweisen zum Eingang "OutputsAdr" (siehe ["Hinweis zu Eingang OutputsAdr" auf Seite 72](#)) wird aber nur ein eindimensionales Feld mit unterschiedlichen Datentypen verwendet. Da aber nur die Adresse übergeben wird, ist dies kein Widerspruch. Allerdings ist bei der späteren Zuweisung des EA-Bereiches notwendig, die Variante mit den unterschiedlichen Datentypen zu verwenden, so wie es in dem nachfolgenden Beispiel auch erfolgt ist.

Der Start des Funktionsbausteines wird vorbereitet:

Damit der Funktionsbaustein gestartet werden kann, müssen die restlichen Variablen zum Funktionsbaustein hinzugefügt und ihre Werte entsprechend gesetzt werden.

1. Der "Enable"-Eingang muss gesetzt werden
2. Im Eingang "EnableMask" muss das entsprechende Bit gesetzt haben: 2#00000100, Bit 2 für dritte Nockenscheibe **rTrack[2]**.
3. Der Eingang "Hysteresis" muss auf den gewünschten Wert 3 gesetzt werden.

ML_TechMotionBase.library // MX_TechMotionBase.library

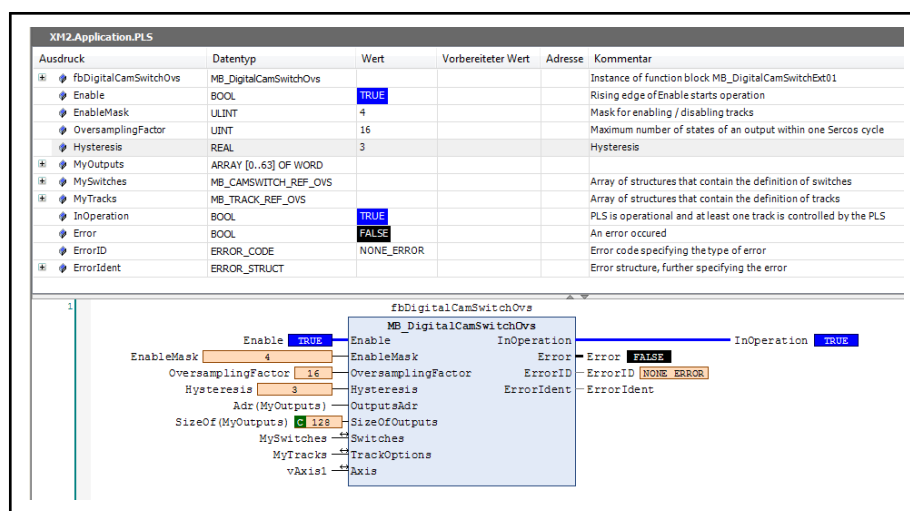


Abb. 3-29: Laufender Funktionsbaustein

Konfiguration des EA-Moduls

Für die Ausgabe der EA-Signale ist ein EA-Modul zu verwenden, das Oversampling unterstützt. In diesem Beispiel wird die Oversampling-Konfiguration mit dem Sercos-Block E/A-Modul R-ILB S3 24 DI16 DIO16/F aufgezeigt. Für die Verwendung von Oversampling muss im Bereich "Erweiterte Einstellungen" der Verbindungstyp "takt synchron" und die Konfiguration mit "IDNs" eingestellt werden (siehe folgendes Bild).

The dialog box shows the configuration for the R_ILB_S3_24_DI16_DIO16_F module. The 'Erweiterte Einstellungen' (Advanced Settings) tab is selected. The 'Verbindungstyp' (Connection Type) is set to 'takt synchron' (clock synchronous) and the 'Konfiguration mit' (Configuration with) is set to 'IDNs'. These two settings are highlighted with a red rectangle.

Abb. 3-30: Dialog Sercos-Slave, Erweiterte Einstellungen

Im Tabulator Oversampling ist die Anzahl der Abtastungen einzustellen.

Die Anzahl der Abtastungen (OversamplingFactor) auf den Ausgangskanälen stellen Sie mit einem Doppelklick auf der Spalte "Anzahl Abtastungen" in der Zeile "OVS OUT" ein.

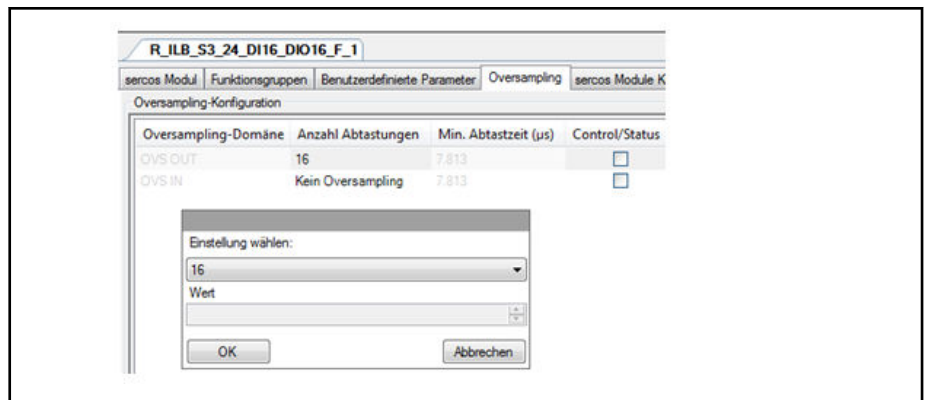


Abb. 3-31: Tabulator Oversampling



Der Funktionsbaustein überprüft nicht, ob der am Bausteineingang „OversamplingFactor“ übergebene Wert mit dem am EA-Modul eingestellten Faktor übereinstimmt.

Wählen Sie im darunter liegenden Bereich die Ausgänge des E/A-Moduls, für die Oversampling aktiviert werden soll. Da im Beispiel nur ein Track aktiviert ist, muss die Oversampling-Funktionalität auch nur bei einem Ausgang aktiviert werden.



Abb. 3-32: Aktivierung Oversampling



Das Sercos-Block-EA-Modul R-ILB S3 24 DI16 DIO16/F unterstützt 8 oversamplingfähige Ausgänge. Falls mehr Nockenscheiben des Funktionsbausteins mit Oversampling ausgegeben werden sollen, so benötigen Sie entsprechend mehr EA-Module. Bei allen EA-Modulen, deren Ausgänge den Tracks eines Funktionsbausteins zugeordnet sind, müssen Sie den gleichen Oversamplingfaktor einstellen wie am Funktionsbaustein.

Zum Schluss muss noch das Ausgangsdatenfeld der dritten Nocke im EA-Abbild gemappt werden: Das folgende Bild zeigt die Abbildung der 16 Bits für die dritte Nocke auf den Ausgang 9 (Fast Digital Output 9) des EA-Moduls.



Nach Aktivierung von Oversampling für mindestens einen Ausgang wird im EA-Abbild der Ordner "OVS OUT" angelegt.

Da im Beispiel Nockenscheibe 3 benutzt wird, müssen Sie im EA-Abbild die Anfangsadresse des Bitstreams für die Nockenscheibe 3 für den oversamplingfähigen Ausgang konfigurieren:

Die Anfangsadresse des Bitstreams für die Nockenscheibe 3 ist „MyOutputs[2].“

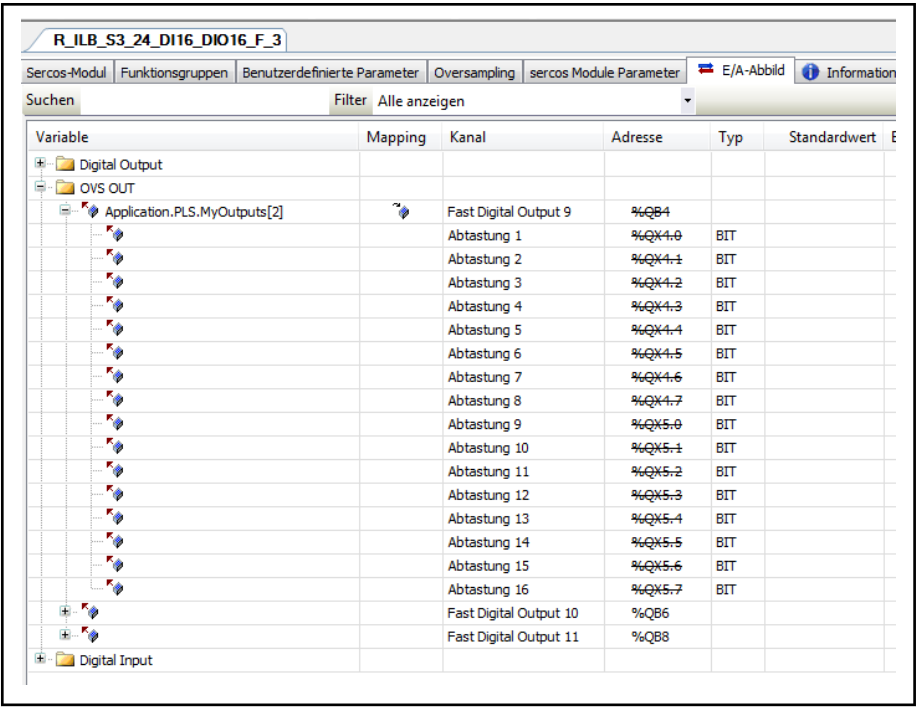


Abb. 3-33: Mapping im EA-Abbild

Zeitdiagramme

Das folgende Beispiel soll die Benutzung der MB_CAMSWITCH_REF_OVS-Struktur und der zeitbasierenden Nocken zeigen. In der folgenden Liste finden Sie eine vorgegebene Konfiguration von Werten:

Parameter	Typ	Switch01	Switch02	Switch03	Switch04	...	SwitchN
FirstOnPosition [u]	REAL	20	25	40	30		
LastOnPosition [u]	REAL	30	30	10	-		
Duration	LTIME	-	-	-	LTIME#250µs		
TrackNumber	INT	1	1	1	2		
AxisDirection	BYTE	1 = Pos	2 = Neg	0 = Beide	0 = Beide		
CamSwitchMode	BYTE	0 = Position	0 = Position	0 = Position	1 = Zeit		

Tab. 3-39: Beispiel: MB_CAMSWITCH_REF_OVS-Struktur



Werte werden in Einheiten [u] angegeben. Diese beziehen sich auf die zugehörige Achse. Ist z. B. eine Einschaltposition bei 20u und die zugehörige Achse wird in Grad gewichtet, ist die Einschaltposition entsprechend 20°.

Dieses Beispiel benutzt die Werte aus dem obigen Beispiel "MB_CAMSWITCH_REF_OVS". Es benutzt weder "On/OffCompensation" noch "Hysteresis".

Dies ist das Verhalten der Ausgänge, wenn die Achse sich kontinuierlich in positive Richtung bewegt. Die Achse ist eine Modulo-Achse mit einer Modulo-Länge von 50u (Einheiten).

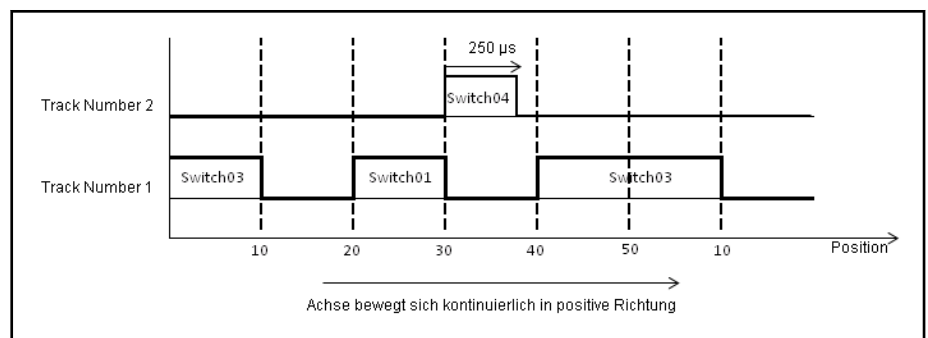


Abb. 3-34: Nocke "Switch01" beim Bewegen in positive Richtung

Dieses Beispiel benutzt zusätzlich

OnCompensation= $-125\mu\text{s}$ und OffCompensation= $+250\mu\text{s}$.

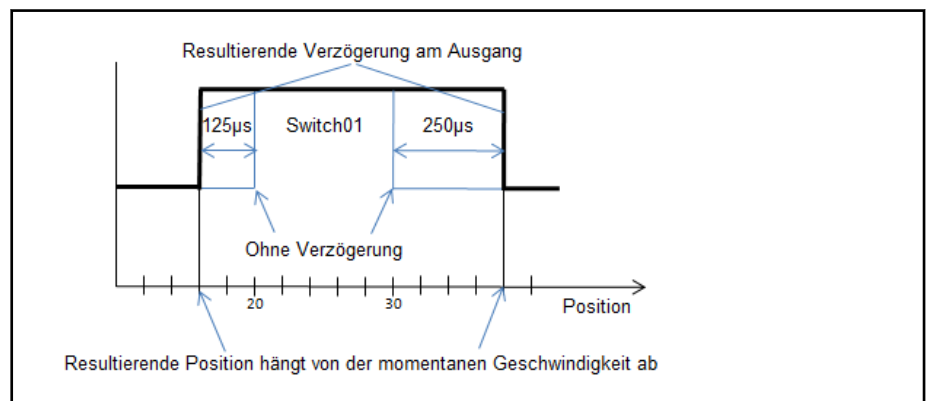


Abb. 3-35: Nocke "Switch01" mit On-/ OffCompensation

Dies ist das Verhalten der Ausgänge, wenn sich die Achse ohne "On-" oder "OffCompensation" und ohne "Hysteresis" kontinuierlich in negative Richtung bewegt:

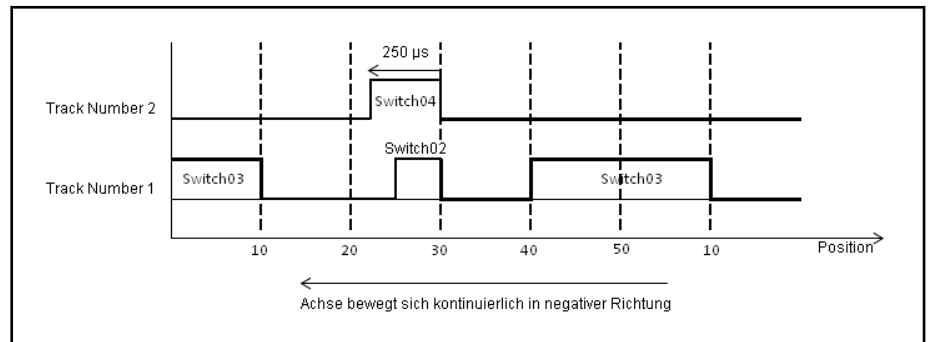


Abb. 3-36: Nocke beim Bewegen in negative Richtung

Realisierungsdetails

Die PLCOpen-Erweiterung spezifiziert das Interface der Nockenschaltwerk-Funktionalität. Jedoch sind einige Anwendungsdetails nicht spezifiziert. Deshalb wurden diese für eine Durchführung mit besserem Laufzeitverhalten definiert.

OnCompensation/OffCompensation

PLCOpen definiert die "On-/OffCompensation" nur für Bewegungen in positive Richtung. In diesem Fall bezieht sich die "OnCompensation" auf die "First-OnPosition", während die "OffCompensation" sich auf die "LastOnPosition" bezieht. Wie ist dies zu definieren für Bewegungen in negative Richtung?

Basierend auf Applikationen bei denen die Vorhaltezeiten benutzt werden, um bei Aktuatoren die An- und Ausschaltverzögerungen zu kompensieren

bedeutet dies, dass die "OnCompensation" für die steigende Flanke des Signals zuständig ist.

Mit anderen Worten: Fährt man in positive Richtung, dann bezieht sich die "OnCompensation" auf die "FirstOnPosition", fährt man in negative Richtung, dann bezieht sich die "OnCompensation" auf die "LastOnPosition". Die "OffCompensation" ist entsprechend definiert

Die folgende Abbildung zeigt das Verhalten für

"OnCompensation= -0.1s und OffCompensation= +0.2s.

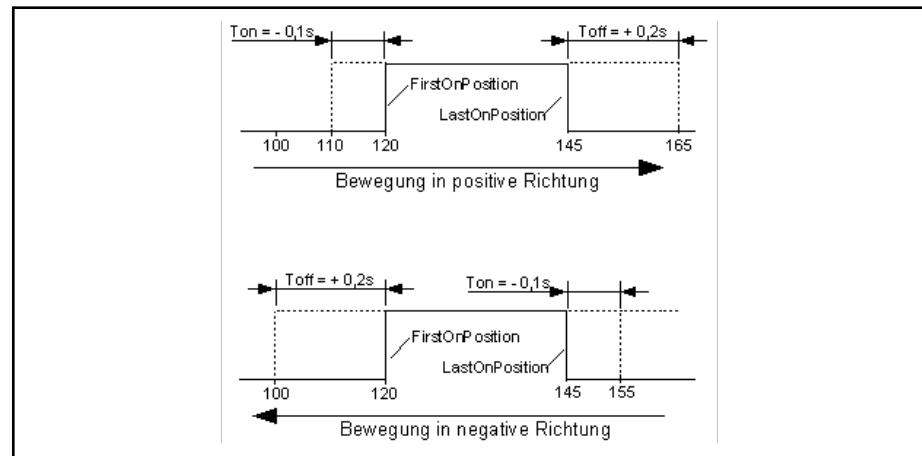


Abb. 3-37: On-/ OffCompensation bei Bewegung in positiver und negativer Richtung

Duration

Die "Duration" (Dauer) ist definiert als die Zeit, die die Nocke eingeschaltet bleibt, nachdem sie eingeschaltet wurde.

Die Frage stellt sich was passiert, wenn die Zeitdauer abgelaufen ist, aber die momentane Position noch innerhalb der "FirstOnPosition" und "LastOn-Position" ist. Soll der Ausgang eingeschaltet bleiben, oder ausgeschaltet werden und erst wieder eingeschaltet werden, nachdem sich die aktuelle Position aus der "FirstOnPosition" und "LastOnPosition" heraus und anschließend wieder hinein bewegt hat?

Es wurde entschieden auf eine steigende Flanke des Nockenausgangs zu warten. Mit anderen Worten: Sobald die Zeitdauer abgelaufen ist, wird der Ausgang ausgeschaltet. Erst wenn sich die aktuelle Position aus den Nockengrenzen heraus und später wieder hinein bewegt hat, wird der Ausgang wieder eingeschaltet und die "Duration" neu gestartet.

Die gleiche Überlegung gilt für die "OnCompensation"/ "OffCompensation", wobei die "Duration" mit der steigenden Flanke einer Nocke beginnt, d.h. an der "LastOnPosition" beim Bewegen in negativer Richtung.

Der Sachverhalt wird in der folgenden Abbildung erläutert:

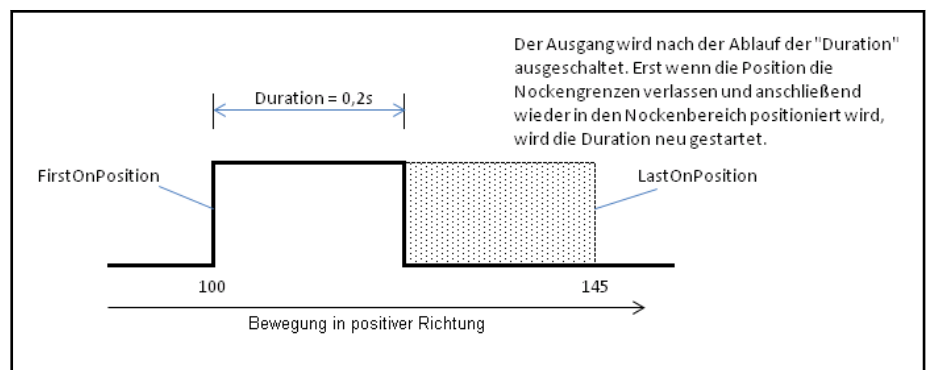


Abb. 3-38: Duration

Kombination von OnCompensation/OffCompensation und Duration

Sollen bei der Benutzung einer zeitbasierenden Nocke mit "On-/OffCompensation"-Zeiten diese dann berücksichtigt werden oder nicht?

Die "OnCompensation" wird berücksichtigt. Daraus resultiert eine frühere/spätere Aktivierung des Ausgangs.

Die "OffCompensation"-Zeit wird bei zeitbasierenden Nocken nicht berücksichtigt. Der Ausgang wird nach Ablauf der „Duration“ ausgeschaltet.

Wie eine Nocke nicht verwendet wird

Das Nockenschaltwerk hat bis zu 64 Nocken. Um festzustellen, ob ein Nocken benutzt wird oder nicht gibt es das Element "TrackNumber" in der MB_CAMSWITCH_REF_OVS-Struktur. Diese Variable beinhaltet die Nockenscheiben-Nummer. Gültige Werte dafür sind 0 bis 63, da es maximal 64 Nockenscheiben gibt.

Die Default-Einstellung dieser Variable ist 64, sie wird als nicht benutzt markiert. Jede ungültige Nummer (64 bis 65535) wird interpretiert als eine nicht benutzte Nocke und wird deshalb auch nicht von der Firmware ausgewertet.

Einschränkungen des Nockenschaltwerkes

Die Einschränkungen des Nockenschaltwerkes werden nachstehend diskutiert. Die Einschränkungen sind lediglich Definitionen, die jedoch zu keinem Fehler führen. Das Systemverhalten wird über diese Definitionen festgelegt. Der Anwender muss deshalb mit unterschiedlichem Verhalten rechnen.

Verlorene Schaltvorgänge

Das Abtast-Theorem gibt an, dass die Abtast-Rate doppelt so hoch ist wie die höchste Signalfrequenz, die man braucht, um das Signal erkennen zu können. Dies gilt auch für das Nockenschaltwerk. Das Nockenschaltwerk tastet die Positionsdaten jedes Mal ab, wenn es von der SPS aufgerufen wird und interpoliert in Abhängigkeit vom Oversampling-Faktor für den aktuellen Sercos-Zyklus eine entsprechende Anzahl von Positionswerten. Dadurch wird die Abtast-Rate um den Oversampling-Faktor erhöht.

Jede Nocke, die kürzer eingeschaltet ist als die aus Sercos-Zykluszeit und Oversampling-Faktor resultierende Abtast-Rate, wird nicht richtig erkannt. Die folgenden Gleichungen zeigen, wie man die Geschwindigkeit oder die Größe der Nocke berechnet, bei denen Schaltvorgänge verloren gehen können.

Rotatorische Systeme:

$$\omega = 2\pi n \quad (\text{Gleichung A1})$$

n ist die Umdrehung pro Sekunde [1/s].

$$\omega = \rho/t \quad (\text{Gleichung A2})$$

ρ ist der Winkel in rad und t die Zeit in Sekunden [s].

Konvertierungsfaktor von Grad to rad: $\pi/180$

Nimmt man Gleichungen A1 und A2 erhält man

$$\rho[\text{rad}] = 2\pi n t$$

für den Winkel, oder

$$n \left[\frac{\text{Umdrehungen}}{\text{Sekunde}} \right] = \frac{\rho}{2\pi t}$$

Passt man die Faktoren für gemeinsame Einheiten an, so erhält man:

$$\rho[^\circ] = n[1/\text{min}] * t[\text{ms}] * 0,006 \quad (\text{Gleichung A3})$$

oder für die Geschwindigkeit:

$$n[1/\text{min}] = \frac{\rho[^\circ]}{t[\text{ms}]} * 166,6 \quad (\text{Gleichung A4})$$

Die Gleichungen A3 und A4 liefern dem Anwender die beiden notwendigen Gleichungen, um festzustellen, ob Schaltvorgänge verloren gehen können.

Zum Beispiel, wenn die maximale Drehzahl von 1000 min⁻¹ bekannt ist, und das Nockenschaltwerk in einer Sercos-synchronen Task mit einer Zykluszeit von 2 ms und einem Oversamplingfaktor von 16 aufgerufen wird. Was ist dann die minimale Größe einer Nocke, um sicherzustellen, dass sie von dem Nockenschaltwerk noch erkannt wird? Nimmt man Gleichung A3, so ergibt das:

$$\rho[^\circ] = 1000 * \frac{2}{16} * 0,006 = 0,75$$

D.h., eine Nocke muss mindestens 0,75° groß sein, um bei einer Drehzahl von max. 1000 min⁻¹, einer Sercos-Zykluszeit von 2ms und einem Oversamplingfaktor von 16 noch sicher erkannt zu werden.

Lineare Systeme:

$$v = s / t \quad (\text{Gleichung B1})$$

s ist die Distanz in Millimeter [mm] und t die Zeit in Sekunden [s].

Nimmt man die Sercos-Zykluszeit dividiert durch den Oversamplingfaktor als Abtastzeit für das Nockenschaltwerk und berücksichtigt die entsprechenden Faktoren, um die Einheiten anzugleichen, so erhält man die folgenden Gleichungen:

$$s[\text{mm}] = v \left[\frac{\text{mm}}{\text{min}} \right] * t[\text{ms}] * 1,66 * 10^{-5} \quad (\text{Gleichung B2})$$

oder für die Geschwindigkeit:

$$v \left[\frac{mm}{min} \right] = \frac{s \left[\frac{mm}{s} \right]}{t \left[\frac{ms}{s} \right]} * 60000 \quad (\text{Gleichung B3})$$

Diese Gleichungen gelten auch für Systeme die Inch [in] als Einheit für den Weg:

$$s \left[\frac{in}{min} \right] = v \left[\frac{in}{min} \right] * t \left[\frac{ms}{s} \right] * 1,66 * 10^{-5} \quad (\text{Gleichung B4})$$

oder für die Geschwindigkeit haben:

$$v \left[\frac{in}{min} \right] = \frac{s \left[\frac{in}{s} \right]}{t \left[\frac{ms}{s} \right]} * 60000 \quad (\text{Gleichung B5})$$

Beispiel: Eine Nocke in einem linearen System ist 0,02 in lang. Die Sercos-Zykluszeit beträgt 2ms bei einem Oversamplingfaktor von 16. Welche maximale Geschwindigkeit ist möglich, damit die Nocke noch sicher erkannt wird?

Nimmt man Gleichung B5, so erhält man das Ergebnis:

$$v \left[\frac{in}{min} \right] = \frac{0,02}{\frac{2}{16}} * 60000 = 9600$$

Nocken haben scheinbar die falsche Größe

Voranstehend wurden die Randbedingungen für "verlorene Schaltvorgänge" beschrieben.

Auch wenn die Bedingungen für verlorene Nocken nicht erfüllt sind, können die Schaltzeitpunkte ungenau sein. Das Nockenschaltwerk kann nur zu den Abtastzeitpunkten den Zustand seiner Ausgänge wechseln. Der Abstand zweier Abtastzeitpunkte ergibt sich aus der eingestellten Sercos-Zykluszeit und dem Oversamplingfaktor. Nehmen wir an, das Nockenschaltwerk wird in einer Sercos-synchronen Task mit 2ms Zykluszeit und mit einem Oversamplingfaktor von 16 aufgerufen.

Wenn z.B. eine Nocke bei einer vorgegebenen Geschwindigkeit für 200 µs aktiv sein soll, dann wird der Ausgang entweder für 250µs (Fall A) oder für 125µs (Fall B) auf "On" gesetzt.

Die Nockeneinschaltdauer und die Größe des sich damit ergebenden Fehlers (Fehler Fall A: 50µs, Fehler Fall B: 75µs) sind abhängig vom Abtastzeitpunkt des Nockenschaltwerkes. Bei Erhöhung des Oversamplingfaktors auf 32 (Fall C) ergibt sich eine Nockeneinschaltdauer von 187,5µs und ein Fehler von 12,5µs.

Eine nähere Erläuterung hierzu sehen Sie in der folgenden Abbildung:

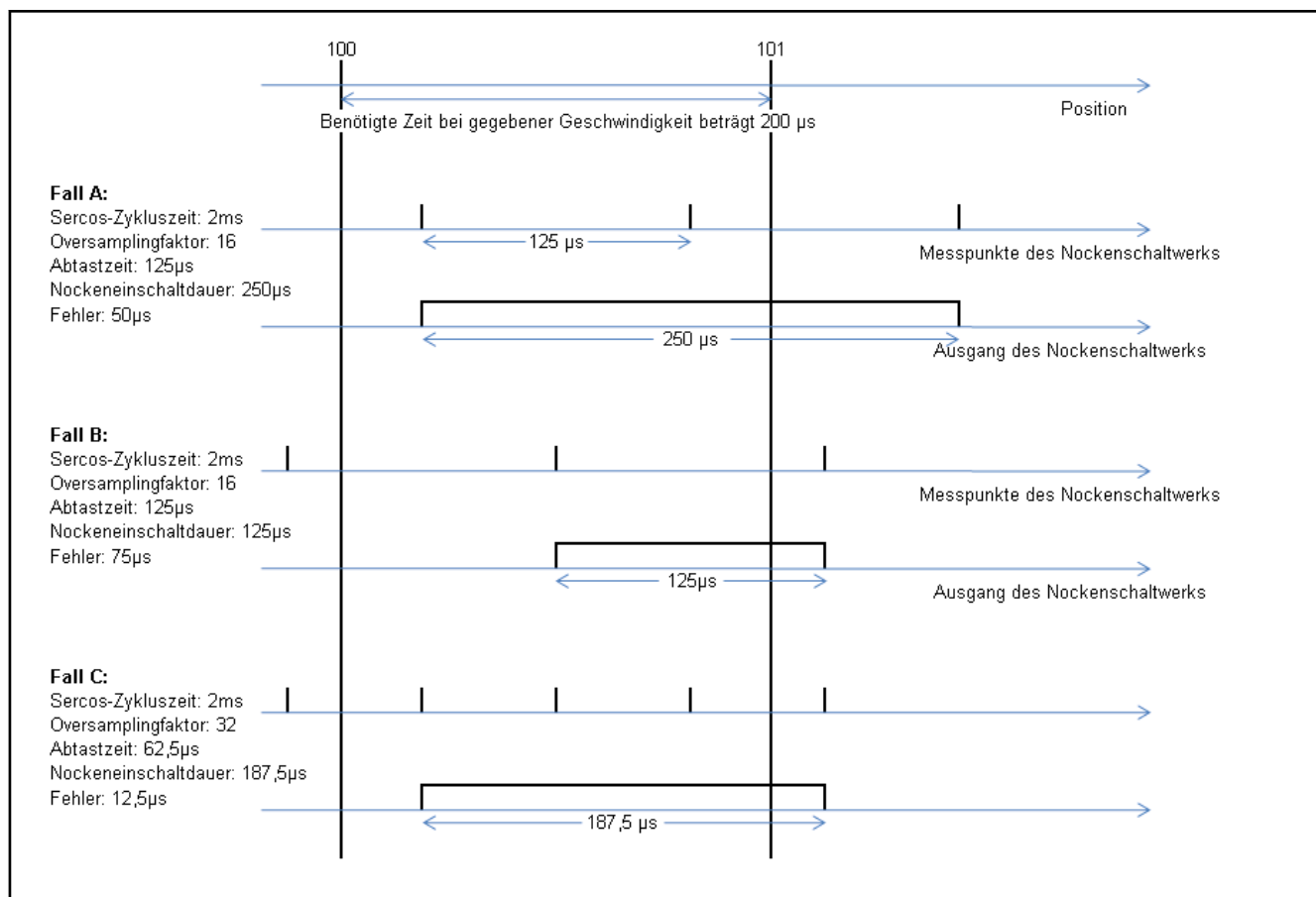


Abb. 3-39: Ungenauigkeit der Nocken

**Schaltvorgang erfolgt zweimal
beim Beschleunigen/Verzögern**

Werden "OnCompensation" und "OffCompensation" für eine Nocke benutzt, ist es möglich, dass sie mehrfach ein- und ausgeschaltet wird, wenn die Nockenschaltwerk Masterposition beschleunigt oder abbremst. Dies passiert aufgrund der Tatsache, dass das Nockenschaltwerk eine konstante Geschwindigkeit voraussetzt, was unter bestimmten Umständen jedoch nicht gegeben ist. Das größere Problem taucht beim Abbremsen auf.

Beispiel:

- Rotatorisches System mit Modulwert 5000°
- Die Masterposition bewegt sich kontinuierlich in positiver Richtung
- FirstOnPosition = 1000
- OnCompensation = 0,2s
- Die Mastergeschwindigkeit wird von 1000 U/min auf 100 U/min abgebremst, siehe Geschwindigkeitsverlauf in folgender Abbildung:

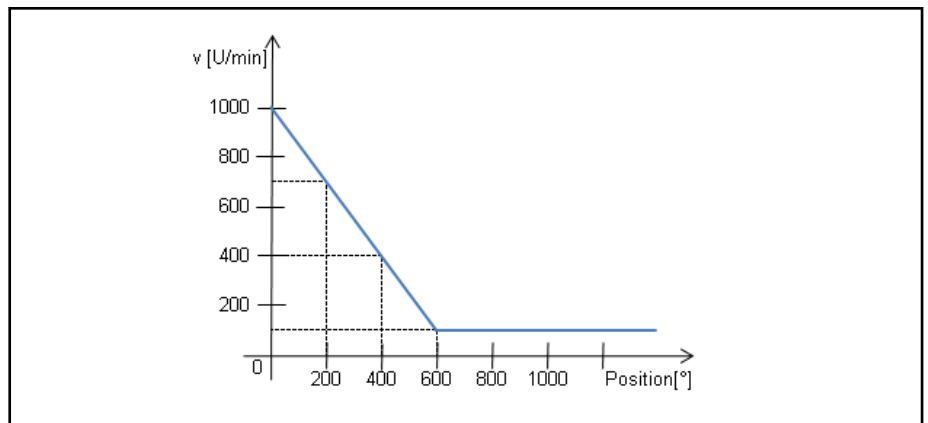


Abb. 3-40: Mastergeschwindigkeit

Die Einschaltposition der Nocke berechnet sich aufgrund der "OnCompensation" und der aktuellen Geschwindigkeit mit folgender Gleichung:

$$\rho[^\circ] = FirstOnPosition[^\circ] - n[1/\text{min}] * t[\text{ms}] * 0,006$$

Wird beispielsweise bei 200° abgetastet, so berechnet sich die Einschaltposition zu:

$$\rho[^\circ] = 1000 - 700 * 200 * 0,006 = 160$$

D.h., der Master hat die Nockeneinschaltposition bereits überschritten und der Ausgang ist eingeschaltet.

Die folgende Tabelle zeigt die berechneten Einschaltpositionen, sowie den zugehörigen Ausgangszustand zu verschiedenen Abtastzeitpunkten:

Abtastposition	Berechnete Einschaltposition	Ausgangszustand
0°	4800°	EIN
200°	160°	EIN
400°	520°	AUS
600°	880°	AUS
800°	880°	AUS
900°	880°	EIN

Obige Tabelle zeigt, dass aufgrund der Verzögerung der Schaltvorgang zweimal erfolgt.

Einige Betrachtungen, wann eine Nocke zu mehreren Zeiten aktiviert werden kann, folgen:

Man kann die folgenden Gleichungen aufstellen:

$$s_v = v_0 * t_{lead} \quad (\text{Gleichung 1})$$

Dabei ist s_v der Weg, der bei der vorgegebenen Geschwindigkeit v_0 in der "OnCompensation"-Zeit t_{lead} zurückgelegt wird.

$$s_b = v_0 * t - \frac{1}{2} a t^2 \quad (\text{Gleichung 2})$$

s_b ist der Weg, der nötig ist zum Abbremsen bei einer vorgegebenen Startgeschwindigkeit v_0 und Verzögerung a in der Zeit t .

Außerdem die allgemeinen Gleichungen:

$$v = a * t \text{ oder } t = v / a$$

Die "OnCompensation"-Zeit eines programmierbaren Nockenschaltwerks "bewegt" die Nocke in Richtung niedriger Positionen entsprechend Gleichung 1.

Der Weg, der benötigt wird, um bei einer vorgegebenen Geschwindigkeit zum Stillstand zu kommen, kann mit Gleichung 2 ermittelt werden.

Die allgemeine Regel lautet deshalb:

Wenn die Position mit $v = v_0$ erreicht ist, damit der Ausgang eines programmierbaren Nockenschaltwerks mit "OnCompensation"-Zeit eingeschaltet werden soll - und die Eingangsposition in einer Weise zum Stoppen abgebremst wird, dass die Stopposition gleichzeitig auch die Einschaltposition des programmierbaren Nockenschaltwerks ohne "OnCompensation"-Zeit ($v = 0$) ist, dann wird der Ausgang dazwischen nicht ausgeschaltet.

Wenn wir Gleichung 1 mit Gleichung 2 gleichsetzen, erhalten wir die Gleichung, mit der die Nocke nur einmal eingeschaltet wird.

Wenn wir etwas schneller abbremsen, dann wird die Nocke mehrfach eingeschaltet.

$$s_b = s_v \Rightarrow v_0 t - \frac{1}{2} a t^2 = v_0 t_{lead} \Rightarrow v_0 \frac{v_0}{a} - \frac{1}{2} a \frac{v_0^2}{a^2} = v_0 t_{lead}$$

$$\Rightarrow \frac{v_0^2}{2a} = v_0 t_{lead}$$

$$\Rightarrow v_0 = 2 t_{lead} a \quad (\text{Gleichung 3})$$

In Gleichung 3 kann man sehen, dass die Startgeschwindigkeit v_0 bei der es möglich ist, dass ein Nockenschalter mehrfach eingeschaltet wird, größer wird durch Erhöhung der "OnCompensation"-Zeit t_{lead} oder Verzögerung a .

Es gibt keine Lösung für dieses Problem. Man kann lediglich die "OnCompensation"-Zeit und Verzögerung für die gewünschte Applikation so klein wie möglich halten.

Eine andere Möglichkeit ist es, die "Hysterese" zu benutzen, damit die Schwankungen aufgrund der Beschleunigung/Verzögerung darin berücksichtigt werden.

Die gleichen Überlegungen können angewandt werden für das Beschleunigen, und/oder negative Geschwindigkeiten.

Laufzeitverhalten

Der Funktionsbaustein MB_DigitalCamSwitchOvs implementiert ein frei programmierbares Nockenschaltwerk mit bis zu 64 Nocken auf 64 Nockenscheiben. Zusätzlich wird Oversampling mit einem maximalen Oversamplingfaktor von 64 unterstützt. In der rechenintensivsten Konfiguration könnten also 64 Nocken verteilt auf 64 Nockenscheiben mit 64-fach Oversampling betrieben werden.



Zur Erreichung möglichst niedriger Rechenzeiten sollten Sie stets alle nicht benötigten Nockenscheiben mit Hilfe des Eingangs "EnableMask" deaktivieren!

Fehlercodes Der Funktionsbaustein MB_DigitalCamSwitchOvs nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170.

Er kann in Additional1/ Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#0006	16#xx	Ungültiger Zustand des Funktionsbausteins. 16#xx in Additional2 beinhaltet die interne Nummer der Zustandsmaschine.
INPUT_INVALID_ERROR	16#000D	16#xx	Ursache: AxisRef liegt außerhalb des gültigen Bereiches. Lösung: AxisRef in den gültigen Bereich bringen. 16#xx in Additional2 beinhaltet die ungültige Achsadresse.
RESSOURCE_ERROR	16#0012	16#xx	Steuerungsnummer ist ungültig. 16#xx in Additional2 beinhaltet die ungültige Steuerungsnummer.
INPUT_INVALID_ERROR	16#2B00	16#0000	Ursache: Ungültige Kombination von Einheiten und Wichtung der Achse Lösung: Achse korrekt parametrieren. Gültige Kombinationen sind: Lineare Wichtung mit Längeneinheiten in [m] oder [inch] und Geschwindigkeitseinheiten in [m]/[min] / [inch]/[min] oder [m]/[s] / [inch]/[s] oder translatorische Wichtung mit Einheit [Grad] und Geschwindigkeitseinheiten in [U]/[min] oder [U]/[s]
RESOURCE_ERROR	16#2B01	Runtime Error Codes (CmpErrors.library)	Ursache: Bei der Ermittlung des Handles der Task, in der der Baustein aufgerufen wird, ist ein Fehler aufgetreten. Lösung: Bausteininitialisierung mit steigender Flanke am Enable-Eingang neu starten.
RESOURCE_ERROR	16#2B02	Runtime Error Codes (CmpErrors.library)	Ursache: Bei der Ermittlung der Eigenschaften der Task, in der der Baustein aufgerufen wird, ist ein Fehler aufgetreten. Lösung: Bausteininitialisierung mit steigender Flanke am Enable-Eingang neu starten.

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#2B03	16#0000	Ursache: Die Task, in der der Baustein aufgerufen wird, ist nicht synchron zum Sercos-Takt. Lösung: Der Baustein muss in einer Sercos-synchronen Task aufgerufen werden. Typ der Task "Extern", externes Ereignis "EVENT_OP_MODE_SERCOS_CYCLIC" (MLC/XLC), bzw. "CCD_SYNCHRONIZED_TASK" (MLD)
RESOURCE_ERROR	16#2B04	Additional1 von MB_ReadParameter	Ursache: Beim Lesen der Sercos-Zykluszeit (Parameter C-0-0503 (MLC/XLC) oder S-0-0001 (MLD)) ist ein Fehler aufgetreten. Additional2 enthält die Fehlernummer des fehlererzeugenden Bausteins MB_ReadParameter (MLC/XLC); MB_ReadParameter (MLD). Lösung: Bausteininitialisierung mit steigender Flanke am Enable-Eingang neu starten.
INPUT_RANGE_ERROR	16#2B05	Maximal zulässiger Oversampling Faktor	Ursache: Der am Bausteineingang OversamplingFactor vorgegebene Wert für die Oversamplingrate ist zu groß. Lösung: Der Maximalwert des Eingangs Oversampling-Factor ist 64.
INPUT_INVALID_ERROR	16#2B06	16#0000	Ursache: Der am Bausteineingang OversamplingFactor vorgegebene Wert für die Oversamplingrate ist ungültig. Lösung: Für den OversamplingFactor werden nur 2er Potenzen (1, 4, 8, 16, ...) unterstützt. Hinweis: Da das Sercos-Block E/A-Modul R-ILB S3 24 DI16 DIO16/F den Oversampling-Faktor 2 nicht unterstützt, wird der Faktor 2 auch vom Funktionsbaustein nicht zugelassen.

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR	16#2B07	Mindestgröße des Ausgangsdatenfelds	Ursache: Die Größe des Ausgangsdatenfeldes (Eingang SizeOfOutputs) ist zu klein. Lösung: Der Oversampling-Faktor bestimmt die Mindestgröße des Ausgabedatenfelds. Die Mindestgröße des Ausgabedatenfelds beträgt 8 Byte (bis Oversampling-Faktor 8). Ab 8 Bytes ergibt sich die Mindestgröße des Ausgabedatenfelds aus folgender Formel: Mindestgröße in Bytes = Anzahl der Nockenscheiben (32) x Oversampling-Faktor / 8 (Anzahl Bits in einem Byte). Beispiel: Bei 32-fachem Oversampling ist die Mindestgröße des Ausgangsdatenfeld $32 \times 32 / 8 = 128$ Bytes.
RESOURCE_ERROR	16#00002B09	16#0000	Ursache: Die Motion Zykluszeit (C-0-0400.0.0) und die Sercos-Zykluszeit (C-0-0503.0.0) haben unterschiedliche Werte. Lösung: Stellen Sie für die Motion Zykluszeit (C-0-0400.0.0) und die Sercos-Zykluszeit (C-0-0503.0.0) die gleichen Werte ein.
RESOURCE_ERROR	16#00002B0A	Additional1 von MB_ReadParameter	Ursache: Beim Lesen der Motion-Zykluszeit (Parameter C-0-0400) ist ein Fehler aufgetreten. Additional2 enthält die Fehlernummer des fehlererzeugenden Bausteins MB_ReadParameter. Lösung: Bausteininitialisierung mit steigender Flanke am Enable-Eingang neu starten.

Tab. 3-40: Fehlercodes FB MB_DigitalCamSwitchOvs

3.4.9 MB_CAMSWITCH_REF_OVS

Typdefinitionen Die folgende Definition beschreibt einen Datentyp, der im Funktionsbaustein [MB_DigitalCamSwitchOvs](#), Seite 71, verwendet wird.

Der Datentyp "MB_CAMSWITCH_REF_OVS" enthält ein Array von 64 Basisstrukturen, die jede eine einzelne Nocke beschreibt ([ML_BASIC_CAMSWITCH_REF_OVS](#), Seite 98).

Typdefinition

```

TYPE MB_CAMSWITCH_REF_OVS :
STRUCT
  // This is an array of basic structures, one for each individual switch.
  // Numbering starts at 0!
  // I.e. the firstswitch is addressed by rSwitch[0].xxxx
  rSwitch : ARRAY [0..63] OF ML_BASIC_CAMSWITCH_REF_OVS;
END_STRUCT
END_TYPE

```

3.4.10 ML_BASIC_CAMSWITCH_REF_OVS

Typdefinitionen Die folgende Definition beschreibt einen Datentyp, der im Funktionsbaustein [MB_DigitalCamSwitchOvs, Seite 71](#), verwendet wird.

Der Datentyp [MB_CAMSWITCH_REF_OVS, Seite 97](#), enthält ein Array von 64 Basis-Strukturen, die jede eine einzelne Nocke beschreibt.

Werte werden in Einheiten [u] angegeben. Diese beziehen sich auf die zugehörige Achse.

Ist z. B. eine Einschaltposition bei 20u und die zugehörige Achse wird in Grad gewichtet, ist die Einschaltposition entsprechend 20°.

Die Basis-Struktur "ML_BASIC_CAMSWITCH_REF_OVS" sieht wie folgt aus:

Strukturelement	Typ	Beschreibung
FirstOnPosition [u]	REAL	Untere Grenze bei der der Ausgang AN ist (Einschaltposition)
LastOnPosition [u]	REAL	Obere Grenze bei der der Ausgang AN ist (Ausschaltposition)
Duration	LTIME	Gekoppelt mit zeitbezogenem "CamSwitchMode". D.h., der Ausgang bleibt für die angegebene Zeit an
TrackNumber	INT	"TrackNumber" ist die Referenz zur Nockenscheibe
AxisDirection	INT	Richtung in der das Nockenschaltwerk wirksam ist: beide (=0; Default); Positiv (1); Negativ (2)
CamSwitchMode	INT	Positionsbezogen (=0; Default); zeitbezogen (=1)

Tab. 3-41: Struktur ML_BASIC_CAMSWITCH_REF_OVS

3.4.11 MB_TRACK_REF_OVS

Typdefinitionen Die folgende Definition beschreibt einen Datentyp, der im Funktionsbaustein [MB_DigitalCamSwitchOvs, Seite 71](#), verwendet wird.

Der Datentyp MB_TRACK_REF_OVS enthält ein Array mit 64 Basis-Strukturen ([ML_BASIC_TRACK_REF_OVS, Seite 98](#)), d. h., 64 Nockenscheiben, die 64 oversamplingfähigen Ausgängen zugeordnet werden können.

Typdefinition

```

TYPE MB_TRACK_REF_OVS :
STRUCT
  // This is an array of basic structures, one for each individual track.
  // Numbering starts at 0!
  // I.e. the firstswitch is addressed by rTrack[0].xxxx
  rTrack: ARRAY [0..63] OF ML_BASIC_TRACK_REF_OVS;
END_STRUCT
END_TYPE

```

Jeder Nockenscheibe können mehrere Nocken zugeordnet sein.

3.4.12 ML_BASIC_TRACK_REF_OVS

Typdefinitionen Die folgende Definition beschreibt einen Datentyp, der im Funktionsbaustein [MB_DigitalCamSwitchOvs, Seite 71](#), verwendet wird.

Eine Nockenscheibe ist ein Element, das die Eigenschaften des Nockenschaltwerkes bezogen auf einen oversamplingfähigen Ausgang beschreibt.

Die Basis-Struktur ML_BASIC_TRACK_REF_OVS enthält Elemente, mit der das Verhalten für eine Nockenscheibe eingestellt werden kann.

Strukturelement	Typ	Beschreibung
OnCompensation	LTIME	Vorhaltezeit der Nockenscheibe bezogen auf den Einschaltzeitpunkt einer Nocke
OffCompensation	LTIME	Vorhaltezeit der Nockenscheibe bezogen auf den Ausschaltzeitpunkt einer Nocke
TimeDef	WORD	Zeitdefinition: Bit 0 gesetzt: "OnCompensation"-Zeit ist negativ; Bit 1 gesetzt: "OffCompensation"-Zeit ist negativ Eine negative Zeit bedeutet, dass der Ausgang früher schaltet. Eine positive Zeit, dass er - gemessen am Schaltzeitpunkt – später schaltet.

Tab. 3-42: Struktur ML_BASIC_TRACK_REF_OVS

3.5 Download von Kurvenscheiben

3.5.1 Einführung und Übersicht

Der nachfolgende Baustein ist für den Download von Kurvenscheiben auf den Antrieb und die Steuerung gedacht, um die vorhandenen Detailunterschiede zu kapseln. Die Struktur [MB_CAMTABLE](#), Seite 102, unterstützt dabei den Funktionsbaustein [MB_LoadCamData](#), Seite 99, beim Laden der Kurvenscheibentabellen.

Die Enumeration [MB_CAMPREF](#), Seite 102, erlaubt es, für die Abarbeitung von Kurventabellen **im Antrieb**, auch das alte Kurventabellen-Format zu verwenden.

3.5.2 MB_LoadCamData

Kurzbeschreibung

Der MB_LoadCamData übernimmt den Download von Kurvenscheiben auf den Antrieb und die Steuerung, um die vorhandenen Detailunterschiede zu kapseln.



Der Funktionsbaustein "MB_LoadCamData (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

MLC/XLC 14VRS	ML_TechMotionBase
Funktionspaket MOT	
MLD (IL2G)	MX_TechMotionBase

Tab. 3-43: Referenztable MB_LoadCamData

Schnittstellenbeschreibung

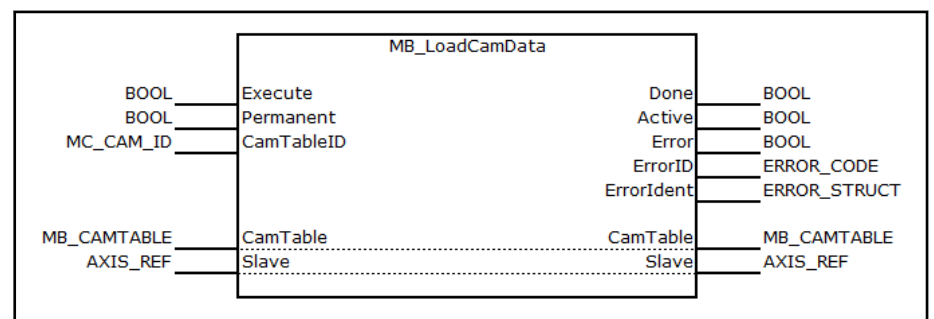


Abb. 3-41: Funktionsbaustein MB_LoadCamData

I/O-Typ	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	Permanent	BOOL	Einstellmöglichkeit, ob Kurvenscheibe permanent in den Parameter geschrieben werden soll, oder nur temporär in den flüchtigen Systemspeicher
	CamTableID	MC_CAM_ID	Auswahl der gewünschten Kurvenscheibentabelle des Antriebs oder der Steuerung
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe Fehlerbehandlung , Seite 102)
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellenelementen und Information über die Anzahl der Elemente MB_CAMTABLE , Seite 102,
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Auswahl der Referenz (Antrieb oder Steuerung)

Tab. 3-44: Schnittstellenvariablen Funktionsbaustein MB_LoadCamData

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Standardwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
Permanent	BOOL			FALSE	Steigende Flanke an Execute
CamTableID	MC_CAM_ID			CAM_TABLE_1	Steigende Flanke an Execute

Tab. 3-45: Min-/Max- und Default-Werte Funktionsbaustein MB_LoadCamData

Die Enumeration MC_CAM_ID, RIL_CommonTypes, unterstützt die Auswahl der gewünschten Kurventabelle.

Element	Wert	Beschreibung
CAM_TABLE_1	1	Kurventabelle 1
CAM_TABLE_2	2	Kurventabelle 2
.....		
CAM_TABLE_255	255	Kurventabelle 255

Tab. 3-46: Elemente des Aufzählungstyps MC_CAM_ID

Die Struktur [MB_CAMTABLE](#), [Seite 102](#), besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

Funktionsbeschreibung



Bei der Verwendung des Zielsystems "MLD (IL2G)" sind die Ausführungen zu Kurvenscheiben-Downloads auf eine XLC/MLC nicht relevant.

Nach der Bearbeitungsfreigabe mittels "Execute" prüft der Baustein, in welchem Format die über "CamTableID" ausgewählte Kurvenscheibentabelle beschrieben werden kann.

Die in der Struktur "MB_CAMTABLE" enthaltene Kurvenscheibentabelle wird dann in dieses Format gewandelt und mit Hilfe eines WriteParameter-Bausteins in die Steuerung oder den Antrieb geladen.

Die Auswahl des Speicherortes (Steuerung oder Antrieb) der Kurvenscheibentabelle wird durch den Eingang "Axis " festgelegt.

Durch den Eingang "Permanent" kann zwischen der permanenten Ablage der Kurvenscheibe oder der temporären Ablage im flüchtigen Speicher (RAM) des Geräts entschieden werden. Die Realisierung der unterschiedlichen Ablage wird auf der Steuerung durch zwei verschiedene WriteParameter-Bausteine durchgeführt.

Der Funktionsbaustein setzt den Parameter S-0-0269 "Speicherungsmodus" des Antriebes auf den Wert "1"(nicht speichernd) wenn dieser mit dem Eingang "Permanent"= "false" aufgerufen wird.

Über das Element "NumberOfElements" der "MB_CAMTABLE" Struktur ist ersichtlich, wie viele gültige Kurvenscheibenpunkte das Array "CamTable" enthält. Ist über "CamTableID" eine Antriebskurvenscheibe mit <1024 Punkten ausgewählt, "NumberOfElements" signalisiert aber mehr Elemente, so wird ein Fehler generiert.

Die umgerechnete Kurvenscheibentabelle wird im Falle der fehlerfreien Bearbeitung des Funktionsbausteins an einen WriteParameter-Baustein übergeben und in die Steuerung oder den Antrieb geladen, wobei die Gültigkeit der Umrechnung und des Ladevorgangs durch den Ausgang "Done" signalisiert wird.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" sowie "Axis" werden im Fehlerfall nicht aktualisiert.

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins MB_LoadCamData weist folgende Eigenschaft auf:

- Der Algorithmus zur Umrechnung der Kurvenscheibenstützpunkte ist auf mehrere Zyklen, d. h. Aufrufe der Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden

	Kurvenscheibe in der Steuerung		Kurvenscheibe im Antrieb	
Axis	CntrlNo = LOCAL_CNTRL AxisNo = NO_OBJECT		CntrlNo = LOCAL_CNTRL AxisNo = AXIS_1..AXIS_99	
Permanent	FALSE (temporäre Kurvenscheibe)	TRUE (permanente Kurvenscheibe)	FALSE (temporäre Kurvenscheibe)	TRUE (permanente Kurvenscheibe)
CamTableID	1..99 (3-1024 Punkte)		1..4 (3-1024 Punkte); 5..8 (3-128 Punkte)	

Tab. 3-47: *Eingangsbelegungen des Funktionsbausteins MB_LoadCamData um die Kurvenscheibe in den Antrieb bzw. die Steuerung zu laden*

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#00000006	16#00000000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR	16#00002200	16#00000001	Eingang NumberOfElements ist kleiner als Minimum oder größer als Maximum
INPUT_INVALID_ERROR	16#00002201	16#00000001	Adresse von Feld CamTable ist null
INPUT_INVALID_ERROR	16#00002210	16#00000001	Eingang CamTableID ist zu groß für Auswahl vom Eingang Axis
INPUT_INVALID_ERROR	16#00002210	16#00000002	Eingang NumberOfElements ist zu groß für ausgewählte CamTableID

Tab. 3-48: *Fehlercodes Funktionsbaustein MB_LoadCamData*

3.5.3 MB_CAMPROF

Typdefinitionen Die Enumeration wird verwendet, um zwischen verschiedenen Kurvenscheibenprofilen zu wählen. Die Auswahl CAMPROF_STANDARD und CAMPROF_ADDITIVE beinhalten das alte Format, bei dem das letzte Element der Kurvenscheibentabelle der Stützpunkt $\varphi = 360^\circ - d\varphi$ ist. Während die Auswahl CAMPROF_ADDITIVE_SEGMENTABLE das neue Format beinhaltet, bei dem das letzte Element der Kurvenscheibentabelle der Stützpunkt $\varphi = 360^\circ$ ist.

Das verwendete Format kann bei Kurventabellen im Antrieb über den Parameter P-0-0086 gewählt werden. Bei Kurventabellen in der Steuerung ist nur das neue Format verfügbar.

Element	Wert	Beschreibung
CAMPROF_STANDARD	0	Kurvenscheibenprofil ohne Untersetzung, nicht segmentierbar (altes Format)
CAMPROF_ADDITIVE	1	Überlagerungsprofil mit Untersetzung, nicht segmentierbar (altes Format)
CAMPROF_ADDITIVE_SEGMENTABLE	2	Überlagerungsprofil mit Untersetzung, segmentierbar (neues Format)

Tab. 3-49: *Elemente des Aufzählungstyps MB_CAMPROF*

3.5.4 MB_CAMTABLE

Die Struktur besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche

die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
IrCamTable	ARRAY [0..1023] OF LREAL				Steigende Flanke an Execute	Feld mit den berechneten Kurvenscheibenpunkten
uiNumber- OfElements	UINT	3	1024		Steigende Flanke an Execute	Anzahl der gültigen Kurven- scheibenpunkte des Arrays CamTable

Tab. 3-50: Min-/Max-/Default-Werte der Struktur MB_CAMTABLE

3.6 PositionFiltering (Filtern inkrementeller Positionen)

3.6.1 Einführung und Übersicht

Der PositionFiltering-Baustein filtert eine inkrementelle Position und berechnet Position und Geschwindigkeit in LREAL. Die Position kann z.B. von einem Sercos-Encoder oder einem beliebigen anderen Geber eingelesen werden.

1. [MB_PositionFilterType01](#), Seite 103, filtert eine inkrementelle Position und berechnet Position und Geschwindigkeit in LREAL.

3.6.2 MB_PositionFilterType01

Kurzbeschreibung

Der Funktionsbaustein MB_PositionFilterType01 filtert eine inkrementelle Position und berechnet Position und Geschwindigkeit in LREAL. Die Position kann z.B. von einem Sercos-Encoder oder einem beliebigen anderen Eingang eingelesen werden.

Die Ausgänge des Funktionsbausteines können neben einer Verarbeitung im SPS-Programm auch als Eingänge für eine MLC-Reglerachse (über den Funktionsbaustein ML_ControlWriteActualData) verwendet werden.

Zuordnung Zielsystem/Bibliothek

MLC/XLC 14VRS	ML_TechMotionBase
Funktionspaket MOT	
MLD (IL2G)	MX_TechMotionBase

Tab. 3-51: Referenztabelle MB_PositionFilterType01

Schnittstellenbeschreibung

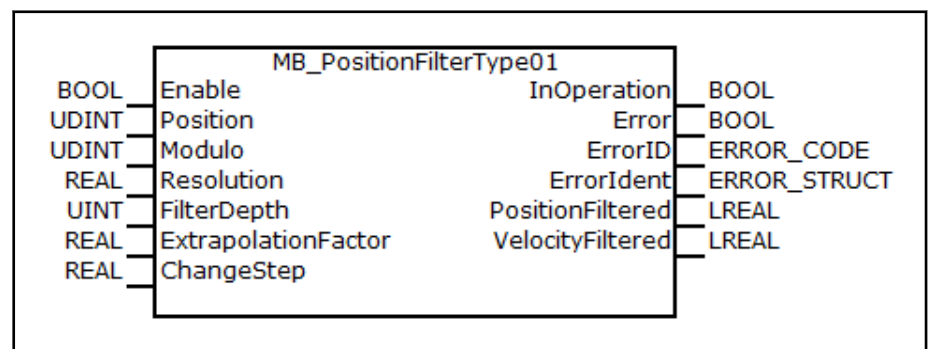


Abb. 3-42: Funktionsbaustein MB_PositionFilterType01

I/O-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (zyklisch, zustandsgesteuert)
	Position	UDINT	Nicht skalierte Position in Inkrementen. Wird zyklisch verarbeitet
	Modulo	UDINT	Modulowert in Inkrementen pro Umdrehung. 0 = absolut. Wird bei steigender Flanke am Enable gelatched
	Resolution	REAL	Auflösung in Inkrementen je Einheit, z.B. Inc/Grad oder Inc/mm. Wird bei steigender Flanke am Enable gelatched
	FilterDepth	UINT	Tiefe des Filters. Mittelwertbildung über N Werte. Wird zyklisch verarbeitet
	ExtrapolationFactor	REAL	Totzeitkompensation. Angabe in Anzahl Zyklen des Aufrufintervall des Funktionsbausteines. Wird zyklisch verarbeitet
	ChangeStep	REAL	Änderungen an den Eingängen "FilterDepth" und "ExtrapolationFactor" werden mit dieser Schrittweite verstellt, um Übergangsvorgänge klein zu halten. Wird bei steigender Flanke am Enable gelatched
VAR_OUTPUT	InOperation	BOOL	Der Baustein verarbeitet die Position zyklisch
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe Fehlerbehandlung , Seite 106)
	PositionFiltered	LREAL	Gefilterte Position. Einheit je nach Eingang "Resolution" z.B. in Grad oder mm
	VelocityFiltered	LREAL	Gefilterte Geschwindigkeit. Einheit je nach Eingängen "Resolution" und "Modulo" z.B. in U/min oder mm/min

Tab. 3-52: Schnittstellenvariablen Funktionsbaustein MB_PositionFilterType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Standardwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Position	UDINT			0	Kontinuierlich
Modulo	UDINT			0	Steigende Flanke an Enable
Resolution	REAL			0 (0 ist ungültig)	Steigende Flanke an Enable
FilterDepth	UINT	1	256	10	Kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ExtrapolationFactor	REAL	-16.0	+16.0	2.0 (z.B. für sercos Encoder)	Kontinuierlich
ChangeStep	REAL	0.01	10.0	0.1	Steigende Flanke an Enable

Tab. 3-53: Min-/Max- und Default-Werte Funktionsbaustein MB_PositionFilterType01

Funktionsbeschreibung

Nach der Bearbeitungsfreigabe mittels "Enable" prüft der Baustein die Eingänge auf gültige Werte. Sobald der Ausgang "InOperation" gesetzt wird, wird der Eingang "Position" mit Hilfe der Werte von "Resolution" und "Modulo" zyklisch in die Ausgänge für Position und Geschwindigkeit umgerechnet.



Der Funktionsbaustein muss zyklisch mit konstanter Zykluszeit aufgerufen werden, um korrekte Werte zu liefern. Es empfiehlt sich der Aufruf in einer motion-synchronen Task. Ist die Sercos-Zykluszeit kleiner als die Motion-Zykluszeit, bringt der Aufruf in einer Sercos-synchronen Task Vorteile in der Filterfunktion.

Der Eingang "FilterDepth" bestimmt die Anzahl der Zyklen, über die der gleitende Mittelwertfilter gerechnet wird. Die Filtertiefe kann während des Betriebs geändert werden, um z. B. den Filter je nach Geschwindigkeit unterschiedlich zu parametrieren. Die Filterlaufzeit wird automatisch kompensiert (extrapoliert).

Die Totzeitkompensation wird durch den Eingang "ExtrapolationFactor" parametrieren. Der Wert wird in Anzahl Zyklen des Aufrufintervall des Funktionsbausteines angegeben. Der Defaultwert von zwei Zyklen passt z.B. für einen Sercos-Encoder.

Durch den Eingang "ChangeStep" wird definiert, wie bei einer Änderung der Eingänge "FilterDepth" und/oder "ExtrapolationFactor" unter Freigabe die Werte schrittweise übernommen werden. Eine Änderung z.B. des "ExtrapolationFactor" kann am Ausgang zu einem Sprung in der Position führen. Durch einen kleinen "ChangeStep" werden die Übergangsvorgänge klein gehalten, z.B. um empfindliche Mechanik zu schonen.

Bei Werten kleiner Eins des "ChangeStep" wird eine Änderung der "FilterDepth" über mehrere Zyklen verteilt. Bei z.B. "ChangeStep" = 0.1 wird "FilterDepth" jeden zehnten Zyklus um 1 geändert.

Die Verstellung der Filterzeitkonstante ist erst nach mehreren Zyklen abgeschlossen. Wird z.B. die "FilterDepth" von 10 auf 16 geändert bei "ChangeStep" = 0.1 ist die Verstellung erst nach 60 Zyklen abgeschlossen.



Als Beispiel für die Anwendung des Funktionsbausteines MB_PositionFilterType01 steht das GAT Plug-In "SercosEncoderRealMaster" "Anbindung eines Sercos-Encoders mit einer Regelachse" auf dem Bosch Rexroth internen Motion-Logic Wiki zum Download zur Verfügung.

Mittelwertfilter

Der Funktionsbaustein MB_PositionFilterType01 implementiert einen gleitenden Mittelwertfilter mit Totzeitkompensation. Die Filtertiefe kann im Betrieb geändert werden.

Empfehlung: für den Stillstand und kleine Geschwindigkeiten eine große Filtertiefe einstellen. Für hohe Geschwindigkeiten dann die Filtertiefe auf kleinere Werte setzen.

Als Näherung für die Filtereckfrequenz kann folgende Formel verwendet werden:

$$f \approx \sqrt{2} / (N * T_{cyc})$$

f Filtereckfrequenz in Hertz

N Eingang FilterDepth

T_{cyc} Zykluszeit. Aufruf des Funktionsbausteines in Sekunden

Abb. 3-43: Berechnung der Filter Grenzfrequenz

Beispiel mit FilterDepth = 16 und Zykluszeit = 2ms:

$f \sim \sqrt{2} / (10 * 0.002s) \sim 70,71$ Hertz

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR	16#0002	16#0001	Eingang "Resolution" ist 0
INPUT_INVALID_ERROR	16#0002	16#0002	Eingang "FilterDepth" ist nicht im gültigen Bereich
INPUT_INVALID_ERROR	16#0002	16#0003	Eingang "ExtrapolationFactor" ist nicht im gültigen Bereich
INPUT_INVALID_ERROR	16#0002	16#0004	Eingang "ChangeStep" ist nicht im gültigen Bereich
INPUT_INVALID_ERROR	16#0002	16#0005	"Resolution" (Inkrement pro Grad) ist kleiner als "Modulo" (Inkrement pro Umdrehung).

Tab. 3-54: Fehlercodes FB MB_PositionFilterType01

3.7 Positionsüberwachung

3.7.1 Einführung und Übersicht

Die PositionMonitoring-Bausteine dienen der Positionsüberwachung zweier Achsen.

Die Bausteine sind in mehreren Ausführungen verfügbar:

1. [MB_PositionMonitoring_GantryCant, Seite 106](#), Überwachung der relativen Position zweier Achsen bezüglich Verkantung zueinander.
2. [MB_PositionMonitoring_PosCollision, Seite 109](#), Überwachung der absoluten und relativen Position zweier Achsen bezüglich Auszahnung und Kollision.

3.7.2 MB_PositionMonitoring_GantryCant

Kurzbeschreibung Der Funktionsbaustein "MB_PositionMonitoring_GantryCant" überwacht die relative Position zweier Achsen bezüglich Verkantung zueinander.

Mittels der Toleranzband-Funktionalität ist die Rückzugsbewegung möglich, wobei die Bewegungsrichtung aus der Verkantung weiter überwacht wird.



Der Funktionsbaustein "MB_PositionMonitoring_GantryCant (ML_TechMotion-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Der Funktionsbaustein eignet sich zur Überwachung von:

- Zustellachsen von Flexo-Druckwerken
- Gantry-Achsen mit mechanisch parallelen Verfahrenswegen u.a.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-55: Referenztable MB_PositionMonitoring_GantryCant

Schnittstellenbeschreibung

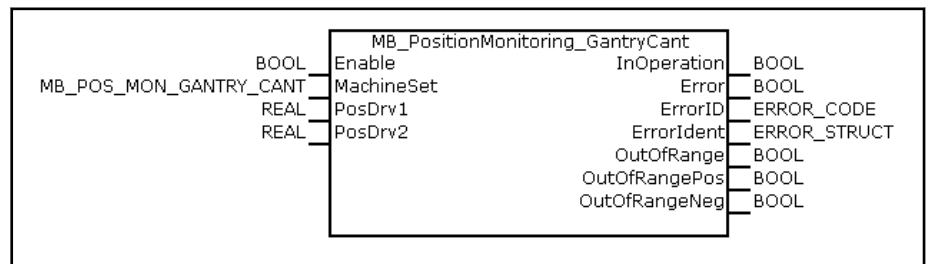


Abb. 3-44: Funktionsbaustein MB_PositionMonitoring_GantryCant

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (zyklisch, zustandsgesteuert)
	MachineSet	MB_POS_MON_GANTRY_CANT	maschinenspezifische Kenndaten MB_POS_MON_GANTRY_CANT, Seite 108
	PosDrv1	REAL	aktuelle Position der Achse 1
	PosDrv2	REAL	aktuelle Position der Achse 2
VAR_OUTPUT	InOperation	BOOL	Überwachung der Achsen ohne Fehler abgeschlossen, Ausgangssignale gültig
	Error	BOOL	Überwachung der Achsen mit Fehler abgeschlossen, Ausgangsvariable "ErrorIdent" gültig
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Beschreibung der Diagnose im Fehlerfall siehe Fehlerbehandlung, Seite 108
	OutOfRange	BOOL	Überschreitung des zulässigen Bereiches
	OutOfRangePos	BOOL	Überschreitung des zulässigen Bereiches, Achsposition der Achse 1 größer bzw. gleich der Achsposition der Achse 2
	OutOfRangeNeg	BOOL	Überschreitung des zulässigen Bereiches, Achsposition der Achse 1 kleiner der Achsposition der Achse 2

Tab. 3-56: Schnittstellenvariablen Funktionsbaustein MB_PositionMonitoring_GantryCant

Funktionsbeschreibung

Der Funktionsbaustein "MB_PositionMonitoring_GantryCant" überwacht nach Bearbeitungsfreigabe mittels "Enable" zyklisch die Achspositionen zweier Achsen bezüglich Verkantung zueinander. Überschreitet der Abstand der Achsen zueinander den maximal zulässigen Bereich "Range", wird dies über

die Ausgänge "OutOfRange" und "OutOfRangePos" bzw. "OutOfRangeNeg" signalisiert.

Im Folgezyklus wird der Ausgang "OutOfRange" zurück gesetzt, die Ausgänge "OutOfRangePos" bzw. "OutOfRangeNeg" signalisieren weiterhin bis zum Erreichen des zulässigen Abstandes abzüglich Toleranz, dass eine Positionskorrektur notwendig ist.

Wird innerhalb des Zeitraumes dieser Positionskorrektur der maximal erreichte Abstand zuzüglich Toleranz erneut bzw. weiter überschritten, wird dies durch erneutes Setzen des Ausganges "OutOfRange" signalisiert.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, wird dieser durch den Ausgang "Error" signalisiert sowie durch die, in diesem Fall aktualisierten Elemente der Ausgangsstruktur "ErrorIdent" spezifiziert. Die weiteren Ausgänge werden im Fehlerfall nicht aktualisiert.

- Fehlercodes

Der Funktionsbaustein "MB_PositionMonitoring_GantryCant" generiert im Fehlerfall eine umfangreiche Diagnose, die lokal interpretiert werden kann.
- Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170. Er kann in Additional1 und Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Eingänge liegen außerhalb des zulässigen Bereichs
STATE_MACHINE_ERROR (16#0005)	16#0006	16#????	ungültiger Zustand des Funktionsbausteins

Tab. 3-57: Fehlercodes Funktionsbaustein MB_PositionMonitoring_GantryCant

3.7.3 MB_POS_MON_GANTRY_CANT

Der Datentyp MB_POS_MON_GANTRY_CANT bildet den Maschinensatz des Funktionsbausteins MB_PositionMonitoring_GantryCant ab.

Die Struktur enthält folgende Daten:

Strukturelement	Typ	Min-Wert	Max-Wert	Default-Wert	Beschreibung
Range	REAL	$\geq 2 \cdot \text{Tolerance}$	n.def.	10.0	maximal zulässige Positionsdifferenz beider Achsen
Tolerance	REAL	0.0	n.def.	5.0	Toleranz der Grenzen

Tab. 3-58: Datenstruktur MB_POS_MON_GANTRY_CANT

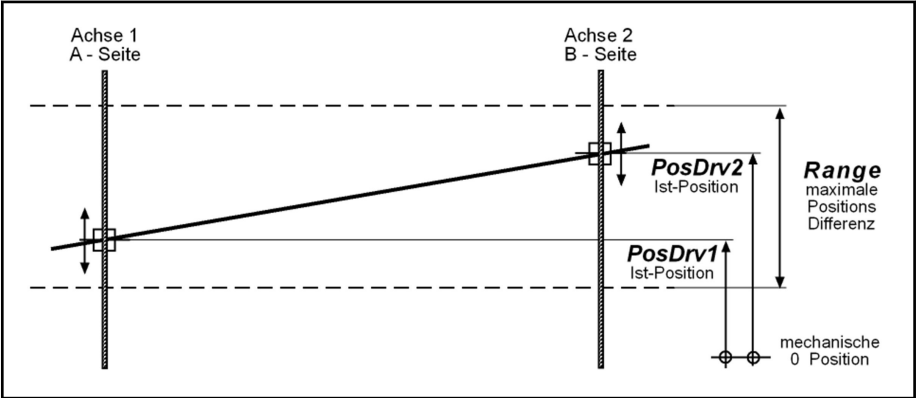


Abb. 3-45: Druckwerk Draufsicht – maximale Positionsdifferenzbildung (Range)

Die angegebene Toleranz muss nachfolgender Bedingung genügen:

$$Range > 2 * Tolerance$$

Abb. 3-46: Bedingung Maschinensatz Type 01

3.7.4 MB_PositionMonitoring_PosCollision

Kurzbeschreibung Der Funktionsbaustein "MB_PositionMonitoring_PosCollision" überwacht die absolute und relative Position zweier Achsen bezüglich Auszahnung und Kollision.



Der Funktionsbaustein "MB_PositionMonitoring_PosCollision" (ML_TechMotion-Bibliothek) wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-59: Referenztable MB_PositionMonitoring_PosCollision

Schnittstellenbeschreibung

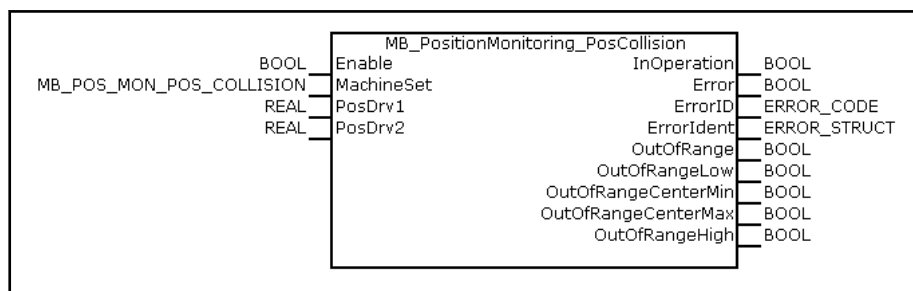


Abb. 3-47: Funktionsbaustein MB_PositionMonitoring_PosCollision

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (zyklisch, zustandsgesteuert)
	MachineSet	MB_POS_MON_POS_COLLISION	maschinenspezifische Kenndaten MB_POS_MON_POS_COLLISION, Seite 111
	PosDrv1	REAL	Aktuelle Position der Achse 1
	PosDrv2	REAL	Aktuelle Position der Achse 2

E/A-Typ	Name	Datentyp	Beschreibung
VAR_OUTPUT	InOperation	BOOL	Überwachung der Achsen ohne Fehler abgeschlossen, Ausgangssignale gültig
	Error	BOOL	Überwachung der Achsen mit Fehler abgeschlossen, Ausgangsvariable "ErrorIdent" gültig
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Beschreibung der Diagnose im Fehlerfall siehe Fehlercodes, Seite 111
	OutOfRange	BOOL	Überschreitung des zulässigen Bereiches
	OutOfRangeLow	BOOL	Überschreitung des zulässigen Bereiches ("DeltaLow")
	OutOfRangeCenterMin	BOOL	Überschreitung des zulässigen Bereiches ("DeltaCenterMin")
	OutOfRangeCenterMax	BOOL	Überschreitung des zulässigen Bereiches ("DeltaCenterMax")
	OutOfRangeHigh	BOOL	Überschreitung des zulässigen Bereiches ("DeltaHigh")

Tab. 3-60: Schnittstellenvariablen Funktionsbaustein MB_PositionMonitoring_PosCollision



Für den Funktionsbaustein "MB_PositionMonitoring_PosCollision" gelten nachfolgende Hinweise:

- Die Sollposition der Achse 2 ergibt sich aus den Kenngrößen des Maschinensatzes 2
- Die aktuelle Position der Achse 1 muss stets größer als die Summe der aktuellen Position der Achse 2 und der Radien beider Achsen sein
- Der Funktionsbaustein kann für drei oder mehr Achsen kaskadiert werden

Funktionsbeschreibung

Der Funktionsbaustein "MB_PositionMonitoring_PosCollision" überwacht nach Bearbeitungsfreigabe mittels "Enable" zyklisch die Achspositionen zweier Achsen bezüglich Auszahnung und Kollision. Überschreitet der Abstand der Achsen zueinander einen maximal zulässigen Bereich, wird dies über die Ausgänge "OutOfRange" und die entsprechenden Bereiche "OutOfRangeLow", "OutOfRangeCenterMin", "OutOfRangeCenterMax" und/oder "OutOfRangeHigh" signalisiert.

Im Folgezyklus wird der Ausgang "OutOfRange" zurück gesetzt, die weiteren Ausgänge signalisieren weiterhin bis zum Erreichen des jeweilig zulässigen Abstandes abzüglich Toleranz, dass eine Positions Korrektur notwendig ist.

Wird innerhalb des Zeitraumes dieser Positions Korrektur der maximal erreichte Abstand zuzüglich Toleranz erneut bzw. weiter überschritten, wird dies durch erneutes Setzen des Ausgangs "OutOfRange" signalisiert.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, wird dieser durch den Ausgang "Error" signalisiert sowie durch die, in diesem Fall aktualisierten Elemente der Ausgangsstruktur "ErrorIdent" spezifiziert. Die weiteren Ausgänge werden im Fehlerfall nicht aktualisiert.

Fehlercodes Der Funktionsbaustein MB_PositionMonitoring_PosCollision generiert im Fehlerfall eine umfangreiche Diagnose, welche lokal interpretiert werden kann.

Der Funktionsbaustein nutzt die Fehlertabelle "F-RELATED-TABLE", 16#0170.

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Eingänge liegen außerhalb des zulässigen Bereichs
STATE_MACHINE_ERROR	16#0006	16#????	Eingänge liegen außerhalb des zulässigen Bereichs
CALCULATION_ERROR	16#000C	16#0000	Berechnungsfehler; falsche Achsposition

Tab. 3-61: Fehlercodes Funktionsbaustein MB_PositionMonitoring_PosCollision

3.7.5 MB_POS_MON_POS_COLLISION

Datentypen Der Datentyp MB_POS_MON_POS_COLLISION bildet den Maschinensatz des Funktionsbausteins MB_PositionMonitoring_PosCollision ab.

Die Struktur enthält folgende Daten:

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Beschreibung
Position1	REAL	n.def.	n.def.	50.0	Sollposition der Achse 1
Diameter1	REAL	0.0	n.def.	20.0	Durchmesser der Achse 1
Diameter2	REAL	0.0	n.def.	20.0	Durchmesser der Achse 2
DeltaLow	REAL	0.0	n.def.	5.0	Begrenzung des Positionsoffsets der Achse 2
DeltaCenter-Min	REAL	0.0	$\leq \Delta \text{Center-Max} - 2 * \text{Tolerance}$	2.0	Minimale Distanz der Achsen 1 und 2 zueinander
DeltaCenter-Max	REAL	$\geq 2 * \text{Tolerance}$	n.def.	5.0	Maximale Distanz der Achsen 1 und 2 zueinander
DeltaHigh	REAL	$\geq 2 * \text{Tolerance}$	n.def.	5.0	Begrenzung Achsposition 2
Tolerance	REAL	0.0	n.def.	1.0	Toleranz der Grenzen

Tab. 3-62: Datenstruktur MB_POS_MON_POS_COLLISION

Die angegebene Toleranz muss nachfolgender Bedingung genügen:

$$\text{Kenngrößen} > 2 * \text{Tolerance}$$

Abb. 3-48: Bedingung Maschinensatz PosCollision

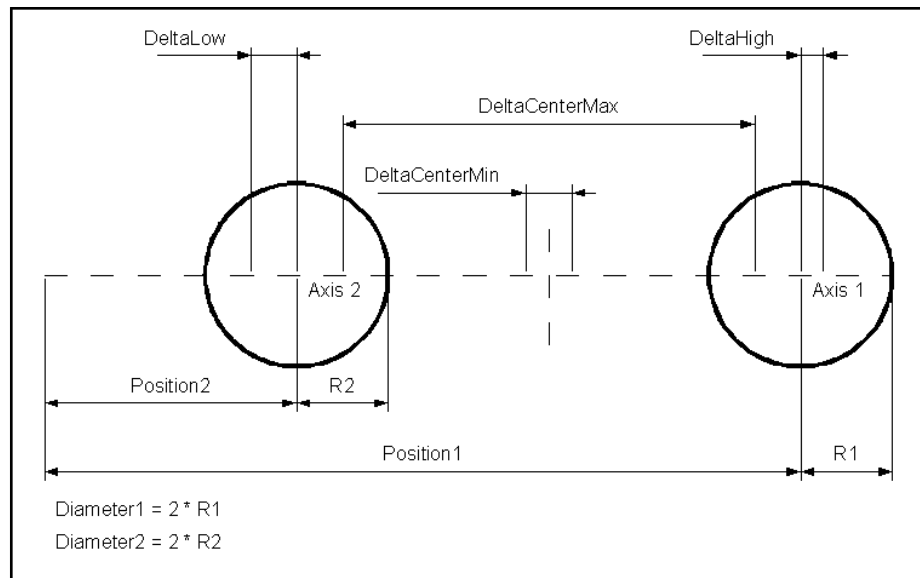


Abb. 3-49: Maschinensatz des Funktionsbausteins MB_PositionMonitoring_Pos-Collision

3.8 S-Kurve, Programmierung von Bewegungsabläufen

3.8.1 Einführung und Übersicht

Der Funktionsbaustein [MB_ReachVelocityType02](#), Seite 112, berechnet den entsprechenden Ruck, die Beschleunigung und die Zielgeschwindigkeit zum Abfahren bestimmter ruckbegrenzter Geschwindigkeitsverläufe (z.B. S-Kurve) und lässt hierbei bis zu 3 Änderungen der Zielgeschwindigkeiten zu.

3.8.2 MB_ReachVelocityType02

Kurzbeschreibung

Der Funktionsbaustein berechnet den entsprechenden Ruck, die Beschleunigung und die Zielgeschwindigkeit zum Abfahren bestimmter ruckbegrenzter Geschwindigkeitsverläufe (z.B. S-Kurve) und lässt hierbei bis zu 3 Änderungen der Zielgeschwindigkeiten zu.

Es kann eine Ruckbegrenzung von 0-100% ausgewählt werden. Ist der Eingabewert 0, so ist der Verlauf der Geschwindigkeit linear, bei 100% S-kurvenförmig. Aus der Beschleunigungsdauer, der aktuellen Geschwindigkeit, der Zielgeschwindigkeit und dem JerkFactor werden die entsprechenden Sollwerte für Geschwindigkeit, Beschleunigung und Ruck berechnet. Die Beschleunigungsphase kann über StopAcc gestoppt werden.

Die Zielgeschwindigkeit kann während des Beschleunigungsvorgangs geändert werden. Neue Sollwerte für Geschwindigkeit, Beschleunigung und Ruck werden berechnet, um die neue Zielgeschwindigkeit zu erreichen. In der restlichen Zeit wirkt immer der neue Ruck, deswegen ist der angelegte JerkFactor nach einer Änderung der Zielgeschwindigkeit nicht mehr gültig. Die gesamte Beschleunigungsdauer ändert sich nur dann, wenn die neue Zielgeschwindigkeit sehr nahe an der aktuellen Geschwindigkeit liegt.



Der Funktionsbaustein "ReachVelocityType02 (ML_TechMotionBibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-63: Referenztabelle MB_ReachVelocityType02

Schnittstellenbeschreibung

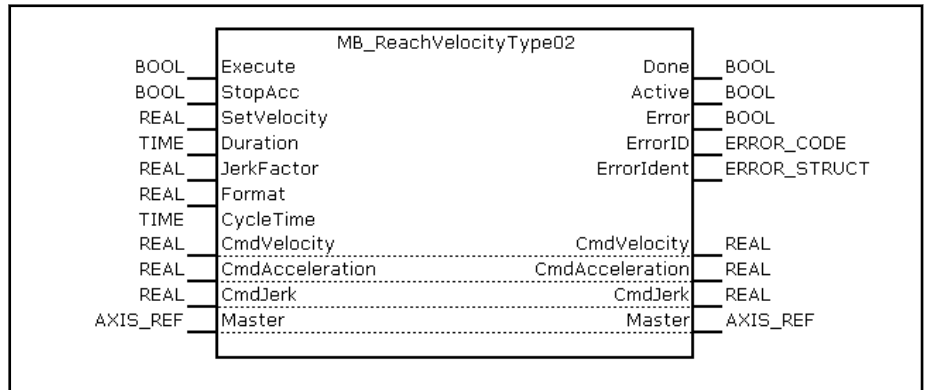


Abb. 3-50: Funktionsbaustein MB_ReachVelocityType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins
	StopAcc	BOOL	Stoppt weitere Beschleunigung
	SetVelocity	REAL	Zielgeschwindigkeit [m/min]
	Duration	TIME	Beschleunigungszeit
	CycleTime	TIME	Zykluszeit der Task, in der der FB aufgerufen wird
	JerkFactor	REAL	Verrampung des Beschleunigungsprofils [%]
	Format	REAL	Format [mm]
	CmdVelocity	REAL	Sollgeschwindigkeit [U/min]
VAR_OUTPUT	Done	BOOL	Bearbeitung abgeschlossen. Die Datenausgänge sind gültig
	Active	BOOL	Funktionsbaustein arbeitet und befindet sich in einem Zwischenzustand
	Error	BOOL	Signalisiert einen Fehler
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
VAR_IN_OUT	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 116
	CmdAcceleration	REAL	Sollbeschleunigung [rad/s ²]
	CmdJerk	REAL	Sollruck [rad/s ³]
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Referenz zur Leitachse Es wird empfohlen eine virtuelle Leitachse anzulegen

Tab. 3-64: Schnittstellenbeschreibung MB_ReachVelocityType02

Min-/Max- und Defaultwerte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
StopAcc	BOOL			FALSE	kontinuierlich
SetVelocity	REAL	n.def.	n.def.	0	kontinuierlich zwischen steigender Flanke an Execute und Done = TRUE
Duration	TIME	> 0 s	n.def.	30 s	steigende Flanke an Execute
JerkFactor	REAL	0	100	50	steigende Flanke an Execute
Format	REAL	> 0	n.def.	1000	steigende Flanke an Execute
CycleTime	TIME	> 1 ms	n.def.	10 ms	steigende Flanke an Execute

Tab. 3-65: Eingangsverhalten MB_ReachVelocityType02

Funktionsbeschreibung

Die Beschleunigungsphase startet mit einer steigenden Flanke an Execute. Beschleunigungsdauer (Duration), die anzufahrende Bahngeschwindigkeit (SetVelocity) sowie das Beschleunigungsprofil (JerkFactor) werden übernommen. Eine Besonderheit ist, dass Active bei steigender Flanke an Execute gesetzt wird, die Beschleunigungsphase aber erst nach einer Sekunde beginnt. D.h. nachfolgende Funktionsbausteine (beispielsweise der Registerregelung) können Active verwenden, um sich den Beginn einer Beschleunigungsphase anzeigen zu lassen. Nachdem die Zielgeschwindigkeit erreicht wurde, steht Active ebenfalls eine weitere Sekunde an, bis schließlich die Bearbeitung mit Done abgeschlossen wird.

Die Beschleunigungsdauer (Duration), die anzufahrende Bahngeschwindigkeit (SetVelocity), sowie das Beschleunigungsprofil (JerkFactor) werden für eine virtuelle Leitachse vorgegeben, da die zu referenzierende Achse am Master eine virtuelle Achse sein muss, um die gewünschte Genauigkeit gewährleisten zu können.

Um den Kurvenverlauf an einer realen Achse abzubilden, muss diese als Folgeachse synchron zur virtuellen Leitachse definiert werden.

S-Kurve

Der Ruck ist die zeitliche Ableitung der Beschleunigung. Der JerkFactor gibt den prozentualen zeitlichen Anteil der Beschleunigungsphase an, in der der Ruck ungleich Null ist.

In nachfolgender Abbildung werden 3 Beschleunigungsphasen gezeigt:

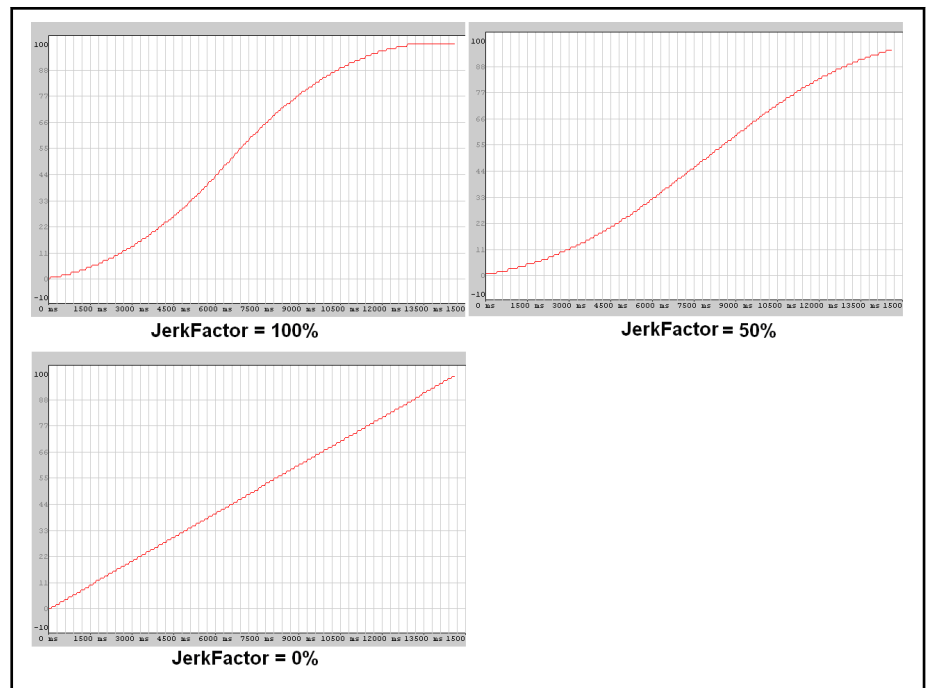


Abb. 3-51: Beschleunigungsphasen mit unterschiedlichem JerkFactor

Änderung an SetVelocity

Nach der Zielgeschwindigkeitsänderung, vor dem Erreichen der Endgeschwindigkeit, werden CmdVelocity, CmdAcceleration und CmdJerk neu berechnet, indem die neue Zielgeschwindigkeit (SetVelocity) übernommen wird und die aktuelle Zeit, Geschwindigkeit und Beschleunigung intern gemessen werden. Der Funktionsbaustein erlaubt eine dreimalige Änderung der Zielgeschwindigkeit.

Nachstehendes Bild veranschaulicht die Leitachsdrehzahl in U/min einer Beschleunigungsphase von 0 auf 200 m/min (SetVelocity) in 30 Sekunden (Duration) und einem Format von 1 m mit JerkFactor 30%. Ist die Geschwindigkeit größer als 50 m/min, wird SetVelocity auf 300 m/min verändert. Nach der Änderung wird der neue Ruck berechnet und die verbleibende Beschleunigungsphase verläuft S-kurvenförmig. Der ursprüngliche JerkFactor wird nach der ersten Änderung nicht mehr berücksichtigt.

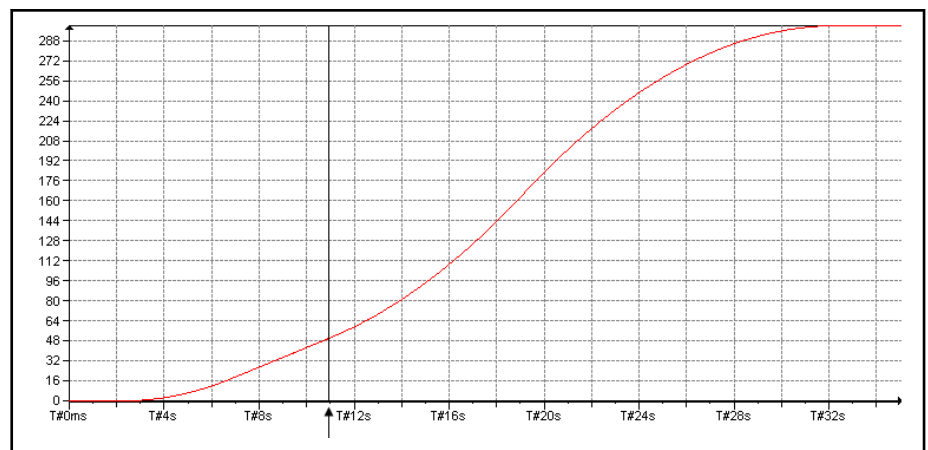


Abb. 3-52: Auswirkung der Änderung der Zielgeschwindigkeit

Stoppen der Beschleunigung

Die mit Execute angestoßene Beschleunigungsphase kann mit StopAcc sofort angehalten werden. Der Funktionsbaustein beendet augenblicklich die weitere Beschleunigung. Der Ausgang Done wird gesetzt. Soll der gesetzte

Geschwindigkeitswert SetVelocity dennoch angefahren werden, muss StopAcc deaktiviert und eine weitere steigende Flanke von Execute gesetzt werden. Das Erreichen des SetVelocity-Wertes wird bis zu einer fallenden Flanke an Execute durch den Ausgang Done signalisiert.

In nachfolgender Abbildung wird der Abbruch der linearen und der S-kurvenförmigen Beschleunigungsphase über StopAcc gezeigt:

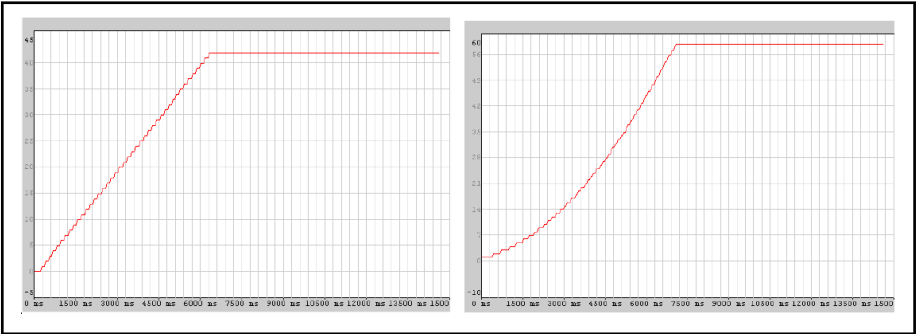


Abb. 3-53: Abbruch der Beschleunigungsphase (links Abbruch bei 6300 ms, rechts Abbruch bei 7000 ms)

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle F_RELATED_TABLE, 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000000	Ungültiges Druckzylinderformat
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000023	Ungültige Zykluszeit
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000024	Ungültige Zeitdauer
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000025	Ungültiger JerkFactor
INPUT_INVALID_ERROR (16#0001)	16#0000000D	16#00000000	AxisNo von "Master" ist außerhalb des gültigen Bereichs

Tab. 3-66: Fehlercodes von MB_ReachVelocityType02

3.9 Messtasterfunktionsbausteine

3.9.1 Einführung und Übersicht

Die Funktionsbausteine MB_InitTouchProbe, MB_TouchProbe, MB_WriteExpectWindow und MB_TouchProbeContinuous steuern und verwalten die von den Bosch Rexroth IndraDrive Antrieben unterstützte Messtasterfunktionalität.

Die Messtasterfunktionalität der Steuerungshardware wird über die folgenden Funktionsbausteine der ML_PLCopen-Bibliothek bereitgestellt:

Funktionsbaustein	Beschreibung
ML_InitTouchProbe	Aktiviert und konfiguriert den Messtaster in der Steuerung
ML_WriteExpectWindow	Konfiguriert das Erwartungsfenster für die Messtasterfunktionalität

Funktionsbaustein	Beschreibung
ML_TouchProbe	Aktiviert den ausgewählten Messtaster, wertet den Status aus und stellt die Messwerte nach dem Triggerereignis zur Verfügung
ML_TouchProbeContinuous	Aktiviert den ausgewählten Messtaster, wertet den Status aus und stellt die Messwerte nach dem Triggerereignis zur Verfügung
ML_AbortTrigger	Bricht eine aktive Messung ab

Tab. 3-67: Funktionsbausteine zur Messtasterfunktionalität der Steuerungshardware

Die Messtasterfunktionalität der Rexroth IndraDrive Antriebe wird über die folgenden Funktionsbausteine der ML_TechMotionBase-Bibliothek bereitgestellt:

Funktionsbaustein	Beschreibung
MB_InitTouchProbe, Seite 127	Aktiviert und konfiguriert den Messtaster im Antrieb
MB_WriteExpectWindow, Seite 142	Konfiguriert das Erwartungsfenster für die Messtasterfunktionalität
MB_TouchProbe, Seite 135	Aktiviert den ausgewählten Messtaster, wertet den Status aus und stellt die Messwerte nach dem Triggerereignis einmalig zur Verfügung
MB_TouchProbeContinuous, Seite 138	Aktiviert den ausgewählten Messtaster, wertet den Status aus und stellt die Messwerte nach dem Triggerereignis fortlaufend zur Verfügung
MB_AbortTrigger, Seite 145	Bricht eine aktive Messung ab
Datentyp	
MB_PROBE_EXECUTION_MODE, Seite 135	Auswahl, ob einzelne oder fortlaufende Messung

Tab. 3-68: Funktionsbausteine zur Messtasterfunktionalität des Rexroth IndraDrive

Die folgende Dokumentation befasst sich mit den Funktionsbausteinen zur Messtasterfunktionalität des Rexroth IndraDrive. Die Funktionsbausteine zur Messtasterfunktionalität der Steuerung werden in den Dokumentationen "MLC 14VRS Funktionsbeschreibung", "Rexroth IndraLogic XLC 14VRS Funktionsbeschreibung" und "MLC/XLC PLCopen Funktionsbausteine und Datentypen" beschrieben.

Die Messtasterfunktionalität des IndraDrives erlaubt das Erfassen der Positionen oder der internen Systemzeit in [us] bei der Instanz des schnellen, digitalen Eingangssignals. Der Messtasterfunktionsbaustein verwendet diese Antriebsfunktionalität und stellt eine Benutzerschnittstelle bereit. Die folgenden Schritte sind notwendig um die Messtasterfunktionsbausteine zu verwenden:

3.9.2 Messtasterfunktionsbausteine – Komponenten und Parametrierung

Hardware

Die folgenden Bosch Rexroth Hardwarekomponenten werden benötigt:

- MLC/XLC
- IndraDrive C oder IndraDrive M mit Advanced oder Basic Performance

Die folgenden Steuerungen werden unterstützt:

- ADVANCED (Typenschlüssel: CSB01.XC-...)
- BASIC Sercos (Einzelachse; Typenschlüssel: CSB01.XN-SE-...)
- BASIC PROFIBUS (Einzelachse; Typenschlüssel: CSB01.XN-PB-...)
- BASIC UNIVERSAL (Einzelachse; Typenschlüssel: CSB01.XC-...)

Firmware

Die folgende Firmware wird benötigt und mit den oben genannten Hardwarekomponenten von Bosch Rexroth verwendet:

- Steuerungsfirmware 14VRS
- Antriebsfirmware MPx07VRS oder höher

Parametrierung der Messtasterfunktionsbausteine für MLC/XLC und MLD

Die folgenden Antriebsparametrierschritte sind notwendig um die Technologiefunktionsbausteine MB_InitTouchProbe, MB_TouchProbe und MB_TouchProbeContinuous zu verwenden:

MLC/XLC-Konfiguration

Die folgenden Antriebsparametereinstellungen sind notwendig, wenn die Messtasterfunktionalität in der MLC/XLC verwendet wird:

Öffnen Sie IndraWorks und gehen Sie mit dem entsprechenden Projekt Online.

- Nehmen Sie konfigurierte Messtastersteuerbits in das Signalsteuerwort mit auf.
- **IndraWorks ► Projekt ► Motion ► Reale Achsen ► Rechtsklick auf Achse: Kommunikation ► Signalsteuerwort**
 - Messtaster 1: S-0-0405, Bit 0
 - Messtaster 2: S-0-0406, Bit 0

	Status	Quellparameter	Bitnummer
Bit 0	1	S-0-0346: Positioniersteuerwort	0
Bit 1	0	S-0-0346: Positioniersteuerwort	3
Bit 2	0	S-0-0346: Positioniersteuerwort	4
Bit 3	0	S-0-0346: Positioniersteuerwort	5
Bit 4	0	S-0-0393: Sollwertmodus	0
Bit 5	0	S-0-0393: Sollwertmodus	1
Bit 6	0	P-0-0088: Steuerwort Synchronbetriebsarten	0
Bit 7	0	P-0-0088: Steuerwort Synchronbetriebsarten	1
Bit 8	0	P-0-0155: Aufsynchonisier-Modus	1
Bit 9	0	P-0-0154: Aufsynchonisier-Richtung	0
Bit 10	0	P-0-0154: Aufsynchonisier-Richtung	1
Bit 11	0	S-0-0000: <reserviert>	0
<Achse>.UserCmdDataBitA	0	S-0-0405: Messtaster 1 Freigabe	0
<Achse>.UserCmdDataBitB	0	S-0-0000: <leer>	0
<Achse>.UserCmdDataBitC	0	S-0-0000: <leer>	0
<Achse>.UserCmdDataBitD	0	S-0-0000: <leer>	0

Anwenden

Abb. 3-54: Signalsteuerwort

- Nehmen Sie konfigurierte Messtasterstatusbits in das Signalstatuswort mit auf.

IndraWorks ► Projekt ► Motion ► Reale Achsen ► Rechtsklick auf Achse: Kommunikation ► Signalstatuswort

- Messtaster 1, positiv erfasst: S-0-0409, Bit 0
- Messtaster 1, negativ erfasst: S-0-0410, Bit 0
- Messtaster 2, positiv erfasst: S-0-0411, Bit 0
- Messtaster 2, negativ erfasst: S-0-0412, Bit 0

	Status	Quellparameter	Bitnummer
Bit 0	1	S-0-0330: Meldung n_ist = n_soll	0
Bit 1	1	S-0-0331: Meldung n_ist = 0	0
Bit 2	0	S-0-0437: Positionierstatus	2
Bit 3	0	S-0-0403: Status Lageistwerte	1
Bit 4	0	S-0-0419: Positioniersollwert-Quittung	0
Bit 5	0	P-0-0089: Statuswort Synchronbetriebsarten	0
Bit 6	0	P-0-0089: Statuswort Synchronbetriebsarten	1
Bit 7	1	P-0-0089: Statuswort Synchronbetriebsarten	8
Bit 8	0	S-0-0824: Meldung Drehmoment-/Kraftsollwert erreicht	0
Bit 9	0	P-0-0089: Statuswort Synchronbetriebsarten	5
Bit 10	1	S-0-0336: Meldung In-Position	0
Bit 11	1	P-0-0089: Statuswort Synchronbetriebsarten	6
<Achse>.UserActualDataBitA	0	S-0-0409: Messtaster 1 positiv erfasst	0
<Achse>.UserActualDataBitB	0	S-0-0410: Messtaster 1 negativ erfasst	0
<Achse>.UserActualDataBitC	0	S-0-0000: <leer>	0
<Achse>.UserActualDataBitD	0	S-0-0000: <leer>	0

Anwenden

Abb. 3-55: Signalstatuswort

- Nehmen Sie konfigurierte Messtasterwerte in das optionale, zyklische AT-Telegramm auf.

IndraWorks ► Projekt ► Motion ► Reale Achsen ► Rechtsklick auf Achse: Kommunikation ► Zyklischer Sercos-Kanal

- Messtaster 1, positive Flanke: S-0-0130
- Messtaster 1, negative Flanke: S-0-0131

ML_TechMotionBase.library // MX_TechMotionBase.library

- Messtaster 2, positive Flanke: S-0-0132
- Messtaster 2, negative Flanke: S-0-0133
- Messtaster 1, Anzahl der Markenausfälle: P-0-0224
- Messtaster 2, Anzahl der Markenausfälle: P-0-0225
- Messtaster 1, Differenzwert des Messtasters: P-0-0202
- Messtaster 2, Differenzwert des Messtasters: P-0-0203

Konfiguration zyklischer Schreibzugriff (MDT)			
<Achse>.UserCmdDataA	S-0-0000: <leer>		
<Achse>.UserCmdDataB	S-0-0000: <leer>		
<Achse>.UserCmdDataC	S-0-0000: <leer>		
<Achse>.UserCmdDataD	S-0-0000: <leer>		

Konfiguration zyklischer Lesezugriff (AT)			
<Achse>.UserActualDataA	S-0-0130: Messwert 1 positive Flanke	224,9635	Grad
<Achse>.UserActualDataB	S-0-0131: Messwert 1 negative Flanke	0,0000	Grad
<Achse>.UserActualDataC	P-0-0202: Differenz Messwerte 1	135,0365	Grad
<Achse>.UserActualDataD	P-0-0224: Messtaster 1, Anzahl Markenausfälle	0	--

Abb. 3-56: Konfigurieren eines zyklischen AT-Telegramms

Konfiguration der MLD-S bzw. MLD-M (Leitachse)

Die folgenden Antriebsparametereinstellungen sind notwendig, um die Messtasterfunktionalität innerhalb einer MLD-S bzw. MLD-M als Leitachse zu verwenden:

- Nehmen Sie konfigurierte Messtastersteuerbits in das Signalsteuerwort mit auf.

IndraWorks ► Projekt ► MLD ► Rechtsklick auf: AxisData

- Messtaster 1: S-0-0405, Bit 0
- Messtaster 2: S-0-0406, Bit 0

Achse #1 (Lokale Achse) ▼

Sollwerte		
wUserCmdDataBitA_q	S-0-0405 : Messtaster 1 Freigabe	0 ▼
wUserCmdDataBitB_q	S-0-0406 : Messtaster 2 Freigabe	0 ▼
wUserCmdDataBitC_q	S-0-0000 : Frei	0 ▼
wUserCmdDataBitD_q	S-0-0000 : Frei	0 ▼
dwUserCmdDataA_q	S-0-0000 : Frei	
dwUserCmdDataB_q	S-0-0000 : Frei	
dwUserCmdDataC_q	S-0-0000 : Frei	
dwUserCmdDataD_q	S-0-0000 : Frei	

Istwerte		
wUserActualDataBitA_i	S-0-0409 : Messtaster 1 positiv erfasst	0 ▼
wUserActualDataBitB_i	S-0-0410 : Messtaster 1 negativ erfasst	0 ▼
wUserActualDataBitC_i	S-0-0000 : Frei	0 ▼
wUserActualDataBitD_i	S-0-0000 : Frei	0 ▼
dwUserActualDataA_i	S-0-0130 : Messwert 1 positive Flanke	
dwUserActualDataB_i	S-0-0131 : Messwert 1 negative Flanke	
dwUserActualDataC_i	P-0-0202 : Differenz Messwerte 1	
dwUserActualDataD_i	P-0-0224 : Messtaster 1, Anzahl Markenausfälle	

Konfiguration übernehmen

Abb. 3-57: Sollwertdatenbits

- Nehmen Sie konfigurierte Messtasterstatusbits in das Signalstatuswort mit auf.

IndraWorks ► Rechtsklick auf Sercos III ► Querkommunikation Antriebe ► Signalstatuswort

- Messtaster 1, positiv erfasst: S-0-0409, Bit 0
- Messtaster 1, negativ erfasst: S-0-0410, Bit 0
- Messtaster 2, positiv erfasst: S-0-0411, Bit 0
- Messtaster 2, negativ erfasst: S-0-0412, Bit 0

Bit	Status	Quellparameter	Quellbit
0	0	P-0-0115 : Gerätesteuerung: Statuswort	12
1	0	S-0-0331 : Meldung n_ist = 0	0
2	0	S-0-0403 : Status Lageistwerte	0
3	0	S-0-0419 : Positioniersollwert-Quittung	0
4	0	S-0-0521 : Lageregler-Statuswort	0
5	0	S-0-0521 : Lageregler-Statuswort	2
6	0	P-0-0152 : Aufsynchonisieren beendet	0
7	0	P-0-0115 : Gerätesteuerung: Statuswort	7
8	0	S-0-0409 : Messtaster 1 positiv erfasst	0
9	0	S-0-0410 : Messtaster 1 negativ erfasst	0
10	0	S-0-0411 : Messtaster 2 positiv erfasst	0
11	0	S-0-0412 : Messtaster 2 negativ erfasst	0
12	0	S-0-0409 : Messtaster 1 positiv erfasst	0
13	0	S-0-0410 : Messtaster 1 negativ erfasst	0
14	0	S-0-0411 : Messtaster 2 positiv erfasst	0
15	0	S-0-0412 : Messtaster 2 negativ erfasst	0

Die Bits 12 bis 15 entsprechen den Elementen wUserActualDataBitA_i bis wUserActualDataBitD_i in der Struktur MX_AXISDATA

[Einstellungen](#) [CCD-Signalsteuerwort](#) [Konfiguration übernehmen](#)

Abb. 3-58: Istwertdatenbits

- Nehmen Sie konfigurierte Messtasterwerte in das optionale, zyklische AT-Telegramm auf.

IndraWorks ► Projekt ► MLD ► Rechtsklick auf: AxisData

- Messtaster 1, positive Flanke: S-0-0130
- Messtaster 1, negative Flanke: S-0-0131
- Messtaster 2, positive Flanke: S-0-0132
- Messtaster 2, negative Flanke: S-0-0133
- Messtaster 1, Anzahl der Markenausfälle: P-0-0224
- Messtaster 2, Anzahl der Markenausfälle: P-0-0225
- Messtaster 1, Differenzwert des Messtasters: P-0-0202
- Messtaster 2, Differenzwert des Messtasters: P-0-0203
- Nehmen Sie die folgenden Parameter für jedes konfigurierte, gemessene Messtastersignal in den Istwertdatencontainer auf:
 - TP_ACT_POS_FEEDBACK1 => Keine zusätzlichen Daten werden benötigt
 - TP_ACT_POS_FEEDBACK2 => Keine zusätzlichen Daten werden benötigt
 - TP_ACT_POSIN_ACT_CYCLE => Keine zusätzlichen Daten werden benötigt
 - TP_SYNC_CMD_POS => P-0-0779
 - TP_PROBE_TIME => Keine zusätzlichen Daten werden benötigt
 - TP_ACT_MEASURING_ENC_POS => P-0-0332
 - TP_RES_MASTER_AXIS_POS => P-0-0764

- TP_EFF_MASTER_AXIS_POS => P-0-0777
- TP_CAM_SHAFT_PROF_ACCESS_ANGLE => P-0-0777

Achse #1 (Lokale Achse)

Sollwerte

wUserCmdDataBitA_q	S-0-0405 : Messtaster 1 Freigabe	0
wUserCmdDataBitB_q	S-0-0406 : Messtaster 2 Freigabe	0
wUserCmdDataBitC_q	S-0-0000 : Frei	0
wUserCmdDataBitD_q	S-0-0000 : Frei	0
dwUserCmdDataA_q	S-0-0000 : Frei	
dwUserCmdDataB_q	S-0-0000 : Frei	
dwUserCmdDataC_q	S-0-0000 : Frei	
dwUserCmdDataD_q	S-0-0000 : Frei	

Istwerte

wUserActualDataBitA_i	S-0-0409 : Messtaster 1 positiv erfasst	0
wUserActualDataBitB_i	S-0-0410 : Messtaster 1 negativ erfasst	0
wUserActualDataBitC_i	S-0-0000 : Frei	0
wUserActualDataBitD_i	S-0-0000 : Frei	0
dwUserActualDataA_i	S-0-0130 : Messwert 1 positive Flanke	
dwUserActualDataB_i	S-0-0131 : Messwert 1 negative Flanke	
dwUserActualDataC_i	P-0-0202 : Differenz Messwerte 1	
dwUserActualDataD_i	P-0-0224 : Messtaster 1, Anzahl Markenausfälle	

Konfiguration übernehmen

Abb. 3-59: Sollwertdatenbits

- Setzen Sie P-0-1367, Bit 6 (Update der verwendeten Achsdatenstruktur aktivieren, nur MLD-M)

Konfiguration der MLD-M (Folgeachse)

Die folgenden Antriebsparametereinstellungen sind notwendig, wenn die Messtasterfunktionalität in der MLD-M als Folgeachse verwendet wird:

- Nehmen Sie konfigurierte Messtastersteuerbits in das Signalsteuerwort mit auf.

IndraWorks ► Rechtsklick auf Sercos III ► Querkommunikation Antriebe ► Signalsteuerwort

- Messtaster 1: S-0-0405, Bit 0
- Messtaster 2: S-0-0406, Bit 0

Achse #1 (Lokale Achse) ▼

Bit	Status	Quellparameter	Quellbit
0	0	S-0-0520 : Achsregler-Steuerwort	0
1	0	S-0-0520 : Achsregler-Steuerwort	2
2	0	P-0-0155 : Aufsynchonisier-Modus	1
3	0	P-0-0154 : Aufsynchonisier-Richtung	0
4	0	P-0-0154 : Aufsynchonisier-Richtung	1
5	0	S-0-0000 : Frei	0
6	0	S-0-0000 : Frei	0
7	0	S-0-0000 : Frei	0
8	0	S-0-0000 : Frei	0
9	0	S-0-0000 : Frei	0
10	0	S-0-0000 : Frei	0
11	0	S-0-0000 : Frei	0
12	0	S-0-0405 : Messtaster 1 Freigabe ▼	0 ▼
13	0	S-0-0406 : Messtaster 2 Freigabe ▼	0 ▼
14	0	S-0-0000 : Frei ▼	0 ▼
15	0	S-0-0000 : Frei ▼	0 ▼

Die Bits 12 bis 15 entsprechen den Elementen
wUserCmdDataBitA_q bis wUserCmdDataBitD_q in der Struktur MX_AXISDATA

Abb. 3-60: Signalsteuerwort

- Nehmen Sie konfigurierte Messtasterstatusbits in das Signalstatuswort mit auf.

IndraWorks ► Rechtsklick auf Sercos III ► Querkommunikation Antriebe ► Signalstatuswort

- Messtaster 1, positiv erfasst: S-0-0409, Bit 0
- Messtaster 1, negativ erfasst: S-0-0410, Bit 0
- Messtaster 2, positiv erfasst: S-0-0411, Bit 0
- Messtaster 2, negativ erfasst: S-0-0412, Bit 0

Bit	Status	Quellparameter	Quellbit
0	0	P-0-0115 : Gerätesteuerung: Statuswort	12
1	0	S-0-0331 : Meldung n_ist = 0	0
2	0	S-0-0403 : Status Lageistwerte	0
3	0	S-0-0419 : Positioniersollwert-Quittung	0
4	0	S-0-0521 : Lageregler-Statuswort	0
5	0	S-0-0521 : Lageregler-Statuswort	2
6	0	P-0-0152 : Aufsynchronisieren beendet	0
7	0	P-0-0115 : Gerätesteuerung: Statuswort	7
8	0	S-0-0409 : Messtaster 1 positiv erfasst	0
9	0	S-0-0410 : Messtaster 1 negativ erfasst	0
10	0	S-0-0411 : Messtaster 2 positiv erfasst	0
11	0	S-0-0412 : Messtaster 2 negativ erfasst	0
12	0	S-0-0409 : Messtaster 1 positiv erfasst	0
13	0	S-0-0410 : Messtaster 1 negativ erfasst	0
14	0	S-0-0411 : Messtaster 2 positiv erfasst	0
15	0	S-0-0412 : Messtaster 2 negativ erfasst	0

Die Bits 12 bis 15 entsprechen den Elementen wUserActualDataBitD_i bis wUserActualDataBitD_i in der Struktur MX_AXISDATA

Einstellungen CCD-Signalsteuerwort Konfiguration übernehmen

Abb. 3-61: Signalstatuswort

- Nehmen Sie konfigurierte Messtasterwerte in das optionale, zyklische AT-Telegramm auf.

IndraWorks ► Rechtsklick auf Sercos III ► Querkommunikation Antriebe ► Zyklischer Sercos-Kanal

- Messtaster 1, positive Flanke: S-0-0130
- Messtaster 1, negative Flanke: S-0-0131
- Messtaster 2, positive Flanke: S-0-0132
- Messtaster 2, negative Flanke: S-0-0133
- Messtaster 1, Anzahl der Markenausfälle: P-0-0224
- Messtaster 2, Anzahl der Markenausfälle: P-0-0225
- Messtaster 1, Differenzwert des Messtasters: P-0-0202
- Messtaster 2, Differenzwert des Messtasters: P-0-0203
- Nehmen Sie die folgenden Parameter für jedes konfigurierte, gemessene Messtastersignal in den optionalen, zyklischen ATs auf:
 - TP_ACT_POS_FEEDBACK1 => Keine zusätzlichen Daten werden benötigt
 - TP_ACT_POS_FEEDBACK2 => Keine zusätzlichen Daten werden benötigt
 - TP_ACT_POSIN_ACT_CYCLE => Keine zusätzlichen Daten werden benötigt
 - TP_SYNC_CMD_POS => P-0-0779
 - TP_PROBE_TIME => Keine zusätzlichen Daten werden benötigt
 - TP_ACT_MEASURING_ENC_POS => P-0-0332
 - TP_RES_MASTER_AXIS_POS => P-0-0764

ML_TechMotionBase.library // MX_TechMotionBase.library

- TP_EFF_MASTER_AXIS_POS => P-0-0777
- TP_CAM_SHAFT_PROF_ACCESS_ANGLE => P-0-0777

Abb. 3-62: Konfigurieren eines zyklischen AT-Telegramms

- Setzen Sie P-0-1367, Bit 6 (Update der verwendeten Achsdatenstruktur aktivieren, nur MLD-M)

Konfiguration Probe1

Messtaster 1	AT-Container	Steuerbit	Statusbit
Positive Flanke	S-0-0130	S-0-0405.0	S-0-0409.0
Negative Flanke	S-0-0131	S-0-0405.0	S-0-0410.0
Positive und negative Flanke, Differenz, Markenausfall	S-0-0130, S-0-0131, P-0-0224 UND P-0-0202	S-0-0405.0	S-0-0409.0, S-0-0410.0

Tab. 3-69: Konfiguration Probe1

Konfiguration Probe2

Probe2	AT-Container	Steuerbit	Statusbit
Positive Flanke	S-0-0132	S-0-0406.0	S-0-0411.0
Negative Flanke	S-0-0133	S-0-0406.0	S-0-0412.0
Positive und negative Flanke, Differenz, Markenausfall	S-0-0132, S-0-0133, P-0-0225 UND P-0-0203	S-0-0406.0	S-0-0411.0, S-0-0412.0

Tab. 3-70: Konfiguration Probe2

Beispielkonfiguration des Signalstatuswortes, Signalsteuerwortes und zyklischen Sercos-Kanals

- IndraDrive Probe1
 - Negative Flanke
 - Erwartungsfenster deaktiviert
 - Markenausfall deaktiviert
 - Differenzmessung deaktiviert
1. Konfiguration eines Signalsteuerwortes

	Status	Quellparameter	Bitnummer
Bit 0	1	S-0-0346: Positioniersteuerwort	0
Bit 1	0	S-0-0346: Positioniersteuerwort	3
Bit 2	0	S-0-0346: Positioniersteuerwort	4
Bit 3	0	S-0-0346: Positioniersteuerwort	5
Bit 4	0	S-0-0393: Sollwertmodus	0
Bit 5	0	S-0-0393: Sollwertmodus	1
Bit 6	0	P-0-0088: Steuerwort Synchronbetriebsarten	0
Bit 7	0	P-0-0088: Steuerwort Synchronbetriebsarten	1
Bit 8	0	P-0-0155: Aufsynchonisier-Modus	1
Bit 9	0	P-0-0154: Aufsynchonisier-Richtung	0
Bit 10	0	P-0-0154: Aufsynchonisier-Richtung	1
Bit 11	0	S-0-0000: <reserviert>	0
<Achse>.UserCmdDataBitA	1	S-0-0405: Messtaster 1 Freigabe	0
<Achse>.UserCmdDataBitB	0	S-0-0000: <leer>	0
<Achse>.UserCmdDataBitC	0	S-0-0000: <leer>	0
<Achse>.UserCmdDataBitD	0	S-0-0000: <leer>	0

Anwenden

Abb. 3-63: Beispiel eines Signalsteuerwortes

2. Konfiguration eines Signalstatuswortes

	Status	Quellparameter	Bitnummer
Bit 0	1	S-0-0330: Meldung n_ist = n_soll	0
Bit 1	1	S-0-0331: Meldung n_ist = 0	0
Bit 2	0	S-0-0437: Positionierstatus	2
Bit 3	0	S-0-0403: Status Lageistwerte	1
Bit 4	0	S-0-0419: Positioniersollwert-Quittung	0
Bit 5	0	P-0-0089: Statuswort Synchronbetriebsarten	0
Bit 6	0	P-0-0089: Statuswort Synchronbetriebsarten	1
Bit 7	0	P-0-0089: Statuswort Synchronbetriebsarten	8
Bit 8	0	S-0-0824: Meldung Drehmoment-/Kraftsollwert erreicht	0
Bit 9	1	P-0-0089: Statuswort Synchronbetriebsarten	5
Bit 10	1	S-0-0336: Meldung In-Position	0
Bit 11	1	P-0-0089: Statuswort Synchronbetriebsarten	6
<Achse>.UserActualDataBitA	1	S-0-0410: Messtaster 1 negativ erfasst	0
<Achse>.UserActualDataBitB	0	S-0-0000: <leer>	0
<Achse>.UserActualDataBitC	0	S-0-0000: <leer>	0
<Achse>.UserActualDataBitD	0	S-0-0000: <leer>	0

Anwenden

Abb. 3-64: Beispiel eines Signalstatuswortes

3. Konfiguration eines zyklischen Sercos-Kanals

Konfiguration zyklischer Schreibzugriff (MDT)

<Achse>.UserCmdDataA	S-0-0000: <leer>	
<Achse>.UserCmdDataB	S-0-0000: <leer>	
<Achse>.UserCmdDataC	S-0-0000: <leer>	
<Achse>.UserCmdDataD	S-0-0000: <leer>	

Konfiguration zyklischer Lesezugriff (AT)

<Achse>.UserActualDataA	S-0-0131: Messwert 1 negative Flanke	0.0000	Deg
<Achse>.UserActualDataB	S-0-0000: <leer>		
<Achse>.UserActualDataC	S-0-0000: <leer>		
<Achse>.UserActualDataD	S-0-0000: <leer>		

Abb. 3-65: Beispiel eines zyklischen Sercos-Kanals

3.9.3 MB_InitTouchProbe

Kurzbeschreibung

Der Funktionsbaustein MB_InitTouchProbe initialisiert die Messtasterfunktionalität.

ML_TechMotionBase.library // MX_TechMotionBase.library



Der Funktionsbaustein "MB_InitTouchProbe (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-71: Referenztable MB_Init_TouchProbe

Schnittstellenbeschreibung

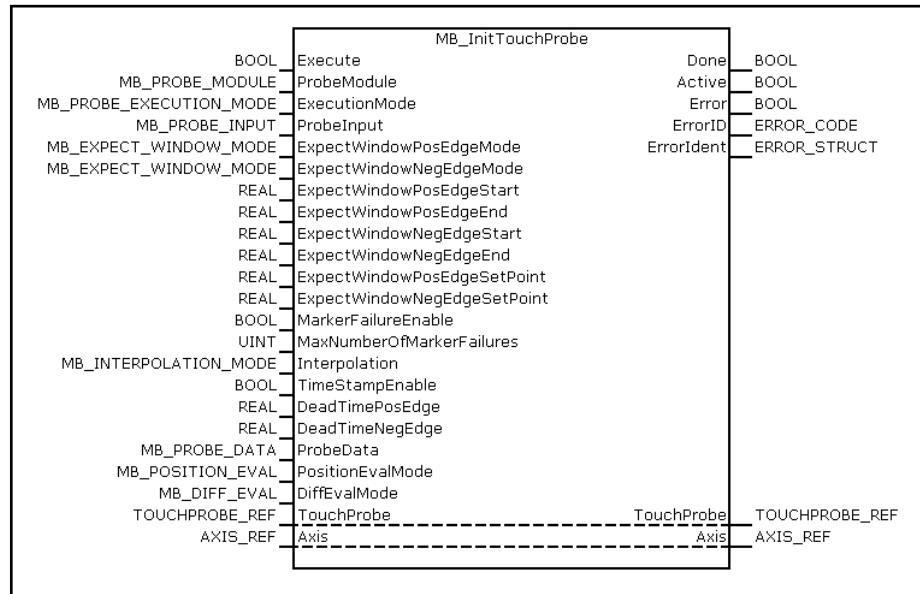


Abb. 3-66: Funktionsbaustein MB_InitTouchProbe

E/A-Typ	Name	Datentyp	Kommentar
VAR_INPUT	Execute	BOOL	Startet die Initialisierung der Messtasterfunktionalität bei steigender Flanke
	ProbeModule	MB_PROBE_MODULE	Position des Messtasters auf der MLC (onboard, Funktionsmodule), wird derzeit vom Antriebsmesstaster nicht ausgewertet
	ExecutionMode	MB_PROBE_EXECUTION_MODE	Auswahl, ob einzelne oder fortlaufende Messung MB_PROBE_EXECUTION_MODE, Seite 135
	ProbeInput	MB_PROBE_INPUT	Wählt den digitalen Eingang für den MLC-Messtaster aus, wird derzeit vom Antriebsmesstaster nicht ausgewertet
	ExpectWindowPosEdgeMode	MB_EXPECT_WINDOW_MODE	Erwartungsfenstermodus, wenn positive Flanke ausgewertet wird
	ExpectWindowNegEdgeMode	MB_EXPECT_WINDOW_MODE	Erwartungsfenstermodus, wenn negative Flanke ausgewertet wird
	ExpectWindowPosEdgeStart	REAL	Start des Erwartungsfensters, wenn positive Flanke ausgewertet wird
	ExpectWindowPosEdgeEnd	REAL	Ende des Erwartungsfensters, wenn positive Flanke ausgewertet wird

E/A-Typ	Name	Datentyp	Kommentar
	ExpectWindowNegEdgeStart	REAL	Start des Erwartungsfensters, wenn negative Flanke ausgewertet wird
	ExpectWindowNegEdgeEnd	REAL	Ende des Erwartungsfensters, wenn negative Flanke ausgewertet wird
	ExpectWindowPosEdgeSetPoint	REAL	Sollwert für das relative Erwartungsfenster, wenn positive Flanke ausgewertet wird
	ExpectWindowNegEdgeSetPoint	REAL	Sollwert für das relative Erwartungsfenster, wenn negative Flanke ausgewertet wird
	MarkerFailureEnable	BOOL	Markenausfall aktivieren
	MaxNumberOfMarkerFailure	UINT	Schwelle für die registrierte Anzahl der Markenausfälle
	Interpolation	MB_INTERPOLATION_MODE	Interpolation für den MLC-Messtaster, wird derzeit vom Antriebsmesstaster nicht ausgewertet
	TimeStampEnable	BOOL	Zeitstempel für den MLC-Messtaster aktivieren, wird derzeit vom Antriebsmesstaster nicht ausgewertet
	DeadTimePosEdge	REAL	Totzeitkompensation für die positive Flanke, wenn ausgewertet
	DeadTimeNegEdge	REAL	Totzeitkompensation für die negative Flanke, wenn ausgewertet
	ProbeData	MB_PROBE_DATA	Gemessenes Messtastersignal
	PositionEvalMode	MB_POSITION_EVAL	Wählt die zu auszuwertende Flanke aus
	DiffEvalMode	MB_DIFF_EVAL	Auswahl der Flanke zur Differenzmessung
VAR_OUTPUT	Done	BOOL	Initialisierung durchgeführt
	Active	BOOL	Der Funktionsbaustein ist momentan aktiv
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 132
VAR_IN_OUT CONSTANT	TouchProbe	TOUCHPROBE_REF	Referenz zur Signaltriggerquelle
	Axis	AXIS_REF	Referenz zur Achse

Tab. 3-72: Schnittstellenvariablen Funktionsbaustein MB_InitTouchProbe

Min-/Max- und Standard-Werte

In der folgenden Tabelle sind der Min-/Max- und der Defaultwert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
Execute	BOOL			FALSE	Kontinuierlich
ProbeModule	MB_PROBE_MODULE			ONBOARD_MODULE	Steigende Flanke an "Execute"
ExecutionMode	MB_PROBE_EXECUTION_MODE			CONTINUOUS	Steigende Flanke an "Execute"

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
ProbeInput	MB_PROBE_INPUT			D_INPUT_1	Steigende Flanke an "Execute"
ExpectWindow-PosEdgeMode	MB_EXPECT_WINDOW_MODE			DISABLED_EXP	Steigende Flanke an "Execute"
ExpectWindow-NegEdgeMode	MB_EXPECT_WINDOW_MODE			DISABLED_EXP	Steigende Flanke an "Execute"
ExpectWindow-PosEdgeStart	REAL	0.0	ModuloValue	0.0	Steigende Flanke an "Execute"
ExpectWindow-PosEdgeEnd	REAL	0.0	ModuloValue	0.0	Steigende Flanke an "Execute"
ExpectWindow-NegEdgeStart	REAL	0.0	ModuloValue	0.0	Steigende Flanke an "Execute"
ExpectWindow-NegEdgeEnd	REAL	0.0	ModuloValue	0.0	Steigende Flanke an "Execute"
ExpectWindow-PosEdgeSetPoint	REAL	0.0	ModuloValue	0.0	Steigende Flanke an "Execute"
ExpectWindow-NegEdgeSetPoint	REAL	0.0	ModuloValue	0.0	Steigende Flanke an "Execute"
MarkerFailureEnable	BOOL			FALSE	Steigende Flanke an "Execute"
MaxNumberOfMarkerFailure	UINT			0	Steigende Flanke an "Execute"
TimeStampEnable	BOOL			FALSE	Steigende Flanke an "Execute"
DeadTimePosEdge	REAL	0.0		0.0	Steigende Flanke an "Execute"
DeadTimeNegEdge	REAL	0.0		0.0	Steigende Flanke an "Execute"
ProbeData	MB_PROBE_DATA			TP_ACT_POS_FEEDBACK	Steigende Flanke an "Execute"
PositionEvalMode	MB_POSITION_EVAL			DISABLED_EVAL	Steigende Flanke an "Execute"
DiffEvalMode	MB_DIFF_EVAL			DISABLED_DIFF	Steigende Flanke an "Execute"

Tab. 3-73: Min-/Max- und Standard-Werte für die Eingänge des MB_InitTouchProbe

Funktionsbeschreibung des MB_InitTouchProbe



Der Funktionsbaustein MB_InitTouchProbe ist in der Sercos-Phase 4 auszuführen bevor die Funktionsbausteine [MB_TouchProbe](#), [Seite 135](#), und [MB_TouchProbeContinuous](#), [Seite 138](#), verwendet werden können.

TouchProbe, ProbeModule, ProbeInput

Der Eingang "TouchProbe" definiert verwendete Steuerung und Messtaster. Das Element "CntrlNo" muss auf "LOCAL_CNTRL" gesetzt sein. Das Element "TouchProbeNo" kann den Wert "TOUCHPROBE_DRIVE_1" oder "TOUCHPROBE_DRIVE_2" annehmen.

Die Eingänge "ProbeModule" und "ProbeInput" werden derzeit nicht ausgewertet.

ExecutionMode

Am Eingang "ExecutionMode", wird die Messtasterfunktion auf Einzelmessung oder fortlaufende Messung gesetzt. Wird die Einzelmessung verwendet, muss der Funktionsbaustein MB_TouchProbe verwendet werden. Für eine fortlaufende Messung ist der Funktionsbaustein MB_TouchProbeContinuous zu verwenden.

ExpectWindow, PositionEvalMode

ExpectWindow beschreibt das Erwartungsfenster des Messtasters. Momentan unterstützt das Erwartungsfenster des Antriebes kein separates Erwartungsfenster für die negativen und positiven Flanken gleichzeitig. Wird nur die positive Flanke ausgewertet ("PositionEvalMode"=POSITION_POS_EDGE), werden die Eingänge "ExpectWindowPosEdgeMode", "ExpectWindowPosEdgeStart", "ExpectWindowPosEdgeEnd" und "ExpectWindowPosEdgeSetPoint" ausgewertet.

Wird die negative Flanke ausgewertet ("PositionEvalMode"=POSITION_NEG_EDGE), werden "ExpectWindowNegEdgeMode", "ExpectWindowNegEdgeStart", "ExpectWindowNegEdgeEnd" und "ExpectWindowNegEdgeSetPoint" ausgewertet.

Werden beide Flanken ausgewertet ("PositionEvalMode"=POSITION_POS_NEG_EDGE), werden nur die Eingänge "ExpectWindowPosEdgeMode", "ExpectWindowPosEdgeStart", "ExpectWindowPosEdgeEnd", "ExpectWindowPosEdgeSetPoint" für das Erwartungsfenster des Antriebes ausgewertet. Das Erwartungsfenster gilt für beide ausgewerteten Flanken.

DeadTimePosEdge, DeadTimeNegEdge

Die Eingänge "DeadTimePosEdge" und "DeadTimeNegEdge" bestimmen die Totzeitkompensation der Flanken. Der Eingang "DeadTimePosEdge" ist gültig, wenn die positive Flanke ausgewertet wird. Der Eingang "DeadTimeNegEdge" ist gültig, wenn die negative Flanke ausgewertet wird.

ProbeData, DiffEvalMode

Der Eingang "ProbeData" definiert das gemessene Signal der Messtasterfunktion. Die folgenden Signale stehen zur Auswahl:

ENUM MB_PROBE_DATA (Auswahl)

Signal	Beschreibung
TP_ACT_POS_FEEDBACK1	Positionsistwert Geber 1
TP_ACT_POS_FEEDBACK2	Positionsistwert Geber 2
TP_ACT_POSIN_ACT_CYCLE	Positionsistwert im Istwertzyklus
TP_SYNC_CMD_POS	ELS, Synchroner Positionssollwert
TP_PROBE_TIME	Zeitstempel
TP_ACT_MEASURING_ENC_POS	Positionsistwert Messgeber
TP_RES_MASTER_AXIS_POS	ELS, Resultierende Leitachsposition
TP_EFF_MASTER_AXIS_POS	ELS, Wirksame Leitachsposition

Signal	Beschreibung
TP_CAM_SHAFT_PROF_ACCESS_ANGLE	ELS, Kurvenscheibe, Zugriffswinkel
TP_ELS_CMD_POS	ELS, Positionssollwert

Tab. 3-74: Mögliche Werte des Eingangs ProbeData

Der Eingang "DiffEvalMode" wählt die Differenzmessung der Messtasterfunktion aus. Momentan wird nur "STANDARD_MODE" unterstützt.

ModuloValue

Der Ausgang "ModuloValue" zeigt den Modulowert des gemessenen Signals an.

Interpolation

Der Eingang "Interpolation" gibt an, wie Positionswerte interpoliert werden sollen. Der Funktionsbaustein MB_InitTouchProbe unterstützt zur Zeit nur die Werte "NO_INTERPOLATION" und "LINEAR_INTERPOLATION". Er wird jedoch nicht weiter ausgewertet.

Zeitdiagramm Zeitdiagramm gemäß der Spezifikation PLCopen.

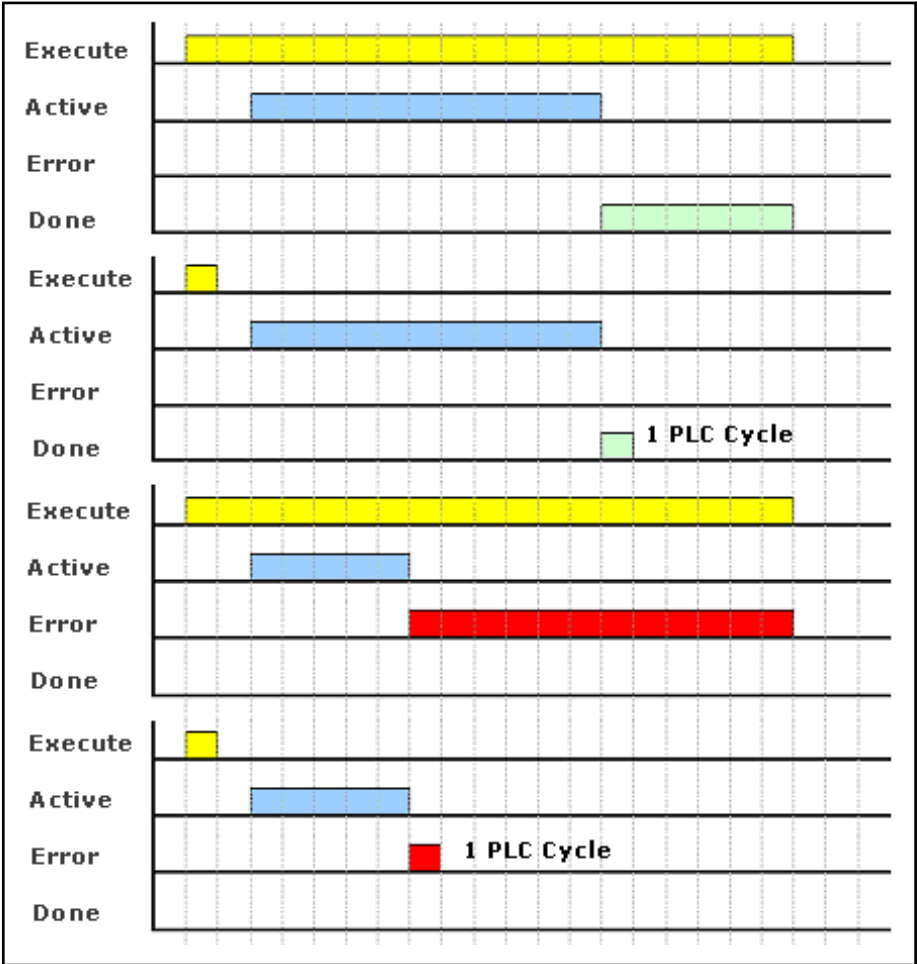


Abb. 3-67: Zeitdiagramm Funktionsbaustein MB_InitTouchProbe

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für "F_RELATED_TABLE" 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo von „Axis“ ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (Axis_Ref) wurde geändert während der FB in Betrieb ist
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0001	Der Eingang "ProbeModule" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0002	Der Eingang "ExecutionMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0003	Der Eingang "ProbeInput" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0004	Der Eingang "DeadTimePosEdge" ist ungültig (überprüfe, ob < 0)
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0005	Der Eingang "DeadTimeNegEdge" ist ungültig (überprüfe, ob < 0)
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0006	Der Eingang "ProbeData" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0007	Der Eingang "PositionEvalMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0008	Der Eingang "DiffEvalMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0009	Der Eingang "MaxNumberOfMarkerFailure" ist ungültig (überprüfe, ob < 0)
INPUT_INVALID_ERROR (16#0001)	16#0404	16#000A	Der Eingang "TouchProbe.TouchProbeNo" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0404	16#000B	Der Eingang "Interpolation" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0001	Der Eingang "ExpectWindowPosEdgeMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0002	Der Eingang "ExpectWindowNegEdgeMode" ist ungültig
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0003	Der Eingang "ExpectWindowPosEdgeStart" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0004	Der Eingang "ExpectWindowPosEdgeEnd" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0005	Der Eingang "ExpectWindowNegEdgeStart" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0006	Der Eingang "ExpectWindowNegEdgeEnd" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0007	Der Eingang "ExpectWindowPosEdgeSetPoint" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0008	Der Eingang "ExpectWindowNegEdgeSetPoint" ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0403	16#0001	Benötigtes Messtastersteuerbit (S-0-0405, Bit 0) ist nicht im Signalsteuerwort konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0002	Benötigtes Messtastersteuerbit (S-0-0406, Bit 0) ist nicht im Signalsteuerwort konfiguriert

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0403	16#0003	Benötigtes Messtasterstatusbit (S-0-0409, Bit 0) ist nicht im Signalstatuswort konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0004	Benötigtes Messtasterstatusbit (S-0-0410, Bit 0) ist nicht im Signalstatuswort konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0005	Benötigtes Messtasterstatusbit (S-0-0411, Bit 0) ist nicht im Signalstatuswort konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0006	Benötigtes Messtasterstatusbit (S-0-0412, Bit 0) ist nicht im Signalstatuswort konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0007	Benötigter Messtasterwert S-0-0130 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0008	Benötigter Messtasterwert S-0-0131 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#0009	Benötigter Messtasterwert S-0-0132 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#000A	Benötigter Messtasterwert S-0-0133 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#000B	Benötigter Messtasterparameter P-0-0224 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#000C	Benötigter Messtasterparameter P-0-0225 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#000D	Benötigter Messtasterparameter P-0-0202 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#000E	Benötigter Messtasterparameter P-0-0203 ist nicht zyklisch im AT konfiguriert
RESOURCE_ERROR (16#0003)	16#0403	16#00XX	P-0-1367, Bit 6 der MLD-M-Masterachse ist nicht auf TRUE gesetzt. Dies ist jedoch für ein Update der AxisData-Struktur notwendig

Tab. 3-75: Fehlercodes Funktionsbaustein MB_InitTouchProbe

Parametrierte Parameter

Der Funktionsbaustein MB_InitTouchProbe parametriert die folgenden Parameter:

- P-0-0226, Messtaster, erweitertes Steuerwort
- S-0-0169, Messtastersteuerparameter
- S-0-0170, Kommando Messtasterzyklus
- S-0-0426 Signalauswahl Messtaster1
- S-0-0427 Signalauswahl Messtaster2



Bei Verwendung der Messtasterfunktion im Antrieb müssen die zyklischen Kanäle vor Start manuell konfiguriert werden.

Hier müssen die Bits des Signal-Steuerwortes sowie die des Signal-Statuswortes erscheinen und die der ausgewählten Messtasterfunktion zugehörigen Messwerte S-0-0130, S-0-0131 bzw. P-0-0202.

3.9.4 MB_PROBE_EXECUTION_MODE

Der Aufzählungstyp gestattet Auswahl, ob eine einzelne oder eine fortlaufende Messung durchgeführt werden soll.

Name	Wert	Beschreibung
CONTINUOUS	0	Messtasterfunktion fortlaufende Messung
SINGLE_SHOT	1	Messtasterfunktion Einzelmessung

Tab. 3-76: ENUM MB_PROBE_EXECUTION_MODE

3.9.5 MB_TouchProbe

Kurzbeschreibung

Der Funktionsbaustein MB_TouchProbe wird zum Erfassen einer Achsposition, Leitachposition oder zur Zeiterfassung bei einem Triggerereignis verwendet. Dabei handelt es sich um einen Funktionsbaustein für die Einzelmessung.

Der Funktionsbaustein muss erneut ausgeführt werden, um ein weiteres Signal zu erfassen.

Das zu messende Signale wird am Eingang "TouchProbe" des Funktionsbausteins [MB_InitTouchProbe](#), Seite 127, definiert.



Die Verwendung dieses Funktionsbausteins ist nur möglich, wenn die Messtasterfunktion zuvor durch Verwendung des Funktionsbausteins [MB_InitTouchProbe](#), Seite 127, konfiguriert wurde.

Eine manuelle Konfiguration der Messtasterfunktion reicht nicht aus!



Der Funktionsbaustein "MB_TouchProbe (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-77: Referenztable MB_TouchProbe

ML_TechMotionBase.library // MX_TechMotionBase.library

Schnittstellenbeschreibung

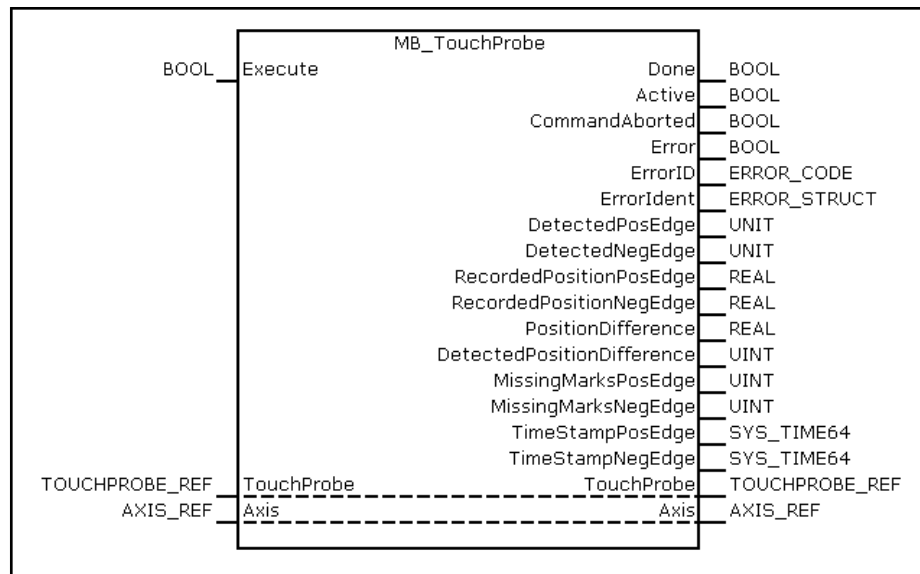


Abb. 3-68: Funktionsbaustein MB_TouchProbe

E/A-Typ	Name	Datentyp	Kommentar
VAR_INPUT	Execute	BOOL	Startet die Messtasteraufzeichnung bei steigender Flanke
VAR_OUTPUT	Done	BOOL	Triggerereignis aufgezeichnet
	Active	BOOL	Der Funktionsbaustein ist momentan aktiv
	CommandAborted	BOOL	Befehl durch MB_AbortTrigger abgebrochen
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 138
	DetectedPosEdge	UINT	Positive Flanke erkannt
	DetectedNegEdge	UINT	Negative Flanke erkannt
	RecordedPositionPosEdge	REAL	Position, an der die positive Flanke auftrat (in technischen Einheiten)
	RecordedPositionNegEdge	REAL	Position, an der die negative Flanke auftrat (in technischen Einheiten)
	PositionDifference	REAL	Zeigt die Differenz zwischen den Messtasterwerten in Abhängigkeit von "DiffEvalMode" an
	DetectedPositionDifference	UINT	Differenzmessung erkannt
	MissingMarksPosEdge	UINT	Zähler der fehlenden Marken, wenn die positive Flanke ausgewertet wird
	MissingMarksNegEdge	UINT	Zähler der fehlenden Marken, wenn die negative Flanke ausgewertet wird
	TimeStampPosEdge	SYS_TIME64	Zeitpunkt, wann die positive Flanke auftrat (nur MLC/XLC-Messtaster)

E/A-Typ	Name	Datentyp	Kommentar
	TimeStampNegEdge	SYS_TIME64	Zeitpunkt, wann die negative Flanke auftrat (nur MLC/XLC-Messtaster)
VAR_IN_OUT CONSTANT	TouchProbe	TOUCHPROBE_REF	Referenz zur Signaltriggerquelle
	Axis	AXIS_REF	Referenz zur Achse

Tab. 3-78: Schnittstellenvariablen Funktionsbaustein MB_TouchProbe

Standardwert In der folgenden Tabelle ist der Standardwert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
Enable	BOOL	n/a	n/a	FALSE	Kontinuierlich

Tab. 3-79: Standardwert für den Eingang des MB_TouchProbe

Funktionsbeschreibung des MB_TouchProbe

Der Funktionsbaustein MB_TouchProbe führt eine Einzelmessungen aus. Der Funktionsbaustein wird bei steigender Flanke am Eingang "Execute" aktiviert.

DetectedPosEdge, DetectedNegEdge

Ist die Messtasterfunktion eingestellt um eine positive/negative Flanke des Eingangssignals auszuwerten, erhöht sich der Ausgang "DetectedPosEdge"/"DetectedNegEdge", wenn eine positiven/negativen Flanke erkannt wird.

RecordedPositionPosEdge, RecordedPositionNegEdge

Die Ausgänge "RecordedPositionPosEdge" bzw. "RecordedPositionNegEdge" enthalten die Position, an der die positive bzw. negative Flanke auftrat.

PositionDifference, DetectedPositionDifference

Der Ausgang "PositionDifference" enthält im Falle einer aktivierten Differenzmessung die Differenz der beiden zuletzt gemessenen Werte des gleichen Messtasters. Bei jeder Messung wird der Ausgang "DetectedPositionDifference" erhöht.

Die folgende Berechnung gilt, wenn die Antriebsmesstasterfunktion für den Ausgang "PositionDifference" aktiviert wird:

$$\text{PositionDifference} = |(\text{S-0-0130}) - (\text{S-0-0131})|$$

$$\text{PositionDifference} = |(\text{S-0-0132}) - (\text{S-0-0133})|$$

Abb. 3-69: Berechnung der Positionsdivergenz für Achsen

MissingMarksPosEdge, MissingMarksNegEdge

Die Ausgänge "MissingMarksPosEdge" bzw. "MissingMarksNegEdge" werden nur ausgewertet, wenn das Erwartungsfenster aktiv ist und der Markenausfall aktiviert wurde. Sie zählen die Marken außerhalb des Erwartungsfensters. Die Ausgänge werden jedesmal auf 0 gesetzt, wenn eine Marke innerhalb des Erwartungsfensters erkannt wird. Ist der Messtaster eingestellt um beide Flanken auszuwerten, ist nur der Ausgang "MissingMarksPosEdge" gültig. Momentan unterstützt die Antriebsmesstasterfunktion nicht gleichzeitig individuelle Markenausfälle für jede Flanke.

Zeitdiagramm

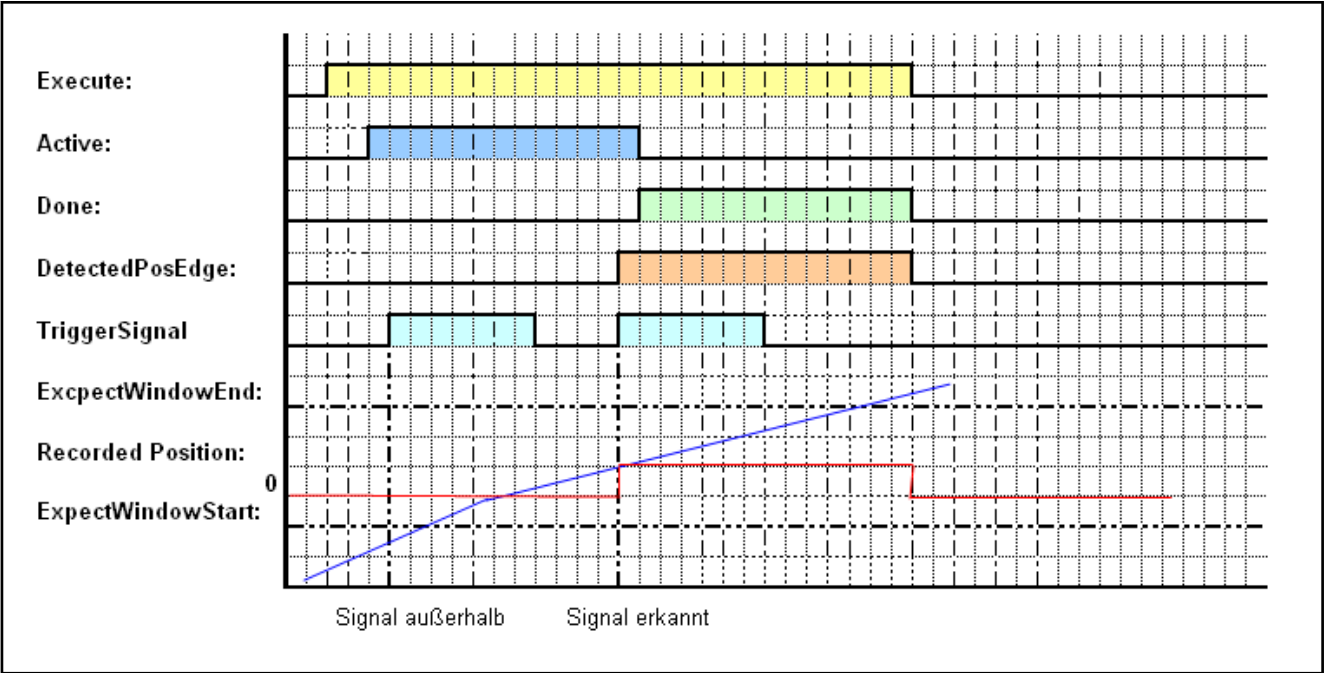


Abb. 3-70: Zeitdiagramm Funktionsbaustein MB_TouchProbe

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für "F_RELATED_TABLE" 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo von "Axis" ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (Axis_Ref) wurde geändert während der FB in Betrieb ist
ACCESS_ERROR (16#0003)	16#0406	16#0001	Leeres Handle für ATPosEdgeVal
ACCESS_ERROR (16#0003)	16#0406	16#0002	Leeres Handle für ATNegEdgeVal
ACCESS_ERROR (16#0003)	16#0406	16#0003	Leeres Handle für ATMarkerFailure
ACCESS_ERROR (16#0003)	16#0406	16#0004	Leeres Handle für ATDiffProbeVal
ACCESS_ERROR (16#0003)	16#0406	16#0005	Leeres Handle für ATDeadTimeComp
ACCESS_ERROR (16#0003)	16#0406	16#0006	Leeres Handle für SCWEnableProbe
ACCESS_ERROR (16#0003)	16#0406	16#0007	Leeres Handle für SSWPosLatched
ACCESS_ERROR (16#0003)	16#0406	16#0008	Leeres Handle für SSWNegLatched

Tab. 3-80: Fehlercodes Funktionsbaustein MB_TouchProbe

3.9.6 MB_TouchProbeContinuous

Kurzbeschreibung Der Funktionsbaustein MB_TouchProbeContinuous wird zum Erfassen einer Achsposition, Leitachposition oder zur Zeiterfassung bei einem Triggerereignis verwendet. Er führt eine fortlaufende Messung aus, solange der Eingang

"Enable" TRUE ist. Das zu erfassende Signale, wird am Eingang "TouchProbe" des Funktionsbausteins [MB_InitTouchProbe](#), [Seite 127](#), definiert.



Die Verwendung dieses Funktionsbausteins ist nur möglich, wenn die Messtasterfunktion zuvor durch Verwendung des Funktionsbausteins [MB_InitTouchProbe](#), [Seite 127](#), konfiguriert wurde.

Eine manuelle Konfiguration der Messtasterfunktion reicht nicht aus!



Der Funktionsbaustein "MB_TouchProbeContinuous (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-81: Referenztabelle MB_TouchProbeContinuous

Schnittstellenbeschreibung

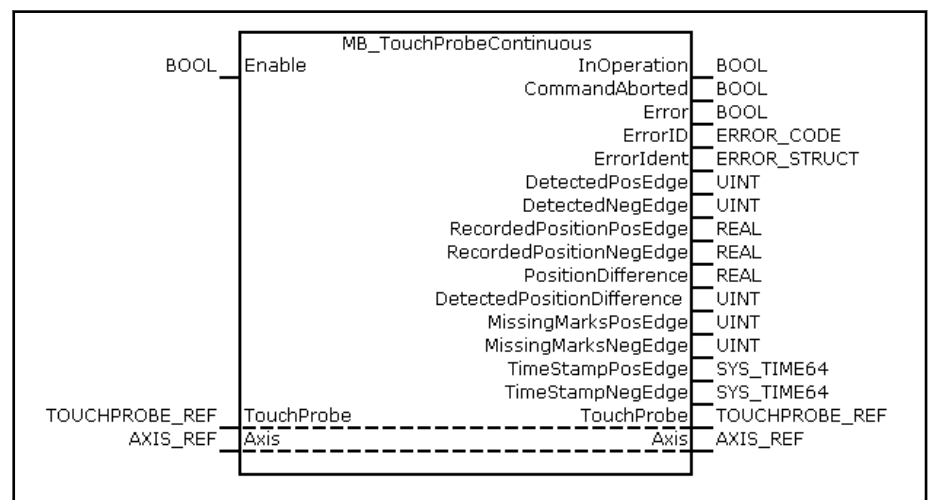


Abb. 3-71: Funktionsbaustein MB_TouchProbeContinuous

E/A-Typ	Name	Datentyp	Kommentar
VAR_INPUT	Enable	BOOL	Startet die Messtasteraufzeichnung bei TRUE
VAR_OUTPUT	InOperation	BOOL	Messtaster in Betrieb
	CommandAborted	BOOL	Befehl durch MB_AbortTrigger abgebrochen
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung
	DetectedPosEdge	UINT	Positive Flanke erkannt
	DetectedNegEdge	UINT	Negative Flanke erkannt
	RecordedPositionPosEdge	REAL	Position, an der die positive Flanke auftrat (in technischen Einheiten)

ML_TechMotionBase.library // MX_TechMotionBase.library

E/A-Typ	Name	Datentyp	Kommentar
	RecordedPositionNegEdge	REAL	Position, an der die negative Flanke auftrat (in technischen Einheiten)
	PositionDifference	REAL	Zeigt die absolute Differenz zwischen den Messtasterwerten an
	DetectedPositionDifference	UINT	Differenzmessung erkannt
	MissingMarksPosEdge	UINT	Zähler der fehlenden Marken für die positive Flanke, wenn ausgewertet
	MissingMarksNegEdge	UINT	Zähler der fehlenden Marken für die negative Flanke, wenn ausgewertet
	TimeStampPosEdge	SYS_TIME64	Zeitpunkt, wann die positive Flanke auftrat (nur MLC/XLC-Messtaster)
	TimeStampNegEdge	SYS_TIME64	Zeitpunkt, wann die negative Flanke auftrat (nur MLC/XLC-Messtaster)
VAR_IN_OUT CONSTANT	TouchProbe	TOUCHPROBE_REF	Referenz zur Signaltriggerquelle
	Axis	AXIS_REF	Referenz zur Achse

Tab. 3-82: Schnittstellenvariablen Funktionsbaustein MB_TouchProbeContinuous

Standardwert In der folgenden Tabelle ist der Standardwert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
Enable	BOOL			FALSE	Kontinuierlich

Tab. 3-83: Standardwert für den Eingang des MB_InitTouchProbe

Funktionsbeschreibung

Der Funktionsbaustein MB_TouchProbeContinuous wird ausschließlich für die fortlaufende Messung verwendet. Der Funktionsbaustein ist aktiviert, solange der Eingang "Enable" TRUE ist.

DetectedPosEdge, DetectedNegEdge

Ist die Messtasterfunktion eingestellt um eine positive/negative Flanke des Eingangssignals auszuwerten, erhöht sich der Ausgang "DetectedPosEdge"/"DetectedNegEdge", wenn eine positiven/negativen Flanke erkannt wird.

RecordedPositionPosEdge, RecordedPositionNegEdge

Die Ausgänge "RecordedPositionPosEdge" bzw. "RecordedPositionNegEdge" enthalten die Position, an der die positive bzw. negative Flanke auftrat.

PositionDifference, DetectedPositionDifference

Der Ausgang "PositionDifference" enthält im Falle einer aktivierten Differenzmessung die Differenz der beiden zuletzt gemessenen Werte des gleichen Messtasters. Bei jeder Messung wird der Ausgang "DetectedPositionDifference" erhöht.

Die folgende Berechnung gilt, wenn die Antriebsmesstasterfunktion für den Ausgang "PositionDifference" aktiviert wird:

$$\text{PositionDifference} = |(S-0-0130) - (S-0-0131)|$$

$$\text{PositionDifference} = |(S-0-0132) - (S-0-0133)|$$

Abb. 3-72: Berechnung der Positionsdifferenz für Achsen

MissingMarksPosEdge, MissingMarksNegEdge

Die Ausgänge "MissingMarksPosEdge" bzw. "MissingMarksNegEdge" werden nur ausgewertet, wenn das Erwartungsfenster aktiv ist und der Markenausfall aktiviert wurde. Sie zählen die Marken außerhalb des Erwartungsfensters. Die Ausgänge werden jedesmal auf 0 gesetzt, wenn eine Marke innerhalb des Erwartungsfensters erkannt wird. Ist der Messtaster eingestellt um beide Flanken auszuwerten, ist nur der Ausgang "MissingMarksPosEdge" gültig. Momentan unterstützt die Antriebsmesstasterfunktion nicht gleichzeitig individuelle Markenausfälle für jede Flanke.

Zeitdiagramm

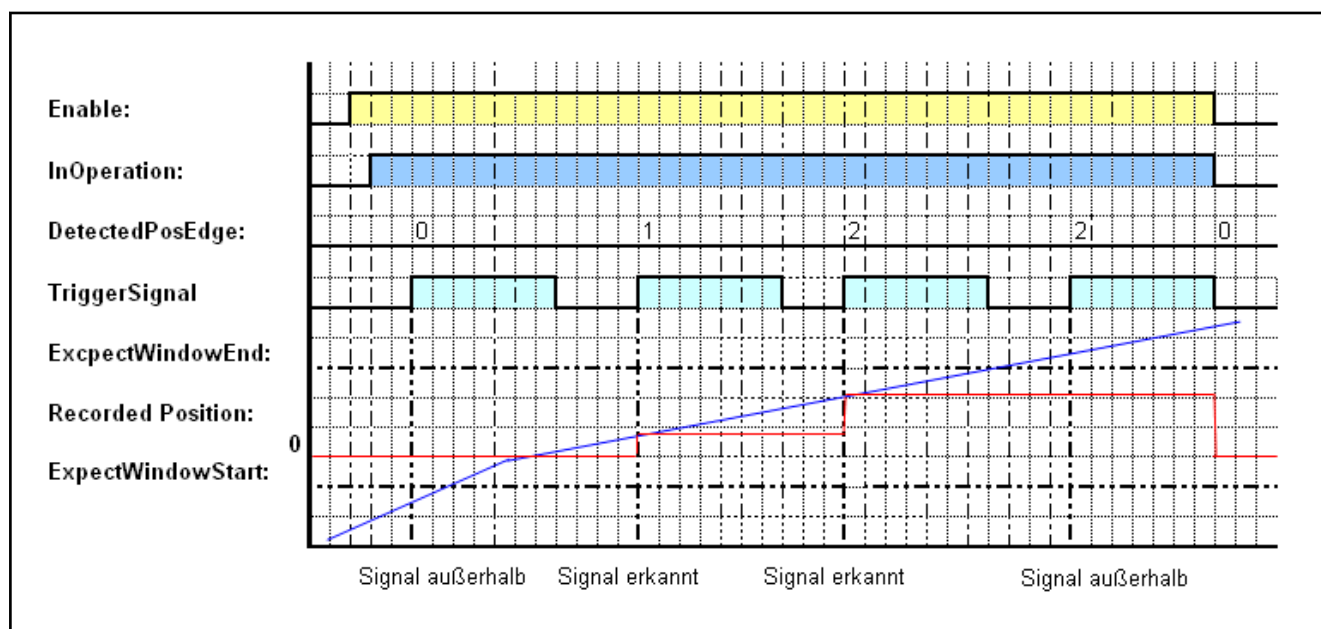


Abb. 3-73: Zeitdiagramm Funktionsbaustein MB_TouchProbeContinuous

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für "F_RELATED_TABLE" 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo von „Axis“ ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (Axis_Ref) wurde geändert während der FB in Betrieb ist
ACCESS_ERROR (16#0003)	16#0406	16#0001	Leeres Handle für ATPosEdgeVal
ACCESS_ERROR (16#0003)	16#0406	16#0002	Leeres Handle für ATNegEdgeVal
ACCESS_ERROR (16#0003)	16#0406	16#0003	Leeres Handle für ATMarkerFailure

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0003)	16#0406	16#0004	Leeres Handle für ATDiffProbeVal
ACCESS_ERROR (16#0003)	16#0406	16#0005	Leeres Handle für ATDeadTimeComp
ACCESS_ERROR (16#0003)	16#0406	16#0006	Leeres Handle für SCWEnableProbe
ACCESS_ERROR (16#0003)	16#0406	16#0007	Leeres Handle für SSWPosLatched
ACCESS_ERROR (16#0003)	16#0406	16#0008	Leeres Handle für SSWNegLatched

Tab. 3-84: Fehlercodes Funktionsbaustein MB_TouchProbeContinuous

3.9.7 MB_WriteExpectWindow

Kurzbeschreibung Der Funktionsbaustein MB_WriteExpectWindow konfiguriert das Erwartungs-
fenster für die Messtasterfunktionalität.



Der Funktionsbaustein "MB_WriteExpectWindow (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-85: Referenztable MB_WriteExpectWindow

Schnittstellenbeschreibung

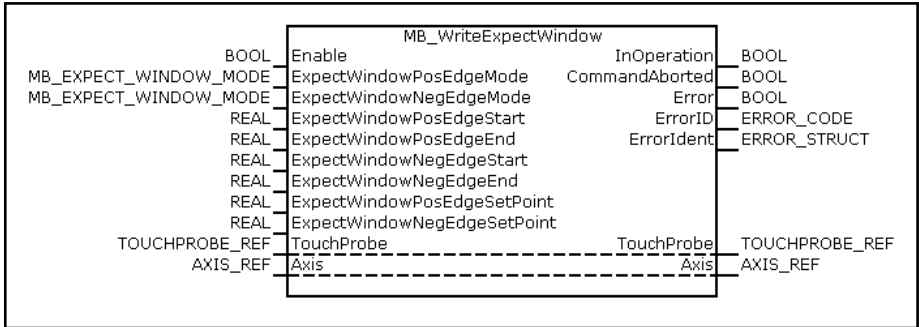


Abb. 3-74: Funktionsbaustein MB_WriteExpectWindow

E/A-Typ	Name	Datentyp	Kommentar
VAR_INPUT	Enable	BOOL	Aktiviert des Funktionsbaustein
	ExpectWindowPosEdgeMode	MB_EXPECT_WINDOW_MODE	Erwartungsfenstermodus, wenn positive Flanke ausgewertet wird
	ExpectWindowNegEdgeMode	MB_EXPECT_WINDOW_MODE	Erwartungsfenstermodus, wenn negative Flanke ausgewertet wird
	ExpectWindowPosEdgeStart	REAL	Start des Erwartungsfensters, wenn positive Flanke ausgewertet wird
	ExpectWindowPosEdgeEnd	REAL	Ende des Erwartungsfensters, wenn positive Flanke ausgewertet wird
	ExpectWindowNegEdgeStart	REAL	Start des Erwartungsfensters, wenn negative Flanke ausgewertet wird

E/A-Typ	Name	Datentyp	Kommentar
	ExpectWindowNegEdgeEnd	REAL	Ende des Erwartungsfensters, wenn negative Flanke ausgewertet wird
	ExpectWindowPosEdgeSetPoint	REAL	Sollwert für das relative Erwartungsfenster, wenn positive Flanke ausgewertet wird
	ExpectWindowNegEdgeSetPoint	REAL	Sollwert für das relative Erwartungsfenster, wenn negative Flanke ausgewertet wird
VAR_OUTPUT	InOperation	BOOL	FB in Betrieb
	CommandAborted	BOOL	FB abgebrochen
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 144
VAR_IN_OUT CONSTANT	TouchProbe	TOUCHPROBE_REF	Referenz zur Signaltriggerquelle
	Axis	AXIS_REF	Referenz zur Achse

Tab. 3-86: Schnittstellenvariablen Funktionsbaustein MB_WriteExpectWindow

Min-/Max- und Standard-Werte

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
Enable	BOOL			FALSE	Kontinuierlich
ExpectWindow-PosEdgeMode	MB_EXPECT_WINDOW_MODE			DISABLED_EXP	Ansteigende Flanke bei „Enable“
ExpectWindow-NegEdgeMode	MB_EXPECT_WINDOW_MODE			DISABLED_EXP	Ansteigende Flanke bei „Enable“
ExpectWindow-PosEdgeStart	REAL	0.0	ModuloValue	0.0	Kontinuierlich
ExpectWindow-PosEdgeEnd	REAL	0.0	ModuloValue	0.0	Kontinuierlich
ExpectWindow-NegEdgeStart	REAL	0.0	ModuloValue	0.0	Kontinuierlich
ExpectWindow-NegEdgeEnd	REAL	0.0	ModuloValue	0.0	Kontinuierlich
ExpectWindow-PosEdgeSetPoint	REAL	0.0	ModuloValue	0.0	Kontinuierlich
ExpectWindow-NegEdgeSetPoint	REAL	0.0	ModuloValue	0.0	Kontinuierlich

Tab. 3-87: Min-/Max- und Standard-Werte für die Eingänge des Funktionsbausteins MB_WriteExpectWindow

Funktionsbeschreibung des Antriebsmesstasters

Momentan unterstützt das Erwartungsfenster des Antriebes kein separates Erwartungsfenster für die negativen und positiven Flanken gleichzeitig. Wird nur die positive Flanke ausgewertet, werden die Eingänge "ExpectWindow-

PosEdgeMode", "ExpectWindowPosEdgeStart", "ExpectWindowPosEdgeStart" und "ExpectWindowPosEdgeSetPoint" ausgewertet.

Wird die negative Flanke ausgewertet, werden "ExpectWindowNegEdgeMode", "ExpectWindowNegEdgeStart", "ExpectWindowNegEdgeStart" und "ExpectWindowNegEdgeSetPoint" ausgewertet.

Werden beide Flanken ausgewertet, werden nur die Eingänge "ExpectWindowPosEdgeMode", "ExpectWindowPosEdgeStart", "ExpectWindowPosEdgeEnd", "ExpectWindowPosEdgeSetPoint" für das Erwartungsfenster des Antriebes ausgewertet. Das Erwartungsfenster gilt für beide ausgewerteten Flanken.

Zeitdiagramm

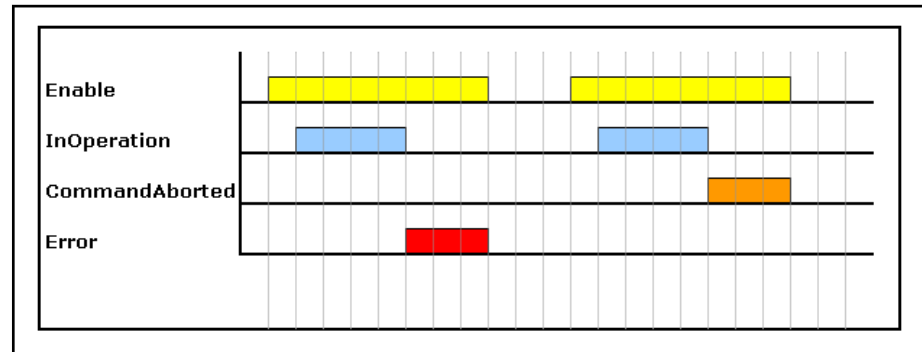


Abb. 3-75: Zeitdiagramm Funktionsbaustein MB_WriteExpectWindow

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für "F_RELATED_TABLE" 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo von „Axis“ ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (Axis_Ref) wurde geändert während der FB in Betrieb ist
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0001	Der Eingang "ExpectWindowPosEdgeMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0002	Der Eingang "ExpectWindowNegEdgeMode" ist ungültig
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0003	Der Eingang "ExpectWindowPosEdgeStart" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0004	Der Eingang "ExpectWindowPosEdgeEnd" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0005	Der Eingang "ExpectWindowNegEdgeStart" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0006	Der Eingang "ExpectWindowNegEdgeEnd" ist außerhalb des gültigen Bereichs
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0007	Der Eingang "ExpectWindowPosEdgeSetPoint" ist außerhalb des gültigen Bereichs

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0008	Der Eingang "ExpectWindowNegEdgeSetPoint" ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0403	16#000F	P-0-1367, Bit 6 der MLD-M-Masterachse ist nicht auf TRUE gesetzt. Dies ist jedoch für ein Update der AxisData-Struktur notwendig.

Tab. 3-88: Fehlercodes Funktionsbaustein MB_WriteExpectWindow

3.9.8 MB_AbortTrigger

Kurzbeschreibung Der Funktionsbaustein MB_AbortTrigger wird zum Abbrechen des Funktionsbausteins [MB_TouchProbe](#), Seite 135, verwendet.



Der Funktionsbaustein "MB_AbortTrigger (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-89: Referenztablette MB_AbortTrigger

Schnittstellenbeschreibung

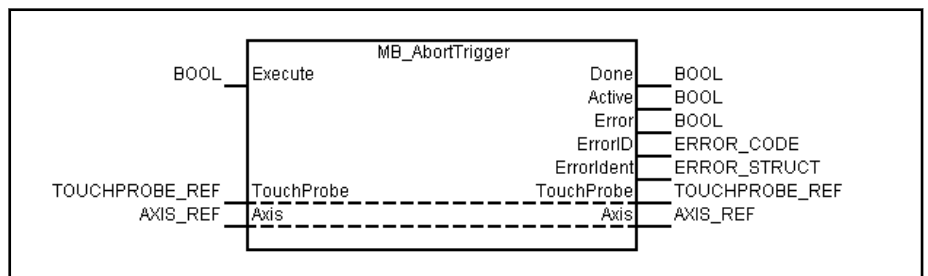


Abb. 3-76: Funktionsbaustein MB_AbortTrigger

E/A-Typen	Name	Datentyp	Kommentar
VAR_INPUT	Execute	BOOL	Startet den Funktionsbaustein bei steigender Flanke
VAR_OUTPUT	Done	BOOL	Messtaster abgebrochen
	Active	BOOL	Der Funktionsbaustein ist momentan aktiv
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung , Seite 146,
VAR_IN_OUT CONSTANT	TouchProbe	TOUCHPROBE_REF	Referenz zur Signaltriggerquelle
	Axis	AXIS_REF	Referenz zur Achse

Tab. 3-90: Schnittstellenvariablen Funktionsbaustein MB_AbortTrigger

Standardwert In der folgenden Tabelle ist der Standardwert des Bausteineingangs aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standard-Wert	Wirksam
Execute	BOOL			FALSE	Kontinuierlich

Tab. 3-91: Standardwert für den Eingang des Funktionsbausteins MB_AbortTrigger

Funktionsbeschreibung Der Funktionsbaustein MB_AbortTrigger deaktiviert die Messtasterfunktion. Er kann nur in Verbindung mit dem Baustein MB_TouchProbe verwendet werden. Um die fortlaufende Messung zu deaktivieren, muss der Eingang "Enable" des Funktionsbausteins MB_TouchProbeContinuous auf FALSE gesetzt werden.

Zeitdiagramm Zeitdiagramm gemäß der PLCopen-Spezifikation.

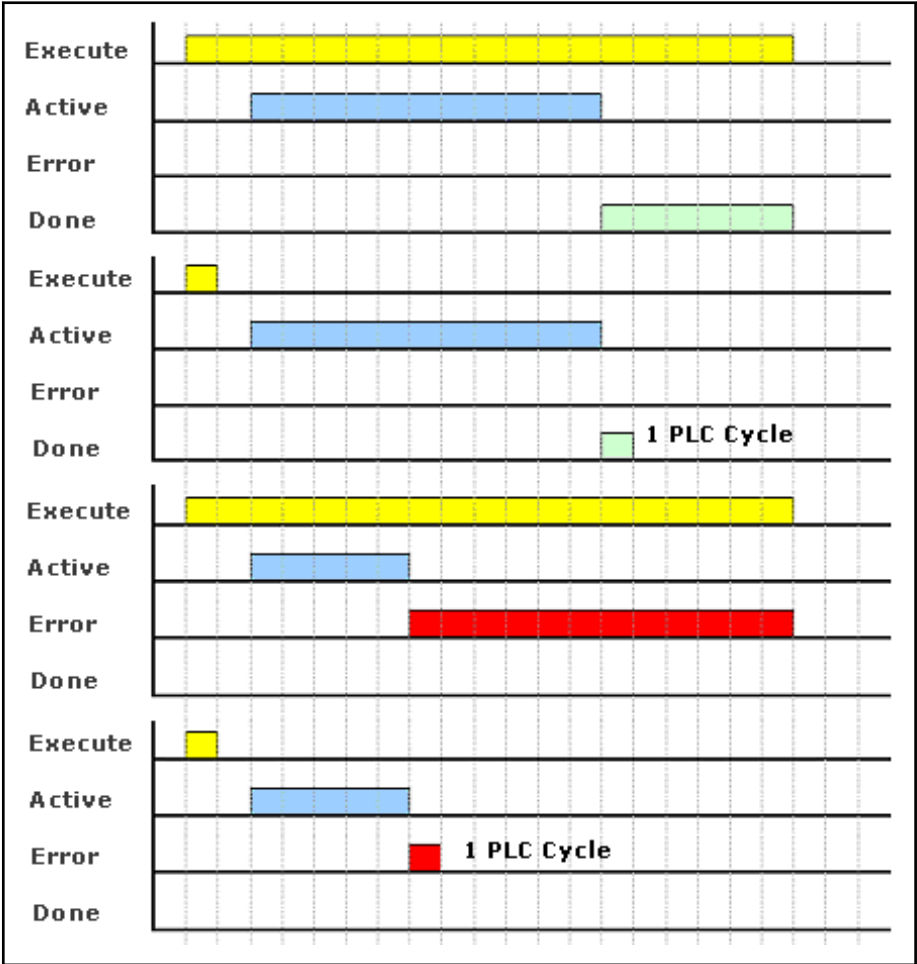


Abb. 3-77: Zeitdiagramm Funktionsbaustein MB_AbortTrigger

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für "F_RELATED_TABLE" 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo von „Axis“ ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (Axis_Ref) wurde geändert während der FB in Betrieb ist
ACCESS_ERROR (16#0003)	16#0406	16#0006	Leeres Handle für SCWEnableProbe

Tab. 3-92: Fehlercodes Funktionsbaustein MB_AbortTrigger

3.10 Zyklischer Datenkanäle, Administration

3.10.1 Einführung und Übersicht

Das Sercos-Interface erlaubt das Auflegen von frei konfigurierbaren zyklischen Daten. Diese werden zyklisch bei jedem Sercos-Takt sowohl von der Steuerung in die Antriebe als auch von den Antrieben zur Steuerung übertragen. Pro Achse stehen jeweils vier Parameter in Lese- bzw. Schreibrichtung zur Verfügung. Zusätzlich können jeweils vier Bits gelesen oder geschrieben werden. Die Konfiguration der zyklischen Daten erfolgt im Parametriermodus (PM, Sercos-Phase 2) und ist im Betriebsmodus (BB, Sercos-Phase 4) nicht änderbar.

Im Parametriermodus (PM, Sercos-Phase 2) kann durch den Funktionsbaustein MB_AllocCyclicDataParameter ein zyklischer Datenkanal für einen zu übertragenden Parameter angefordert werden. Ist der Parameter bereits im zyklischen Datenkanal aufgelegt, so erfolgt keine Änderung der Konfiguration. Andernfalls wird ein freier Datenkanal gesucht und der entsprechende Parameter aufgelegt.

Bei Änderung der Konfiguration kann ein einmal zyklisch aufgelegter Parameter mit [MB_FreeCyclicParameter, Seite 150](#), auch wieder freigegeben werden. Dadurch steht dieser Kanal wieder für eine neue Konfiguration zur Verfügung. Eine explizite Freigabe ist nur notwendig, wenn die Datenkanäle zur Laufzeit aus der SPS heraus neu konfiguriert werden.

Die Konfiguration kann auch in IndraWorks über den AxisData- bzw. den zyklischen Ist- und Sollwerte-Dialog erfolgen. Eine über IndraWorks festgelegte Konfiguration wird automatisch verwendet. Werden einige Parameter über IndraWorks konfiguriert und später zur Laufzeit weitere Parameter mit [MB_AllocCyclicParameter, Seite 148](#), angefordert, werden die neuen Parameter auf freie Kanäle verteilt. Mit [MB_GetCyclicChannelConfig, Seite 152](#), kann die aktuelle Konfiguration der zyklisch aufgelegten Parameter ausgelesen werden.

Im Betriebsmodus (BB) sucht der Funktionsbaustein [MB_GetCyclicParameterHandle, Seite 154](#), einen zyklisch aufgelegten Parameter in der Konfiguration einer Achse und schreibt alle benötigten Informationen in ein Handle. Mit diesem Handle kann dann in jedem Sercos-Zyklus durch die [MB_ReadCyclicParameter, Seite 155](#), und [MB_WriteCyclicParameter, Seite 157](#), vom Parameter gelesen oder auf den Parameter geschrieben werden. Für nachkommastellenbehaftete zyklische Parameter stehen die Funktionen [MB_ReadCyclicRealParameter, Seite 156](#), und [MB_WriteCyclicRealParameter, Seite 157](#), zur Verfügung. Aufgetretene Fehler während des Zugriffs auf die zyklischen Datenkanäle können über die Funktion [MB_GetLastCyclicParameterError, Seite 158](#), abgefragt werden.

Die zyklisch aufgelegten Parameter stehen parallel auch in der AxisData-Struktur der jeweiligen Achse zur Verfügung. Die Istwerte stehen in den UserActualDataX-Einträgen, die Sollwerte werden über UserCmdDataX beschrieben. Hierbei muss jedoch je nach aufgelegtem Parameter der richtige Datentyp beschrieben werden. Dieser Aufwand entfällt bei Verwendung der Funktionsbausteine MB_ReadCyclicParameter und MB_WriteCyclicParameter.

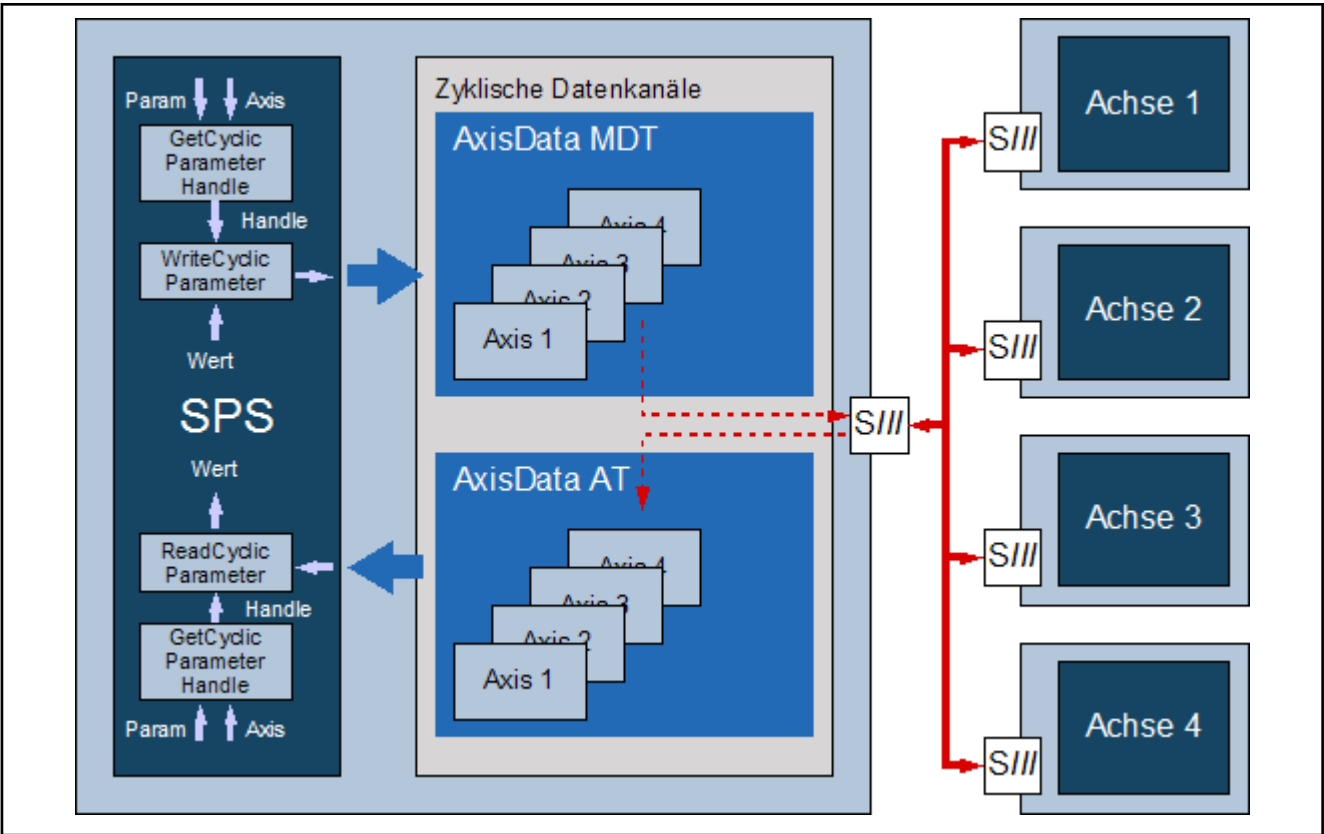


Abb. 3-78: Zugriff auf zyklische Daten mit Hilfe der CyclicParameter Funktionsbausteine und Funktionen

3.10.2 MB_AllocCyclicParameter

Kurzbeschreibung Dieser Funktionsbaustein überprüft die Konfiguration eines zyklisch zu lesen-
den oder zu beschreibenden Parameters im zyklischen Datenkanal einer
Achse und konfiguriert bei Bedarf den zyklischen Datenkanal.
Belegte Parameter werden in einer globalen Datenstruktur vermerkt.

 Der Funktionsbaustein "MB_AllocCyclicParameter (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-93: Referenztable MB_AllocCyclicParameter

Schnittstellenbeschreibung

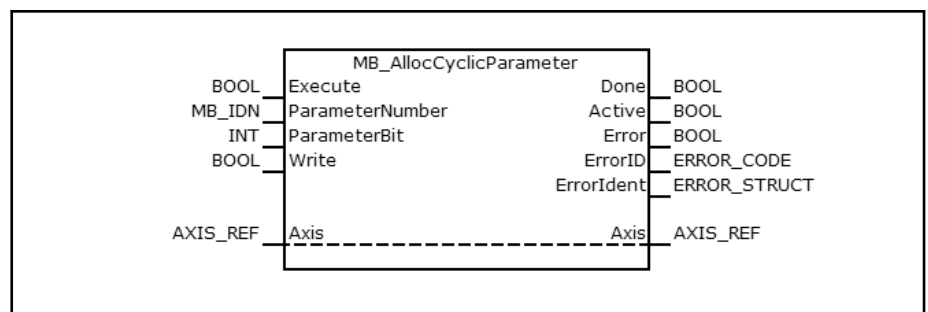


Abb. 3-79: Funktionsbaustein MB_AllocCyclicParameter

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	Startet die Konfiguration des Parameter
	ParameterNumber	MB_IDN	Aufzulegender Parameter
	ParameterBit	INT	Bit innerhalb des Parameters; "CYCLIC_ALLBITS" für gesamten Parameter
	Write	BOOL	FALSE: Parameter lesen TRUE: Parameter schreiben
VAR_OUTPUT	Done	BOOL	Parameter wurde aufgelegt
	Active	BOOL	Der Funktionsbaustein ist aktiv
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Grundsätzliche Fehlerursache
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe Fehlercodes, Seite 150
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Achse des Parameters

Tab. 3-94: Schnittstellenvariablen Funktionsbaustein MB_AllocCyclicParameter

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
ParameterNumber	DINT	0	n.def.	0	steigende Flanke an "Execute"
ParameterBit	INT	-1	31	CYCLIC_ALLBITS	steigende Flanke an "Execute"
Write	BOOL			FALSE	steigende Flanke an "Execute"

Tab. 3-95: Eingangsverhalten MB_AllocCyclicParameter

Funktionsbeschreibung

Dieser Funktionsbaustein arbeitet grundsätzlich nur in Sercos-Phase 2. Bei steigender Flanke an "Execute" werden die Eingänge "ParameterNumber", "ParameterBit" und "Write" übernommen. Anhand des Eingangs "Write" wird zwischen den Istdaten (UserActualData) und den Solldaten (UserCmdData) unterschieden. "ParameterBit" gibt die Bitnummer innerhalb des Parameters an. Liegt "ParameterBit" zwischen 0 und 31, wird ein Bit-Datenkanal verwendet. Ist "ParameterBit" gleich "CYCLIC_ALLBITS", wird der gesamte Parameter aufgelegt. "ParameterNumber" gibt den aufzulegenden Parameter an.

MB_AllocCyclicParameter überprüft im ersten Schritt, ob der Parameter sich eventuell schon in der aktuellen Konfiguration befindet. Ist dies nicht der Fall,

ML_TechMotionBase.library // MX_TechMotionBase.library

wird ein freier zyklischer Kanal gesucht und der Parameter dort aufgelegt. Anschließend wird die Belegung in einer internen statischen Variable gespeichert. Wird in "ParameterBit" eine Bitnummer angegeben, wird vorzugsweise ein freier Bitkanal verwendet. Sind alle Bitkanäle belegt, wird ein Parameterkanal verwendet. Nach erfolgreichem Abschluss der Konfiguration des zyklischen Datenkanals wird der "Done"-Ausgang für mindestens einen Zyklus auf TRUE gesetzt. "Done" bleibt TRUE, solange der Eingang „Execute“ aktiv ist. Steht kein freier zyklischer Kanal zur Verfügung, wird eine entsprechende Fehlermeldung ausgegeben.

Fehlerverhalten Kann der Parameter nicht im zyklischen Datenkanal aufgelegt werden oder tritt ein anderer Fehler auf, ist "Error" TRUE.

Fehlercodes Der Funktionsblock generiert die folgenden Fehlermeldungen der "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#000D	AxisNo	AxisRef liegt außerhalb des gültigen Bereichs
ACCESS_ERROR	16#0010	16#0000	Sercos-Phase 2 ist nicht aktiv
ACCESS_ERROR	16#1504	16#0000	AxisData-Struktur nicht aktiviert
INPUT_INVALID_ERROR	16#F212	16#0000	Parametertyp Struktur nicht unterstützt
INPUT_INVALID_ERROR	16#F213	16#0000	Parametertyp Funktion nicht unterstützt
INPUT_INVALID_ERROR	16#F214	16#0000	Parametertyp nicht unterstützt
INPUT_INVALID_ERROR	16#F215	16#0000	Parameter ist nicht beschreibbar
RESOURCE_ERROR	16#F216	16#0000	Kein freier zyklischer Kanal
INPUT_INVALID_ERROR	16#F217	16#0000	Parameter nicht im zyklischen Datenkanal konfigurierbar

Tab. 3-96: Fehlercodes Funktionsbaustein MB_AllocCyclicParameter

3.10.3 MB_FreeCyclicParameter

Kurzbeschreibung Dieser Funktionsbaustein gibt einen bereits konfigurierten zyklisch zu lesen oder zu beschreibenden Parameter im zyklischen Datenkanal einer Achse wieder frei. Die Freigabe des Parameters wird in einer globalen Struktur vermerkt.



Der Funktionsbaustein "MB_FreeCyclicParameter (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-97: Referenztable MB_FreeCyclicParameter

Schnittstellenbeschreibung

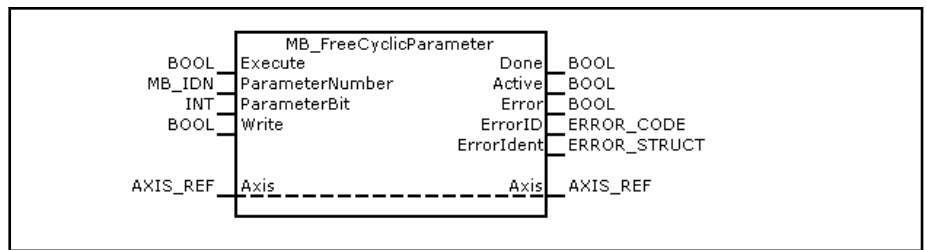


Abb. 3-80: Funktionsbaustein MB_FreeCyclicParameter

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Startet die Freigabe des Parameters
	ParameterNumber	MB_IDN	Nummer des freizugebenden Parameters
	ParameterBit	INT	Bit innerhalb des Parameters; "CYCLIC_ALLBITS" für gesamten Parameter
	Write	BOOL	FALSE: Parameter lesen TRUE: Parameter schreiben
VAR_OUTPUT	Done	BOOL	Parameter wurde aufgelegt
	Active	BOOL	Der Funktionsbaustein ist aktiv
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Grundsätzliche Fehlerursache
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe Fehlercodes , Seite 152
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Achse des Parameters

Tab. 3-98: Schnittstellenvariablen Funktionsbaustein MB_FreeCyclicParameter

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
ParameterNumber	MB_IDN	0	n.def.	0	steigende Flanke an "Execute"
ParameterBit	INT	-1	31	CYCLIC_ALLBITS	steigende Flanke an "Execute"
Write	BOOL			FALSE	steigende Flanke an "Execute"

Tab. 3-99: Eingangsverhalten Funktionsbaustein MB_FreeCyclicParameter

Funktionsbeschreibung

Dieser Funktionsbaustein arbeitet grundsätzlich nur in sercos Phase 2. Bei steigender Flanke an "Execute" werden die Eingänge "ParameterNumber", "ParameterBit" und "Write" übernommen. Anhand des Eingangs "Write" wird zwischen den Istdaten (UserActualData) und den Sollwerten (UserCmdData) unterschieden. "ParameterBit" gibt die Bitnummer innerhalb des Parameters an. Liegt "ParameterBit" zwischen 0 und 31, wird ein Bit-Datenkanal verwendet. Ist "ParameterBit" gleich "CYCLIC_ALLBITS", wird der gesamte Parameter aufgelegt. "ParameterNumber" gibt den freizugebenden Parameter an.

MB_FreeCyclicParameter überprüft im ersten Schritt, ob der Parameter sich in der aktuellen Konfiguration befindet. Ist dies nicht der Fall, wird eine entsprechende Fehlermeldung ausgegeben. Anschließend wird die Freigabe in

einer internen statischen Variable gespeichert. MB_FreeCyclicParameter ändert die eigentliche Konfiguration der zyklischen Datenkanäle nicht. Die Freigabe wird lediglich in der internen Kontrollstruktur vermerkt. Nach erfolgreichem Abschluss der Freigabe des zyklischen Datenkanals wird der "Done"-Ausgang für mindestens einen Zyklus auf TRUE gesetzt. "Done" bleibt TRUE, solange der Eingang "Execute" aktiv ist.

- Fehlerverhalten**
- Ist der Parameter nicht im zyklischen Datenkanal aufgelegt oder tritt ein anderer Fehler auf, ist "Error" TRUE.
- Fehlercodes**
- Der Funktionsblock generiert die folgenden Fehlermeldungen der "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#000D	16#0000	AxisRef liegt außerhalb des gültigen Bereichs
ACCESS_ERROR	16#0010	16#0000	sercos Phase 2 ist nicht aktiv
ACCESS_ERROR	16#1504	16#0000	AxisData-Struktur nicht aktiviert
ACCESS_ERROR	16#F211	16#0000	Parameter nicht im zyklischen Datenkanal gefunden

Tab. 3-100: Fehlercodes Funktionsbaustein MB_FreeCyclicParameter

3.10.4 MB_GetCyclicChannelConfig

- Kurzbeschreibung**
- Der Funktionsbaustein MB_GetCyclicChannelConfig liest die Konfiguration der zyklischen Datenkanäle einer Achse.
- Es werden sowohl die Lese-, als auch die Schreibkanäle ausgelesen und in einer Datenstruktur abgelegt.
- Die Anzahl der Einträge ist geräteabhängig und wird in einem gesonderten Eintrag ausgegeben.



Der Funktionsbaustein "MB_GetCyclicChannelConfig (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-101: Referenztable MB_GetCyclicChannelConfig

Schnittstellenbeschreibung

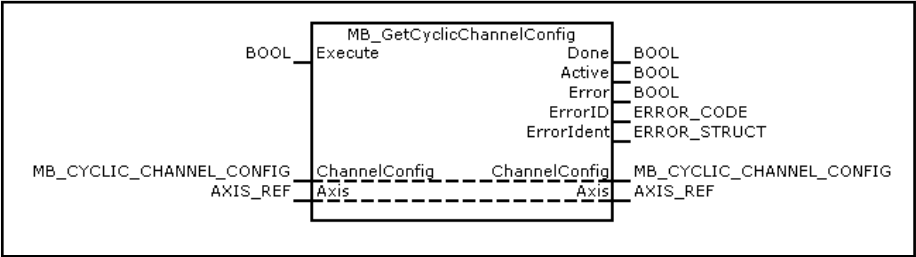


Abb. 3-81: Funktionsbaustein MB_GetCyclicChannelConfig

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Startet die Konfiguration des Parameter
VAR_OUTPUT	Done	BOOL	Parameter wurde aufgelegt
	Active	BOOL	Der Funktionsbaustein ist aktiv
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Grundsätzliche Fehlerursache
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe Fehlercodes , Seite 153
VAR_IN_OUT	ChannelConfig	MB_CYCLIC_CHANNEL_CONFIG	Ausgelesene Konfiguration MB_CYCLIC_CHANNEL_CONFIG , Seite 161
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Achse des Parameters

Tab. 3-102: Schnittstellenvariablen Funktionsbaustein MB_GetCyclicChannelConfig

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich

Tab. 3-103: Eingangsverhalten MB_GetCyclicChannelConfig

Funktionsbeschreibung

Dieser Funktionsbaustein arbeitet grundsätzlich nur in den sercos Phasen 2 und 4. Zyklische Datenkanäle sind nur bei realen Achsen vorhanden. Demzufolge kann der Baustein nur für reale Achsen verwendet werden. Bei steigender Flanke an "Execute" wird die angelegte Achse überprüft und die Konfiguration der zyklischen Datenkanäle ausgelesen. Diese Operation dauert mehrere Zyklen. Während dieser Zeit ist der Ausgang "Active" gesetzt. Nach erfolgreichem Abschluss des Auslesevorgangs wird "Active" zurückgesetzt und "Done" gesetzt. Die Konfiguration steht jetzt am Ausgang „ChannelConfig“ zur Verfügung. Der Ausgang "ChannelConfig" ist nur nach gesetztem "Done"-Ausgang gültig.

Die Anzahl der verfügbaren Datenkanäle ist von der Achse abhängig und ist in den "...Count" Elementen der "ChannelConfig"-Struktur hinterlegt. Die Elemente "ReadChannel", "WriteChannel", "ReadBitChannel" und "WriteBitChannel" enthalten die aufgelegten Parameternummern. Ist kein Parameter aufgelegt, so ist 0 eingetragen. Die Elemente "ReadBitChannelBit" und "WriteBitChannelBit" enthalten die aktive Bitnummer des Parameters.

Fehlerverhalten

Tritt während der Abarbeitung ein Fehler auf, werden die Ausgänge "Active" und "Done" zurückgesetzt und "Error" gesetzt. "ErrorID" und "ErrorIdent" enthalten bei gesetztem "Error"-Ausgang nähere Informationen über den Fehler.

Fehlercodes

Der Funktionsblock generiert die folgenden Fehlermeldungen der "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#000D	16#0000	AxisRef liegt außerhalb des gültigen Bereichs

Tab. 3-104: Fehlercodes Funktionsbaustein MB_GetCyclicChannelConfig

3.10.5 MB_GetCyclicParameterHandle

Kurzbeschreibung Dieser Funktionsbaustein sucht einen zyklisch zu lesenden oder zu beschreibenden Parameter im zyklischen Datenkanal einer Achse und legt alle benötigten Informationen in einem Handle ab.

 Die Funktion "MB_GetCyclicParameterHandle (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-105: Referenztabelle MB_GetCyclicParameterHandle

Schnittstellenbeschreibung

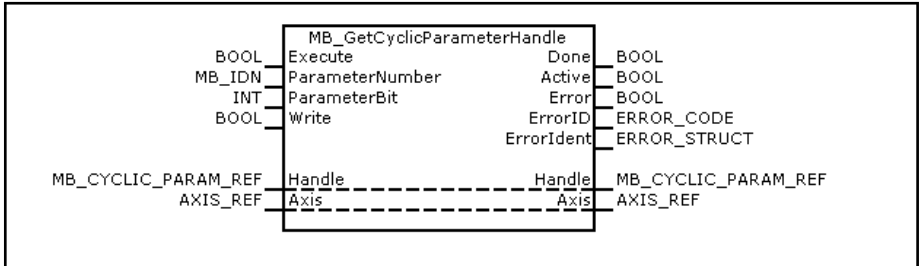


Abb. 3-82: Funktionsbaustein MB_GetCyclicParameterHandle

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	startet die Suche nach dem Parameter
	ParameterNumber	MB_IDN	Nummer des zyklisch aufgelegten Parameters
	ParameterBit	INT	Bit innerhalb des Parameters; "CYCLIC_ALLBITS" für den gesamten Parameter
	Write	BOOL	FALSE: Parameter lesen TRUE: Parameter schreiben
VAR_OUTPUT	Done	BOOL	Parameter wurde gefunden und "Handle" ist gültig
	Active	BOOL	der Funktionsbaustein ist aktiv
	Error	BOOL	zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	grundsätzliche Fehlerursache
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe Fehlercodes, Seite 155
VAR_IN_OUT	Handle	MB_CYCLIC_PARAM_REF	enthält Informationen für den Zugriff auf den zyklischen Parameter MB_CYCLIC_PARAM_REF, Seite 159
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Achse des Parameters

Tab. 3-106: Schnittstellenvariablen Funktionsbaustein MB_GetCyclicParameterHandle

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
ParameterNumber	MB_IDN	0	n.def.	0	steigende Flanke an "Execute"
ParameterBit	INT	-1	31	CYCLIC_ALLBITS	steigende Flanke an "Execute"
Write	BOOL			FALSE	steigende Flanke an "Execute"

Tab. 3-107: Eingangsverhalten MB_GetCyclicParameterHandle

Funktionsbeschreibung

Dieser Funktionsbaustein wird im Betriebsmodus (sercos Phase 4) verwendet. Nach einer steigenden Flanke an "Execute" werden die Eingänge "ParameterNumber", "ParameterBit" und "Write" gelesen. Der "Write"-Eingang entscheidet, ob Sollwerte des MDT oder Istwerte des AT gesucht werden. Ist "ParameterBit"="CYCLIC_ALLBITS", wird der komplette Parameter aufgelegt. Wird in "ParameterBit" eine Bitnummer angegeben, wird erst in dem Bitkanal gesucht. Wurde der Parameter nicht in den Bitkanälen gefunden, und handelt es sich um einen Istwert, werden die Parameterkanäle durchsucht. Nach erfolgreichem Abschluss der Bearbeitung ist "Done" aktiv und die Daten im angelegten "Handle" sind gültig.

Fehlerverhalten

Wird der Parameter nicht im zyklischen Datenkanal gefunden oder tritt ein anderer Fehler auf, ist "Error" TRUE und "Handle" wird nicht verändert.

Fehlercodes

Der Funktionsblock generiert die folgenden Fehlermeldungen der "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#000D	16#0000	AxisRef liegt außerhalb des gültigen Bereichs
ACCESS_ERROR	16#1504	16#0000	AxisData-Struktur nicht aktiviert
ACCESS_ERROR	16#F211	16#0000	Parameter nicht im zyklischen Datenkanal gefunden
INPUT_INVALID_ERROR	16#F212	16#0000	Parametertyp Struktur nicht unterstützt
INPUT_INVALID_ERROR	16#F213	16#0000	Parametertyp Funktion nicht unterstützt
INPUT_INVALID_ERROR	16#F214	16#0000	Parametertyp nicht unterstützt
INPUT_INVALID_ERROR	16#F215	16#0000	Parameter ist nicht beschreibbar

Tab. 3-108: Fehlercodes Funktionsbaustein MB_GetCyclicParameterHandle

3.10.6 MB_ReadCyclicParameter

Kurzbeschreibung

Diese Funktion liest einen zyklischen Parameter über den "Handle" und gibt den aktuellen Inhalt als DINT zurück. Nachkommastellenbehaftete Werte werden entsprechend der im Sercos-Attribut abgelegten Wichtung gewandelt. Diese Funktion kann auch für Bit-Werte verwendet werden.



Die Funktion "MB_ReadCyclicParameter (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek	Zielsystem	Bibliothek
	MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase
	MLD (IL2G)	MX_TechMotionBase

Tab. 3-109: Referenztabelle MB_ReadCyclicParameter

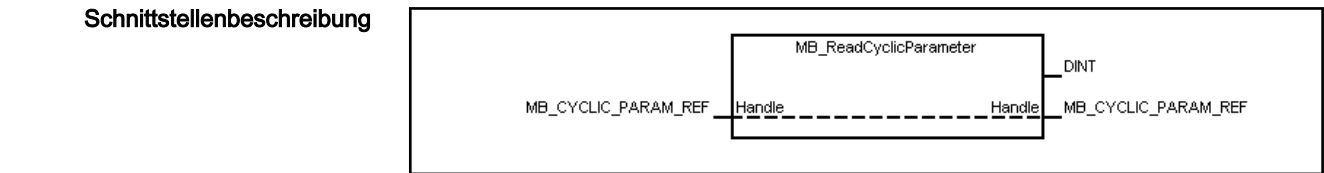


Abb. 3-83: Funktion MB_ReadCyclicParameter

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	Handle	MB_CYCLIC_PARAM_REF	Enthält Informationen für den Zugriff auf den zyklischen Parameter MB_CYCLIC_PARAM_REF , Seite 159
Rückgabewert		DINT	Wert des Parameters

Tab. 3-110: Schnittstellenvariablen Funktion MB_ReadCyclicParameter

3.10.7 MB_ReadCyclicRealParameter

Kurzbeschreibung Diese Funktion liest einen zyklischen Parameter über den "Handle" und gibt den aktuellen Inhalt als REAL zurück. Ist der Datentyp des Parameters nicht REAL, findet eine Typwandlung statt. Diese Funktion kann nicht für Bit-Werte verwendet werden.

 Die Funktion "MB_ReadCyclicRealParameter (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek	Zielsystem	Bibliothek
	MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase
	MLD (IL2G)	MX_TechMotionBase

Tab. 3-111: Referenztabelle MB_ReadCyclicRealParameter

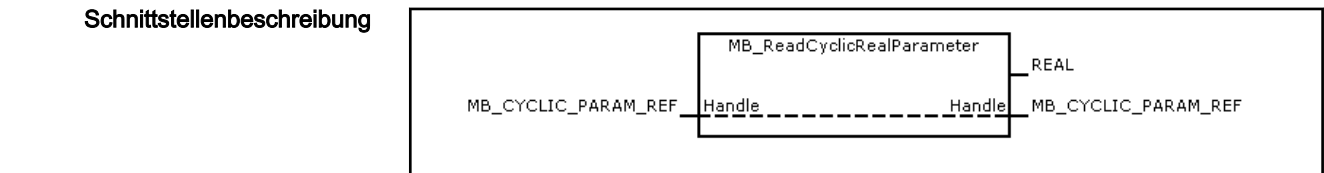


Abb. 3-84: Funktion MB_ReadCyclicRealParameter

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	Handle	MB_CYCLIC_PARAM_REF	Enthält Informationen für den Zugriff auf den zyklischen Parameter MB_CYCLIC_PARAM_REF , Seite 159
Rückgabewert		REAL	Wert des Parameters

Tab. 3-112: Schnittstellenvariablen Funktion MB_ReadCyclicRealParameter

3.10.8 MB_WriteCyclicParameter

Kurzbeschreibung Diese Funktion schreibt einen zyklischen Parameter über den "Handle". Ist der Zieldatentyp nachkommastellenbehaftet, wird der Wert entsprechend der im Sercos-Attribut angegebenen Nachkommastellen gewandelt. Diese Funktion kann auch für Bit-Werte verwendet werden.



Die Funktion "MB_WriteCyclicParameter (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-113: Referenztabelle MB_WriteCyclicParameter

Schnittstellenbeschreibung

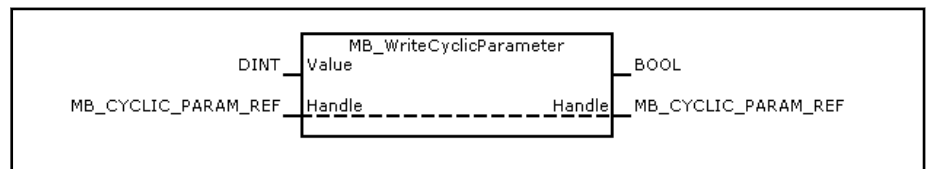


Abb. 3-85: Funktion MB_WriteCyclicParameter

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Value	DINT	Zu schreibender Wert
VAR_IN_OUT	Handle	MB_CYCLIC_PARAM_REF	Enthält Informationen für den Zugriff auf den zyklischen Parameter MB_CYCLIC_PARAM_REF, Seite 159
Rückgabewert		BOOL	Dummy Ausgang ohne Bedeutung; ist immer TRUE

Tab. 3-114: Schnittstellenvariablen Funktion MB_WriteCyclicParameter

3.10.9 MB_WriteCyclicRealParameter

Kurzbeschreibung Diese Funktion schreibt einen zyklischen Parameter über den "Handle". Bei ganzzahligem Zieldatentyp ohne Nachkommastellen wird gerundet. Diese Funktion kann nicht für Bit-Werte verwendet werden.

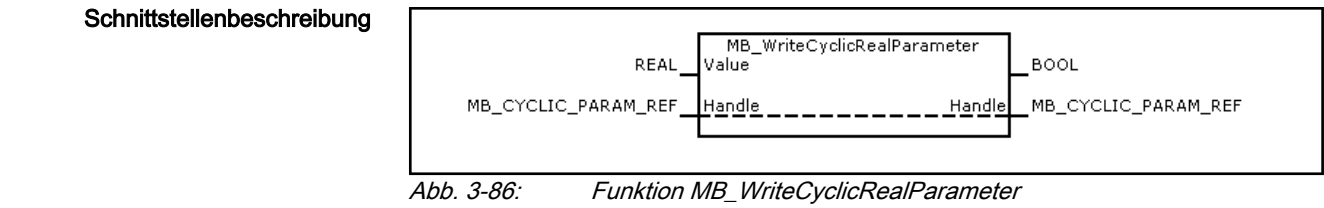


Die Funktion "MB_WriteCyclicRealParameter (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-115: Referenztabelle MB_WriteCyclicRealParameter



E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Value	REAL	Zu schreibender Wert
VAR_IN_OUT	Handle	MB_CYCLIC_PARAM_REF	Enthält Informationen für den Zugriff auf den zyklischen Parameter MB_CYCLIC_PARAM_REF , Seite 159
Rückgabewert		BOOL	Dummy Ausgang ohne Bedeutung; ist immer TRUE

Tab. 3-116: Schnittstellenvariablen Funktion MB_WriteCyclicRealParameter

3.10.10 MB_GetLastCyclicParameterError

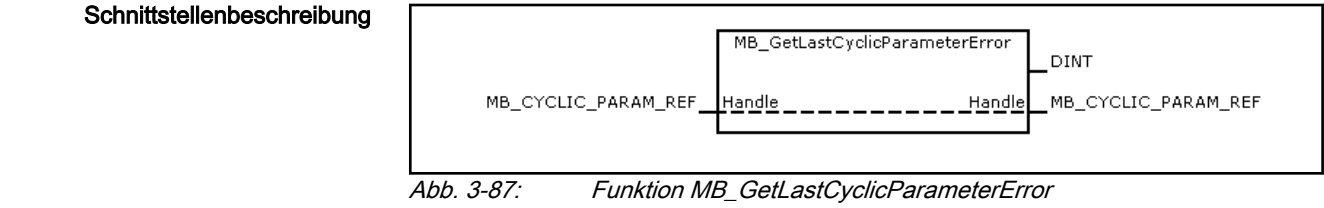
Kurzbeschreibung Diese Funktion gibt den letzten aufgetretenen Fehler beim Lesen oder Schreiben eines zyklischen Wertes für einen "Handle" zurück und löscht den Fehler.

Diese Funktion ist optional und muss nicht aufgerufen werden. Es wird jedoch empfohlen die zyklische Datenkommunikation in regelmäßigen Abständen auf aufgetretene Fehler zu prüfen.

Die Funktion "MB_GetLastCyclicParameterError (ML_TechBase-Bibliothek)" wurde mit Version 14VRS in die "TechMotionBase-Bibliothek" übernommen.

Zuordnung Zielsystem/Bibliothek	Zielsystem	Bibliothek
	MLC/XLC 14VRS Funktionspaket MOT	ML_TechMotionBase
	MLD (IL2G)	MX_TechMotionBase

Tab. 3-117: Referenztable MB_GetLastCyclicParameterError



E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT	Handle	MB_CYCLIC_PARAM_REF	Enthält Informationen für den Zugriff auf den zyklischen Parameter MB_CYCLIC_PARAM_REF , Seite 159
Rückgabewert		DINT	Fehlercode der letzten lesenden oder schreibenden Aktion

Tab. 3-118: Schnittstellenvariablen Funktion MB_GetLastCyclicParameterError

Fehlercodes Die Funktion MB_GetLastCyclicParameterError generiert die folgenden Fehler:

ErrorID	Beschreibung
16#0001	kann Parameter nicht schreiben - Parameter nur zum Lesen konfiguriert
16#0002	falscher Typ für Bit-Zugriff
16#0003	falscher Typ für Parameter-Zugriff
16#0004	ungültige Achsnummer
16#0005	ungültiger Datenkanal
16#0006	ungültiger Handle

Tab. 3-119: Fehlercodes Funktion *GetLastCyclicParameterError*

3.10.11 MB_CYCLIC_PARAM_REF

Typdefinitionen Die folgende Definition beschreibt Datentypen, die im Funktionsbaustein [MB_GetCyclicParameterHandle](#), Seite 154, verwendet werden.

Die Typ-Definitionen sind in der Bibliothek enthalten.

Die Struktur "MB_CYCLIC_PARAM_REF" darf gelesen, jedoch nicht beschrieben werden. Eine Veränderung des Inhalts kann zu Fehlfunktionen führen. Der Aufbau der Struktur kann sich in späteren Versionen ändern. Der Inhalt sollten nur zu Debug-Zwecken gelesen werden.

Die Struktur "MB_CYCLIC_PARAM_REF" sieht wie folgt aus:

Strukturelement	Typ	Beschreibung
ParameterNumber	MB_IDN	FP-Nummer des Parameters
AxisNo	OBJECTS	Logische Nummer der Achse
ParamType	MB_CYCLIC_TYPES	Datentyp des Parameters MB_CYCLIC_TYPES , Seite 159
Bit	INT	Bit-Nummer innerhalb des Parameters
AccessChannel	MB_CYCLIC_CHANNELS	Datenkanal in dem der Parameter gefunden wurde MB_CYCLIC_CHANNELS , Seite 160
ParamAdr	POINTER TO SM_TYPES	Zeiger auf den zyklischen Parameter
BitParamAdr	POINTER TO WORD	Zeiger auf den zyklischen Bit-Parameter
Factor	DINT	Wird zum Umrechnen von REAL zu DINT verwendet
ErrorCode	DWORD	Fehlercode des letzten aufgetretenen Fehlers, siehe Fehlercodes , Seite 158

Tab. 3-120: Definition der Basiselemente der MB_CYCLIC_PARAM_REF-Struktur

3.10.12 MB_CYCLIC_TYPES

Der Datentyp "MB_CYCLIC_TYPES" ist ein Aufzählungstyp und hat die folgenden Einträge:

Element	Wert
tINT	1
tUINT	2

Element	Wert
tDINT	3
tUDINT	4
tREAL	5
tBIT	6
tDV_SINT	7
tDV_USINT	8
tDV_INT	9
tDV_UINT	10
tDV_DINT	11
tDV_UDINT	12
tDV_LINT	13
tDV_ULINT	14
tDV_REAL	15
tDV_LREAL	16

Tab. 3-121: Definition des Aufzählungstyps MB_CYCLIC_TYPES

3.10.13 MB_CYCLIC_CHANNELS

Der Datentyp "MB_CYCLIC_CHANNELS" ist ein Aufzählungstyp und hat die folgenden Einträge:

Element	Wert
UserCmdDataA	0
UserCmdDataB	1
UserCmdDataC	2
UserCmdDataD	3
UserCmdDataE	4
UserCmdDataF	5
UserCmdDataBitA	32
UserCmdDataBitB	33
UserCmdDataBitC	34
UserCmdDataBitD	35
WriteDV	255
UserActualDataA	256
UserActualDataB	257
UserActualDataC	258
UserActualDataD	259
UserActualDataE	260
UserActualDataF	261

Element	Wert
UserActualDataBitA	288
UserActualDataBitB	289
UserActualDataBitC	290
UserActualDataBitD	291
ReadDV	511

Tab. 3-122: Aufzählungstyp MB_CYCLIC_CHANNELS

3.10.14 MB_CYCLIC_CHANNEL_CONFIG

Die Struktur "MB_CYCLIC_CHANNEL_CONFIG" wird vom Funktionsbaustein [MB_GetCyclicChannelConfig](#), Seite 152 beschrieben und enthält die aufgelegten Parameter im zyklischen Datenkanal der Achse.

Die Struktur "MB_CYCLIC_CHANNEL_CONFIG" sieht wie folgt aus:

Strukturelement	Typ	Min-Wert	Max-Wert	Default-Wert	Beschreibung
ReadChannelCount	INT	0	10	0	Anzahl der Lesekanäle
WriteChannelCount	INT	0	10	0	Anzahl der Schreibkanäle
ReadBitChannelCount	INT	0	10	0	Anzahl der Bit-Lesekanäle
WriteBitChannelCount	INT	0	10	0	Anzahl der Bit-Schreibkanäle
ReadChannel	ARRAY[1..10] OF MB_IDN	n.def.	n.def.	S-0-0000	Konfigurierte Parameter der Lesekanäle
WriteChannel	ARRAY[1..10] OF MB_IDN	n.def.	n.def.	S-0-0000	Konfigurierte Parameter der Schreibkanäle
ReadBitChannel	ARRAY[1..10] OF MB_IDN	n.def.	n.def.	S-0-0000	Konfigurierte Parameter der Bit-Lesekanäle
ReadBitChannelBit	ARRAY[1..10] OF UINT	0	63	0	Konfigurierte Bits der Bit-Lesekanäle
WriteBitChannel	ARRAY[1..10] OF MB_IDN	n.def.	n.def.	S-0-0000	Konfigurierte Parameter der Bit-Schreibkanäle
WriteBitChannelBit	ARRAY[1..10] OF UINT	0	63	0	Konfigurierte Bits der Bit-Schreibkanäle

Tab. 3-123: Struktur MB_CYCLIC_CHANNEL_CONFIG

4 ML_TechMotion.library // MX_TechMotion

4.1 Übersicht über die Funktionsbausteine



Die Bibliothek benötigt das TEC (Technologie) Funktionspaket der Steuerung.

Die Funktionsbausteine von ML_TechMotion

Anwendungsgebiet (Ordner in der Bibliothek)	
Funktionsbausteine	Beschreibung
Taktsynchrones Ein- und Auskoppeln (CamLock)	
MB_CamLock, Seite 170	Wird verwendet um die durch den Funktionsbaustein MB_PrepareCams berechneten und gespeicherten Run-, LockOn- und LockOff-Kurvenscheibenprofile zu aktivieren
MB_PrepareCams, Seite 168	Verwendet ein Polynom 5. Ordnung um 3 Kurvenscheibenprofile zu berechnen, zu erzeugen und sie in Parameter, die durch die MC_CAM_ID-Eingänge festgelegt sind, zu laden. Die erzeugten Bewegungsprofile enthalten Randbedingungen für die Position und Geschwindigkeit
Fliegende Säge (FlyingShear)	
ML_FlyingShear / MX_FlyingShear, Seite 178	Enthält den typischen Ablauf einer "Fliegenden Säge"
MagicBelt (MagicBelt)	
MB_SmartBeltType01, Seite 238	Führt am Produkt eine Phasenkorrektur mittels eines dreieckigen Geschwindigkeitsprofils durch
Schlaufenregelung (SagControl)	
ML_SagControlA, Seite 234	Enthält einen Schlaufenregler mit analoger Istwerterfassung der Schlaufe
ML_SagControlC, Seite 231	Enthält einen Schlaufenregler mit binären Min/Max-Kontakten
SmartBelt (SmartBelt)	
MB_MagicBeltType01, Seite 192	Enthält Funktionalitäten zur Ansteuerung einer Kette einer MagicBelt (=Mehrzugkette) Applikation
Kurvenscheibentabellen für Technologiefunktionen	
MB_PilgrimStepCalcType01, Seite 262	Übernimmt die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunkten sowie die Hübe des Vorwärts- und Rückwärtstransportes und die Stopppositionen für den Pilgerschrittbetrieb
MB_PrintLengthCorrCalcType01, Seite 272	Übernimmt die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunkten für die Technologiefunktion Drucklängenkorrektur, um die Länge eines Druckes auf dem Druckgut (Warenbahn bzw. Bogen) zu verändern
Zugkraftregler (TensionControl)	

MB_TensionControlLoadCellType01, Seite 278	Der Zugkraftregler regelt, abhängig vom Messwert einer Kraftmessdose, die Zugkraft auf den gewünschten Sollwert. Der Zugkraftregler wird direkt über die Reglerparameter parametrier
MB_TensionControlLoadCellMatParType01, Seite 292	Der Zugkraftregler regelt, abhängig vom Messwert einer Kraftmessdose, die Zugkraft auf den gewünschten Sollwert. Der Zugkraftregler wird über Material- und Streckenparameter parametrier

Tab. 4-1: Bibliothek ML_TechMotion

_PhaseOut Die nachstehend genannten Funktionsbausteine (nebst zugehöriger Datentypen) wurden aus organisatorischen Gründen in andere Bibliotheken verschoben.

Dies geschah teilweise unter Anpassung des Namens.

Neue Bibliothek: ML_TechMotionBase ab 14.3.0.0

Ordner (neu)	Name (neu)	POU-Typ	Name (alt)
DistanceAccumulator	MB_AxisDistanceAccumulator	FB	
	MB_DistanceAccumulator	FB	
SCurve	MB_ReachVelocityType02	FB	
PositionMonitoring	MB_PositionMonitoring_GantryCant	FB	
	MB_PositionMonitoring_PosCollision	FB	

Tab. 4-2: POU's, die von ML_TechMotion nach ML_TechMotionBase verschoben wurden

4.2 CamLock - Taktsynchronen Ein- und Auskoppeln

4.2.1 Einführung und Übersicht

Die MLC Steuerungen ab Firmware-Version MLC 03VRS wurden durch Bereitstellung von CamLock-Technologie-Funktionsbausteinen um die "Lock On/Lock Off" Funktionalität erweitert.

Der Funktionsbaustein [MB_PrepareCams, Seite 168](#), verwendet ein Polynom 5. Ordnung um 3 Kurvenscheibenprofile zu berechnen, zu erzeugen und sie in eine Slave-Achse zu laden. Diese Kurvenscheibenprofile werden als Lock On, Lock Off und Eins-zu-Eins bezeichnet. Sie ermöglichen einer synchronisierten Slave-Achse sich vom Master auszukoppeln (Lock Off) und an einer vorher festgelegten Position zu stoppen, bis sie wieder mit dem Master synchronisiert wird (Lock On).

Der Funktionsbaustein [MB_CamLock, Seite 170](#), wird verwendet um die Lock On/Lock Off Funktionalität zu aktivieren. Ein Anwender-Kurvenscheibenprofil wird ebenfalls unterstützt.

Solange eine Slave-Achse mit dem Master synchronisiert ist, folgt sie dem Master unter Nutzung der Eins-zu-Eins Kurvenscheibe.

Ein Phasenoffset zwischen der Master- und der Slave-Position kann definiert werden.

Wenn die Lock Off Kurvenscheibe aktiviert ist (über den MB_CamLock Funktionsbaustein), wechselt die Slave-Achse von der Master-Position zu einer vorher festgelegten Auskoppel-Position und kommt dann zum Stillstand.

Ist die Lock Off Kurvenscheibe nicht aktiv (Lock On), synchronisiert sich die Slave-Achse von ihrer Stopp-Position aus wieder mit dem Master. Sobald sie sich aufsynchronisiert hat, schaltet die Slave-Achse wieder auf die Eins-zu-Eins Kurvenscheibe und folgt weiter der Position des Master-Eingangs.

4.2.2 CamLock - Applikationsbeispiel

Diese Funktionalität wird normalerweise in Verpackungsmaschinen verwendet, wobei vorausgesetzt wird, dass die ankommenden Produkte einen gleichmässigen Abstand aufweisen, bevor sie eingepackt werden können.

Im Fall dass der Abstand zu groß ist, wird der Verpackungsprozess in der Maschine vom Master für einen oder mehrere Zyklen ausgekoppelt (Lock Off) bis das Produkt wieder erkannt wird. Diesen Zustand nennt man "No Product, No Seal". Sobald das Produkt erkannt ist, wird der Verpackungsprozess erneut auf den Master aufsynchronisiert (Lock On) und die Produkte werden weiter verpackt.

Diese nicht gleichmässig vorhandenen Abstände erfordern das Beschleunigen und Abbremsen der Slave-Achse, um sie dadurch auf den Master zu synchronisieren. Die folgende Abbildung zeigt eine typische Verpackungsmaschine zum Füllen und Versiegeln:

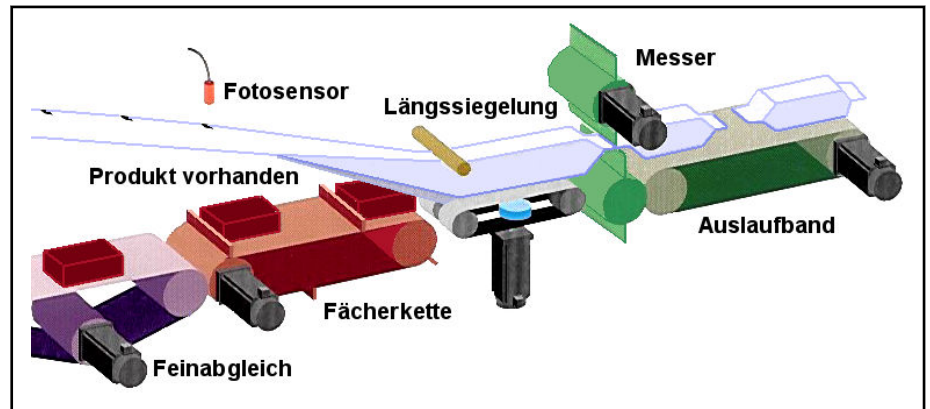


Abb. 4-1: Horizontale Form einer Verpackungsmaschine zum Füllen und Versiegeln

Das folgende Diagramm zeigt den Bearbeitungs-, den Abbrems-, Stopp-, Beschleunigungs- und den erneuten Bearbeitungs-Prozess, dem eine Slave-Achse während des Lock On/Lock Off Prozesses folgt.

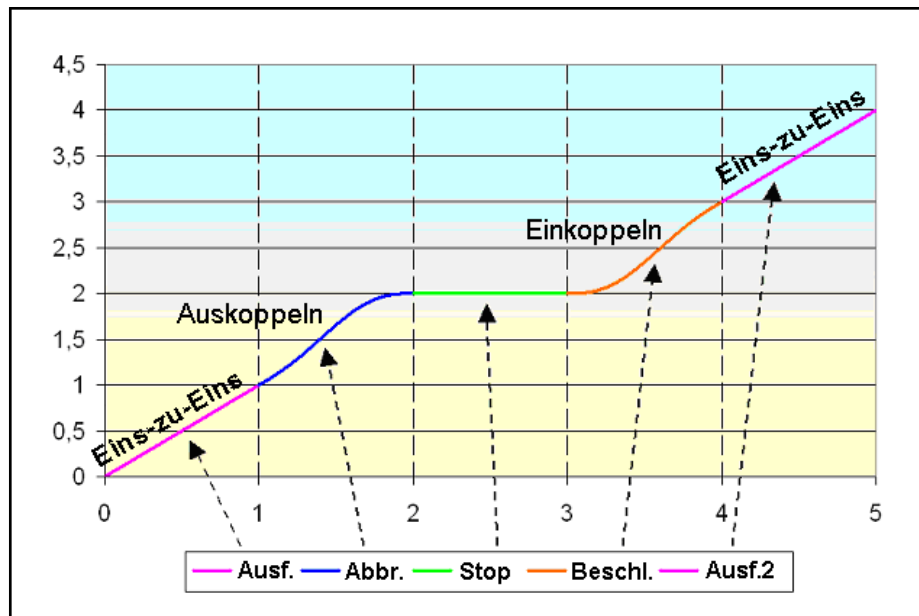


Abb. 4-2: CamLock Kurvenscheibenprofil

Eins-zu-Eins Kurvenscheibenprofil "Synchronisiert auf Master"

Das Eins-zu-Eins Kurvenscheibenprofil ist aktiv und auf den Master-Eingang synchronisiert, wenn die Lock On/Lock Off Funktionalität nicht aktiv ist. Die Lock On/Lock Off Funktionalität wird aktiv, indem der "Enable"-Eingang des MB_CamLock Funktionsbausteins auf High gesetzt wird. Der LockOff-Eingang muss Low sein. Unter normalen Arbeitsbedingungen ist dieses Kurvenscheibenprofil aktiv und folgt den Vorgaben des Master-Eingangs.

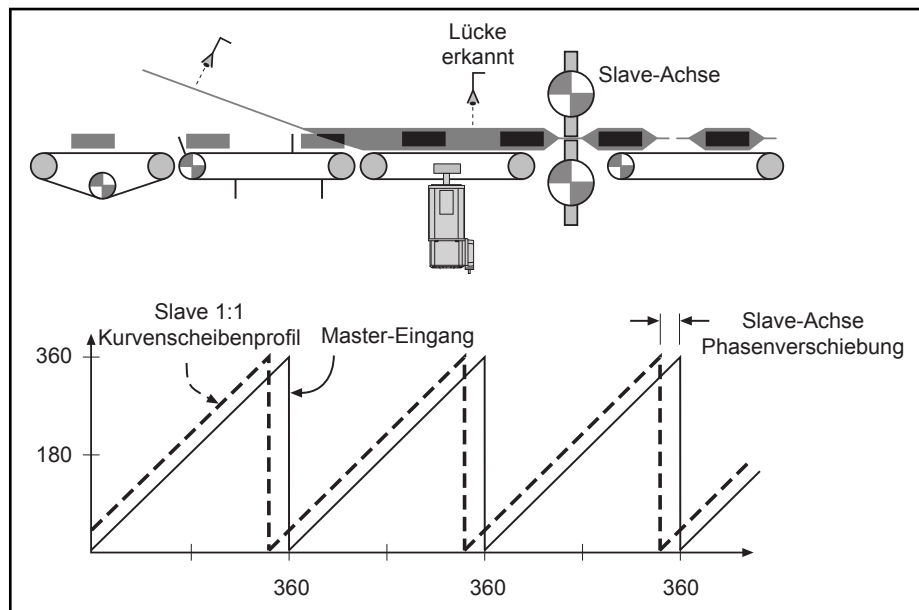


Abb. 4-3: Bearbeitungs-Kurvenscheibe aktiv, Normalbetrieb der Verpackungs-Applikation

Lock Off Kurvenscheibenprofil

Das Lock Off Kurvenscheibenprofil bremst die Slave-Achse bis zum Stillstand über einen Zyklus des Masters. Die Lock Off Kurvenscheibe ist aktiv, solange die beiden Eingänge "Enable" und "LockOff" des MB_CamLock Funktionsbausteins auf High sind. Nach diesem Zyklus wird die Geschwindigkeit der Slave-Achse gestoppt und erst wieder neu gestartet, wenn das Lock On Kurvenscheibenprofil aktiv ist.



Einmal eingeschaltet, ist das Lock Off Kurvenscheibenprofil nach dem Nulldurchgang der Slave-Achse aktiv.

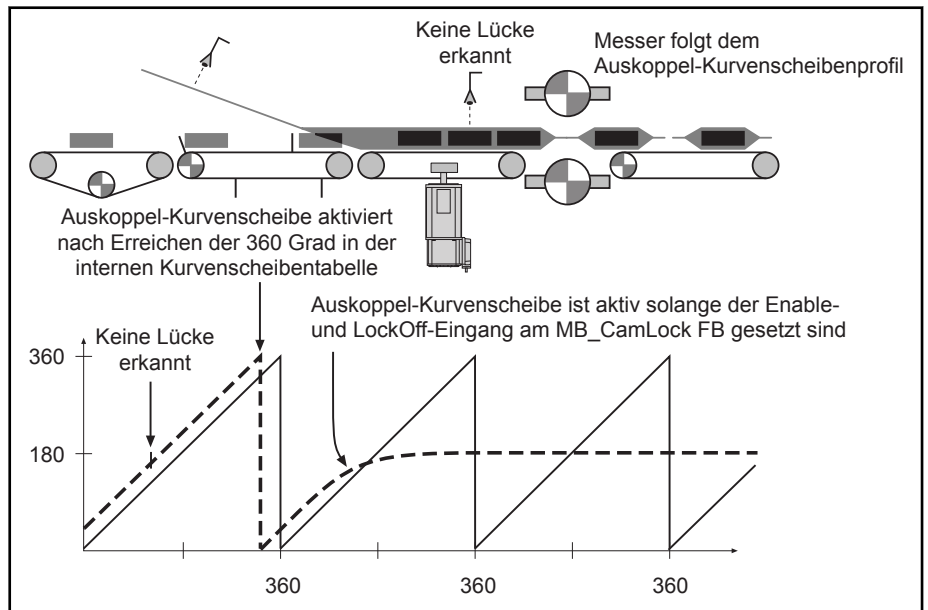


Abb. 4-4: Lock Off Kurvenscheibe aktiv, No Product - No Seal

Lock On Kurvenscheibenprofil

Das Lock On Kurvenscheibenprofil beschleunigt von einer Halteposition bis die Geschwindigkeit des Master-Eingangs über einen Zyklus des Masters (360 Grad) erreicht ist. Nach diesem Zyklus stimmt die Geschwindigkeit von Slave-Achse und Master überein. Die Lock On Kurvenscheibe ist nur solange aktiv bis die Slave-Achse mit dem Master synchronisiert ist. Danach folgt die Slave-Achse der Eins-zu-Eins Kurvenscheibe während des Normalbetriebs. Das Lock On Kurvenscheibenprofil ist aktiv, solange am MB_CamLock Funktionsbaustein der "Enable"-Eingang High und der "LockOff"-Eingang Low ist.



Einmal eingeschaltet, ist das Lock On Kurvenscheibenprofil nach dem Nulldurchgang der Master-Achse aktiv.

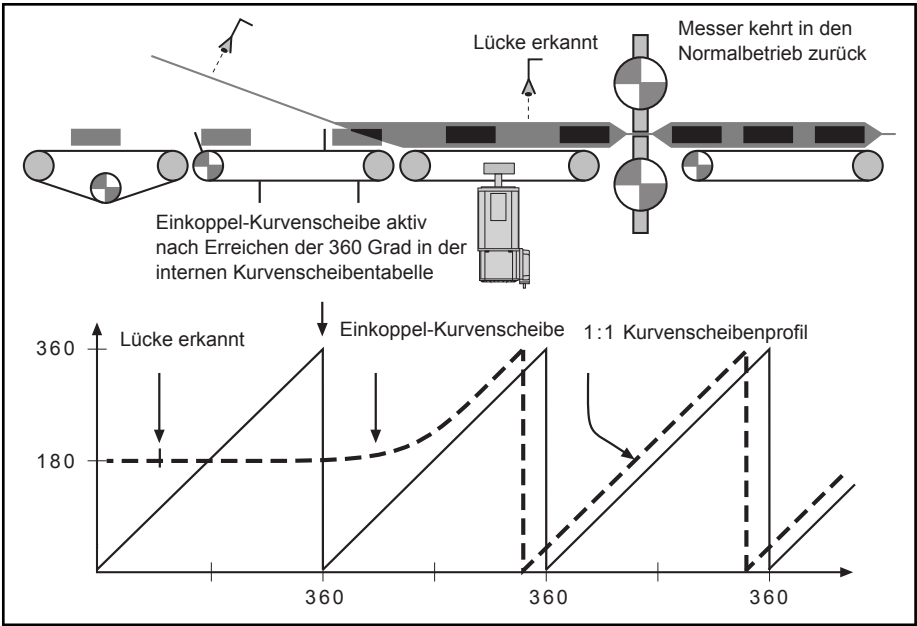


Abb. 4-5: Lock On Kurvenscheibe aktiv, Produkt ist vorhanden

4.2.3 MB_PrepareCams

Kurzbeschreibung Der Funktionsbaustein benutzt ein Polynom 5. Ordnung um 3 Kurvenscheibenprofile zu berechnen, zu erzeugen und sie in Parameter, die durch die MC_CAM_ID-Eingänge festgelegt sind, zu laden.

Die erzeugten Bewegungsprofile enthalten Randbedingungen für die Position und Geschwindigkeit.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechMotion

Tab. 4-3: Referenztable MB_PrepareCams

Schnittstellenbeschreibung

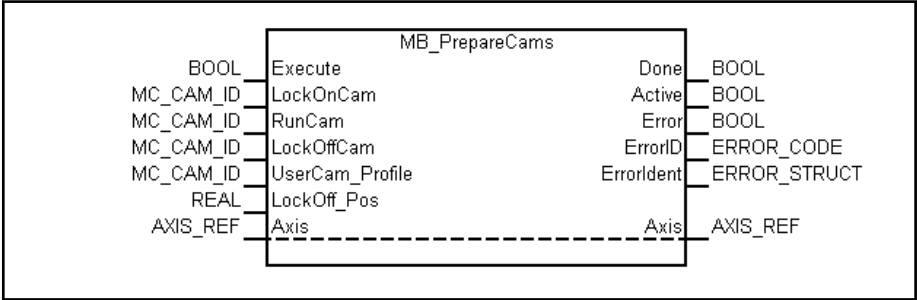


Abb. 4-6: Funktionsbaustein MB_PrepareCams



Der MC_CAM_ID-Wert legt nur fest, welcher Parameter zum Speichern des berechneten Kurvenscheibenprofils benutzt wird. Die aktuelle Funktionalität (z.B. LockOnCam) wird durch den Eingang des Funktionsbausteins bestimmt.

Für ein MLC Steuerungssystem wird bei einem Wert von 1 das Kurvenscheibenprofil im Steuerungsparameter C-0-2001 abgespeichert. Um jedoch den MB_PrepareCams-Funktionsbaustein ohne Fehler ausführen zu können, ist nach wie vor ein Achs-Eingang für "Axis" erforderlich.

Die berechneten Kurvenscheibenprofile werden in Steuerungsparametern abgespeichert.

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Positive Flanke startet die Berechnung von drei Kurvenscheibenprofilen
	LockOnCam	MC_CAM_ID	Bestimmt die Zieladresse der Lock On-Kurvenscheibentabelle. Defaultwert ist 4
	RunCam	MC_CAM_ID	Bestimmt die Zieladresse der Bearbeitungs-Kurvenscheibentabelle. Defaultwert ist 3
	LockOffCam	MC_CAM_ID	Bestimmt die Zieladresse der Lock Off-Kurvenscheibentabelle. Defaultwert ist 2
	LockOff_Pos	REAL	Dieser Eingang wird zum Berechnen des entsprechenden Geschwindigkeitsprofils benutzt
VAR_OUTPUT	Done	BOOL	Drei Kurvenscheibenprofile wurden berechnet, erzeugt und geladen
	Active	BOOL	Funktionsbaustein ist aktiv
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 170,
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Referenz zur Slave-Achse für die Kurvenscheibenprofile abgespeichert sind

Tab. 4-4: Schnittstellenvariablen Funktionsbaustein MB_PrepareCams

Funktionsbeschreibung

Eine MLC unterstützt einen Block von 98 Steuerungsparametern zum Speichern von CamLock-Kurvenscheibenprofilen.

Im Gegensatz zu den Antriebsparametern ist die Zuordnung zwischen MC_CAM_ID-Wert und Steuerungsparameter fortlaufend.

Beginnend mit Steuerungsparameter C-0-2001 wird ein MC_CAM_ID-Wert von 1 in Steuerungsparameter C-0-2001, ein MC_CAM_ID-Wert von 98 in C-0-2098 gespeichert.

Der Steuerungsparameter C-0-2099 kann nicht benutzt werden, um ein Kurvenscheibenprofil abzuspeichern; dieser Parameter ist reserviert für das "Stoppen der Slave-Achse".

Der Funktionsbaustein muss direkt nach dem Hochlauf der Steuerung gestartet werden, damit die drei Kurvenscheibenprofile berechnet und in die entsprechenden Parameter geladen werden können. Benutzen Sie den DONE-

Ausgang, um zu überprüfen, dass dieser Vorgang beendet ist, bevor das SPS-Programm weiter läuft.



Der MB_PrepareCams Funktionsbaustein wird nur einmal am Anfang des SPS-Programms und vor dem Start des Funktionsbausteins [MB_CamLock](#), [Seite 170](#), ausgeführt.

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle **F_RELATED_TABLE** folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#0003	16#0000	FB wurde durch einen anderen FB beendet
RESOURCE_ERROR	16#0004	16#0000	Diese Antriebsfirmware-Version wird nicht unterstützt
INPUT_RANGE_ERROR	16#13A1	16#0001	Die kurvenscheiben-bezogenen Werte wurden nicht korrekt initialisiert
INPUT_RANGE_ERROR	16#13A1	16#0002	Slave Axis_Ref; die AxisNo liegt ausserhalb des zulässigen Bereichs
INPUT_RANGE_ERROR	16#13A1	16#0003	LockOff_Pos muss grösser als 0 und kleiner 360 sein, Default=180 Grad
CALCULATION_ERROR	16#13A2	16#0000	Berechnung der Schrittweite, Ergebnis = 0
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Zustand der State Machine

Tab. 4-5: Fehlercodes Funktionsbaustein MB_PrepareCams

4.2.4 MB_CamLock

Kurzbeschreibung Der Funktionsbaustein MB_CamLock wird verwendet, um die durch den Funktionsbaustein [MB_PrepareCams](#), [Seite 168](#), berechneten und gespeicherten Run-, LockOn- und LockOff-Kurvenscheibenprofile zu aktivieren.

Zusätzlich zum Aktivieren von Kurvenscheibenprofilen bietet dieser Funktionsbaustein noch die folgenden Funktionalitäten:

- Elektronische Getriebeübersetzung
- Synchronisationsrichtung (SyncMode)
- MC_CamIn- und MC_CamOut-Funktionalität für Antriebs-Kurvenscheiben
- Master-Feinabgleich



Der Betrieb des Funktionsbausteines erfordert den zyklischen Aufruf in der Motion-Task. Bitte beachten Sie dazu die Parametrierung in [MB_CamLock Parametrierung](#), [Seite 177](#).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechMotion

Tab. 4-6: Referenztable MB_CamLock

Schnittstellenbeschreibung

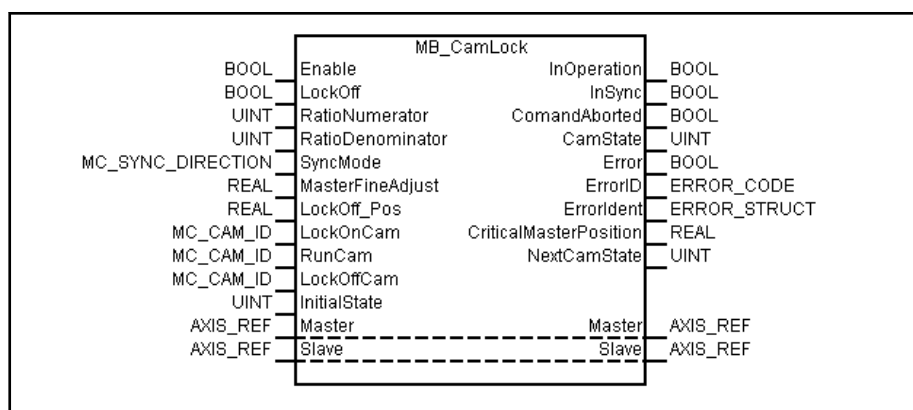


Abb. 4-7: Funktionsbaustein MB_CamLock



Für das MLC-Steuerungssystem können Steuerungskurvenscheiben verwendet werden, um dem Slave-Eingang eine virtuelle Achse zuzuweisen.

Die folgende Tabelle zeigt die verschiedenen Kurvenscheibenprofile, die durch den MB_CamLock Funktionsbaustein gesteuert werden:

Kurvenscheibenprofil	Enable-Eingang	Lock Off-Eingang	Beschreibung
Run	High	Low	Normalbetrieb unter Nutzung eines 1:1 Kurvenscheibenprofils
LockOff	High	Low nach High	Die Slave-Achse schaltet von der Run-Kurvenscheibe zum LockOff-Kurvenscheibenprofil und kommt zum Stillstand an der durch den LockOff_Pos-Eingang festgelegten Position
LockOn	High	High nach Low	Die Slave-Achse beschleunigt von einer Halteposition unter Nutzung des LockOn-Kurvenscheibenprofils, synchronisiert sich mit dem Master und schaltet zur Run-Kurvenscheibe

Tab. 4-7: Normaler CamLock-Betrieb

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert den MB_CamLock FB. Durch eine steigende Flanke wird der Slave vor Eintritt in den Run-Zustand dynamisch aufsynchronisiert. Eine fallende Flanke bewirkt eine Getriebe- Auskopplung
	LockOff	BOOL	TRUE: Locked off (ausgekoppelt) FALSE: Locked on (eingekoppelt) Ist dieser Eingang vor Setzen von Enable TRUE, dann synchronisiert sich die Achse dynamisch auf, geht in den Run-zustand über und koppelt sich dann aus (LockOff)
	RatioNumerator	UINT	Elektronische Getriebeübersetzung
	RatioDenominator	UINT	Elektronische Getriebeübersetzung

E/A-Typ	Name	Datentyp	Beschreibung
	SyncMode	MC_SYNC_DIRECTION	0: = kürzeste Wegstrecke 1: = positive Richtung 2: = negative Richtung
	MasterFineAdjust	REAL	Dieser Eingang wird für den MB_MotionProfile FB benötigt
	LockOff_Pos	REAL	Die Position in Grad an der die Slave-Achse stoppt wenn sie vom Master ausgekoppelt wird. Der Defaultwert ist 180 Grad
	LockOnCam	MC_CAM_ID	Bestimmt die Bezugsquelle der LockOn-Kurvenscheibentabelle. Defaultwert ist 4
	RunCam	MC_CAM_ID	Bestimmt die Bezugsquelle der Run-Kurvenscheibentabelle. Defaultwert ist 3
	LockOffCam	MC_CAM_ID	Bestimmt die Bezugsquelle der LockOff-Kurvenscheibentabelle. Defaultwert ist 2
	InitialState	UINT	Ausgangszustand der Zustandsmaschine
VAR_OUTPUT	InOperation	BOOL	Ist TRUE, solange der FB aktiviert ist
	InSync	BOOL	Der InSync-Ausgang ist TRUE, wenn sich die Run-Kurvenscheibe im synchronisierten Zustand befindet
	CommandAborted	BOOL	Kommando wurde durch einen anderen FB abgebrochen
	CamState	UINT	Aktueller Status des CamLock FB 1: in Dynamic Sync (dynamischer Synchronisation) 2: im Run (Bearbeitungszustand) 3: im LockOff (ausgekoppelten Zustand) 4: im Standstill (Stoppzustand) 5: im LockOn (eingekoppelten Zustand) 6: in Gear Out [Getriebe-Auskopplung (fortlaufende Bewegung)]
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 176
	CriticalMasterPosition	REAL	Stellt die notwendige kritische Master-Position für den Schaltzustandsübergang bereit
	NextCamState	UINT	Zeigt den nächsten zu verarbeitenden Ausgang an
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Reale oder virtuelle Achse
	Master	AXIS_REF	Reale oder virtuelle Achse

Tab. 4-8: Schnittstellenvariablen Funktionsbaustein MB_CamLock

Signal-Zeit-Diagramm

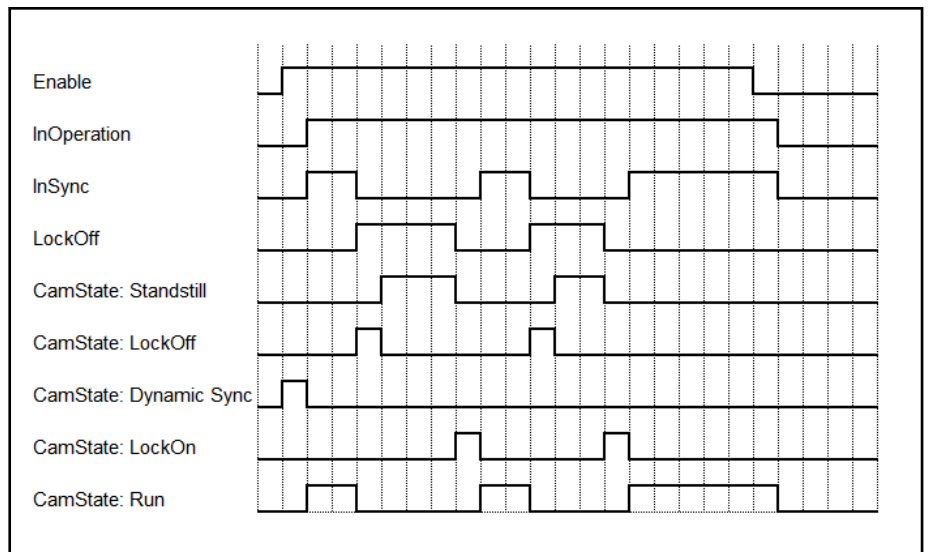


Abb. 4-8: MB_CamLock Signal-Zeit-Diagramm

Funktionsbeschreibung

Ist der **Enable**-Eingang auf High und ist gleichzeitig der **LockOff**-Eingang auf Low gesetzt, synchronisiert sich die Slave-Achse dynamisch mit dem Master und schaltet sofort in den Run-Zustand. Der **CamState**-Ausgang wechselt von Dynamic Sync (1) in den Run-Zustand (2). Der Funktionsbaustein bleibt im Run-Zustand (bearbeitet eine 1:1 Kurvenscheibe) bis der **LockOff**-Eingang auf High geht.

Geht der **LockOff**-Eingang während der Bearbeitung auf High, führt der Funktionsbaustein eine LockOff-Kurvenscheibe aus und stoppt an der Position, die über den **LockOff_Pos**-Eingang vorgegeben ist. Während des Übergangs von Run zu Stop geht der Ausgang **CamState** von Run (2) über LockOff (3) zu Standstill (4). Der Defaultwert für die LockOff_Pos ist 180 Grad, welches die halbe Wegstrecke von LockOn zu Run und von Run zu LockOff ist.

Zu diesem Zeitpunkt bleibt der Funktionsbaustein im Stoppzustand (Standstill) bis LockOff auf Low geht. Dies beendet den Zyklus. Um den CamLock-Funktionsbaustein zu deaktivieren wird empfohlen, dass man sich im CamState=4 (Stoppzustand) befindet, und den Enable-Eingang auf Low setzt.

Ist der Eingang **Enable** während des Run-Zustands auf Low gesetzt, entsynchronisiert sich der Slave von der LockOn/LockOff-Kurvenscheibenfunktionalität und geht in die kontinuierliche Bewegung über.

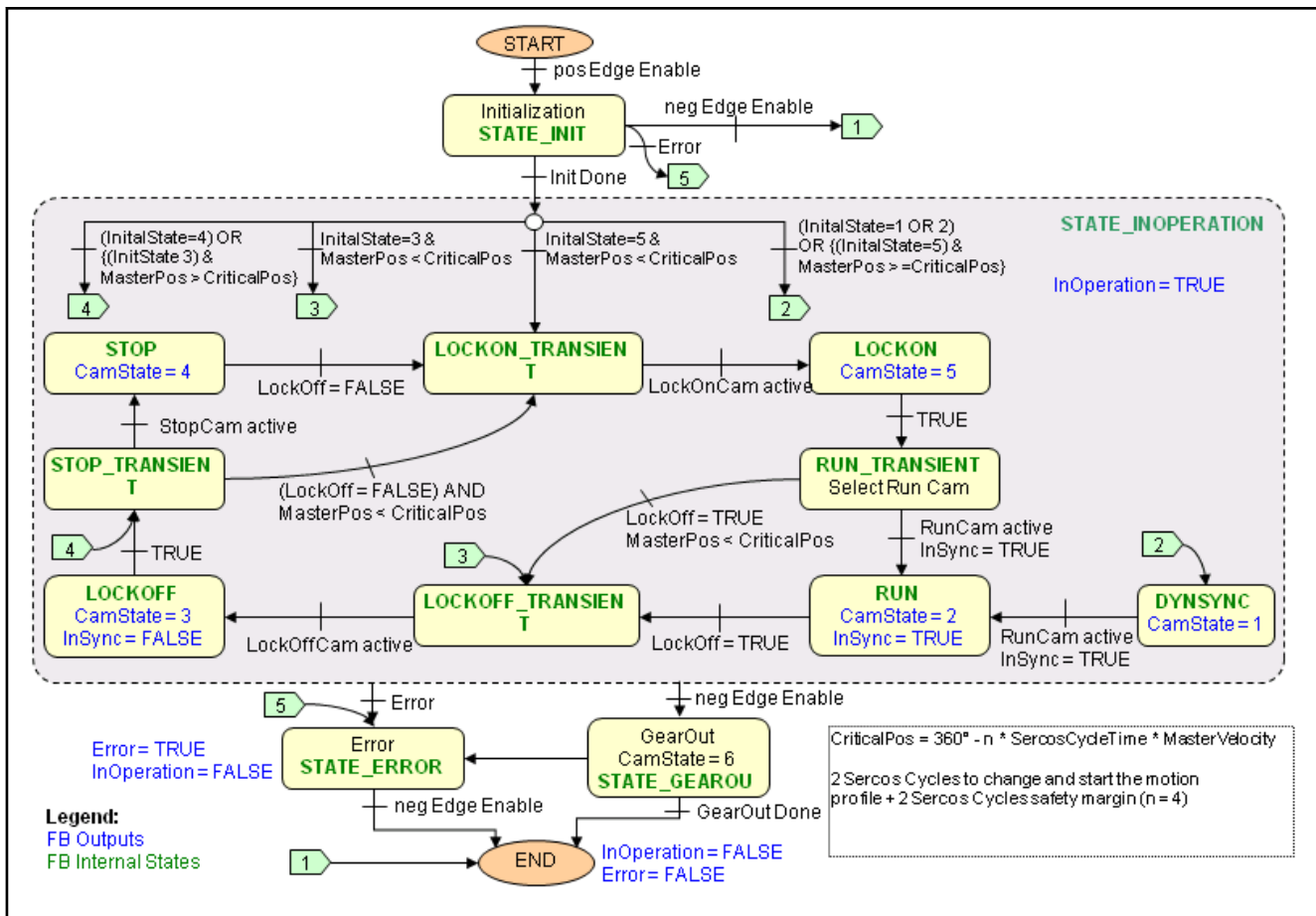
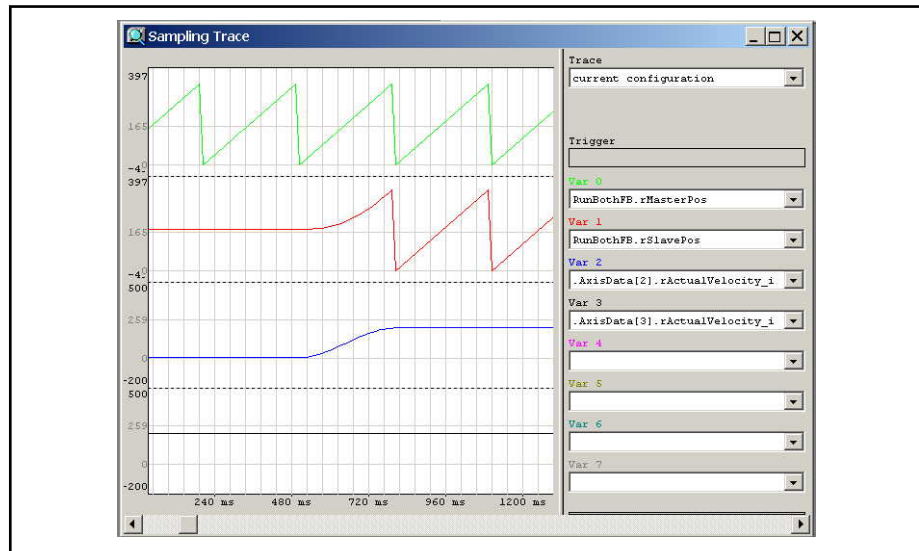


Abb. 4-9: CamLock Zustandsmaschine

LockOn/LockOff Ablaufbeispiele



Kanal 0 (grün)= Master-Position
 Kanal 1 (rot) = Slave-Position
 Kanal 2 (blau)= Slave-Geschwindigkeit
 Master-Geschwindigkeit= 200 U/min

Abb. 4-10: LockOn-Profil mit einer LockOff_Pos von 180 Grad

Die obige Abbildung zeigt das Positions- und Geschwindigkeits-Profil während des LockOn-Profiles. Die LockOff_Pos beträgt 180 Grad (Default). Die

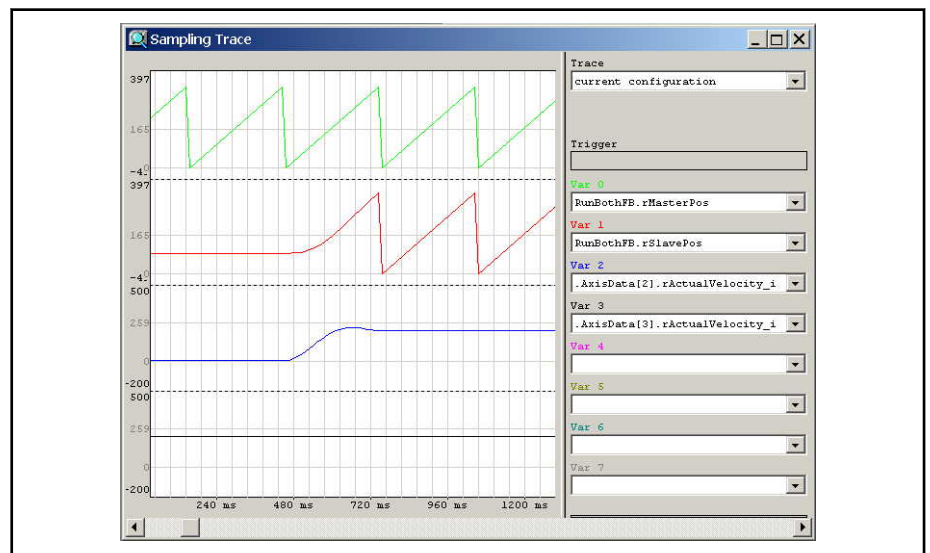
obige Abbildung zeigt das Positions- und Geschwindigkeits-Profil während des LockOff-Profiles ohne Sprünge und Überschwüngen. Die LockOff-Position ist 180 Grad (Default).



Kanal 0 (grün)= Master-Position
 Kanal 1 (rot) = Slave-Position
 Kanal 2 (blau)= Slave-Geschwindigkeit
 Master-Geschwindigkeit= 200 U/min

Abb. 4-11: LockOff-Profil mit einer LockOff_Pos von 180 Grad

Die obige Abbildung zeigt das Positions- und Geschwindigkeits-Profil während des LockOff-Profiles. Die LockOff_Pos beträgt 180 Grad (Default). Die obige Abbildung zeigt das Positions- und Geschwindigkeits-Profil während des LockOff-Profiles ohne Sprünge und Überschwüngen. Die LockOff-Position ist 180 Grad (Default).



Kanal 0 (grün)= Master-Position
 Kanal 1 (rot) = Slave-Position
 Kanal 2 (blau)= Slave-Geschwindigkeit
 Master-Geschwindigkeit= 200 U/min

Abb. 4-12: LockOn-Profil mit einer LockOff_Pos von 90 Grad

Die obige Abbildung zeigt das Positions- und Geschwindigkeits-Profil während des LockOn-Profiles, wenn die LockOff_Pos 90 Grad ist. Zu sehen ist,

dass bei einer LockOff-Position von 90 Grad die Positions- und Geschwindigkeits-Profile noch in Ordnung sind (es gibt eine Geschwindigkeits-Überschwingung, aber keine Rückwärtsbewegung).

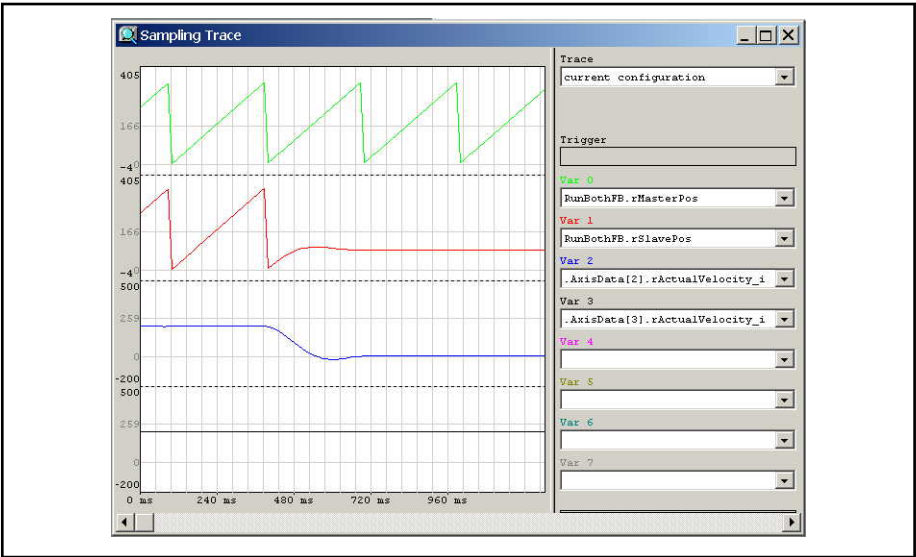


Abb. 4-13: LockOff-Profil mit einer LockOff_Pos von 90 Grad

Die obige Abbildung zeigt das Positions- und Geschwindigkeits-Profil während des LockOff-Profils wenn die LockOff-Position 90 Grad ist. Zu sehen ist, dass bei einer LockOff-Position von 90 Grad die Positions- und Geschwindigkeits-Profile ein Überschwingen haben. Dieses Positionsüberschwingen verursacht eine Rückwärtsbewegung.

Es wird empfohlen für den LockOff_Pos-Eingang einen Wert von 180 Grad zu verwenden (der Wert dieses Eingangs muss sowohl für den MB_PrepareCams als auch für den MB_CamLock Funktionsbaustein übereinstimmen).

Wird eine andere LockOff-Position als 180 Grad verwendet, kommt es, je weiter man von 180 Grad weg ist, zu einem Überschwingen in der Position und der Geschwindigkeit.

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle **F_RELATED_TABLE** folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#0001	16#0000	Spannung der Slave-Achse ist nicht auf AN
RESOURCE_ERROR	16#0003	16#0000	FB wurde durch einen anderen FB beendet
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Zustand der Zustandsmaschine
ACCESS_ERROR	16#0011	16#0000	Steuerung befindet sich nicht in der BB-Phase (Sercos P4)
INPUT_RANGE_ERROR	16#1301	16#0001	LockOff_Pos muss größer als 0 und kleiner 360 sein, Default=180 Grad
INPUT_RANGE_ERROR	16#1301	16#0002	InitialState liegt außerhalb des Bereiches 1...5
INPUT_RANGE_ERROR	16#1301	16#0003	RatioNumerator =< 0

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR	16#1301	16#0004	RatioDenominator =< 0
INPUT_RANGE_ERROR	16#1301	16#0005	SyncMode < 0 ODER SyncMode > 2
INPUT_RANGE_ERROR	16#1301	16#0006	LockOnCam < 1 OR LockOnCam > 98
INPUT_RANGE_ERROR	16#1301	16#0007	RunCam < 1 ODER RunCam > 98
INPUT_RANGE_ERROR	16#1301	16#0008	LockOffCam < 1 ODER LockOffCam > 98
INPUT_RANGE_ERROR	16#1301	16#0009	UserCam_Profile < 1 ODER UserCam_Profile > 98
INPUT_RANGE_ERROR	16#1301	16#0010	Die CamTable-Eingänge sind gleich
ACCESS_ERROR	16#1302	16#0003	Der Achsenmodulowert des Slaves beträgt nicht 360°
ACCESS_ERROR	16#1302	16#0004	Der Mastermodulowert des Slaves beträgt nicht 360°

Tab. 4-9: Fehlercodes Funktionsbaustein MB_CamLock

4.2.5 MB_CamLock Parametrierung

Der Funktionsbaustein [MB_CamLock](#), Seite 170, benötigt die folgende Parametrierung:

1. Die Checkbox "Interpolation im Antrieb" muss bei Hinzufügen von einer realen Achse in IndraWorks abgewählt werden, wenn diese Achse als Slave Achse des MB_CamLock verwendet wird.
2. Leit- und Folgeachse sind ins Moduloformat 360° zu setzen.
3. Der Antrieb ist vor dem Gebrauch des FB zu referenzieren (absolute Position bevorzugt).
4. Der FB [MB_PrepareCams](#), Seite 168, muss einmal ausgeführt werden, um die drei benötigten Kurvenscheiben zu berechnen und zu speichern.
 - LockOnCam, standardmäßig in CAM_TABLE_4 gespeichert
 - RunCam, standardmäßig in CAM_TABLE_3 gespeichert
 - LockOffCam, standardmäßig in CAM_TABLE_2 gespeichert
 - User_Cam_Profile, standardmäßig in CAM_TABLE_1 gespeichert
5. Die Leistung muss zum Antrieb zugeschaltet sein (Antrieb im Zustand "Enabled").
6. Der MB_CamLock enthält zeitkritische Motion -Funktionalität und sollte daher in der MotionTask aufgerufen werden.
7. Der FB arbeitet nur für eine Leitachse, die sich in positiver Richtung bewegt.



CamID 99 wird immer als Stopp-Kurvenscheibe verwendet.

4.3 Fliegende Säge

4.3.1 Einführung und Übersicht

Bei der Applikation "Fliegende Säge" wird eine Bearbeitungsstation auf ein kontinuierlich durchlaufendes Material (wie z.B. Metall, Kunststoff etc.) auf-synchronisiert und die Bearbeitung in dem Synchronlauf der Bearbeitungsstation ausgeführt.

Die Materialposition wird üblicherweise über ein Messrad erfasst und über einen optionalen Encodereingang an den Antriebsregler übertragen. Der An-

triebsregler synchronisiert dann den angeschlossenen Servomotor auf das kontinuierlich durchlaufende Material, wenn die vorgegebene Schnittlänge durchlaufen wurde.

Der Funktionsbaustein [ML_FlyingShear / MX_FlyingShear](#), Seite 178, übernimmt die Aufsynchronisierung und setzt den Ausgang "InSync", sobald die Bearbeitungsstation im Synchronlauf ist.

In diesem Zustand kann die Bearbeitung durchgeführt werden, da sich nun die Bearbeitungsstation synchron zum Material bewegt.

Nach Abschluss der Bearbeitung kann der Synchronlauf über den Eingang "MoveReturn" unterbrochen werden, und die Bearbeitungsstation fährt wieder in die vorgegebene Ausgangslage zurück.

Für Testzwecke kann auch (anstelle des Messrades) eine virtuelle Achse verwendet werden. Über diese virtuelle Achse ist es dann möglich das Messrad zu simulieren.



Für die Funktionalität "Fliegende Säge" gibt es für die Zielsysteme unterschiedliche Funktionsbausteine:

- Zielsystem MLD: Funktionsbaustein MX_FlyingShear
- Zielsystem MLC: Funktionsbaustein ML_FlyingShear

Die nachfolgende Beschreibung bezieht sich auf beide Funktionsbausteine.

Die systemspezifischen Einstellungen finden Sie in

- Zielsystem MLD: siehe [MX_FlyingShear - Besonderheiten für MLD](#), Seite 187
- Zielsystem MLC: siehe [ML_FlyingShear - Besonderheiten für MLC](#), Seite 186

4.3.2 ML_FlyingShear / MX_FlyingShear

Kurzbeschreibung

Der Funktionsbaustein ML_FlyingShear/MX_FlyingShear enthält den typischen Ablauf einer fliegenden Säge und führt die folgenden Schritte aus, sobald der Eingang "Start" TRUE ist:

- Folgeachse in die Ausgangslage "ReturnPos" fahren und warten, bis die Achse in Position ist
- Folgeachse über ein Lock-On-Kurvenscheibenprofil auf die Leitachse aufsynchronisieren
- Ausgang "InSync" setzen, sobald die Folgeachse (Trennstation) auf das Material aufsynchronisiert ist
- Folgeachse wieder in Ausgangslage positionieren, wenn am Eingang "MoveReturn" das entsprechende Signal anliegt



Der Betrieb des Funktionsbausteines erfordert besondere Randbedingungen (z.B. Interpolation im Antrieb, zyklischer Aufruf des FB in der Motion-Task...).

Beachten Sie bitte für das Zielsystem MLD die Inbetriebnahmehinweise in [MX_FlyingShear - Besonderheiten für MLD](#), Seite 187.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechMotion
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_Technology07.lib

Tab. 4-10: Referenztabelle ML_FlyingShear / MX_FlyingShear

Schnittstellenbeschreibung

ML_FlyingShear			
BOOL	Start	InSync	BOOL
BOOL	CropCut	InCycle	BOOL
BOOL	ImmediateCut	CropCutDone	BOOL
BOOL	MoveReturn	ImmediateCutDone	BOOL
BOOL	ResetCutCounter	PreSyncSignal	BOOL
REAL	CutLength	MaterialMoving	BOOL
REAL	MWFeedConst	ShortPrdWarning	BOOL
REAL	SyncDist	Reserve	DINT
REAL	ReturnPos	CycleState	UINT
REAL	ReturnVel	CutCounter	UINT
REAL	ReturnAcc	Error	BOOL
REAL	PreSyncPos	ErrorID	ERROR_CODE
TIME	PreSyncTime	ErrorIdent	ERROR_STRUCT
AXIS_REF	Master	Master	AXIS_REF
AXIS_REF	Slave	Slave	AXIS_REF
MB_FS_RETAIN_DATA	FSRetain	FSRetain	MB_FS_RETAIN_DATA

Abb. 4-14: Funktionsbaustein ML_FlyingShear / MX_FlyingShear

E/A-Typ	Name	Datentyp	Erläuterung
VAR_INPUT	Start	BOOL	Start=TRUE: Start des FlyingShear Funktionsbausteins. Je nach den Zuständen der Eingänge "ImmediateCut" und "CropCut" führt der Funktionsbaustein die folgenden Funktionen aus: Start=TRUE und positive Flanke am Eingang "ImmediateCut": Die Trennstation wird sofort aufsynchronisiert und führt die Schnittlänge (Cutlength) aus, sobald der Sofortschnitt (ImmediateCut) ausgeführt ist. Start=TRUE und positive Flanke am Eingang "CropCut": Die Trennstation wird aufsynchronisiert, nachdem die Schnittlänge (Cutlength) die Maschine passiert hat, und führt die Schnittlänge (Cutlength) aus, sobald der Kopfschnitt (CropCut) ausgeführt ist. Start=TRUE und kein Signal "ImmediateCut" und "CropCut": Die Trennstation wird aufsynchronisiert, nachdem die Schnittlänge (Cutlength), gemessen vom letzten Schnitt aus, die Maschine passiert hat
	CropCut	BOOL	Auf eine steigende Flanke hin beginnt die Folgeachse mit dem Aufsynchronisieren, nachdem die Schnittlänge die Maschine passiert hat. Befindet sich gerade ein Schneidezyklus in Ausführung (InCycle=TRUE), wird der Kopfschnitt (CropCut) im nächsten Schnittzyklus ausgeführt
	ImmediateCut	BOOL	Start = TRUE: Dieser Fall ist für einen Sofortschnitt bei laufendem Material vorgesehen. Der Sofortschnitt (ImmediateCut) wird durch eine positive Flanke ausgelöst. Die Folgeachse wird sofort aufsynchronisiert. Wird gerade ein Schnitt ausgeführt (InCycle = TRUE), erfolgt der Sofortschnitt im nächsten Schnittzyklus. Wenn (Start = FALSE) und ($V_{\text{measuring\_wheel}} < 5 \text{ U/min}$): Dieser Fall ist für einen Sofortschnitt bei Materialstillstand vorgesehen. Der Sofortschnitt (ImmediateCut) wird durch eine positive Flanke ausgelöst. Die Folgeachse bleibt im Stillstand und wird nicht synchronisiert. Der Ausgang "InSync" wird sofort gesetzt, und der Funktionsbaustein erwartet als Abschluss das Eingangssignal "MoveReturn". Nach Abschluss dieses Vorgangs ist die fliegende Säge auf das Material referenziert
	MoveReturn	BOOL	Beendet den Synchronlauf der Folgeachse mit dem Material und positioniert die Folgeachse wieder in der Ausgangslage "ReturnPos"
	ResetCutCounter	BOOL	Die positive Flanke setzt den Schnitzzähler (Cut Counter) zurück

E/A-Typ	Name	Datentyp	Erläuterung
	Cutlength	REAL	Die vorgegebene Materialschnittlänge* ¹ * ²
	MWFeedconst	REAL	Die Vorschubkonstante des Messrades pro Umdrehung * ¹ * ²
	SyncDist	REAL	Aufsynchronisierweg, den die Folgeachse zum Aufsynchronisieren auf die Leitachse benötigt* ¹ * ² . Je kürzer der Aufsynchronisierweg, desto höher die Aufsynchronisierbeschleunigung.
	ReturnPos	REAL	Die Folgeachse verfährt in diese Ausgangslage zu Beginn sowie nachdem sie aufsynchronisiert wurde und der Eingang MoveReturn den Zustand TRUE annimmt * ¹ * ²
	ReturnVel	REAL	Die Folgeachse verfährt mit der Geschwindigkeit ReturnVel in die Ausgangslage "ReturnPos" * ¹ * ²
	ReturnAcc	REAL	Die Folgeachse fährt mit der Beschleunigung ReturnAcc in die Ausgangslage "ReturnPos" * ¹ * ¹
	PreSyncPos	REAL	Das Ausgangssignal "PreSyncSignal" wird zur Vorbereitung der Synchronfahrt vor dem Synchronpunkt "PreSyncPos" mit der hier angegebenen Entfernung gesetzt (TRUE)* ¹
	PreSyncTime	TIME	Zeitdauer von "PreSyncSignal"
VAR_OUTPUT	InSync	BOOL	Die Folgeachse läuft synchron mit dem Material. Sie befindet sich im Synchronisationsfenster (siehe Antriebsparameter S-0-0228)
	InCycle	BOOL	Die Folgeachse führt gerade einen Schnittzyklus aus. Ein Sofort- oder Kopfschnitt wird erst im nächsten Zyklus ausgeführt (CutCycle = 0)
	CropCutDone	BOOL	Kopfschnitt ist ausgeführt
	ImmediateCutDone	BOOL	Sofortschnitt ist ausgeführt
	PreSyncSignal	BOOL	Vorsignal vor dem Synchronpunkt "PreSyncPos"
	ShortPrdWarning	BOOL	Die Folgeachse hat nicht genügend Zeit, um die Ausgangslage "ReturnPos" für den nächsten Schnitt zu erreichen. Die Rückfahrt wird unterbrochen, und die Aufsynchronisierrampe beginnt erneut. Dieser Fall sollte vermieden werden, da der Antrieb sonst überlastet wird und die Trennstation an das Wagenbetten- de fahren könnte. Gegenmaßnahmen: • Rückfahrgeschwindigkeit erhöhen (Funktionsbaustein-Eingang "ReturnVel") • Rückfahrbeschleunigung erhöhen (Funktionsbaustein-Eingang "ReturnAcc") • Aufsynchronisierweg verkürzen (Funktionsbaustein-Eingang "SyncDist") • Materialvorschubgeschwindigkeit verringern

E/A-Typ	Name	Datentyp	Erläuterung
	MaterialMoving	BOOL	Die Drehzahl des Material-Gebers ist höher als 5 U/min. Ein Sofortschnitt bei Materialstillstand ist nur zulässig, wenn MaterialMoving=FALSE
	Reserve	DINT	Reserve-Inkremente des Aufsynchronisiervorgangs zu Diagnosezwecken. Aufsynchronisierung nicht möglich, wenn Reserve ≤ 0 -> In diesem Fall gibt der FB eine Fehlermeldung aus
	CycleState	UINT	Aktueller Zustand des Schneidezyklus: 0: Stillstands- und Wartephase 1: Beschleunigungsphase 2: Synchronlaufphase 3: Rücklaufphase
	CutCounter	UINT	Mit jedem Schnitt wird der Schnitzzähler "CutCounter" erhöht. Durch (Start = FALSE) oder eine positive Flanke am Eingang "ResetCutCounter" wird der Zähler rückgesetzt
	Error	BOOL	Weist auf einen Fehler hin. Wird gelöscht wenn Start=FALSE
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 184
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Referenz zur Leitachse
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Referenz zur Folgeachse (Trennstation)
VAR_IN_OUT	FSRetain	MB_FS_RETAIN_DATA	Referenz auf die benötigten Retain-Daten dieses Funktionsbausteins MB_FS_RETAIN_DATA, Seite 186

*1:
*2:

Maßeinheiten je nach Antriebswichtung, z. B. mm
Neue Eingangswerte werden beim Übergang von der Synchronlauf- zur Rücklaufphase übernommen. Deshalb muss eine neue Schnittlänge, die im nächsten Zyklus aktiviert werden soll, zum Funktionsbaustein-Eingang kopiert werden, bevor die Achse in die Rücklaufposition gefahren wird.

Tab. 4-11: *Schnittstellenvariablen Funktionsbaustein ML_FlyingShear / MX_FlyingShear*

Signal-Zeit-Diagramm

Das nachstehende Diagramm zeigt den Signalablauf bei Sofortschnitt und stillstehendem Material (Start = FALSE):

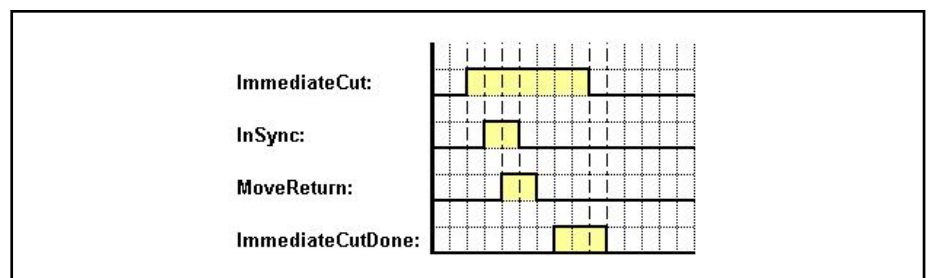


Abb. 4-15: Signal-Zeit-Diagramm: Sofortschnitt bei Materialstillstand

Das folgende Diagramm zeigt den kompletten Signalablauf des Funktionsbausteins "Fliegende Säge" bei kontinuierlich durchlaufendem Material und Sofortschnitt:

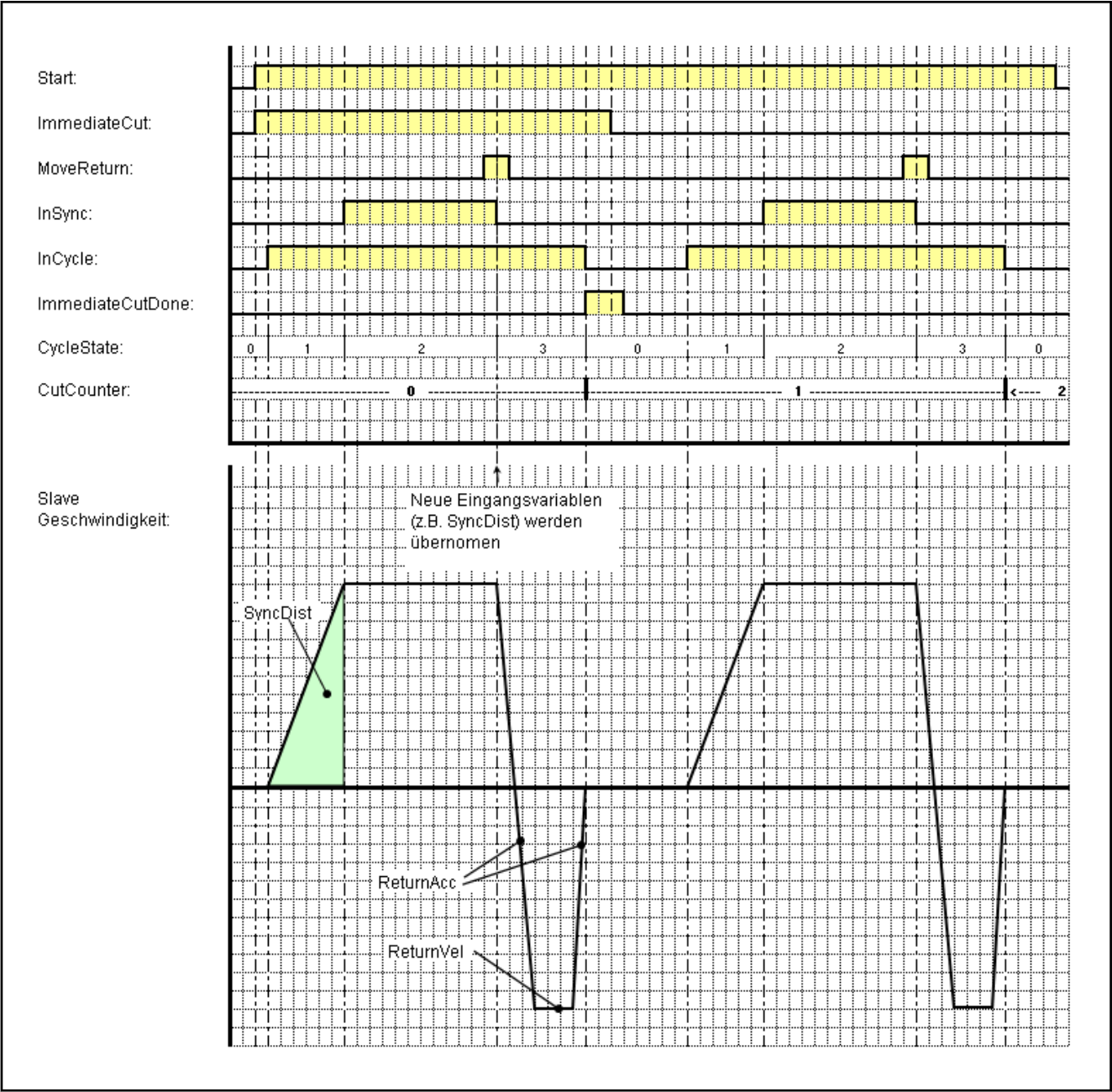


Abb. 4-16: Kompletter Signalablauf des Funktionsbausteins "Fliegende Säge" bei Sofortschnitt

Fehlerbehandlung Der Funktionsbaustein ML_FlyingShear / MX_FlyingShear erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb nicht freigeschaltet oder Antriebsfehler
ACCESS_ERROR (16#0004)	16#0003	16#0000	Funktionsbaustein wurde von einem anderen Funktionsbaustein unterbrochen
ACCESS_ERROR (16#0004)	16#0004	16#0000	Nicht unterstützte Antriebs-Firmware

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (Axis_Ref) wurde während der Abarbeitung des Funktionsbausteins verändert
INPUT_RANGE_ERROR (16#0006)	16#0201	16#0001	CutLenght <= 0
INPUT_RANGE_ERROR (16#0006)	16#0201	16#0002	MWFeedconst <= 0
INPUT_RANGE_ERROR (16#0006)	16#0201	16#0003	SyncDist <= 0
INPUT_RANGE_ERROR (16#0006)	16#0201	16#0004	ReturnAcc <= 0
INPUT_RANGE_ERROR (16#0006)	16#0201	16#0005	ReturnVel <= 0
SYSTEM_ERROR (16#7FFF)	16#0202	16#0000	Synchronpunkt ist zu weit entfernt. Daher ist kein Aufsynchronisieren möglich. Fehler rücksetzen und Sofortschnitt veranlassen
ACCESS_ERROR (16#0004)	16#0203	16#0001	S-0-0048 ist nicht im optionalen zyklischen MDT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0002	P-0-0054 ist nicht im optionalen zyklischen MDT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0003	P-0-0691 ist nicht im optionalen zyklischen MDT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0004	P-0-0692 ist nicht im optionalen zyklischen MDT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0005	P-0-0052 ist nicht im optionalen zyklischen AT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0006	P-0-0053 ist nicht im optionalen zyklischen AT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0007	P-0-0761 ist nicht im optionalen zyklischen AT-Kanal der Folgeachse konfiguriert
ACCESS_ERROR (16#0004)	16#0203	16#0008	P-0-1367 Bit 6 ist nicht TRUE (Struktur AxisData wird nicht aktualisiert)
OTHER_ERROR (16#00FE)	16#0204	16#0001	Das ELS-Konfigurationswort (P-0-0086) der Folgeachse ist falsch konfiguriert. Die Konfiguration muss wie folgt lauten: P-0-0086 = 2#xxx01000
OTHER_ERROR (16#00FE)	16#0204	16#0002	Parameter P-0-0060 der Folgeachse ist falsch konfiguriert. Der Wert muss lauten: P-0-0060=0
OTHER_ERROR (16#00FE)	16#0204	16#0003	Parameter P-0-0693 der Folgeachse ist falsch konfiguriert. Der Wert muss lauten: P-0-0693=0
OTHER_ERROR (16#00FE)	16#0204	16#0004	Parameter P-0-0750 der Folgeachse ist falsch konfiguriert. Der Wert muss lauten: P-0-0750=0
OTHER_ERROR (16#00FE)	16#0204	16#0005	Parameter P-0-0090 der Folgeachse ist falsch konfiguriert. Bit 2 von P-0-0090 muss FALSE sein

ErrorID	Additional1	Additional2	Beschreibung
SYSTEM_ERROR (16#7FFF)	16#0205	16#0000	Sofortschnitt (im Stillstand) wurde angefordert, obwohl sich das Material weiterbewegte
OTHER_ERROR (16#00FE)	16#0206	16#0000	Der Parameter A-0-0200 der Slaveachse entspricht nicht der logischen Achsnummer der Masterachse

Tab. 4-12: Fehlercodes Funktionsbaustein ML_FlyingShear / MX_FlyingShear

4.3.3 MB_FS_RETAIN_DATA

Die Struktur MB_FS_RETAIN_DATA enthält die für den Baustein ML_FlyingShear / MX_FlyingShear notwendigen Retain-Daten.

Typvereinbarung

```

TYPE MB_FS_RETAIN_DATA :
STRUCT
  bCutNotCompleted : BOOL;      // Der letzte Schnitt war nicht vollständig
  diMasterSyncPosition : DINT;  // Synchronisationsposition für Folgeachse
  iRevCounter: INT;             // Leitachs-Umdrehungszähler [-2048...2047]
END_STRUCT
END_TYPE

```

4.3.4 ML_FlyingShear - Besonderheiten für MLC

Notwendige Hardware

- IndraDrive C oder M
- Zusätzliche Encoder-Interface-Karte zum Einlesen der Messradposition, falls Messgeber verwendet werden soll (kann entfallen, falls virtuelle oder reale Achse verwendet wird)
- Zusätzlicher Encoder zum Einlesen der Messradposition (gemäß IndraDrive Projektierung), falls Messgeber verwendet werden soll (kann entfallen, falls virtuelle oder reale Achse verwendet wird)



Ein hochauflösender Messgeber mit einer Auflösung von min. 4096 Inkremente/Umdrehung (möglichst Sinus-Encoder) wird empfohlen. Grob aufgelöste Gebersignale verursachen geringe Schnittgenauigkeit, sowie ggf. Laufgeräusche im Synchronbetrieb.

Notwendige Firmware

- Antriebs-Firmware MPX07 oder höher
- Die folgenden Funktionspakete müssen im Antrieb freigeschaltet sein
 - Closed Loop
 - Synchronization

Notwendige Parametrierung

Die nachfolgende Parametrierung muss am Antrieb vorgenommen werden, um eine ordnungsgemäße Funktion des Funktionsbausteins "ML_FlyingShear" sicherzustellen.

1. Anlegen der Slave-Achse in IndraWorks
Der Funktionsbaustein benötigt die Lagesollwertgenerierung im Antrieb. Daher darf beim Anlegen der Slave-Achse in IndraWorks die Checkbox "Interpolation im Antrieb" nicht abgewählt werden.
2. Laden des Parameterfiles "FlyingShearSettings.par"
Mit dem Laden des Parameterfiles "FlyingShearSettings.par" werden automatisch auf der Fliegenden Säge Achse wichtige Konfigurationen und die Kurvenscheiben geladen. Das Parameterfile wird über die

Indraworks MLC Installations-CD 3 im Pfad "Add ons\ Tech FB_Parameterfile" bereitgestellt.

Das Parameterfile enthält folgende Informationen:

- P-72 Lock On Cam Poly 5
- P-92 Run Cam
- P-750=1
- P-693=0
- P-60=0
- P-86=0x0000.0000.0000.1000
- P-90=0x0000.0000.0000.0000
- A-500=P-691
- A-502=P-692

3. Parametrierung Defaultleitachse

Im Parameter A-0-0200 der FlyingShear Slave Achse die logische Achsnummer der Masterachse eintragen

Über den IndraWorks Parametereditor den Parameter A-0-0200 der Slave Achse mit der logischen Achsnummer der Masterachse beschreiben.

4. Parametrierung der Fliegenden Säge Achse

Parametrierung der Fliegenden Säge gemäß Anwendung.

Wichtung sollte linear, absolut eingestellt werden.

5. Parametrieren des optionalen Encoders

IndraWorks ► Hinzufügen einer Geberachse in das Projekt

Die Wichtung sollte rotatorisch Modulo 360° eingestellt werden.

Einstellen der Encoder-Parameter gemäß Encoderdaten

6. Sicherstellen der Encoder Bewegungsrichtung

Der optionale Encoder (Messrad) muss sich im Normalbetrieb in positive Richtung bewegen. Die Bewegungsrichtung kann über den IndraWorks-Dialog der Geberachse beobachtet werden. Der Bewegungssinn muss invertiert werden, falls sich der Geber in negative Richtung bewegt.

7. Aufsynchonisierbeschleunigung (P-0-0142) und Aufsynchonisiergeschwindigkeit (P-0-0143) mit möglichst großen Werten parametrieren um einen dynamischen Aufsynchonisiervorgang sicherzustellen

IndraWorks ► Parametereditor.

8. Synchronlauffenster (S-0-0228) möglichst klein parametrieren, um eine möglichst hohe Schnittgenauigkeit zu erreichen

IndraWorks ► Parametereditor.

Notwendige IndraLogic-Schritte

- Der Funktionsbaustein "ML_FlyingShear" enthält hochpriore Motion-Befehle und sollte deshalb in einer hochprioren zyklischen SPS-Task mit einer Zykluszeit $\leq 4\text{ms}$ zyklisch aufgerufen werden (MotionTask)
- Vor Start des FB muss der MC_Power fehlerfrei abgearbeitet werden

4.3.5 MX_FlyingShear - Besonderheiten für MLD

Notwendige Hardware

- IndraDrive C oder M mit Advanced Steuerteil

- Zusätzliche Encoder-Interface-Karte zum Einlesen der Messradposition (kann entfallen, falls virtueller Leitachsgenerator oder reale Achse als Leitachse verwendet wird)
- Zusätzlicher Encoder zum Einlesen der Messradposition (gemäß IndraDrive Projektierung), falls Messgeber verwendet werden soll (kann entfallen, falls virtueller Leitachsgenerator oder reale Achse als Leitachse verwendet wird)



Ein hochauflösender Messgeber mit einer Auflösung von min. 4096 Inkremente/Umdrehung (möglichst Sinus-Encoder) wird empfohlen. Grob aufgelöste Gebersignale verursachen geringe Schnittgenauigkeit, sowie ggf. Laufgeräusche im Synchronbetrieb.

Notwendige Firmware

- Antriebs-Firmware MPx07 oder höher
- Die folgenden Funktionspakete müssen im Antrieb freigeschaltet sein:
 - Closed Loop
 - Synchronisation

Notwendige Parametrierung

Die nachfolgende Parametrierung muss am Antrieb vorgenommen werden, um eine ordnungsgemäße Funktion des Funktionsbausteins "MX_FlyingShear" sicherzustellen.



Die unten genannten Parameter-Dateien sind im "AddOn"-Verzeichnis der IndraWorks-MLD Installation CD im Ordners "Parameterfiles" zu finden.

1. Laden Sie die Parameterdatei **FS_Settings_Local_MPH07.par** falls der MX_FlyingShear FB auf die lokale MLD-Achse wirkt. Die nachfolgenden Schritte 2 und 3 entfallen in diesem Fall.
2. Laden Sie die Parameterdatei **FS_Settings_MLDM_Slave_MPH07.par** auf die MLD-M Slaveachse, falls der MX_FlyingShear FB auf diese Achse wirken soll.

Folgende Parameter werden damit eingestellt:

- P-0-0072 ⇒ Aufsynchronisierungskurvenscheibe (Polynom 5. Ordnung)
- P-0-0092 ⇒ 1:1 Kurvenscheibe
- P-0-0765 ⇒ 0: Modulofaktor Messgeber
- P-0-0750 ⇒ 0: Leitachsumdrehungen je Leitachszyklus
- P-0-0693 ⇒ 0: Filterzeitkonstante Leitachs-Versatz
- P-0-0060 ⇒ 0: Filterzeitkonstante additiver Lagesollwert
- P-0-0086 ⇒ 0000.0000.0000.1000: Konfigurationswort Synchronbetriebsart
- P-0-0918 ⇒ 104.8576: Wegstrecke virtuelle Leitachse
- P-0-0916 ⇒ P-0-0758: Signalauswahl Leitwert-Umwandler
- P-0-0917 ⇒ 0000.0000.0000.0001: Steuerwort virtueller Leitwertgenerator
- P-0-0756 ⇒ 0000.0000.0000.0010: Wichtung virtueller Leitwertgenerator
- P-0-0142 ⇒ 10000: Dynamische Synchronisationsbeschleunigung
- P-0-0143 ⇒ 40000: Dynamische Synchronisationsgeschwindigkeit
- S-0-0048 ⇒ 0: Additiver Lagesollwert

3. Laden Sie die Parameterdatei **FS_Settings_MLDM_Master_MPH07.par** in die MLD-M Masterachse, sofern der MX_FlyingShear auf die **erste** MLD-M Slaveachse wirken soll.

Die Parameterdatei enthält die folgenden Einstellungen:

- P-0-1367 ⇒ 0000.0010.0101.0000: SPS-Konfigurationswort
- P-0-1623 ⇒ Konfiguration Leitachs-Sollwerte
- P-0-1624 ⇒ Konfiguration Leitachs-Istwerte
- P-0-1625 ⇒ Konfiguration Folgeachs-Sollwerte
- P-0-1626 ⇒ Konfiguration Folgeachs-Istwerte
- P-0-0918 ⇒ 104.8576: Wegstrecke virtuelle Leitachse
- P-0-0916 ⇒ P-0-0758: Signalauswahl Leitwert-Umwandler
- P-0-0917 ⇒ 0000.0000.0000.0001: Steuerwort virtueller Leitwert-generator
- P-0-0756 ⇒ 0000.0000.0000.0010: Wichtung virtueller Leitwert-generator



Die **FS_Settings_MLDM_Master_MPH07.par** Parameterdatei enthält die Voreinstellung, dass die MLD-M Masterachse über den FlyingShear-Funktionsblock die **erste** MLD-M Slaveachse ansteuert. Passen Sie die Konfiguration der optionalen zyklischen Daten an, wenn eine andere MLD-M Slaveachse angesteuert werden soll.

Die folgenden optionalen zyklischen MDT-Daten (Sollwerte) werden benötigt und müssen auf der MLD-M Masterachse entsprechend konfiguriert werden:

- P-0-0692 ⇒ Leitachsposition additiv, Prozessregler
- P-0-0691 ⇒ Lagesollwert additiv, Prozessregler

Die folgenden optionalen zyklischen AT-Daten (Istwerte von der MLD-Slaveachse zur Masterachse) werden benötigt und müssen auf der MLD-M Masterachse entsprechend konfiguriert werden:

- P-0-0052 ⇒ Lageistwert Messgeber
- P-0-0053 ⇒ Leitachsposition
- P-0-0761 ⇒ Leitachsposition für Folgeachse

Die Einstellungen können in IndraWorks durch einen Klick mit der rechten Maustaste auf den Sercos III Knoten und Auswahl **Cross Communication Drive: Prozessdaten, Sollwerte** für MDT und **Cross Communication Drive: Prozessdaten, Istwerte** für die AT-Konfiguration vorgenommen werden.

4. Konfigurieren Sie die mechanischen Achseinstellungen der Fliegenden Säge (lineare Achse, kein Modulo, Verfahrbereich, Bereichsgrenzen usw.) der gegebenen Applikation entsprechend.
5. Konfigurieren Sie den zusätzlichen zweiten Geber/Messgeber.
 - Schalten Sie IndraWorks auf Onlinebetrieb und öffnen Sie den Achsdialog **Mess-Systeme ▶ Optionaler Geber**.
6. Stellen Sie sicher, dass sich der Messgeber in die positive Richtung bewegt, da der Aufsynchonisiermechanismus nur für die positive Richtung ausgelegt ist.

- Schalten Sie IndraWorks auf Onlinebetrieb, und öffnen Sie den Achsdialog **Messgeber**

Beobachten Sie die Position des Messgebers, während Sie den Geber in Materialrichtung bewegen. Die Geberposition muss sich in positiver Richtung verändern. Die Geberposition wird nur in den Antriebszuständen bb/Ab, AH und AF aktualisiert.



Werden im Antrieb Software-Begrenzungsschalter aktiviert, muss als Reaktion "Fehler" konfiguriert werden. Die Einstellung "Warnungsmeldung" kann zur Folge haben, dass der Synchronisationsvorgang nach einem Achsstillstand erneut ausgelöst wird.

Relevante Parameter für MX_FlyingShear

Folgende Parameter sind für den Betrieb des Funktionsbausteines MX_FlyingShear relevant.

Parameter-IDN	Parametername	Beschreibung
P-0-0750	Leitachsumdrehungen pro Leitachszyklus	Gibt den Modulobereich der Leitachse an. Verwenden Sie immer P-0-0750 = 0
P-0-0765	Modulofaktor Messgeber	Gibt den Modulobereich des Messgebers an. Verwenden Sie immer P-0-0765 = 0
P-0-0329	Lageistwert 3 Glättung Messgeber	Dies ist die Filterzeitkonstante des Messgeberfilters. Dieser Filter reduziert Störsignale wenn die Fliegende Säge Achse synchron zum Material läuft
P-0-0142	Aufsynchronisierbeschleunigung	Dieser Parameter legt die Beschleunigung fest, die anzuwenden ist, wenn die Folgeachse nicht genügend Zeit hat, um die Ausgangslage "ReturnPos" für den nächsten Schnitt zu erreichen. Die Folgeachse unterbricht die Rückfahrt und beschleunigt sich diesem Wert entsprechend für den nächsten Schnitt. Die Beschleunigung sollte so hoch wie möglich gewählt werden
P-0-0143	Aufsynchronisiergeschwindigkeit	Dieser Parameter legt die Geschwindigkeit fest, die anzuwenden ist, wenn die Folgeachse nicht genügend Zeit hat, um die Ausgangslage "ReturnPos" für den nächsten Schnitt zu erreichen. Die Folgeachse unterbricht die Rückfahrt und beschleunigt sich diesem Wert entsprechend für den nächsten Schnitt. Die Geschwindigkeit sollte so hoch wie möglich gewählt werden
S-0-0228	Synchronlauffenster Lage	Der Funktionsbaustein MX_FlyingShear setzt den Ausgang InSync = TRUE, sobald der Aufsynchronisierweg (Sync-Dist) abgefahren wurde und sich die Achs-Istposition im Synchronlauffenster befindet. Das Synchronlauffenster sollte klein bemessen sein, um eine hohe Schnittgenauigkeit zu gewährleisten

Parameter-IDN	Parametername	Beschreibung
P-0-0694	Feinabgleich Getriebeübersetzung, Prozessregler	Dieser Getriebe-Feinabgleich bewirkt, dass sich die Achse der Fliegenden Säge schneller bewegt als das Material, wenn P-0-0694 > 0% ist. Ist P-0-0694 < 0%, bewirkt der Getriebe-Feinabgleich, dass sich die Achse der Fliegenden Säge langsamer bewegt als das Material. Ein Wert von <> 0% verursacht eine zusätzliche Schnittgenauigkeit
S-0-0193	Positionierungsruck	Dieser Parameter begrenzt die Beschleunigungsänderung (Ruck) in der Rückkehrbewegung. S-0-0193=0 deaktiviert die Ruckbegrenzung

Tab. 4-13: Relevante Parameter für MX_FlyingShear

4.4 MagicBelt

4.4.1 Einführung und Überblick

Bei der "MagicBelt"-Anwendung werden Produkte aus einem nahezu kontinuierlichen Produktstrom gruppiert und typischerweise einer Zweitverpackung (wie z. B. Karton) zugeführt. "MagicBelt"-Anwendungen sind auch unter den Begriffen Zweizugkette, DualBelt, MagicChain bekannt. Jede Kette wird bei dieser Anwendung über eine eigene Instanz des nachfolgend beschriebenen Funktionsbausteines MB_MagicBeltType01 angesteuert. Die Funktionsbausteine kommunizieren untereinander durch die angelegten Datenstrukturen.

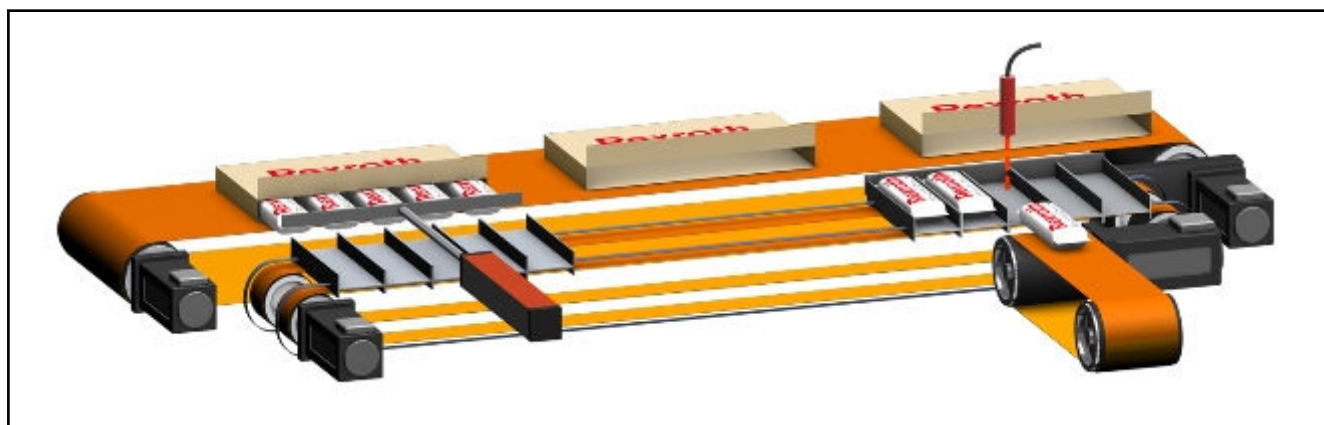


Abb. 4-17: MagicBelt Applikation

Begriffsdefinitionen

Die nachfolgend verwendeten Begriffe sind wie folgt definiert:

- **Kette:** Die Kette (Chain) wird durch die Servomotoren angetrieben. Eine Kette kann bis zu 4 Züge enthalten
- **Zug:** Der Zug (Train) dient zum Gruppieren der Produkte und beinhaltet mehrere Fächer
- **Fach:** Ein Fach (Pocket) kann ein oder mehrere Produkte beinhalten und ist ein Teil eines Zuges
- **InfeedMaster:** Antriebs-Achse (oder Encoder) des Synchronrades bei Synchronradzuführung
- **InfeedAxis:** Intern genutzte virtuelle Achse um Infeed-Bewegungen auszuführen

Definitionen und Randbedingungen

In der Praxis werden oftmals mehrere Züge pro Kette eingesetzt, da dies zu kürzeren Verfahrbewegungen und Geschwindigkeiten führt. Folglich kann da-

mit ein "kleinerer" Motor eingesetzt werden, und der mechanische Verschleiß von Kette und Motor wird reduziert.

Sind mehrere Züge auf einer Kette angebracht wird im Folgenden davon ausgegangen dass:

- der Abstand von Zuganfang zum nächsten Zuganfang gleich ist (symmetrische Zugverteilung)
- alle Züge einer Kette den gleichen Fächeraufbau haben (z. B. Anzahl und Größe der Fächer)
- alle Ketten mit der gleichen Anzahl von Zügen bestückt sind

Der Modulowert eines Ketten-Antriebes wird damit durch den Abstand von Zug-Vorderkante zu der nächsten Zug-Vorderkante definiert (translatorische Einheit). Befindet sich z. B. nur ein Zug auf der Kette, dann ist der Modulowert gleich der Kettenlänge.

Der Modulowert wird damit über die folgende Gleichung festgelegt:

$$\text{ModuloValue} = \frac{\text{Chain Length}}{\text{ZugsPerChain}}$$

Abb. 4-18:

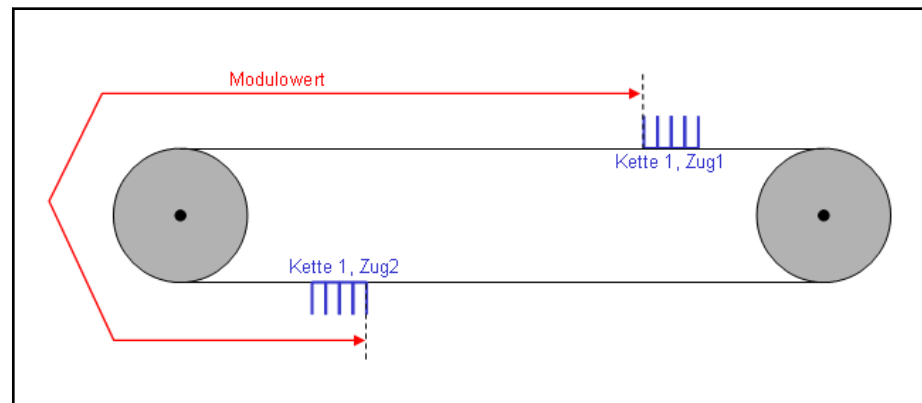


Abb. 4-19: MagicBelt Definitionen Modulowert

4.4.2 MB_MagicBeltType01

Kurzbeschreibung

Der Funktionsbaustein enthält Funktionalitäten zur Ansteuerung einer Kette einer MagicBelt Applikation.

Je nach angewählter Betriebsart wird die jeweilige Kette entsprechend angesteuert:

- **Jog-Mode, MagicBelt, Seite 214:** Die Züge können einzeln oder synchron wahlweise kontinuierlich oder inkrementell vertippt werden
- **Tune-Mode, MagicBelt, Seite 214:** Um Dynamikgrenzwerte (z. B. maximale Beschleunigung) zu ermitteln, werden die Züge synchronisiert und fortlaufend um eine einstellbare Distanz verfahren
- **Automatik-Mode, MagicBelt, Seite 215:** Im Wechselspiel fahren die Züge jeweils die Be- und Entladeposition(en) an



Der Betrieb des Funktionsbausteins erfordert besondere Randbedingungen (z. B. Modulo-Parametrierung, zyklischer Aufruf des FB in der Motion-Task...).

Bitte beachten Sie daher [Inbetriebnahmehinweise, MagicBelt, Seite 229](#).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechMotion

Tab. 4-14: Referenztabelle MB_MagicBeltType01

Leistungsmerkmale

- Maximal können jeweils 3 Ketten verwaltet werden (über 3 FB-Instanzen für Kette A, B, C)
- Auf einer Kette können maximal 4 Züge mit gleichen Abständen angeordnet sein
- Es können maximal 40 Fächer mit **unterschiedlichen** Fächerbreiten pro Zug verwendet werden. Bei mehreren **gleichlangen** Fächern vervielfacht sich die maximale Fächer-Anzahl entsprechend (siehe auch Strukturelement MB_MBLET_INFEED.InfeedStepData)
- Getaktetes und synchronisiertes Beladen wird unterstützt
- Es können maximal 10 verschiedene Entladepositionen angefahren werden

Schnittstellenbeschreibung

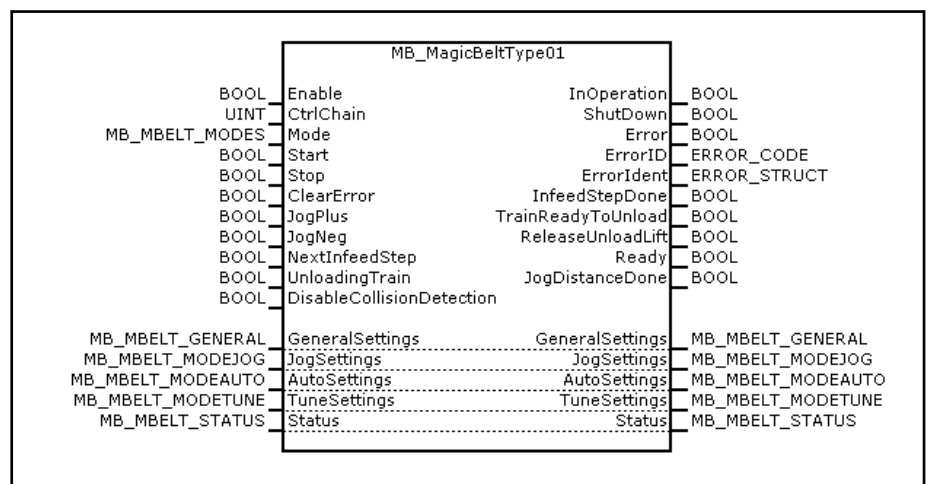


Abb. 4-20: Funktionsbaustein MB_MagicBeltType01

Über die Funktionsbaustein-Eingänge (z. B. Start, Stop...) werden die Ketten von Anwendungsprogramm angesteuert. Die Konfigurationsdaten für den Betrieb werden über die VAR_IN_OUT-Datenstrukturen (z. B. AutoSettings...) angelegt.

Über eine weitere "Status"-VAR_IN_OUT Datenstruktur stehen zusätzliche Diagnose-Informationen zur Verfügung.

I/O-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins, zustandsgesteuert
	CtrlChain	UINT	Wählt die anzusteuern Kettensnummer aus: 1: Kette A, 2: Kette B, 3: Kette C
	Mode	MB_MBELT_MODES	Betriebsarten-Anwahl. Ein Wechsel zwischen den Betriebsarten wird nur im "Stopped"-Zustand ausgeführt MB_MBELT_MODES, Seite 200
	Start	BOOL	Positive Flanke von "Start" startet die Achsbewegung (siehe auch StartType)

I/O-Typ	Name	Datentyp	Beschreibung
	Stop	BOOL	Positive Flanke von "Stop" stoppt die Achsbewegung (siehe auch StopType)
	ClearError	BOOL	Positive Flanke löscht einen anstehenden Fehler
	JogPlus	BOOL	Vertippen der Achsen in positive Richtung (nur bei aktivem Jog-Mode)
	JogNeg	BOOL	Vertippen der Achsen in negative Richtung (nur bei aktivem Jog-Mode)
	NextInfeedStep	BOOL	Bei Infeed-Typ "FEED_INDEXED": Nach positiver Flanke und Ablauf der Zeit "StartDelayTime" wird der Zug zum nächsten Fach weitergetaktet Bei Infeed-Typ "FEED_SYNCHRONIZED": Eine positive Flanke innerhalb des Erwartungsfensters ("ProductInExpectationWindow" = TRUE) signalisiert ein Produkt im Fach des Eintaktrades "InfeedMaster"
	UnloadingTrain	BOOL	Der Eingang muss beim Entladen des Zuges solange gesetzt bleiben, bis der Entladevorgang abgeschlossen ist. Der Zug verbleibt solange an der Entladeposition wie dieses Signal gesetzt ist. Mit negativer Flanke wird der Zug zur nächsten Ent- bzw. Beladeposition weitergetaktet
	DisableCollisionDetection	BOOL	Schaltet die Kollisionsüberwachung ab, um z. B. nach einer Kollision die Ketten auseinander zu fahren
VAR_OUTPUT	InOperation	BOOL	Fehlerfreier Betrieb, Ausgangsdaten sind gültig
	ShutDown	BOOL	Definiertes Stillsetzen nach negativer Flanke an "Enable"
	Error	BOOL	Signalisiert fehlerhaften Zustand
	ErrorID	ERROR_CODE	Fehler Diagnose
	ErrorIdent	ERROR_STRUCT	Detail Fehler Diagnose Fehlercodes, Seite 197
	InfeedStepDone	BOOL	Ausgabe nur bei FB der Kette A (CtrlChain = 1 ChainA). Ausgang ist nur für die Infeed-Variante „FEED_INDEXED“ relevant. Der Ausgang wird gesetzt wenn der "InfeedStep" ausgeführt wurde. Mit positiver Flanke von "NextInfeedStep" wird der Ausgang zurückgesetzt
	TrainReadyToUnload	BOOL	Zug steht an der Entladeposition und ist bereit zur Entladung
	ReleaseUnloadLift	BOOL	Freigabe des Entlade-Rechens

I/O-Typ	Name	Datentyp	Beschreibung
	Ready	BOOL	Wird nur bei FB mit "CtrlChain" = 1 (ChainA) ausgegeben: Der Ausgang wird gesetzt wenn der einfließende Produktstrom vom MagicBelt aufgenommen werden kann, siehe auch Produktstau-Überwachung
	JogDistanceDone	BOOL	Der Ausgang wird gesetzt wenn bei inkrementellem Joggen die vorgegebene Distanz zurückgelegt wurde. Der Ausgang wird zurückgesetzt wenn "Jog-Plus"/"JogNeg" zurückgesetzt wird
VAR_IN_OUT	GeneralSettings	MB_MBELT_GENERAL	Allgemeine Konfiguration der Magic Belt Anwendung MB_MBELT_GENERAL, Seite 200
	JogSettings	MB_MBELT_MODEJOG	Einstellungen für den Jog-Mode MB_MBELT_MODEJOG, Seite 202
	AutoSettings	MB_MBELT_MODEAUTO	Einstellungen für den Auto-Mode MB_MBELT_MODEAUTO, Seite 203
	TuneSettings	MB_MBELT_MODETUNE	Einstellungen für den Tune-Mode MB_MBELT_MODETUNE, Seite 202
	Status	MB_MBELT_STATUS	Statusinformationen zu den Ketten MB_MBELT_STATUS, Seite 211

Tab. 4-15: Schnittstellenvariablen Funktionsbaustein MB_MagicBeltType01

Min./Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet:

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
CtrlChain	UINT	1	3	1	Pos. Flanke von "Enable"
Mode	MB_MBELT_MODES	MODE_AUTO	MODE_JOG	MODE_AUTO	Kontinuierlich
Start	BOOL			FALSE	Kontinuierlich im Auto- und Tune-Mode
Stop	BOOL			FALSE	Kontinuierlich im Auto- und Tune-Mode
ClearError	BOOL			FALSE	Kontinuierlich im Error-Mode
JogPlus	BOOL			FALSE	Kontinuierlich im Jog-Mode
JogNeg	BOOL			FALSE	Kontinuierlich im Jog-Mode

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
NextInfeedStep	BOOL			FALSE	Kontinuierlich im Auto-Mode
UnloadingTrain	BOOL			FALSE	Kontinuierlich im Auto-Mode
DisableCollision-Detection	BOOL			FALSE	Kontinuierlich

Tab. 4-16: Min-/Max-/Default-Werte Funktionsbaustein MB_MagicBeltType01

Verwendete Datenstrukturen**Verwendete Datenstrukturen**

Struktur // Unterstruktur	Verwendungszweck
MB_MBELT_MODES, Seite 200	Betriebsarten des Magic Belt-Funktionsbausteins
MB_MBELT_GENERAL, Seite 200	Betriebsartenübergreifende Einstellungen (z.B. Kettenlänge)
MB_MBELT_MODEJOG, Seite 202	Einstellungen für die Jog-Betriebsart (z.B. Jog Geschwindigkeit)
MB_MBELT_MODETUNE, Seite 202	Einstellungen für die Tune-Betriebsart (z.B. Tune Geschwindigkeit)
MB_MBELT_MODEAUTO, Seite 203	Einstellungen für die Auto-Betriebsart (z.B. Tune Geschwindigkeit)
MB_MBELT_STARTTYPE, Seite 204	Start-Typ (z.B. START_AT_BEGIN)
MB_MBELT_STOPTYPE, Seite 204	Stop-Typ (z.B. STOP_IMMEDIATE)
MB_MBELT_INFEED, Seite 204	Einstellungen zum Beladen in Auto-Betriebsart
MB_MBELT_FEEDTYPE, Seite 207	Beladeverfahren (getaktet, synchronisiert, mit Geschwindigkeit)
MB_MBELT_INFEEDSTEPS, Seite 207	Kettenspezifische Daten für das Beladen im Automatik-Mode
MB_MBELT_OUTFEED, Seite 209	Einstellungen zum Entladen in Auto-Betriebsart
MB_MBELT_MOVE, Seite 210	Einstellungen zum Fahren zwischen Infeed ↔ Outfeed in Auto-Betriebsart
MB_MBELT_STATUS, Seite 211	Kettenübergreifende Statusinformationen

Funktionsbeschreibung

Mit "Enable"=TRUE startet der Funktionsbaustein und die am "Mode"-Eingang angewählte Betriebsart wird nach der Initialisierung aktiv. Entsprechend der angewählten Betriebsart wird die Kette angesteuert. Ein Wechsel zwischen den Betriebsarten ist nur im sicheren "Stopped"-Zustand erlaubt, wobei die Ketten im Stillstand sind. Die Betriebsarten und deren Funktionalität werden nachfolgend beschrieben.

Folgende Eingänge sind betriebsartenunabhängig und wirken immer:

- Enable
- Mode
- CtrlChain
- ClearError
- DisableCollisionDetection

Folgende Ausgänge sind betriebsartenunabhängig und wirken immer:

- InOperation
- Error
- ErrorID
- ErrorIdent

- Shutdown

Folgende Ein/Ausgänge sind betriebsartenunabhängig und wirken immer:

- GeneralSettings
- Status



Alle Chain-Achsen müssen bei Start des FB (über Enable=TRUE) referenziert sein. Ist diese Bedingung nicht erfüllt, meldet der FB einen Fehler.

Fehlercodes Die folgenden Fehlercodes der "F_RELATED_TABLE" werden vom Funktionsbaustein generiert:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR	16#0001	16#0000	Keine Leitung am Kettenantrieb zugeschaltet
RESOURCE_ERROR	16#0003	16#0000	FB wurde von einem anderen FB unterbrochen
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Zustand der internen Zustandsmaschine
RESOURCE_ERROR	16#0011	16#0000	Steuerung ist nicht betriebsbereit
INPUT_INVALID_ERROR	16#1A00	16#0001	Die verbleibende Positionierzeit ist ≤ 0 . Ursache: InfeedLoadTime ist größer als die Zeit, die sich aus der MaxProductRate ergibt. Lösung: InfeedLoadTime und/oder MaxProductRate reduzieren.
INPUT_INVALID_ERROR	16#1A00	16#0002	Products per pocket = 0
INPUT_INVALID_ERROR	16#1A00	16#0003	Feedrate override ≤ 0 OR > 100
INPUT_INVALID_ERROR	16#1A00	16#0004	CtrlChain ≤ 0 OR > 3
INPUT_INVALID_ERROR	16#1A00	16#0005	ChainLength ≤ 0
INPUT_INVALID_ERROR	16#1A00	16#0006	NoOfChains ≤ 0 OR > 3
INPUT_INVALID_ERROR	16#1A00	16#0007	TrainsPerChain ≤ 0 OR > 3
INPUT_INVALID_ERROR	16#1A00	16#0008	ProductBuffer ≤ 0
INPUT_INVALID_ERROR	16#1A00	16#0009	InfeedAxis AxisRef.AxisNo ist außerhalb des gültigen Wertebereiches
INPUT_INVALID_ERROR	16#1A00	16#000A	ChainAxisRef ist außerhalb des gültigen Wertebereiches
INPUT_INVALID_ERROR	16#1A00	16#000B	StopRamp zu klein. StopRamp muss größer oder gleich als alle anderen vorgegebenen Beschleunigungen sein
INPUT_INVALID_ERROR	16#1A00	16#000C	InfeedType ist ungültig
INPUT_INVALID_ERROR	16#1A00	16#000D	FirstInfeedPosition ist außerhalb gültigem Bereich
INPUT_INVALID_ERROR	16#1A00	16#000E	BeginReducedVel1 ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#000F	EndReducedVel1 ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0010	BeginReducedVel2 ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0011	EndReducedVel2 ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0012	MaxProductRate ist außerhalb des gültigen Bereiches

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#1A00	16#0013	MinDistanceToOutfeed ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0014	MaxInfeedVelocity ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0015	MaxInfeedAcceleration ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0016	MaxInfeedJerk ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0017	StartDelayTime ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0018	InfeedLoadTime ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0019	InfeedMotionLaw ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#001A	ProdExpectationWindowStart ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#001B	ProdExpectationWindowEnd ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#001C	AdditionalSteps ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#001D	InfeedStepData ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#001E	FirstOutfeedPosition ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#001F	MaxOutfeedVelocity ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0020	MaxOutfeedAcceleration ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0021	MaxOutfeedJerk ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0022	OutfeedDistances ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0023	MaxMoveVelocity ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0024	MaxMoveAcceleration ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0025	MaxMoveJerk ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0026	LoadTrainAcceleration ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0027	ReducedVelocity ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0028	JogAxisSelection ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0029	JogVelocity ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#002A	JogAcceleration ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#002B	JogDistance ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#002C	TrainDistance ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#002D	SafetyDistance ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#002E	Mode-Eingang außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#002F	Erwartungsfenster Ende "ProdExpectationWindowEnd" ist zu groß
INPUT_INVALID_ERROR	16#1A00	16#0030	InfeedMaster AxisRef ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0031	CatchUpAcceleration ist außerhalb des gültigen Bereiches
INPUT_INVALID_ERROR	16#1A00	16#0032	Infeed Schritt übersteigt die maximale Beschleunigung "MaxInfeedAcceleration"
INPUT_INVALID_ERROR	16#1A00	16#0033	LoadTrainDeceleration ist außerhalb des gültigen Bereiches

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR	16#1A00	16#0034	WaitDistBlockedOutfeed ist außerhalb des gültigen Bereiches
RESOURCE_ERROR	16#1A01	16#0000	Ein oder mehrere Kettenantriebe sind nicht referenziert
RESOURCE_ERROR	16#1A02	16#0000	Outfeed Position ist gesperrt
SYSTEM_ERROR	16#1A03	16#0000	Kollision mit Vorderzug erkannt
SYSTEM_ERROR	16#1A04	16#0000	Infeed-Beschleunigung ist größer als Move-Beschleunigung
RESOURCE_ERROR	16#1A05	16#0000	Fallende Flanke an TrainUnload aber UnloadLift ist nicht in "eingezogen" oder "ausgefahren"
RESOURCE_ERROR	16#1A06	16#0000	Modulowert des Kettenantriebes oder Infeed-Masters sind nicht richtig eingestellt
RESOURCE_ERROR	16#1A07	16#0000	Folgefehler verursacht durch voreilende Kette
SYSTEM_ERROR	16#1A08	16#xxxx	Unerwarteter Zustand einer anderen Kette xxxx: Kettennummer
SYSTEM_ERROR	16#1A09	16#xxxx	Ungültiger interner Array-Index xxxx: Array-Index
RESOURCE_ERROR	16#1A0A	16#xxxx	UnloadLiftLowPos UND UnloadLiftHighPos sind zum gleichen Zeitpunkt gesetzt xxxx = Outfeed-Index
RESOURCE_ERROR	16#1A0B	16#xxxx	Resultierender Fehler verursacht durch eine andere Kette xxxx = Kettennummer mit der Fehlerursache
RESOURCE_ERROR	16#1A0C	16#xxxx	Repositionierung ist erforderlich (START_AT_LAST_STEP) aber Entladevorrichtung ist blockiert xxxx = Kettennummer mit der Fehler Ursache
SYSTEM_ERROR	16#1A0D	16#0000	Ungültige interne Schrittweite
SYSTEM_ERROR	16#1A0E	16#0000	Intern berechneter Kollisions-Zug ist ungültig
SYSTEM_ERROR	16#1A0F	16#0000	Ein nächster Infeed-Schritt wurde getriggert, der vorausgegangene Infeed-Schritt war jedoch nicht abgeschlossen, und ein REST_IN_REST Infeed-Bewegungsgesetz wurde ausgewählt
RESOURCE_ERROR	16#1A10	16#0001	Positionswichtung des Kettenantriebes ist unzulässig
RESOURCE_ERROR	16#1A10	16#0002	Geschwindigkeitswichtung des Kettenantriebes ist unzulässig
RESOURCE_ERROR	16#1A10	16#0003	Beschleunigungswichtung des Kettenantriebes ist unzulässig
RESOURCE_ERROR	16#1A10	16#0004	Positionswichtung der InfeedAxis ist unzulässig
RESOURCE_ERROR	16#1A10	16#0005	Geschwindigkeitswichtung der InfeedAxis ist unzulässig
RESOURCE_ERROR	16#1A10	16#0006	Beschleunigungswichtung der InfeedAxis ist unzulässig
RESOURCE_ERROR	16#1A10	16#0007	Positionswichtung des InfeedMaster ist unzulässig
RESOURCE_ERROR	16#1A10	16#0008	Geschwindigkeitswichtung des InfeedMaster ist unzulässig
RESOURCE_ERROR	16#1A10	16#0009	Beschleunigungswichtung des InfeedMaster ist unzulässig
RESOURCE_ERROR	16#1A11	16#0000	Funktionsbaustein-Ausführungszeit ist größer als Motion-Zykluszeit (C-0-0400)

Tab. 4-17: Fehlercodes Funktionsbaustein MB_MagicBeltType01

4.4.3 Datentypen, MagicBelt

MB_MBELT_MODES

Die Enumeration "MB_MBELT_MODES" enthält die auswählbaren Betriebsarten des Magic Belt Funktionsbausteins:

Element	Wert	Beschreibung
MODE_AUTO	16#100	Auto-Mode
MODE_TUNE	16#300	Tune-Mode
MODE_JOG	16#D00	Jog-Mode

Tab. 4-18: Elemente des Aufzählungstyps MB_MBELT_MODES

MB_MBELT_GENERAL

Die Datenstruktur "MB_MBELT_GENERAL" enthält ketten- und betriebsartenübergreifende Einstellungen:

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
ChainLength	REAL	4000.0	pos. Flanke Enable	Kettenlänge [mm]; Minimum = 1 Max = undefiniert
NoOfChains	UINT	2	pos. Flanke Enable	Kettenanzahl Minimum = 1 Maximum = 3
TrainsPerChain	UINT	1	pos. Flanke Enable	Züge pro Kette Minimum = 1 Maximum = 4
DischargeBackwards	BOOL	FALSE	kontinuierlich	Bestimmt, durch welche Bewegungsrichtung die Züge entleert werden. FALSE: Fahrt über 2/3 Bandlänge in positive Richtung TRUE: Fahrt über 2/3 Bandlänge in negative Richtung
AllowNegDirection	BOOL	FALSE	kontinuierlich	Erlaubt die Bewegung der Züge in negative Richtung
TrainDistance	REAL	5.0	kontinuierlich	Ein nacheilender Zug wird mit diesem Abstand auf einen voreilenden Zug aufschließen (Verwendung im Automatik-Mode) [mm]. Die "Train-Distance" muss größer als die "SafetyDistance" vorgegeben werden, da ansonsten die Kollisionsüberwachung auslöst. Minimum = 0.0mm Maximum = Modulowert/4
SafetyDistance	REAL	0.5	kontinuierlich	Sicherheitsabstand der Kollisions-Überwachung [mm] Minimum = -10mm Maximum = 100mm

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
StopRamp	REAL	10000.0	kontinuierlich	Stopprampen-Verzögerung bei Fehlerreaktion oder Stopp mit der Stoppvariante "STOP_IMMEDIATE" . Die Stopprampe muss größer oder mindestens gleich als der Maximalwert aller anderen Beschleunigungen (MaxMoveAcceleration; MaxInfeedAcceleration; MaxOutfeedAcceleration) sein. Minimum = siehe Beschreibung oben Maximum = undefiniert
StartAtLastStep-Window	REAL	100	kontinuierlich	Bei der Startoption "StartAtLastStep" werden die aktuellen Achspositionen mit den gespeicherten Positionen nach Erreichen des Stillstandes verglichen. Nur wenn die Positionsdivergenz aller Achsen kleiner als das hier vorgegebene Fenster ist, wird gemäß der Startoption "StartAtLastStep" weiter verfahren. Falls mindestens eine Achse außerhalb dieses Fensters liegt, so wird mit der Startoption "Start_At_Begin" weiter verfahren. Minimum: 0mm Maximum: ModuloValue
Feedrate	UINT	100	kontinuierlich	Alle Geschwindigkeiten im "Auto-Mode" werden entsprechend reduziert, falls "FeedrateOverride" < 100 % vorgegeben wird. Minimum: 10% Maximum: 100%
InfeedMaster	AXIS_REF	NO_OBJECT	pos. Flanke Enable	Definiert die Master-Achse zum synchronisierten Beladen mit Radzuführung. Nur notwendig wenn synchronisiertes Beladen ausgewählt ist
InfeedAxis	AXIS_REF	NO_OBJECT	pos. Flanke Enable	Für die Infeed-Bewegung genutzte virtuelle Achse "Infeed Achse"
DummyMaster	AXIS_REF	NO_OBJECT	pos. Flanke Enable	Zur internen "FlexProfile"-Positionierung muss hier eine Dummy-Achse angelegt werden. Der Achstyp und die Achseinstellungen sind nicht relevant. Die Achse wird nicht aktiv angesteuert und kann anwendungsseitig für andere Zwecke genutzt werden. Es wird nur eine gültige Achsreferenz benötigt
ChainAxes	ARRAY [1..3] OF AXIS_REF		pos. Flanke Enable	Achszuordnung der Ketten ChainAxes[1] = Chain A ChainAxes[2] = Chain B ChainAxes[3] = Chain C (optional)

Tab. 4-19: Schnittstellenbeschreibung und Default-Werte der Datenstruktur MB_MBELT_GENERAL

MB_MBELT_MODEJOG

Diese Datenstruktur "MB_MBELT_MODEJOG" enthält die kettenübergreifenden Daten für den Jog-Mode:

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
Incremental	BOOL	FALSE	Pos. Flanke Jog-Plus/JogNeg	TRUE: Mit einer positiven Flanke an "Jog-Plus"/"JogNeg" wird die "JogDistance" in positive oder negative Richtung verfahren FALSE: Fahrt in positive bzw. negative Richtung, solange "JogPlus"/"JogNeg" betätigt ist
SingleAxisJog	BOOL	FALSE	Pos. Flanke Jog-Plus/JogNeg	FALSE: Alle Züge werden synchron vertippt TRUE: Es wird nur der durch "JogAxisSelection" ausgewählte Zug vertippt
JogAxisSelection	UINT	1	Pos. Flanke Jog-Plus/JogNeg	Auswahl der Achse zum Einzelachstippen 1: Chain A, 2: Chain B, 3: Chain C
JogVelocity	REAL	100.0	Pos. Flanke Jog-Plus/JogNeg	Jog-Geschwindigkeit
JogAcceleration	REAL	1000.0	Pos. Flanke Jog-Plus/JogNeg	Jog-Beschleunigung
JogDistance	REAL	2.0	Pos. Flanke Jog-Plus/JogNeg	Jog-Distanz bei inkrementellen Joggen

Tab. 4-20: Schnittstellenbeschreibung und Default-Werte der Datenstruktur MB_MBELT_MODEJOG

MB_MBELT_MODETUNE

Diese Datenstruktur "MB_MBELT_MODETUNE" enthält die kettenübergreifenden Daten für den Tune-Mode:

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
PosDirectionOnly	BOOL	FALSE	bei Bewegungsstart	TRUE: Es wird mit der "TuneDistance" relativ in positive Richtung gefahren FALSE: Von der Ausgangslage beim Start wird die "TuneDistance" im Wechselspiel Vor- und Rückwärts verfahren
MotionLaw	MC_STEP_MODE	X_FIT_SINE_TRAPEZE_ALIM	bei Bewegungsstart	Spezifiziert das Bewegungsgesetz für einen Tune-Positioniervorgang. <i>Verfügbare Bewegungsgesetze sind:</i> • X_FIT_VEL_TRAPEZE_ALIM • X_FIT_SINE_TRAPEZE_ALIM • X_FIT_ACC_TRAPEZE_JLIM • REST_IN_REST_INCLINEDSINE • REST_IN_REST_MOD_TRAPEZE • X_MOTION_IN_MOTION_POLY5_VLIM Detailinformationen zu den Bewegungsgesetzen und dem Datentyp MC_STEP_MODE sind in der FlexProfile-Dokumentation enthalten

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
TuneVelocity	REAL	1000.0	bei Bewegungs-start	Maximale Positionier-Geschwindigkeit
TuneAcceleration	REAL	100.0	bei Bewegungs-start	Maximale Positionier-Beschleunigung
TuneJerk	REAL	200.0	bei Bewegungs-start	Tune-Ruckbegrenzung (wirkt nur bei ruckbegrenzten Bewegungsgesetzen)
TunePosTime	REAL	500	bei Bewegungs-start	<i>Tune-Positionierzeit, wirkt bei allen Bewegungsgesetzen außer:</i> • X_FIT_VEL_TRAPEZE ALIM • X_FIT_SINE_TRAPEZE ALIM • X_FIT_ACC_TRAPEZE JLIM
TuneDistance	REAL	10.0	bei Bewegungs-start	Verfahrweg pro Positioniervorgang

Tab. 4-21: Schnittstellenbeschreibung und Default-Werte der Datenstruktur MB_MBELT_MODETUNE

MB_MBELT_MODEAUTO

Die Datenstruktur "MB_MBELT_MODEAUTO" enthält kettenübergreifende Daten für den Automatik-Mode:

Strukturelement	Typ	Default-Wert	Beschreibung
StartType	MB_BELT_START-TYPE	START_AT_BEGIN	Legt das Startverhalten für den Automatik-Mode fest MB_MBELT_STARTTYPE, Seite 204
StopType	MB_BELT_STOPTYPE	STOP_CYCLE_COMPLETE	Legt das Stopverhalten für den Automatik-Mode fest MB_MBELT_STOPTYPE, Seite 204 Übernahme bei pos. "Stop"- oder neg. "Enable"-Flanke
Infeed	MB_BELT_INFEED		Enthält Daten für das Beladen der Züge MB_MBELT_INFEED, Seite 204
Outfeed	MB_BELT_OUTFEED		Enthält Daten für das Entladen der Züge MB_MBELT_OUTFEED, Seite 209
MoveTo	MB_BELT_MOVE		Enthält Daten für die Bewegung zwischen Be- und Entladen MB_MBELT_MOVE, Seite 210

Tab. 4-22: Schnittstellenbeschreibung und Default-Werte der Datenstruktur MB_MBELT_MODEAUTO

MB_MBELT_STARTTYPE

Die Enumeration "MB_MBELT_STARTTYPE" enthält die möglichen Startvarianten für den Automatik-Mode:

Element	Wert	Beschreibung
START_AT_BEGIN	1	Definierter Start von Grundstellung
START_WITH_TRAIN_UNLOAD	2	Definierter Start von Grundstellung mit vorgeschalteter Entladung der Züge
START_AT_LAST_STEP	3	Start mit letztem (unterbrochenen) Zustand (z. B. nach Fehler oder Stopp)

Tab. 4-23: Elemente des Aufzählungstyps MB_MBELT_STARTTYPE

MB_MBELT_STOPTYPE

Die Enumeration "MB_MBELT_STOPTYPE" enthält die möglichen Stopvarianten für den Automatik-Mode:

Element	Wert	Beschreibung
STOP_IMMEDIATE	1	Sofortiges Stoppen mit Rampe MB_MBELT_GENERAL.StopRamp
STOP_CYCLE_COMPLETE	2	Aktiver Produkt-Zyklus (Infeed-Step) wird noch ausgeführt, danach wird angehalten (definiertes Stillsetzen). Alle Züge die nicht mit der InfeedAxis synchronisiert sind (z. B. bei Fahrten zwischen Outfeedpositionen) werden sofort gestoppt
STOP_IMMEDIATE_SYNC	3	Diese Stopvariante ist nur für Anwendungen mit synchronisiertem Infeed sinnvoll. Die Stopvariante verhält sich wie STOP_IMMEDIATE mit dem Unterschied, dass die Achsen, die im Synchronverbund mit dem InfeedMaster stehen, weiter synchronisiert bleiben. Diese Achsen werden durch das Verzögern des externen InfeedMasters mit verzögert. Das Verzögern des InfeedMasters muss außerhalb des FB sichergestellt werden

Tab. 4-24: Elemente des Aufzählungstyps MB_MBELT_STOPTYPE

MB_MBELT_INFEED

Die Datenstruktur "MB_MBELT_INFEED" enthält die kettenübergreifenden Daten für das Beladen im Automatik-Mode:

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
InfeedType	MB_MBELT_FEEDTYPE	FEED_INDEXED	pos. Flanke Enable	Beladeverfahren (getaktet, synchronisiert, mit Geschwindigkeit MB_MBELT_FEEDTYPE, Seite 207)
StartDelayTime	TIME	T#50ms	pos. Flanke von NextInfeedStep	Nur für getaktetes Beladen relevant: Verzugszeit (pos. Flanke "NextInfeedStep"--> Start Bewegung) zum Weitertakten Minimum = T#0ms Maximum = T#1h

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
InfeedLoadTime	TIME	T#10ms	pos. Flanke von NextInfeedStep	Nur für getaktetes Beladen relevant: Zeitbedarf für die Zug Beladung. Innerhalb dieser Zeit verweilt der Zug im Stillstand um das Produkt beim Beladen nicht zu beschädigen. Eingang ist unwirksam bei den Bewegungsgesetzen: X_FIT_ACC_TRAPEZE_JLIM X_FIT_SINE_TRAPEZE_ALIM X_FIT_VEL_TRAPEZE_ALIM Da in diesem Fall die vorgegeben a,v, j Maximalwerte verwendet werden
MaxProductRate	UINT	200	pos. Flanke von NextInfeedStep	Nur für getaktetes Beladen relevant: Maximale Anzahl von Produkten pro Minute die eingetaktet werden sollen [1/min] (Produktstrom). Eingang ist unwirksam bei den Bewegungsgesetzen: X_FIT_ACC_TRAPEZE_JLIM X_FIT_SINE_TRAPEZE_ALIM X_FIT_VEL_TRAPEZE_ALIM Da in diesem Fall die vorgegeben a,v, j Maximalwerte verwendet werden
MotionLaw	MC_STEP_MODE	VELOCITY_IN_VELOCITY_MOD_SINE	pos. Flanke von NextInfeedStep	Nur für getaktetes Beladen relevant: Spezifiziert das Bewegungsgesetz für einen Infeed-Schritt (Detail-Beschreibung der Bewegungsgesetze siehe FlexProfile-Dokumentation). <i>Folgende Bewegungsgesetze werden für MagicBelt Anwendungen empfohlen:</i> • VELOCITY_IN_VELOCITY_MOD_SINE: Modifizierte Sinoide / ruckarm aber hohe Beschleunigungsspitze • X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM: Modifiziertes Beschleunigungstrapez / stark ruckbehaftet aber geringste Beschleunigungsspitze • X_FIT_ACC_TRAPEZE_JLIM: Ruckbegrenzte Bewegung. Bei diesem Bewegungsgesetz wirkt der Eingang "MaxProductRate" nicht, da die a,v, j Maximalwerte verwendet werden Achtung: Bei "REST_IN_REST_XXX"- Bewegungsgesetze muss der vorausgegangene Infeed-Schritt bei Start eines neuen Infeed-Schrittes abgeschlossen sein. Es wird der Fehler Add1 = 16#1A0F ausgegeben wenn diese Bedingung nicht erfüllt ist

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
MinDistance-ToOutfeed	REAL	400.0	kontinuierlich	Minimale Distanz zwischen Zug im "Infeed"-Zustand und einem weiteren Zug im "Outfeed"-Zustand. Wird die Distanz unterschritten, wird der "Ready"-Ausgang zurückgesetzt um einen Produktstau zu vermeiden. Details siehe Produktstau-Überwachung
MaxInfeedVelocity	REAL	300000.0	kontinuierlich	Infeed-Schritt Maximalgeschwindigkeit: <i>Nur wirksam bei den Bewegungsgesetzen:</i> • X_FIT_ACC_TRAPEZE_JLIM • X_FIT_SINE_TRAPEZE_ALIM • X_FIT_VEL_TRAPEZE_ALIM • X_MOTION_IN_MOTION_POLY5_VLIM • X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM
MaxInfeedAcceleration	REAL	5000.0	kontinuierlich	Infeed Schritt Maximalbeschleunigung: <i>Nur wirksam bei den Bewegungsgesetzen:</i> • X_FIT_ACC_TRAPEZE_JLIM • X_FIT_SINE_TRAPEZE_ALIM • X_FIT_VEL_TRAPEZE_ALIM • X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM
MaxInfeedJerk	REAL	1000000.0	kontinuierlich	Infeed Schritt Maximalruck: <i>Nur wirksam bei Bewegungsgesetz:</i> • X_FIT_ACC_TRAPEZE_JLIM
ProductBuffer	UINT	3	pos. Flanke Enable	Nur für synchronisiertes Beladen: Anzahl der Produkte zwischen Produkterkennung und Produktauswurf Minimum: 0 Maximum: 64
ProdExpectationWindowStart	REAL	30.0	kontinuierlich	Nur für synchronisiertes Beladen: Start des Produkt-Erwartungsfensters. Innerhalb des Erwartungsfensters wird der Eingang "NextInfeedStep" ausgewertet. Minimum = 0° Maximum = ProdExpectationWindowEnd
ProdExpectationWindowEnd	REAL	180.0	kontinuierlich	Nur für synchronisiertes Beladen: Ende des Produkt-Erwartungsfensters. Innerhalb des Erwartungsfensters wird der Eingang "NextInfeedStep" ausgewertet. Minimum = ProdExpectationWindowStart Maximum = 360°

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
FirstInfeedPosition	REAL	500.0	kontinuierlich	Erste (absolute) Ladeposition Minimum = 0 Maximum = ModuloValue
AdditionalSteps	UINT	0	kontinuierlich	Der bereits vollständig beladene Zug kann mit "AdditionalSteps" noch einen oder mehrere Steps vor dem nächsten Zug hergeschoben werden, bevor er zur Outfeedposition fährt (z. B. bis Niederhalter erreicht wird)
InfeedStepData	ARRAY [1..3, 1..40] OF MB_MBELT_INFEEDSTEPS		kontinuierlich	Liste mit der Größe der Fächer und Produktanzahl pro Fach für alle Ketten. "InfeedStepData"[1,x] = Kette A "InfeedStepData"[2,x] = Kette B "InfeedStepData"[3,x] = Kette C (optional) Bei dem synchronisierten Beladen wird immer von einem Produkt pro Fach ausgegangen MB_MBELT_INFEEDSTEPS, Seite 207

Tab. 4-25: Schnittstellenbeschreibung und Default-Werte Datenstruktur MB_MBELT_INFEED

MB_MBELT_FEEDTYPE

Die Enumeration "MB_MBELT_FEEDTYPE" enthält die möglichen Be- und Entladeverfahren der Züge:

Element	Wert	Beschreibung
FEED_INDEXED	1	Getaktetes Beladen
FEED_SYNCHRONIZED	2	Synchronisiertes Beladen

Tab. 4-26: Elemente des Aufzählungstyps MB_MBELT_FEEDTYPE

MB_MBELT_INFEEDSTEPS

Die Datenstruktur "MB_MBELT_INFEEDSTEPS" enthält kettenspezifische Daten für das Beladen im Automatik-Mode:

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
PocketWidth	REAL	50.0	kontinuierlich	Abstand von Vorderkante des aktuellen zur Vorderkante des nachfolgenden Faches. Sind mehrere Fächer gleich groß, so kann die Anzahl der gleich großen Fächer über das Element "PocketQuantity" vorgegeben werden
PocketQuantity	UINT	1	kontinuierlich	Anzahl der Fächer mit der gleichen Größe -1 = Kennung für Gruppenbildung (siehe Beispiel) 0 = Kennung für Zugende (siehe Beispiel)
ProductsPerPocket	UINT	0	kontinuierlich	Produktanzahl pro Fach (nur für getaktetes Beladen relevant)

Tab. 4-27: Schnittstellenbeschreibung und Default-Werte der Datenstruktur MB_MBELT_INFEEDSTEPS

Beispiel:

Beispiel 1: Gleiche Fächer (siehe auch Abbildung unten):

Zug besteht aus 10 gleichen Fächern mit 50mm Fachbreite mit jeweils einem Produkt

ChainA_InfeedData[1].PocketWidth = 50mm

ChainA_InfeedData[1].PocketQuantity = 10

ChainA_InfeedData[1].ProductsPerPocket = 1

ChainA_InfeedData[2].PocketWidth = don't care

ChainA_InfeedData[2].PocketQuantity = 0 <- Zug Ende Kennung

ChainA_InfeedData[2].ProductsPerPocket = don't care

Beispiel:

Beispiel 2: Unterschiedliche Fächer (siehe auch Abbildung unten):

Der Zug besteht aus 3 unterschiedlichen Fächern (40mm, 80mm, 120mm) mit (1,2,3) Produkten pro Fach

ChainA_InfeedData[1].PocketWidth = 40mm

ChainA_InfeedData[1].PocketQuantity = 1

ChainA_InfeedData[1].ProductsPerPocket = 1

ChainA_InfeedData[2].PocketWidth = 80mm

ChainA_InfeedData[2].PocketQuantity = 1

ChainA_InfeedData[2].ProductsPerPocket = 2

ChainA_InfeedData[3].PocketWidth = 120mm

ChainA_InfeedData[3].PocketQuantity = 1

ChainA_InfeedData[3].ProductsPerPocket = 3

ChainA_InfeedData[4].PocketWidth = don't care

ChainA_InfeedData[4].PocketQuantity = 0 <- Zug Ende Kennung

ChainA_InfeedData[4].ProductsPerPocket = don't care

Beispiel:

Beispiel 3: Fächergruppe (siehe auch Abbildung unten):

Der Zug besteht aus einer Fächergruppe mit 5 * 3 unterschiedlichen Fächern (40mm, 80mm, 120mm) mit (1,2,3) Produkten pro Fach

ChainA_InfeedData[1].PocketWidth = 40mm

ChainA_InfeedData[1].PocketQuantity = -1 <-Gruppen-Kennung

ChainA_InfeedData[1].ProductsPerPocket = 1

ChainA_InfeedData[2].PocketWidth = 80mm

ChainA_InfeedData[2].PocketQuantity = -1 <-Gruppen-Kennung

ChainA_InfeedData[2].ProductsPerPocket = 2

ChainA_InfeedData[3].PocketWidth = 120mm

ChainA_InfeedData[3].PocketQuantity = 5 <-Gruppen-Ende-Kennung (Wert > 0) mit Anzahl der Gruppen

ChainA_InfeedData[3].ProductsPerPocket = 3

ChainA_InfeedData[4].PocketWidth = don't care

ChainA_InfeedData[4].PocketQuantity = 0 <-Zug-Ende-Kennung

ChainA_InfeedData[4]. ProductsPerPocket = don't care

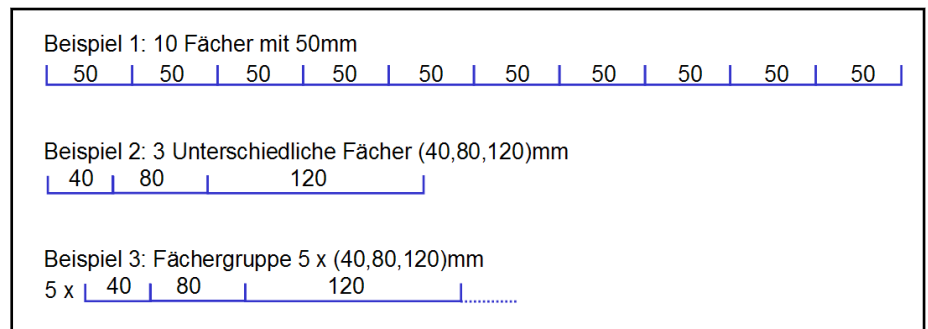


Abb. 4-21: Zugkonfigurationen der oben gezeigten Beispiele

MB_MBELT_OUTFEED

Die Datenstruktur "MB_MBELT_OUTFEED" enthält die kettenübergreifenden Daten für das Entladen im Automatik-Mode:

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
MotionLaw	MC_STEP_MODE	X_FIT_SINE_TRAPEZE ALIM	pos. Flanke von UnloadingTrain	Spezifiziert das Bewegungsgesetz für die Bewegung zur nächsten Outfeedposition (Detail-Beschreibung der Bewegungsgesetze siehe FlexProfile-Dokumentation) Beschleunigungsbegrenzte Bewegung (X_FIT_VEL_TRAPEZE ALIM) Ruckbegrenzte Bewegung (X_FIT_ACC_TRAPEZE ALIM) Beschleunigungsbegrenzte Sinoide (X_FIT_SINE_TRAPEZE ALIM)
MaxOutfeedVelocity	REAL	60000.0	kontinuierlich	Maximale Geschwindigkeit zum Weiterfahren auf die nächste Entladeposition
MaxOutfeedAcceleration	REAL	5000.0	kontinuierlich	Maximale Beschleunigung zum Weiterfahren auf die nächste Entladeposition
MaxOutfeedJerk	REAL	0.0	kontinuierlich	Maximaler Ruck zum Weiterfahren auf die nächste Entladeposition Nur wirksam bei Bewegungsgesetz X_FIT_ACC_TRAPEZE ALIM
FirstOutfeedPosition	REAL	2000.0	kontinuierlich	Erste (absolute) Entladeposition
WaitDistBlockedOutfeed	REAL	0.0	kontinuierlich	Bei Fahrt auf eine belegte Outfeed-Position stoppt der Zug mit dem hier eingetragenen Abstand zum Vorderzug. Mit WaitDistBlockedOutfeed = 0 wird keine Warteposition angefahren (Verhalten wie bei Version 12 und vorher)

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
EnableUnload-Handshake	BOOL	FALSE	kontinuierlich	FALSE: Entladehandshake abgeschaltet. Für Anwendungen ohne Kollisionsgefahr der Entnahmeverrichtung mit dem Zug (z. B. Entnahmeroboter) TRUE: Entladehandshake eingeschaltet. Für Anwendungen mit Kollisionsgefahr der Entnahmeverrichtung mit dem Zug. "UnloadLiftLowPos" und "UnloadLiftHighPos" müssen entsprechend gesetzt werden (z. B. Entnahmerechen)
OutfeedDistances	ARRAY [1..3, 2..10] OF REAL	0.0	kontinuierlich	Diese Liste ist nur notwendig wenn mehr als eine Entladestelle angefahren werden soll. In diesem Fall können über diese Liste die Entfernungen von Entladestelle zur nachfolgenden Entladestelle angegeben werden. Mit jeder negativen Flanke des Eingangs "UnloadingTrain" wird zur nächsten Entladestelle weitergetaktet. Das Ende der Liste ist erreicht wenn der Wert 0 eingetragen ist. "OutfeedDistances" [1,x] = Kette A "OutfeedDistances" [2,x] = Kette B "OutfeedDistances" [3,x] = Kette C (optional)
UnloadLiftLowPos	ARRAY [1...10] OF BOOL	TRUE	kontinuierlich	Entladerechen Lage-Endschalter/TRUE = Entladerechen eingezogen
UnloadLiftHighPos	ARRAY [1...10] OF BOOL	TRUE	kontinuierlich	Entladerechen Lage-Endschalter/TRUE = Entladerechen ausgefahren

Tab. 4-28: Schnittstellenbeschreibung und Default-Werte der Datenstruktur MB_MBELT_OUTFEED

MB_MBELT_MOVE

Die Datenstruktur "MB_MBELT_MOVE" enthält die kettenübergreifenden Daten für das Fahren zwischen der Be- und Entladestation:

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
MaxMoveVelocity	REAL	300000.0	Bei Start der Move-Bewegung	Maximale Geschwindigkeit zum Fahren zwischen Be- und Entladestation und zum Aufsynchronisieren
MaxMoveAcceleration	REAL	10000.0	Bei Start der Move-Bewegung	Maximale Beschleunigung/Verzögerung zum Fahren zwischen Be- und Entladestation
MaxMoveJerk	REAL	0.0	Bei Start der Move-Bewegung	Maximaler Ruck zum Fahren zwischen Be- und Entladestation
LoadTrainAcceleration	UINT	100	Bei Start der Move-Bewegung	Bei der Bewegung von der Beladestation zu der ersten Entladestation kann über den Eingang "LoadTrainAcceleration" die Beschleunigung reduziert werden. Die wirksame Beschleunigung berechnet sich wie folgt: $\text{MaxMoveAcceleration} * \text{LoadTrainAcceleration} [\%] / 100 [\%]$

Strukturelement	Typ	Default-Wert	Übernahme	Beschreibung
LoadTrainDeceleration	UINT	0	Bei Start der Move-Bewegung	Bei der Bewegung von der Beladestation zu der ersten Entladestation kann über den Eingang "LoadTrainDeceleration" die Verzögerung reduziert werden. Die wirksame Verzögerung berechnet sich wie folgt: $\text{MaxMoveAcceleration} * \text{LoadTrainDeceleration} [\%] / 100[\%]$ Für den Sonderfall "LoadTrainDeceleration = 0" wirkt der Eingang "LoadTrainAcceleration" auch auf die Verzögerung
BeginReducedVel1	REAL	0.0	Bei Start der Move-Bewegung	Ab dieser Position (der Zug-Vorderkante) fährt der Zug mit reduzierter Geschwindigkeit. Damit können die entstehenden Fliehkräfte, welche durch die Umlenkung in den Kurven entstehen, reduziert werden
EndReducedVel1	REAL	0.0	Bei Start der Move-Bewegung	Ab dieser Position (der Zug-Vorderkante) ist die Geschwindigkeitsbegrenzung wieder aufgehoben
BeginReducedVel2	REAL	0.0	Bei Start der Move-Bewegung	Ab dieser Position (der Zug-Vorderkante) fährt der Zug mit reduzierter Geschwindigkeit. Damit können die entstehenden Fliehkräfte, welche durch die Umlenkung in den Kurven entstehen, reduziert werden
EndReducedVel2	REAL	0.0	Bei Start der Move-Bewegung	Ab dieser Position (der Zug-Vorderkante) ist die Geschwindigkeitsbegrenzung wieder aufgehoben
ReducedVelocity	REAL	100000.0	Bei Start der Move-Bewegung	Geschwindigkeitsbegrenzung zur Reduzierung der Fliehkräfte in den Kurven
AddaptiveSyncRamp	BOOL	TRUE	Bei Start der Move-Bewegung	Aktiviert adaptive Synchronisations-Rampe
CatchUpAcceleration	REAL	100.0	Bei Start der Move-Bewegung	Maximale Beschleunigung der Aufsynchro-nisierbewegung zur Synchronisation auf den In-feed. Vorgabe in % der MaxMoveAcceleration

Tab. 4-29: Schnittstellenbeschreibung und Default-Werte der Datenstruktur MB_MBELT_MOVE

MB_MBELT_STATUS

Die Datenstruktur " MB_MBELT_STATUS" enthält Statusinformationen des Zuges:

Strukturelement	Typ	Beschreibung
ChainStates	ARRAY [1..3] OF MB_MBELT_MODES	Aktueller Ist-Zustand der Ketten MB_MBELT_MODES, Seite 200
PocketsToBeLoaded	UINT	Ein Zug im Zustand "Auto-Infeed" gibt über dieses Strukturelement aus, wie viele Fächer noch in dem Zug beladen werden müssen. Damit kann z. B. übergeordnet berechnet werden, wie lange der Ladevorgang voraussichtlich noch andauert

Strukturelement	Typ	Beschreibung
ActInfeedStep	UINT	Ein Zug im Zustand "Auto-Infeed" gibt über dieses Strukturelement aus welcher "Infeed" Schritt aktuell bearbeitet wird
ProductInExpectationWindow	BOOL	Im Infeed-Modus "FEED_SYNCHRONIZED" wird dieser Ausgang gesetzt, wenn sich die Achse "InfeedMaster" in dem Erwartungsfenster befindet. Das Erwartungsfenster wird durch die Strukturelemente "MB_BELT_INFEED", "ProdExpectationWindowStart" und "MB_MBELT_INFEED" und "ProdExpectationWindowEnd" definiert
AllChainsHomed	BOOL	Alle Ketten sind referenziert
Error	BOOL	Kettenübergreifendes Error-Flag
ErrorSourceChain	UINT	Fehlerverursachende Kettennummer
MeasuredInfeedStepRate	REAL	Gemessene InfeedStepRate [min-1] (wird durch den zeitlichen Abstand der "NextInfeedStep" Flanken ermittelt)
MaxProdExpectationWindowEnd	REAL	Maximal mögliche Endposition des Erwartungsfensters "ProdExpectationWindowEnd" berechnet aus der aktuellen Geschwindigkeit der InfeedMaster Achse [°]
ChainInAutoInfeed	UINT	Kettennummer, die sich im Zustand "AUTO_INFEED" befindet
SupressedInfeedSteps	UINT	Anzahl der unterdrückten InfeedSteps. Wird inkrementiert, bei der Beladevariante "FEED_INDEXED" wenn eine "NextInfeedStep" Flanke angelegt wird und der Infeed-Zug aber den Infeed-Step nicht ausführen kann (für Diagnosezwecke)

Strukturelement	Typ	Beschreibung
StartAtLastStepPossible	ARRAY [1..3] OF BOOL	Ist TRUE wenn die Startoption "START_AT_LAST_STEP" für die jeweilige Kette möglich ist. Index 1 gilt für Kette A Index 2 gilt für Kette B Index 3 gilt für Kette C In dem Zustand "STATE_MBELT_AUTO_STOPPED" werden die Bits für die Ketten nach folgendem Kriterium gebildet. Das Bit der jeweiligen Kette wird gesetzt wenn: Absolute Positions Differenz zwischen aktueller Zugposition und Position nach Stopp (Status.RestoreData) kleiner als GeneralSettings.StartAtLastStepWindow ist UND Zustand bei Stopp (Status.RestoreData) war ein Automatik Zustand UND <i>Zustand bei Stopp war nicht:</i> • STATE_MBELT_AUTO_STOPPING oder • STATE_MBELT_AUTO_STOPPED oder • STATE_MBELT_AUTO_STARTING UND (EnableUnloadHandshake = FALSE ODER Absolute Positions Differenz zwischen aktueller Zugposition und Position nach Stopp ist kleiner als 2mm ODER alle Entladevorrichtungen UnloadLiftHighPos sind eingefahren)
MaxNoOfInfeedSteps	UINT	Maximale Infeed-Step Anzahl für den aktuellen Zug an der Infeed-Position
FirstMasterChain	UINT	Nummer der Kette, die zuerst die FirstInfeedPosition erreicht (nur für interne Zwecke)
OutfeedPosOccupied	ARRAY [1...10] OF BOOL	Wird für jede Outfeedposition gesetzt, wenn diese durch einen Zug oder den Entlade-Rechen besetzt ist
OccupiedChain	ARRAY [1...10] OF UINT	Kettennummer welche die Outfeed-Position belegt
ChainCmdStates	ARRAY [1..3] OF MB_MBELT_STATES	Berechneter Zielzustand (nur für interne Zwecke)
RestoreData	ARRAY [1..3] OF MB_MBELT_RESTORE	Gespeicherte Informationen zum Stopp-Zeitpunkt um bei dem "StartType" = "START_AT_LAST_STEP" den letzten Zustand wieder herzustellen (nur für interne Zwecke)

Tab. 4-30: Schnittstellenbeschreibung und Default-Werte der Datenstruktur MB_MBELT_STATUS

4.4.4 Betriebsarten, MagicBelt

Betriebsarten, MagicBelt, Einführung und Überblick

Der Funktionsbaustein [MB_MagicBeltType01](#), Seite 192, enthält Funktionalitäten zur Ansteuerung einer Kette einer MagicBelt-Applikation.

Je nach angewählter Betriebsart wird die jeweilige Kette entsprechend angesteuert:

- **Jog-Mode, MagicBelt, Seite 214:** Die Züge können einzeln oder synchron wahlweise kontinuierlich oder inkrementell vertippt werden
- **Tune-Mode, MagicBelt, Seite 214:** Um Dynamikgrenzwerte (z. B. maximale Beschleunigung) zu ermitteln, werden die Züge synchronisiert und fortlaufend um eine einstellbare Distanz verfahren
- **Automatik-Mode, MagicBelt, Seite 215:** Im Wechselspiel fahren die Züge jeweils die Be- und Entladeposition(en) an



Der Betrieb des Funktionsbausteins erfordert besondere Randbedingungen (z. B. Modulo-Parametrierung, zyklischer Aufruf des FB in der Motion-Task...).

Bitte beachten Sie daher Inbetriebnahmehinweise, MagicBelt, Seite 229.

Jog-Mode, MagicBelt

Jog-Betriebsart In der Jog Betriebsart sind folgende Ein/Ausgänge aktiv:

Funktionsbaustein-Eingänge:

- JogPlus
- JogNeg

Funktionsbaustein-Ausgänge:

- JogDistanceDone

Funktionsbaustein-Ein/Ausgänge:

- JogSettings

Mit der Jog-Betriebsart können zum Einrichten der Maschine die Züge inkrementell oder kontinuierlich vertippt werden. Nach Aktivierung der Jog-Betriebsart sind die Züge zunächst im Stillstand (Stopped State). Das Joggen der Achsen ist mit den optional enthaltenen Entlade-Rechen verriegelt um Kollisionen zu vermeiden.

Daher ist das anschließende Joggen nur möglich wenn:

- Der FB-Eingang "UnloadingTrain" zurückgesetzt ist UND
- die optionalen Entlade-Rechen eingezogen sind, d. h. die Entlade-Rechen Endschalter-Signale sind: "MB_MBELT_OUTFEED. UnloadLift-LowPos[1..10]" = TRUE

Ist die o. g. Bedingung erfüllt, können über die Eingänge "JogPlus" oder "JogNeg" die Züge in positive bzw. negative Richtung vertippt werden.

Die folgenden Optionen sind dabei möglich:

- Über das Strukturelement "MB_MBELT_MODEJOG.Incremental" kann zwischen inkrementellem und kontinuierlichem Joggen umgeschaltet werden (Standardeinstellung = kontinuierliches Joggen)
- Über das Strukturelement "MB_MBELT_MODEJOG.SingleAxisJog" kann das Einzelachs-Joggen aktiviert werden. Der zu vertippende Zug wird dann über das Strukturelement "MB_MBELT_MODEJOG.JogAxis-Selection" ausgewählt (Standardeinstellung = synchronisiertes Joggen)

Tune-Mode, MagicBelt

Tune-Betriebsart In der Tune-Betriebsart sind folgende Ein/Ausgänge aktiv:

Funktionsbaustein-Eingänge:

- Start
- Stop

Funktionsbaustein-Ein/Ausgänge:

- TuneSettings

Mit der Tune Betriebsart können beim Einrichten der Maschine die optimalen Dynamikeinstellungen/Bewegungsgesetze ermittelt werden. Im permanenten Start/Stop-Betrieb werden die Züge hierbei gemäß des ausgewählten Bewegungsgesetzes wahlweise in eine Richtung oder bipolar bewegt. Wie bei der Jog-Betriebsart ist das Starten der Achsen mit den optionalen Entlade-Rechen verriegelt (Detailbeschreibung siehe Jog-Betriebsart).

Nach positiver Flanke von Start werden die Ketten synchronisiert und fortlaufend zyklisch um die Distanz "MB_MBELT_MODETUNE.TuneDistance" verfahren. Je nach Einstellung von "MB_MBELT_MODETUNE.PosDirectionOnly" werden die Züge entweder fortlaufend in positive Richtung oder im Wechselspiel Vor- und Zurück verfahren. Während der Fahrt können die Dynamikeinstellungen (z. B. "MB_MBELT_MODETUNE.TuneAcceleration" verändert werden um z. B. Maximalwerte für die Beschleunigungen zu ermitteln.

Automatik-Mode, MagicBelt**Automatik-Mode, MagicBelt, Allgemeines**

Auto-Betriebsart In der Auto-Betriebsart sind folgende Ein/Ausgänge aktiv:

Funktionsbaustein-Eingänge:

- Start
- Stop
- NextInfeedStep
- UnloadingTrain

Funktionsbaustein Ausgänge:

- InfeedStepDone
- TrainReadyToUnload
- ReleaseUnloadLift
- Ready

Funktionsbaustein-Ein/Ausgänge:

- AutoSettings
- Status

Im Automatik-Mode werden die Züge im Wechselspiel zwischen den Be- und Entladepositionen bewegt, um den nahezu kontinuierlichen Produktstrom an der Beladeposition aufzunehmen und zu gruppieren. Ist der Zug voll beladen, werden die Entladepositionen angefahren, um die Produktgruppen über einen Entlade-Rechen oder einen Entladeroboter zu entnehmen. Nach der Entladung fährt der jeweilige Zug wieder die Beladeposition an.

Start, Automatik-Mode, MagicBelt

Die positive Start-Flanke gibt die Bewegung der Züge frei. Je nach Startoption ("AutoSettings.StartType") wird wie folgt gestartet:

START_AT_BEGIN:

Der Zug mit der geringsten Entfernung zur "MB_MBELT_INFEED.FirstInfeed-Position" ist der führende Zug und damit der Master. Hierbei wird auch be-

rücksichtigt ob sich die Züge nur in positive Richtung oder bipolar bewegen dürfen (einstellbar über "MB_MBELT_GENERAL.AllowNegDirection").

Alle Slave-Züge schließen mit dem Zugabstand "MB_MBELT_GENERAL.TrainDistance" auf den führenden Zug auf. Der führende Zug fährt zur "MB_MBELT_INFEED.FirstInfeedPosition", alle anderen Züge folgen synchron. Nach Erreichen der "FirstInfeedPosition" startet das Beladen des Zuges.

START_WITH_TRAIN_UNLOAD:

Vorgehensweise wie bei START_AT_BEGIN mit dem Unterschied dass eine zusätzliche Zug-Entladebewegung um $\frac{2}{3}$ * "MB_MBELT_GENERAL.ChainLength" vor Fahrt auf die "FirstInfeed"-Position ausgeführt wird, um die Züge sicher zu entladen. Die Richtung der Entladebewegung kann hierbei durch das Strukturelement "MB_MBELT_GENERAL.DischargeBackwards" beeinflusst werden. Die Entladebewegung wird mit der maximalen Geschwindigkeit "MB_MBELT_MOVE.MaxMoveVelocity" ausgeführt.

START_AT_LAST_STEP:

Mit dieser Startoption kann der letzte Zustand nach Fehler oder Stopp wiederhergestellt werden. Bei dieser Startoption wird zunächst geprüft, ob alle Achs-Positionen noch innerhalb des einstellbaren Fensters MB_MBELT_GENERAL.StartAtLastStepWindow liegen (im Vergleich zur abgespeicherten Position nach Stopp). Falls mindestens eine der Achsen außerhalb dieses Fensters liegt, so wird nach Startoption "START_AT_BEGIN" verfahren. Liegen alle Achsen innerhalb dieses Fensters, so werden im nächsten Schritt alle Achsen auf die zuletzt aktive Position vor Stopp oder Fehler repositioniert. Anschließend wird der letzte Zustand, vor Stopp oder Fehler, wiederhergestellt. Zur Wiederherstellung werden die Achsen - wenn notwendig - repositioniert.

Diese Startoption kann daher nur ausgeführt werden, wenn folgende Randbedingungen erfüllt sind:

- Positionsdivergenz zwischen aktueller Ist-Position und letzter Position der Chain-Achsen im Stillstand liegt innerhalb des "MB_MBELT_GENERAL.StartAtLastStepWindow". Ist die Bedingung nicht erfüllt, wird START_AT_BEGIN ausgeführt
- Die abgespeicherten letzten Zustände vor der Unterbrechung von allen Chain-Achsen entsprechen dem Auto-Mode. Ist die Bedingung nicht erfüllt wird START_AT_BEGIN ausgeführt
- Ist eine Entladevorrichtung vorhanden (EnableUnloadHandshake=TRUE) und ist mindestens ein Entladerechen ausgefahren (UnloadLiftHighPos[n] = TRUE) und ist eine Repositionierung von mehr als 2mm an mindestens einer Achse erforderlich, so kann kein "START_AT_LAST_STEP" ausgeführt werden, da ein mechanischer Schaden auftreten könnte. In diesem Fall wird daher ein Fehler ausgegeben

Das Repositionieren wird wie folgt durchgeführt:

- Ketten werden synchronisiert (relative Synchronisation)
- Nur wenn 3 Ketten konfiguriert sind: Kette C positioniert auf letzte gültige abgespeicherte Position. Alle weiteren Ketten folgen synchron. Weiter wenn Kette C die Zielposition erreicht hat
- Kette B positioniert auf letzte gültige abgespeicherte Position. Kette C (wenn vorhanden) bleibt im Stillstand. Kette A folgt synchron. Weiter wenn Kette B die Zielposition erreicht hat

- Kette A positioniert auf letzte gültige abgespeicherte Position. Alle anderen Ketten bleiben im Stillstand. Weiter wenn Kette A die Zielposition erreicht hat
- Der letzte Zustand und alle notwendigen internen Variablen werden wieder hergestellt, um den Zustand vor der Unterbrechung wieder zu erreichen

Beladen, Automatik-Mode, MagicBelt

Alle Einstellungen zu dem Beladevorgang werden über das Strukturelement "MB_MBELT_MODEAUTO.Infeed" vorgegeben.

Grundsätzlich wird zwischen getaktetem und synchronisiertem Beladen des Zuges unterschieden. Bei dem getaktetem Beladen wird der Zug an der Beladeposition über den FB Eingang "NextInfeedStep" weitergetaktet bis alle Fächer des Zuges gefüllt sind.

Bei dem synchronisierten Beladen ist der Zug beim Beladevorgang synchronisiert mit der Produktzuführung (Produktzuführung z. B. über Eintaktrad). Dieses Beladeverfahren ist im allgemeinen produktschonender für höhere Taktzeiten. Bei diesem Verfahren wird immer von einer gleichen Fächerbreite ausgegangen. Falls ein Fach der Produktzuführung leer ist, muss der Zug beim Beladen taktsynchron für eine Fächerbreite aussetzen (Leererteilungskompensation).

Das Zug-Beladeverfahren wird über das Strukturelement "MB_MBELT_MODEAUTO.Infeed.InfeedType" festgelegt.

Getaktetes Beladen, Automatik-Mode, MagicBelt

Beim getakteten Beladen wirken alle Elemente der Datenstruktur "MB_MBELT_MODEAUTO.Infeed" (Ausnahme: ProductBuffer und ProductExpectationWindow wirken nicht).

Beim getakteten Beladen (AutoSettings.Infeed.InfeedType = FEED_INDEXED) kommen die Produkte über Transportbänder mit mehr oder weniger gleichen Abständen, aber mit einem Mindestabstand zueinander. Auch bei quasi gleichen Abständen können einzelne oder mehrere Produkte fehlen. Das Produkt löst bei dem Weg in das Fach ein Sensorsignal zur Weitertaktung des Zuges aus. Dieses Signal wird an den FB-Eingang "NextInfeedStep" angelegt.

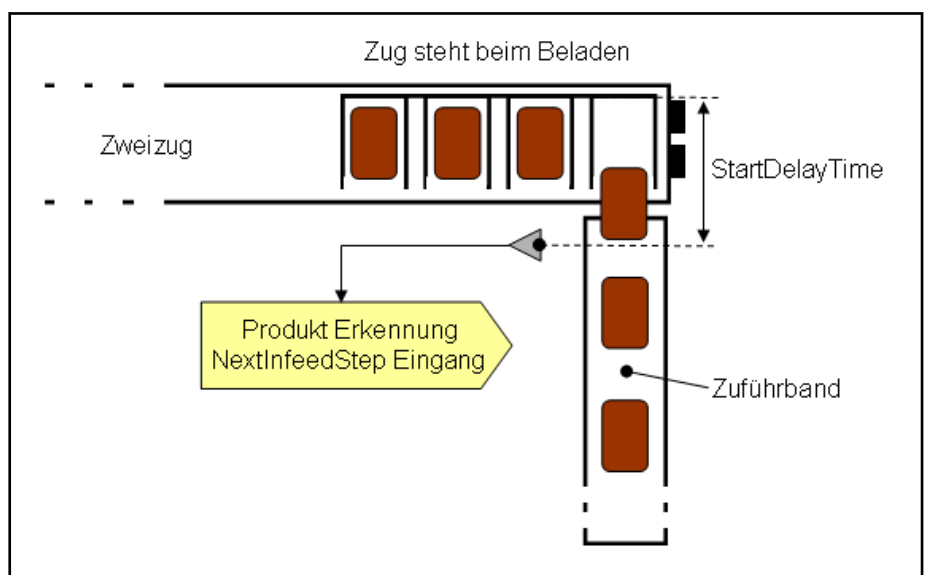


Abb. 4-22: Getaktete Produktzuführung/Beladen im Stillstand von Vorne

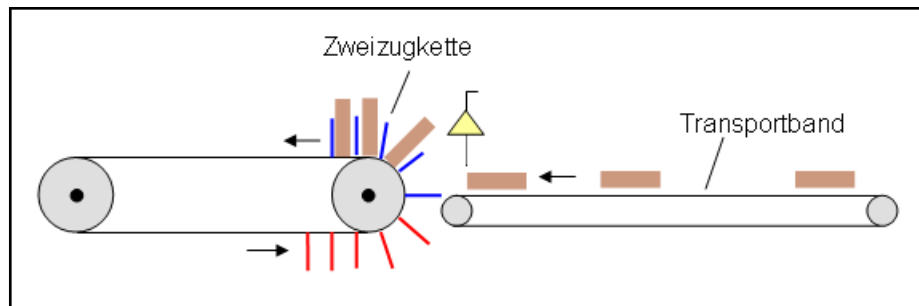


Abb. 4-23: Getaktete Produktzuführung / Beladen im Stillstand seitlich

Über eine einstellbare Zeit "AutoSettings.Infeed.StartDelayTime" kann der Transportweg vom Erkennen des Sensorsignales bis zum Start der Weiertakt-Bewegung eingestellt werden. Nach Abschluss der Weiertaktbewegung wird der FB-Ausgang "InfeedStepDone" zur Quittierung gesetzt. Die Fächerbreite sowie die Produkte pro Tasche kann von Tasche zu Tasche variieren und wird über das Strukturelement "MB_MBELT_MODEAUTO.Infeed.InfeedStepData" vorgegeben.

Die Eintaktbewegung wird hierbei durch folgenden Strukturelemente festgelegt (siehe auch nachfolgende Abbildung):

- "MB_MBELT_MODEAUTO.Infeed.MotionLaw" spezifiziert das genutzte Bewegungsgesetz
- "MB_MBELT_MODEAUTO.Infeed.StartDelayTime" spezifiziert die Zeitverzögerung zum Start der Infeed-Bewegung
- "MB_MBELT_MODEAUTO.Infeed.InfeedLoadTime" spezifiziert die Zeit, die der Zug im Stillstand stehen soll (und dabei beladen wird)
- "MB_MBELT_MODEAUTO.Infeed.MaxProductRate" spezifiziert die maximale Anzahl von Produkten pro Minute

Das ausgewählte Bewegungsgesetz "MB_MBELT_MODEAUTO.Infeed.MotionLaw" hat große Bedeutung auf die ausgeführte Infeed-Bewegung.

Hierbei ist folgendes zu beachten:

- Grundsätzlich sollten möglichst "VELOCITY_IN_VELOCITY..."-Bewegungsprofile verwendet werden, da diese auch eine Zug-Weiertaktung unter Geschwindigkeit erlauben (wenn z. B. der vorausgegangene Infeed Step noch nicht vollständig abgeschlossen ist). In Sonderfällen können auch alle "Rast in Rast" FlexProfile-Bewegungsgesetze verwendet werden, wenn sichergestellt ist, dass der zu beladene Zug nach jedem Infeed-Step den Stillstand erreicht.

Empfohlene Infeed-Bewegungsgesetze sind:

- Modifizierte Sinoide (ruckarm aber hohe Beschleunigungsspitze) ("VELOCITY_IN_VELOCITY_MOD_SINE")
- Modifiziertes Beschleunigungstrapez (stark ruckbehaftet aber geringste Beschleunigungsspitze) ("X_VELOCITY_IN_VELOCITY_TRAPEZE ALIM")
- Mit dem Bewegungsgesetz "X_FIT_ACC_TRAPEZE JLIM" wird eine ruckbegrenzte Bewegung ausgeführt. Bei diesem Bewegungsgesetz wirkt der Eingang "MB_MBELT_MODEAUTO.Infeed.MaxProductRate" nicht, da die a,v, j Maximalwerte verwendet werden



Der Abstand zwischen dem Produktsensor und der Beladestation sollte kleiner als eine Produktlänge sein. Innerhalb der "StartDelayTime" werden alle weiteren positive Flanken an "NextInfeedStep" ausgeblendet.

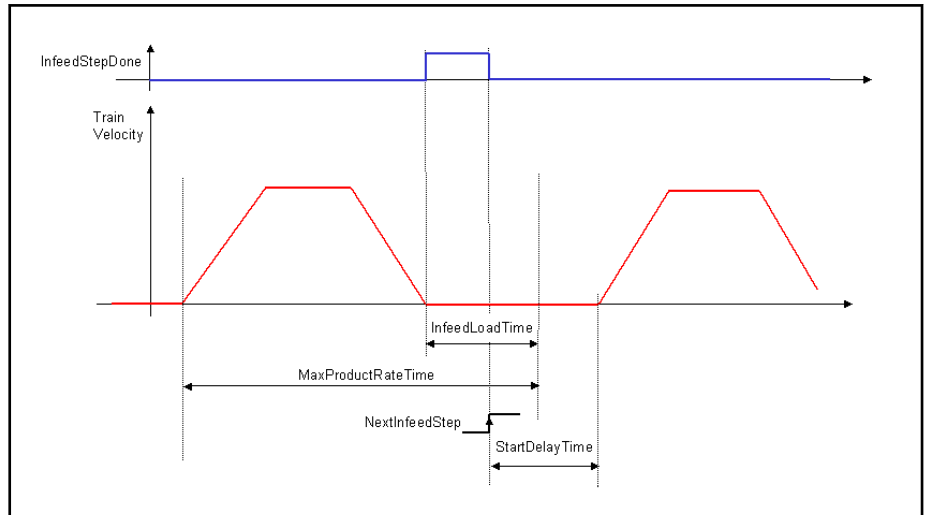


Abb. 4-24: Zusammenhang und Wirkungsweise von InfeedLoadTime, StartDelayTime und MaxProductRate

MaxMoveVelocity, MaxMoveAcceleration, MaxMoveJerk Einstellung

Der beladende Zug muss "schneller" vom Infeed wegfahren als sich der nacheilende Zug am Infeed bewegt. Daher müssen die Eingangsdaten MaxMoveVelocity, MaxMoveAcceleration, MaxMoveJerk wie folgt eingestellt werden:

X_FIT_ACC_TRAPEZE_JLIM

Wird das Bewegungsgesetz **X_FIT_ACC_TRAPEZE_JLIM** bei der Infeed Bewegung verwendet, müssen die **MaxMove...** Eingangswerte wie folgt eingestellt werden:

- MaxMoveVelocity $\geq$ MaxInfeedVelocity
- MaxMoveAcceleration $\geq$ MaxInfeedAcceleration
- MaxMoveJerk $\geq$ MaxInfeedJerk

VELOCITY_IN_VELOCITY_MOD_SINE

Wird das Bewegungsgesetz **VELOCITY_IN_VELOCITY_MOD_SINE** bei der Infeed-Bewegung verwendet, müssen die **MaxMove...** Eingangswerte wie folgt eingestellt werden:

Einstellung von MaxMoveVelocity:

$$\text{MaxMoveVelocity} \left[\frac{\text{mm}}{\text{min}} \right] \geq 1,76 \cdot \frac{S_{\text{max\_infeed}} [\text{mm}]}{\frac{1}{\text{MaxProductRate} [\text{min}^{-1}]} - \frac{\text{InfeedLoadTime} [\text{ms}]}{60000 \left[\frac{\text{ms}}{\text{min}} \right]}}$$

Abb. 4-25: Berechnung von MaxMoveVel
Smax_infeed = Maximale Infeed Schrittweite

Einstellung von MaxMoveAcceleration:

$$\text{MaxMoveAcceleration} \left[\frac{\text{mm}}{\text{s}^2} \right] \geq 5,53 \cdot \frac{S_{\text{max\_infeed}} [\text{mm}]}{\left(\frac{60 \left[\frac{\text{s}}{\text{min}} \right]}{\text{MaxProductRate} [\text{min}^{-1}]} - \frac{\text{InfeedLoadTime} [\text{ms}]}{1000 \left[\frac{\text{ms}}{\text{s}} \right]} \right)^2}$$

Abb. 4-26: Berechnung von MaxMoveAcceleration
Smax_infeed = Maximale Infeed Schrittweite

Einstellung von MaxMoveJerk:

$$\text{MaxMoveJerk} \left[\frac{\text{mm}}{\text{s}^3} \right] \geq 69,47 \cdot \frac{S_{\text{max\_infeed}} [\text{mm}]}{\left(\frac{60 \left[\frac{\text{s}}{\text{min}} \right]}{\text{MaxProductRate} [\text{min}^{-1}]} - \frac{\text{InfeedLoadTime} [\text{ms}]}{1000 \left[\frac{\text{ms}}{\text{s}} \right]} \right)^3}$$

Abb. 4-27: Berechnung von MaxMoveJerk
Smax_infeed = Maximale Infeed Schrittweite

X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM

Wird das Bewegungsgesetz X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM bei der Infeed-Bewegung verwendet, müssen die **MaxMove...** Eingangswerte wie folgt eingestellt werden:

Einstellung von MaxMoveVelocity:

$$\text{MaxMoveVelocity} \left[\frac{\text{mm}}{\text{min}} \right] \geq 60 \left[\frac{\text{s}}{\text{min}} \right] \cdot \left(\frac{t_{\text{move}} \cdot a_{\text{max}}}{2} - \sqrt{\left(\frac{t_{\text{move}} \cdot a_{\text{max}}}{2} \right)^2 - a_{\text{max}} \cdot S_{\text{max\_infeed}} [\text{mm}]} \right)$$

$$a_{\text{max}} = \text{MaxMoveAcceleration} \left[\frac{\text{mm}}{\text{s}^2} \right]$$

$$t_{\text{move}} [\text{s}] = \frac{60 \left[\frac{\text{s}}{\text{min}} \right]}{\text{MaxProductRate} [\text{min}^{-1}]} - \frac{\text{InfeedLoadTime} [\text{ms}]}{1000 \left[\frac{\text{ms}}{\text{s}} \right]}$$

Abb. 4-28: Berechnung von MaxMoveVel
Smax_infeed = Maximale Infeed Schrittweite

Einstellung von MaxMoveAcceleration und MaxMoveJerk:

- MaxMoveAcceleration ≥ MaxInfeedAcceleration
- MaxMoveJerk = 0 (keine Ruckbegrenzung)

Getaktetes Beladen mit Eintaktung in fahrenden Zug, Automatik-Mode, MagicBelt

Unter Vorgabe entsprechender Werte für "MB_MBELT_MODEAUTO.Infeed.StartDelayTime" kann auch erreicht werden, dass sich der Zug bei dem Beladevorgang bewegt. Damit können auch Anwendungen umgesetzt werden, bei denen der Zug unter Bewegung beladen wird (siehe nachfolgende Abbildung).

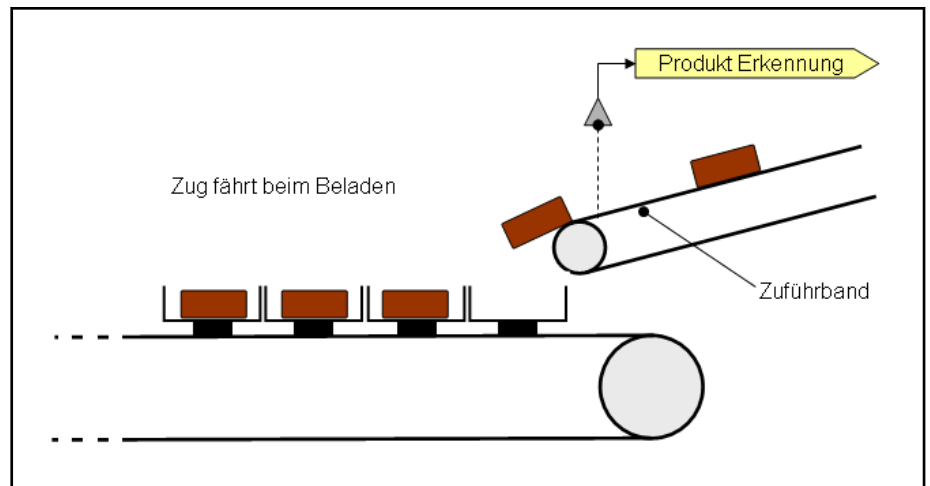


Abb. 4-29: Getaktete Produktzuführung/Beladung bei fahrendem Zug



Typischerweise wird bei der Beladevariante "FEED_INDEXED" der Produktstrom über den Eingang "NextInfeedStep" dem System vorgegeben. Der Zug muss dem vorgegebenen Produkt-Takt folgen. Bei diesen Anwendungen muss typischerweise immer ein Zug am Infeed zur Verfügung stehen um die Produkte aufzunehmen.

Es gibt jedoch auch Anwendungsfälle bei denen die Produkte aufgestaut zur Verfügung stehen, und der Zug die Produkte "anfordert" (Beispiel: Schachteltransport-Applikationen). Der MagicBelt FB gibt dabei die Taktrate über den Ausgang "InfeedStepDone" vor. Bei diesen Anwendungen muss nicht zwangsläufig ein Zug an der Beladeposition stehen. Auch diese Anwendungen können mit dem MagicBelt FB umgesetzt werden. Für diese Anwendungsfälle ist folgende Vorgehensweise zu empfehlen:

Ist die UND-Verknüpfung "InfeedStepDone" UND "Ready" = TRUE, so steht ein Zug am Infeed und fordert ein Produkt an. Das Produkt (z. B. Karton) sollte in das Zug-Fach eingetaktet werden. Ist die Beladung abgeschlossen, so kann mit einer positiven Flanke an "NextInfeedStep" der Zug weitergetaktet werden. Ist die Beladezeit nahezu konstant, so kann die "InfeedLoadTime" entsprechend eingestellt werden, und der Eingang "NextInfeedStep" wie folgt verdrahtet werden:

$\text{NextInfeedStep} := \text{Ready} \text{ AND } \text{InfeedStepDone} \text{ AND } \text{ProductsAvailable}$

Mit dieser Beschaltung taktet der Zug automatisch weiter, solange aufgestaute Produkte ($\text{ProductsAvailable} = \text{TRUE}$) zur Verfügung stehen. Das Signal "ProductsAvailable" wird im Anwendungsprogramm berechnet.

Infeed mit konstanter Geschwindigkeit, Automatik-Mode, MagicBelt

Grundsätzlich erlauben alle "VELOCITY_IN_VELOCITY..." Infeed Bewegungsgesetze (wie z. B. X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM) ein Weitertakten in den nächsten "Infeed Step" wenn sich der Zug noch durch den vorausgegangenen "Infeed Step" bewegt (d. h. Geschwindigkeit > 0).

Werden die positiven Flanken am Eingang "NextInfeedStep" in kürzeren Zeitintervallen als die maximale Produktrate "MB_MBELT_INFEED.MaxProductRate" vorgegeben, so erreicht der Zug den Stillstand nicht mehr (Vorausset-

zung: MB_MBELT_INFEED.InfeedLoadTime = 0). Bei immer kürzeren Zeitintervallen des "NextInfeedStep" werden hierbei die Geschwindigkeitseinbrüche immer kleiner. Wenn die positiven Flanken am Eingang "NextInfeedStep" mit doppelter Produktrate "MB_MBELT_INFEED.MaxProductRate" (oder höher) angelegt werden, so fährt der Zug mit konstanter Geschwindigkeit.

Beispiel:

MB_MBELT_MODEAUTO.Infeed.MaxProductRate = 60 Produkte/Min --> 1 Produkt entspricht 1s --> Der Zug am Infeed bewegt sich mit konstanter Geschwindigkeit wenn alle 0,5s eine positive Flanke an "NextInfeedStep" angelegt wird.

Über das Status-Strukturelement "MB_MBELT_STATUS.MeasuredInfeedStepRate" kann die Frequenz der positiven Flanken am Eingang "NextInfeedStep" für Diagnosezwecke überprüft werden. Im oben genannten Beispiel wird dabei NextInfeedStep = 120 angezeigt. Der zu beladene Zug bewegt sich dann mit einer konstanten Geschwindigkeit die 120 Produkte/Minute entspricht.

Synchrones Beladen, Automatik-Mode, MagicBelt

Beim synchronisieren Beladen wirken folgende Elemente der Datenstruktur MB_MBELT_MODEAUTO.Infeed:

- ProductBuffer
- ProductExpectationWindowStart
- ProductExpectationWindowEnd
- AdditionalSteps
- MinDistanceToOutfeed

Beim synchronen Beladen ist der Zug während dem Beladevorgang mit der Produktzuführung synchronisiert. Die Produktzuführung wird in diesem Fall über ein kontinuierlich drehendes Eintaktrad ("MB_MBELT_GENERAL.InfeedMaster") durchgeführt, welches sich mit jedem Produkt um einen Masterzyklus (= 360°) weiter bewegt. Ein Produktsensor ist auf den Eingang "NextInfeedStep" verdrahtet, und erkennt, ob ein Produkt im Fach des Eintaktrades enthalten ist. Die Produkterkennung ist aktiviert, wenn sich das Produkt im aufgespannten Erwartungsfenster (ProdExpectationWindowStart ... ProdExpectationWindowEnd) im Bezug zur Position der InfeedMasters befindet. Das erkannte Produkt wird in ein Schieberegister eingetragen, welches mit jedem Masterzyklus weitertaktet. Der Zug fährt mit jedem Masterzyklus eine Fachlänge weiter, wenn im Eintaktrad-Fach ein Produkt enthalten ist. Falls kein Produkt enthalten ist, wird der Zug entsprechend aussetzen, damit kein Leerfach entsteht. Bei einem Leerfach kann der Zug nicht genau auf der "InfeedPosition" stehen bleiben, da noch der Bremsweg zurückgelegt werden muss. Der Zug bleibt dann ein halbes Fach später stehen (taktsynchrones Aussetzen). Zwischen Produkterkennung und Zugbeladung können mehrere Produkte enthalten sein. Die Anzahl dieser Produkte wird über "MB_MBELT_MODEAUTO.Infeed.ProductBuffer" dem FB vorgegeben. Auch beim synchronen Beladen sind unterschiedliche Fächergrößen einstellbar, es wird jedoch immer fest von einem Produkt pro Fach ausgegangen. Der optional einstellbare Zugabstand wird mit dem letzten Infeed-Schritt auf die Fachbreite aufaddiert um diesen zu kompensieren.



Im Sonderfall "MB_MBELT_MODEAUTO.INFEED.PRODUCT-BUFFER" = 0 befindet sich die Produkterkennung direkt vor der Zugbeladung. In diesem Fall wird im gleichem Masterzyklus das Produkt erfasst und das notwendige Bewegungsgesetz für den nächsten Masterzyklus vorbereitet. Die Vorbereitung des Bewegungsgesetzes für den nächsten Masterzyklus benötigt 3 sercos Zyklen. Das Ende des Erwartungsfensters "MB_MBELT_INFED.ProdExpectationWindowEnd" darf damit nicht größer als der berechnete Ausgangswert "MB_MBELT_STATUS.MaxProdExpectationWindowEnd" werden. Mit gegebener sercos Zykluszeit und maximaler Geschwindigkeit der InfeedMaster-Achse kann dieser Wert auch extern mit der nachfolgenden Formel berechnet werden:

$$\varphi_{MaxProdWindowEnd} [^\circ] = 360^\circ - 3 \cdot t_{SercosCycle} [ms] \cdot v_{InfeedMaster} [min^{-1}] \cdot 0,006$$

Abb. 4-30: Berechnung MB_MBELT_STATUS.MaxProdExpectationWindowEnd

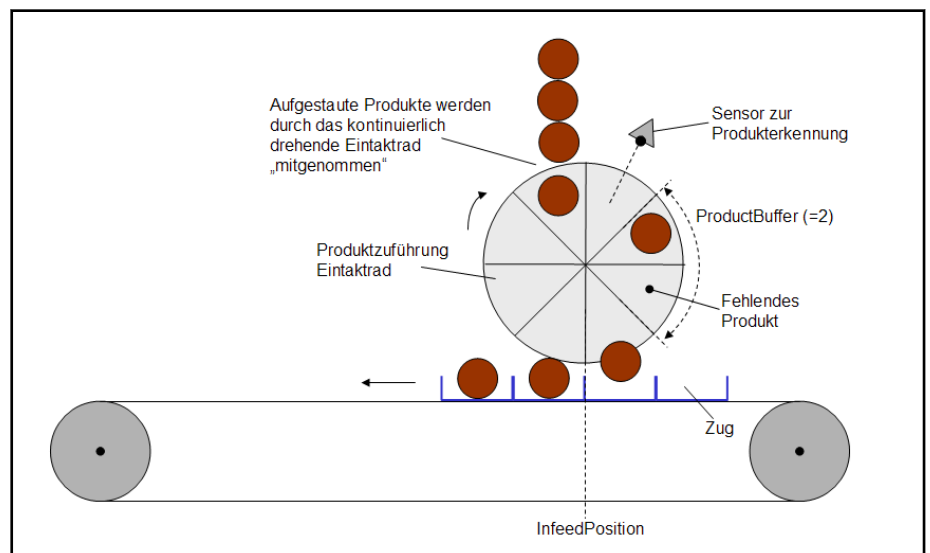


Abb. 4-31: Synchrone Produktzuführung

Beispiel für die synchrone Produktzuführung (zeitlicher Ablauf):

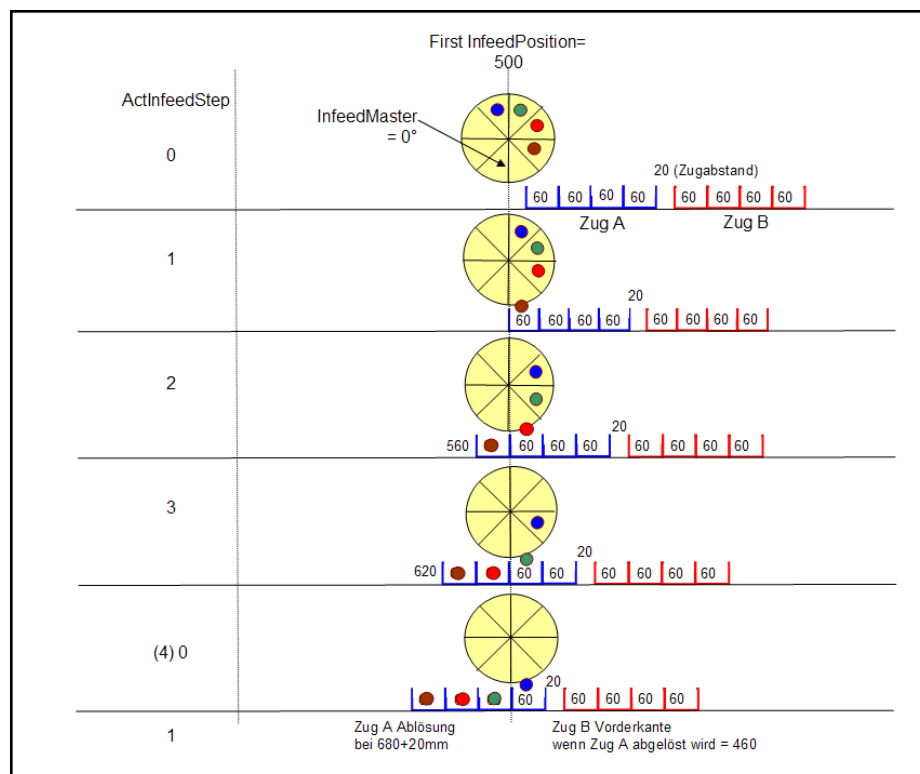


Abb. 4-32: Synchroner Produktzuführung (zeitlicher Ablauf)

Fahrten zwischen Be- und Entladestation, Automatik-Mode, MagicBelt

Nachdem der Zug be- bzw. entladen ist muss er schnellstmöglich die nächste Station anfahren. Die Fahrten zwischen den Be- und Entladestationen werden über die Datenstruktur "MB_MBELT_MOVE" spezifiziert. Hierbei kann für einen beladenen Zug die effektive Beschleunigung und Verzögerung über die Strukturelemente "MB_MBELT_MOVE.LoadTrainAcceleration" und "MB_MBELT_MOVE.LoadTrainDeceleration" herabgesetzt werden.

Weiterhin kann im Bereich der Umlenkungen die Geschwindigkeit reduziert werden um die resultierenden Fliehkräfte zu reduzieren.



Die "MB_MBELT_MOVE" Geschwindigkeit/Beschleunigung muss größer/gleich als die maximale Infeed Geschwindigkeit/Beschleunigung eingestellt werden. Ansonsten besteht Kollisionsgefahr wenn sich ein Zug von der Beladebewegung ablöst. Bei Fehleinstellungen meldet die Dynamiküberwachung einen Fehler um Kollisionen zu vermeiden.

Entladen, Automatik-Mode, MagicBelt

Alle Einstellungen zu dem Entladevorgang werden über das Strukturelement "MB_MBELT_MODEAUTO.Outfeed" vorgegeben.

Zum Entladen fährt zunächst der Zug an die erste absolute Entladeposition ("MB_MBELT_MODEAUTO.Outfeed.FirstOutfeedPosition"). Der FB setzt den Ausgang "TrainReadyToUnload" wenn der Zug die "Outfeedposition" erreicht.

Über eine negative Flanke am FB-Eingang "UnloadingTrain" wird der Zug zur nächsten Ent- bzw. Beladeposition weitergetaktet. Weitere Entladepositionen können über das Strukturelement "MB_MBELT_MODEAUTO.Outfeed.OutfeedDistances" festgelegt werden. Die hier festgelegten Distanzen entsprechen der Entfernung von der letzten Entladeposition. Ist als Distanz = 0 (de-

fault) vorgegeben, so wird nach negativer Flanke am FB Eingang "Unloading-Train" wieder die "FirstInfeedPosition" angefahren.

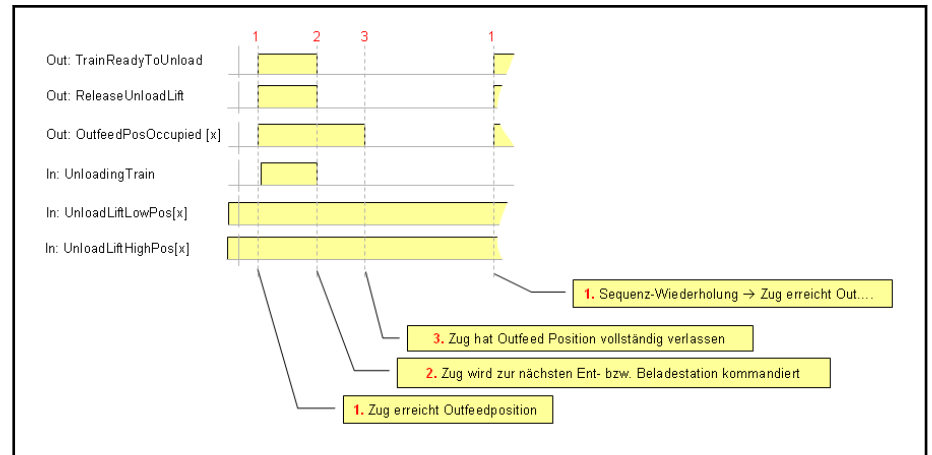


Abb. 4-33: Signal Zeit Diagramm beim Entladen ohne Unload-Handshake



Ab Version 13: Bei der Fahrt auf eine belegte Outfeedposition stoppt der Zug auf einer Warteposition. Der Zug fährt weiter, sobald die Outfeedposition freigegeben wird. Die Warteposition wird durch das Strukturelement "MB_MBELT_MODEAUTO.Outfeed.WaitDistBlockedOutfeed" vorgegeben. Hier wird der gewünschte Abstand zu dem Vorderzug angegeben der die Outfeedposition belegt.

Mit der Einstellung MB_MBELT_MODEAUTO.Outfeed.WaitDistBlockedOutfeed = 0 wird keine Warteposition angefahren (Verhalten wie bei Version 12 und vorher).

Wird der Zug über einen Entlade-Rechen entladen, so müssen die Bewegung des Zuges und des Entlade-Rechens verriegelt werden um Kollisionen zwischen Zug und Rechen zu vermeiden. Das Entlade-Handshake wird über das Strukturelement "MB_MBELT_MODEAUTO.Outfeed.EnableUnloadHandshake" aktiviert.

Im einfachsten Fall wird der Zug erst bei eingezogenem Entlade-Rechen zur nächsten Ent- bzw. Beladeposition weitergetaktet. Das entsprechende Handshake und das Signal-Zeit Diagramm zu diesem Fall sind in der nachfolgenden Abbildung dargestellt.

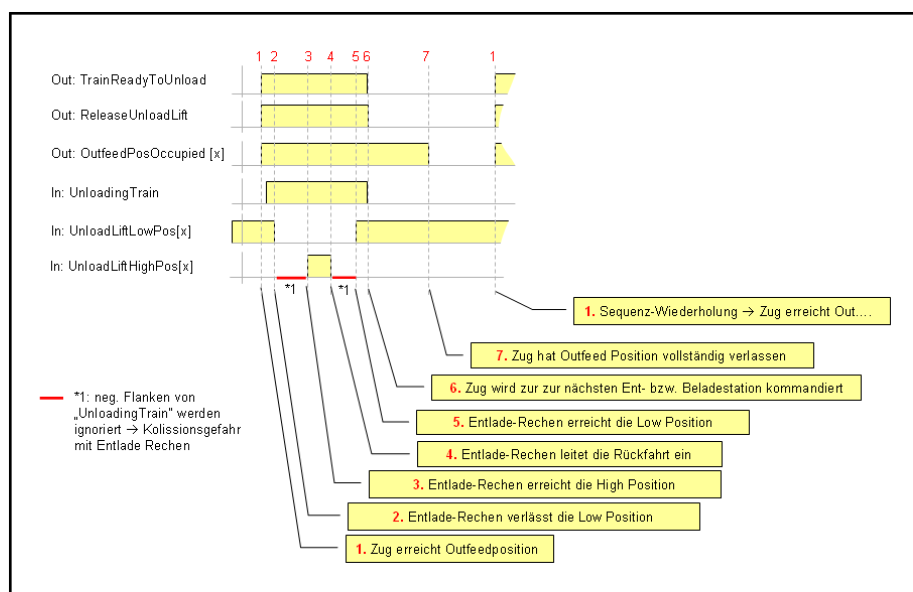


Abb. 4-34: Signal-Zeit-Diagramm beim Entladen mit Handshake/Zug-Weiterfahrt bei eingefahrenem Entlade-Rechen

In einem weiteren Anwendungsfall wird der Zug schon bei vollständig ausgefahrenem Entlade-Rechen weitergetaktet. Der Entlade-Rechen darf dann erst wieder zurückfahren wenn der Zug vollständig aus dem Durchdringungsbe-
reich des Rechens gefahren ist. Die Entladeposition ist solange belegt, bis der Entlade-Rechen wieder vollständig zurückgefahren ist. Das entsprechende Handshake und das Signal-Zeit-Diagramm zu diesem Fall sind in der nachfolgenden Abbildung dargestellt.

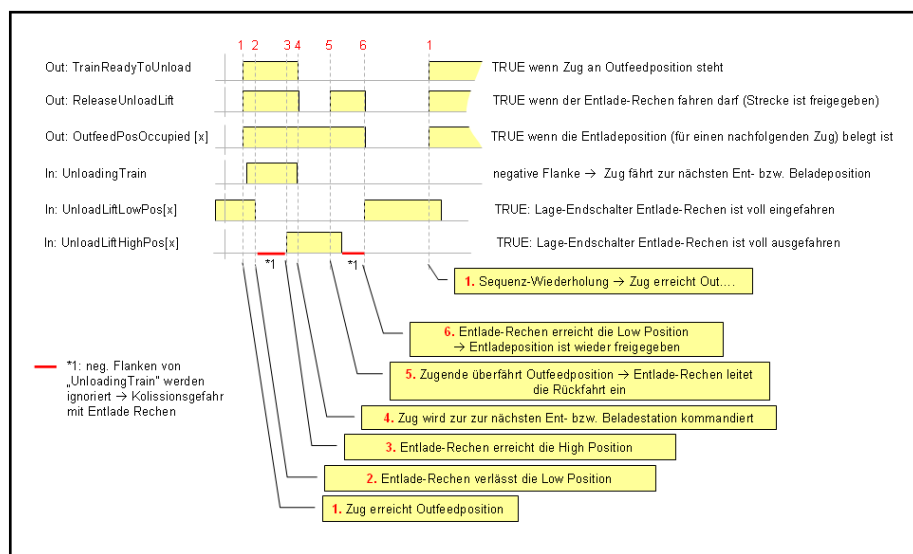


Abb. 4-35: Signal-Zeit-Diagramm beim Entladen mit Handshake/Zug-Weiterfahrt bei ausgefahrenem Entlade-Rechen

Stopp, Automatik-Mode, MagicBelt

Über eine positive Flanke am "Stop"-Eingang wird der Zug gemäß spezifizierter Stoppvariante stillgesetzt ("MB_MBELT_MODEAUTO. StopType"). Bei der synchronen Infeed Variante wird empfohlen, das Anhalten der Ketten durch Verzögern des InfeedMasters durchzuführen und anschließend die Ketten über den StopType = STOP_IMMEDIATE stillzusetzen.

STOP_IMMEDIATE:

Bei dieser Stoppvariante werden alle Züge und die InfeedAxis sofort über die Rampe "MB_MBELT_GENERAL.StopRamp" verzögert (z. B. für schwerwiegende Störungen).

STOP_CYCLE_COMPLETE:

Bei dieser Stoppvariante wird der aktive Produkt-Zyklus (Infeed Step) noch ausgeführt. Danach wird angehalten (definiertes Stillsetzen). Alle Züge die nicht mit der InfeedAxis synchronisiert sind (z. B. bei Fahrten zwischen Outfeedpositionen) werden sofort gestoppt.

STOP_IMMEDIATE_SYNC:

Diese Stoppvariante ist nur für Anwendungen mit synchronisiertem Infeed sinnvoll. Die Stoppvariante verhält sich wie STOP_IMMEDIATE mit dem Unterschied, dass die Achsen, die im Synchronverbund mit dem InfeedMaster stehen, weiter synchronisiert bleiben. Diese Achsen werden durch das Verzögern des externen InfeedMasters mit verzögert. Das Verzögern des InfeedMasters muss außerhalb des FB sichergestellt werden.

Fehlerreaktion

Im Fehlerfall werden alle Achsen über die "MB_MBELT_GENERAL.StopRamp" sofort angehalten, sofern sie noch den Sollwerten folgen können (Verhalten wie bei "STOP_IMMEDIATE"). Weiterhin wird der "Error"-Ausgang gesetzt, und Fehlercodes an den entsprechenden FB-Ausgängen ausgegeben. Die Fehlerquittierung kann durch eine steigende Flanke am "ClearError"-Eingang erfolgen. Je nach gewähltem Start-Typ kann nach Löschen des Fehlers der Automatik-Betrieb an der unterbrochenen Stelle ("MB_MBELT_STARTTYPE" = START_AT_LAST_STEP), oder von der definierten Grundstellung ("MB_MBELT_STARTTYPE" = START_AT_BEGIN) fortgesetzt werden.

Überwachungen, Automatik-Mode, MagicBelt

Produktstau-Überwachung

Durch unplanmäßige Verzögerungen (z. B. beim Entladen) können Situationen entstehen bei denen der kontinuierliche Produktstrom nicht mehr aufgenommen werden kann, und folglich ein Produktstau entstehen kann. Solche Situationen müssen möglichst frühzeitig erkannt werden, um den Produktstrom zu drosseln oder zu unterbrechen.

Im ersten Zug (Zug A) ist daher eine Produktstau-Überwachung aktiv, die den "Ready"-Ausgang setzt wenn folgende Bedingung eintritt:

- Es befindet sich ein Zug an der Beladestation

UND

*Für den Fall **ohne** Entladevorrichtung ("MB_MBELT_INFEED.EnableUnload-Handshake" = FALSE):*

- Distanz zwischen gerade beladenem Zug (Zug in "Infeed"-Zustand) und weiteren Zügen an den "Outfeed"-Positionen ist größer als "MB_MBELT_INFEED.MinDistanceToOutfeed"

*Für den Fall **mit** Entladevorrichtung ("MB_MBELT_INFEED.EnableUnload-Handshake" = TRUE):*

- Distanz zwischen gerade beladenem Zug (Zug in "Infeed"-Zustand) und als belegt gekennzeichneten "Outfeed"-Positionen (siehe "MB_MBELT_STATUS.OutfeedPosOccupied") ist größer als "MB_MBELT_INFEED.MinDistanceToOutfeed". Ist eine "Outfeed"-Position belegt, wird eine feste Schranke an die Position ("Outfeed"-Position – Zuglänge) gelegt. Solange der Zug im "Infeed"-Zustand mit der Distanz "MB_MBELT_INFEED.MinDistanceToOutfeed" von dieser Schranke entfernt ist, wird der "Ready"-Ausgang gesetzt.

Sind die Bedingungen nicht erfüllt, wird der "Ready"-Ausgang zurückgesetzt. Bei zurückgesetztem "Ready"-Ausgang sollte die vorgeschaltete Station den Produktstrom unterbrechen (z. B. Produkte ausblasen, wenn "Ready" = FALSE)

Bleibt der Produktstrom auch bei "Ready"=FALSE anstehen, so wird überwacht ob der Zug im "Infeed"-Zustand mit dem Zug an der/den belegten "Outfeed"-Position/Positionen kollidieren würde. Hierzu wird die "Stop"-Rampe mit aktueller Zug-Geschwindigkeit und Verzögerung ("MB_MBELT_GENERAL.StopRamp") genutzt. Wird eine Kollision erkannt, wird der betroffene Zug mit der "Stopp"-Rampe sofort stillgesetzt und ein Fehler ausgegeben.

Bei der Ermittlung der Zug-Distanzen wird der Fall mit/ohne Entladevorrichtung unterschieden:

- **Ohne** Entladevorrichtung ("MB_MBELT_INFEED.EnableUnloadHandshake" = FALSE) wird die Distanz zwischen gerade beladenem Zug (Zug in "Infeed"-Zustand) und weiteren Zügen an den "Outfeed"-Positionen überwacht
- **Mit** Entladevorrichtung wird die Distanz zwischen gerade beladenem Zug (Zug in "Infeed"-Zustand) und der als belegt gekennzeichneten "Outfeed"-Positionen überwacht. Ist eine "Outfeed"Position belegt, wird eine feste Schranke an die Position ("Outfeed"-Position – Zuglänge) gelegt. Würde der Infeed-Zug mit seiner Verzögerungsrampe "MB_MBELT_GENERAL.StopRamp" über diese Schranke hinausfahren, wird eine Kollision erkannt und behandelt

Kollisionsüberwachung

Die Kollisionsüberwachung wird automatisch nach Start des FB aktiviert und läuft permanent zyklisch. Die Kollisionsüberwachung ist zweistufig aufgebaut:

In Betriebszuständen mit großen Abständen zum Vorderzug (wie z. B. Fahren auf freier Strecke) muss der Zugabstand größer als der Bremsweg auf den Vorderzug + Sicherheitsabstand "MB_MBELT_GENERAL.SafetyDistance" sein. Die Kollisionsüberwachung löst aus, wenn diese Bedingung nicht erfüllt ist.

In Betriebszuständen mit sehr geringen Abständen zum Vorderzug (wie z. B. das Aufschließen auf einen Vorderzug) muss der Zugabstand größer als der Sicherheitsabstand MB_MBELT_GENERAL.SafetyDistance sein. Die Kollisionsüberwachung löst aus, wenn diese Bedingung nicht erfüllt ist.



Die Kollisionsüberwachung basiert systembedingt auf Lagesollwerten. Ein Überspringen der Züge durch ungünstig parametrisierte Lageregler kann zu Kollisionen führen.

Die Kollisionsüberwachung kann durch den Eingang **DisableCollisionDetection** = TRUE deaktiviert werden. Diese Funktion sollte jedoch ausschließlich zum Auseinanderfahren der Züge nach einer Kollision verwendet werden. Sobald die Züge auseinander gefahren sind, ist die Kollisionsüberwachung wieder zu aktivieren.

Dynamiküberwachung

Die "MagicBelt"-Applikation ist so einzustellen, dass die Bewegungsdynamik (z. B. Beschleunigung, Geschwindigkeit) beim Fahren zwischen "In- und Outfeed"-Positionen mit größeren oder gleichen Werten im Bezug zur "In- bzw. Outfeed"-Bewegung eingestellt ist. Nur so kann sichergestellt werden, dass ein Zug der eine "In- bzw. Outfeed"-Bewegung ausführt nicht mit einem langsameren Vorderzug kollidiert. Der FB generiert einen Fehler und verzögert alle Achsen mit der Rampe "MB_MBELT_GENERAL.StopRamp", falls die Dynamikdaten nicht den o. g. Regeln entsprechen.

Entladeüberwachung

Bei der Produktentnahme mit Greifer können im Fehlerfall Produkte im Zug verbleiben. Die Erkennung der unvollständigen Produktentnahme muss - bei

- Bedarf - außerhalb des "MagicBelt"-FB erfolgen. Wird ein solcher Fall extern erkannt, können über den Eingang "Stop" die Ketten stillgesetzt werden. Nach positiver Flanke von "Start" wird der Betrieb wieder fortgesetzt.
- Fehlerbehandlung** Im Fehlerfall werden grundsätzlich alle Züge mit Rampe stillgesetzt. Es wird keine Unterscheidung getroffen, welcher Zug den Fehler verursachte.

Wiederanlauf, Automatik-Mode, MagicBelt

Nach Beseitigung der Fehlerursache kann der Fehler durch eine positive Flanke am Eingang **ClearError** gelöscht werden. Über eine positive Flanke an "Start" starten die Züge gemäß vorgegebener Startoption. Mit der Startoption "START_AT_LAST_STEP" kann z. B. der Zustand vor der Unterbrechung wiederhergestellt werden, und z. B. ein noch beladener Zug wieder weiter auf die Entladeposition gefahren werden.

4.4.5 Inbetriebnahmehinweise, MagicBelt

Die folgenden Punkte müssen zur Inbetriebnahme des Funktionsbausteins "MB_MagicBeltType01" berücksichtigt werden:

- Pro Kette muss jeweils eine eigene Instanz des Funktionsbausteins aufgerufen werden
- Der Funktionsbaustein "MB_MagicBeltType01" muss in einer **sercos synchronen Task** zyklisch aufgerufen werden. Der Task-Watchdog muss eingeschaltet sein (Empfindlichkeit = 1)
- Die Achsen des Kettenantriebs **Chain A, B, (C)** müssen mit IndraWorks angelegt werden

Hierbei ist auf folgende Parametrierung zu achten:

- Achstyp: real oder virtuell (für Testzwecke)
 - "Interpolation auf Antrieb" bei Hinzufügen des Antriebes **muss ausgewählt werden** (Lagesollwertgenerierung erfolgt auf der Steuerung)
 - Wichtungsart "**Translatorisch, Modulo, Einheit [mm]**" muss eingestellt werden
 - Der Modulowert muss wie folgt eingestellt werden: **Modulowert = ChainLength/TrainsPerChain**
 - Die **InfeedAxis** muss mit IndraWorks angelegt werden
- Hierbei ist auf folgende Parametrierung zu achten:*
- Achstyp: **virtuell**
 - Wichtungsart "**Translatorisch, Modulo, Einheit [mm]**" muss eingestellt werden
 - Der Modulowert muss wie folgt eingestellt werden: **Modulowert = ChainLength/TrainsPerChain**
 - Die **DummyMaster**-Achse muss in IndraWorks angelegt werden, falls keine andere zur Verfügung steht. Hierbei sind alle Achstypen zugelassen. Es ist keine besondere Parametrierung erforderlich, da diese Achse nur passiv genutzt wird
 - Bei synchronem Beladen muss zusätzlich die **InfeedMaster**-Achse mit IndraWorks angelegt werden

Hierbei ist auf folgende Parametrierung zu achten:

- Es können grundsätzlich alle Achstypen (Real, Virtuell, Encoder...) verwendet werden
- Wichtungsart "**translatorisch, rotatorisch**" muss eingestellt werden

– Modulwert = **360°**

- Die in IndraWorks angelegten Achsen müssen dann über die Datenstruktur [MB_MBELT_GENERAL](#), Seite 200, den entsprechenden Achsen (wie z. B. "ChainAxes") zugeordnet werden (über AxisRef).

4.5 Schlaufenregelung

4.5.1 Einführung und Übersicht

Schlaufenregelungen werden zur Entkoppelung von Anlagenteilen eingesetzt.

Ein typischer Anwendungsfall ist nachstehende Bandanlage.

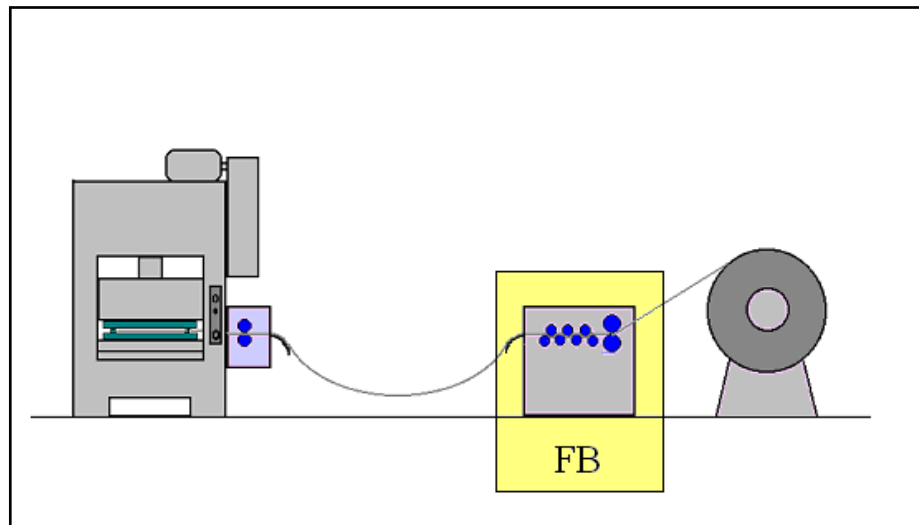


Abb. 4-36: Bandanlage, Prinzipdarstellung

Die Schlaufe befindet sich zwischen den Vorschubwalzen der Stanzmaschine und dem vorgeschalteten Materialzufuhrsystem. Sie soll die beiden Bewegungssysteme (Start/Stop bzw. kontinuierlich) entkoppeln. Der Schlaufeninhalt hängt dabei von Walzenvorschubdaten ab.

Im Normalbetrieb läuft der eine Bereich der Bandanlage kontinuierlich mit einer vom Bediener vorgegebenen durchschnittlichen Geschwindigkeit v .

Der andere Anlagenbereich läuft im Start-Stop-Betrieb. Während der Blechbearbeitung geht dieser Teil der Anlage in Stopp und der Schlaufeninhalt wächst an. Wenn das Material für die nächste Bearbeitung nachgeführt wird, baut sich der Schlaufeninhalt wieder ab. In einem Arbeitstakt bewegt sich die Schlaufe immer zwischen den Min/Max-Kontakten.

Voraussetzung für den Einsatz des FB zur Schlaufenkontrolle:

Beim Einrichten der Bandanlage wird das neue Blechband vom Coil durch die Richtmaschine zur Stanzmaschine dergestalt geführt, dass bereits eine Schlaufe zwischen den Min/Max-Kontakten (oder Min/Max-Analogwerte) existiert.

- [ML_SagControlC](#), Seite 231, regelt die Schlaufe mit Min/Max-Kontakten.
- [ML_SagControlA](#), Seite 234, regelt die Schlaufe mit analogen Schaltsignalen.

4.5.2 ML_SagControlC

Kurzbeschreibung

Der Funktionsbaustein "ML_SagControlC" enthält einen Schlaufenregler mit binären Min/Max-Kontakten.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechMotion

Tab. 4-31: Referenztable ML_SagControlA

Schnittstellenbeschreibung

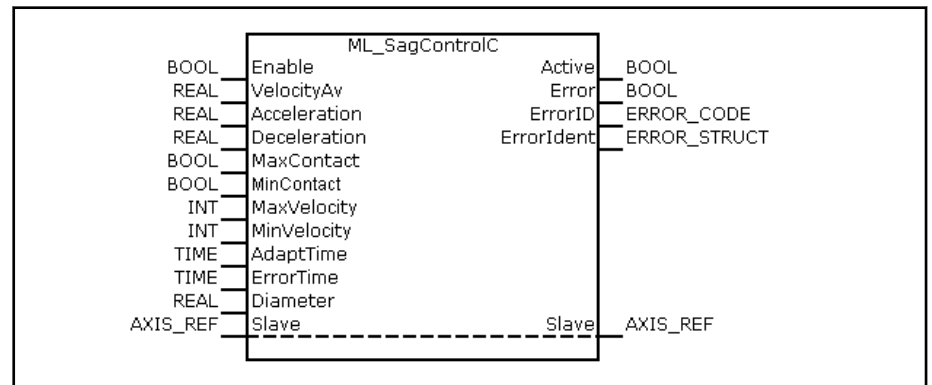


Abb. 4-37: Funktionsbaustein ML_SagControlC

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert/deaktiviert die Geschwindigkeitsregelung
	VelocityAv	REAL	Durchschnittsgeschwindigkeit der Bandanlage
	Acceleration	REAL	Beschleunigung für Geschwindigkeitsänderungen
	Deceleration	REAL	Verzögerung für Geschwindigkeitsänderungen
	MaxContact	BOOL	Max-Kontakt der Schlaufe
	MinContact	BOOL	Min-Kontakt der Schlaufe
	MaxVelocity	INT	Maximale Geschwindigkeit in % bezogen auf die Durchschnittsgeschwindigkeit der Bandanlage
	MinVelocity	INT	Minimale Geschwindigkeit in % bezogen auf die Durchschnittsgeschwindigkeit der Bandanlage
	AdaptTime	TIME	Adaptionszeit (Ta)
	ErrorTime	TIME	Fehlerzeit (Te)
	Diameter	REAL	Coildurchmesser ⇒ Bewertung der Geschwindigkeit in %. Ist der Eingang nicht beschaltet, wird der Default-Wert 100% gesetzt
VAR_OUTPUT	Active	BOOL	Schlaufenregelung arbeitet
	Error	BOOL	Signalisiert einen Fehler. Mit "Enable"= FALSE wird Fehler gelöscht
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung

E/A-Typ	Name	Datentyp	Beschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung in Abhängigkeit von der Fehlertabelle Fehlerbehandlung, Seite 232
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Slave-Achse

Tab. 4-32: Schnittstellenvariablen Funktionsbaustein ML_SagControlC

Funktionsbeschreibung

Der "Enable"-Eingang aktiviert/deaktiviert die Geschwindigkeitsregelung. Die erste aktive Geschwindigkeit ist die vom Bediener vorgegebene Durchschnittsgeschwindigkeit der Anlage.



Wird der "Enable"-Eingang zurückgesetzt, geht der zu regelnde Antrieb in Stop.

Der Funktionsbaustein für die Schleifenregelung wirkt ausschließlich auf den Antrieb für die Materialzufuhr. Hierbei wird versucht, die Schleife in einem definierten Fenster zu halten (Min/Max-Wert), indem je nach Abweichung die aktuelle Geschwindigkeit der Bandanlage angepasst wird, optional mit einem Sensor, der den Coildurchmesser erfasst.

Bei Erreichen des Min-Kontaktes wird die Geschwindigkeit des Materialzufuhrsystems auf die am Baustein anliegende Maximalgeschwindigkeit erhöht, um eine größere Schleife aufzubauen. Wird der Min-Kontakt innerhalb der Fehlerreaktionszeit "Te" wieder in Richtung Max-Kontakt verlassen, so wird noch für die Nachhaltezeit "Ta" mit der erhöhten Geschwindigkeit gefahren, um dann wieder mit der vorgegebenen Durchschnittsgeschwindigkeit weiterzufahren. Vergrößert sich die Schleife so weit, dass der Max-Kontakt anspricht, so wird die Geschwindigkeit auf die am Baustein anliegende Minimalgeschwindigkeit reduziert. Bei Verlassen des Max-Kontaktes innerhalb der Fehlerreaktionszeit "Te" fährt der Antrieb noch für die Zeit "Ta" mit reduzierter Geschwindigkeit weiter. Danach wird wieder mit der vorgegebenen Durchschnittsgeschwindigkeit weitergefahren (siehe v-t Diagramm).

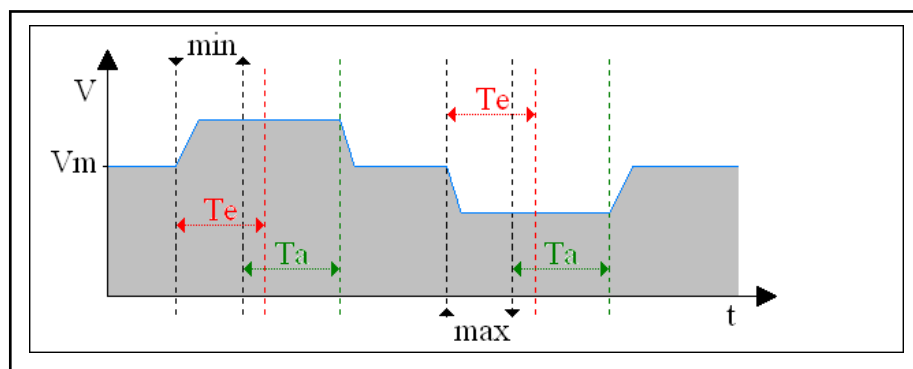


Abb. 4-38: Bausteinverhalten, fehlerfrei

Fehlerbehandlung

Steht der Min-Kontakt für eine Zeit größer "Te" immer noch an, so wird ein Fehler generiert und mit gleicher Geschwindigkeit weitergefahren. Dieser Zustand bleibt so lange bestehen, bis der Kontakt wieder verlassen wird. Hiernach fährt der Antrieb noch für die Nachhaltezeit "Ta" mit der erhöhten Geschwindigkeit und danach mit der vorgegebenen Durchschnittsgeschwindigkeit. Bleibt hingegen der Max-Kontakt für die Zeit "Te" anstehen, so wird die Geschwindigkeit auf Null reduziert. Schaltet der Max-Kontakt wieder, so bleibt der Antrieb weiter für die Zeit "Ta" stehen, um dann wieder auf die vor-

gegebenen Durchschnittsgeschwindigkeit "Vm" zu beschleunigen und mit dieser Geschwindigkeit weiterzufahren.

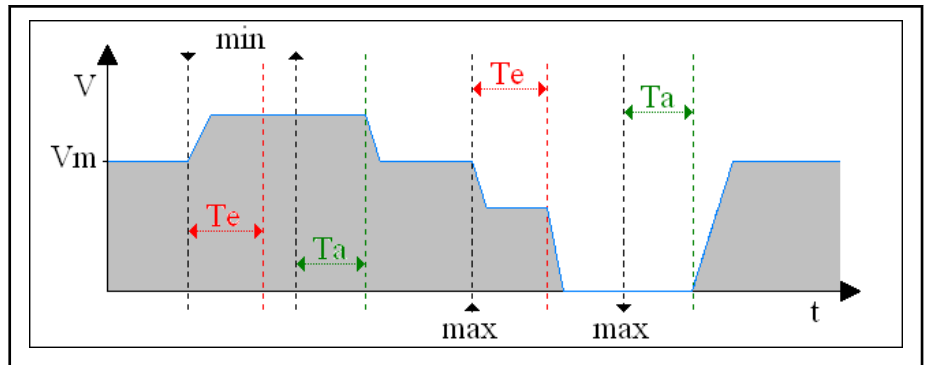


Abb. 4-39: Bausteinverhalten, fehlerhaft

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle F_RELATED_TABLE, 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Eingänge außerhalb des Gültigkeitsbereichs: Geschwindigkeit ≤ 0 Beschleunigung ≤ 0 Verzögerung ≤ 0 Max. Geschw. $\leq$ Min. Geschw. Adaptionszeit ≤ 0 Fehlerreaktionszeit ≤ 0 Coildurchmesser ≤ 0
SYSTEM_ERROR (16#7FFF)	16#0501	16#0000	Geschwindigkeitsadaption erfolglos, Adaptionszeit und Fehlerzeit abgelaufen, Max-Kontakt steht noch an
SYSTEM_ERROR (16#7FFF)	16#0502	16#0000	Geschwindigkeitsadaption erfolglos, Adaptionszeit und Fehlerzeit abgelaufen, Min-Kontakt steht noch an
SYSTEM_ERROR (16#7FFF)	16#0503	16#0000	Max/Min-Kontakt stehen zur gleichen Zeit an. Kontaktproblem ?
SYSTEM_ERROR (16#7FFF)	16#0504	16#0000	Die vorgegebene Geschwindigkeit konnte nicht erreicht werden. Geschwindigkeitsvorgaben überprüfen

Tab. 4-33: Fehlercodes Funktionsbaustein ML_SagControlC

Firm-, Soft- und Hardware-Anforderungen

Folgende Firm-, Soft- und Hardware wird benötigt:

- Hardware - IndraDrive C oder M
- Firmware - Antriebsfirmware (Funktionspaket Closed Loop)
- Software - IndraWorks Engineering 14VRS

Applikationsbeispiel 1 In der einfachsten Form sind es 2 Kontakte, ein Min- und ein Max-Kontakt, die als Regelparameter zur Verfügung stehen.

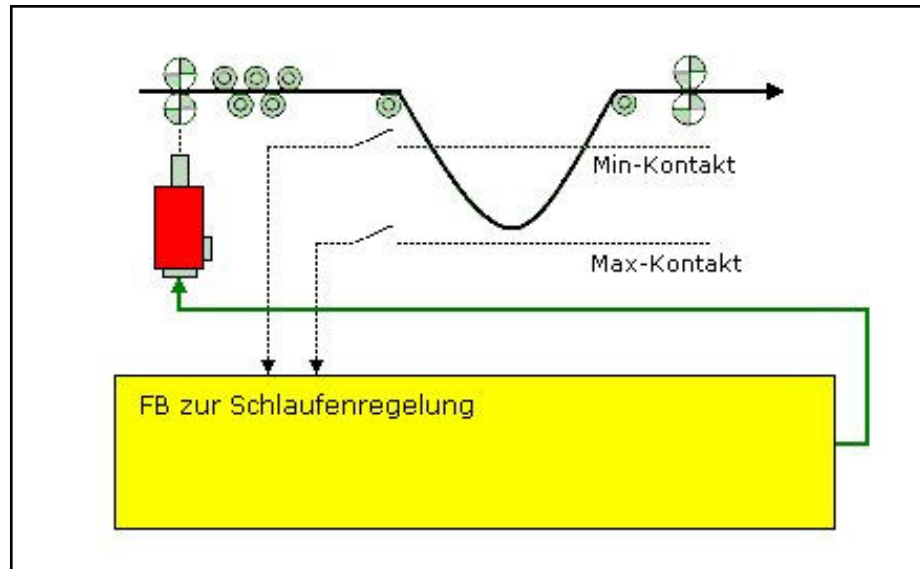


Abb. 4-40: Anwendung 1, ML_SagControlC

Im Normalbetrieb läuft die Bandanlage mit einer durch den Bediener vorgegebenen durchschnittlichen Geschwindigkeit "Vm". Der FB für die Schlaufenregelung versucht nun, die Schlaufe in einem definierten Fenster zu halten (Min/Max-Kontakt), indem je nach Abweichung die aktuelle Geschwindigkeit der Bandanlage angepasst wird.

Applikationsbeispiel 2 Bei dieser Variante ist der zu regelnde Antrieb der Abwickler (Coil) selbst. Hierbei kommt neben den Min/Max-Kontakten auch der abnehmende Coildurchmesser als Regelgröße hinzu.

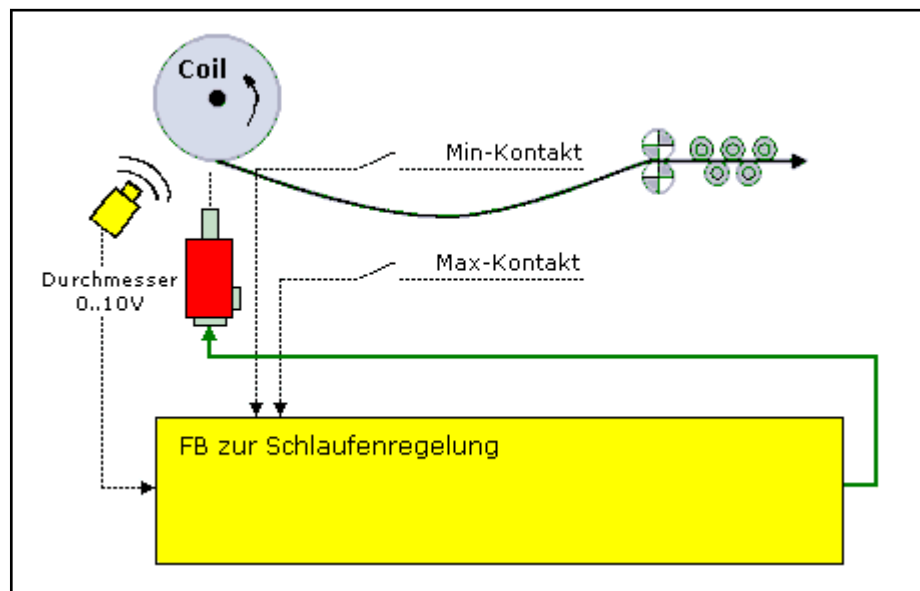


Abb. 4-41: Anwendung 2, ML_SagControlC

4.5.3 ML_SagControlA

Kurzbeschreibung Der Funktionsbaustein "ML_SagControlA" enthält einen Schlaufenregler mit analoger Istwerterfassung der Schlaufe.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechMotion

Tab. 4-34: Referenztable ML_SagControlA

Schnittstellenbeschreibung

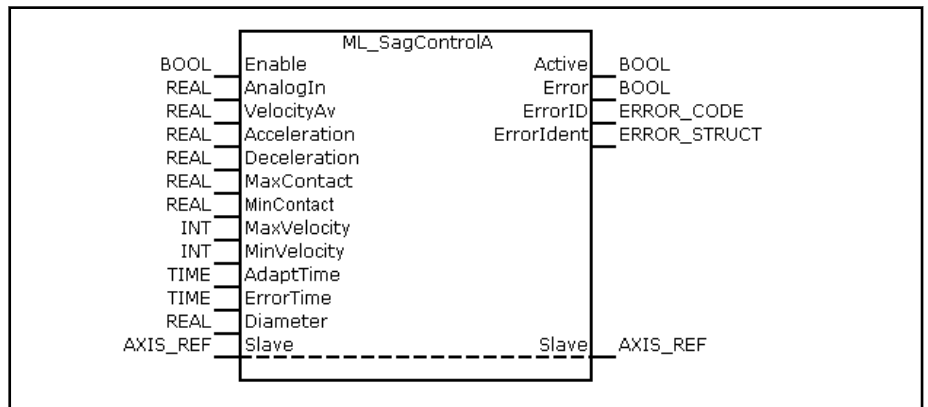


Abb. 4-42: Funktionsbaustein ML_SagControlA

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert / deaktiviert die Geschwindigkeitsregelung
	AnalogIn	REAL	Analogwert der Schleife (0..10)
	VelocityAv	REAL	Durchschnittsgeschwindigkeit der Bandanlage
	Acceleration	REAL	Beschleunigung für Geschwindigkeitsänderungen
	Deceleration	REAL	Verzögerung für Geschwindigkeitsänderungen
	MaxContact	REAL	Max-Kontakt Schleife
	MinContact	REAL	Min-Kontakt Schleife
	MaxVelocity	INT	Maximale Geschwindigkeit in % bezogen auf die Durchschnittsgeschwindigkeit der Bandanlage
	MinVelocity	INT	Minimale Geschwindigkeit in % bezogen auf die Durchschnittsgeschwindigkeit der Bandanlage
	AdaptTime	TIME	Adaptionszeit (Ta)
	ErrorTime	TIME	Fehlerzeit (Te)
	Diameter	REAL	Coildurchmesser ⇒ Bewertung der Geschwindigkeit in %. Ist der Eingang nicht beschaltet, wird der Default-Wert 100% gesetzt
VAR_OUTPUT	Active	BOOL	Schlaufenregelung arbeitet
	Error	BOOL	Signalisiert einen Fehler. Mit "Enable"= FALSE wird Fehler gelöscht
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung

E/A-Typ	Name	Datentyp	Beschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung in Abhängigkeit von der Fehlertabelle Fehlerbehandlung, Seite 236
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Slave-Achse

Tab. 4-35: Schnittstellenvariablen Funktionsbaustein ML_SagControlA

Funktionsbeschreibung

Der Enable-Eingang aktiviert/deaktiviert die Geschwindigkeitsregelung. Die erste aktive Geschwindigkeit ist die vom Bediener vorgegebene Durchschnittsgeschwindigkeit der Anlage.



Wird der Enable-Eingang zurückgesetzt, geht der zu regelnde Antrieb in Stopp.

Der Funktionsbaustein für die Schleifenregelung wirkt ausschließlich auf den Antrieb für die Materialzufuhr. Hierbei wird versucht, die Schleife in einem definierten Fenster zu halten (Analogwert-Sensor), indem je nach Abweichung die aktuelle Geschwindigkeit der Bandanlage angepasst wird, optional mit einem Sensor, der den Coildurchmesser erfasst.

Bei Erreichen des Min-Kontaktes wird die Geschwindigkeit des Materialzufuhrsystems auf die am Baustein anliegende Maximalgeschwindigkeit erhöht, um eine größere Schleife aufzubauen. Wird der Min-Kontakt innerhalb der Fehlerreaktionszeit "Te" wieder in Richtung Max-Kontakt verlassen, so wird noch für die Nachhaltezeit "Ta" mit der erhöhten Geschwindigkeit gefahren, um dann wieder mit der vorgegebenen Durchschnittsgeschwindigkeit weiterzufahren. Vergrößert sich die Schleife so weit, dass der Max-Kontakt anspricht, so wird die Geschwindigkeit auf die am Baustein anliegende Minimalgeschwindigkeit reduziert. Bei Verlassen des Max-Kontaktes innerhalb der Fehlerreaktionszeit "Te" fährt der Antrieb noch für die Zeit "Ta" mit reduzierter Geschwindigkeit weiter. Danach wird wieder mit der vorgegebenen Durchschnittsgeschwindigkeit weitergefahren (siehe v-t Diagramm).

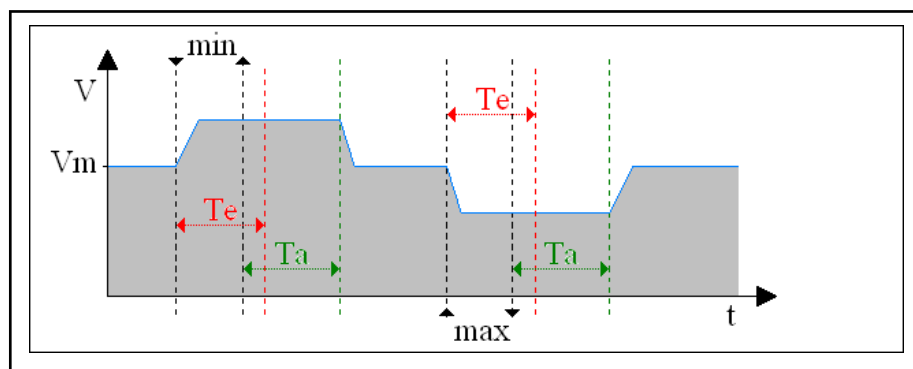


Abb. 4-43: Bausteinverhalten, fehlerfrei

Fehlerbehandlung

Steht der Min-Kontakt für eine Zeit größer "Te" immer noch an, so wird ein Fehler generiert und mit gleicher Geschwindigkeit weitergefahren. Dieser Zustand bleibt so lange bestehen, bis der Kontakt wieder verlassen wird. Hiernach fährt der Antrieb noch für die Nachhaltezeit "Ta" mit der erhöhten Geschwindigkeit und danach mit der vorgegebenen Durchschnittsgeschwindigkeit. Bleibt hingegen der Max-Kontakt für die Zeit "Te" anstehen, wird die Geschwindigkeit auf Null reduziert. Schaltet der Max-Kontakt wieder, bleibt der Antrieb weiter für die Zeit "Ta" stehen, um dann wieder auf die vorgegebenen

Durchschnittsgeschwindigkeit "Vm" zu beschleunigen und mit dieser Geschwindigkeit weiterzufahren.

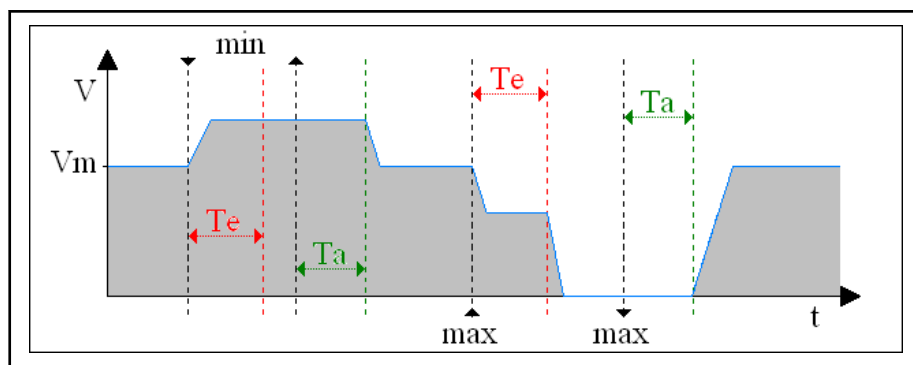


Abb. 4-44: Bausteinverhalten, fehlerhaft

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle F_RELATED_TABLE, 16#0170 folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Eingänge außerhalb des Gültigkeitsbereichs: Geschwindigkeit ≤ 0 Beschleunigung ≤ 0 Verzögerung ≤ 0 Max. Geschw. $\leq$ Min. Geschw. Adaptionszeit ≤ 0 Fehlerreaktionszeit ≤ 0 Coil Durchmesser ≤ 0
SYSTEM_ERROR (16#7FFF)	16#0501	16#0000	Geschwindigkeitsadaption erfolglos, Adaptionszeit und Fehlerzeit abgelaufen, Max-Kontakt steht noch an
SYSTEM_ERROR (16#7FFF)	16#0502	16#0000	Geschwindigkeitsadaption erfolglos, Adaptionszeit und Fehlerzeit abgelaufen, Min-Kontakt steht noch an
SYSTEM_ERROR (16#7FFF)	16#0504	16#0000	Die vorgegebene Geschwindigkeit konnte nicht erreicht werden. Geschwindigkeitsvorgaben überprüfen

Tab. 4-36: Fehlercodes Funktionsbaustein ML_SagControlA

Firm-, Soft- und Hardware-Anforderungen

Folgende Firm-, Soft- und Hardware wird benötigt:

- Hardware - IndraDrive C oder M
- Firmware - Antriebsfirmware (Funktionspaket Closed Loop)
- Software - IndraWorks Engineering 14VRS

Applikationsbeispiel 1

In dieser Variante bildet ein Sensor die Schlaufenlage als analogen Eingang (0-10V) ab, der zur Geschwindigkeitsregelung herangezogen wird.

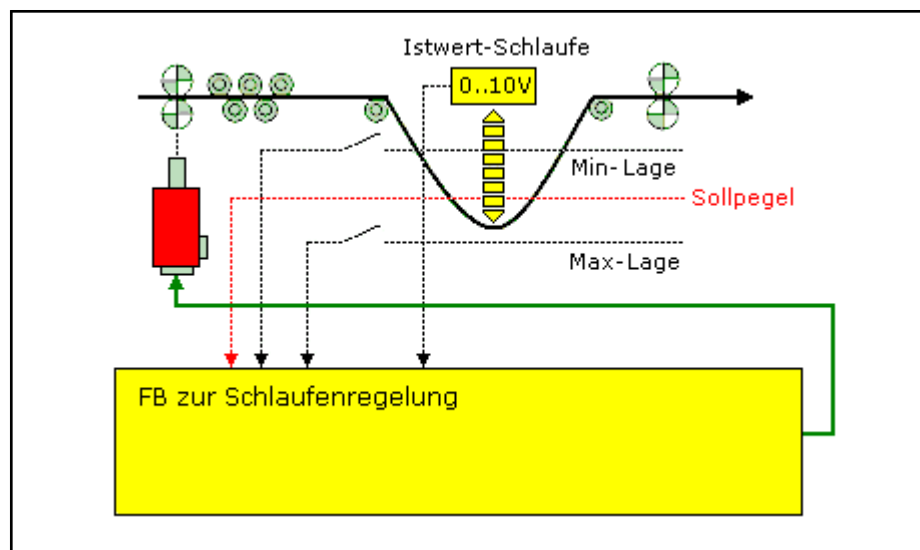


Abb. 4-45: Anwendung 1, ML_SagControlA

Im Normalbetrieb läuft die Bandanlage mit einer durch den Bediener vorgegebenen durchschnittlichen Geschwindigkeit V_m . Der FB für die Schlaufenregelung versucht nun, die Schlaufe in einem definierten Fenster zu halten (Min/Max-Lage), indem je nach Abweichung die aktuelle Geschwindigkeit der Bandanlage angepasst wird.

Applikationsbeispiel 2

Bei dieser Variante ist der zu regelnde Antrieb der Abwickler (Coil) selbst. Hierbei kommt neben dem Analogwert-Eingang auch der abnehmende Coildurchmesser als Regelgröße hinzu.

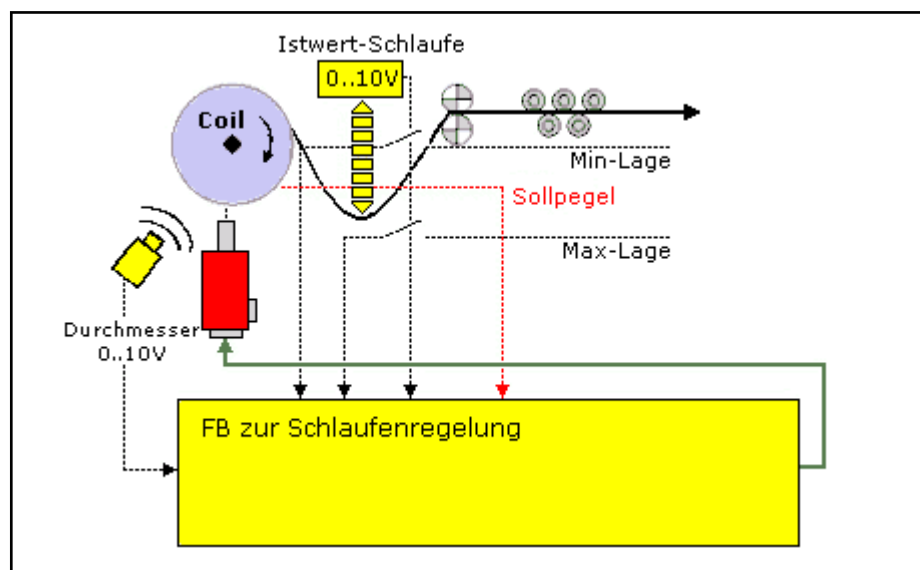


Abb. 4-46: Anwendung 2, ML_SagControlA

4.6 SmartBelt

4.6.1 MB_SmartBeltType01

Kurzbeschreibung

Dieser Funktionsbaustein misst die Winkelverschiebung zwischen Produktposition und gewünschter Leitachspolition. Wird eine Winkelverschiebung erkannt, wird eine Winkelkorrektur an der Produktposition mittels eines dreieckigen Geschwindigkeitsprofils vorgenommen.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechMotion
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 4-37: Referenztabelle MB_SmartBeltType01

Schnittstellenbeschreibung

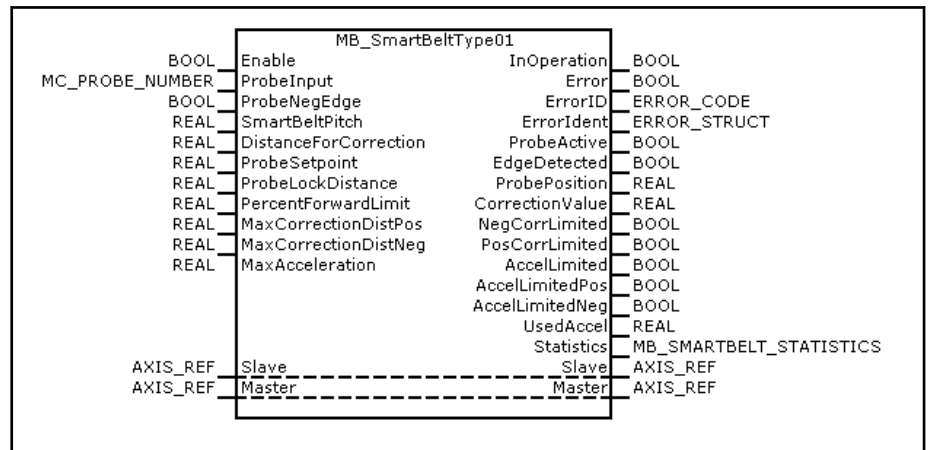


Abb. 4-47: Funktionsbaustein MB_SmartBeltType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Steigende Flanke aktiviert Funktionsbaustein
	ProbeInput	MC_PROBE_NUMBER	Auswahl des Messtastereingangs auf Antrieb (1= Probe1, 2= Probe2). Wird bei steigender Flanke am Eingang "Enable" gelesen
	ProbeNegEdge	BOOL	FALSE: Positive Flanke des Messtasters wird ausgewertet TRUE: Negative Flanke des Messtasters wird ausgewertet. Wird bei steigender Flanke am Eingang "Enable" gelesen
	SmartBeltPitch	REAL	Wirksame SmartBelt-Länge (Inch [in] bzw. Millimeter [mm]). Wird bei steigender Flanke am Eingang "Enable" gelesen
	DistanceForCorrection	REAL	Verbleibender Abstand für die Korrektur auf dem Band. Dies bezieht sich normalerweise auf die Entfernung, von der der Messtaster am Ende des Bands angebracht ist. Wird bei steigender Flanke am Eingang "Enable" gelesen
	ProbeSetpoint	REAL	Sollwert des Messtasters (Inch [in] bzw. Millimeter [mm]) Wird zyklisch gelesen

E/A-Typ	Name	Datentyp	Beschreibung
	ProbeLockDistance	REAL	Entfernung innerhalb der Messtasterereignisse ignoriert werden (Inch [in] oder Millimeter [mm]). Bei Erkennung eines Messtasterereignisses wird das nächste Messtasterereignis erst berücksichtigt, nachdem das Band die Entfernung in "ProbeLockDistance" zurückgelegt hat. Wird zyklisch gelesen
	PercentForwardLimit	REAL	Der prozentuale Anteil des gesamten Produktzyklus, der die Verteilung zwischen Vorwärts- und Rückwärtskorrektur festlegt. Beispiel: Ein Wert von 70,0 würde bis zu 70 mm vorwärts und bis zu 30 mm rückwärts für einen Produktzyklus von 100 mm Länge korrigiert werden. Wird zyklisch gelesen
	MaxCorrectionDistPos	REAL	Maximaler Korrekturweg für positive Korrektur (Inch [in] oder Millimeter [mm]). Wird zyklisch gelesen
	MaxCorrectionDistNeg	REAL	Maximaler Korrekturweg für negative Korrektur (Inch [in] oder Millimeter [mm]). Wird zyklisch gelesen
	MaxAcceleration	REAL	Maximal zulässige Beschleunigung in Antriebseinheiten (in/s^2 , mm/s^2). Wird zyklisch gelesen
VAR_OUTPUT	InOperation	BOOL	SmartBelt in Betrieb
	Error	BOOL	Weist auf einen Fehler hin. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 243
	ProbeActive	BOOL	SmartBelt-Messtasterfunktionalität aktiv. Wartet auf Ereignis
	EdgeDetected	BOOL	Messtasterereignis erkannt
	ProbePosition	REAL	Erfasste Position in Antriebseinheiten basierend auf "SmartBeltPitch"
	CorrectionValue	REAL	Korrekturwert in Antriebseinheiten basierend auf "SmartBeltPitch".
	NegCorrLimited	BOOL	Negative Korrektur ist auf den in "MaxCorrectionDistanceNeg" eingestellten Weg begrenzt
	PosCorrLimited	BOOL	Positive Korrektur ist auf den in "MaxCorrectionDistancePos" eingestellten Weg begrenzt
	AccelLimited	BOOL	Korrektur ist auf die in "MaxAcceleration" eingestellte Beschleunigung begrenzt
	AccelLimitedPos	BOOL	Positive Beschleunigung ist begrenzt

E/A-Typ	Name	Datentyp	Beschreibung
	AccelLimitedNeg	BOOL	Negative Beschleunigung ist begrenzt
	UsedAccel	REAL	Berechnete Beschleunigung, die während der Winkelkorrektur verwendet wird (in/s ² oder mm/s ²)
	Statistics	MB_SMART-BELT_STATISTICS	Struktur, die verschiedene statistische Daten enthält MB_SMARTBELT_STATISTICS, Seite 245
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Adresse der SmartBelt-Achse
	Master	AXIS_REF	Adresse der Leitachse

Tab. 4-38: Schnittstellenvariablen Funktionsbaustein MB_SmartBeltType01

Min/Max- und Standard-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
Enable	BOOL			FALSE
ProbeInput	MC_PROBE_NUMBER	PROBE1	PROBE2	PROBE1
ProbeNegEdge	BOOL	-	-	FALSE
SmartBeltPitch	REAL	>0.0	10000.0	200.0
DistanceForCorrection	REAL	0.0	SmartBeltPitch	100.0
ProbeSetpoint	REAL	0.0	Modulowert der Leitachse	0.0
ProbeLockDistance	REAL	0.0	SmartBeltPitch	0.0
PercentForwardLimit	REAL	0.0	100.0	50.0
MaxCorrectionDistPos	REAL	0.0	SmartBeltPitch	100.0
MaxCorrectionDistNeg	REAL	-SmartBeltPitch	0.0	-100.0
MaxAcceleration	REAL	> 0.0	∞	100
Slave	AXIS_REF	AxisNo >= 1	<= MB_MAX_AXIS_NUMBER	-
Master	AXIS_REF	AxisNo >= 1	<= MB_MAX_AXIS_NUMBER	-

Tab. 4-39: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Funktionsbeschreibung**Detaillierte Beschreibung der Eingangs- und Ausgangsvariablen**

Die Maßeinheiten müssen für alle Eingänge gleich sein (Inch [in] oder Millimeter [mm]). Ist der "SmartBeltPitch" beispielsweise in mm angegeben, muss "DistanceForCorrection" in mm sein und auch "MaxAcceleration" in mm/s².

"PercentForwardLimit" wird verwendet um die Verteilung von Vorwärts- zu Rückwärts-Korrektur zu bestimmen. Wenn das Produkt z.B. nur in Vorwärtsrichtung korrigiert werden soll, muss als Wert 100.0 eingegeben werden. Damit werden all möglichen Positionsfehler durch eine Korrektur nach vorne

ausgeglichen. Dieser Wert gibt die prozentuale Verteilung nach vorne, bzw. nach hinten an.

Der maximal gemessene Fehler für ein Produkt kann zwischen $(-0.5 * \text{Produktzyklus})$ und $(+0.5 * \text{Produktzyklus})$ liegen. Der Begriff Produktzyklus ist identisch mit "CamShaftDistance" oder entspricht der Produktlänge plus dem Abstand zwischen den Produkten. Die folgende Abbildung zeigt ein Beispiel:

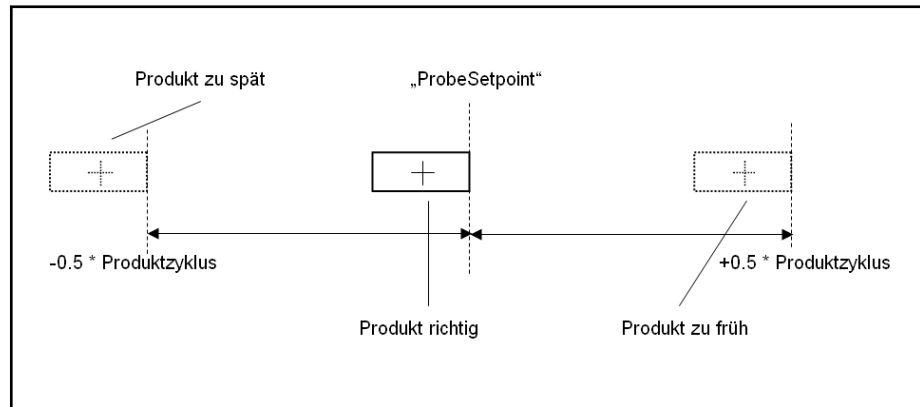


Abb. 4-48: Bedeutung der Korrektur in positiver, bzw. negativer Richtung

Das Produkt kann jetzt entweder in positive oder negative Richtung korrigiert werden. Somit ist es möglich, die Länge eines gesamten Produktzyklus zu korrigieren. "PercentForwardLimit" gibt an, wie der gemessene Fehler in Bezug auf die gesamte Produktlänge zu korrigieren ist. Wird "PercentForwardLimit" auf 50% gesetzt, wird die Hälfte der Fehler vorwärts, die andere Hälfte rückwärts korrigiert.

Wird "PercentForwardLimit" auf 70% gesetzt, werden die Korrekturen wie folgt vorgenommen:

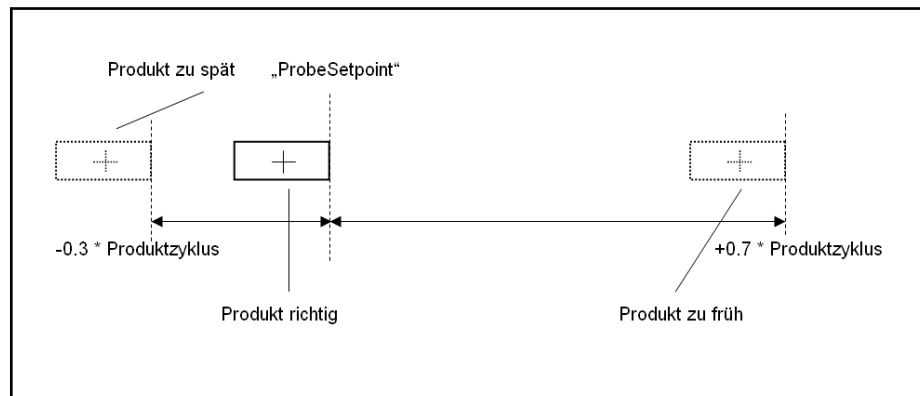


Abb. 4-49: PercentForwardLimit

"MaxCorrectionDistancePos" und "MaxCorrectionDistanceNeg" begrenzen den absoluten Weg, der in beide Richtungen kompensiert wird. Mit "PercentForwardLimit" können Kombinationen erstellt werden, mittels derer alle Fehler oder nur Teilfehler in beide Richtungen kompensiert werden können. Wenn beispielsweise 80% aller Fehler in die positive Richtung korrigiert werden sollen (aber nicht mehr als 60 mm in die positive Richtung und nicht mehr als 20 mm in die negative Richtung der Länge eines gesamten Produktzyklus korrigiert werden sollen), werden die Eingänge wie folgt konfiguriert:

"PercentForwardLimit" = 80

"MaxCorrectionDistancePos" = 60

"MaxCorrectionDistanceNeg" = -20

Erläuterung: 80% (160 mm) werden vorwärts und 20% (40 mm) werden rückwärts für einen 200mm langen Produktzyklus mit "PercentForwardLimit" = 80 korrigiert. "MaxCorrectionDistancePos" = 60 begrenzt die vorhandenen 160 mm auf 60 mm und "MaxCorrectionDistanceNeg" = -20 begrenzt die zulässige Rückwärtskorrektur auf -20 mm.

Der Gesamtprozess wird im Folgenden beschrieben:

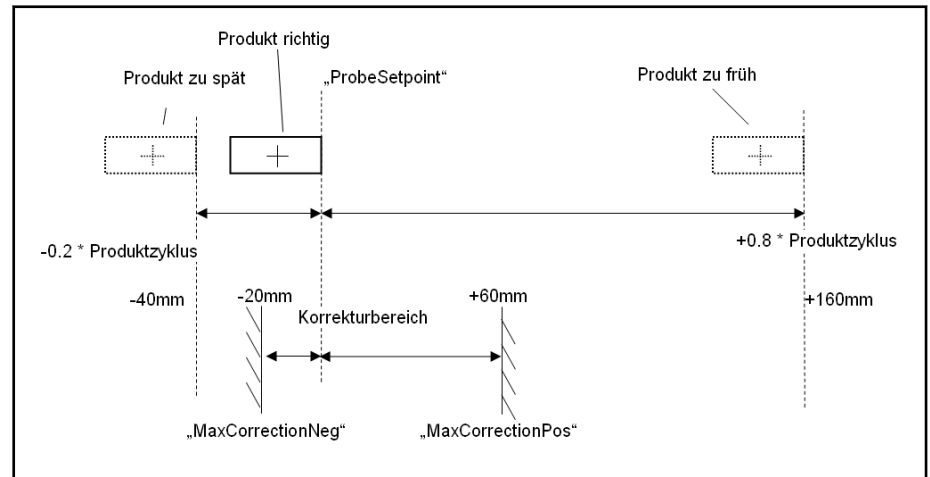


Abb. 4-50: *MaxCorrectionDistancePos und MaxCorrectionDistanceNeg*

Ein Fehler, der sich außerhalb der "MaxCorrectionDistancePos"- und "MaxCorrectionDistanceNeg"-Begrenzungen befindet, wird auf "MaxCorrectionDistancePos" und "MaxCorrectionDistanceNeg" begrenzt. Der verbleibende Fehler muss auf dem nächsten Band korrigiert werden. Damit ist es möglich die Korrekturen auf alle Bänder zu verteilen, so dass das erste Band nicht die Hauptlast der Korrektur tragen muss.

Je länger der Korrekturweg, desto höher die Beschleunigung. Alternativ kann die maximale Beschleunigung auch mittels "MaxAcceleration" begrenzt werden. Die maximale Beschleunigung und der maximale Korrekturabstand sind gleichzeitig aktiv. Die niedrigere der beiden Beschränkungen gilt.

Zeitanforderungen

Der Funktionsbaustein benötigt bis zu 400 SPS-Zyklen, um die Initialisierung abzuschließen. Die genaue Zyklenanzahl hängt von der Belastung der Steuerung ab.

Fehlerbehandlung

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0006)	16#0012	16#0000	Achse ohne Leistung
RESOURCE_ERROR (16#0006)	16#0001	x	Im Antrieb ist keine Leistung zu geschaltet "x" gibt die Adresse der fehlerhaften Achse an
ACCESS_ERROR (16#0004)	16#0003	16#0000	Der intern benutzte Funktionsbaustein MB_TouchProbeContinuous wurde durch einen anderen Funktionsbaustein unterbrochen
ACCESS_ERROR (16#0004)	16#0003	16#0001	Der intern benutzte Funktionsbaustein MB_PhasingSlave wurde durch einen anderen Funktionsbaustein unterbrochen

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Fehler in Zustandsmaschine
INPUT_INVALID_ERROR (16#0001)	16#000D	x	Achse (Leit- oder Folgeachse) besitzt nicht gültige Adresse. x gibt die Adresse der fehlerhaften Achse an
INPUT_INVALID_ERROR (16#0001)	16#0F02	16#0010	SmartBeltPitch außerhalb des gültigen Wertebereichs (0..10000)
INPUT_INVALID_ERROR	16#0F02	16#0011	ProbeInput ungültig (PROBE1, PROBE2)
INPUT_INVALID_ERROR	16#0F02	16#0012	DistanceForCorrection ungültig (0..SmartBeltPitch)
INPUT_INVALID_ERROR	16#0F02	16#0013	MaxCorrectionDistPos ungültig (0..SmartBeltPitch)
INPUT_INVALID_ERROR	16#0F02	16#0014	MaxCorrectionDistNeg ungültig (-SmartBeltPitch..0)
INPUT_INVALID_ERROR	16#0F02	16#0015	Achse ist keine IndraDrive-Achse (Packprofil nicht zulässig)
INPUT_INVALID_ERROR	16#0F02	16#0017	Folgeachswichtung ist nicht die bevorzugte Wichtung
INPUT_INVALID_ERROR	16#0F02	16#0018	Leitachswichtung ist nicht die bevorzugte Wichtung
INPUT_INVALID_ERROR	16#0F02	16#0019	Beschleunigungswert zu niedrig (>0)
INPUT_INVALID_ERROR	16#0F02	16#001A	Folgeachse hat keine Modulowichtung
INPUT_INVALID_ERROR	16#0F02	16#001B	Leitachse hat keine Modulowichtung
INPUT_INVALID_ERROR	16#0F02	16#001C	Einstellungen in P-0-0154 (MLD) / A-0-2793 (MLC) müssen auf "ShortestPath" gesetzt sein
INPUT_INVALID_ERROR	16#0F02	16#001D	ProbeSetpoint-Eingang ist außerhalb des gültigen Bereichs (0..Leitachsmodulo)
INPUT_INVALID_ERROR	16#0F02	16#001F	ProbeLockDistance-Eingang ist außerhalb des gültigen Bereichs (0..SmartBeltPitch)
INPUT_INVALID_ERROR	16#0F02	16#0020	PercentForwardLimit ist außerhalb des gültigen Bereichs (0..100.0)
INPUT_INVALID_ERROR	16#0F02	16#0021	MaxCorrectionDistPos+ ABS(MaxCorrectionDistNeg) ist größer als SmartBeltPitch
INPUT_INVALID_ERROR	16#0F02	16#0022	Modulowert der Folgeachse muss mindestens doppelt so groß wie der SmartBeltPitch sein
DEVICE_ERROR (16#0008)	16#0F06	16#0000	Geschwindigkeitswichtung und Wichtung der Lagedaten sind nicht gleich

Tab. 4-40: Fehlercodes Funktionsbaustein MB_SmartBeltType01

Zusätzlich kann der Funktionsbaustein alle folgenden Fehler der Unterfunktionsbausteine ausgeben:

- MB_GetCyclicParameterHandle
- MB_PhasingSlave
- MB_ReadParameter
- MB_ReadRealParameter
- MB_ReadStringParameter
- MB_InitTouchProbe
- MB_TouchProbeContinuous

Siehe Dokumentation zu den Fehlercodes des jeweiligen Bausteins.

4.6.2 MB_SMARTBELT_STATISTICS

Diese Struktur enthält statistische Daten, die für die Überwachung und Auswertung der Anwendung verwendet werden können. Die einzelnen Elemente sind in der nachfolgenden Tabelle aufgeführt:

Strukturelement	Typ	Default-Wert	Beschreibung
CurRate	REAL	0.0	Produktrate basierend auf dem Abstand zwischen aktuellem und vorherigem Produkt
NbrOnBelt	UINT	0	Aktuelle Produktanzahl auf dem Band. Ein Produkt ist "auf dem Band", wenn es weniger als eine ganze SmartBelt-Länge seit der Registrierung gefahren ist
MaxGap	REAL	0.0	Maximaler Abstand zwischen zwei aufeinanderfolgenden Produkten in Einheiten basierend auf dem "SmartBeltPitch"
MinGap	REAL	"SmartBeltPitch"	Minimaler Abstand zwischen zwei aufeinanderfolgenden Produkten in Einheiten basierend auf dem "SmartBeltPitch"
TotalNbr	UINT	0	Gesamtproduktanzahl, die vom SmartBelt registriert wurde

Tab. 4-41: Struktur des MB_SMARTBELT_STATISTICS

Alle Elemente in der Struktur MB_SMARTBELT_STATISTICS werden auf ihren Standardwert bei steigender Flanke am Eingang "Enable" gesetzt.



Die Werte der Elemente können falsch sein, wenn die Produkte nicht richtig registriert wurden.

"CurRate" Wert wird auf der Basis von Zeitmessungen berechnet. Die Zeitauflösung beträgt 1ms. Für hohe Produktmengen pro Zeiteinheit können die berechneten Geschwindigkeiten im Bereich der 1ms-Auflösung variieren. Wird beispielsweise eine Produktrate von 400 Produkten pro Minute verwendet, beträgt die Zeit für ein Produkt $1 / (400 / 60) = 150$ ms. Bei einer Auflösung von 1ms können Werte im Bereich von 149ms, 150ms und 151ms gemessen werden, was in Produktraten von 403, 400 und 397 resultiert.

Die Werte für "MaxGap" und "MinGap" werden auf Basis der zurückgelegten Distanz von der Leitachse berechnet. Dies merkt man besonders auf dem ersten Band, wenn dieses beschleunigt oder abbremst. Der SmartBelt legt in diesem Fall wesentlich mehr/weniger Weg zurück als die Leitachse.

4.6.3 Einführung in die SmartBelt-Verwendung

SmartBelt-Verwendung

SmartBelts werden verwendet, um die Produkte vor ihrer Bearbeitung korrekt zu verteilen. Produkte werden durch eine Zuführeinheit den SmartBelts, wo der Winkel eingestellt wird, zur Verfügung gestellt. Dann übernimmt eine Bearbeitungseinheit die Produkte (z.B. Schlauchbeutelmaschine).

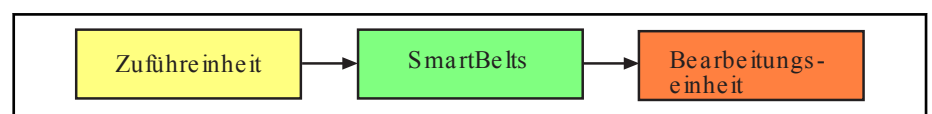


Abb. 4-51: Beispiel einer Schlauchbeutelmaschine

Je nach Anwendung sind mehrere Situationen denkbar:

- Zuführeinheiten stellen Produkte mit richtigem Abstand zueinander bereit, aber die Bearbeitungseinheiten können nicht synchron zur Zuführeinheit starten bzw. stoppen. In diesem Fall können SmartBelts dazu benutzt werden, die Abstände zwischen den Produkten zu korrigieren
- Zuführeinheiten stellen Produkte aus dem Storage-on-Demand bereit. SmartBelts werden nicht benötigt
- Zuführeinheiten sorgen für einen zufälligen, aber fast immer gleichen Abstand zwischen den Produkten. SmartBelts können verwendet werden
- Zuführeinheiten sorgen für einen zufälligen Abstand zwischen den Produkten. SmartBelts können zusammen mit den folgenden Maßnahmen verwendet werden:
 - Die Geschwindigkeit der Zuführeinheiten kann gesteuert werden, um eine konstante Produktgeschwindigkeit zu erreichen
 - Die Geschwindigkeit der Bearbeitungseinheiten kann gesteuert werden, um eine konstante Produktgeschwindigkeit zu erreichen

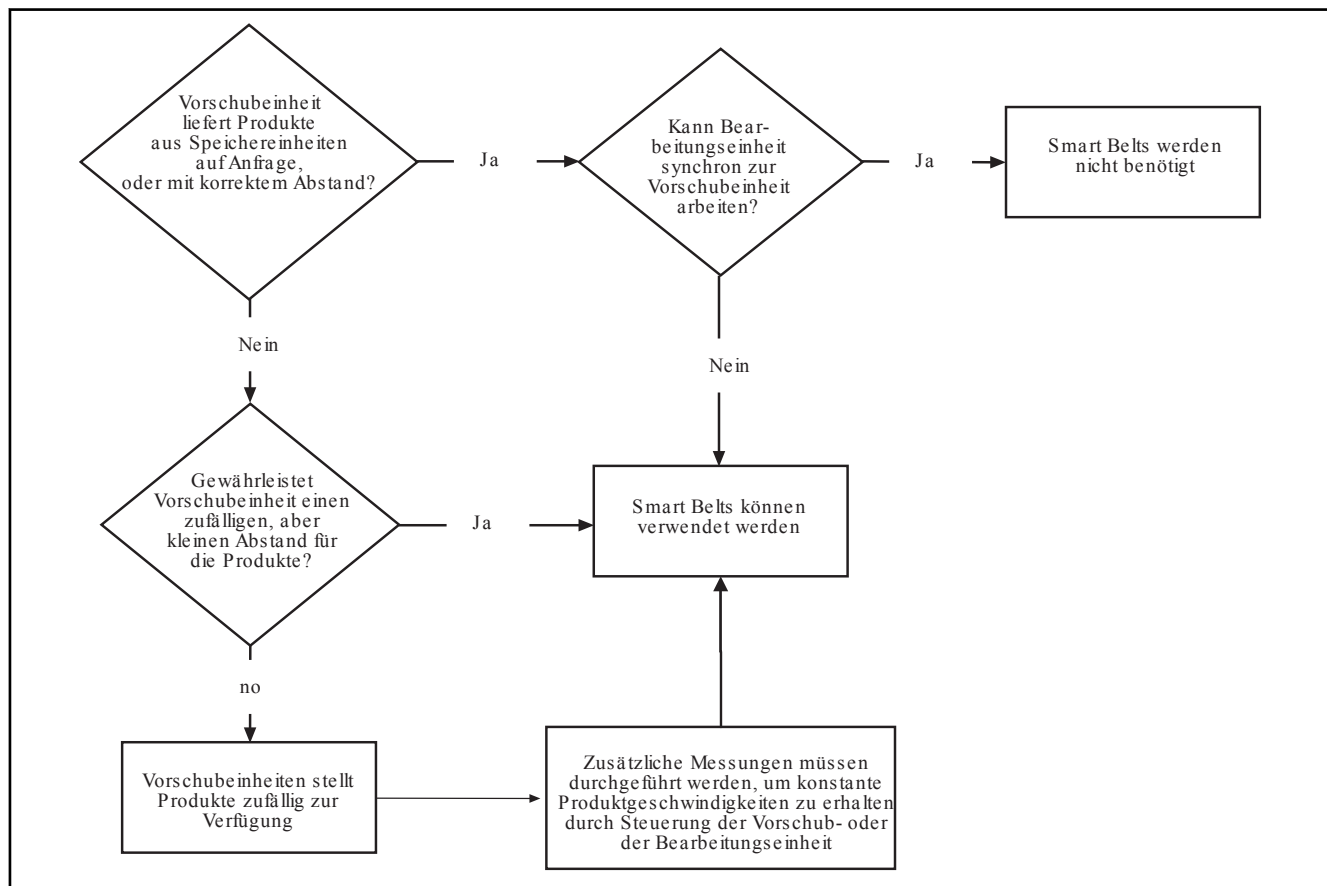


Abb. 4-52: Verwendung des SmartBelts

Stauförderer

Die Verwendung eines Stauförderers erhöht die Produktabfolge. In der folgenden Abbildung ist Band 2 als Stauförderer dargestellt. Seine Länge (L2) muss länger als die Länge der Produkte auf dem vorhergehenden Band (L1) sein. Sonst schieben die Produkte von Band 1 die Produkte auf Band 2.

Damit sich die Produkte ansammeln können, muss die Geschwindigkeit des Stauförderers geringer sein als die Geschwindigkeit von Band 1.

Damit die Produkte einen vordefinierten Abstand voneinander erhalten, muss Band 3 mit einer höheren Geschwindigkeit laufen als der Stauförderer.

Abstände zwischen den Produkten auf dem Stauförderer resultieren in noch größere Abständen auf Band 3. Diese Abstände sind jedoch nicht zwingenderweise ein Vielfaches des gewünschten Abstands. Um diese Situation zu vermeiden, kann die Geschwindigkeit des Stauförderers gesteuert werden.

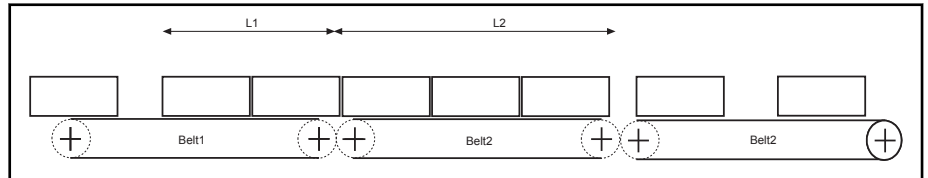


Abb. 4-53: Stauförderer

Abfolgegrad

Auswertung mittels Produktzyklen und Abständen

Der Abfolgegrad ist entscheidend für die Anzahl der SmartBelts. Je höher der Grad, desto weniger SmartBelts werden benötigt. Der Abfolgegrad hat direkte Auswirkungen auf die SmartBelt-Betragskorrektur. Dieser wird mit der folgenden Formel berechnet:

Die folgende Abbildung erläutert die verschiedenen Parameter:

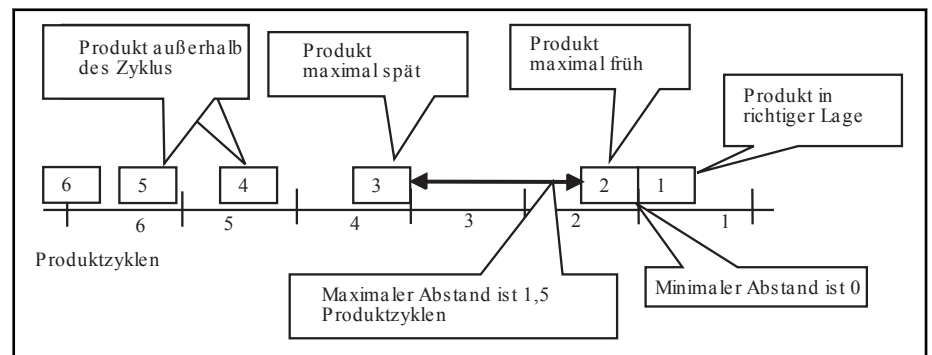


Abb. 4-54: Auswertung mittels Produktzyklen und Abständen

Allgemein wird angenommen, dass die positive Produktflanke der Referenzpunkt ist. Solange immer derselbe Referenzpunkt verwendet wird, spielt es bei den folgenden Schlüssen keine Rolle, ob die positive/negative Flanke oder die Produktmitte betrachtet wird.

In der Abbildung oben, befindet sich Produkt 1 in der gewünschten Position und muss nicht korrigiert werden. Produkt 2 ist zu nahe an Produkt 1, aber immer noch innerhalb des Produktzyklus. Produkt 2 hat einen minimalen Abstand von 0 zum vorherigen Produkt. Produkt 3 ist zu spät, aber immer noch innerhalb des Produktzyklus. Die Produkte 4 und 5 befinden sich außerhalb ihres Produktzyklus, Produkt 6 hingegen ist innerhalb seines Produktzyklus.

Die Betragskorrektur wird mit Hilfe der folgenden Variablen berechnet:

- Produktlänge (L_{Prod})
- Nominaler Produktabstand bei vorgegebener Geschwindigkeit (L_{Abs})
- Nominaler Produktzyklus bei vorgegebener Geschwindigkeit (L_{Abs})
- Minimaler Produktabstand (L_{Absmin})
- Maximaler Produktabstand (L_{Absmin})
- Aufeinanderfolgende Produkte außerhalb des Produktzyklus (N_{Abs})

Die folgenden Werte gelten für die obenstehende Abbildung:

L_{Prod} : 40 mm (angenommen)

L_{Abs} : 50 mm (angenommen)

L_{zykl} : 90 mm (= $L_{\text{Prod}} + L_{\text{Abs}}$)

$L_{\text{Abs. min}}$: 0 mm (Abstand zwischen den Produkten 1 und 2)

$L_{\text{Abs. max}}$: 1.5 Produktzyklus = $1.5 * L_{\text{zykl}} = 135$ mm (Abstand zwischen den Produkten 2 und 3)

N_{Abs} : 2 (Abstand zwischen den Produkten 4 und 5)

Die benötigte Beträgskorrektur für die SmartBelts wird wie folgt angegeben:

$$\text{Korrektur} = \min\left(L_{\text{max Abs}}, \left(1.5 * L_{\text{zykl}} - L_{\text{Prod}} + L_{\text{Abs}} * (N_{\text{Abs}} - 1) - (N_{\text{Abs}} * L_{\text{min Abs}})\right)\right)$$

Der folgende Wert gilt für dieses Beispiel:

$$\begin{aligned} \text{Korrektur} &= \min(135 \text{ mm}, (1.5 * 90 \text{ mm} - 40 \text{ mm} + 50 \text{ mm} * (2 - 1) - (2 * 0 \text{ mm}))) \\ &= \min(135 \text{ mm}, 145 \text{ mm}) \\ &= 135 \text{ mm} \end{aligned}$$

In diesem Beispiel sind 135 mm die Entfernung, die der SmartBelt durchfahren muss, um die Produktpositionen in beiden Richtungen zu korrigieren.

Auswertung mittels Produktgeschwindigkeit und Dauer

Auch mittels der Produktgeschwindigkeit kann geprüft werden, ob das Korrekturvermögen ausreicht. Die Geschwindigkeit sollte konstant sein, es gibt aber Abweichungen, die sich in der Nähe des Mittelwerts bewegen. Die Dauer und die Höhe dieser Abweichung kann auch dazu verwendet werden, um festzustellen, ob ein SmartBelt diese Anwendung unterstützen kann.

Die Formel zur Berechnung der Zeitabweichung ist folgende:

$$t = \frac{1}{\frac{R}{60} * \frac{\text{Abweichung}[\%]}{100}}$$

R ist die gewünschte Produktgeschwindigkeit in [1/min] und "Abweichung" ist der Prozentsatz um den die Produktgeschwindigkeit abweicht.

Beispiel:

Bei einer gewünschten Produktgeschwindigkeit von 200 Produkten pro Minute kommt es zu einer Abweichung von 10%. Das heißt, dass 180 bis 220 Produkte pro Minute über das Band laufen können. Die oben genannte Formel ergibt eine Zeit von:

$$t = \frac{1}{\frac{200}{60} * \frac{10\%}{100}} = 3 \text{ s}$$

Bei einer Dauer von weniger als 3 Sekunden in der die Produktrate von der nominalen Rate abweicht, kann der SmartBelt die Anwendung unterstützen. Ansonsten kommen Produkte ohne Korrektur vom Band oder es kommt zu leeren Plätzen.

Reibung

Der Zusammenhang zwischen Reibung und möglicher Beschleunigung lässt sich wie folgt erklären:

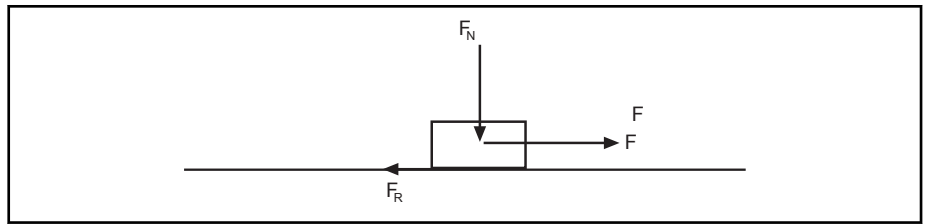


Abb. 4-55: Reibung

Statische Reibung ist definiert als:

$$F_R = \mu_H F_N$$

mit

$$F_N = F_G = m \cdot g$$

und ersetzt die zweite Formel durch die erste Formel

$$F_R = \mu_H \cdot m \cdot g$$

Benötigte Kraft für die Beschleunigung ist:

$$F = m \cdot a$$

Die dynamische Reibung beginnt bei $F = F_R$

oder ersetzt durch

$$\mu_H \cdot m \cdot g = m \cdot a_{\max}$$

gelöst für $a_{\max}$:

$$a_{\max} = \mu_H \cdot g$$

$a_{\max}$ ist die maximale Beschleunigung, μ_H ist der Reibungskoeffizient und g ist die Schwerkraftbeschleunigung.

μ_H kann auch anders visualisiert werden, wie die folgende Abbildung zeigt. μ_H ist das Äquivalent zu $\tan \phi$. In anderen Worten, ein Objekt mit einer Reibung von 1 würde bei einem Winkel von 45° anfangen zu rutschen.

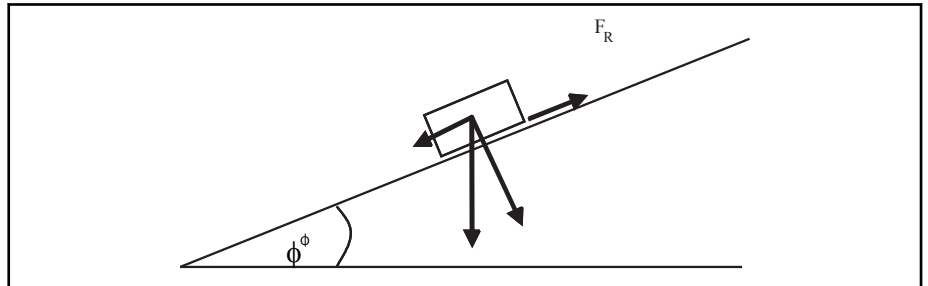


Abb. 4-56: Zusammenhang zwischen Reibungskoeffizient und Winkel

4.6.4 Projektplanung

Allgemeines

Dieser Abschnitt beschreibt die durchzuführenden Einstellungen und Berechnungen, wenn der Funktionsbaustein MB_SmartBeltType01 verwendet wird.

Mechanische Anforderungen an Band und Maschine

Die Bandlänge sollte ungefähr die Produktgröße plus den gewünschten Abstand haben (Produktzyklus). Mit Bandlänge ist die physikalische Länge des gesamten Bands gemeint und nicht der Umfang. Es gibt keinen Vorteil, wenn das Band deutlich länger oder kürzer als der Produktzyklus ist. Bei einem längeren Band sind mehrere Produkte auf einem Band. Daher muss das letzte Produkt darauf warten, dass das erste Produkt das Band verlässt. Bei

kürzeren Bändern werden weniger Zeit und schnellere Beschleunigungen benötigt um die gleichen Abstände zu erhalten.

Der Messtaster muss für jedes Band an den entsprechenden Antriebseingang angeschlossen sein. Merken Sie sich den verwendeten Messtastereingang. Der Messtaster muss so nah wie möglich an den Bandanfang platziert werden, wenn die positive Flanke erkannt wird. Das Produkt muss seinen Massenschwerpunkt auf dem Band haben. Bei symmetrischen Produkten ist dies die Hälfte der Produktlänge. Der Messtaster muss auf der Hälfte der Produktlänge plus die Hälfte des Achsdurchmessers des Bandes angebracht sein.

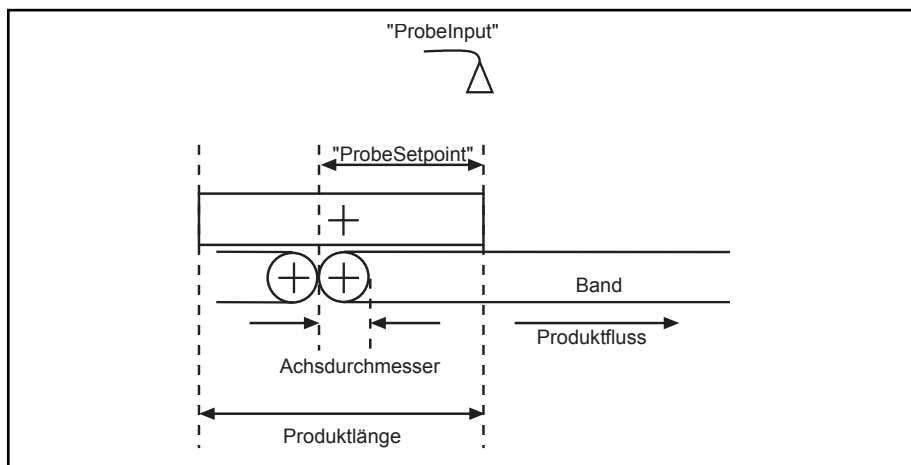


Abb. 4-57: Messtaster mit Erkennung der positiven Flanke

Wird die negative Flanke des Produkts verwendet, wird eine ähnliche Logik angewendet. Der Messtaster muss dann auf der Hälfte der Produktlänge minus der Hälfte des Achsdurchmessers des Bandes vor dem Band angebracht werden.

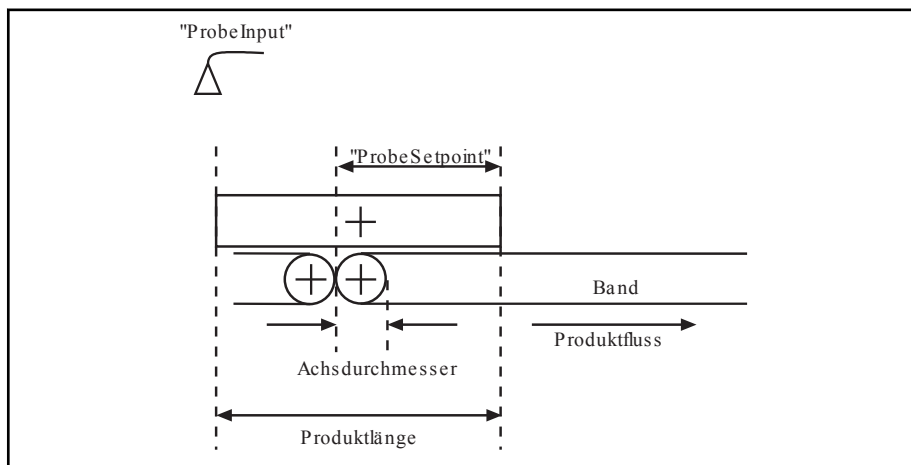


Abb. 4-58: Messtaster mit Erkennung der negativen Flanke

Wenn mehrere Produkte mit den unterschiedlichen Formaten auf demselben SmartBelt verwendet werden sollen, muss der Messtaster für die Registrierung mit der negativen Flanke angebracht werden. In diesem Fall bestimmt das kleinste Produkt den Ort des Messtasters. In diesem Fall wird die Kompensationsfähigkeit größerer Produkte verringert.

Antriebsanforderungen

Der Antrieb muss mit einem Funktionspaket ausgestattet sein, das synchrone Betriebsarten und Messtastereingänge unterstützt.

Folgende Anforderungen gelten für die MLC:

- Nur IndraDrives mit Interpolation im Antrieb können verwendet werden. Die Antriebsfirmware muss mindestens MPx04, V22 oder höher sein. MPx03 wird nicht unterstützt

Folgende Anforderungen gelten für die MLD:

- Die Antriebsfirmware muss mindestens MPx05 oder höher sein. MPx03 und MPx04 werden nicht unterstützt
- Das MA Paket muss aktiviert sein

Antriebsparametereinstellungen

Es wird empfohlen, die Basisparameter vor der Antriebskonfiguration zu laden. In diesem Abschnitt werden die Unterschiede zu den Standardparametern aufgelistet.

Vor der Verwendung des Funktionsbausteins, müssen zusätzliche Einstellungen vorgenommen werden. Die Messtastereinstellungen müssen korrekt sein, da der Funktionsbaustein die Messtasterfunktionalität nutzt. Abhängig von dem verwendeten Messtastereingang und der Flanke zum Auslösen, werden die folgenden Einstellungen für den optionalen zyklischen Kanal benötigt. Um verschiedene Möglichkeiten zur Laufzeit zu haben, können mehrere Einstellungen konfiguriert werden. Setzen Sie beispielsweise S-0-0130 und S-0-0131, sowie S-0-0409 und S-0-0410 im zyklischen Lesekanal. Dies erlaubt die Verwendung von entweder positiven oder negativen Flanken zur Laufzeit.



Die folgenden Parameter werden in der Steuerung gesetzt, damit die zyklischen Parameter zum Antrieb übertragen werden können. Zusätzlich im Antrieb gesetzte Parameter sind für die Messtasterfunktionalität notwendig.

Der Antriebsparametereditor von IndraWorks kann für die Parametereinstellungen verwendet werden.

MLC-Parametereinstellungen

Gewünschte Konfiguration	Zyklischer Lesezugriff	Signalsteuerwort
Messtaster 1, positive Flanke	S-0-0130, S-0-0409	S-0-0405, Bit 0
Messtaster 1, negative Flanke	S-0-0131, S-0-0410	S-0-0405, Bit 0
Messtaster 2, positive Flanke	S-0-0132, S-0-0411	S-0-0406, Bit 0
Messtaster 2, negative Flanke	S-0-0133, S-0-0412	S-0-0406, Bit 0

MLD-Parametereinstellungen

Gewünschte Konfiguration	Istwerte	Sollwerte
Messtaster 1, positive Flanke	S-0-0130 S-0-0409	S-0-0405, Bit 0
Messtaster 1, negative Flanke	S-0-0131 S-0-0410	S-0-0405, Bit 0

Gewünschte Konfiguration	Istwerte	Sollwerte
Messtaster 2, positive Flanke	S-0-0132 S-0-0411	S-0-0406, Bit 0
Messtaster 2, negative Flanke	S-0-0133 S-0-0412	S-0-0406, Bit 0

Messtasterfunktionalität im Antrieb

Die nachfolgenden Parameter sind im Antrieb gesetzt um die Antriebsmess-
tasterfunktionalität zu aktivieren:

Parameter	Wert	Beschreibung
P-0-0300	S-0-0401 (Messtaster 1) bzw. S-0-0402 (Messtaster 2)	Digitale Eingangsbelegung der Messtasterfunktionalität
P-0-0301	0 (Bit 0)	Bitbelegung zur Messtasterfunktionalität
S-0-0426	P-0-0775	Leitachsposition für die Messtastermessung
S-0-0169	Bit 0 bis 3	Aktivierung der positiven bzw. negativen Flankenerkennung für die Messtaster 1 und 2
P-0-0226	Bit 9 = TRUE	Messtasterfunktion wird automatisch bei Eintritt in Phase 4 im Antrieb aktiviert

Zusätzlich zu den anwendungsspezifischen Einstellungen, müssen die folgenden Parameter gesetzt sein.

Hardwareunabhängige Einstellungen

Parameter	Wert	Beschreibung
P-0-0072 P-0-0092 P-0-0780 P-0-0781 P-0-0782 P-0-0783 P-0-0784 P-0-0785	Lineare Kurvenscheibe mit dem Verhältnis 1:1 zwischen Leitachse und Folgeachse. Es muss nur ein Parameter (Kurvenscheibe) gesetzt werden. Die CamBuilder-Software kann verwendet werden um eine Kurvenscheibe zu erstellen und herunterzuladen	Die dazugehörige Kurvenscheibennummer (CamTableID) muss im MC_CamIn Baustein referenziert werden, wenn ein synchroner Betriebsmodus erstellt wird
S-0-0121	Eingangsumdrehungen des Lastgetriebes	Eingangsumdrehungen des mechanischen Getriebes
S-0-0122	Ausgangsumdrehungen des Lastgetriebes	Ausgangsumdrehungen des mechanischen Getriebes
S-0-0123	Lineare Bewegung pro Umdrehung	Vorschubkonstante für den Antrieb

Achsparametereinstellungen der MLC

Parameter	Wert	Beschreibung
A-0-0045	Doppelte Bandlänge	Modulowert der Achse
A-0-0059	Bit 0 = TRUE, Bit 1 und 2 = FALSE, Bit 3 = FALSE, Bit 7 = TRUE	Auswahl der Linear-, Modulo-, und bevorzugten Wichtung
A-0-0059	Bit 4 = FALSE für Einheiten in Millimeter [mm], Bit 4 = TRUE für Einheiten in Inch [in]	Auswahl der Bandeinheiten

Parameter	Wert	Beschreibung
A-0-2790	Hoch genug für die Anwendung	Aufsynchronisierungsgeschwindigkeit
A-0-2791	Hoch genug für die Anwendung	Aufsynchronisierungsbeschleunigung
A-0-0032	Hoch genug für die Anwendung	Geschwindigkeitsgrenzwert positiv
A-0-0033	Hoch genug für die Anwendung	Geschwindigkeitsgrenzwert negativ
A-0-0034	Hoch genug für die Anwendung	Beschleunigungsgrenzwert

Die Leitachse für das Band muss auch eine Modulo- und bevorzugte Wichtung (A-0-0059: Bit 3 = FALSE, Bit 7 = TRUE) sowie eine rotatorische bzw. eine lineare Wichtung haben. Eine Leitachsumdrehung entspricht einem Produktzyklus (Produktlänge plus Abstand zwischen den Produkten) für eine rotatorische Leitachse. Bei einer linearen Leitachse, muss der Modulowert dem Produktzyklus entsprechen. Entspricht ein Leitachszyklus mehreren oder Teilen eines Produktzyklus, muss das Getriebeverhältnis entsprechend eingestellt werden. Entspricht beispielsweise eine Leitachsumdrehung 3 Produkten, sind die Eingänge des Getriebes wie folgt:

- "RatioNumerator"/"InputRevolution" muss auf 1 gesetzt sein
- "RatioDenominator"/"OutputRevolution" muss auf 3 gesetzt sein

MLD-Parametereinstellungen

Parameter	Wert	Beschreibung
S-0-0103	Doppelte Bandlänge	Modulowert der Achse
S-0-0076	Bit 0 = TRUE, Bit 1 und 2 = FALSE, Bit 3 = FALSE, Bit 7 = TRUE	Auswahl der Linear-, Modulo-, und bevorzugten Wichtung
S-0-0076	Bit 4 = FALSE für Einheiten in Millimeter [mm], Bit 4 = TRUE für Einheiten in Inch [in]	Auswahl der Bandeinheiten
P-0-0143	Hoch genug für die Anwendung	Aufsynchronisierungsgeschwindigkeit
P-0-0142	Hoch genug für die Anwendung	Aufsynchronisierungsbeschleunigung
S-0-0038	Hoch genug für die Anwendung	Geschwindigkeitsgrenzwert positiv
S-0-0039	Hoch genug für die Anwendung	Geschwindigkeitsgrenzwert negativ
S-0-0138	Hoch genug für die Anwendung	Beschleunigungsgrenzwert

Die Leitachse für das Band muss auch über eine Modulo- und eine bevorzugte Wichtung (P-0-0756 bei einem Achsgenerator) sowie über rotatorische bzw. lineare Wichtung verfügen. Eine Leitachsumdrehung entspricht einem Produktzyklus (Produktlänge plus Abstand zwischen den Produkten) für eine rotatorische Leitachse. Bei einer linearen Leitachse, muss der Modulowert dem Produktzyklus entsprechen. Entspricht ein Leitachszyklus mehreren oder Teilen eines Produktzyklus, muss das Getriebeverhältnis entsprechend im SPS Programm eingestellt werden. Wenn beispielsweise eine Leitachsumdrehung drei Produkten entspricht, müssen die Eingänge "RatioNumerator"/"InputRevolution" auf 1 und "RatioDenominator"/"OutputRevolution" auf 3 gesetzt werden.

SPS-Programmierung

Die Achse mit der der SmartBelt betrieben werden soll, muss in einer synchronen Betriebsart sein. Wie diese einzustellen ist kann anhand der folgenden Code-Fragmente nachvollzogen werden.

Strukturierter Text:**Programm:**

```

myMC_CamIn(Execute:= TRUE,
           RatioNumerator:= 1,
           RatioDenominator:= 1,
           MasterFineadjust:= 0,
           CamShaftDistance:= myProductCycle,
           SyncMode:= SHORTEST_WAY,
           StartMode:= RELATIVE,
           CamTableID:= myCamTableID,
           Master:= mySmartBeltMaster,
           Slave:= myBelt,
           InSync=> ,
           Active=> ,
           CommandAborted=> ,
           Error=> ,
           ErrorID=> ,
           ErrorIdent=> ,
           CamActiveID=>
);

```

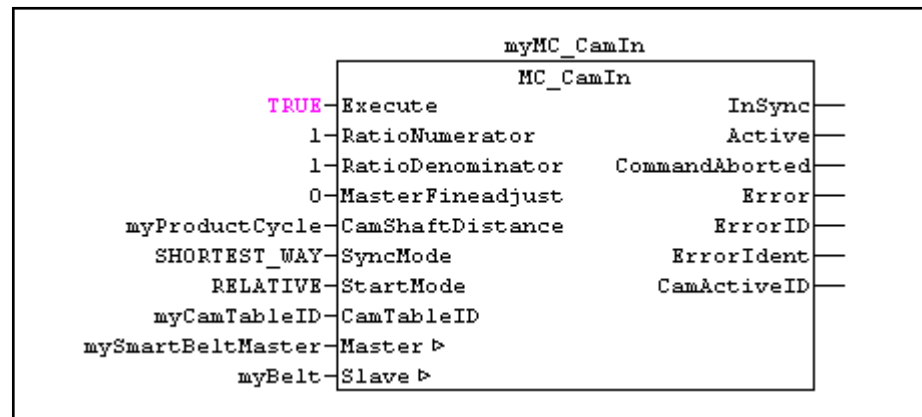
Funktionsplan:

Abb. 4-59: Einstellungen des MC_CamIn für den Einsatz mit dem SmartBelt

Wird das **AchsInterface** programmiert, muss der Betriebsmodus für den Antrieb auch auf "CamIn" gesetzt werden. Nachfolgend ein Beispiel:

Strukturierter Text:**Programm:**

```

arAxisCtrl[myBelt.AxisNo].SyncMode.Master:= mySmartBeltMaster;
arAxisCtrl[myBelt.AxisNo].SyncMode.SyncDirection:= SHORTEST_WAY;
arAxisCtrl[myBelt.AxisNo].SyncMode.StartMode:= RELATIVE;
arAxisCtrl[myBelt.AxisNo].SyncMode.InputRevolution:= 1;
arAxisCtrl[myBelt.AxisNo].SyncMode.OutputRevolution:= 1;
arAxisCtrl[myBelt.AxisNo].SyncMode.FineAdjust:= 0;
arAxisCtrl[myBelt.AxisNo].SyncMode.CamShaftDistance:= myProductCycle;
arAxisCtrl[myBelt.AxisNo].SyncMode.CamTableID:= CAM_TABLE_1;
arAxisCtrl[myBelt.AxisNo].Admin._OpMode:= ModeSyncCam;

```

Der Synchronisationsabstand muss SHORTEST_WAY sein, um sicherzustellen, dass sich der SmartBelt vorwärts und rückwärts bewegt.

Der Startmodus muss auf RELATIVE gesetzt sein. Dadurch kann der Benutzer den SmartBelt mit einer Leitachse im Stillstand synchronisieren, ohne dass sich das Band bewegt.

Der Eingang "CamShaftDistance" muss auf den Produktzyklus der Anwendung gesetzt werden. Dies ist die Länge des Produkts plus der Abstand zwischen den Produkten. Bei dieser Länge muss der SmartBelt festlegen, ob die korrekte Bewegung vorwärts oder rückwärts ist. Auch wenn das Verhältnis zwischen Leitachse und SmartBelt nicht 1:1 ist, muss dieser Wert für den Produktzyklus gesetzt werden. In diesem Fall wird das Getriebe eingesetzt, um das Verhältnis zwischen Leitachse und SmartBelt zu wählen.

Die im Antrieb verwendete ID der Kurvenscheibe kann beliebig sein, muss aber entsprechend dem "CamTableID"-Element gesetzt werden. Vor seiner Verwendung muss die Kurvenscheibe auf den Antrieb heruntergeladen werden. Die Kurvenscheibe muss im Verhältnis 1:1 sein. Diese kann mit Hilfe der CamBuilder-Software erstellt werden. Um die Downloadzeit zu verringern, kann die Anzahl der Kurvenscheibenpunkte auf die minimal möglichen Punkte reduziert werden.

Einem Übersetzungsverhältnis von 1:1 folgt einem 1:1-Verhältnis von Leitachszyklus zu Produktzyklus. Ist das Verhältnis zwischen den Zyklen anders, muss das Übersetzungsverhältnis entsprechend eingestellt werden.

Angabe des "SmartBeltPitch"

Der "SmartBeltPitch" gibt die physikalisch wirksame Länge des SmartBelt an. Die Länge des SmartBelts sollte ungefähr der Länge des Produktes + Lücke zwischen den Produkten (= Produktzyklus) entsprechen. Der Messtaster ist am Anfang des SmartBelts montiert und erfasst entweder den Produktanfang oder das Produktende. Der SmartBelt Funktionsbaustein nimmt dabei an, dass sich das Produkt nach der Erfassung durch den Messtaster um mindestens den "SmartBeltPitch" weiter bewegen muss, bevor es das Band verlässt und dann erst kann mit der Korrektur eines nachfolgenden Produktes begonnen werden kann. Dies ist in der nachfolgenden Zeichnung zu sehen. Hier ist der Produktzyklus etwas kleiner als die Länge des SmartBelts. Im gezeichneten Fall würde sich das Produkt 1 noch bis zur gestrichelt dargestellten Position weiter bewegen, bevor das nachfolgende Produkt (2) korrigiert werden kann.

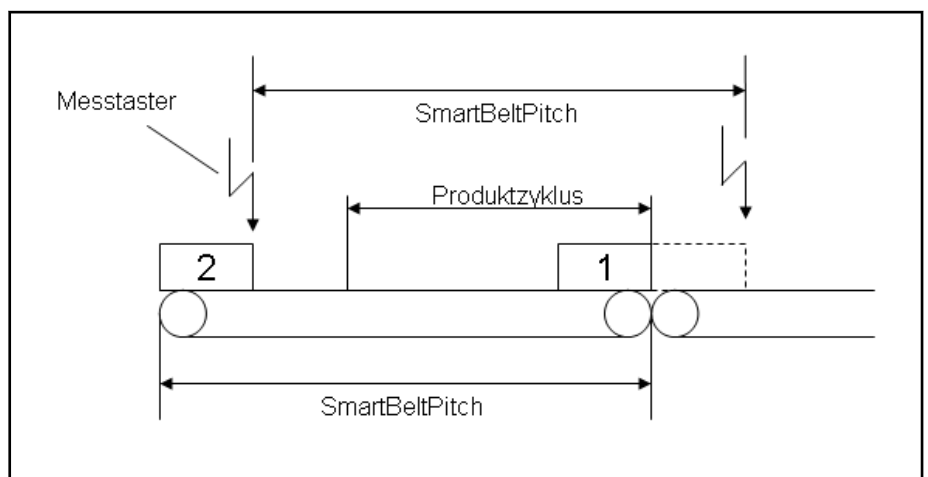


Abb. 4-60: SmartBelt bei dem die physikalische Länge in etwa dem Produktzyklus entspricht

Diese Annahmen sind korrekt solange der Messtaster am Anfang des SmartBelts in dem Bereich montiert wird, die maximal der Länge des Produktes entspricht (siehe nachfolgendes Bild). In diesem Fall ist der SmartBelt etwas

länger als zwei Produktzyklen. Ausgehend vom zweiten Band ist in dieser Zeichnung auch noch ein Produktzyklus A eingezeichnet, der ausgehend vom Messtaster des zweiten Bandes rückwärts eingezeichnet ist. Daraus ist ersichtlich, dass ein nachfolgendes Produkt sich immer noch auf dem ersten Band befindet, wenn das erste Produkt die Länge von "SmartBeltPitch" durchfahren hat. Damit ist dann eine Korrektur des zweiten Produktes noch möglich.

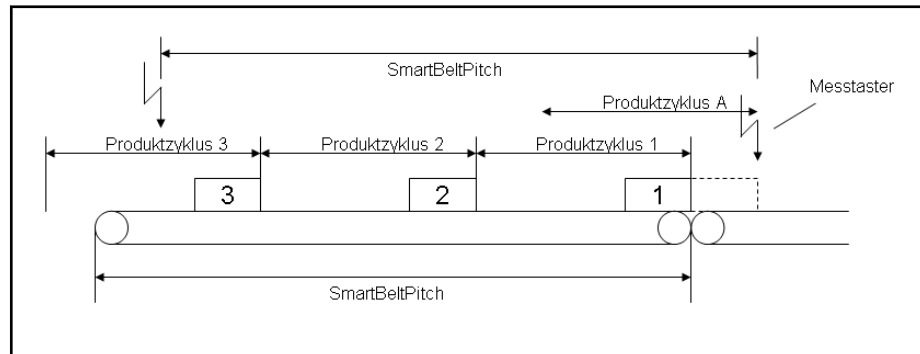


Abb. 4-61: SmartBelt mit physikalischer Länge die ein vielfaches des Produktzyklus entspricht

Wird der Messtaster allerdings nicht am Anfang des SmartBelts montiert, so arbeitet der SmartBelt nicht korrekt, da er nach dem Erkennen des Produktes dieses noch eine volle "SmartBeltPitch" Länge weiter bewegen möchte. In diesem Fall ist das Produkt bereits lange vom Band bevor das nachfolgende Produkt korrigiert wird. Im eingezeichneten Fall wäre auch das nachfolgende Produkt bereits vom Band, da Produktzyklus A vollständig auf dem nachfolgenden Band liegt. Eine Korrektur des nachfolgenden Produktes wäre damit nicht mehr möglich.

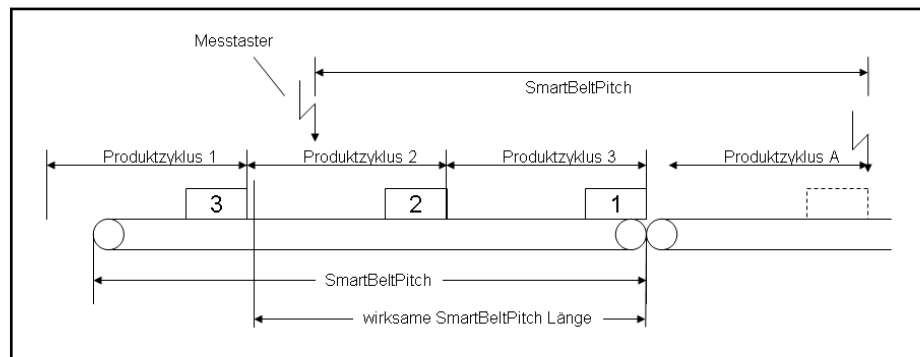


Abb. 4-62: SmartBelt mit Messtaster der nicht am Anfang des Bandes montiert ist

Um dies sicherzustellen, darf nicht die physikalische Länge angegeben werden, sondern es muss die wirksame Länge des SmartBelts angegeben werden. Diese entspricht dem Abstand vom Ende des Bandes bis zum Messtaster + einer Produktlänge. Diese Strecke ist auch in dem Bild eingezeichnet. Die Differenz zwischen der wirksamen Länge und der physikalischen Länge ist der Bereich am Anfang des SmartBelts in dem das Produkt zwar auf dem Band ist, aber noch nicht vom Messtaster erfasst wurde. Damit wird es in diesem Bereich auch noch nicht korrigiert und ist damit unerheblich für den Funktionsbaustein.



Für die nachfolgende Berechnung des "ProbeSetpoint" muss die physikalische Länge des SmartBelts und nicht die wirksame Länge verwendet werden!

Berechnung des "ProbeSetpoint"

Der Eingang "ProbeSetpoint" erkennt die gewünschte Leitachsposition in dem Moment wenn das Produkt den Messtaster des SmartBelts passiert. Dieser Punkt kann berechnet werden, indem die Entfernung der Messtasterposition und das Ende aller SmartBelts, z.B. wo das Produkt ausgerichtet sein muss, benutzt werden. Die folgende Abbildung stellt dies dar:

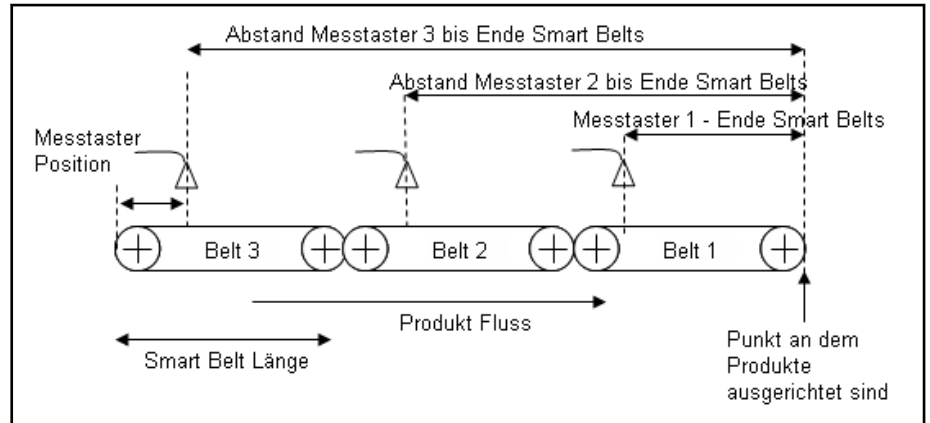


Abb. 4-63: Messungen um den "ProbeSetpoint" zu berechnen

Zunächst müssen alle Abstände zwischen den Messtastern bis zum Ende aller SmartBelts, z.B. wo die Produkte ausgerichtet werden, gemessen werden. Anschließend kann die nachfolgende Formel benutzt werden, um den Wert für die **positive Flankenerkennung** auf dem Messtastereingang zu berechnen:

$$\text{ProbeSetpoint}_k = L_{\text{cycl}} - \left(\left(L_{\text{SmartBelt}_n} - L_{\text{ProbePos}_n} \right) + \sum_{n=1}^{n=k-1} L_{\text{SmartBelt}_n} \right) \text{MOD} (L_{\text{cycl}})$$

Abb. 4-64: Formel für die positive Flankenerkennung

Die nachfolgende Formel gilt für die **negative Flankenerfassung** des Messtastereingangs:

$$\text{ProbeSetpoint}_k = L_{\text{cycl}} - \left(\left(L_{\text{SmartBelt}_n} - L_{\text{ProbePos}_n} - L_{\text{Prod}} \right) + \sum_{n=1}^{n=k-1} L_{\text{SmartBelt}_n} \right) \text{MOD} (L_{\text{cycl}})$$

Abb. 4-65: Formel für die negative Flankenerkennung

Die nachfolgenden Variablen werden in den Berechnungen verwendet:

- L_{cycl} : Länge des Produktzyklus (Produktlänge (L_{Prod}) plus Länge des Abstands)
- k : Anzahl der verwendeten SmartBelts
- L_{ProbePos} : Messtasterposition an einem bestimmten SmartBelt (gemessen vom Anfang des jeweiligen SmartBelts)
- $L_{\text{SmartBelt}}$: Länge des SmartBelts

Angenommen 3 SmartBelts, alle 150 mm lang, mit einem Messtaster, der 30mm vom Anfang des SmartBelts angebracht ist und einem Produktzyklus von 120 mm sowie der Vorgabe die positive Produktflanke zu erkennen. Die folgenden "ProbeSetpoint"-Werte können dann wie folgt berechnet werden:

- SmartBelt 1: $k=1$; $\text{ProbeSetpoint}_1 = 120 - (((150-120) + 0) \text{MOD} (120)) = 120 - (30 \text{MOD} 120) = 90$

- SmartBelt 2: $k=2$; $\text{ProbeSetpoint}_2 = 120 - (((150-120) + 150) \text{ MOD } (120)) = 120 - (180 \text{ MOD } 120) = 60$
- SmartBelt 3: $k=3$; $\text{ProbeSetpoint}_3 = 120 - (((150-120) + 300) \text{ MOD } (120)) = 120 - (330 \text{ MOD } 120) = 30$

4.6.5 SmartBelt-Beschränkungen und -Verhalten

Beschränkungen

Der Funktionsbaustein wird für Winkelkorrekturen von Produkten auf einem Band verwendet. Die Produkte müssen mit einem Abstand auf dem Band platziert werden, damit jedes Produkt identifiziert werden kann. Ist die Distanz zu gering, wird das zweite Produkt nicht erkannt und führt zu einer falschen Korrektur.

Die Grenzwerte für Beschleunigung und/oder maximale und minimale Entfernungen müssen Werte sein, die erreicht werden können. Ansonsten rutschen die Produkte auf dem Band. Dies führt zu falschen Korrekturen.

Idealerweise ist auf dem Band immer nur ein Produkt. Sind zwei oder mehr Produkte auf dem Band, wird das zweite Produkt erst korrigiert, nachdem das erste Produkt das Band verlassen hat. Der verbleibende Weg für die Korrektur ist daher geringer. Zusätzlich erfährt auch das zweite Produkt die Korrekturen des ersten Produktes. Beide Werte werden beim Berechnen der Betragskorrektur für das zweite Produkt berücksichtigt. Jedoch ist es nicht möglich, dass das zweite Produkt genauso korrigiert wird als wäre es alleine auf einem Band.

Zwei oder mehr Produkte auf einem Band

Sind zwei Produkte auf einem Band, wird das zweite Produkt nicht korrigiert bis das erste Produkt das Band verlassen hat. In diesem Fall heißt "das Band verlassen", dass das erste Produkt einen Weg "SmartBeltPitch" vom Registrierungspunkt zurückgelegt hat. In der nachfolgenden Abbildung hat das Produkt, beschrieben mittels einer gepunkteten Linie, "das Band verlassen", da es seit der Registrierung auf Band 1 (Blickwinkel: positive Flanke des Produkts) eine ganze "SmartBeltPitch" gefahren ist. Dieser Wert ist fest vorgegeben.

Das zweite Produkt hat weniger Zeit/Abstand, um Korrekturen vorzunehmen. Der verbleibende Abstand ist der "DistanceForCorrection" minus des bereits gefahrenen Weges. Mit diesem Wert können alle Variablen (Geschwindigkeit, Beschleunigung und Zeit) für das zweite Produkt berechnet werden.

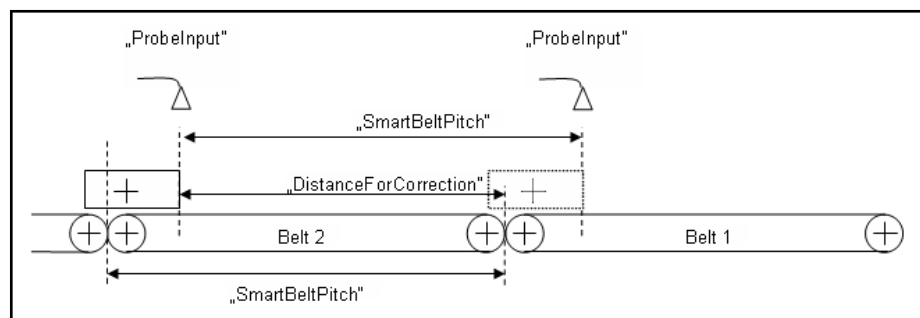


Abb. 4-66: Zwei oder mehr Produkte auf einem Band

Ein zusätzliches Problem ist, dass das zweite Produkt stets rückwärts korrigiert werden muss. Die Korrektur wird aus den Einstellungen in "PercentForwardLimit", "MaxCorrectionDistancePos" und "MaxCorrectionDistanceNeg" berechnet. Wenn ein zweites Produkt auf dem Band ist und das erste Produkt gerade von Band genommen wurde, berechnet der Algorithmus einen neuen Korrekturwert für das zweite Produkt. Dieser könnte bewirken, dass

das zweite Produkt nach vorne geschoben wird. Dies ist jedoch nicht möglich, da dort das erste Produkt ist. In diesem Fall wird ein ganzer Produktzyklus abgezogen, um das Produkt auf den nächsten freien Platz zu schieben. Jede Rückwärtsbewegung ist auf "MaxCorrectionDistanceNeg" begrenzt.

Mehrere Produkte pro Leitachszzyklus

Wenn mehrere Produkte oder Teile eines Produkts einem Leitachszzyklus entsprechen, müssen zusätzliche Einstellungen vorgenommen werden. Das Übersetzungsverhältnis bestimmt die Beziehung zwischen den Zyklen.

Wenn beispielsweise eine Leitachsumdrehung drei Produkten entspricht, müssen die Eingänge "RatioNumerator"/"InputRevolution" auf 1 und "RatioDenominator"/"OutputRevolution" auf 3 gesetzt werden (siehe auch [SPS-Programmierung, Seite 253](#)).



Es ist wichtig zu vermerken, dass der Eingang "CamShaftDistance" nicht verwendet werden kann, um das Verhältnis zwischen Leitachse und SmartBelt zu ändern. Ist "CamShaftDistance" nicht auf den Produktzyklus abgestimmt, kommt es zu falschen Korrekturen!

4.6.6 Tipps und Tricks

Korrekturabstand

Die vorhergehenden Abschnitte in diesem Kapitel beschreiben die Konfiguration der SmartBelt-Funktionalität. Dieser Abschnitt gibt zusätzliche Informationen, damit Abhängigkeiten deutlich werden und Optimierungen an dem bereits laufenden System durchgeführt werden können.

Der SmartBelt verwendet die Daten des Eingangs "DistanceForCorrection" um die zur Korrektur benötigten Abstände zwischen den Produkten zu berechnen. Dies ist normalerweise der Abstand zwischen der Messtasterposition und dem Bandende. Für Produkt A ist das in der Abbildung unten dargestellt. Dies ist sicherer, da das Produkt nur solange korrigiert wird, wie es sich noch vollständig auf dem Band befindet.

Produkt B verfügt über einen längeren Weg für die Korrektur. Der Massenschwerpunkt des Produktes ist an derselben Stelle wie die Mitte der SmartBelt-Achse. Hier ist die Möglichkeit des Schlupfes immer noch gering.

Bei Produkt C ist der Masseschwerpunkt zwischen den beiden angrenzenden SmartBelts. In diesem Fall kommt es bereits zu Schlupf.

Der Wert "DistanceToCorrect" kann von A bis C erhöht werden, um die Betragskorrektur zu erhöhen, während das Rutschen überwacht wird. Es hängt von der Anwendung, hauptsächlich von der Reibung und dem Durchmesser der SmartBelt-Achse ab, wie stark der Wert des "DistanceToCorrect" erhöht werden kann. Mit diesem Wissen kann die Betragskorrektur optimiert werden.

4.6.7 Verbreitete Fehler und Missverständnisse

Frage	Antwort:
Der Antrieb synchronisiert nicht oder braucht sehr lange für die Synchronisation	Wenn der Antrieb in den linearen Modus geschaltet ist, ändern sich die Einheiten, aber die Standardwerte bleiben gleich. Die Beschleunigung und die Geschwindigkeit sind normalerweise sehr niedrig, so dass eine Synchronisation sehr langsam ist. Erhöhen Sie die Werte für A-0-2790 und A-0-2791 (MLC) bzw. P-0-0142 und P-0-0143 (MLD)
Fehler F2002028 (MLC) "Exzessive Regelabweichung" und F2028 (MLD) treten ständig auf	Setzen Sie die Parameter A-0-0032, A-0-0033 und A-0-0034 (MLC) bzw. S-0-0038, S-0-0039 und S-0-0138 (MLD) auf richtige Werte. Die Standardwerte für diese Parameter passen zu einer rotatorischen Wichtung, aber für die lineare Wichtung sind sie normalerweise zu klein und müssen angepasst werden
Der Funktionsbaustein ist so konfiguriert, dass er 100% korrigiert, aber das Produkt befindet sich nicht in der korrekten Position, nachdem es vom Band gekommen ist	Dafür kann es mehrere Gründe geben: Erstens, die Winkeldifferenz war wahrscheinlich zu groß, um innerhalb einer Bandlänge korrigiert zu werden. In diesem Fall werden die Ausgänge "Neg-CorrLimited", "PosCorrLimited" oder "AccelLimited" gesetzt. Zweitens, wenn die Werte "MaxCorrectionDistancePos", "MaxCorrectionDistanceNeg" oder "MaxAcceleration" zu hoch gesetzt werden, kann es sein, dass die Produkte physisch nicht dazu in der Lage sind, dem Band zu folgen und rutschen. Diese Bedingung wird nicht von den Statusausgängen angezeigt. Es können Maßnahmen ergriffen werden um die Reibung zu erhöhen (Vakuum Bänder) oder es müssen mehr Bänder verwendet werden, um die Fehler auf mehreren Bändern zu kompensieren. Drittens, sind zwei oder mehr Produkte auf dem Band, siehe nächste Frage
Befinden sich zwei oder mehr Produkte auf dem Band, so wird das zweite oder dritte Produkt nicht korrekt korrigiert	Mit der Korrektur des zweiten Produkts muss gewartet werden, bis das erste Produkt das Band verlassen hat. Dann wird es korrigiert. Ist der verbleibende Abstand zu klein, kann die notwendige Korrektur nicht erreicht werden
Befinden sich zwei Produkte Stoß auf Stoß auf dem Band, wird das zweite Produkt nicht korrigiert	Wenn die Produkte Stoß auf Stoß liegen, kann die Messtasterfunktion das zweite Produkt nicht erkennen. Der SmartBelt sieht nur ein Produkt. Es muss sichergestellt werden, dass der - den SmartBelts vorgelagerte Prozess - die Produkte mit einem Minimalabstand voneinander trennt, damit die Messtasterfunktionalität das nächste Produkte erkennen kann
Alles ist konfiguriert, aber der SmartBelt nimmt keine Korrekturen vor und es liegt auch kein Fehler vor	Setzen Sie Bit 9 des Parameters P-0-0226 im Antrieb und schalten Sie in Phase 2 und dann wieder in Phase 4. Dies aktiviert die Messtasterfunktionalität automatisch. Sonst ist die Messtasterfunktionalität zwar konfiguriert aber nicht aktiviert. Dies führt dazu, dass keine Messtastermessungen durchgeführt werden und dadurch der Smart Belt auch kein Produkt erkennt

4.7 Kurvenscheibentabellen für Technologiefunktionen

4.7.1 Einführung und Übersicht

Die Funktionsbausteine dienen der Erstellung von Kurvenscheibentabellen. Der Funktionsbaustein [MB_PilgrimStepCalcType01](#), Seite 262, übernimmt die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunkten sowie der Hübe des Vorwärts- und Rückwärtstransportes und die Stopppositionen für den Pilgerschrittbetrieb.

Der Funktionsbaustein [MB_PrintLengthCorrCalcType01](#), Seite 272, übernimmt die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunkten für die Technologiefunktion Drucklängenkorrektur, um die Länge eines Druckes auf dem Druckgut (Warenbahn bzw. Bogen) zu verändern.

4.7.2 MB_PilgrimStepCalcType01

Kurzbeschreibung Der Funktionsbaustein berechnet eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen sowie die Hübe des Vorwärts- und Rückwärtstransportes sowie die Stoppositionen für den Pilgerschrittbetrieb. Er berücksichtigt dabei, dass der synchrone Bereich des Druckzylinders größer sein kann, als die zu druckende Formatlänge. Beispielsweise ist bei Offsetdruckwerken die Plattenlänge durch die mechanische Einspannvorrichtung fest vorgegeben und damit immer konstant. Das auf der Druckplatte befindliche Druckformat kann jedoch beliebig kürzer sein.

Durch die vom Baustein berechnete Kurvenscheibe werden die Infeed- und Outfeed-Walze der Druckmaschine betrieben.

Zuordnung Zielsystem/Bibliothek

MLC/XLC 14VRS Funktionspaket TEC	ML_TechMotion

Tab. 4-42: Referenztable MB_PilgrimStepCalcType01

Schnittstellenbeschreibung

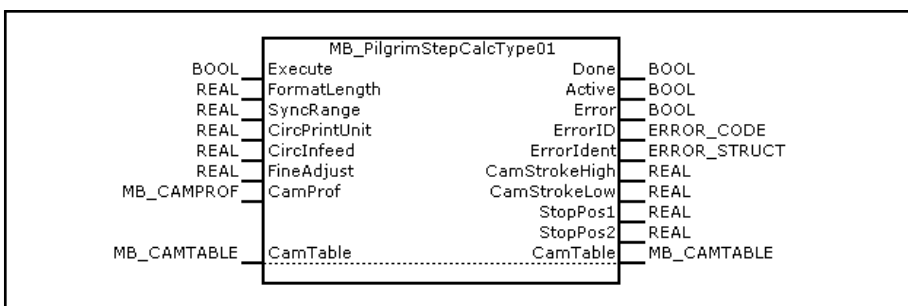


Abb. 4-68: Funktionsbaustein MB_PilgrimStepCalcType01

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	FormatLength	REAL	Länge des zu druckenden Formates [mm]
	SyncRange	REAL	Länge des synchronen Bereichs [mm]
	CircPrintUnit	REAL	Umfang des Druckzylinders [mm]
	CircInfeed	REAL	Umfang der Zugwalzen [mm]
	FineAdjust	REAL	Feinabgleich [%] – definiert die gewünschte Bahnspannung
	CamProf	MB_CAMPROF	Auswahl des Kurvenscheibenprofils MB_CAMPROF, Seite 102
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe Fehlerbehandlung, Seite 272)
	CamTable	MB_CAMTABLE	

E/A-Typ	Name	Typ	Kommentar
	CamStrokeHigh	REAL	Hub P-0-0073 Outfeed / P-0-0093 Infeed Der P-0-0093 wird über den Funktionsbaustein MC_CamIn / Eingang "CamShaftDistance" vorgegeben Der P-0-0073 wird über den Funktionsbaustein MB_WriteRealParameter vorgegeben
	CamStrokeLow	REAL	Hub P-0-0073 Infeed / P-0-0093 Outfeed Der P-0-0093 wird über den Funktionsbaustein MC_CamIn / Eingang "CamShaftDistance" vorgegeben Der P-0-0073 wird über den Funktionsbaustein MB_WriteRealParameter vorgegeben Verwendete Beschleunigung in Relation zum Eingang "MaxAcc" in [%]
	StopPos1	REAL	Umkehrpunkt 1 [°] mit Geschwindigkeit = 0
	StopPos2	REAL	Umkehrpunkt 2 [°] mit Geschwindigkeit = 0
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellenelementen und Information über die Anzahl der Elemente MB_CAMTABLE, Seite 102

Tab. 4-43: Schnittstellenvariablen Funktionsbaustein MB_PilgrimStepCalcType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteinneingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
FormatLength	REAL	>0.0	n.def.	100.0	Steigende Flanke an "Execute"
SyncRange	REAL	>0.0	n.def.	100.0	Steigende Flanke an "Execute"
CircPrintUnit	REAL	>0.0	n.def.	100.0	Steigende Flanke an "Execute"
CircInfeed	REAL	>0.0	n.def.	100.0	Steigende Flanke an "Execute"
FineAdjust	REAL	0.0	100.0	1.0	Steigende Flanke an "Execute"
CamProf	MB_CAMPROF			CAM_PROF_STANDARD	Steigende Flanke an "Execute"

Tab. 4-44: Min-/Max- und Default-Werte Funktionsbaustein MB_PilgrimStepCalcType01

Die Enumeration MB_CAMPROF, ML_TechMotionBase, wird verwendet um zwischen verschiedenen Kurvenscheibenprofilen zu wählen.



Bei der Auswahl des Kurvenscheibenprofils "CAMPROF_STANDARD" muss der Linearpfad deaktiviert sein, da es sich um ein Profil ohne Untersetzung handelt. Während bei den Profilen "CAMPROF_ADDITIVE" und "CAMPROF_ADDITIVE_SEGMENT" der Linearpfad über ein Häkchen aktiviert werden muss und die Reduktion auf den Wert "1" zu stellen ist, da es sich um Profile mit Untersetzung handelt.

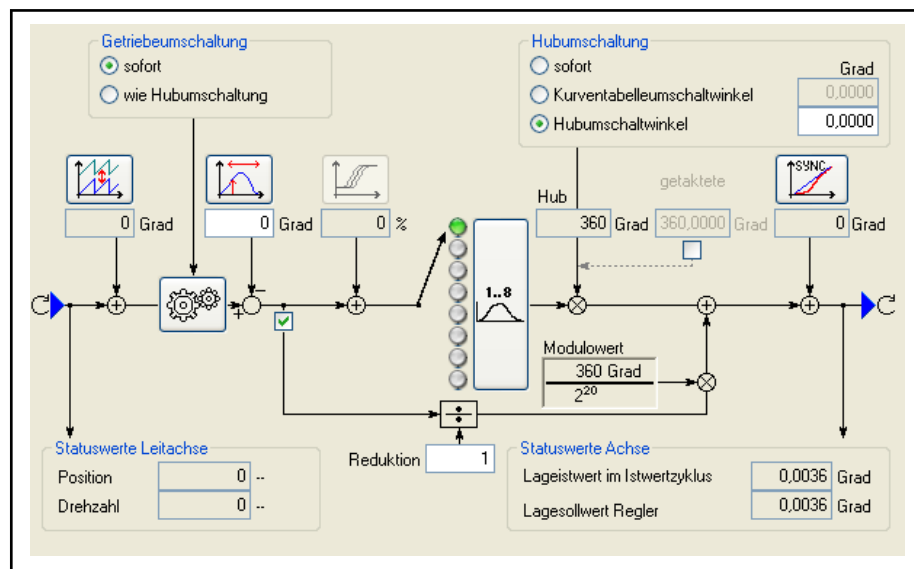


Abb. 4-69: Aktivierung des Linearpfades

Die Struktur MB_CAMTABLE, ML_TechMotionBase, besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

Funktionsbeschreibung

Wird nicht der gesamte Umfang eines Druckzylinders mit einer Druckplatte bzw. Druckklischee bedeckt, wechseln sich im kontinuierlichen Betrieb bedruckte mit nicht bedruckten Stellen ab. Um diese Druck-Lücken zu vermeiden wird die Materialbahn durch die Transportwalzen abgebremst, einen definierten Weg zurückgezogen und erneut auf synchrone Geschwindigkeit beschleunigt, so dass sich der neue Druck sofort nach dem alten anschließt.

Diese Ausgleichsbewegung wird durch den Funktionsbaustein berechnet und ist durch eine Stetigkeit bis zur Beschleunigung charakterisiert.

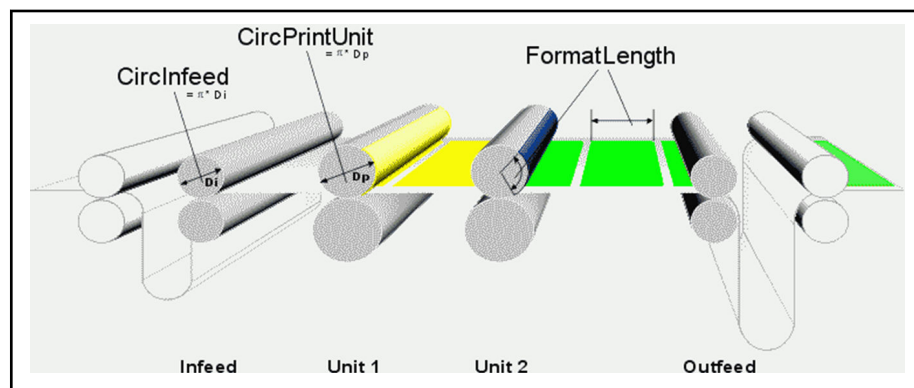


Abb. 4-70: Prinzipdarstellung Pilgerschritt

Voraussetzungen zur Nutzung des Funktionsbausteins

- Der Funktionsbaustein ist nur für Achsen mit Interpolation auf Antrieb nutzbar
- Die Antriebsfunktion „Getaktete Zugwalze“ (siehe Antriebsdokumentation P-0-0086) muss vom Anwender eingeschaltet werden

Aus den mechanischen Angaben des Umfangs der Infeed- bzw. Outfeedwalze "CircInfeed", des Umfangs des Druckzylinders "CircPrintUnit", der Klichscheellänge "FormatLength" auf dem Druckzylinder und der Gesamtlänge des Synchronbereichs "SyncRange" berechnet der Funktionsbaustein MB_PilgrimStepCalcType01 nach der Bearbeitungsfreigabe mittels "Execute" einmalig eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen, sowie die beiden Hubfaktoren "CamStrokeHigh" und "CamStrokeLow" und die Stoppositionen "StopPos1" und "StopPos2". Hierbei wird die Anzahl der Stützpunktelemente einer Kurvenscheibentabelle durch das Element "NumberOfElements" der Struktur "CamTable" eingestellt.

Es wird davon ausgegangen, dass der Umfang der Infeedwalze und der Umfang der Outfeedwalze identisch sind (CircInfeed).

Entsprechend der durch den Feinabgleich "FineAdjust" vorgegebenen Voreilung werden 2 Hübe berechnet, die dazu dienen, die Bahnspannung konstant zu halten. Der jeweils größere Hub ist für die in Transportrichtung vordere Walze bestimmt. Zur Berechnung der Hübe verwendet der Baustein folgende Formeln:

$$CamStrokeLow = 360^\circ * \frac{FormatLength}{CircInfeed}$$

Der Wert von CamStrokeLow gilt für den Parameter P-0-0073 der Einzugswalze und für den Parameter P-0-0093 der Auszugswalze

Abb. 4-71: CamStrokeLow Formel

$$CamStrokeHigh = 360^\circ * \frac{FormatLength}{CircInfeed} * \left(1 + \frac{FineAdjust}{100} \right)$$

Der Wert von CamStrokeHigh gilt für den Parameter P-0-0073 der Auszugswalze und für den Parameter P-0-0093 der Einzugswalze

Abb. 4-72: CamStrokeHigh Formel

Die Ausgänge "StopPos1" und "StopPos2" enthalten die Leitachspalten, bei denen die Kurvenscheibe einen Umkehrpunkt hat, um z.B. an einer dieser Positionen während einer Regelung einen Formatwechsel durchzuführen oder per Registerregler den Hub zu ändern.

Die zu berechnende Kurvenscheibentabelle wird in der Struktur "CamTable" mit Wichtung in Prozent abgelegt. Die Gültigkeit der Kurvenscheibenelemente wird durch den Ausgang "Done" signalisiert. Die Kurvenscheibentabelle kann mittels des FB MB_LoadCamData, ML_TechMotionBase, in eine Kurvenscheibe in der Steuerung oder in den Antrieb geladen werden.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" sowie die Ausgänge "CamStrokeHigh", "CamStrokeLow", "StopPos1" und "StopPos2" werden im Fehlerfall nicht aktualisiert.

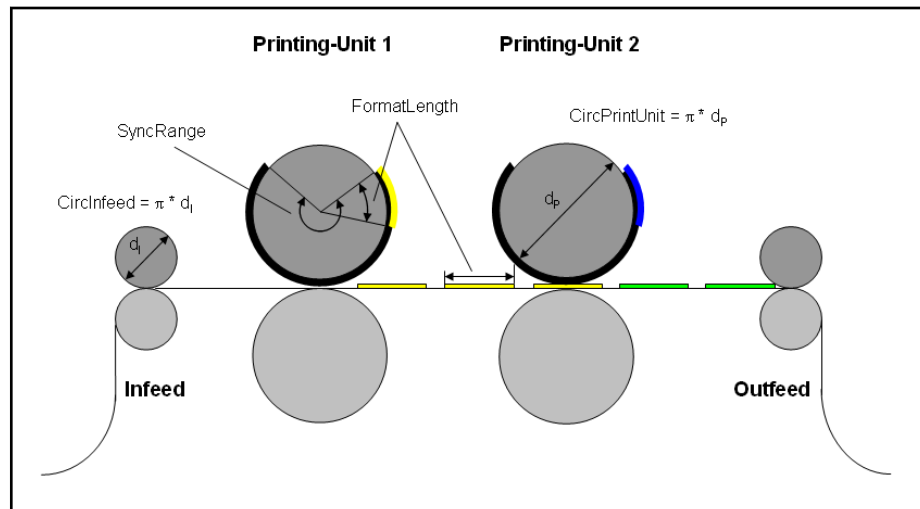


Abb. 4-73: Prozess-Schema ($FormatLength < SyncRange$)

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins MB_PilgrimStepCalc-Typ01 weist folgende Eigenschaften auf:

- Der Funktionsbaustein basiert auf dem Funktionsbaustein MSV_Cam-TableType04 und deckt zusätzlich Fälle ab, bei denen der Synchronbereich des Druckzylinders größer sein muss, als der Druckbereich
- Die durch die Kurvenscheibentabelle charakterisierte Ausgleichsbewegung zum Rücktransport der Bahn ist stetig bis zur Beschleunigung (s , v , a) und weist einen Sprung im Ruck (j) auf
- Die Ausgleichsbewegung beginnt beim ersten Tabellenelement, nachfolgend schließt sich eine lineare Steigung winkelsynchron zum Druckzylinder an
- Der Algorithmus zur Berechnung der Kurvenscheibenstützpunkte ist auf mehrere Zyklen, d.h. Aufrufe der Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden
- Die berechneten Hübe des Funktionsbausteins weisen folgende Eigenschaften auf:
 - Die Hübe sind abhängig vom Druckformat und dem Umfang der Infeedwalze
 - Der Hub CamStrokeHigh ist für den Parameter P-0-0093 der Infeedwalze und den Parameter P-0-0073 der Outfeedwalze, der Hub CamStrokeLow für den Parameter P-0-0073 der Infeedwalze und den Parameter P-0-0093 der Outfeedwalze vorgesehen

Zur Erzeugung der Bahnspannung sind der Hub für den Vortransport der Outfeedwalze und der Hub für den Rücktransport der Infeedwalze um den Feinabgleich größer als der Hub für den Vortransport der Infeedwalze und den Hub für den Rücktransport der Outfeedwalze.

Es werden drei Arten von Kurvenscheiben unterschieden:

- **FormatLength = SyncRange = CircPrintUnit**
Die Druckplatte bzw. das Druckklischee bedeckt die gesamte synchrone Zone des Zylinders und die synchrone Zone erstreckt sich über den kompletten Umfang des Druckzylinders. In diesem Fall findet keine Ausgleichsbewegung statt. SyncRange gleich CircPrintUnit, muss auf jeden Fall auch FormatLength diesen Wert haben, da keine Ausgleichsbewegung

gung durchgeführt werden kann. Ist FormatLength in diesem Fall kleiner als SyncRange, bricht der Funktionsbaustein die Bearbeitung mit einem Fehler ab.

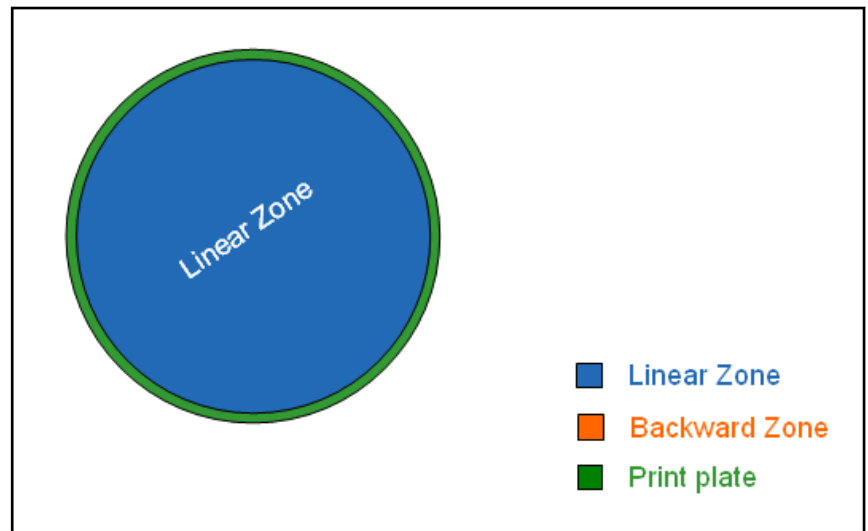
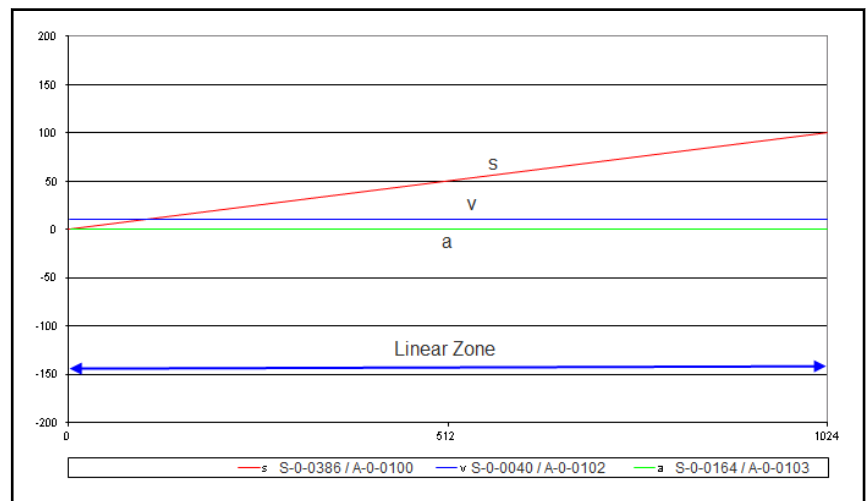


Abb. 4-74: MB_PilgrimStepCalcType01 mit FormatLength = SyncRange = CircPrintUnit



FormatLength = 2000.0; SyncRange = 2000.0; CircPrintUnit = 2000.0; CircInFeed = 2000.0

Abb. 4-75: MB_PilgrimStepCalcType01 Bewegungsprofil

- **FormatLength = SyncRange**

Die Druckplatte bzw. das Druckklischee bedeckt die gesamte synchrone Zone des Zylinders. Der gesamte restliche Zylinderbereich kann für die Ausgleichsbewegung genutzt werden.

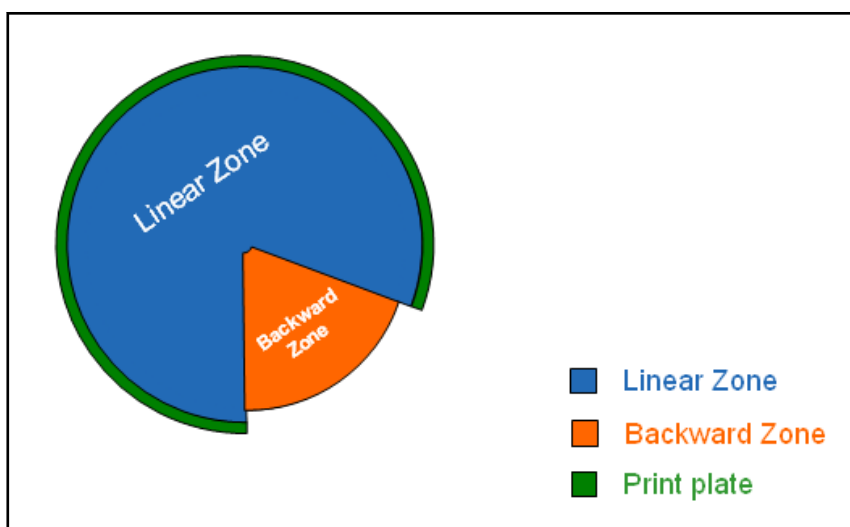
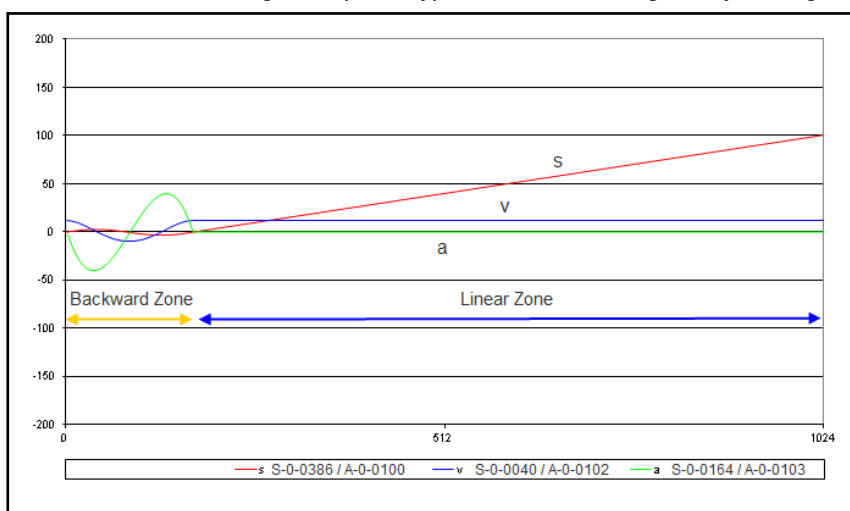
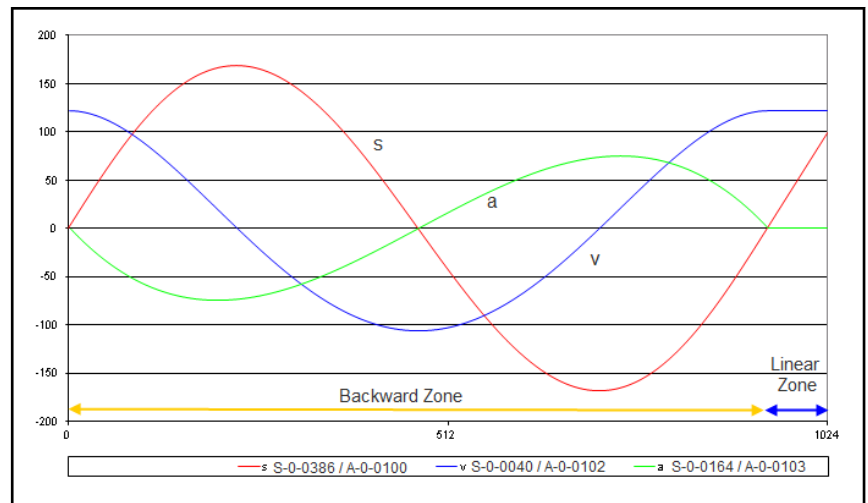


Abb. 4-76: *MB_PilgrimStepCalcType01* mit *FormatLength* = *SyncRange*



FormatLength = 500.0; *SyncRange* = 500.0; *CircPrintUnit* = 600.0; *CircInFeed* = 600.0

Abb. 4-77: *MB_PilgrimStepCalcType01* Bewegungsprofil



FormatLength = 200.0; SyncRange = 200.0; CircPrintUnit = 2500.0; CircInFeed = 2500.0

Abb. 4-78: MB_PilgrimStepCalcType01 Bewegungsprofil

- **FormatLength < SyncRange**

Die Druckplatte bzw. das Druckklischee bedeckt nur teilweise die synchrone Zone des Zylinders. Dieser muss sich bis zum Ende des synchronen Bereichs weiterhin mit synchroner Geschwindigkeit bewegen. Erst wenn der Zylinder keinen Bahnkontakt mehr hat, kann die Materialbahn abgebremst, zurückgezogen und erneut beschleunigt werden. Hier muss auch der zusätzliche Weg des nichtdruckenden synchronen Bereichs zurückgezogen werden.

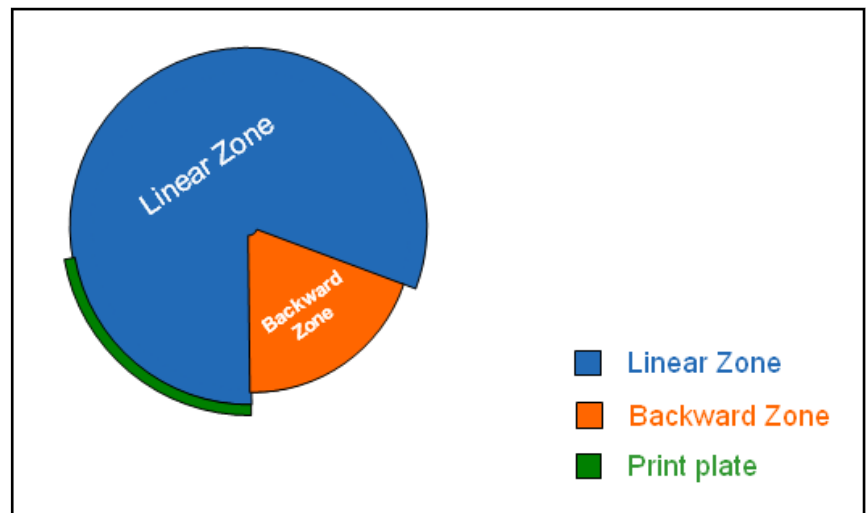


Abb. 4-79: MB_PilgrimStepCalcType01 mit FormatLength < SyncRange

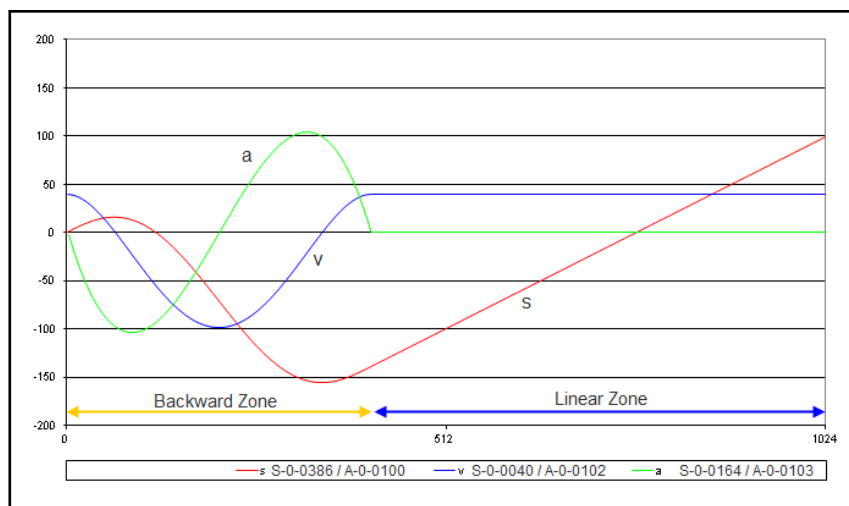


Abb. 4-80: MB_PilgrimStepCalcType01 Bewegungsprofil

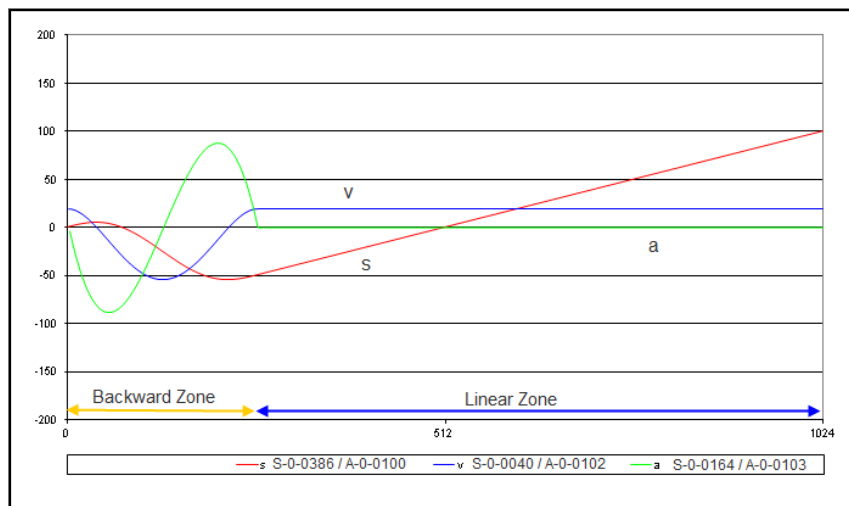
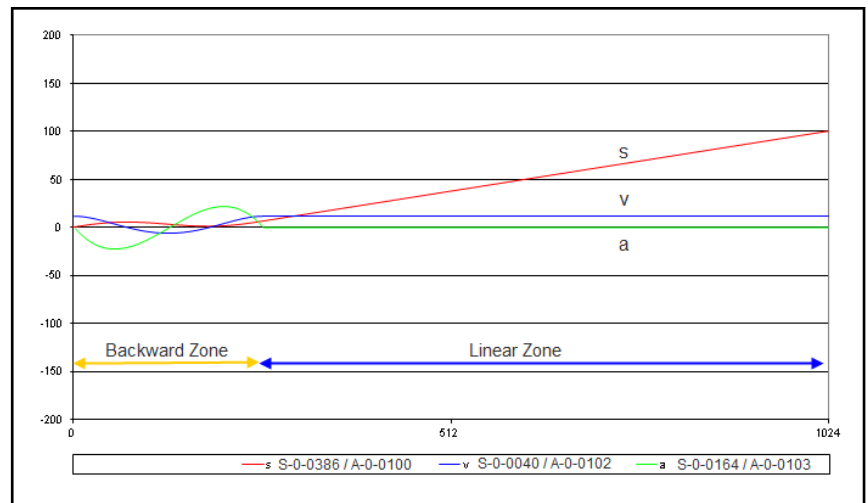


Abb. 4-81: MB_PilgrimStepCalcType01 Bewegungsprofil

- **FormatLength > SyncRange**

Für spezielle Produktionen kann die Formatlänge auch größer als der SyncRange eines Druckzylinders sein.



FormatLength = ; SyncRange = ; CircPrintUnit = ; CircIn-
Feed =

Abb. 4-82: MB_PilgrimStepCalcType01 Bewegungsprofil

Anwendungsfall Spezialprodukt gedruckt mit zwei Druckwerken:



Abb. 4-83: Normales Produkt (Formatlänge 350mm):



Abb. 4-84: Spezielles Produkt (Formatlänge 700mm):

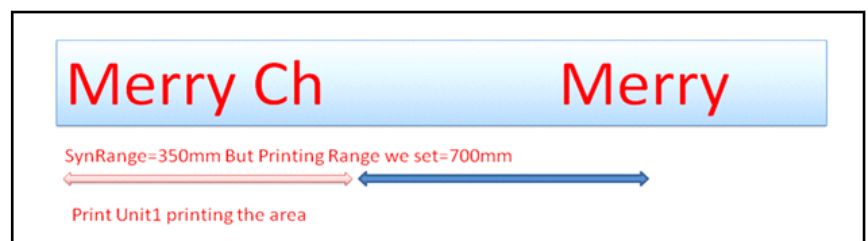


Abb. 4-85: Druckwerk 1 zum Drucken des roten Bereiches



Abb. 4-86: Druckwerk 2 zum Drucken des blauen Bereiches

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR	16#2200	16#0001	Eingang "NumberOfElements" ist < Minimum oder > Maximum
INPUT_RANGE_ERROR	16#2260	16#0001	Eingang "FormatLength" ist < Minimum
INPUT_RANGE_ERROR	16#2260	16#0002	Eingang "SyncRange" ist < Minimum
INPUT_RANGE_ERROR	16#2260	16#0003	Eingang "CircPrintUnit" ist < Minimum
INPUT_RANGE_ERROR	16#2260	16#0004	Eingang "CircInfeed" ist < Minimum
INPUT_RANGE_ERROR	16#2260	16#0005	Eingang "FineAdjust" ist < Minimum oder > Maximum
INPUT_INVALID_ERROR	16#2261	16#0001	Adresse von Feld "CamTable" ist null
INPUT_INVALID_ERROR	16#2261	16#0001	Eingang "FormatLength" ist > CircPrintUnit
INPUT_INVALID_ERROR	16#2261	16#0002	Eingang "SyncRange" ist > CircPrintUnit

Tab. 4-45: Fehlercodes FB MB_PilgrimStepCalcType01

4.7.3 MB_PrintLengthCorrCalcType01

Kurzbeschreibung Der Funktionsbaustein berechnet eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen für die Technologiefunktion Drucklängenkorrektur.

Zuordnung Zielsystem/Bibliothek

MLC 14VRS	ML_TechMotion
Funktionspaket TEC	

Tab. 4-46: Referenztable MB_PrintLengthCorrCalcType01

Schnittstellenbeschreibung

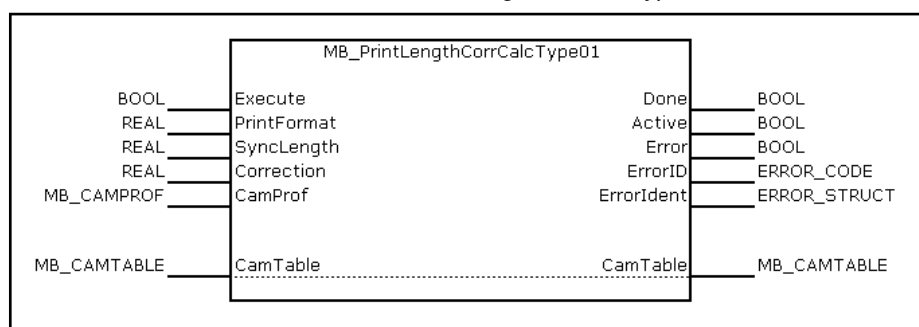


Abb. 4-87: Funktionsbaustein MB_PrintLengthCorrCalcType01

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	PrintFormat	REAL	Erwartetes, nominelles Druckformat [mm]
	SyncLength	REAL	Synchrone Formatlänge, d.h. Umfang des Druckzylinders [mm]
	Correction	REAL	Druckkorrektur zur Kompensation der Abweichung [mm]
	CamProf	MB_CAMPROF	Auswahl des Kurvenscheibenprofils MB_CAMPROF, Seite 102
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe Fehlerbehandlung, Seite 277)
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellenelementen und Information über die Anzahl der Elemente MB_CAMTABLE, Seite 102

Tab. 4-47: Schnittstellenvariablen Funktionsbaustein MB_PrintLengthCorrCalcType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
PrintFormat	REAL	>0.0	n.def	1000.0	Steigende Flanke an "Execute"
SyncLength	REAL	>0.0	n.def	1500.0	Steigende Flanke an "Execute"
Correction	REAL	-9999.9	9999.9	100.0	Steigende Flanke an "Execute"
CamProf	MB_CAMPROF			CAM_PROF_STANDARD	Steigende Flanke an "Execute"

Tab. 4-48: Min-/Max- und Default-Werte Funktionsbaustein MB_PrintLengthCorrCalcType01

Die Enumeration MB_CAMPROF, ML_TechMotionBase, wird verwendet, um zwischen verschiedenen Kurvenscheibenprofilen zu wählen.



Bei der Auswahl des Kurvenscheibenprofils "CAMPROF_STANDARD" muss der Linearpfad deaktiviert sein, da es sich um ein Profil ohne Untersetzung handelt. Während bei den Profilen "CAMPROF_ADDITIVE" und "CAMPROF_ADDITIVE_SEGMENTS" der Linearpfad über ein Häkchen aktiviert werden muss und die Reduktion auf den Wert "1" zu stellen ist, da es sich um Profile mit Untersetzung handelt.

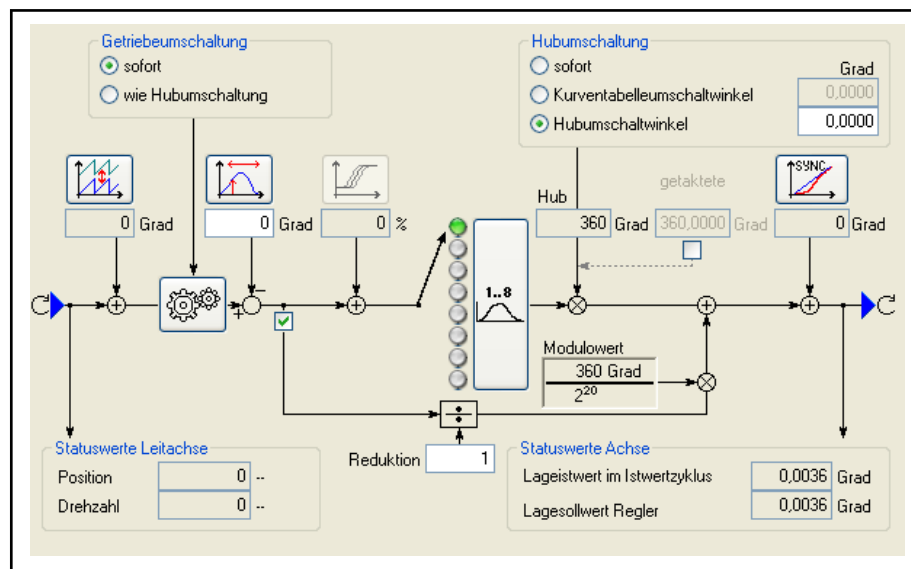


Abb. 4-88: Aktivierung des Linearpfades

Die Struktur MB_CAMTABLE, ML_TechMotionBase, besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

Funktionsbeschreibung

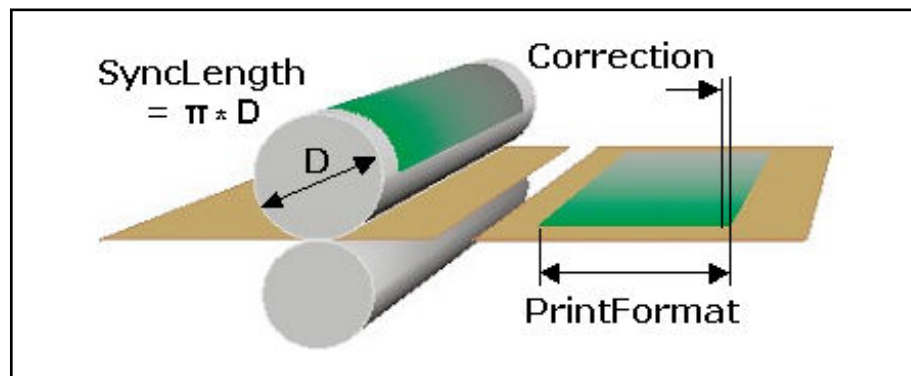


Abb. 4-89: Prinzipdarstellung der Drucklängenkorrektur

Die Drucklängenkorrektur dient dazu, die Länge eines Druckes auf dem Druckgut (Warenbahn bzw. Bogen) zu verändern. Hierzu wird die Geschwindigkeit der Druckwalze zum Druckgut während des Druckkontaktes der Druckwalze vergrößert bzw. verringert. Innerhalb der darauffolgenden druckfreien Zone wird die Position des Druckzylinders derart korrigiert, dass der nächste Druck wieder registerhaltig begonnen wird.

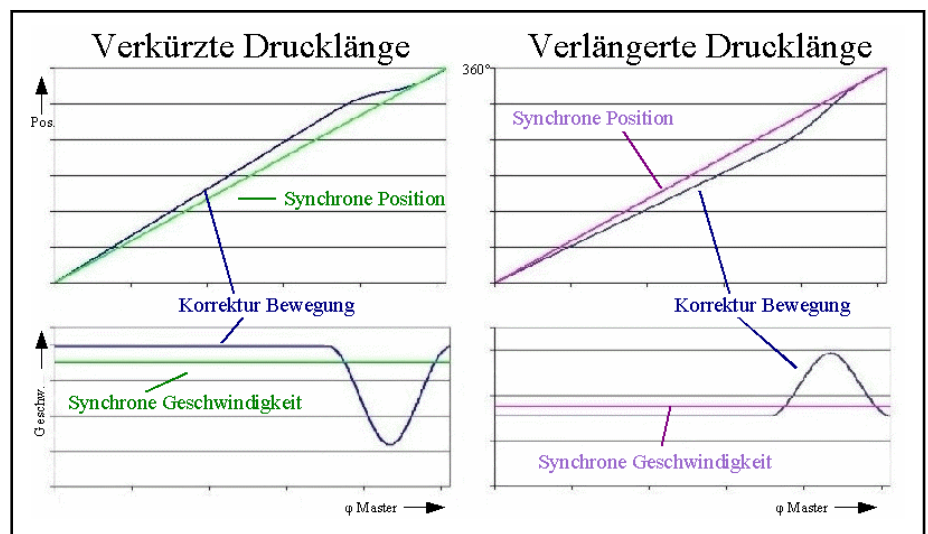


Abb. 4-90: Profile der Drucklängenkorrektur

Mit der Vorgabe der Formate "PrintFormat" sowie "SyncLength" und der Korrektur "Correction" berechnet der Funktionsbaustein "MB_PrintLengthCorr-CalcType01" nach der Bearbeitungsfreigabe mittels "Execute" einmalig auf Basis der Eingangsgrößen eine Kurvenscheibentabelle mit bis zu 1024 Stützpunkten. Hierbei wird die Anzahl der Stützpunktelemente einer Kurvenscheibentabelle durch das Element "NumberOfElements" der Struktur "CamTable" eingestellt.

Die Summe aus Druckformat und Druckkorrektur darf den Umfang der Druckwalze nicht überschreiten. Die notwendigen positiven und negativen Beschleunigungswerte steigen mit sinkender druckfreier Zone.

- Ein positiver Korrekturwert vergrößert den vom Druckzylinder zurückgelegten Weg bei gleicher Leitachseposition und verringert somit das Druckformat

Die zu berechnende Kurvenscheibentabelle wird in der Struktur "CamTable" mit Wichtung in Prozent abgelegt. Die Gültigkeit der Kurvenscheibenelemente wird durch den Ausgang "Done" signalisiert. Die Kurvenscheibentabelle kann mittels des FB MB_LoadCamData, ML_TechMotionBase, in eine Kurvenscheibe in der Steuerung oder in den Antrieb geladen werden.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" werden im Fehlerfall nicht aktualisiert.

Eigenschaften der Kurvenscheibe

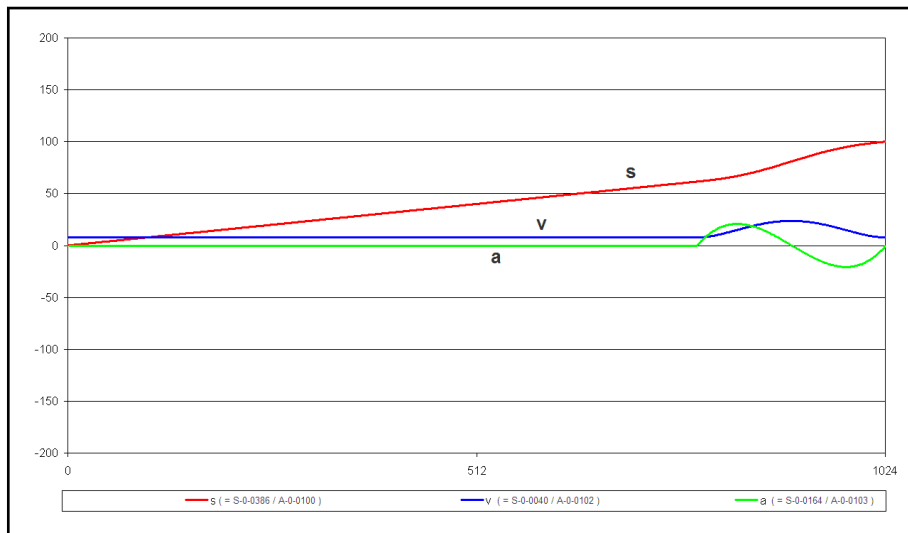
Die Kurvenscheibentabelle des Funktionsbausteins MB_PrintLengthCorr-CalcType01 weist folgende Eigenschaften auf:

- Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist stetig bis zur Beschleunigung (s,v,a) und weist einen Sprung im Ruck (j) auf
- Die Drucklängenkorrektur, d.h. die Verlängerung bzw. Verkürzung des Drucklängenformates basiert auf einer Vergrößerung bzw. Verringerung der Geschwindigkeit des Druckzylinders während des Druckes und abschließenden Ausgleichsbewegung
- Das Druckformat, d.h. das Klischee, darf inkl. der Druckkorrektur nur einen Teil der synchronen Formatlänge betragen, d.h. zwischen zwei Dr-

cken muss eine druckfreie Zone bestehen, welche für die Ausgleichsbewegung genutzt wird

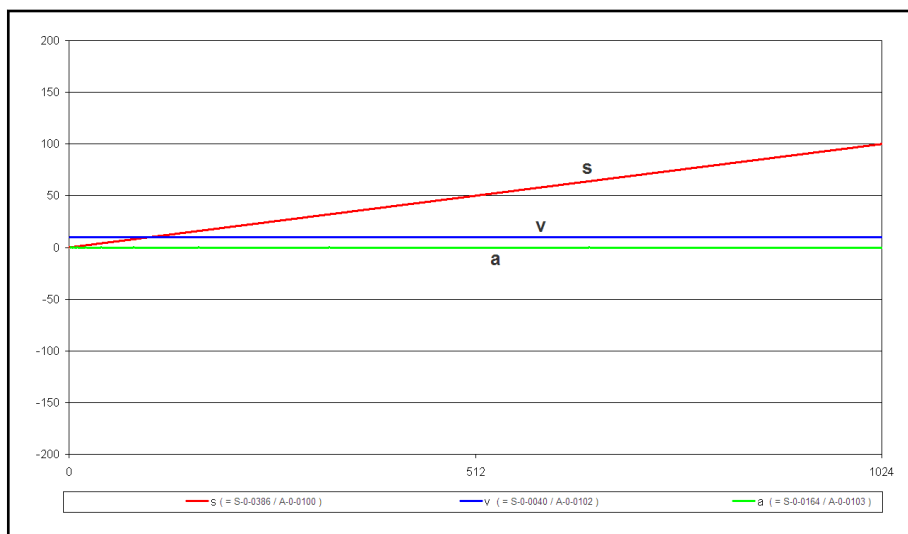
- Der Algorithmus zur Berechnung der Kurvenscheibenstützpunkte ist auf mehrere Zyklen, d.h. Aufrufe der FB-Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden

Bewegungsprofile mit verschiedenen Belegungen des Eingangs "Correction"



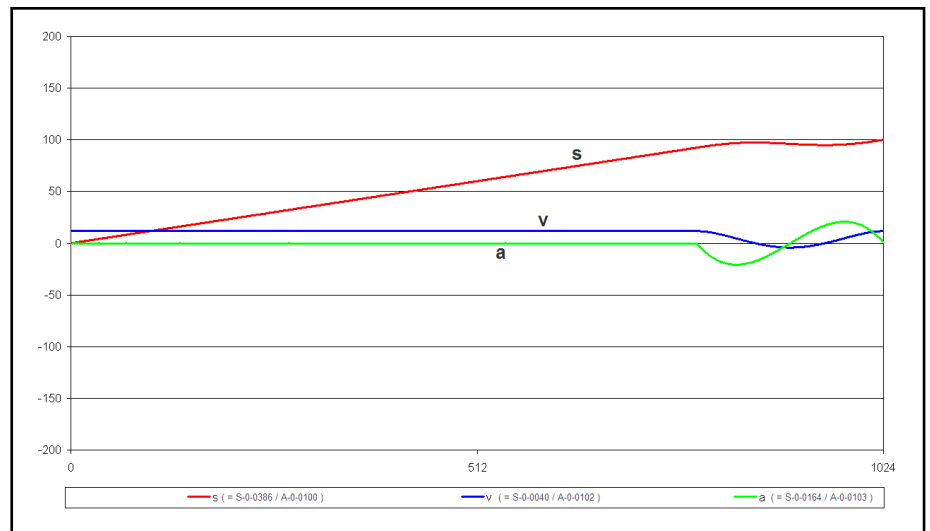
SyncLength: 1.3 * PrintLength; Correction: - 200

Abb. 4-91: MB_PrintLengthCorrCalcType01 Bewegungsprofil



SyncLength: 1.3 * PrintLength; Correction: 0

Abb. 4-92: MB_PrintLengthCorrCalcType01 Bewegungsprofil



SyncLength: 1.3 * PrintLength; Correction: 200

Abb. 4-93: MB_PrintLengthCorrCalcType01 Bewegungsprofil

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR	16#2200	16#0001	Eingang "NumberOfElements" ist < Minimum oder > Maximum
INPUT_RANGE_ERROR	16#2240	16#0001	Eingang "PrintFormat" ist < Minimum
INPUT_RANGE_ERROR	16#2240	16#0002	Eingang "SyncLength" ist < Minimum
INPUT_RANGE_ERROR	16#2240	16#0003	Eingang "Correction" ist < Minimum oder > Maximum
INPUT_INVALID_ERROR	16#2201	16#0001	Adresse von Feld "CamTable" ist null
INPUT_INVALID_ERROR	16#2241	16#0001	Ungültige Berechnung: (PrintFormat + Correction) > SyncLength oder SyncLength <= PrintFormat

Tab. 4-49: Fehlercodes Funktionsbaustein MB_PrintLengthCorrCalcType01

4.8 Zugkraftregler

4.8.1 Einführung und Übersicht

Die Zugkraftregler ([MB_TensionControlLoadCellType01](#), Seite 278, und [MB_TensionControlLoadCellMatParType01](#), Seite 292) regeln, abhängig vom Messwert einer Kraftmessdose, die Zugkraft auf den gewünschten Sollwert. Der Istwert muss dem Regler zur Verfügung gestellt werden, und wird in der Regel über einen Analog- oder Feldbus-Eingang erfasst.

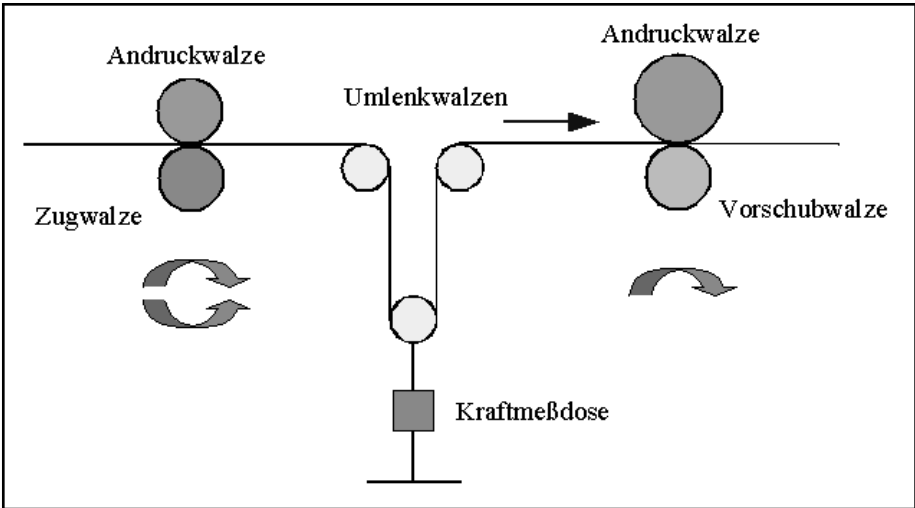


Abb. 4-94: *Prinzipdarstellung zur Zugkraftregelung*

Der Zugregler ist als PI-Regler ausgeführt. Der Reglerausgang wirkt auf den Getriebefeinabgleich der Folgeachse (Zugwalze).

Für die Regelung im Stillstand ist zusätzlich ein P-Regler integriert. Der Reglerausgang im Stillstand wirkt auf die additive Leitachsisposition der Folgeachse (Zugwalze).

Eine Parametrierung des dynamischen Regelkreisverhaltens ist beim Zugkraftregler "MB_TensionControlLoadCellType01" durch Vorgabe der Reglerparameter, und beim Zugkraftregler "MB_TensionControlLoadCellMatParType01" durch Vorgabe von Material- und Streckenparametern, möglich.

Die Folgeachse kann sowohl in geschwindigkeitssynchroner oder winkelsynchroner Betriebsart betrieben werden.

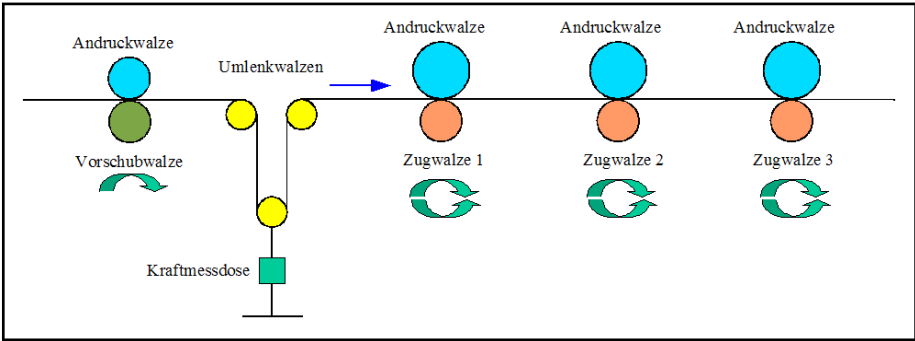


Abb. 4-95: *Multiachs-Zugkraftregler*

4.8.2 **MB_TensionControlLoadCellType01**

Kurzbeschreibung Der Zugkraftregler regelt abhängig vom Messwert einer Kraftmessdose die Zugkraft auf den gewünschten Sollwert.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechMotion
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_Technology_07.lib
MLD ab MPx20 mit MA Funktionspaket	MX_TechMotion

Tab. 4-50: *Referenztablelle MB_TensionControlLoadCellType01*



Ab MLC 12VRS werden auch Slave-Achsen mit Interpolation auf der Steuerung unterstützt.



Der Eingang "ResponseCurve" ist in der Bibliothek RIL_LoopControl deklariert. Daher muss die Bibliothek RIL_LoopControl zusätzlich im Bibliotheksverwalter aufgenommen werden, wenn der Eingang "ResponseCurve" genutzt wird.

Schnittstellenbeschreibung

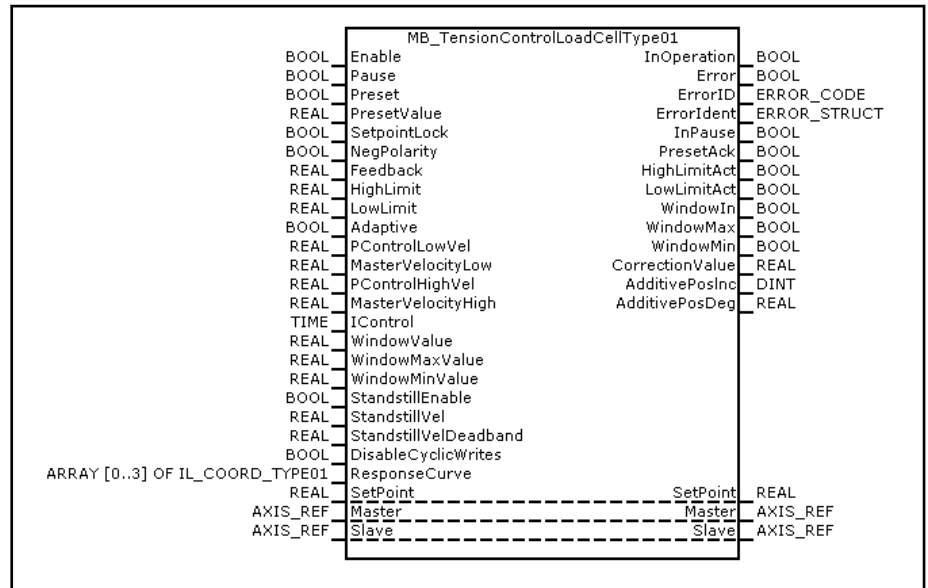


Abb. 4-96: Funktionsbaustein MB_TensionControlLoadCellType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Wird dieser Eingang gesetzt, arbeitet der Funktionsbaustein
	Pause	BOOL	Bei gesetztem Eingang wird der Zugkraftregler angehalten und der I-Anteil des Ausgangswertes eingefroren. Ausgang "InPause" signalisiert ein aktives "Pause"
	Preset	BOOL	Bei der steigenden Flanke dieses Eingangs wird der Reglerausgang auf den in PresetValue parametrierbaren Wert gesetzt und bei Erfolg der Ausgang "PresetAck" gesetzt
	PresetValue	REAL	Wert auf den der Reglerausgang gesetzt wird, wenn der Eingang "Preset" gesetzt ist
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben. Der Wert wird nur bei steigender Flanke von "Enable" ausgewertet
	SetpointLock	BOOL	Bei steigender Flanke am Eingang wird der aktuelle Zugkraftwert an "Feedback" als neuer Sollwert für den Zugkraftregler übernommen

E/A-Typ	Name	Datentyp	Beschreibung
	NegPolarity	BOOL	Bei gesetztem Eingang wird die Stellgröße "CorrectionValue" und "AdditivePosInc" mit einem negativen Vorzeichen versehen
	Feedback	REAL	aktueller Istwert der Zugkraft
	HighLimit	REAL	Maximalwert den der Reglerausgang annehmen kann. Wird die vom Regler berechnete Stellgröße größer als dieser Wert, wird die Stellgröße begrenzt und der I-Anteil des Reglers nicht weiter aufsummiert
	LowLimit	REAL	Minimalwert den der Reglerausgang annehmen kann. Wird die vom Regler berechnete Stellgröße kleiner als dieser Wert, wird die Stellgröße begrenzt und der I-Anteil des Reglers nicht weiter aufsummiert
	Adaptive	BOOL	Bei gesetztem Eingang wird eine adaptive P-Verstärkung für die Regelung verwendet
	PControlLowVel	REAL	P-Verstärkung des Reglers wenn "Adaptive" = FALSE Ansonsten entspricht der Wert der P-Verstärkung bei niedrigen Leitachsgeschwindigkeiten
	MasterVelocityLow	REAL	Nur wirksam wenn "Adaptive" = TRUE. Leitachsgeschwindigkeit bis zu der "PControlLowVel" verwendet wird
	PControlHighVel	REAL	P-Verstärkung des Reglers bei hohen Leitachsgeschwindigkeiten. Wenn "Adaptive" = FALSE, hat dieser Eingang keine Wirkung
	MasterVelocityHigh	REAL	Nur wirksam, wenn "Adaptive" = TRUE. Leitachsgeschwindigkeit ab der "PControlHighVel" verwendet wird
	IControl	TIME	Nachstellzeit des I-Reglers Wird der Wert auf 0 sec. gesetzt, wird der I-Regler unwirksam
	WindowValue	REAL	Prozessgrößenfenster
	WindowMaxValue	REAL	Maximalwert der Prozessgröße
	WindowMinValue	REAL	Minimalwert der Prozessgröße
	StandstillEnable	BOOL	Bei gesetztem Eingang ist die in "StandstillVel" angegebenen additive Geschwindigkeit wirksam
	StandstillVel	REAL	Additive Geschwindigkeit für die Regelung im Stillstand
	StandstillVelDeadband	REAL	Dieser Wert bestimmt den Wert der Regelabweichung unterhalb dessen die Spanngeschwindigkeit bei der Regelung im Stillstand reduziert wird
	ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	Kennlinie zur Gewichtung der Regelabweichung im Zweig "InVelocity Control" IL_COORD_TYPE01, RIL_LoopControl.library
VAR_OUTPUT	InOperation	BOOL	Der Zugkraftregler arbeitet und die Ausgänge sind gültig

E/A-Typ	Name	Datentyp	Beschreibung
	Error	BOOL	Signalisiert einen Fehler Mit "Enable" = FALSE wird der Fehler gelöscht
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung gemäß Fehlertabelle (siehe Fehlerbehandlung , Seite 290)
	InPause	BOOL	"Pause" ist aktiv
	PresetAck	BOOL	Ist dieser Ausgang aktiv, wurde der "Preset" Eingang ausgewertet und der Ausgang "CorrectionValue" hat den Wert "PresetValue" angenommen
	HighLimitActive	BOOL	Dieser Ausgang wird gesetzt, wenn die vom Regler berechnete Stellgröße größer als der in "HighLimit" angegebene Wert ist
	LowLimitActive	BOOL	Dieser Ausgang wird gesetzt, wenn die vom Regler berechnete Stellgröße kleiner als der in "LowLimit" angegebene Wert ist
	WindowIn	BOOL	Wenn TRUE, befindet sich die Prozessgröße innerhalb des parametrisierten Fensters
	WindowMax	BOOL	Wenn TRUE, befindet sich die Prozessgröße oberhalb des parametrisierten Maximalwertes
	WindowMin	BOOL	Wenn TRUE, befindet sich die Prozessgröße unterhalb des parametrisierten Minimalwertes
	CorrectionValue	REAL	Stellgröße des Zugkraftreglers
	AdditivePosInc	DINT	Additive Leitachsposition in Inkrementen je Sercos-Zyklus, genutzt für reale Slave-Achse mit Interpolation auf dem Antrieb (Zielparameter: P-0-0692) [-]
	AdditivePosDeg	REAL	Additive Leitachsposition in Grad je Sercos-Zyklus, genutzt für Slave-Achse mit Interpolation auf der Steuerung (z. B. für virtuelle Achsen oder reale Achse mit Interpolation auf der Steuerung, Zielparameter: A-0-2600) [°/ms]
VAR_IN_OUT	Setpoint	REAL	Sollwert der Zugkraft Diese wird in der Regel durch den Anwender gesetzt, kann aber auch vom Funktionsbaustein selbst geändert werden (siehe "SetpointLock")
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Referenz zur Leitachse
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Referenz zur Folgeachse

Tab. 4-51: Schnittstellenvariablen Funktionsbaustein MB_TensionControlLoad-CellType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
Pause	BOOL			FALSE	kontinuierlich
Preset	BOOL			FALSE	kontinuierlich
DisableCyclicWrites	BOOL			FALSE	steigende Flanke an "Enable"
SetPointLock	BOOL			FALSE	kontinuierlich
NegPolarity	BOOL			FALSE	kontinuierlich
Feedback	REAL	n.def.	n.def.	0.0	kontinuierlich
PresetValue	REAL	n.def.	n.def.	0.0	steigende Flanke an "Preset"
HighLimit	REAL	>LowLimit	n.def.	200.0	kontinuierlich
LowLimit	REAL	-95.0	<HighLimit	-50	kontinuierlich
Adaptive	BOOL			FALSE	kontinuierlich
PControlLowVel	REAL	>0.0	n.def.	1.0	kontinuierlich
MasterVelocity-Low	REAL	>0.0	<=MasterVelocity-High	1.0	kontinuierlich
PControlHighVel	REAL	>0.0	n.def.	1.0	kontinuierlich
MasterVelocity-High	REAL	>=MasterVelocity-Low	n.def.	1.0	kontinuierlich
IControl	TIME	0s	n.def.	0s	kontinuierlich
WindowValue	REAL	0.0	100.0	0.0	kontinuierlich
WindowMaxValue	REAL	>=WindowMinValue	100.0	0.0	kontinuierlich
WindowMinValue	REAL	0.0	<=WindowMaxValue	0.0	kontinuierlich
StandstillEnable	BOOL			FALSE	kontinuierlich
StandstillVel	REAL	0.0	n.def.	0.0	kontinuierlich
StandstillDeadband	REAL	0.0	n.def.	1.0	kontinuierlich
ResponseCurve	ARRAY [0..3] OF IL_COORD_TY-PE01	siehe RIL_Loop-Control	siehe RIL_Loop-Control	siehe Datenstruktur	steigende Flanke an "Enable"

Tab. 4-52: Eingangsverhalten MB_TensionControlLoadCellType01

Funktionsbeschreibung

Im Folgenden werden die einzelnen Funktionen der Eingänge des Funktionsbausteins näher erläutert. Allgemein gilt für alle Eingänge, dass sie kontinuierlich bei gesetztem "Enable"-Eingang ausgewertet werden. Das heißt, Änderungen der Eingänge werden sofort ohne erneutes Toggeln des "Enable"-Eingangs wirksam.

Ausnahme dieser Regel bildet der "PresetValue"-Eingang, der nur bei steigender Flanke am "Preset"-Eingang übernommen wird. Ebenso wird der Ein-

gang "DisableCyclicWrites" nur bei steigender Flanke am "Enable"-Eingang übernommen.

Die gesamte Wirkungsweise des Zugkraftreglers ist in den folgenden Flussdiagrammen dargestellt. Je nach Eingang "StandstillEnable" ist entweder der Pfad "InVelocity" oder "Regelung im Stillstand" (Standstill Control) wirksam.

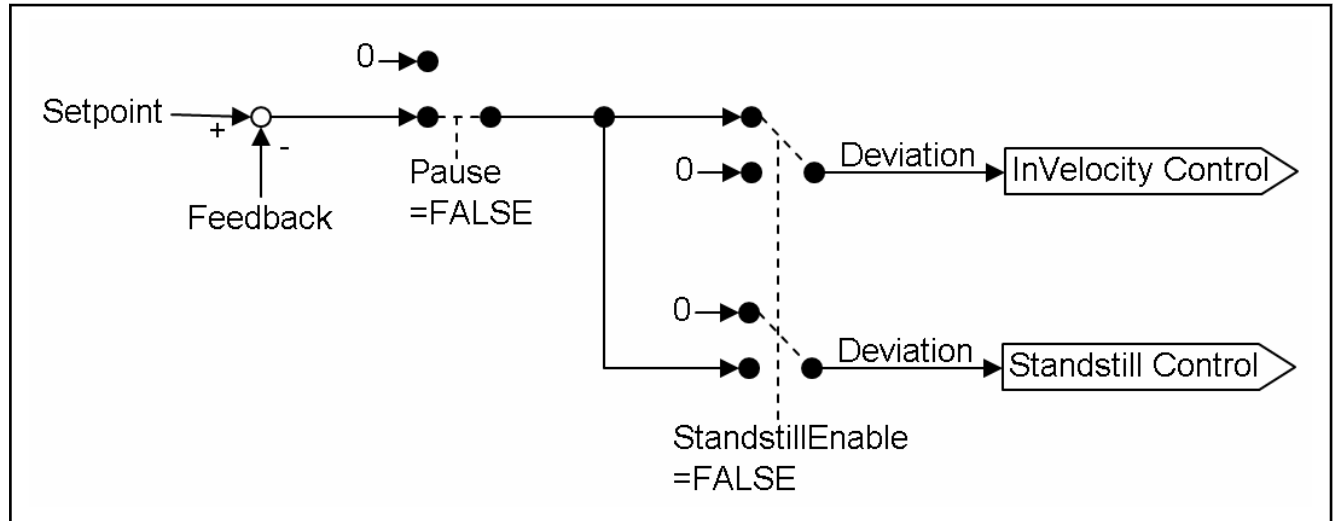


Abb. 4-97: Flussdiagramm des Zugkraftreglers mit Abzweig für Regelung im Stillstand / In Geschwindigkeit

Der Pfad "InVelocity Control" ist wirksam, wenn der Eingang "StandstillEnable" nicht gesetzt ist. Die Zugkraft wird hierbei über den Parameter "Getriebeineinabgleich" P-0-0694 (Slave-Achse mit Interpolation auf dem Antrieb/IPO Drive) bzw. A-0-2605 (Slave-Achse mit Interpolation auf der Steuerung/IPO Control) geregelt. Es wird direkt auf die Parameter geschrieben, wenn der Eingang "DisableCyclicWrites" nicht gesetzt ist. Bei realen Achsen mit Interpolation auf dem Antrieb ist daher auch die Konfiguration des Parameters P-0-0694 in den optionalen zyklischen Sercos-Daten (MDT) erforderlich.

Die Stellgröße wird zusätzlich auch über den Ausgang "CorrectionValue" ausgegeben.

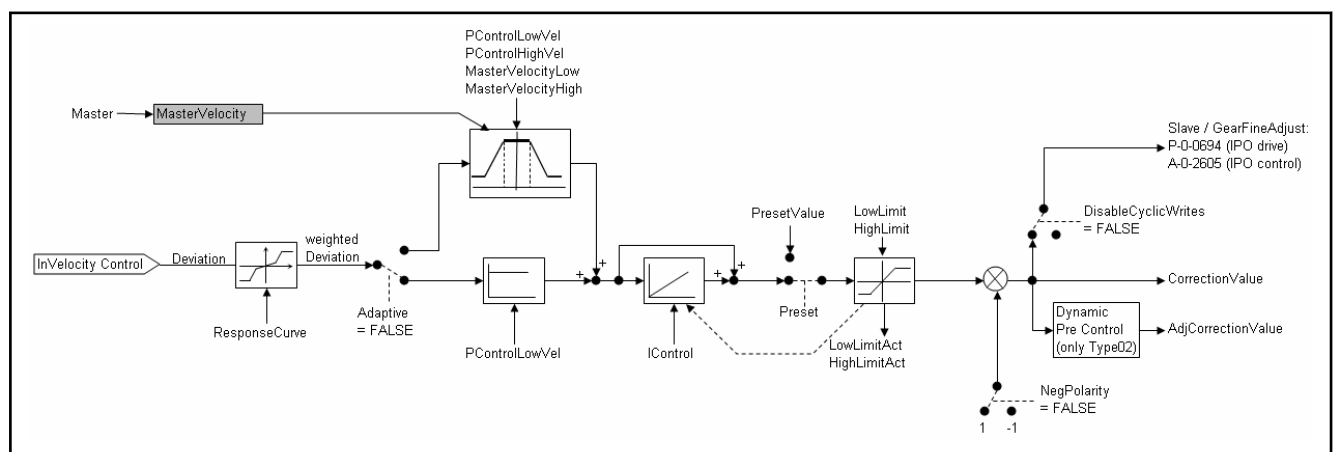


Abb. 4-98: Flussdiagramm des Zugkraftreglers für den Zweig "InVelocity Control"

Für die Regelung im Stillstand kann zusätzlich ein weiterer Pfad aktiv geschaltet werden, in dem auf die additive Leitachsisposition (P-0-0692) geschrieben wird sofern "DisableCyclicWrites" inaktiv ist. Gleichzeitig steht der Wert immer über den Ausgang "AdditivePosInc" zur Verfügung.

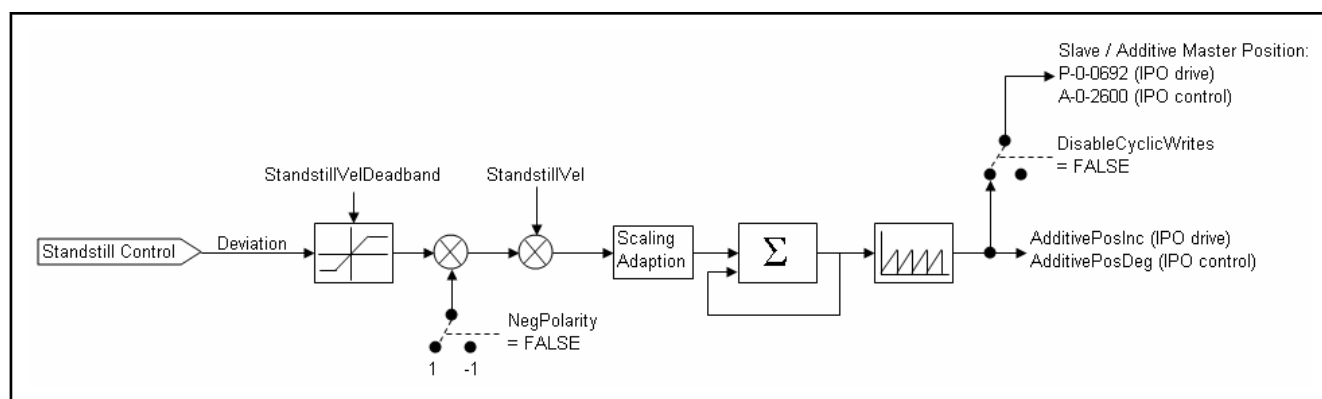


Abb. 4-99: Flussdiagramm des Zugkraftreglers für die Regelung im Stillstand (Standstill Control)

Das Flussdiagramm enthält aus Gründen der Übersichtlichkeit nicht die Begrenzungen "WindowValue", "WindowMaxValue", "WindowMinValue" sowie deren Statusanzeigen.

Inbetriebnahme

Ist "DisableCyclicWrites" inaktiv, müssen die beiden Parameter P-0-0694 (Getriebefeinabgleich) und P-0-0692 (additive Leitachspolition) im zyklischen Sollwertkanal der Folgeachse aufgelegt werden. Bei steigender Flanke an "Enable" werden zyklische Datenkanäle überprüft. Wird einer der beiden Parameter nicht gefunden, erfolgt eine entsprechende Fehlermeldung an den Fehlerausgängen des Funktionsbausteins. Ist "DisableCyclicWrites" aktiv, wird die Konfiguration der zyklischen Sollwertdatenkanäle nicht überprüft.

Die verwendeten Achsen müssen vor dem Aktivieren des "Enable"-Eingangs in eine synchrone Betriebsart gebracht werden. Dies kann durch die PLCopen-Funktionsbausteine "GearIn" oder "GearInPos" geschehen.

Wird der Zugkraftregler in der antriebsintegrierten Steuerung MLD verwendet, so muss die AxisData-Struktur im Antrieb aktiviert werden.

Preset/PresetValid

Durch eine positive Flanke am Eingang "Preset" wird ein Anfangswert über die Variable "PresetValue" für die Stellgröße vorgegeben. Gleichzeitig werden interne Variablen des Reglers so gesetzt, dass ein ruckfreies Anfahren des Reglers erfolgt. Der "Preset" wirkt nur auf den Getriebefeinabgleich im Pfad "InVelocity Control".

Der "Preset" Eingang wird nur ausgewertet, wenn "Enable" gesetzt ist. Bei steigender Flanke an "Preset" wird der neue Wert an "PresetValue" übernommen und "PresetAck" gesetzt. Eine fallende Flanke an "Preset" oder "Enable" löscht "PresetAck". Der "Preset"-Eingang ist dominant über "Pause", d.h. der Wert an "PresetValue" erscheint am Ausgang, auch wenn "Pause" gesetzt ist. Ist bei steigender Flanke an "Enable" "Preset" gesetzt, so wird "PresetValue" sofort als Stellgröße übernommen.



Die Preset Funktion sollte im Stillstand ausgeführt werden (StandstillEnable=TRUE), da ansonsten Geschwindigkeitssprünge durch das setzen des ControlValue auf den PresetValue entstehen können. Der Betrag der Geschwindigkeitssprünge ist proportional zu der Geschwindigkeit.

Der "Preset"-Eingang wirkt nur bei der steigenden Flanke.

DisableCyclicWrites

Der Eingang "DisableCyclicWrites" bestimmt, ob die vom Funktionsbaustein berechneten Werte direkt über den zyklischen Kanal geschrieben werden (FALSE) oder nur am Bausteinausgang verfügbar sind (TRUE).

Adaptive, PControlLowVel, PControlHighVel, MasterVelocityLow, MasterVelocityHigh

Damit ist es möglich mehrere Bausteine zu kaskadieren, indem man "DisableCyclicWrites" auf TRUE setzt. Dann kann man lediglich die Ausgänge ("CorrectionValue" und "AdditivePosInc") verwenden und weiter modifizieren. Allerdings ist es dann noch notwendig, dass der Anwender die Werte selbst in den optional zyklischen Kanal schreibt.

Ist "DisableCyclicWrites" auf TRUE gesetzt, wird außerdem die Prüfung, ob die optional zyklischen Parameter auch konfiguriert sind, nicht durchgeführt.

Die P-Verstärkung kann wahlweise als konstante P-Verstärkung oder als adaptive P-Verstärkung vorgegeben werden. Über den Eingang "Adaptive" wird zwischen adaptiver oder konstanter P-Verstärkung gewählt. Durch Setzen von "Adaptive" auf TRUE wird die adaptive P-Verstärkung ausgewählt.

Ist eine konstante P-Verstärkung gewählt, ist nur der Wert in "PControlLowVel" für die P-Verstärkung wirksam.

Wird die adaptive P-Verstärkung gewählt, müssen die Eingänge "PControlLowVel", "PControlHighVel", "MasterVelocityLow" und "MasterVelocityHigh" beschaltet werden. Die Kennlinie sieht dann folgendermaßen aus:

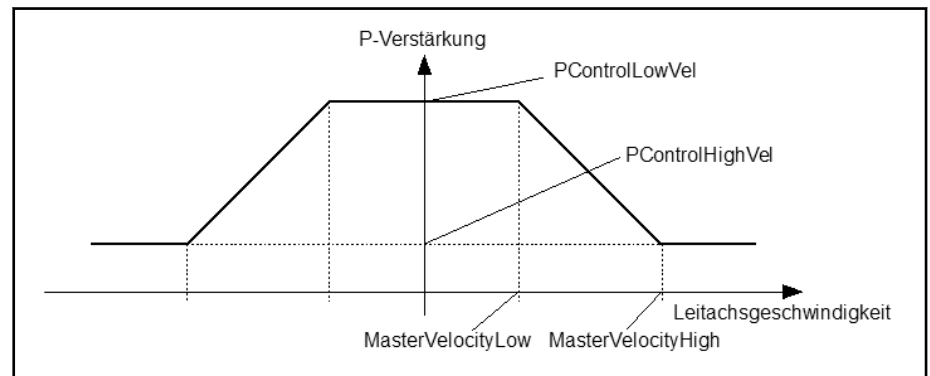


Abb. 4-100: Kennlinie der adaptiven P-Verstärkung

Die einzelnen Eingänge beschreiben die Kennlinie für positive Leitachs-Geschwindigkeiten. Die Kennlinie für negative Leitachs-geschwindigkeiten ergibt sich aus der Spiegelung an der Achse der P-Verstärkung. Das heißt, es sind nur symmetrische Verläufe möglich. Die P-Verstärkung zwischen den Leitachsgeschwindigkeiten "MasterVelocityLow" und "MasterVelocityHigh" wird linear interpoliert. Dadurch ergeben sich einige Bedingungen für die Wertebereiche der Eingaben:

- Die Werte für "MasterVelocityLow" und "MasterVelocityHigh" müssen positiv sein
- Die Werte für "MasterVelocityLow" und "MasterVelocityHigh" dürfen nicht gleich sein
- Der Wert von "MasterVelocityLow" muss kleiner als der Wert von "MasterVelocityHigh" sein

Ist eine dieser Bedingungen nicht erfüllt, wird ein Fehler am Ausgang des Bausteines generiert.

SetpointLock

Durch Setzen des Einganges "SetpointLock" wird der aktuelle Zugkraftistwert als neuer Sollwert für den Zugkraftregler übernommen. Die Funktion erlaubt es, z.B. für den Sollwert der Zugkraft einen sinnvollen Startwert zu erhalten, der den momentanen Zustand der Maschine widerspiegelt.

Pause

Der Eingang "Pause" ermöglicht das Anhalten des Zugkraftreglers. Wird der Eingang auf TRUE gesetzt, wird die Regelabweichung auf "0" gesetzt, d.h. der Reglerausgang wird auf dem aktuellen I-Anteil eingefroren. Ist "Pause" aktiv, wird der Ausgang "InPause" gesetzt.

NegPolarity	Der Eingang "NegPolarity" beeinflusst den Regelsinn. Abhängig von der Lage der Zugwalze, bezüglich der Durchlaufrichtung des Materials und der Lage der Vorschubwalze, muss der Regelsinn entweder positiv oder negativ sein. Ist dieser Eingang FALSE, wird der Regelsinn positiv und eine positive Regelabweichung führt zu einer positiven Stellbewegung. Der Eingang wirkt auch auf die Stillstandsregelung.
Feedback	Dieser Eingang stellt den Istwert der Zugkraft dar. Dieser kann z.B. durch einen Analogeingang am Antrieb oder den Inline I/O eingelesen werden. Der Anwender ist dabei für eine sinnvolle Skalierung verantwortlich.
PresetValue	Durch Setzen des Eingangs "Preset" wird ein Anfangswert über die Variable "PresetValue" für die Stellgröße vorgegeben. Gleichzeitig werden interne Variablen des Reglers so gesetzt, dass ein ruckfreies Anfahren des Reglers erfolgt.
HighLimit, HighLimitAct	Mit diesem Eingang wird die Stellgröße des Reglers auf einen Maximalwert begrenzt. Tritt die Begrenzung ein, wird der I-Anteil des Reglers nicht weiter aufsummiert ("Anti-reset-windup"). Der maximale Wert wird durch den im Antrieb definierten Maximalwert für den Getriebefeinabgleich bestimmt. Tritt eine Begrenzung ein, wird der Ausgang "HighLimitAct" gesetzt.
LowLimit, LowLimitAct	Mit diesem Eingang wird die Stellgröße des Reglers auf einen Minimalwert begrenzt. Tritt die Begrenzung ein, wird der I-Anteil des Reglers nicht weiter aufsummiert ("Anti-reset-windup"). Der minimale Wert liegt bei -95 unabhängig von ggf. kleineren Werten im Antrieb (siehe Getriebefeinabgleich, Seite 288). Tritt eine Begrenzung ein, wird der Ausgang "LowLimitAct" gesetzt.
PControlLowVel	Dieser Wert entspricht der Proportional-Verstärkung des Zugkraftreglers, sofern der Eingang "Adaptive" nicht aktiv ist. Ansonsten stellt er die untere Verstärkung einer adaptiven P-Verstärkung dar.
WindowValue, WindowIn	Mit diesem Wert kann der Anwender ein symmetrisches Fenster definieren innerhalb dessen sich der aktuelle Zugkraftistwert befinden soll. Das Fenster ist wie folgt definiert:

$$\left(SetPoint - |SetPoint| \cdot \frac{WindowValue}{100\%} \right) \leq Feedback \leq \left(SetPoint + |SetPoint| \cdot \frac{WindowValue}{100\%} \right)$$

Abb. 4-101: WindowValue, WindowIn

Ist diese Bedingung erfüllt, wird der Ausgang "WindowIn" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt, es findet aber auch keinerlei Begrenzung statt.

WindowMaxValue, WindowMax	Mit diesem Wert kann der Anwender einen oberen Wert für den Zugkraftistwert festlegen. Der Wert wird wie folgt ausgewertet:

$$Feedback > \left(SetPoint + |SetPoint| \cdot \frac{WindowMaxValue - 100}{100} \right)$$

Abb. 4-102: WindowMaxValue, WindowMax

Ist diese Bedingung erfüllt, wird der Ausgang "WindowMax" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt, es findet aber auch keinerlei Begrenzung statt.

WindowMinValue, WindowMin	Mit diesem Wert kann der Anwender einen unteren Wert für den Zugkraftistwert festlegen. Der Wert wird wie folgt ausgewertet:

$$Feedback < \left(SetPoint + |SetPoint| \cdot \frac{WindowMinValue - 100}{100} \right)$$

Abb. 4-103: WindowMinValue, WindowMin

Ist diese Bedingung erfüllt, wird der Ausgang "WindowMin" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt, es findet aber auch keinerlei Begrenzung statt.

StandstillEnable

Durch Setzen des Eingangs "StandstillEnable" wird die in "StandstillVel" angegebene Geschwindigkeit additiv auf die zu regelnde Achse aufgeschaltet. Damit ist eine Zugkraftregelung auch im Stillstand möglich.



Der Getriebefeinabgleich verliert bei niedrigen Geschwindigkeiten immer mehr an Einfluss. Bei Stillstand der Leitachse besteht sogar ein offener Regelkreis, da der Getriebefeinabgleich dann keinen Einfluss mehr hat. Daher muss der Eingang "StandstillEnable" bei Erreichen des Stillstandsfensters gesetzt werden, damit im Pfad "InVelocity Control" der I-Anteil durch den offenen Regelkreis nicht bis zur Grenze ansteigt.



Bei Achsen mit Interpolation auf der Steuerung wird der Ausgang "AdditivePosInc" im Moduloformat 2²⁰ ausgegeben (ab Version 12).



Wird dieser Eingang verwendet, muss sichergestellt sein, dass der Zugkraftregler nur in der Sercos-synchronen Task verwendet wird. Ist dies nicht der Fall, kann es zu unruhigem Lauf des Motors oder Oszillieren um die Stillstandsfläche führen.

StandstillVel

Dieser Wert gibt die additive Geschwindigkeit an, die beim Setzen des Eingangs "StandstillEnable" an den Antrieb gesendet wird. Die Einheiten der Geschwindigkeit entsprechen dabei der im Antrieb eingestellten Wichtung.



Im Stillstand wird die Geschwindigkeit sprunghaft erzeugt. Da die Geschwindigkeit innerhalb eines Sercos-Zyklus erreicht werden muss, können keine beliebig hohen Werte vorgegeben werden. Wird der Wert der Stillstandsgeschwindigkeit zu hoch gewählt, kann dies zur Oszillation des Antriebes führen. Der Maximalwert richtet sich dabei nach der Wichtung der Geschwindigkeit, sowie der eingestellten maximalen Beschleunigung und angeschlossenen Masse am Antrieb. Um zu große Geschwindigkeitssprünge zu vermeiden, kann eine Filterzeitkonstante im Antriebsparameter P-0-0693 definiert werden, wodurch der Sprung geglättet wird.

ResponseCurve

Mit Hilfe des Eingang "ResponseCurve" kann die Regelabweichung gewichtet werden. Durch diese Funktionalität kann beispielsweise bei kleinen Regelabweichungen eine geringere Gewichtung der Regelabweichung eingestellt werden. Dies ergibt quasi eine geringere Proportionalverstärkung und dadurch ein ruhigeres Regelverhalten im eingeschwungenen Zustand. Bei der Parametrierung des Eingangs sollten die Reglerparameter zunächst ohne Bewertung eingestellt werden und erst danach eine geringere Gewichtung vorgegeben werden. Hierdurch wird erreicht, dass die Reglerparameter zunächst global anhand der Systemdynamik eingestellt werden und anschließend eine lokale Optimierung (z. B. für den eingeschwungenen Zustand) durchgeführt wird.

Parametriert wird diese lokale Optimierung mittels einer Kennlinie mit bis zu 4 Stützstellen. Negative Abweichungen werden durch Punktspiegelung der Kurve am Ursprung behandelt. Daraus ergeben sich insgesamt neun lineare Sektionen. Die Sektion s5 verläuft durch den Ursprung. Die Sektionen s1 und s9 haben einen Anstieg von 1 und laufen jeweils ins positive oder negative Unendliche. Wird der Eingang unbeschaltet gelassen, wird als Default-Wert eine 1-zu-1 Kennlinie verwendet, d. h. es wird keinerlei Gewichtung durchgeführt.

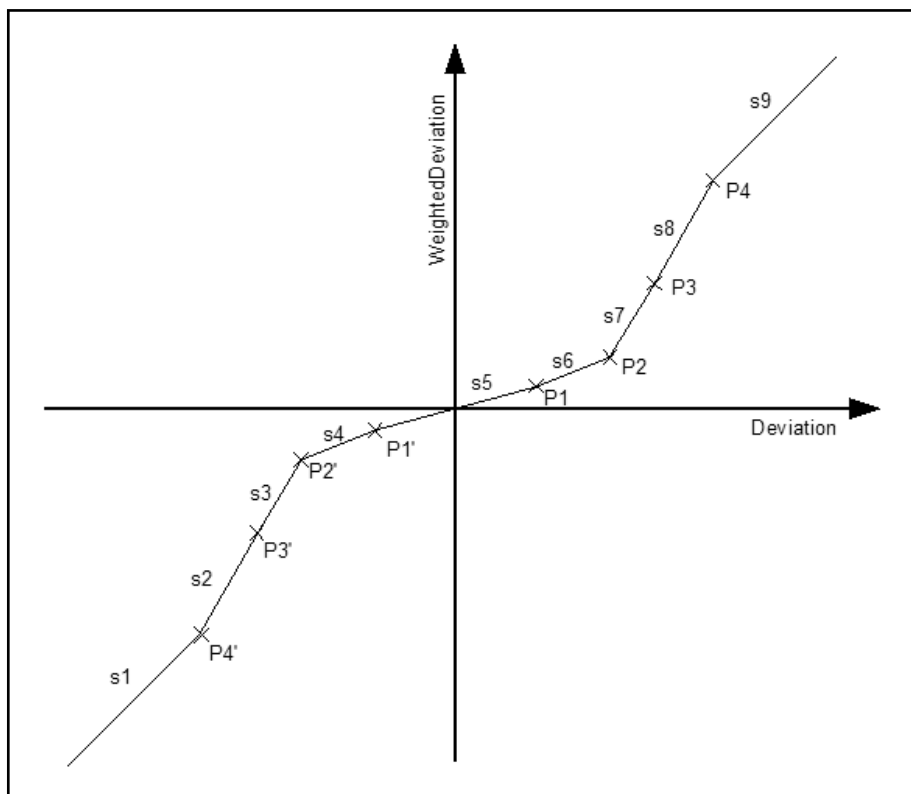


Abb. 4-104: Wichtung der Regelabweichung mit dem Eingang ResponseCurve

Die Punkte P1 bis P4 werden in einem Array aus IL_COORD_TYPE01 abgelegt. Werden weniger als vier Punkte benötigt, muss die X-Koordinate der nicht benötigten Punkte (vom letzten Punkt beginnend) auf 0 gesetzt werden. Die X-Werte müssen positiv sein und in positiver Richtung aufsteigen. Es ist erlaubt, denselben Punkt zweimal abzulegen.

Hintergrundinformationen

Getriebefeinabgleich

Der Zugkraftregler arbeitet in der Regel mit Geschwindigkeiten größer 0. Dabei ist der Getriebefeinabgleich ein probates Mittel, um die Zugkraft aufrechtzuerhalten. Allerdings gibt es dabei auch einige Randbedingungen, die hier erläutert werden sollen.

Der Getriebefeinabgleich wird intern auf ein Minimum von -95% begrenzt. Wird ein kleinerer Wert eingegeben, erzeugt der Baustein einen Fehler.

Hintergrund dieser Begrenzung ist der errechnete Ausgangswert, der bei gegebenem Eingangswert wie folgt definiert ist:

$$\text{Ausgangswert} = \text{Eingangswert} * \left(1 + \frac{\text{Getriebefeinabgleich}}{100} \right)$$

StandstillEnable, StandstillVel, StandstillVelDeadband

Anhand der Gleichung kann man erkennen, dass - wenn der Getriebefeinabgleich gegen -100% geht - sich der Ausgangswert gegen 0 bewegt. Damit würde der Regelkreis praktisch geöffnet werden, da eine Änderung des Eingangswertes keinerlei Reaktion am Ausgang bewirkt. Schlimmer noch, bei einem Getriebefeinabgleich der kleiner als -100% wäre, kehrt sich das Vorzeichen des Ausgangswertes um. Im Regelkreis würde eine Mitkopplung entstehen, die das System instabil machen kann.

Die Regelung im Stillstand kann durch Verwendung der Eingänge "StandstillEnable" und "StandstillVel" erfolgen. Dabei wird eine additive Geschwindigkeit aufgeschaltet. Um eine bessere Regelgüte zu erreichen, wird diese um den Sollwert herum verringert. Dies geschieht mit Hilfe einer abgewandelten Signum-Funktion, die nachfolgende Übertragungsfunktion besitzt:

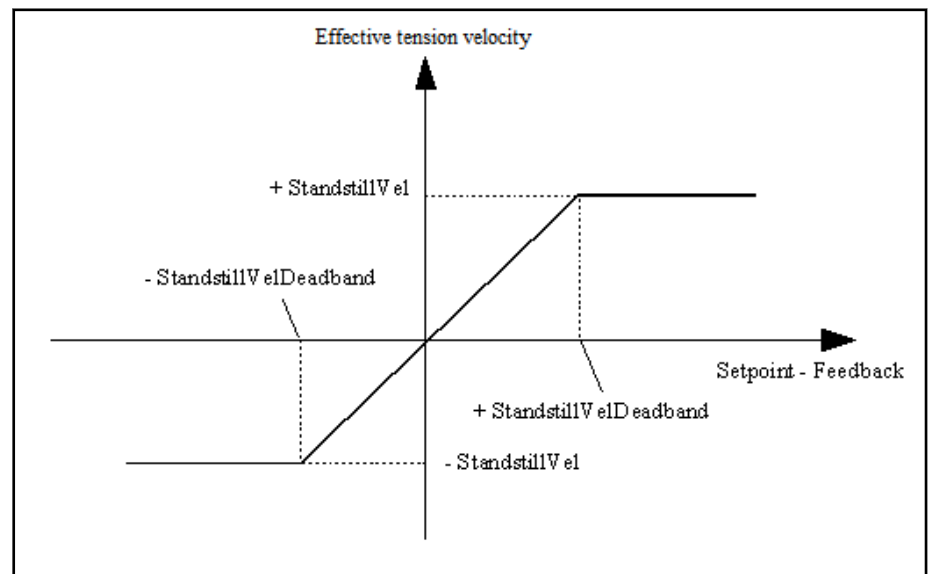


Abb. 4-105: Abgewandelte Signum-Funktion für die Begrenzung der aufgeschalteten StandstillVel

Die Regelabweichung wird dadurch im Bereich des "Setpoint" relativiert. Nähert sich der "Feedback" dem "Setpoint", wird die Regelung "feinfühlig" als ein reiner Zweipunktregler. Der Wert bei dem diese Begrenzung stattfindet, kann man am Eingang "StandstillVelDeadband" einstellen. Die Werte besitzen dabei dieselben Einheiten, wie der "Setpoint"-, bzw. "Feedback"- Eingang. Zwischen den Extremwerten "+StandstillVel" und "-StandstillVel" wird linear interpoliert, so dass sich obige Übertragungsfunktion für Regelabweichung und Spanngeschwindigkeit ergibt.

Anschließend wird der ermittelte Faktor mit der vorgegebenen "StandstillVel" multipliziert. Die Geschwindigkeit wird dabei als Drehzahl, mit der im Antrieb eingestellten Wichtung, verarbeitet. Möchte man anstatt der Drehzahl eine Bahngeschwindigkeit eingeben, ist diese vorher selbst zu berechnen.

Um eine umgekehrte Bewegungsrichtung für die Regelung im Stillstand zu erreichen, ist der Eingang "NegPolarity" zu setzen.

Nachfolgend wird noch mit einem Korrekturfaktor multipliziert. Dieser Korrekturfaktor setzt sich aus dem Getriebeübersetzungsverhältnis zusammen. Jedoch ist der Getriebefeinabgleich nicht berücksichtigt. In der Praxis ist der Getriebefeinabgleich in der Regel sehr klein, so dass dieser nicht ins Gewicht fällt. Das Getriebeübersetzungsverhältnis wird nur bei einer steigenden Flanke des "Enable"-Eingangs übernommen.

Im Flussdiagramm ist ersichtlich, dass die angegebene Geschwindigkeit in Inkremente umgerechnet wird. Dabei sind einige Regeln zu beachten:

- Die angegebene Geschwindigkeit kann nicht beliebig hoch gewählt werden, denn der Antrieb muss die Geschwindigkeit innerhalb eines Sercos-Zyklus erreichen können. Damit sind nur sehr kleine Werte zulässig. Die Größe hängt dabei von der verwendeten Wichtung und der Masse am Motor ab. Die errechneten Inkremente werden über den Parameter P-0-0692 vorgegeben. Dieser Parameter kann zudem über den im Antrieb verfügbaren Filter (P-0-0693) geglättet werden, so dass auch größere Sprünge möglich sind.
- Die errechneten Inkremente werden noch vor dem Getriebe aufaddiert. Das Getriebeübersetzungsverhältnis (A-0-2720 und A-0-2721 bzw. P-0-0156 und P-0-0157) wird dabei mit eingerechnet. Der Getriebefeinabgleich (P-0-0083 und P-0-0694) wird jedoch nicht berücksichtigt, da dieser in der Regel sehr klein ist. Das kann dazu führen, dass die Achse sich mit einer anderen, als der eingestellten Geschwindigkeit bewegt.
- Es wird eine Skalierung der Einheiten durchgeführt. Außerdem wird der eingegebene Geschwindigkeitswert nicht nach Maximum- und Minimumwerten überprüft. Dadurch können sich sehr große Werte für die Anzahl der Inkremente je Sercos-Zyklus ergeben. Überschreitet der errechnete Zwischenwert die Maximalwerte von $\pm$ halber Modulowert, werden diese Werte automatisch begrenzt. Die Folge ist eine niedrigere additive Geschwindigkeit, als am Eingang "StandstillVel" vorgegeben wird.
- Bei sehr kleinen Geschwindigkeiten ist außerdem zu beachten, dass die aktuelle Geschwindigkeit von der angegebenen Geschwindigkeit abweichen kann. Dies ist durch die Granulierung der Inkremente je Sercos-Zyklus zu erklären: Entspricht z. B. 1 Inkrement einer Geschwindigkeit von 0,03 U/min, so können nur Geschwindigkeiten die ein Vielfaches von 0,03 U/min sind, eingestellt werden. Ist dies nicht der Fall, wird zum nächstgelegenen Wert gerundet.
- Die errechneten Inkremente werden mit jedem Sercos-Zyklus verarbeitet. Deshalb muss bei Verwendung der Regelung im Stillstand der Baustein in der Sercos-synchronen Task abgearbeitet werden. Ist dies nicht der Fall, wird sich der Antrieb einen Sercos-Zyklus um die errechnete Anzahl an Inkrementen bewegen und in den folgenden Sercos-Zyklen wieder still stehen, was eine ruckende Bewegung zur Folge hat. Der Einsatz des Filters in P-0-0693 kann auch hier Abhilfe schaffen.

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle **F_RELATED_TABLE**, 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0004)	16#00000A00	16#00000001	Der Parameter P-0-0694 ist nicht im optional zyklischen Telegramm der Folgeachse parametrierbar
ACCESS_ERROR (16#0004)	16#00000A00	16#00000002	Der Parameter P-0-0692 ist nicht im optional zyklischen Telegramm der Folgeachse parametrierbar
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000001	Die Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") ist < 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000002	Die Leitachsgeschwindigkeit im oberen Betriebspunkt ("MasterVelocityHigh") ist < 0

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000003	Die Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") und oberen Betriebspunkt ("MasterVelocityHigh") besitzen die gleichen Werte
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000004	Der Wert der Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") ist größer als der Wert im oberen Betriebspunkt ("MasterVelocityHigh")
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000005	Der Wert des Eingangs "LowLimit" ist kleiner als der minimal zulässige (-95.0)
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000006	Der Wert der Eingänge "PGainLowVel" oder "PGainHighVel" ist kleiner oder gleich 0
DEVICE_ERROR (16#0008)	16#00000A02	16#00000007	Bei dem in "Slave" angegebenem Antrieb ist keine Leistung zugeschaltet. Eine Zugkraftregelung ist daher nicht möglich
DEVICE_ERROR (16#0008)	16#00000A02	16#00000008	Der in "Slave" angegebene Antrieb befindet sich nicht in einer synchronen Betriebsart
RESOURCE_ERROR (16#0003)	16#00001503	16#00000000	AxisData ist in der Steuerung nicht aktiv. Dieser Fehler tritt nur bei der MLD auf

Tab. 4-53: Fehlercodes Funktionsbaustein MB_TensionControlLoadCellType01

Besonderheiten für MLD

Bei Verwendung der Steuerung MLD-S mit übergeordneter Steuerung MLC muss für die Zugspannungsreglerachse Interpolation im Antrieb aktiviert sein (**reale Achsen ► entspr. Achse ► Eigenschaften**). Bei Verwendung der Steuerung MLD-M gibt es diese Einstellung nicht.

Im zyklischen Kanal zwischen Steuerung und Antrieb des entsprechenden Antriebs müssen die Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" und "P-0-0692 Leitachspannung additiv, Prozessregler" eingetragen sein ("UserCmdDataX").

Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus (der Dialog befindet sich im Projektbaum unterhalb des MLD-Knotens **AxisData**

).

Die "AxisData-Struktur" des Antriebes muss parametrierbar sein ("P-0-1367 SPS Konfiguration, Bit 6 ist gesetzt", bzw. Dialog SPS-Konfiguration).

Achsreferenz für die Masterachse:

Für die Technologiebausteine wird als Achsreferenz für die Masterachse empfohlen:

- Für eine virtuelle Leitachse die Achsreferenz **"VmAxis1"** verwenden.
Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 20 (VMA_1) eingestellt werden.
- Für eine reale Leitachse die Achsreferenz **"RmAxis1"** verwenden.
Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 23 (RMA_1) eingestellt werden.

Zusätzlich muss in diesem Fall noch die logische Adresse des Messgebers im Parameter "P-0-1820.0.2 CCD: Quelle des Messgeber-Lageistwertes" eingetragen werden.

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Wird dieser Eingang gesetzt, arbeitet der Funktionsbaustein
	Pause	BOOL	Bei gesetztem Eingang wird der Zugkraftregler angehalten und der Ausgangswert eingefroren. Der Ausgang "InPause" signalisiert ein aktives "Pause"
	Feedback	REAL	Aktueller Istwert der Zugkraft [N]*
	SetpointLock	BOOL	Bei einer steigenden Flanke an diesem Eingang wird der aktuelle Zugkraftistwert "Feedback" als neuer Sollwert "Setpoint" für den Zugkraftregler übernommen
	Preset	BOOL	Bei einer steigenden Flanke an diesem Eingang wird der Reglerausgang auf den in "PresetValue" parametrierbaren Wert gesetzt und bei Erfolg der Ausgang "PresetAck" gesetzt
	PresetValue	REAL	Wert auf den der Reglerausgang "CorrectionValue" gesetzt wird, wenn der Eingang "Preset" gesetzt wird [%]
	NegPolarity	BOOL	Bei gesetztem Eingang wird der Regelsinn invertiert
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben.
	HighLimit	REAL	Maximalwert den der Reglerausgang annehmen kann [%]. Wird die vom Regler berechnete Stellgröße größer als dieser Wert, wird die Stellgröße begrenzt und der I-Anteil des Reglers nicht weiter aufsummiert
	LowLimit	REAL	Minimalwert den der Reglerausgang annehmen kann [%]. Wird die vom Regler berechnete Stellgröße kleiner als dieser Wert, wird die Stellgröße begrenzt und der I-Anteil des Reglers nicht weiter aufsummiert
	WindowValue	REAL	Prozessgrößenfenster (relativ) [%]
	WindowMaxValue	REAL	Maximalwert der Prozessgröße (absolut) [%]
	WindowMinValue	REAL	Minimalwert der Prozessgröße (absolut) [%]
	Circumference	REAL	Achsumfang der Masterachse [mm]*
	DistanceRoll1To2	REAL	Bahnlänge zwischen Masterachse und Slaveachse [mm]*
	DelayFeedbackValue	TIME	Verzugszeit T_u der Bahnzugkraftregelstrecke. Summe aller peripherer Zeitkonstanten. U.a. Motordynamik, Filterzeiten, Messaufnehmer, Motionzykluszeit.
	StandstillEnable	BOOL	StandstillEnable = FALSE: Zweig "InVelocity Control" ist aktiv, Stellgröße: Getriebefeinabgleich StandstillEnable = TRUE: Zweig "Regelung im Stillstand" ist aktiv, Stellgröße: Leitachssposition additiv

E/A-Typ	Name	Datentyp	Beschreibung
	StandstillVelDeadband	REAL	Maximal wirksame Regelabweichung in der Stillstandsregelung [N]*. Entsprechend dieser Regelabweichung wird die Spangsgeschwindigkeit begrenzt
	ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	Kennlinie zur Gewichtung der Regelabweichung im Zweig "InVelocity Control"
	MasterVelocityLow	REAL	Leitachsgeschwindigkeit unterhalb derer die Regleradaptation begrenzt wird [1/min]
	MaterialParameter	MB_TENSION_CONTROL_MATERIAL	Struktur mit materialspezifischen Daten (Typ, Breite, Dicke, Grammatur und E-Modul) MB_TENSION_CONTROL_MATERIAL, Seite 305
	ControlMode	MB_TENSION_CONTROL_MODE	Auswahl des Regelkreisverhaltens. Optimiert auf Führungs- oder Störverhalten MB_TENSION_CONTROL_MODE, Seite 306
	DampingFactor	REAL	Dieser Eingang wirkt auf das Regelkreisverhalten [%]. Es kann stufenlos zwischen einem langsamen aperiodischen und einem schnellen periodischen Verhalten gewählt werden.
VAR_OUTPUT	InOperation	BOOL	Der Zugkraftregler arbeitet und die Ausgänge sind gültig
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung gemäß Fehlertabelle (siehe Fehlerbehandlung, Seite 304)
	InPause	BOOL	"Pause" ist aktiv
	PresetAck	BOOL	Ist dieser Ausgang aktiv, wurde der "Preset" Eingang ausgewertet und der Ausgang "CorrectionValue" hat den Wert "PresetValue" angenommen
	HighLimitActive	BOOL	Dieser Ausgang wird gesetzt, wenn die vom Regler berechnete Stellgröße größer als der in "HighLimit" angegebene Wert ist
	LowLimitActive	BOOL	Dieser Ausgang wird gesetzt, wenn die vom Regler berechnete Stellgröße kleiner als der in "LowLimit" angegebene Wert ist
	WindowIn	BOOL	Wenn TRUE, befindet sich die Prozessgröße innerhalb des parametrisierten Fensters
	WindowMax	BOOL	Wenn TRUE, befindet sich die Prozessgröße oberhalb des parametrisierten Maximalwertes
	WindowMin	BOOL	Wenn TRUE, befindet sich die Prozessgröße unterhalb des parametrisierten Minimalwertes
	CorrectionValue	REAL	Stellgröße des Zugkraftreglers [%]
	AdditivePosInc	DINT	Additive Leitachsposition in Inkrementen, genutzt für reale Slave-Achse mit Interpolation auf dem Antrieb (Zielparameter: P-0-0692) [-]

E/A-Typ	Name	Datentyp	Beschreibung
	AdditivePosDeg	REAL	Additive Leitachsposition in Grad, genutzt für Slave-Achse mit Interpolation auf der Steuerung (z.B. für virtuelle Achsen oder reale Achse mit Interpolation auf der Steuerung, Zielparameter: A-0-2600) [°]
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Referenz zur Leitachse
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Referenz zur Folgeachse
VAR_IN_OUT	Setpoint	REAL	Sollwert der Zugkraft [N]*. Diese wird in der Regel durch den Anwender gesetzt, kann aber auch vom Funktionsbaustein selbst geändert werden (siehe "SetpointLock")

*

Die Einheiten [N] und [mm] sind bei Verwendung der internen Materialdatenbank (E [N/mm²]) obligatorisch. Bei der Verwendung von benutzerdefinierten Elastizitätsmoduln können beliebige Einheiten wie z.B. [kg] und/oder [ft] verwendet werden. Es ist darauf zu achten, dass die gewählten Einheiten einheitlich verwendet werden.

Tab. 4-55:

Schnittstellenvariablen Funktionsbaustein MB_TensionControlLoad-CellMatParType01

Min-/Max- und Default-Werte der Eingänge

In folgender Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL	FALSE	TRUE	FALSE	kontinuierlich
Pause	BOOL	FALSE	TRUE	FALSE	kontinuierlich
Feedback	REAL	n.def.	n.def.	0.0	kontinuierlich
SetpointLock	BOOL	FALSE	TRUE	FALSE	kontinuierlich
Preset	BOOL	FALSE	TRUE	FALSE	kontinuierlich
PresetValue	REAL	-95.0	n.def.	0.0	steigende Flanke an "Preset"
NegPolarity	BOOL	FALSE	TRUE	FALSE	kontinuierlich
DisableCyclicWrites	BOOL	FALSE	TRUE	FALSE	steigende Flanke an "Enable"
HighLimit	REAL	>LowLimit	n.def.	200.0	kontinuierlich
LowLimit	REAL	-95.0	<HighLimit	-50	kontinuierlich
WindowValue	REAL	n.def.	n.def.	0.0	kontinuierlich
WindowMaxValue	REAL	>=WindowMinValue	n.def.	0.0	kontinuierlich
WindowMinValue	REAL	n.def.	<=WindowMaxValue	0.0	kontinuierlich
Circumference	REAL	>0.0	n.def.	100.0	steigende Flanke an "Enable"

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
DistanceRoll1To2	REAL	>0.0	n.def.	1000.0	steigende Flanke an "Enable"
DelayFeedback-Value	TIME	n.def.	n.def.	T#100ms	kontinuierlich
StandstillEnable	BOOL	FALSE	TRUE	FALSE	kontinuierlich
StandstillVel-Deadband	REAL	>0.0	n.def.	1.0	kontinuierlich
ResponseCurve	ARRAY [0..3] OF IL_COORD_TY-PE01	siehe RIL_Loop-Control	siehe RIL_Loop-Control	siehe RIL_Loop-Control	steigende Flanke an "Enable"
MasterVelocity-Low	REAL	>0.0	n.def.	1.0	kontinuierlich
MaterialParameter	MB_TENSION_CONTROL_MATERIAL	siehe Datenstruktur	siehe Datenstruktur	siehe Datenstruktur	kontinuierlich
ControlMode	MB_TENSION_CONTROL_MODE	siehe Datenstruktur	siehe Datenstruktur	siehe Datenstruktur	kontinuierlich
DampingFactor	REAL	20.0	200.0	100.0	kontinuierlich
Master	AXIS_REF	siehe RIL_CommonTypes	siehe RIL_CommonTypes	siehe RIL_CommonTypes	steigende Flanke an "Enable"
Slave	AXIS_REF	siehe RIL_CommonTypes	siehe RIL_CommonTypes	siehe RIL_CommonTypes	steigende Flanke an "Enable"
Setpoint	REAL	n.def.	n.def.	0.0	kontinuierlich

Tab. 4-56: Eingangsverhalten MB_TensionControlLoadCellMatParType01

Funktionsbeschreibung

Die Wirkungsweise des Zugkraftreglers ist in den folgenden Flussdiagrammen dargestellt. Je nach Eingang "StandstillEnable" ist entweder der Pfad "InVelocity Control" oder "Standstill Control" wirksam.

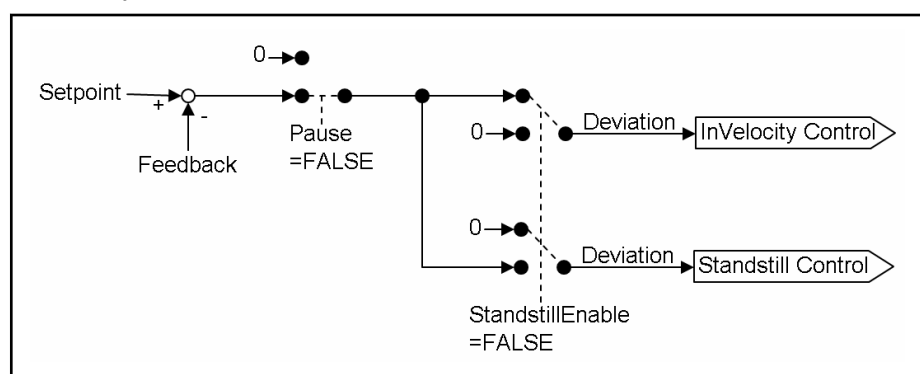


Abb. 4-107: Flussdiagramm des Zugkraftreglers mit Abzweig für Regelung im Stillstand / In Geschwindigkeit

Der Pfad "InVelocity Control" ist wirksam, wenn der Eingang "Standstill-Enable" nicht gesetzt ist. Die Zugkraft wird hierbei über den Parameter "Getriebefeinabgleich" P-0-0694 (Slave-Achse mit Interpolation auf dem Antrieb / IPO Drive) bzw. A-0-2605 (Slave-Achse mit Interpolation auf der Steuerung / IPO Control) geregelt.

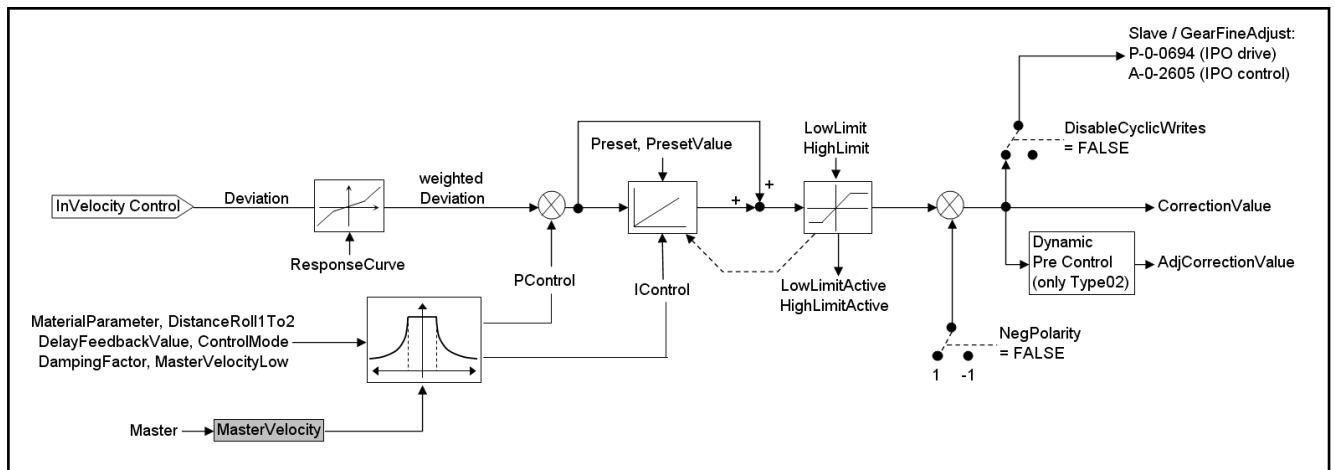


Abb. 4-108: Flussdiagramm des Zugkraftreglers für den Zweig (InVelocity Control)

Für die Regelung im Stillstand kann der Pfad "Standstill Control" aktiv geschaltet werden. Dieser Pfad ist wirksam, wenn der Eingang "Standstill-Enable" gesetzt ist. Die Zugkraft wird hierbei über den Parameter "additive Leitachsposition" P-0-0692 (IPO Drive) bzw. A-0-2600 (IPO Control) geregelt.

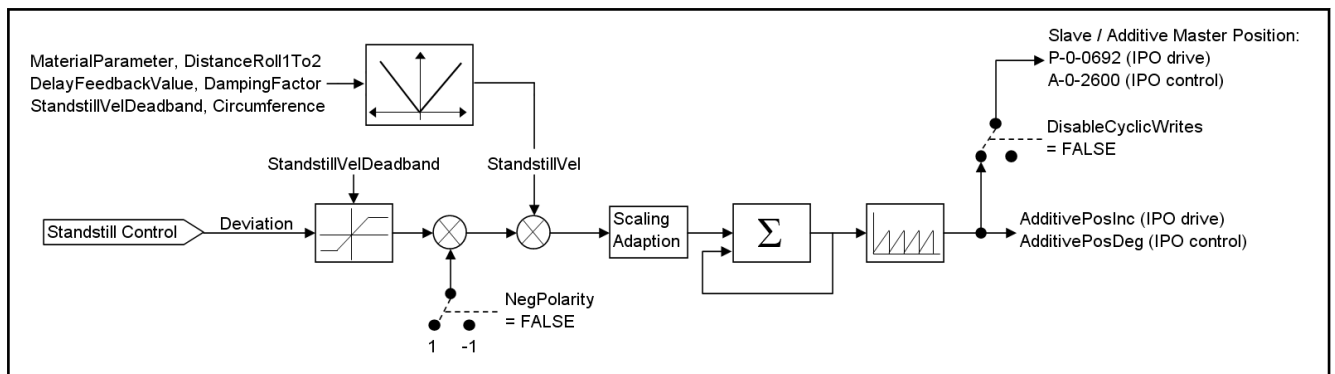


Abb. 4-109: Flussdiagramm des Zugkraftreglers für die Regelung im Stillstand (Standstill Control)

Die Flussdiagramme enthalten aus Gründen der Übersichtlichkeit nicht die Begrenzungen "WindowValue", "WindowMaxValue", "WindowMinValue" sowie deren Statusanzeigen.

Inbetriebnahme

Ist "DisableCyclicWrites" inaktiv, und es wird eine reale Slave-Achse mit Interpolation auf dem Antrieb verwendet, müssen die Parameter P-0-0694 ("In-Velocity Control") und P-0-0692 ("Standstill Control") im zyklischen Sollwertkanal der Folgeachse aufgelegt werden. Bei steigender Flanke an "Enable" werden die zyklischen Datenkanäle überprüft. Wird einer der beiden Parameter nicht gefunden, erfolgt eine entsprechende Fehlermeldung an den Fehlerausgängen des Funktionsbausteins. Ist "DisableCyclicWrites" aktiv, wird die Konfiguration der zyklischen Sollwertdatenkanäle nicht überprüft.

Im Folgenden werden die einzelnen Eingänge des Funktionsbausteins näher erläutert.

Allgemein gilt für alle Eingänge, dass sie kontinuierlich bei gesetztem "Enable"-Eingang ausgewertet werden. Das heißt, Änderungen der Eingänge werden sofort ohne erneutes Toggeln des "Enable"-Eingangs wirksam. Ausnahme dieser Regel bilden die Eingänge "PresetValue", "DisableCyclicWrites", "Circumference", "DistanceRoll1To2", "ResponseCurve", "Master" und "Slave". Der "PresetValue"-Eingang wird hierbei nur bei einer steigenden Flanke am "Preset"-Eingang, und die Eingänge "DisableCyclicWrites", "Cir-

	cumference", "DistanceRoll1To2", "ResponseCurve", "Master" und "Slave" nur bei einer steigenden Flanke am "Enable"-Eingang, übernommen.
Pause, InPause	Der Eingang "Pause" ermöglicht das Anhalten des Zugkraftreglers. Wird der Eingang auf TRUE gesetzt, wird das aktuelle Reglerstellsignal eingefroren. Ist "Pause" aktiv, wird der Ausgang "InPause" gesetzt.
	 Wenn keine Zugkraftregelung im Stillstand genutzt wird (StandstillEnable ist immer FALSE), muss im Maschinenstillstand der Eingang "Pause" gesetzt werden.
Feedback	Dieser Eingang stellt den Istwert der Zugkraft dar. Dieser kann z.B. durch einen Analogeingang am Antrieb oder den Inline I/O eingelesen werden. Der Anwender ist dabei für eine sinnvolle Skalierung verantwortlich.
SetpointLock	Durch Setzen des Einganges "SetpointLock" wird der aktuelle Zugkraftistwert als neuer Sollwert "Setpoint" für den Zugkraftregler übernommen. Die Funktion erlaubt es, z.B. für den Sollwert der Zugkraft einen sinnvollen Startwert zu erhalten, der den momentanen Zustand der Maschine widerspiegelt.
Preset, PresetValue, PresetAck	Durch eine positive Flanke am Eingang "Preset" wird über die Variable "PresetValue" ein Anfangswert für die Stellgröße "CorrectionValue" vorgegeben. Gleichzeitig werden interne Variablen des Reglers so gesetzt, dass ein ruckfreies Anfahren des Reglers erfolgt. Der "Preset" wirkt nur auf den Getriebe-feinabgleich im Pfad "InVelocity Control". Der "Preset" Eingang wird nur ausgewertet, wenn "Enable" gesetzt ist. Bei steigender Flanke an "Preset" wird der neue Wert an "PresetValue" übernommen und "PresetAck" gesetzt. Eine fallende Flanke an "Preset" oder "Enable" löscht "PresetAck". Der "Preset"-Eingang ist dominant über "Pause", d.h. der Wert an "PresetValue" erscheint am Ausgang, auch wenn "Pause" gesetzt ist. Ist bei steigender Flanke an "Enable" "Preset" gesetzt, so wird "PresetValue" sofort als Stellgröße übernommen.
	 Die Preset Funktion sollte im Stillstand ausgeführt werden (StandstillEnable=TRUE), da ansonsten Geschwindigkeitssprünge durch das Setzen des "CorrectionValue" auf den "PresetValue" entstehen können. Der Betrag der Geschwindigkeitssprünge ist proportional zu der Geschwindigkeit. Der "Preset"-Eingang wirkt nur bei der steigenden Flanke.
NegPolarity	Der Eingang "NegPolarity" beeinflusst den Regelsinn. Abhängig von der Lage der Zugwalze ("Slave"-Achse), bezüglich der Durchlaufrichtung des Materials und der Lage der Vorschubwalze ("Master"-Achse), muss der Regelsinn entweder positiv oder negativ sein. Ist dieser Eingang FALSE (positiver Regelsinn), führt eine positive Regelabweichung zu einer positiven Stellbewegung. Der Eingang wirkt auch auf die Stillstandsregelung.
DisableCyclicWrites	Der Eingang "DisableCyclicWrites" bestimmt, ob die vom Funktionsbaustein berechneten Werte direkt über den zyklischen Kanal geschrieben werden (FALSE) oder nur am Bausteinausgang verfügbar sind (TRUE). Ist "DisableCyclicWrites" auf TRUE gesetzt, ist es möglich die Reglerausgänge "CorrectionValue", "AdditivePosInc" und "AdditivePosDeg" abzugreifen und zu modifizieren, ohne dass diese zyklisch auf den Sollwertkanal der Folgeachse aufgelegt werden. Des Weiteren erfolgt in diesem Fall keine Prüfung, ob die optional zyklischen Parameter konfiguriert sind. Infolgedessen ist es in diesem Fall notwendig, die Werte selbst in den optional zyklischen Kanal zu schreiben.

HighLimit, HighLimitAct	Mit diesem Eingang wird die Stellgröße des Reglers auf einen Maximalwert begrenzt. Tritt die Begrenzung ein, wird der I-Anteil des Reglers nicht weiter aufsummiert ("Anti-reset-windup"). Der maximale Wert wird durch den im Antrieb definierten Maximalwert für den Getriebefeinabgleich bestimmt. Tritt eine Begrenzung ein, wird der Ausgang "HighLimitAct" gesetzt.
LowLimit, LowLimitAct	Mit diesem Eingang wird die Stellgröße des Reglers auf einen Minimalwert begrenzt. Tritt die Begrenzung ein, wird der I-Anteil des Reglers nicht weiter aufsummiert ("Anti-reset-windup"). Der minimale Wert liegt bei -95 unabhängig von ggf. kleineren Werten im Antrieb (siehe Getriebefeinabgleich, Seite 303). Tritt eine Begrenzung ein, wird der Ausgang "LowLimitAct" gesetzt.
WindowValue, WindowIn	Mit dem Wert "WindowValue" kann der Anwender ein symmetrisches Fenster definieren innerhalb dessen sich der aktuelle Zugkraftistwert befinden soll. Das Fenster ist wie folgt definiert:

$$\left(\text{SetPoint} - |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right) \leq \text{Feedback} \leq \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right)$$

Abb. 4-110: WindowValue, WindowIn

Ist diese Bedingung erfüllt, wird der Ausgang "WindowIn" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt. Es findet nur eine boolesche Ausgabe, aber keinerlei Begrenzung statt.

WindowMaxValue, WindowMax	Mit diesem Wert kann der Anwender einen oberen Wert für den Zugkraftistwert festlegen. Der Wert wird wie folgt ausgewertet:

$$\text{Feedback} > \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowMaxValue} - 100}{100} \right)$$

Abb. 4-111: WindowMaxValue, WindowMax

Ist diese Bedingung erfüllt, wird der Ausgang "WindowMax" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt. Es findet nur eine boolesche Ausgabe, aber keinerlei Begrenzung statt.

WindowMinValue, WindowMin	Mit diesem Wert kann der Anwender einen unteren Wert für den Zugkraftistwert festlegen. Der Wert wird wie folgt ausgewertet:

$$\text{Feedback} < \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowMinValue} - 100}{100} \right)$$

Abb. 4-112: WindowMinValue, WindowMin

Ist diese Bedingung erfüllt, wird der Ausgang "WindowMin" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt. Es findet nur eine boolesche Ausgabe, aber keinerlei Begrenzung statt.

Circumference, DistanceRoll1To2	Die Eingänge "Circumference" und "DistanceRoll1To2" dienen zur internen Berechnung der Reglerparameter. Sie beschreiben die individuellen konstruktiven Gegebenheiten der Anlage. Der Eingang "Circumference" beschreibt den Umfang der "Master"-Achse, und der Eingang "DistanceRoll1To2" die Materialbahnlänge zwischen der "Master"- und der "Slave"-Achse.

DelayFeedbackValue	Der Eingang "DelayFeedbackValue" dient der internen Berechnung der Reglerparameter. Er beschreibt die Verzugszeit T_u der Bahnzugkraftregelstrecke. Somit entspricht dieser Wert der Summe aller peripherer Zeitkonstanten wie Motordynamik, Filterzeiten, Messaufnehmer, Motionzykluszeit.

StandstillEnable, StandstillVelDeadband

Durch Setzen des Eingangs "StandstillEnable" wird zur Regelung der Bahnzugkraft eine additive Geschwindigkeit auf die zu regelnde Achse aufgeschaltet. Diese Geschwindigkeit ist hierbei - solange die Begrenzung "StandstillVelDeadband" nicht aktiv ist - direkt proportional zur Regeldifferenz.



Der Getriebefeinabgleich verliert bei niedrigen Geschwindigkeiten immer mehr an Einfluss. Bei Stillstand der Leitachse besteht sogar ein offener Regelkreis, da der Getriebefeinabgleich dann keinen Einfluss mehr hat. Daher muss der Eingang "StandstillEnable" bei Erreichen des Stillstandsfensters gesetzt werden, damit im Pfad "InVelocity Control" der I-Anteil durch den offenen Regelkreis nicht bis zur Grenze ansteigt.



Bei Achsen mit Interpolation auf der Steuerung wird der Ausgang "AdditivePosInc" im Moduloformat 2^{20} ausgegeben.

Die Stellsignalbegrenzung "StandstillVelDeadband" entspricht einer Begrenzung der Regeldifferenz und somit - aufgrund des proportionalen Reglers - ebenfalls einer Begrenzung des Stellsignals. Der Wert besitzt dieselbe Einheiten wie der "Setpoint"- bzw. "Feedback"- Eingang. Ist die Differenz aus "Setpoint" und "Feedback" größer als der am Eingang "StandstillVelDeadband" vorgegebene Wert, so wird die Spanngeschwindigkeit begrenzt. Diese Begrenzung der Spanngeschwindigkeit ist unter anderem bei einer neu eingelegten ungespannten Materialbahn relevant. Dieser Eingang sollte entsprechend so eingestellt werden, dass regelungstechnisch relevante Regeldifferenzen ohne Begrenzung ausgeregelt werden können, und prozesstechnisch bedingte Regeldifferenzen durch die begrenzte Spanngeschwindigkeit berücksichtigt werden. Ein "StandstillVelDeadband" von 5% bis 25% des "Setpoint" sollte für die meisten Prozesse ausreichen.



Im Stillstand wird die Achsgeschwindigkeit sprunghaft erzeugt. Da die Geschwindigkeit innerhalb eines sercos Zyklus erreicht werden muss, können keine beliebig hohen Werte vorgegeben werden und der Zugkraftregler muss in der sercos synchronen Task verwendet werden. Der Maximalwert richtet sich dabei nach der maximalen Beschleunigung des Antriebs sowie der angeschlossenen Masse. Um zu große Geschwindigkeitssprünge zu vermeiden, kann zum einen die Filterzeitkonstante im Antriebsparameter P-0-0693 vergrößert werden (es wird die 4-fache sercos Zykluszeit empfohlen) und/oder zum anderen das maximale Stellsignal durch das Stillstandsfenster "StandstillVelDeadband" begrenzt werden.

Um eine umgekehrte Bewegungsrichtung für die Regelung im Stillstand zu erreichen, ist der Eingang "NegPolarity" zu setzen.

ResponseCurve

Mit Hilfe des Eingang "ResponseCurve" kann die Regelabweichung gewichtet werden. Durch diese Funktionalität kann beispielsweise bei kleinen Regelabweichungen eine geringere Gewichtung der Regelabweichung eingestellt werden. Dies ergibt quasi eine geringere Proportionalverstärkung und dadurch ein ruhigeres Regelverhalten im eingeschwungenen Zustand. Bei der Parametrierung des Eingangs sollten die Reglerparameter zunächst ohne Bewertung eingestellt werden und erst danach eine geringere Gewichtung vorgegeben werden. Hierdurch wird erreicht, dass die Reglerparameter zunächst global anhand der Systemdynamik eingestellt werden und anschließend eine lokale Optimierung (z. B. für den eingeschwungenen Zustand) durchgeführt wird.

Parametriert wird diese lokale Optimierung mittels einer Kennlinie mit bis zu 4 Stützstellen. Negative Abweichungen werden durch Punktspiegelung der Kurve am Ursprung behandelt. Daraus ergeben sich insgesamt neun lineare Sektionen. Die Sektion s5 verläuft durch den Ursprung. Die Sektionen s1 und s9 haben einen Anstieg von 1 und laufen jeweils ins positive oder negative Unendliche. Wird der Eingang unbeschaltet gelassen, wird als Default-Wert eine 1-zu-1 Kennlinie verwendet, d. h. es wird keinerlei Gewichtung durchgeführt.

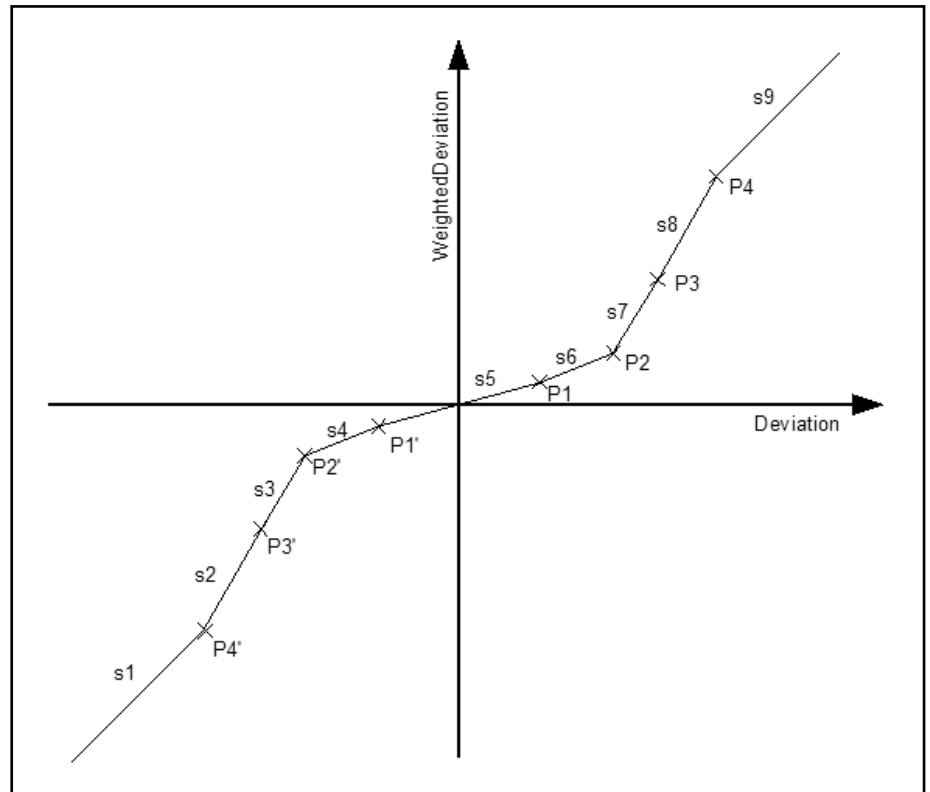


Abb. 4-113: Wichtung der Regelabweichung mit dem Eingang ResponseCurve

Die Punkte P1 bis P4 werden in einem Array aus IL_COORD_TYPE01 abgelegt. Werden weniger als vier Punkte benötigt, muss die X-Koordinate der nicht benötigten Punkte (vom letzten Punkt beginnend) auf 0 gesetzt werden. Die X-Werte müssen positiv sein und in positiver Richtung aufsteigen. Es ist erlaubt, denselben Punkt zweimal abzulegen.

MasterVelocityLow

Der Eingang "MasterVelocityLow" dient zur Begrenzung der Reglerparameter. Infolge des geschwindigkeitsabhängigen Streckenverhaltens ergeben sich geschwindigkeitsabhängige Reglerparameter. Abhängig vom gewünschten Regelkreisverhalten ist entweder die Reglerverstärkung und/oder die Reglernachstellzeit geschwindigkeitsabhängig. Nachfolgend ist stellvertretend die Abhängigkeit der Reglerverstärkung von der Bahngeschwindigkeit grafisch dargestellt.

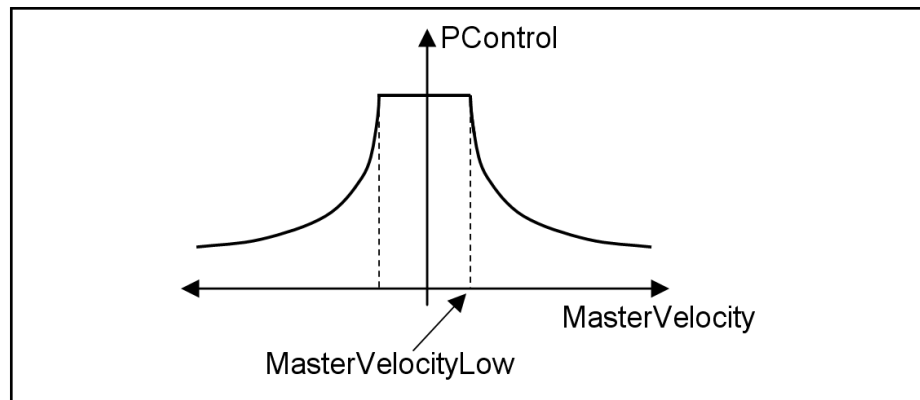


Abb. 4-114: Abhängigkeit der Reglerverstärkung von der Bahngeschwindigkeit

Wie bereits für "StandstillEnable" dargelegt wurde, verliert der Getriebefei-nabgleich bei niedrigen Geschwindigkeiten immer mehr an Einfluss. Dieses Verhalten spiegelt sich direkt in der Reglerverstärkung wieder. Um unreal gro-ßen Reglerparametern vorzubeugen, dient der Eingang "MasterVelocityLow" zur Begrenzung der Reglerparameter bei "kleinen" Bahngeschwindigkeiten. Der Wert dieses Eingangs sollte in den meisten Fällen unterhalb der klein-sten Produktionsgeschwindigkeit und oberhalb der Umschaltgeschwindigkeit in den Stillstandspfad liegen. Somit ist eine uneingeschränkte Regelkreisdyn-amik oberhalb der "MasterVelocityLow"-Geschwindigkeit gewährleistet, und gleichzeitig werden die Reglerparameter bei "kleinen" Bahngeschwindigkeiten begrenzt.

MaterialParameter

Die Eingangsstruktur "MaterialParameter" dient zur internen Berechnung der Reglerparameter. Diese Struktur beinhaltet Informationen über das individu-elle Bahnmaterial. Die einzelnen Elemente dieser Struktur beschreiben den Typ, die Breite, die Dicke bzw. die Grammatur und den Elastizitätsmodul des Bahnmaterials. Bei Bahnmaterialien aus Papier wird hierbei die Grammatur "WebGrammage" und bei anderen Bahnmaterialien die Materialdicke "Web-Thickness" ausgewertet. Bei Auswahl eines vorgegebenen Bahnmaterials "MaterialParameter.MaterialType", wird das Elastizitätsmodul aus der inter-nen Datenbank bezogen. Entsprechend wird der Eingang "MaterialParame-ter.MaterialEModule" ausschließlich ausgewertet, wenn ein benutzerdefinier-ter Materialtyp "CUSTOM_PAPER" oder "CUSTOM_NON_PAPER" ausge-wählt wird. Zur Verdeutlichung sind nachfolgend zwei beispielhafte Beschal-tungen dargestellt:

Beispiel:

Papier aus bausteininterner Datenbank

```
MaterialParameter.MaterialType := OFFSET_PAPER;
MaterialParameter.WebWidth := 500;
MaterialParameter.WebGrammage := 80;
```

Beispiel:

Benutzerdefiniertes Bahnmaterial (z.B. Folie)

```
MaterialParameter.MaterialType := CUSTOM_NON_PAPER;
MaterialParameter.WebWidth := 500;
MaterialParameter.WebThickness := 200;
MaterialParameter.MaterialEModule := 650;
```

Ist das verwendete Bahnmaterial nicht in der internen Datenbank aufgezählt, und es liegen keine Informationen über den Elastizitätsmodul vor, so kann dieses in einer Stillstandsmessung ermittelt werden.

$$\Delta F \approx A \cdot E \cdot \frac{\Delta L}{L} \Rightarrow E \approx \frac{L \cdot \Delta F}{A \cdot \Delta L}$$

Abb. 4-115: Hookesches Gesetz

Entsprechend dieser Hookeschen Formel ist ersichtlich, dass durch eine gemessene Kraftdifferenz, infolge einer vorgegebenen Streckung (Längenänderung infolge einer additiven Leitachspolition), das individuelle Elastizitätsmodul [N/mm²] im Stillstand ermittelt werden kann. Des Weiteren kann ebenfalls direkt das Produkt aus Elastizitätsmodul [N/mm²] und Querschnittsfläche [mm²] ermittelt werden, und durch Normierung der Querschnittsfläche ("MaterialType:=CUSTOM_NON_PAPER", "WebWidth:=1[mm]", "WebThickness:=1000[μm]") auf A=1[mm²] dieses als modifiziertes Elastizitätsmodul [N] vorgegeben werden.

ControlMode Durch den Eingang "ControlMode" kann das primäre Regelkreisverhalten festgelegt werden. Der Bahnzugkraftregler kann entweder auf Führungsverhalten oder auf Störverhalten optimiert werden. Bei einer Festwertregelung der Bahnzugkraft sollte dieser Eingang mit "CONTROL_MODE_OPT_DISTURBANCE" belegt werden. Infolgedessen werden die internen Reglerparameter so bestimmt, dass Störungen optimal ausgeglichen werden. Entgegengesetzt sollte bei einer Folgeregelung dieser Eingang mit "CONTROL_MODE_OPT_COMMAND" belegt werden. Infolgedessen werden die internen Reglerparameter so bestimmt, dass die Regelgröße optimal der Führungsgröße folgt.

DampingFactor Durch den Eingang "DampingFactor" kann die Regelkreisdynamik festgelegt werden. Er wirkt auf beide Zweige, Stillstandsregelung und Regelung im Bahnlaufbetrieb. Die Regelkreisdynamik kann entweder aperiodisch (ohne Überspringen) oder periodisch (mit Überspringen) eingestellt werden. Hierbei entspricht ein "DampingFactor" von 100% - bei exakten Streckenparametern - genau dem aperiodischen Grenzfall, und ein „DampingFactor“ von 120% einem periodischen Einschwingen mit ca. 20% Überspringen. Des Weiteren kann das Regelkreisverhalten bis zu einem "DampingFactor" von 20% verlangsamt, bzw. bis zu einem "DampingFactor" von 200% beschleunigt werden. Liegen vom realen Verhalten abweichende Streckenparameter ("Circumference", "DistanceRoll1To2", "DelayFeedbackValue", "MaterialParameter") oder Systemverhalten (z.B. orts- und zeitabhängige Feuchtigkeit und Temperaturen) vor, so kann des Weiteren über diesen Eingang eine Korrektur des gewünschten dynamischen Regelkreisverhaltens erzielt werden.

Hintergrundinformationen

Getriebefeinabgleich Der Zugkraftregler arbeitet in der Regel mit Geschwindigkeiten größer "0". Dabei ist der Getriebefeinabgleich ein probates Mittel um die Zugkraft zu regeln.

Der Getriebefeinabgleich wird intern auf ein Minimum von -95% begrenzt. Wird ein kleinerer Wert eingegeben, erzeugt der Baustein einen Fehler.

Hintergrund dieser Begrenzung ist der errechnete Ausgangswert der bei gegebenem Eingangswert wie folgt definiert ist:

$$\text{Ausgangswert} = \text{Eingangswert} * \left(1 + \frac{\text{Getriebefeinabgleich}}{100} \right)$$

Abb. 4-116: Ausgangswert

Entsprechend dieser formalen Beschreibung ist erkennbar, dass bei einem Getriebefeinabgleich kleiner "-100%" eine Umkehr der Drehrichtung resultiert.

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle **F_RELATED_TABLE**, 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0004)	16#00000A00	16#00000001	Der Parameter P-0-0694 ist nicht im optional zyklischen Telegramm der Folgeachse parametrier
ACCESS_ERROR (16#0004)	16#00000A00	16#00000002	Der Parameter P-0-0692 ist nicht im optional zyklischen Telegramm der Folgeachse parametrier
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000001	Die Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000005	Der Wert des Eingangs "LowLimit" ist kleiner als der minimal zulässige (-95.0)
DEVICE_ERROR (16#0008)	16#00000A02	16#00000005	Die Sercoszykluszeit (C-0-0503) ist kleiner oder gleich 0
DEVICE_ERROR (16#0008)	16#00000A02	16#00000006	Die Getriebeübersetzung der geregelten Achse ist unzulässig (A-0-2720, A-0-2721)
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000002	Der am Eingang "ControlMode" angegebene Wert ist nicht im Datentyp "MB_TENSION_CONTROL_MODE" enthalten
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000003	Der am Eingang "MaterialParameter.MaterialType" angegebene Wert ist nicht im Datentyp "MB_TENSION_CONTROL_MATERIAL_TYPE" enthalten.
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000007	Der Eingang "StandstillVelDeadband" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000009	Der Eingang "MaterialParameter.WebWidth" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000010	Der Eingang "MaterialParameter.WebThickness" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000011	Der Eingang "MaterialParameter.MaterialEModul" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000012	Der Eingang "MaterialParameter.WebGrammage" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000013	Der Eingang "DampingFactor" ist kleiner als 20 oder größer als 200

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000014	Der Eingang "Circumference" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000015	Der Eingang "DistanceRoll1To2" ist kleiner oder gleich 0
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Zustandsmaschine des Funktionsbausteins befindet sich in einen ungültigen Status
RESOURCE_ERROR (16#0003)	16#00001503	16#00000000	AxisData ist in der Steuerung nicht aktiv. Dieser Fehler tritt nur bei der MLD auf

Tab. 4-57: Fehlercodes Funktionsbaustein MB_TensionControlLoadCellMatParType01

4.8.4 MB_TENSION_CONTROL_MATERIAL

In der folgenden Tabelle sind die Eigenschaften der Struktur MB_TENSION_CONTROL_MATERIAL aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
MaterialType	MB_TENSION_CONTROL_MATERIAL_TYPE, Seite 305	CUSTOM_PAPER	BOPP_FILM	OFFSET_PAPER	kontinuierlich	Art des Bahnmaterials
WebWidth	REAL	>0.0	n.def.	500	kontinuierlich	Breite des Bahnmaterials [mm]*
WebThickness	REAL	>0.0 (Wenn MaterialType ist kein Papier)	n.def.	0	kontinuierlich (Wenn "MaterialType" ist kein Papier)	Dicke des Bahnmaterials [µm]*
WebGrammage	REAL	>0.0 (Wenn MaterialType ist Papier)	n.def.	80	kontinuierlich (Wenn "MaterialType" ist Papier)	Grammatur des Bahnmaterials [g/m²]
MaterialEModule	REAL	>0.0 (Wenn MaterialType ist CUSTOM_...)	n.def.	0	kontinuierlich	Elastizitätsmodul des Bahnmaterials [N/mm²]*

*

Die Einheiten [mm] und [µm] sind bei Verwendung der internen Materialdatenbank (E [N/mm²]) obligatorisch. Bei der Verwendung von benutzerdefinierten Elastizitätsmoduln ist die Basiseinheit [N/mm²]. Es können jedoch beliebige Einheiten wie z.B. [kg/ft²] verwendet werden. Es ist darauf zu achten, dass die gewählten Einheiten einheitlich verwendet werden.

Tab. 4-58: Strukturbeschreibung MB_TENSION_CONTROL_MATERIAL

4.8.5 MB_TENSION_CONTROL_MATERIAL_TYPE

In der folgenden Tabelle sind die Eigenschaften des Aufzählungstyps MB_TENSION_CONTROL_MATERIAL_TYPE aufgelistet.

Aufzählungselement	Nummer	Beschreibung
CUSTOM_PAPER	1	Benutzerdefiniertes Papier
PACKING_PAPER	2	Verpackungspapier
OFFSET_PAPER	3	Offsetpapier (Default)
METALLIZED_PAPER	4	Metallisiertes Papier
CUSTOM_NON_PAPER	5	Benutzerdefiniert kein Papier
PE_FILM	6	Polyethylen Folie (PE-Folie)
PET_FILM	7	Polyethylenterephthalat Folie (PET-Folie)
PP_FILM	8	Polypropylen Folie (PP-Folie)
OPP_FILM	9	Orientierte Polypropylen Folie (OPP-Folie)
BOPP_FILM	10	Biaxial orientierte Polypropylen Folie (BOPP-Folie)

Tab. 4-59: Aufzählungstyp MB_TENSION_CONTROL_MATERIAL_TYPE

4.8.6 MB_TENSION_CONTROL_MODE

In der folgenden Tabelle sind die Eigenschaften des Aufzählungstyps MB_TENSION_CONTROL_MODE aufgelistet.

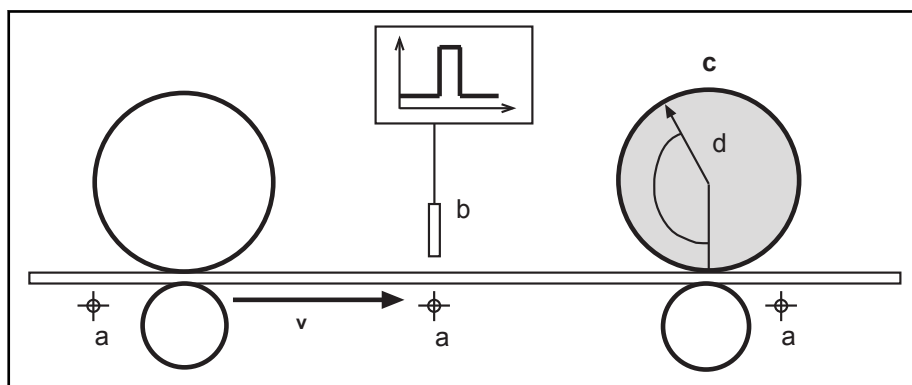
Aufzählungselement	Nummer	Beschreibung
CONTROL_MODE_OPT_DISTURBANCE	1	Baustein regelt optimiert auf Störverhalten den "Feedback"-Wert auf den "Setpoint"-Wert (Default)
CONTROL_MODE_OPT_COMMAND	2	Baustein regelt optimiert auf Führungsverhalten den "Feedback"-Wert auf den "Setpoint"-Wert

Tab. 4-60: Aufzählungstyp MB_TENSION_CONTROL_MODE

5 ML_TechRegi.library // MX_TechRegi.lib

5.1 Einführung und Übersicht

Auf Papierbahnen oder Papierbogen gedruckte Registermarken werden durch einen digitalen Eingang erkannt und anschließend von einem Registerregler als Lagemesswert erkannt (Lagemessung). Die Positionsversatz, die Geschwindigkeit des Antriebs oder die Leitachsantriebsparameter werden durch die Registerregler-FB angepasst.



- a:** Registermarke
b: Registermesstaster
c: Druckzylinder
d: ϕ drucken

Abb. 5-1: Messprinzip mit dem Papierbahndrucken

Der Registerregler regelt die Lage des Materials in Bezug auf die Lage einer Bezugsachse. Diese kann die Lage des Bogens in Bezug auf den Druckzylinder sein. Die Regelung wird mit Hilfe eines elektrischen Eingangssignals an der Bezugsachse durchgeführt. Das elektrische Eingangssignal wird durch das optische Messen einer Registermarke auf dem Material zur Verfügung gestellt.



Die Bibliothek benötigt das **TEC (Technologie) Funktionspaket** der Steuerung

Funktionsbausteine von ML_TechRegi (MLC)/ MX_TechRegi (MLD)

Anwendungsgebiet (Ordner in der Bibliothek)			
Funktionsbaustein	MLC	MLD MPx07/08/17 (IL1G)	MLD MPx20 (IL2G)
Bearbeitung von gemessenen Messtastersignalen (RegiMeasuring)			
MB_RegiMeasuringType01, Seite 308	X	X	
MB_RegiMeasuringType02, Seite 310	X	X	
MB_RegiMeasuringType03, Seite 314	X	X	X
Qualitätsbewertung von Druckbildern (RegiQuality)			
IL_RegiQualityType01, Seite 318	X	X	
Schreiben eines Stellwertes an den Antrieb (RegiRegulate)			

MB_RegiRegulateType01, Seite 321	X	X	
MB_RegiRegulateType02, Seite 326	X	X	X
MB_RegiRegulateType03, Seite 334	X	X	
MB_RegiRegulateType04, Seite 341	X		
Verriegeln eines Werts bei einem Registerregler (RegiSetpointLock)			
MB_SetpointLockType01, Seite 350	X	X	X
Registerregler-FB (RegisterController)			
MB_RegisterControllerType02, Seite 352	X	X	
MB_RegisterControllerType04, Seite 362	X	X	X
Registerregler-FB (RegisterControllerSide)			
MB_RegisterControllerSideType01, Seite 381	X	X	

Tab. 5-1: Überblick über Funktionsbausteinunterstützung durch die Bibliothek

5.2 Messfunktionsbausteine für Registerregler

5.2.1 Einführung und Übersicht

Dieser Abschnitt beschreibt die Funktionsbausteine zum Messen, die dazu eingesetzt werden können kundenspezifische Registerregler-FB, herzustellen.

5.2.2 MB_RegiMeasuringType01

Kurzbeschreibung

Der Funktionsbaustein MB_RegiMeasuringType01 wird zur Bearbeitung des gemessenen Messtastersignals verwendet.

Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechRegi
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechRegi.lib

Tab. 5-2: Referenztabelle MB_RegiMeasuringType01

Schnittstellenbeschreibung

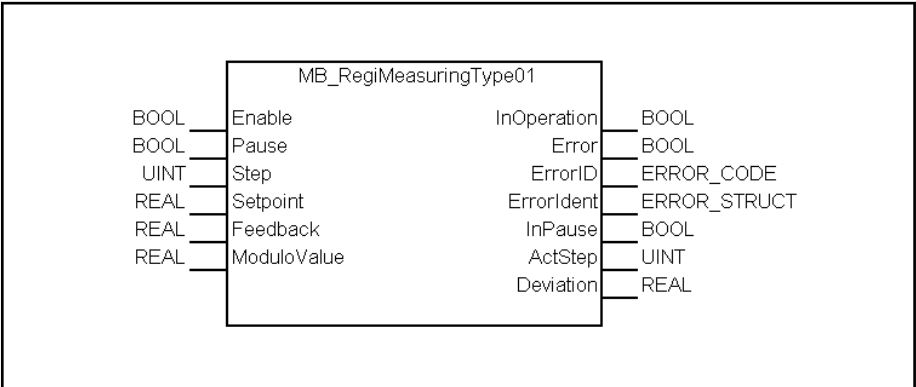


Abb. 5-2: Funktionsbaustein MB_RegiMeasuringType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist
	Step	UINT	Neue Werte werden berechnet wenn sich der "Step"-Eingang ändert
	Setpoint	REAL	Sollwert
	Feedback	REAL	Gemessener Wert
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
VAR_OUTPUT	InOperation	BOOL	Messfunktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt an, dass es zu einem Fehler während des Funktionsbausteinbetriebes kam
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 310
	InPause	BOOL	"Pause" ist aktiv. Der Ausgang "Deviation" behält seinen Wert
	ActStep	UINT	Schritt看t; zählt mit jedem neuen Wert am Ausgang "Deviation" hoch
	Deviation	REAL	"Setpoint" - "Feedback"

Tab. 5-3: Schnittstellenvariablen Funktionsbaustein MB_RegiMeasuring-Type01

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
Setpoint	REAL	0	ModuloValue	0.0	Änderung von "Step"
Feedback	REAL	0	ModuloValue	0.0	Änderung von "Step"
ModuloValue	REAL	0	n.def	0.0	Steigende Flanke an "Enable"

Tab. 5-4: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm

Zeitdiagramm gemäß der Spezifikation PLCopen.

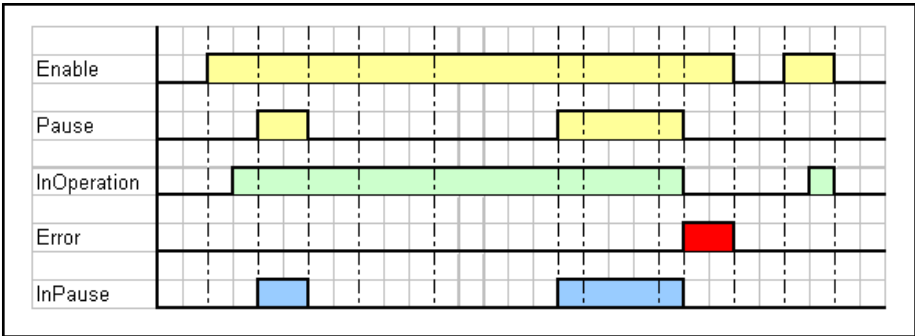


Abb. 5-3: Zeitdiagramm des MB_RegiMeasuringType01

- Funktionsbeschreibung**
- Der Funktionsbaustein MB_RegiMeasuringType01 subtrahiert die Eingänge "Setpoint" und "Feedback" voneinander ("Setpoint" - "Feedback"). Das Ergebnis der Subtraktion ist der Wert des Ausgangs "Deviation".
- Pause**
- Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und der "Deviation"-Ausgang behält seinen Wert.
- Fehlerbehandlung**
- Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse ("Axis_Ref") wurde geändert während der FB in Betrieb war
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0002	Eingang "Setpoint" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0003	Eingang "Setpoint" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0004	Eingang "Feedback" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0005	Eingang "Feedback" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0006	Eingang "ModuloValue" < 0

Tab. 5-5: Fehlercodes Funktionsbaustein MB_RegiMeasuringType01

5.2.3

MB_RegiMeasuringType02

Kurzbeschreibung

Der Funktionsbaustein MB_RegiMeasuringType02 wird zur Bearbeitung des gemessenen Messtastersignals verwendet.

Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechRegi
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechRegi.lib

Tab. 5-6: Referenztable MB_RegiMeasuringType02

Schnittstellenbeschreibung

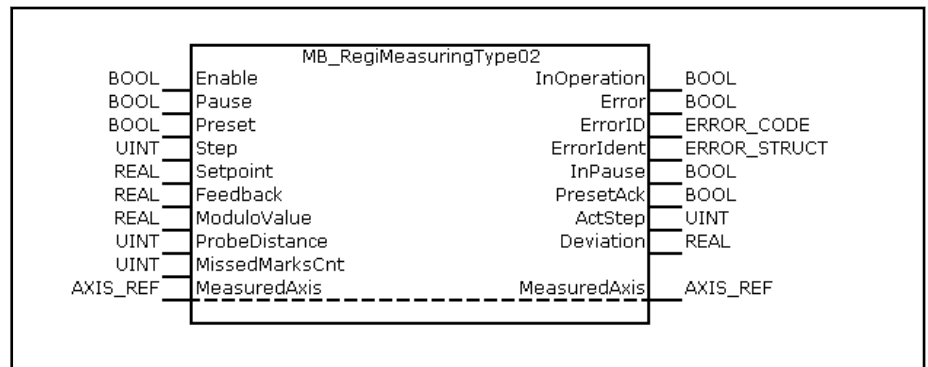


Abb. 5-4: Funktionsbaustein MB_RegiMeasuringType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt den Speicher des Schieberegisters zwangsweise auf "0"
	Step	UINT	Neue Werte werden berechnet wenn sich der "Step"-Eingang ändert
	Setpoint	REAL	Sollwert
	Feedback	REAL	Gemessener Wert
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
	ProbeDistance	UINT	Der Abstand zwischen dem Markensensor und der Mess- (= geregelten) Achse in Leitachsumdrehungen (Formatlängen)
	MissedMarksCnt	UINT	Zähler für Markenausfälle, wenn ein Erwartungsfenster verwendet wird
VAR_OUTPUT	InOperation	BOOL	Messfunktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt an, dass es zu einem Fehler während des Funktionsbausteinbetriebes kam
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 313
	InPause	BOOL	"Pause" ist aktiv. Der Ausgang "Deviation" behält seinen Wert
	PresetAck	BOOL	Preset-Funktion ist abgeschlossen
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "Deviation" hoch

E/A-Typ	Name	Datentyp	Beschreibung
	Deviation	REAL	"Setpoint" - "Feedback" verzögert wenn Schieberegister aktiv ist
VAR_IN_OUT CONSTANT	MeasuredAxis	AXIS_REF	Bezug zur gemessenen Achse

Tab. 5-7: Schnittstellenvariablen Funktionsbaustein MB_RegiMeasuringType02

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	"Enable" TRUE (kontinuierlich)
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
Setpoint	REAL	0	ModuloValue	0.0	Änderung von "Step"
Feedback	REAL	0	ModuloValue	0.0	Änderung von "Step"
ModuloValue	REAL	0	n.def	0.0	Steigende Flanke an "Enable"
ProbeDistance	UINT	0	30	0	Kontinuierlich
MissedMarksCnt	UINT	0	65.535	0	Kontinuierlich

Tab. 5-8: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm

Zeitdiagramm gemäß der Spezifikation PLCopen.

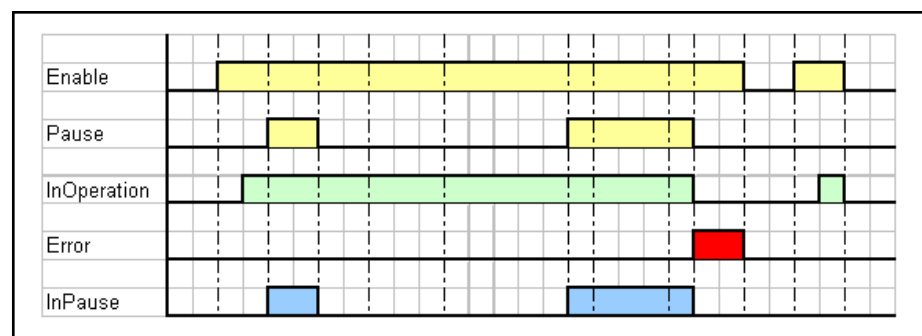


Abb. 5-5: Zeitdiagramm des MB_RegiMeasuringType02

Funktionsbeschreibung

Der Funktionsbaustein MB_RegiMeasuringType02 subtrahiert die Eingänge "Setpoint" und "Feedback" voneinander ("Setpoint" - "Feedback"). Das Ergebnis der Subtraktion ist der Wert des Ausgangs "Deviation" wenn das Schieberegister deaktiviert ist. Wenn das Schieberegister aktiviert ist, wird es um die Anzahl der "ProbeDistance" verzögert.

Pause

Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und der "Deviation"-Ausgang behält seinen Wert.

Preset Das Schieberegister kann vorab mit dem Eingang "Preset" beschrieben werden. Damit der Schieberegisterspeicher vorab beschrieben wird, muss "Enable" aktiv sein. Ein Nullwert wird in den Speicher bei ansteigender Flanke des Eingangs "Preset" geschrieben wenn "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt.

ProbeDistance Der Eingang "ProbeDistance" ist der Abstand zwischen dem Markensensor und der Mess- (= geregelten) Achse, in Leitachsumdrehungen (Formatlängen). Die maximale Formatlänge beträgt 30. Ist "ProbeDistance" 0, ist das Schieberegister deaktiviert. Der Ausgang "Deviation" wird auf "Setpoint" – "Feedback" gesetzt.

Wenn "ProbeDistance" nicht 0 ist, wird das Schieberegister aktiviert. Das bedeutet, dass die Werte beim Ausgang "Deviation" um die Anzahl der Leitachsdrehungen (Formatlängen) am Eingang "ProbeDistance" verzögert werden.

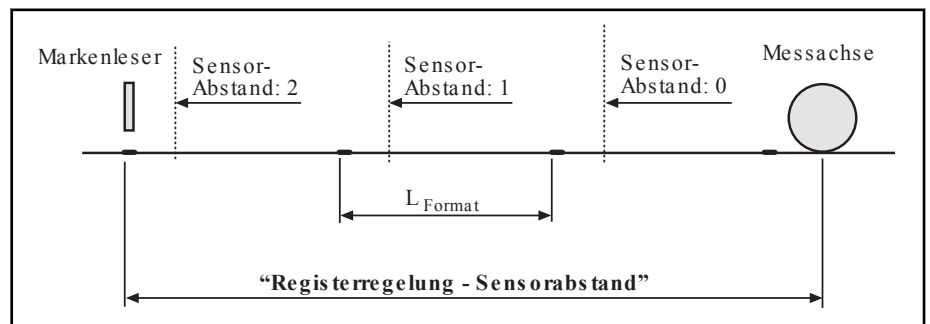


Abb. 5-6: Eingang „ProbeDistance“ setzen

MissedMarkCnt Das ist der Markenausfallüberwachungszähler der Messtasterfunktion wenn das Erwartungsfenster aktiv ist. Wenn eine Marke fehlt, wird eine Null in das Schieberegister geschrieben wenn es aktiviert ist. Im Falle eines deaktivierten Schieberegisters wird die Null in den Ausgang "Deviation" geschrieben.

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse ("Axis_Ref") wurde geändert während der FB in Betrieb war
RESOURCE_ERROR (16#0003)	16#0012	16#0003	"AXIS_REF.CntrlNo" liegt außerhalb des gültigen Bereiches (Achsen von entfernten Steuerungen sind nicht zulässig)
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0001	Eingang "ProbeDistance" ist außerhalb des gültigen Wertebereichs ($0 \leq \text{"ProbeDistance"} \leq 30$)
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0002	Eingang "Setpoint" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0003	Eingang "Setpoint" < 0

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0004	Eingang "Feedback" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0005	Eingang "Feedback" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0006	Eingang "ModuloValue" < 0

Tab. 5-9: Fehlercodes Funktionsbaustein MB_RegiMeasuringType02

5.2.4 MB_RegiMeasuringType03

Kurzbeschreibung Der Funktionsbaustein MB_RegiMeasuringType03 wird zur Bearbeitung des gemessenen Messtastersignals verwendet.

Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechRegi
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechRegi.lib (IL1G)
MLD mit MPx20 mit MA Funktionspaket	MX_TechRegi (IL2G)

Tab. 5-10: Referenztable MB_RegiMeasuringType03

Schnittstellenbeschreibung

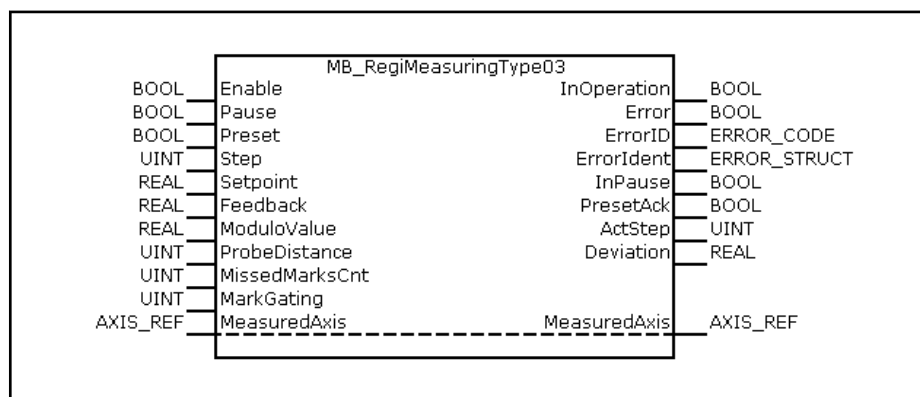


Abb. 5-7: Funktionsbaustein MB_RegiMeasuringType03

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt den Speicher des Schieberegisters zwangsweise auf "0"
	Step	UINT	Neue Werte werden berechnet wenn sich der "Step"-Eingang ändert
	Setpoint	REAL	Sollwert
	Feedback	REAL	Gemessener Wert
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse

E/A-Typ	Name	Datentyp	Beschreibung
	ProbeDistance	UINT	Der Abstand zwischen dem Markensensor und der Messachse in Leitachsumdrehungen (Formatlängen)
	MissedMarksCnt	UINT	Zähler der fehlenden Marken wenn ein Erwartungsfenster verwendet wird
	MarkGating	UINT	Jede n-te Marke wird für die Registrierung verwendet
VAR_OUTPUT	InOperation	BOOL	Messfunktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt an, dass es zu einem Fehler während des Funktionsbausteinbetriebes kam
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 317
	InPause	BOOL	"Pause" ist aktiv. Der Ausgang "Deviation" behält seinen Wert
	PresetAck	BOOL	Preset-Funktion ist abgeschlossen
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "Deviation" hoch
	Deviation	REAL	"Setpoint" - "Feedback" verzögert, wenn Schieberegister aktiv ist
VAR_IN_OUT CONSTANT	MeasuredAxis	AXIS_REF	Bezug zur gemessenen Achse

Tab. 5-11: Schnittstellenvariablen Funktionsbaustein MB_RegiMeasuring-Type03

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	"Enable" TRUE (kontinuierlich)
Step	UINT	0	65.535	0	Kontinuierlich Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
Setpoint	REAL	0	ModuloValue	0.0	Änderung von "Step"
Feedback	REAL	0	ModuloValue	0.0	Kontinuierlich
ModuloValue	REAL	0	n.def	0.0	Steigende Flanke an "Enable"
ProbeDistance	UINT	0	60	0	Kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
MissedMarksCnt	UINT	0	65.535	0	Kontinuierlich
MarkGating	UNIT	0	65.535	1	Steigende Flanke an "Enable". Ein Wert von "0" deaktiviert den Markentakt (identisch mit dem Wert "1")

Tab. 5-12: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm

Zeitdiagramm gemäß der Spezifikation PLCopen.

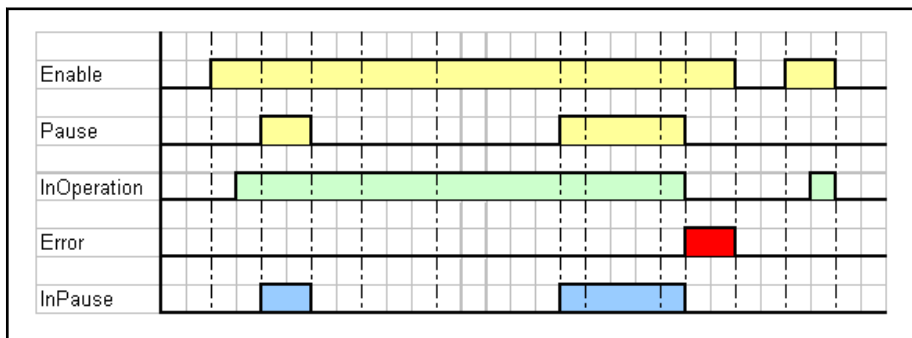


Abb. 5-8: Zeitdiagramm des MB_RegiMeasuringType03

Funktionsbeschreibung

Der Funktionsbaustein MB_RegiMeasuringType03 subtrahiert die Eingänge "Setpoint" und "Feedback" ("Setpoint" - "Feedback"). Das Ergebnis der Subtraktion ist der Wert des Ausgangs "Deviation" wenn das Schieberegister deaktiviert ist. Wenn das Schieberegister aktiviert ist, wird es um die Anzahl der "ProbeDistance" verzögert. Nur jede n-te Marke wird für die Registrierung mit der Markenausfensterungsfunktion des FB verwendet.

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und der "Deviation"-Ausgang behält seinen Wert.

Preset Das Schieberegister kann vorab mit dem Eingang "Preset" beschrieben werden. Damit der Schieberegisterspeicher vorab beschrieben wird, muss "Enable" aktiv sein. Ein Nullwert wird in den Speicher bei ansteigender Flanke des Eingangs "Preset" geschrieben wenn "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt.

ProbeDistance Der Eingang "ProbeDistance" ist der Abstand zwischen dem Markensensor und der Mess- (= geregelten) Achse, in Leitachsumdrehungen (Formatlängen). Die maximale Formatlänge beträgt 60. Ist "ProbeDistance" 0, ist das Schieberegister deaktiviert. Der Ausgang "Deviation" wird auf "Setpoint" - "Feedback" gesetzt.

Wenn "ProbeDistance" nicht 0 ist, wird das Schieberegister aktiviert. Das bedeutet, dass die Werte beim Ausgang "Deviation" um die Anzahl der Leitachsdrehungen (Formatlängen) am Eingang "ProbeDistance" verzögert werden.

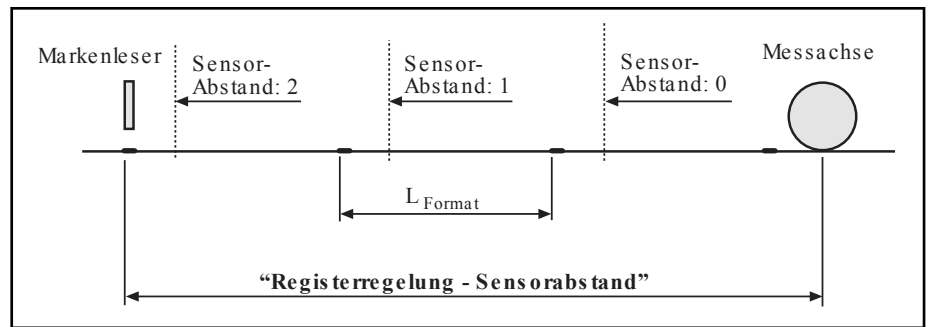


Abb. 5-9: Eingang „ProbeDistance“ setzen



Bei Verwendung der ProbeDistance wird vorausgesetzt, dass die zuvor ins Schieberegister eingetragenen Abweichungen jeweils zu 100% ausgegelt werden ($P\_Control = 1$ / $I\_Control = 0$).

Das heißt: Als neue Abweichung wird jeweils nur das Delta zur Summe der bereits im Schieberegister enthaltenen Abweichungen eingetragen.

MissedMarkCnt

Das ist der Markenausfallüberwachungszähler der Messtasterfunktion wenn das Erwartungsfenster aktiv ist. Wenn eine Marke fehlt, wird eine Null in das Schieberegister geschrieben wenn es aktiviert ist. Im Falle eines deaktivierten Schieberegisters wird die Null in den Ausgang "Deviation" geschrieben.

MarkGating

Mit diesem Feature kann der Anwender die Registrierung nur bei jeder n-ten Marke regeln, indem er ungültige Marken filtert. Man kann dies z. B. verwenden, wenn die Marken auf der Papierbahn mit einer Druckplatte gedruckt wurden, auf dem das gedruckte Bild mehrfach erscheint. Dabei sind die Markenabstände zwischen aufeinanderfolgenden Marken nicht konstant. Die Markenabstände zwischen jeder n-ten aufeinanderfolgenden Marke sind jedoch konstant. Der RegiMeasuring-FB verwendet nur jede n-te Marke für die Registerregelung.

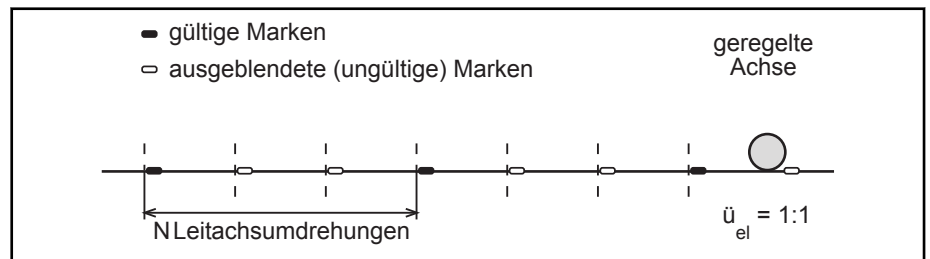


Abb. 5-10: Markenausfensterung

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse ("Axis_Ref") wurde geändert während der FB in Betrieb war

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0012	16#0003	"AXIS_REF.CntrlNo" liegt außerhalb des gültigen Bereiches (Achsen von entfernten Steuerungen sind nicht zulässig)
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0001	Eingang "ProbeDistance" ist außerhalb des gültigen Wertebereichs ($0 \leq \text{"ProbeDistance"} \leq 30$)
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0002	Eingang "Setpoint" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0003	Eingang "Setpoint" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0004	Eingang "Feedback" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0005	Eingang "Feedback" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0006	Eingang "ModuloValue" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0007	Eingang "MarkGating" < 0

Tab. 5-13: Fehlercodes Funktionsbaustein MB_RegiMeasuringType03

5.3 Registerqualitäts-FB

5.3.1 Einführung und Übersicht

Der Registerqualitäts-FB ist in den Bibliotheken ML_TechRegi (MLC) und MX_TechRegi.lib (MLD) enthalten.

5.3.2 IL_RegiQualityType01

Kurzbeschreibung Dieser Funktionsbaustein, der eventgesteuert aufgerufen wird, wertet bis zu 16 Registerabweichungen aus. Anhand von zwei Grenzwerten werden die Druckbilder in "gute/mittlere/schlechte" Qualität eingeteilt und aufsummiert. Entscheidend für das Druckbild ist hierbei der schlechteste Wert einer Farbmarke.

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechRegi
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechRegi.lib

Tab. 5-14: Referenztable IL_RegiQualityType01

Schnittstellenbeschreibung

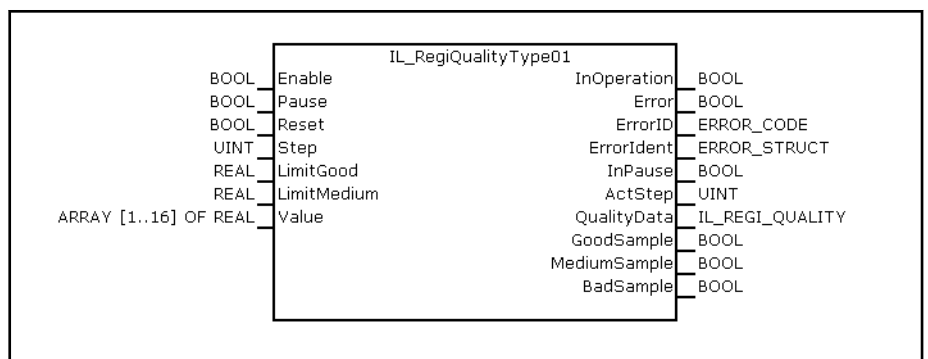


Abb. 5-11: Funktionsbaustein IL_RegiQualityType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Berechnung der Daten in jedem Zyklus solange Enable = TRUE
	Pause	BOOL	Die Daten der Ausgänge werden nicht verändert, solange Pause = TRUE
	Reset	BOOL	Reset des FB
	Step	UINT	Aktueller Berechnungsschritt
	LimitGood	REAL	Absolutwert [mm], den eine Registerabweichung maximal haben kann, damit das Druckbild noch mit "guter" Qualität gezählt wird
	LimitedMedium	REAL	Liegt der absolute Wert einer Registerabweichung [mm] zwischen "LimitGood" und "LimitMedium", zählt das Druckbild zu "mittlerer" Qualität; Registerabweichungen größer als "LimitMedium" haben "schlechte" Qualität
	Value	ARRAY [1..16] OF REAL	Enthält die Registerabweichungen eines Druckbildes (bis zu 16 Farbmarken)
VAR_OUTPUT	InOperation	BOOL	Fehlerfreie Bearbeitung, Ausgangsdaten sind gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorIdent)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlercodes, Seite 321
	InPause	BOOL	Zeigt an, dass der Eingang "Pause" aktiv ist und sich die Werte an den Ausgängen nicht ändern
	ActStep	UINT	Letzter verarbeiteter Schritt
	QualityData	IL_REGI_QUALITY	Enthält die berechneten Qualitätsdaten zum Druckprozess IL_REGI_QUALITY, Seite 321
	GoodSample	BOOL	Ist TRUE, wenn das aktuelle Druckbild "gute" Qualität hat

E/A-Typ	Name	Datentyp	Beschreibung
	MediumSample	BOOL	Ist TRUE, wenn das aktuelle Druckbild "mittlere" Qualität hat
	BadSample	BOOL	Ist TRUE, wenn das aktuelle Druckbild "schlechte" Qualität hat

Tab. 5-15: Schnittstellenvariablen Funktionsbaustein IL_RegiQualityType01

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
Pause	BOOL			FALSE	kontinuierlich
Reset	BOOL			FALSE	kontinuierlich
Step	UINT	0	65535	0	kontinuierlich
LimitGood	REAL	0.0	LimitMedium	1.0	steigende Flanke an "Enable"
LimitMedium	REAL	LimitGood	n. def.	2.0	steigende Flanke an "Enable"
Value	ARRAY [1..16] OF REAL	n. def.	n. def.	0.0	bei Änderung an "Step"

Tab. 5-16: Min.- Max.- und Default-Werte der Eingänge IL_RegiQualityType01

Funktionsbeschreibung

Um die Qualität eines Druckprozesses zu beurteilen, ist ein wichtiges Merkmal die Registerfehler der Druckbilder auszuwerten. Bei einem Druckbild ist der (betragsmäßig) größte Registerfehler einer Farbmarke entscheidend für die Qualität.

Über die Eingänge "LimitGood" und "LimitMedium" lassen sich entsprechende Fenster parametrieren, die die Beurteilung eines Druckbildes abhängig von der größten Registerabweichung in Value in "gute", "mittlere" und "schlechte" Qualität vornehmen. Wird für "LimitGood" und "LimitMedium" der gleiche Wert angegeben, findet nur die Einteilung in "good" und "bad" statt (siehe nachfolgende Abbildung).

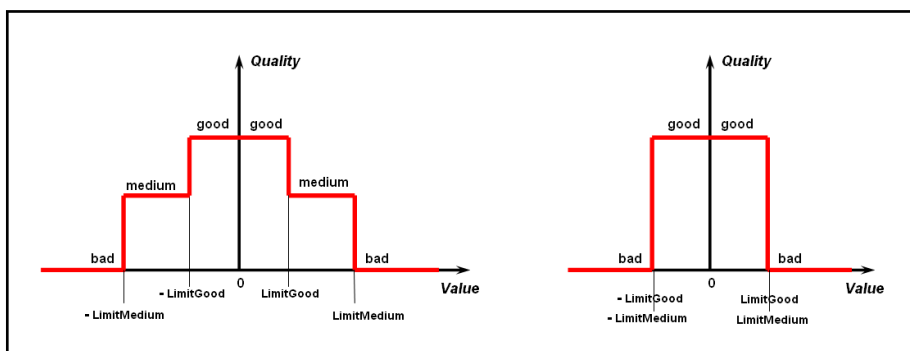


Abb. 5-12: Prinzip der Qualitätsbeurteilung bei IL_RegiQualityType01

In der Datenstruktur am Ausgang "QualityData" werden die Druckbilder der drei Qualitätskategorien aufsummiert, sowie deren Anteil an der Gesamtstückzahl ermittelt. Mit einer steigenden Flanke am Eingang „Reset“ werden alle Werte auf '0' gesetzt.

Die Abarbeitung erfolgt kontinuierlich bei gesetztem **Enable**-Eingang. Der Ausgang "ActStep" zeigt an, dass ein neues Datenpaket mit Messwerten er-

halten und ausgewertet wurde. Die booleschen Ausgänge "Good-/Medium-/BadSample" zeigen zusätzlich an, welche Qualität das aktuelle Druckbild hat. Ist der Eingang "Pause" gesetzt, so wird der aktuelle Zustand eingefroren, der Ausgang "InPause" wird gesetzt und es findet keine weitere Auswertung statt.

Fehlercodes Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_INVALID_ERROR	16#0817	16#0001	Wert von "LimitGood" < 0.0
INPUT_INVALID_ERROR	16#0817	16#0002	Wert von "LimitMedium" < "LimitGood"

Tab. 5-17: Fehlercodes Funktionsbaustein IL_RegiQualityType01

5.3.3 IL_REGI_QUALITY

Die Struktur "IL_REGI_QUALITY" erlaubt es die Anzahl und den prozentualen Anteil guter, mittlerer und schlechter Qualität zu erfassen.

Die Struktur unterstützt den Funktionsbaustein [IL_RegiQualityType01](#), Seite 318.

Name	Typ	Beschreibung
GoodSampleCount	UDINT	Anzahl der Druckbilder mit "guter" Qualität
MediumSampleCount	UDINT	Anzahl der Druckbilder mit "mittlerer" Qualität
BadSampleCount	UDINT	Anzahl der Druckbilder mit "schlechter" Qualität
GoodSampleRatio	REAL	Prozentualer Anteil der Druckbilder "guter" Qualität
MediumSampleRatio	REAL	Prozentualer Anteil der Druckbilder "mittlerer" Qualität
BadSampleRatio	REAL	Prozentualer Anteil der Druckbilder "schlechter" Qualität

Tab. 5-18: Definition der Struktur IL_REGI_QUALITY

5.4 Stellfunktionsbausteine für Registerregler

5.4.1 Einführung und Übersicht

Dieser Abschnitt beschreibt die Funktionsbausteine zum Stellwert an den Antrieb zu schreiben, die dazu eingesetzt werden können kundenspezifische Registerregler-FB, herzustellen.

5.4.2 MB_RegiRegulateType01

Kurzbeschreibung Der Funktionsbaustein MB_RegiRegulateType01 wird verwendet, um einen Stellwert an den Antrieb zu schreiben.

Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechRegi
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechRegi.lib

Tab. 5-19: Referenztable MB_RegiRegulateType01

Schnittstellenbeschreibung

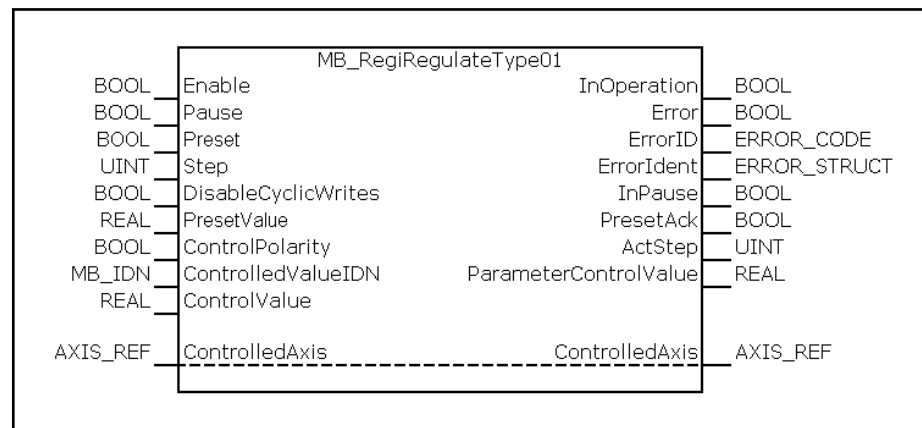


Abb. 5-13: Funktionsbaustein MB_RegiRegulateType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt "ParameterControlValue" zwangsweise auf "PresetValue"
	Step	UINT	Verarbeitungsschritt
	DisableCyclicWrites	BOOL	Deaktiviert zyklisches Schreiben auf ausgewählten Parameter in "ControlledValueIDN" wenn TRUE
	PresetValue	REAL	Wert an den Ausgang "ParameterControlValue" bei steigender Flanke des "Preset"-Eingangs übergeben
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist
	ControlledValueIDN	MB_IDN	sercos IDN des geregelten Parameters (z.B. P-0-0691). Die folgenden IDNs werden unterstützt: MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 MLC: A-0-2730, P-0-0690, A-0-2615, P-0-0691, A-0-2610, P-0-0692, A-0-2600, P-0-0694, A-0-2605, P-0-0695, A-0-2620
	ControlValue	REAL	Berechneter Stellwert von z.B. IL_PIType02
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein ist in Betrieb

E/A-Typ	Name	Datentyp	Beschreibung
	Error	BOOL	Zeigt einen Fehler an. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 325
	InPause	BOOL	"Pause" ist aktiv. Der Ausgang "ParameterControlValue" behält seinen Wert
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in den Ausgang "ParameterControlValue" kopiert
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "ParameterControlValue" hoch
	ParameterControlValue	REAL	Der Registerregler-FB berechnet einen Wert, der auf dem ausgewählten Parameter im "ControlledValueIDN"-Eingang basiert und speichert das Ergebnis im Ausgang "ParameterControlValue". Dieser Wert wird in den ausgewählten Parameter, Eingang "ControlledValueIDN", geschrieben, wenn der Eingang "DisableCyclicWrites" = FALSE ist
VAR_IN_OUT CONSTANT	ControlledAxis	AXIS_REF	Bezug zur geregelten Achse

Tab. 5-20: Schnittstellenvariablen Funktionsbaustein MB_RegiRegulateType01

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
ControlValue	REAL	n.def	n.def	0.0	Kontinuierlich
ModuloValue	REAL	0.0	n.def	0.0	Steigende Flanke an "Enable"

Tab. 5-21: Übersicht über Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Zeitdiagramm gemäß der Spezifikation PLCopen.

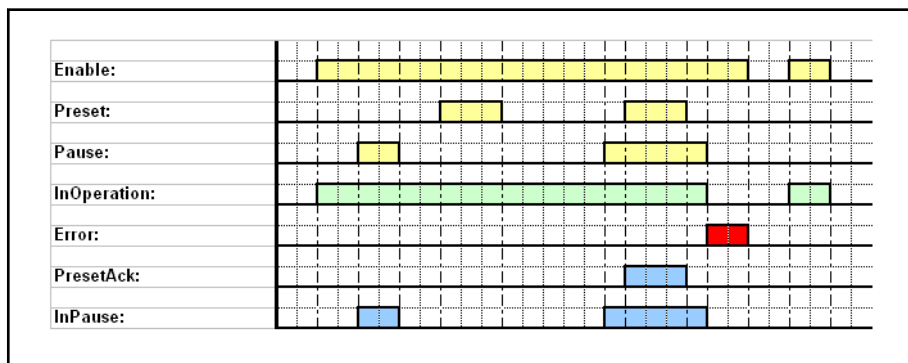


Abb. 5-14: Zeitdiagramm des MB_RegiRegulateType01

Funktionsbeschreibung

Der Funktionsbaustein wird verwendet, um Stellwerte auf einen gewünschten Parameter in einer Achse zu schreiben. Je nach dem gewünschten Zielparame-ter, "ControlledValueIDN", umfasst der Funktionsbaustein eine Integration des Eingangs "ParameterControlValue". Die folgenden Werte von "Control-
ledValueIDN" werden integriert:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Ein-gang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und der "ParameterControlValue"-Ausgang behält seinen Wert.

Preset Der Ausgang "PresetAck" ist aktiv wenn der Wert an "PresetValue" berück-sichtigt wird. Der Eingang "Preset" hat Priorität über den "Pause"-Eingang. Der Wert in "PresetValue" wird bei einer ansteigenden Flanke von "Preset" berücksichtigt wenn der Eingang "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt und der Wert in "PresetVa-lue" an den geregelten Parameter geschrieben.

DisableCyclicWrites Deaktiviert zyklisches Schreiben auf den geregelten Parameter (Eingang "ControlledValueIDN") vom FB.

Normalbetrieb Der RegiRegulateType berechnet nur dann einen neuen Wert wenn sich der "Step"-Eingang ändert. Die geregelte Achse muss eine Modulo-Achse sein.

Notwendige Parametrierung

MLC-Parametrierung

Wird eine **reale Achse mit Interpolation im Antrieb** als geregelte Achse ("Con-trolledAxis"-Eingang) benutzt, muss der verwendete Parameter im "Control-ledValueIDN"-Eingang einer der nachfolgenden unterstützten P-Parameter sein:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 und P-0-0695



Zusätzlich muss dieser P-Parameter im MDT des zyklischen Da-tenkanals im Antrieb konfiguriert sein.

Wird eine **reale Achse mit Interpolation in der Steuerung** oder eine **virtuelle Achse** als geregelte Achse benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten A-Pa-rameter sein:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

MLD-Parametrierung

Der Parameter, der als geregelter Parameter für den "ControlledValueIDN"-Eingang (P-Parameter) verwendet wird, muss in der "AxisData"-Struktur konfiguriert werden. Daher muss die Achsdatenstruktur aktiviert werden. Folgende P-Parameter werden unterstützt:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

Ab MLD Firmware MPx20 (IL2G) werden am Eingang "MeasuredAxis" auch Achsreferenzen unterstützt, die nicht in der "AxisData"-Struktur enthalten sind (> VMA1).

In diesem Fall wird das Korrekturfenster nicht über die aktuelle Position der "MeasuredAxis" ermittelt, sondern über P-0-0761 "Leitachseposition für Folgeachse" der lokalen Achse gebildet.

Damit im P-0-0761 "Leitachseposition für Folgeachse" die benötigte Positionsinformation zur Verfügung steht, muss der Leitachsgenerator entsprechend konfiguriert werden.

Beteiligte Parameter:

- P-0-0760 "Virtuelle Leitachse, Positionssteuerwort"
- P-0-0916 "Leitachsformatwandler Signalauswahl"

Siehe "Virtueller Leitachsgenerator" in der Dokumentation IndraDrive MPx-20 Funktionen Anwendungsbeschreibung, DOK-INDRV*-MP*-20VRS**-APRS-DE-P.

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

INPUT_RANGE_ERROR (16#0006)

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse ("AXIS_REF") wurde geändert während der FB in Betrieb war
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"AXIS_REF" ist außerhalb des gültigen Wertebereichs
RESOURCE_ERROR (16#0003)	16#0012	16#0003	"AXIS_REF.CntrlNo" liegt außerhalb des gültigen Bereiches (Achsen von entfernten Steuerungen sind nicht zulässig)
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000A	S-0-0103 oder A-0-0045 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000B	S-0-0091 oder A-0-0032 der geregelten Achse außerhalb des gültigen Wertebereichs

INPUT_RANGE_ERROR (16#0006)	16#0811	16#000C	P-0-0750 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0001	Ausgewählte "ControlledValueIDN" wird nicht unterstützt
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00A0	Der Typ des zu regelnden Parameters "ControlledValueIDN" passt nicht zum Typ der geregelten Achse (A-Parameter für Achse mit Interpolation im Antrieb verwendet)
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00B0	Der Typ des zu regelnden Parameters "ControlledValueIDN" passt nicht zum Typ der geregelten Achse (P-Parameter für Achse mit Interpolation in der Steuerung verwendet)
ACCESS_ERROR (16#0003)	16#0815	16#0002	Benötigter Parameter P-0-0061 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0003	Benötigter Parameter P-0-0690 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0004	Benötigter Parameter P-0-0691 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0005	Benötigter Parameter P-0-0692 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0006	Benötigter Parameter P-0-0694 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0007	Benötigter Parameter P-0-0695 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert

Tab. 5-22: Fehlercodes Funktionsbaustein MB_RegiRegulateType01

5.4.3 MB_RegiRegulateType02

Kurzbeschreibung Der Funktionsbaustein MB_RegiRegulateType02 wird verwendet, um einen Stellwert in den Antrieb zu schreiben. Wenn das Korrekturfenster aktiv ist, werden die Abweichungen der Regelung innerhalb eines parametrierten Korrekturfensters korrigiert.

Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechRegi

MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechRegi.lib (IL1G)
MLD mit MPx20 mit MA Funktionspaket	MX_TechRegi (IL2G)

Tab. 5-23: Referenztabelle MB_RegiRegulateType02

Schnittstellenbeschreibung

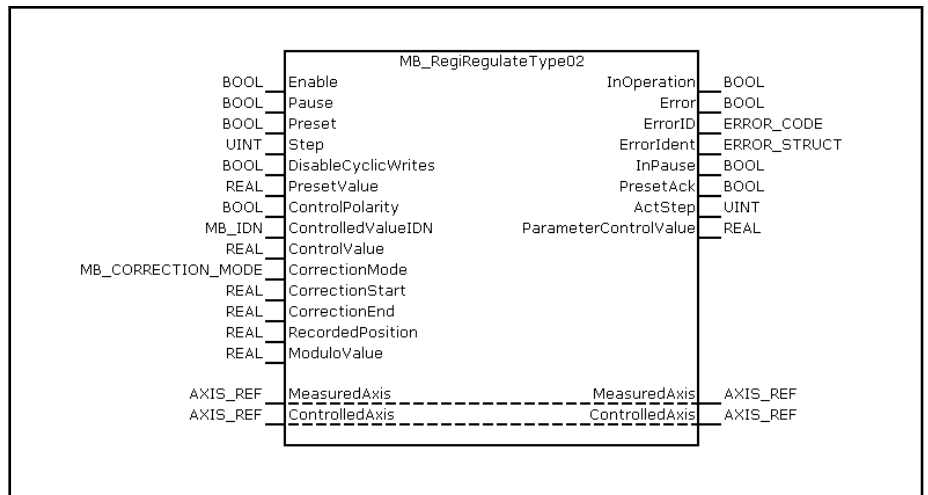


Abb. 5-15: Funktionsbaustein MB_RegiRegulateType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt "ParameterControlValue" zwangsweise auf "PresetValue"
	Step	UINT	Verarbeitungsschritt
	DisableCyclicWrites	BOOL	Deaktiviert zyklisches Schreiben auf ausgewählten Parameter in "ControlledValueIDN" wenn TRUE
	PresetValue	REAL	Wert an den Ausgang "ParameterControlValue" bei steigender Flanke des "Preset"-Eingangs übergeben
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist
	ControlledValueIDN	MB_IDN	sercos IDN des geregelten Parameters (z.B. P-0-0691). Die folgenden IDNs werden unterstützt: MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 MLC: A-0-2730, P-0-0690, A-0-2615, P-0-0691, A-0-2610, P-0-0692, A-0-2600, P-0-0694, A-0-2605, P-0-0695, A-0-2620
	ControlValue	REAL	Berechneter Stellwert von z.B. IL_PIType02
	CorrectionMode	MB_CORRECTION_MODE	Eigenschaften des Korrekturfensters MB_CORRECTION_MODE, Seite 349

E/A-Typ	Name	Datentyp	Beschreibung
	CorrectionStart	REAL	Anfang des Korrekturfensters
	CorrectionEnd	REAL	Ende des Korrekturfensters
	RecordedPosition	REAL	Gemessener Wert
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt einen Fehler an. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 333
	InPause	BOOL	"Pause" ist aktiv. Der Ausgang "ParameterControlValue" behält seinen Wert
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in den Ausgang "ParameterControlValue" kopiert
	ActStep	UINT	Schrittweite; zählt mit jedem neuen Wert am Ausgang "ParameterControlValue" hoch
	ParameterControlValue	REAL	Der Registerregler-FB berechnet einen Wert, der auf dem ausgewählten Parameter im "ControlledValueIDN"-Eingang basiert und speichert das Ergebnis im Ausgang "ParameterControlValue". Dieser Wert wird in den ausgewählten Parametereingang "ControlledValueIDN" geschrieben wenn der Eingang "DisableCyclicWrites" = FALSE ist
VAR_IN_OUT CONSTANT	MeasuredAxis	AXIS_REF	Bezug zur gemessenen Achse
	ControlledAxis	AXIS_REF	Bezug zur geregelten Achse

Tab. 5-24: Schnittstellenvariablen Funktionsbaustein MB_RegiRegulateType02

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
ControlValue	REAL	n.def	n.def	0.0	Kontinuierlich
CorrectionMode	MB_CORRECTION_MODE	0	4	CORR_REL	Kontinuierlich
CorrectionStart	REAL	0.0	ModuloValue	0.0	Kontinuierlich
CorrectionEnd	REAL	0.0	ModuloValue	0.0	Kontinuierlich
RecordedPosition	REAL	0.0	ModuloValue	0.0	Änderung des "Step"-Eingangs
ModuloValue	REAL	0.0	n.def	0.0	Steigende Flanke an "Enable"

Tab. 5-25: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm

Zeitdiagramm gemäß der Spezifikation PLCopen.

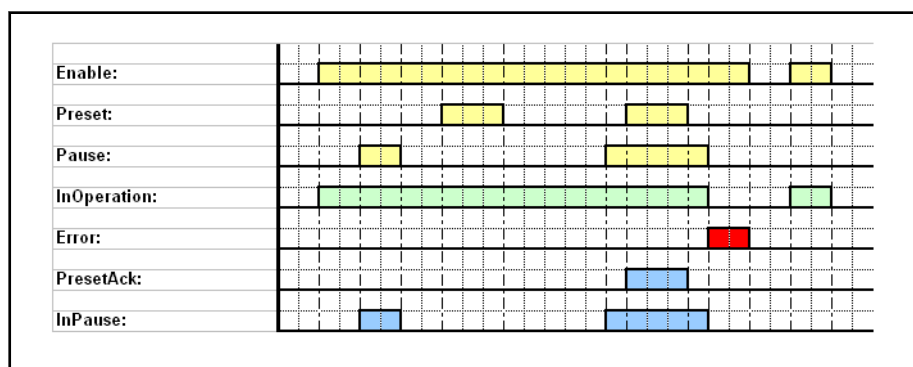


Abb. 5-16: Zeitdiagramm des MB_RegiRegulateType02

Funktionsbeschreibung

Der Funktionsbaustein wird verwendet, um Stellwerte auf einen gewünschten Parameter in einer Achse zu schreiben. Je nach dem gewünschten Zielparame-ter, "ControlledValueIDN", umfasst der Funktionsbaustein eine Integration des Eingangs "ParameterControlValue". Die folgenden Werte von "ControlledValueIDN" werden integriert:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und der "ParameterControlValue"-Ausgang behält seinen Wert.

Preset Der Ausgang "PresetAck" ist aktiv wenn der Wert an "PresetValue" berücksichtigt wird. Der Eingang "Preset" hat Priorität über den "Pause"-Eingang. Der Wert in "PresetValue" wird bei einer ansteigenden Flanke von "Preset" berücksichtigt wenn der Eingang "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt und der Wert in "PresetValue" an den geregelten Parameter geschrieben.

DisableCyclicWrites Deaktiviert zyklisches Schreiben auf den geregelten Parameter (Eingang "ControlledValueIDN") vom FB.

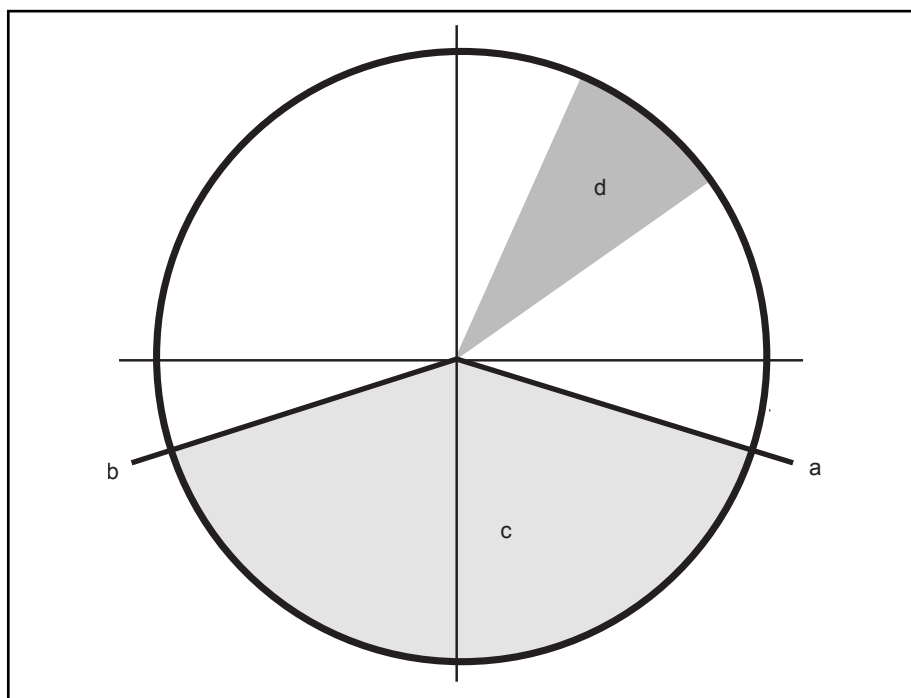
Korrekturfenster Das Korrekturfenster ist ein auf die Messachse bezogener Winkelbereich. Der Registerregler regelt die Regelungsabweichungen innerhalb dieses Fensters. Die Lage und Größe des Korrekturfensters werden mithilfe der folgenden Eingänge gesetzt:

Der Eingang "CorrectionMode" stellt das Korrekturfenster entweder auf absolut oder relativ ein. Wenn "CorrectionMode" auf CORR_WIN_DISABLED gesetzt wird, ist das Korrekturfenster inaktiv.

Der Eingang "CorrectionStart" legt den Anfang des Korrekturfensters fest.

Der Eingang "CorrectionEnd" legt das Ende des Korrekturfensters fest.

Das Korrekturfenster unterstützt nur einen positiven Modulo-Bereich (z.B. 0...+360 Grad) der Messachse.



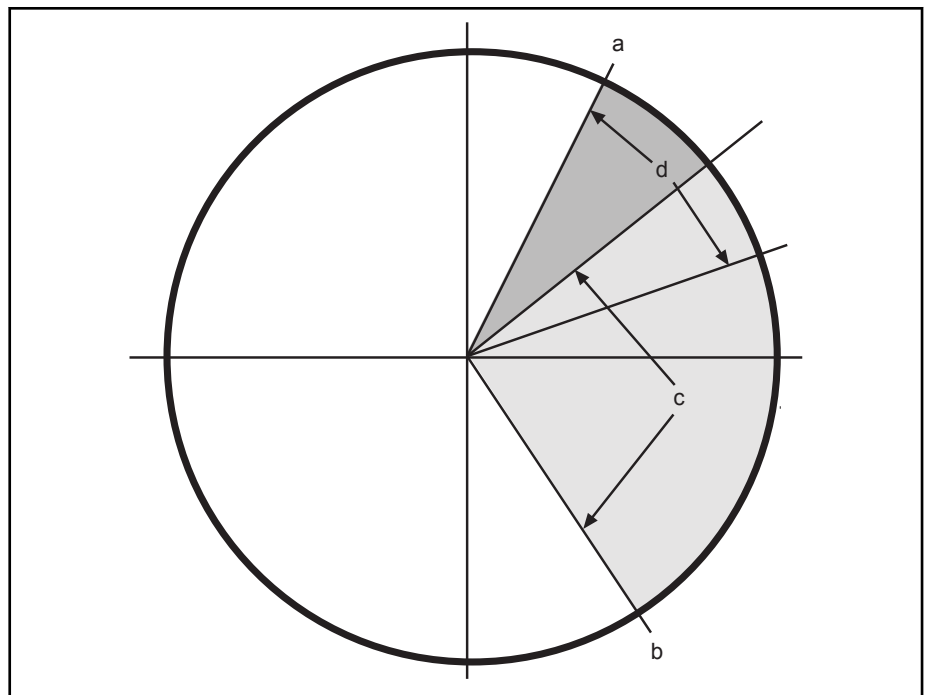
a: Beginn Korrekturfenster

b: Ende Korrekturfenster

c: Korrekturfenster

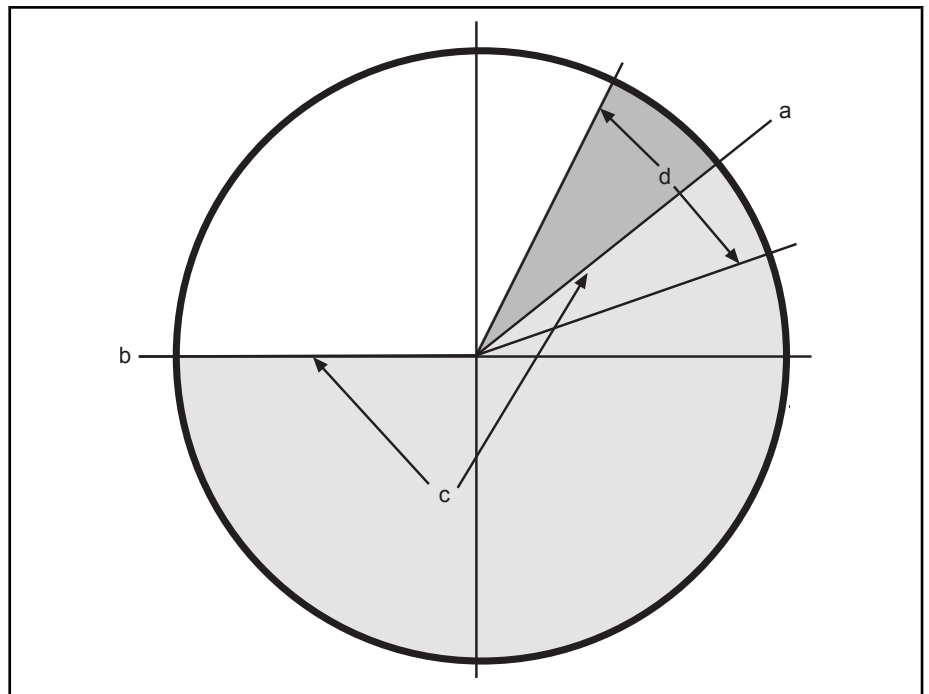
d: Erwartungsfenster

Abb. 5-17: Korrekturfenster absolut



- a:** Anfang Korrekturfenster = Messende
b: Relatives Ende des Korrekturfensters bezogen auf das Messende
c: Korrekturfenster
d: Erwartungsfenster

Abb. 5-18: *Korrekturfenster relativ*



- a:** Anfang Korrekturfenster = Messende
b: Ende des Korrekturfensters = perm. Winkel
c: Korrekturfenster
d: Erwartungsfenster

Abb. 5-19: *Korrekturfenster, relativer Start und absolutes Ende*

Normalbetrieb Der Funktionsbaustein RegiRegulateType02 berechnet nur einen neuen Wert, wenn sich der Eingang "Step" ändert. Die geregelte und gemessene Achse müssen Modulo-Achsen sein.

Notwendige Parametrierung

MLC-Parametrierung

Wird eine **reale Achse mit Interpolation im Antrieb** als geregelte Achse ("ControlledAxis"-Eingang) benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten P-Parameter sein:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 und P-0-0695



Zusätzlich muss dieser P-Parameter im MDT des zyklischen Datenkanals im Antrieb konfiguriert sein.

Wird eine **reale Achse mit Interpolation in der Steuerung** oder eine **virtuelle Achse** als geregelte Achse benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten A-Parameter sein:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

MLD-Parametrierung

Der Parameter, der als geregelter Parameter für den "ControlledValueIDN"-Eingang (P-Parameter) verwendet wird, muss in der "AxisData"-Struktur konfiguriert werden. Daher muss die Achsdatenstruktur aktiviert werden. Folgende P-Parameter werden unterstützt:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

Ab MLD Firmware MPx20 (IL2G) werden am Eingang "MeasuredAxis" auch Achsreferenzen unterstützt, die nicht in der "AxisData"-Struktur enthalten sind (> VMA1).

In diesem Fall wird das Korrekturfenster nicht über die aktuelle Position der "MeasuredAxis" ermittelt, sondern über P-0-0761 "Leitachspannung für Folgeachse" der lokalen Achse gebildet.

Damit im P-0-0761 "Leitachspannung für Folgeachse" die benötigte Positionsinformation zur Verfügung steht, muss der Leitachsgenerator entsprechend konfiguriert werden.

Beteiligte Parameter:

- P-0-0760 "Virtuelle Leitachse, Positionssteuerwort"
- P-0-0916 "Leitachsformatwandler Signalauswahl"

Siehe "Virtueller Leitachsgenerator" in der Dokumentation IndraDrive MPx-20 Funktionen Anwendungsbeschreibung, DOK-INDRV*-MP*-20VRS*-APRS-DE-P.

Datentyp

MB_CORRECTION_MODE

Modus des Korrekturfensters.

Programm:

```
TYPE MB_CORRECTION_MODE:
  (CORR_WIN_DISABLED :=0, (* Korrekturfenster ist ausgeschaltet *)
   CORR_ABS          :=1, (* Korrekturfenster ist absolut *)
   CORR_REL          :=2, (* Korrekturfenster ist relativ *)
   CORR_REL_ABS      :=3, (* Korrekturfenster ist relativer Anfang
                           und absolutes Ende *)
   CORR_ABS_REL      :=4); (* Korrekturfenster ist absoluter Anfang
                           und relatives Ende *)
END_TYPE
```

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

INPUT_RANGE_ERROR (16#0006)

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse ("AXIS_REF") wurde geändert während der FB in Betrieb war
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"AXIS_REF" ist außerhalb des gültigen Wertebereichs
RESOURCE_ERROR (16#0003)	16#0012	16#0003	"AXIS_REF.CntrlNo" liegt außerhalb des gültigen Bereiches (Achsen von entfernten Steuerungen sind nicht zulässig)
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0001	Länge des Korrekturfensters < 0.25% des Werts am Eingang "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0002	Eingang "CorrectionStart" < 0
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0003	Eingang "CorrectionEnd" < 0
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0004	Wert < 0 Eingang "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0005	Wert < 0 Eingang "RecordedPosition"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0006	Eingang "CorrectionStart" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0007	Eingang "CorrectionEnd" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0008	Eingang "RecordedPosition" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0009	Korrekturbahn > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000A	S-0-0103 oder A-0-0045 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000B	S-0-0091 oder A-0-0032 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000C	P-0-0750 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0001	Ausgewählte "ControlledValueIDN" wird nicht unterstützt

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0002	Eingang "CorrectionMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0003	"CorrectionStart" = "CorrectionEnd"
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00A0	Der Typ des zu regelnden Parameters "ControlledValueIDN" passt nicht zum Typ der geregelten Achse (A-Parameter für Achse mit Interpolation im Antrieb verwendet)
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00B0	Der Typ des zu regelnden Parameters "ControlledValueIDN" passt nicht zum Typ der geregelten Achse (P-Parameter für Achse mit Interpolation in der Steuerung verwendet)
ACCESS_ERROR (16#0003)	16#0815	16#0002	Benötigter Parameter P-0-0061 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0003	Benötigter Parameter P-0-0690 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0004	Benötigter Parameter P-0-0691 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0005	Benötigter Parameter P-0-0692 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0006	Benötigter Parameter P-0-0694 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0007	Benötigter Parameter P-0-0695 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0009	Benötigter Parameter P-0-0753 ist nicht in den optionalen zyklischen AT-Daten der gemessenen Achse konfiguriert

Tab. 5-26: Fehlercodes Funktionsbaustein MB_RegiRegulateType02

5.4.4 MB_RegiRegulateType03

Kurzbeschreibung Der Funktionsbaustein MB_RegiRegulateType03 wird verwendet, um einen Stellwert mit Hilfe einer parametrisierten Korrekturrampe in den Antrieb zu schreiben.

Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechRegi
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechRegi.lib

Tab. 5-27: Referenztable MB_RegiRegulateType03

Schnittstellenbeschreibung

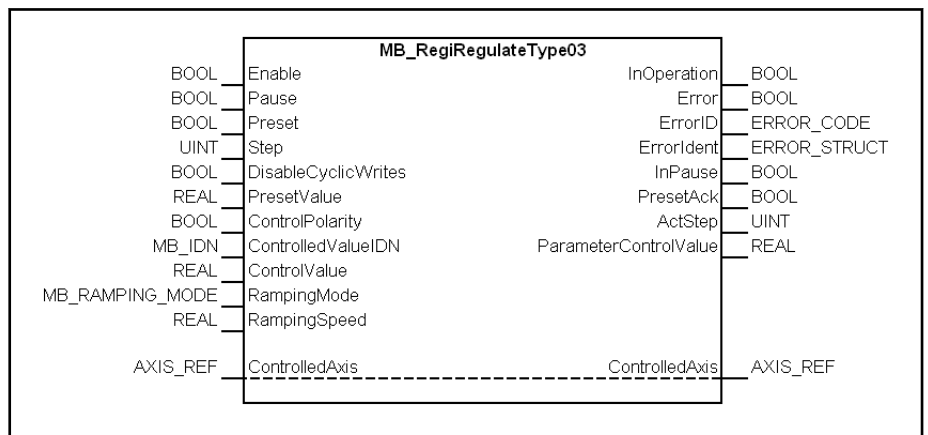


Abb. 5-20: Funktionsbaustein MB_RegiRegulateType03

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt "ParameterControlValue" zwangsweise auf "PresetValue"
	Step	UINT	Verarbeitungsschritt
	DisableCyclicWrites	BOOL	Deaktiviert zyklisches Schreiben auf ausgewählten Parameter in "ControlledValueIDN" wenn TRUE
	PresetValue	REAL	Wert an den Ausgang "ParameterControlValue" bei steigender Flanke des "Preset"-Eingangs übergeben
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist
	ControlledValueIDN	MB_IDN	sercos IDN des geregelten Parameters (z.B. P-0-0691). Die folgenden IDNs werden unterstützt: MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 MLC: A-0-2730, P-0-0690, A-0-2615, P-0-0691, A-0-2610, P-0-0692, A-0-2600, P-0-0694, A-0-2605, P-0-0695, A-0-2620
	ControlValue	REAL	Berechneter Stellwert von z.B. IL_PIType02

E/A-Typ	Name	Datentyp	Beschreibung
	RampingMode	MB_RAMPING_MODE	Vorgabe der Art der Verrampungsgeschwindigkeit MB_RAMPING_MODE, Seite 350
	RampingSpeed	REAL	Dieser Wert verwendet einen Verrampungsgradienten
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt einen Fehler an. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 339
	InPause	BOOL	"Pause" ist aktiv. Der Ausgang "ParameterControlValue" behält seinen Wert
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in den Ausgang "ParameterControlValue" kopiert
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "ParameterControlValue" hoch
	ParameterControlValue	REAL	Der Registerregler-FB berechnet einen Wert, der auf dem ausgewählten Parameter im "ControlledValueIDN"-Eingang basiert und speichert das Ergebnis im Ausgang "ParameterControlValue". Dieser Wert wird in den ausgewählten Parametereingang "ControlledValueIDN" geschrieben wenn der Eingang "DisableCyclicWrites" = FALSE ist
VAR_IN_OUT CONSTANT	ControlledAxis	AXIS_REF	Bezug zur geregelten Achse

Tab. 5-28: Schnittstellenvariablen Funktionsbaustein MB_RegiRegulateType03

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Baustein-
eingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ControlValue	REAL	n.def	n.def	0.0	Kontinuierlich
RampingMode	MB_RAM-PING_MODE	0.0	1	RMPING_CONS	Steigende Flanke an "Enable"
RampingSpeed	REAL	0.0	n.def	1.0	Kontinuierlich

Tab. 5-29: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Zeitdiagramm gemäß der Spezifikation PLCopen.

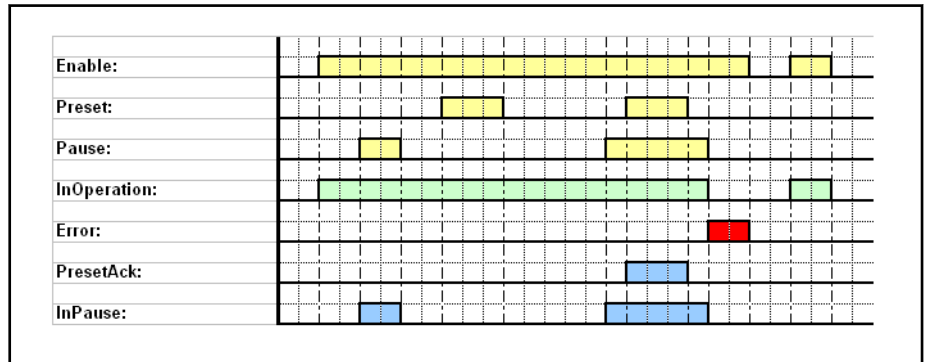


Abb. 5-21: Zeitdiagramm des MB_RegiRegulateType03

Funktionsbeschreibung

Der Funktionsbaustein wird verwendet, um Stellwerte auf einen gewünschten Parameter in einer Achse zu schreiben. Je nach dem gewünschten Parameter, "ControlValueIDN", umfasst der Funktionsbaustein die Integration des Eingangs "ParameterControlValue". Die folgenden Werte von "ControlValueIDN" werden integriert:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und der "ParameterControlValue"-Ausgang behält seinen Wert.

Preset Der Ausgang "PresetAck" ist aktiv wenn der Wert an "PresetValue" berücksichtigt wird. Der Eingang "Preset" hat Priorität über den "Pause"-Eingang. Der Wert in "PresetValue" wird bei einer ansteigenden Flanke von "Preset" berücksichtigt wenn der Eingang "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt und der Wert in "PresetValue" an den geregelten Parameter geschrieben.

DisableCyclicWrites Deaktiviert zyklisches Schreiben auf den geregelten Parameter (Eingang "ControlValueIDN") vom FB.

Verrampungsmodus Es gibt 2 verschiedene Verrampungsmodi. Ein Modus ist das konstante Verrampen mit dem Gradienten des "RampingSpeed" ("RAMPING_CONSTANT")-Eingangs. Der andere Modus hängt von der Änderung des "Step"-Eingangs ab. Je schneller sich der "Step"-Eingang ändert, desto steiler ist das Verrampen des "ParameterControlValue" ("RAMPING_ADAPTIVE").

- **RAMPING_CONSTANT:**
In diesem Modus hängt die Änderung am "ParameterControlValue"-Ausgang vom "RampingSpeed"-Eingang ab. Ist der "RampingSpeed"-

Eingang beispielsweise auf 1 gesetzt, dann ist das Verrampen des "ParameterControlValue"-Ausgangs 1 Grad / Sekunde für den Parameter P-0-0691.

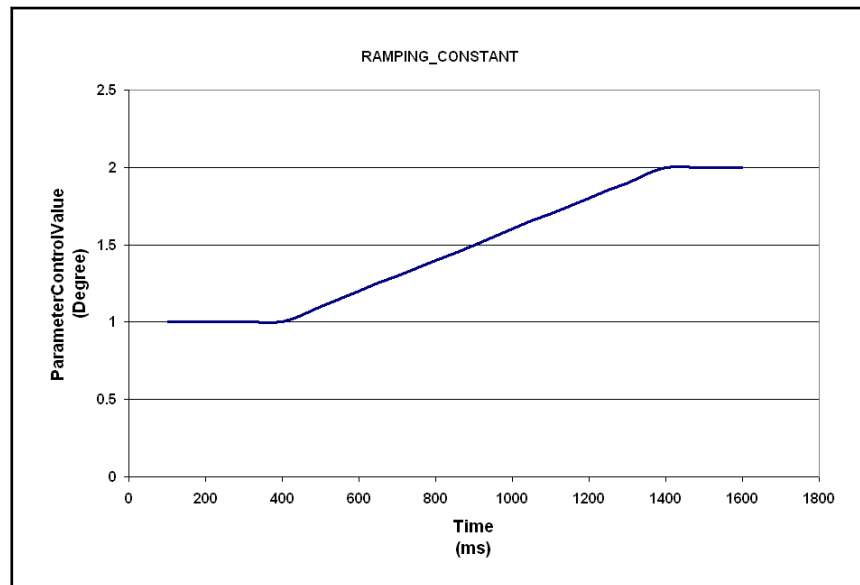


Abb. 5-22: Verrampungskonstante

- **RAMPING_ADAPTIVE:**

In diesem Modus hängt die Änderung des "ParameterControlValue" von der Zeitdifferenz der Änderungen ein "Step"-Eingangs ab. Beispielsweise verursacht eine Änderung nach jeden 0.6 (0.3) Sekunden am "Step"-Eingang, eine Verrampung des "ParameterControlValue"-Ausgang von 1(2) Grad / Sekunden für den Parameter P-0-0691.

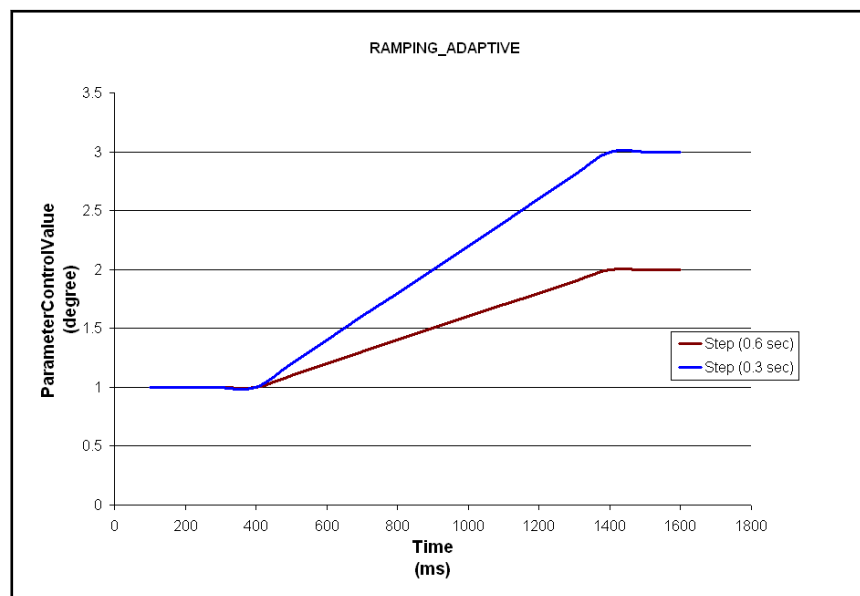


Abb. 5-23: Adaptive Verrampung

Verrampungsgeschwindigkeit	Die Verrampungsgeschwindigkeit ist nur in Verbindung mit dem Verrampungsmodus RAMPING_CONSTANT gültig.
Normalbetrieb	Der RegiRegulateType berechnet nur einen neuen Wert wenn sich der "Step"-Eingang ändert. Die geregelte Achse muss eine Modulo-Achse sein.
Notwendige Parametrierung	MLC-Parametrierung

Wird eine **reale Achse mit Interpolation im Antrieb** als geregelte Achse ("ControlledAxis"-Eingang) benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten P-Parameter sein:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 und P-0-0695



Zusätzlich muss dieser P-Parameter im MDT des zyklischen Datenkanals im Antrieb konfiguriert sein.

Wird eine **reale Achse mit Interpolation in der Steuerung** oder eine **virtuelle Achse** als geregelte Achse benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten A-Parameter sein:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

MLD-Parametrierung

Der Parameter, der als geregelter Parameter für den "ControlledValueIDN"-Eingang (P-Parameter) verwendet wird, muss in der "AxisData"-Struktur konfiguriert werden. Daher muss die Achsdatenstruktur aktiviert werden. Folgende P-Parameter werden unterstützt:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

Ab MLD Firmware MPx20 (IL2G) werden am Eingang "MeasuredAxis" auch Achsreferenzen unterstützt, die nicht in der "AxisData"-Struktur enthalten sind (> VMA1).

In diesem Fall wird das Korrekturfenster nicht über die aktuelle Position der "MeasuredAxis" ermittelt, sondern über P-0-0761 "Leitachspanposition für Folgeachse" der lokalen Achse gebildet.

Damit im P-0-0761 "Leitachspanposition für Folgeachse" die benötigte Positionsinformation zur Verfügung steht, muss der Leitachsgenerator entsprechend konfiguriert werden.

Beteiligte Parameter:

- P-0-0760 "Virtuelle Leitachse, Positionssteuerwort"
- P-0-0916 "Leitachsformatwandler Signalauswahl"

Siehe "Virtueller Leitachsgenerator" in der Dokumentation IndraDrive MPx-20 Funktionen Anwendungsbeschreibung, DOK-INDRV*-MP*-20VRS**-APRS-DE-P.

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

INPUT_RANGE_ERROR (16#0006)

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse ("AXIS_REF") wurde geändert während der FB in Betrieb war
RESOURCE_ERROR (16#0003)	16#0012	16#0003	"AXIS_REF.CntrlNo" liegt außerhalb des gültigen Bereiches (Achsen von entfernten Steuerungen sind nicht zulässig)
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000A	S-0-0103 oder A-0-0045 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000B	S-0-0091 oder A-0-0032 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000C	P-0-0750 der geregelten Achse außerhalb des gültigen Wertebereichs
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0001	Ausgewählte "ControlledValueIDN" wird nicht unterstützt
INPUT_INVALID_ERROR (16#0001)	16#0812	16#000C	Wert < 0 für den "RampingSpeed"-Eingang
INPUT_INVALID_ERROR (16#0001)	16#0812	16#000D	Modulowert der geregelten Achse konnte nicht gelesen werden oder ist 0
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00A0	Der Typ des zu regelnden Parameters "ControlledValueIDN" passt nicht zum Typ der geregelten Achse (A-Parameter für Achse mit Interpolation im Antrieb verwendet)
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00B0	Der Typ des zu regelnden Parameters "ControlledValueIDN" passt nicht zum Typ der geregelten Achse (P-Parameter für Achse mit Interpolation in der Steuerung verwendet)
ACCESS_ERROR (16#0004)	16#0815	16#0002	Benötigter Parameter P-0-0061 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0004)	16#0815	16#0003	Benötigter Parameter P-0-0690 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0004)	16#0815	16#0004	Benötigter Parameter P-0-0691 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0004)	16#0815	16#0005	Benötigter Parameter P-0-0692 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0004)	16#0815	16#0006	Benötigter Parameter P-0-0694 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert
ACCESS_ERROR (16#0004)	16#0815	16#0007	Benötigter Parameter P-0-0695 ist nicht in den optionalen zyklischen MDT-Daten der geregelten Achse konfiguriert

Tab. 5-30: Fehlercodes Funktionsbaustein MB_RegiRegulateType03

5.4.5 MB_RegiRegulateType04

Kurzbeschreibung Der Funktionsbaustein MB_RegiRegulateType04 wird verwendet, um einen Regelwert an bis zu acht Antriebe zu schreiben. Wenn das Korrekturfenster aktiv ist, werden die Abweichungen der Regelung innerhalb eines parametrisierten Korrekturfensters korrigiert.

Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechRegi

Tab. 5-31: Referenztabelle MB_RegiRegulateType04

Schnittstellenbeschreibung

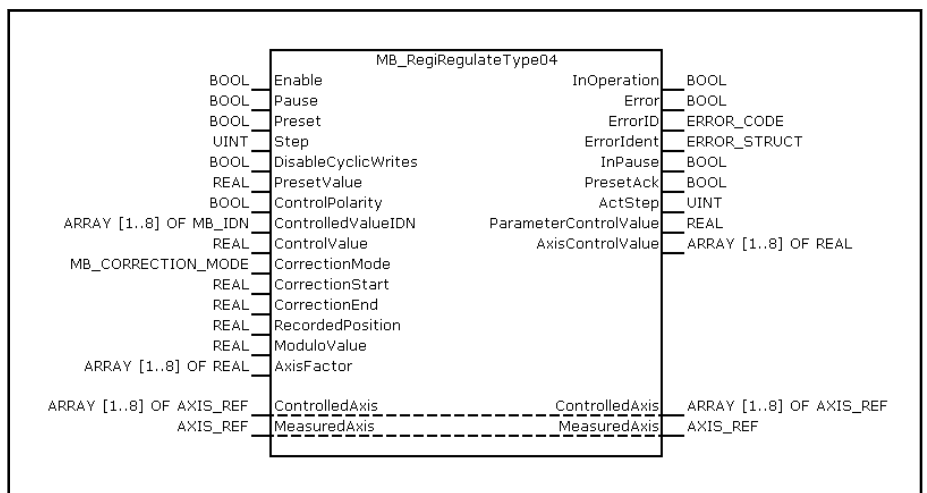


Abb. 5-24: Funktionsbaustein MB_RegiRegulateType04

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert den Funktionsbaustein, wenn er auf TRUE gesetzt ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist

E/A-Typ	Name	Datentyp	Beschreibung
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt "ParameterControlValue" zwangsweise auf "PresetValue"
	Step	UINT	Verarbeitungsschritt
	DisableCyclicWrites	BOOL	Deaktiviert zyklisches Schreiben auf ausgewählten Parameter in "ControlledValueIDN" wenn TRUE
	PresetValue	REAL	Wert an den Ausgang "ParameterControlValue" bei steigender Flanke des "Preset"-Eingangs übergeben
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist
	ControlledValueIDN	ARRAY [1..8] OF MB_IDN	sercos IDN des geregelten Parameters (z.B. P-0-0691). Die folgenden IDNs werden unterstützt: MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 MLC: A-0-2730, P-0-0690, A-0-2615, P-0-0691, A-0-2610, P-0-0692, A-0-2600, P-0-0694, A-0-2605, P-0-0695, A-0-2620
	ControlValue	REAL	Berechneter Stellwert von z.B. IL_PIType02
	CorrectionMode	MB_CORRECTION_MODE	Eigenschaften des Korrekturfensters MB_CORRECTION_MODE, Seite 349
	CorrectionStart	REAL	Anfang des Korrekturfensters
	CorrectionEnd	REAL	Ende des Korrekturfensters
	RecordedPosition	REAL	Gemessener Wert
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
	AxisFactor	ARRAY [1..8] OF REAL	Faktor zum Archivieren einer achsabhängigen Wichtung. Der Wert von "ParameterControlValue" wird mit diesem achsabhängigen Faktor (ARRAY) multipliziert und an jede geregelte Achse verteilt
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt einen Fehler an. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 347
	InPause	BOOL	"Pause" ist aktiv. Die Ausgänge "ParameterControlValue" und "AxisControlValue" behalten ihre Werte.
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in den Ausgang "ParameterControlValue" kopiert
	ActStep	UINT	Schrittwerk; zählt mit jedem neuen Wert am Ausgang "ParameterControlValue" hoch

E/A-Typ	Name	Datentyp	Beschreibung
	ParameterControlValue	REAL	Der Registerregler-FB berechnet einen Wert, der auf dem ausgewählten Parameter im "ControlledValueIDN"-Eingang basiert und speichert das Ergebnis im Ausgang "ParameterControlValue". Dieser Wert wird in den ausgewählten Parametereingang "ControlledValueIDN" geschrieben wenn der Eingang "DisableCyclicWrites" = FALSE ist
	AxisControlValue	ARRAY [1..8] OF REAL	Berechneter Wert des Registerregler-FB wird in den "ControlledValueIDN"-Eingang geschrieben. "AxisControlValue" ist der gewichtete "ParameterControlValue". Zur Diagnose oder weiteren Verarbeitung verwendet, wenn "DisableCyclicWrites" = TRUE
VAR_IN_OUT CONSTANT	ControlledAxis	ARRAY [1..8] OF AXIS_REF	Bezug auf die geregelten Achsen
	MeasuredAxis	AXIS_REF	Bezug zur gemessenen Achse

Tab. 5-32: Schnittstellenvariablen Funktionsbaustein MB_RegiRegulateType04

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	ARRAY [1..8] OF MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
ControlValue	REAL	n.def	n.def	0.0	Kontinuierlich
CorrectionMode	MB_CORRECTION_MODE	0	4	CORR_REL	Kontinuierlich
CorrectionStart	REAL	0.0	ModuloValue	0.0	Kontinuierlich
CorrectionEnd	REAL	0.0	ModuloValue	0.0	Kontinuierlich
RecordedPosition	REAL	0.0	ModuloValue	0.0	Änderung des "Step"-Eingangs

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ModuloValue	REAL	0.0	n.def	0.0	Steigende Flanke an "Enable"
AxisFactor	ARRAY [1..8] OF REAL	n.def	n.def	1.0	Kontinuierlich

Tab. 5-33: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Zeitdiagramm gemäß der Spezifikation PLCopen.

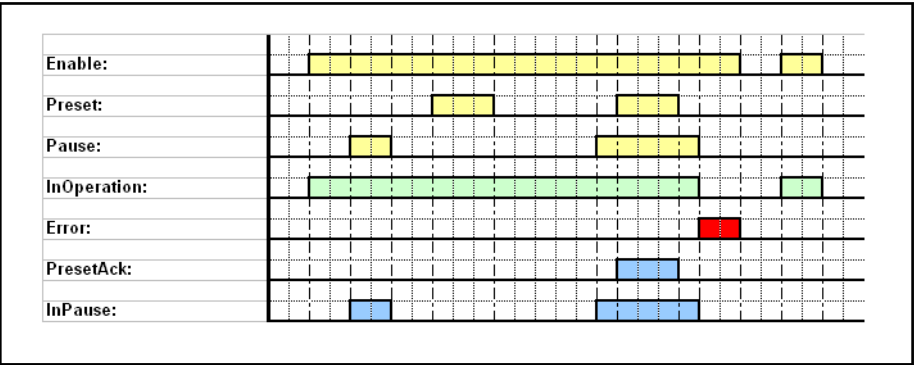


Abb. 5-25: Zeitdiagramm des MB_RegiRegulateType04

Funktionsbeschreibung

Der Funktionsbaustein wird verwendet, um Stellwerte auf einen gewünschten Parameter in einer Achse zu schreiben. Je nach dem gewünschten Parameter, "ControlledValueIDN", umfasst der Funktionsbaustein die Integration des Eingangs "ParameterControlValue". Für die folgenden Werte von ControlledValueIDN wird diese Integration vollzogen:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Während "Pause" aktiv ist, ist der "InPause"-Ausgang TRUE und die Ausgänge "ParameterControlValue" und "AxisControlValue" behalten ihre Werte.

Preset Der Ausgang "PresetAck" ist aktiv wenn der Wert an "PresetValue" berücksichtigt wird. Der Eingang "Preset" hat Priorität über den "Pause"-Eingang. Der Wert in "PresetValue" wird bei einer ansteigenden Flanke von "Preset" berücksichtigt wenn der Eingang "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt und der Wert in "PresetValue" an den geregelten Parameter geschrieben.

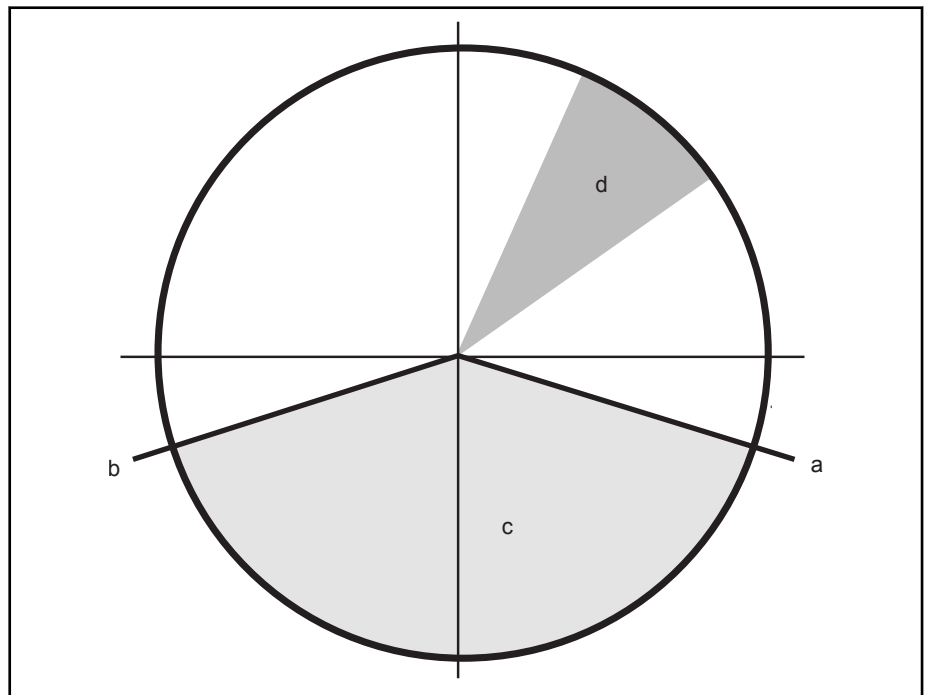
DisableCyclicWrites Deaktiviert zyklisches Schreiben auf den geregelten Parameter (Eingang "ControlledValueIDN") vom FB.

Korrekturfenster Das Korrekturfenster ist ein auf die Messachse bezogener Winkelbereich. Der Registerregler regelt die Regelungsabweichungen innerhalb dieses Fensters. Die Lage und Größe des Korrekturfensters werden mithilfe der folgenden Eingänge gesetzt:

Der Eingang "CorrectionMode" stellt das Korrekturfenster entweder auf absolut oder relativ ein. Wenn "CorrectionMode" auf CORR_WIN_DISABLED gesetzt wird, ist das Korrekturfenster inaktiv.

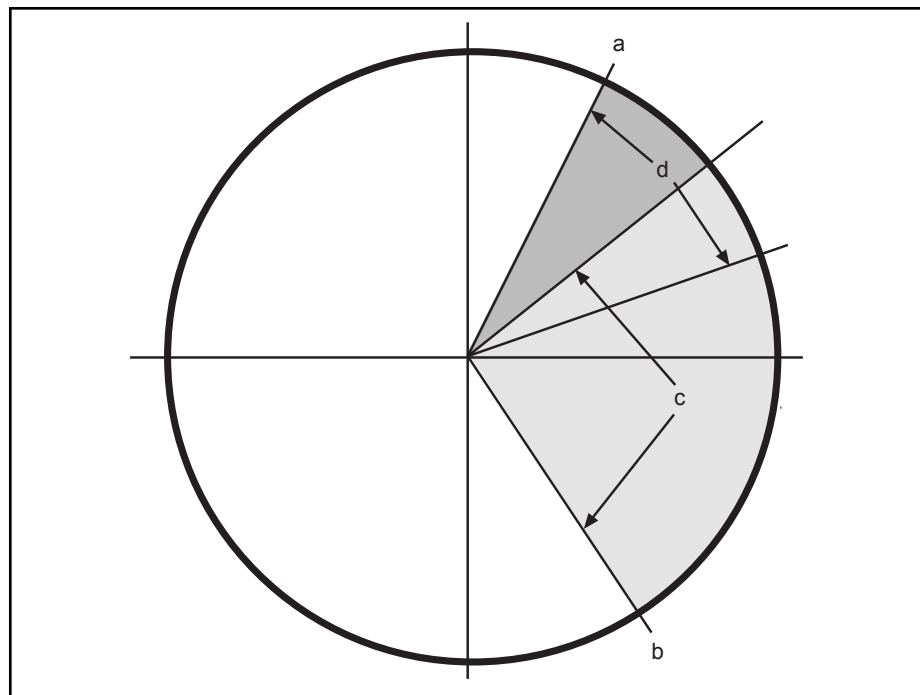
Der Eingang "CorrectionStart" legt den Anfang des Korrekturfensters fest.

Der Eingang "CorrectionEnd" legt das Ende des Korrekturfensters fest.
Das Korrekturfenster unterstützt nur einen positiven Modulo-Bereich (z.B. 0...+360 Grad) der Messachse.



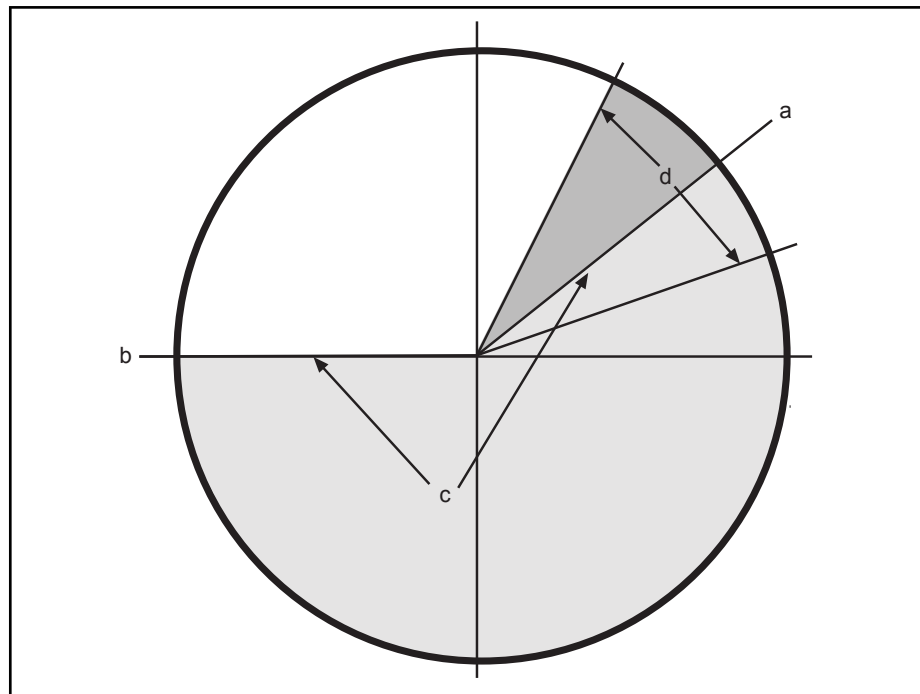
a: Beginn Korrekturfenster
b: Ende Korrekturfenster
c: Korrekturfenster
d: Erwartungsfenster

Abb. 5-26: Korrekturfenster absolut



- a:** Anfang Korrekturfenster = Messende
b: Relatives Ende des Korrekturfensters bezogen auf das Messende
c: Korrekturfenster
d: Erwartungsfenster

Abb. 5-27: *Korrekturfenster relativ*



- a:** Anfang Korrekturfenster = Messende
b: Ende des Korrekturfensters = perm. Winkel
c: Korrekturfenster
d: Erwartungsfenster

Abb. 5-28: *Korrekturfenster, relativer Start und absolutes Ende*

Mehrere Achsen	Die berechneten Stellwerte werden in die entsprechenden Parameter ("ControlValueIDN") auf bis zu 8 geregelte Achsen geschrieben. Die Achsen sind mit dem Eingang "ControlledAxis[x]" referenziert.
Individuelle Stellwertwichtung	Der Stellwert jeder einzelnen Achse kann mit dem Eingang "AxisFactor[x]" gewichtet werden.
Normalbetrieb	Der Funktionsbaustein RegiRegulateType04 berechnet nur einen neuen Wert wenn sich der Eingang "Step" ändert. Die geregelte und gemessene Achse müssen Modulo-Achsen sein.
Notwendige Parametrierung	MLC-Parametrierung Wird eine reale Achse mit Interpolation im Antrieb als geregelte Achse ("ControlledAxis"-Eingang) benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten P-Parameter sein: • P-0-0690, P-0-0691, P-0-0692, P-0-0694 und P-0-0695  Zusätzlich muss dieser P-Parameter im MDT des zyklischen Datenkanals im Antrieb konfiguriert sein. Wird eine reale Achse mit Interpolation in der Steuerung oder eine virtuelle Achse als geregelte Achse benutzt, muss der verwendete Parameter im "ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten A-Parameter sein: • A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620 MLD-Parametrierung Der Parameter, der als geregelter Parameter für den "ControlledValueIDN"-Eingang (P-Parameter) verwendet wird, muss in der "AxisData"-Struktur konfiguriert werden. Daher muss die Achsdatenstruktur aktiviert werden. Folgende P-Parameter werden unterstützt: • P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 Ab MLD Firmware MPx20 (IL2G) werden am Eingang "MeasuredAxis" auch Achsreferenzen unterstützt, die nicht in der "AxisData"-Struktur enthalten sind (> VMA1). In diesem Fall wird das Korrekturfenster nicht über die aktuelle Position der "MeasuredAxis" ermittelt, sondern über P-0-0761 "Leitachspannung für Folgeachse" der lokalen Achse gebildet. Damit im P-0-0761 "Leitachspannung für Folgeachse" die benötigte Positionsinformation zur Verfügung steht, muss der Leitachsgenerator entsprechend konfiguriert werden. Beteiligte Parameter: • P-0-0760 "Virtuelle Leitachse, Positionssteuerwort" • P-0-0916 "Leitachsformatwandler Signalauswahl" Siehe "Virtueller Leitachsgenerator" in der Dokumentation IndraDrive MPx-20 Funktionen Anwendungsbeschreibung, DOK-INDRV*-MP*-20VRS*-APRS-DE-P.
Fehlerbehandlung	Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen: INPUT_RANGE_ERROR (16#0006)

ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Antrieb ist nicht freigeschaltet bzw. Antriebsfehler
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Ausgewählte Achse (AXIS_REF) wurde geändert, während der FB in Betrieb ist
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AXIS_REF ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#0012	16#0003	"AXIS_REF.CntrlNo" liegt außerhalb des gültigen Bereiches (Achsen von entfernten Steuerungen sind nicht zulässig)
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0001	Länge des Korrekturfensters < 0.25% des Werts am Eingang "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0002	Eingang "CorrectionStart" < 0
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0003	Eingang "CorrectionEnd" < 0
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0004	Wert < 0 Eingang "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0005	Wert < 0 Eingang "RecordedPosition"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0006	Eingang "CorrectionStart" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0007	Eingang "CorrectionEnd" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0008	Eingang "RecordedPosition" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0009	Korrekturbahn > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0x10	S-0-0103 oder A-0-0045 der Controlled Axis[x] außerhalb des Gültigkeitsbereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0x20	S-0-0091 oder A-0-0032 der Controlled Axis[x] außerhalb des Gültigkeitsbereichs
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0x30	P-0-0750 der Controlled Axis[x] außerhalb des Gültigkeitsbereichs
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0x01	Gewählte "ControlledValueIDN[x]" wird nicht unterstützt
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0002	Eingang "CorrectionMode" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0003	"CorrectionStart" = "CorrectionEnd"

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0xA0	Der Typ des zu regelnden Parameters "ControlledValueIDN[x]" passt nicht zum Typ der geregelten Achse "Controlled Axis[x]" (A-Parameter für Achse mit Interpolation im Antrieb verwendet)
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0xB0	Der Typ des zu regelnden Parameters "ControlledValueIDN[x]" passt nicht zum Typ der geregelten Achse "Controlled Axis[x]" (P-Parameter für Achse mit Interpolation in der Steuerung verwendet)
ACCESS_ERROR (16#0003)	16#0815	16#0x02	Benötigter Parameter P-0-0061 ist nicht in den optionalen zyklischen MDT-Daten der Controlled Axis[x] konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0x03	Benötigter Parameter P-0-0690 ist nicht in den optionalen zyklischen MDT-Daten der Controlled Axis[x] konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0x04	Benötigter Parameter P-0-0691 ist nicht in den optionalen zyklischen MDT-Daten der Controlled Axis[x] konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0x05	Benötigter Parameter P-0-0692 ist nicht in den optionalen zyklischen MDT-Daten der Controlled Axis[x] konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0x06	Benötigter Parameter P-0-0694 ist nicht in den optionalen zyklischen MDT-Daten der Controlled Axis[x] konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0x07	Benötigter Parameter P-0-0695 ist nicht in den optionalen zyklischen MDT-Daten der Controlled Axis[x] konfiguriert
ACCESS_ERROR (16#0003)	16#0815	16#0009	Benötigter Parameter P-0-0753 ist nicht in den optionalen zyklischen AT-Daten der gemessenen Achse konfiguriert

Tab. 5-34: Fehlercodes Funktionsbaustein MB_RegiRegulateType04

5.4.6 MB_CORRECTION_MODE

Der Aufzählungstyp "MB_CORRECTION_MODE" enthält mögliche Formen von Korrekturfenstern für die Funktionsbausteine [MB_RegiRegulateType02, Seite 326](#), und [MB_RegiRegulateType04, Seite 341](#).

Name	Wert	Beschreibung
CORR_WIN_DISABLED	0	Korrekturfenster ist deaktiviert
CORR_ABS	1	absolutes Korrekturfenster [CorrectionStart .. CorrectionEnd].

Name	Wert	Beschreibung
CORR_REL	2	relatives Korrekturfenster [Anfang = "RecordedPosition" + "CorrectionStart" .. Ende = "RecordedPosition" + "CorrectionEnd"]
CORR_REL_ABS	3	relatives Anfangs... absolutes Endkorrekturfenster [Anfang = "RecordedPosition" + "CorrectionStart" .. Ende = "CorrectionEnd"]
CORR_ABS_REL	4	absolutes Anfangskorrekturfenster... relatives Endkorrekturfenster [Anfang = "CorrectionStart" .. Ende = "RecordedPosition" + "CorrectionEnd"]

Tab. 5-35: ENUM MB_CORRECTION_MODE

5.4.7 MB_RAMPING_MODE

Der Aufzählungstyp "MB_RAMPING_MODE" erlaubt für den Funktionsbaustein [MB_RegiRegulateType03](#), Seite 334, die Auswahl aus den folgenden Verrampungsmodi:

Name	Wert	Bedeutung
RAMPING_CONSTANT	0	Verrampungsgeschwindigkeit ist konstant
RAMPING_ADAPTIVE	1	Die Verrampungsgeschwindigkeit wird basierend auf der Zeitdifferenz im "Step"-Eingang berechnet

Tab. 5-36: ENUM MB_RAMPING_MODE

5.5 Registersollwertübernahme-FB

5.5.1 Einführung und Übersicht

Der Registersollwertübernahme-FB ist in den Bibliotheken ML_TechRegi (MLC) und MX_TechRegi.lib (MLD) enthalten.

5.5.2 MB_SetpointLockType01

Kurzbeschreibung Der Funktionsbaustein MB_RegiSetpointLockType01 dient zum Verriegeln eines Werts bei einem Registerregler.

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechRegi
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechRegi.lib (IL1G)
MLD mit MPx20 mit MA Funktionspaket	MX_TechRegi (IL2G)

Tab. 5-37: Referenztable MB_SetpointLockType01

Schnittstellenbeschreibung

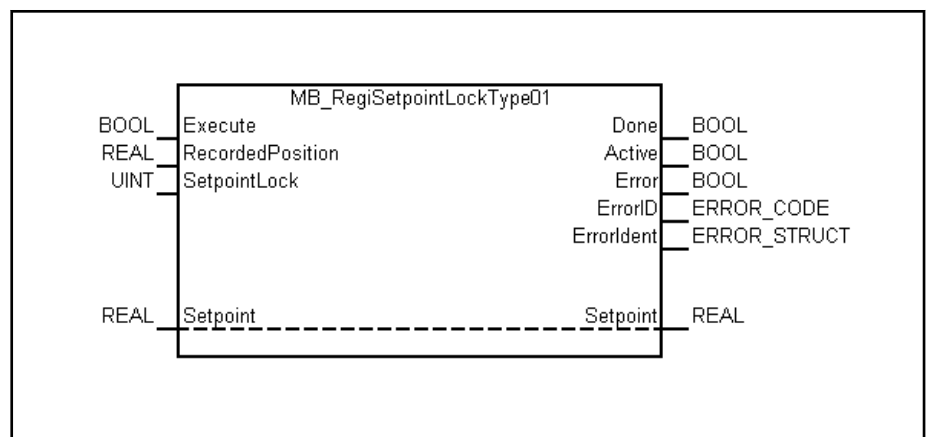


Abb. 5-29: Funktionsbaustein MB_SetpointLockType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN	Execute	BOOL	Der Registersollwertübernahme-FB wird bei einer ansteigenden Flanke aktiv
	RecordedPosition	REAL	Die aufgenommene Position wird in den Eingang "Setpoint" kopiert wenn sich "SetpointLock" ändert
	SetpointLock	UINT	Der Wert muss sich erhöhen wenn der Wert in "RecordedPosition" übernommen wird. Bei 65,535 schlägt der Wert nach 0 um (Wert vom Modulo-Typ). Normalerweise wird ein "actStep"-Ausgang eines Technologiefunktionsbausteins als Eingang verwendet
VAR_OUT	Done	BOOL	Der Funktionsbaustein ist abgeschlossen und der Wert im "RecordPosition"-Eingang wird zu "Setpoint" kopiert
	Active	BOOL	Funktionsbaustein ist aktiv
	Error	BOOL	Zeigt an, dass es zu einem Fehler während des Funktionsbausteinbetriebes kam
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 352
VAR_IN_OUT	Setpoint	REAL	Setpoint

Tab. 5-38: Schnittstellenvariablen Funktionsbaustein MB_RegiSetpointLockType01

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
RecordedPosition	REAL			0.0	Änderung von "SetpointLock"
SetpointLock	UINT	0	65.535	0	Kontinuierlich. Initialisiert bei ansteigender Flanke von "Execute"

Tab. 5-39: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Das folgende Diagramm zeigt den Signal-Zeit-Verlauf des Funktionsbausteins MB_RegiSetpointLockType01.

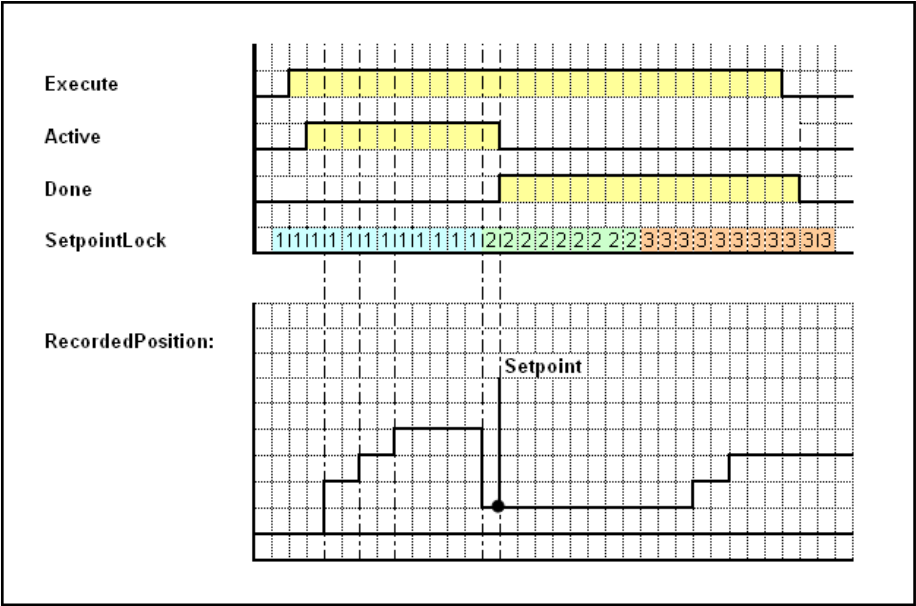


Abb. 5-30: Zeitdiagramm des MB_RegiSetpointLockType01

Funktionsbeschreibung Der Funktionsbaustein setzt "Setpoint" auf den als nächsten auftretenden Wert am "RecordedPosition"-Eingang. Das nächste Auftreten wird erkannt wenn sich der Eingang "SetpointLock" ändert. Man kann dies verwenden, um einen neuen Sollwert z.B. für einen Registerregler zu setzen.

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle F_RELATED_TABLE, 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins

Tab. 5-40: Fehlercodes Funktionsbaustein MB_RegiSetpointLockType01

5.6

Registerregler-FB

5.6.1

Einführung und Übersicht

Die Registerregler-FB sind in den Bibliotheken ML_TechRegi (MLC) und MX_TechRegi.lib (MLD) enthalten.

5.6.2

MB_RegisterControllerType02

- Kurzbeschreibung**
- Der Registerregler-FB MB_RegisterControllerType02 unterstützt die folgende Funktionalität:
- Berechnung eines Korrekturwerts auf Grundlage eines gemessenen Werts und eines Sollwerts unter Verwendung eines P- oder PI-Regelkreises
 - Preset-Funktion
 - Pause-Funktion
 - Min/Max-Begrenzung des berechneten Stellwerts
 - Korrekturfenster-Funktion
 - Sollwert übernehmen

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechRegi
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechRegi.lib

Tab. 5-41: Referenztabelle MB_RegisterControllerType02

Schnittstellenbeschreibung

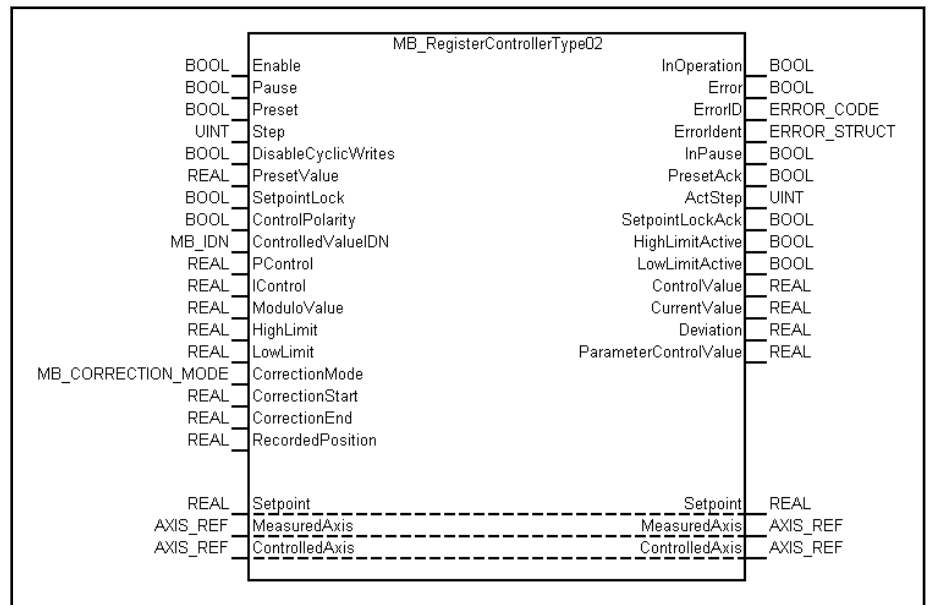


Abb. 5-31: Funktionsbaustein MB_RegisterControllerType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert den Registerregler
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist. Die Ausgänge "ControlValue" und "ParameterControlValue" sind konstant, solange "Pause" aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt den Regelkreis zwangsweise auf "PresetValue". Zusätzlich setzt der "Preset"-Eingang "ParameterControlValue" auf den "PresetValue"-Wert
	Step	UINT	Verarbeitungsschritt
	DisableCyclicWrites	BOOL	Deaktiviert zyklisches Schreiben auf ausgewählten Parameter in "ControlledValueIDN" wenn TRUE
	PresetValue	REAL	Wert an die Ausgänge "ControlValue" und "ParameterControlValue" bei steigender Flanke des "Preset"-Eingangs übergeben
	SetpointLock	BOOL	Aktiviert die Sollwertübernahme. Wenn TRUE, wird die nächste gemessene Marke gelatcht
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist

E/A-Typ	Name	Datentyp	Beschreibung
	ControlledValueIDN	MB_IDN	sercos IDN des geregelten Parameters (z.B. P-0-0691). Die folgenden IDNs werden unterstützt: MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 MLC: A-0-2730, P-0-0690, A-0-2615, P-0-0691, A-0-2610, P-0-0692, A-0-2600, P-0-0694, A-0-2605, P-0-0695, A-0-2620
	PControl	REAL	Proportionalverstärkung des PI-Reglers
	IControl	REAL	Nachstellzeit T_n des PI-Reglers
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
	HighLimit	REAL	Maximalwert für "ControlValue"
	LowLimit	REAL	Minimalwert für "ControlValue"
	CorrectionMode	MB_CORRECTION_MODE	Eigenschaften des Korrekturfensters MB_CORRECTION_MODE, Seite 349
	CorrectionStart	REAL	Anfang des Korrekturfensters
	CorrectionEnd	REAL	Ende des Korrekturfensters
	RecordedPosition	REAL	Gemessener Wert der Achsposition
VAR_OUTPUT	InOperation	BOOL	Funktionsbaustein ist in Betrieb
	Error	BOOL	Zeigt einen Fehler an. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 361
	InPause	BOOL	"Pause" ist aktiv. Die Ausgänge "ControlValue" und "ParameterControlValue" behalten ihre Werte
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in die Ausgänge "ControlValue" und "ParameterControlValue" kopiert
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "ControlValue" hoch
	SetpointLockAck	BOOL	"Setpoint" übernommen. Der aktuelle Wert in "RecordedPosition" wird nach "Setpoint" kopiert
	HighLimitActive	BOOL	Obergrenze ist aktiv. "ControlValue" wäre größer, ist aber auf "HighLimit" beschränkt
	LowLimitActive	BOOL	Untergrenze ist aktiv. "ControlValue" wäre niedriger, ist aber auf "LowLimit" beschränkt
	ControlValue	REAL	Vom Registerregler errechneter Stellwert
	CurrentValue	REAL	Die aktuell aufgezeichnete Lage, die im Registerregler verwendet wird
	Deviation	REAL	"Deviation" = ("Setpoint" - "CurrentValue") des Registerreglers

E/A-Typ	Name	Datentyp	Beschreibung
	ParameterControlValue	REAL	Der Registerregler berechnet einen Wert basierend auf dem ausgewählten Parameter im "ControlledValueIDN"-Eingang und speichert das Ergebnis im Ausgang "ParameterControlValue". Dieser Wert wird in den ausgewählten Parametereingang "ControlledValueIDN" geschrieben wenn der Eingang "DisableCyclicWrites" = FALSE ist
VAR_IN_OUT	Setpoint	REAL	Sollwert. Dieser Wert wird mit "RecordedPosition" verglichen
VAR_IN_OUT CONSTANT	MeasuredAxis	AXIS_REF	Bezug zur gemessenen Achse
VAR_IN_OUT CONSTANT	ControlledAxis	AXIS_REF	Bezug zur geregelten Achse

Tab. 5-42: Schnittstellenvariablen Funktionsbaustein MB_RegisterController-Type02

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	"Enable" TRUE (kontinuierlich)
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
SetpointLock	BOOL			FALSE	Kontinuierlich
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
Setpoint	REAL	n.def	n.def	0.0	Änderung von "Step"
PControl	REAL	0.0	n.def	1.0	Kontinuierlich
IControl	REAL	0.0	n.def	0.0	Kontinuierlich
ModuloValue	REAL	0.0	n.def	0.0	Steigende Flanke an "Enable"
HighLimit	REAL	LowLimit	n.def	0.0	Kontinuierlich
LowLimit	REAL	n.def	HighLimit	0.0	Kontinuierlich
CorrectionMode	MB_CORRECTION_MODE			CORR_REL	Kontinuierlich
CorrectionStart	REAL	0.0	ModuloValue	0.0	Kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
CorrectionEnd	REAL	0.0	ModuloValue	0.0	Kontinuierlich
RecordedPosition	REAL	n.def	n.def	0.0	Änderung von "Step"

Tab. 5-43: Übersicht über Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm

Das nachfolgende Diagramm zeigt den Signal-Zeit-Verlauf des FB MB_RegisterControllerType02 einschließlich der Funktionen "Pause" und "Preset".

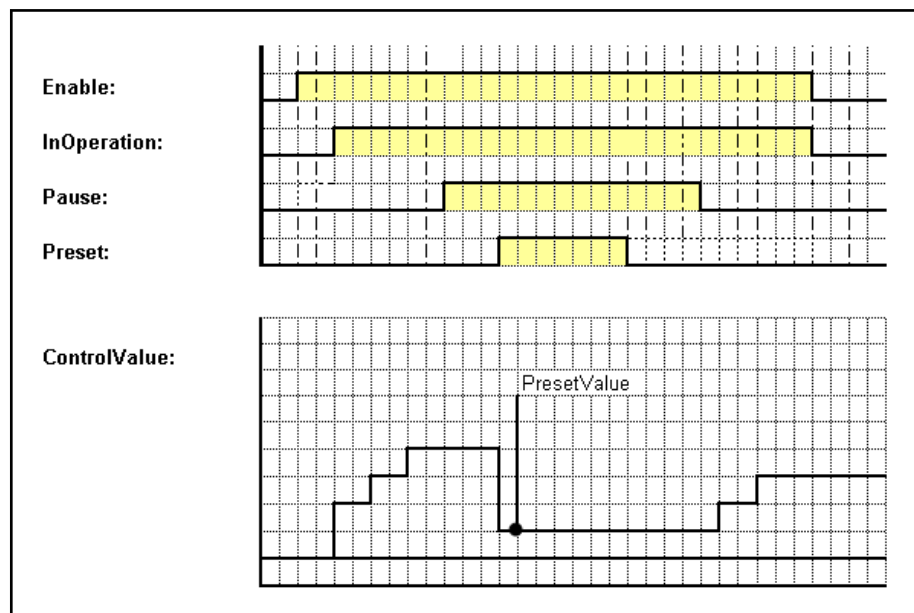


Abb. 5-32: Zeitdiagramm des FB MB_RegisterControllerType02 mit den Funktionen "Pause" und "Preset"

Funktionsbeschreibung

Ein Sensor erkennt Marken, Perforationen, Schnitte oder einen geklebten Stoß auf dem Material und meldet dies dem Antrieb als (binäres) Signal.

Die Messtasterfunktion des Antriebs bestimmt die Flanke des Sensorsignals und verriegelt, abhängig von der Signalauswahl des Messtasters, die folgenden Stellungen:

Antriebsistposition Parameter	S-0-0051, S-0-0053, P-0-0753
Leitachspostion Parameter	P-0-0052, P-0-0227, P-0-0775, P-0-0776, P-0-0778

Die Messtasterfunktion zeichnet Positionsdaten mit einer Auflösung von 0,5µs auf. Der Sensor muss ein 24V-Signal mit einer Anstiegszeit im µs-Bereich bereitstellen. Die sensorspezifische Verzögerungszeit kann mithilfe der Eingänge "DeadTimePosEdge" oder "DeadTimeNegEdge" des Funktionsbausteins MB_InitTouchProbe kompensiert werden.

Der Registerregler berechnet die Regelungsabweichung zwischen der gemessenen und der Sollwertposition und ermittelt den erforderlichen Korrekturwert, sobald eine neue Flanke des Sensorsignals ermittelt wird.

Änderungen des Korrekturwertes werden über das trapezförmiges Profil, die Geschwindigkeitsrampe, den PT1-Filter oder ohne Verzögerung aktiv, abhängig vom ausgewählten Regelungsparametereingang ("ControlledValueIDN") des Funktionsbausteins. Die nachstehende Tabelle führt die verfügbaren Regelungsparameter ("ControlledValueIDN") sowie das daraus resultierende MotionProfile aller verfügbaren Antriebsbetriebsarten mit Synchronisation auf.

Parametername	Controlled-ValueIDN	Bewegungsverhalten, wenn sich ein Parameterwert ändert	Verfügbar in den folgenden Betriebsmodus			
			Velocity-Synchronization	Phase-Synchronization	Cam Shaft	Electronic Motion Profile
Winkelverschiebung Anfang des Profils	P-0-0061 A-0-2730	Änderungen von P-0-0061 werden über Rampe mithilfe P-0-0158 (Geschwindigkeit) aktiv. Wenn P-0-0158=0, werden Änderungen an P-0-0061 sofort aktiv.	Nein	Nein	Ja	Ja
Geschwindigkeits-Sollwert additiv	P-0-0690 A-0-2615	Änderungen von P-0-0690 werden sofort aktiv	Ja	Nein	Nein	Nein
Lagesollwert additiv, Prozessregler	P-0-0691 A-0-2610	Wenn P-0-0155 Bit0 = 0: Änderungen von P-0-0691 werden über PT1-Filter mittels P-0-0060 (Filterzeitkonstante) aktiv. Wenn P-0-0060=0, werden Änderungen an P-0-0691 sofort aktiv. Wenn P-0-0155 Bit0 = 1: Änderungen von P-0-0691 werden sofort aktiv	Ja	Ja	Ja	Ja
Leitachspannung additiv, Prozessregler	P-0-0692 A-0-2600	Änderungen von P-0-0692 werden mit einem Filter 1. Ordnung geglättet bei Verwendung von P-0-0693 (Filterzeitkonstante) und dann zur Summe aus P-0-0054, Leitachspannung additiv, und P-0-0053, Leitachspannung, oder P-0-0052, Lageistwert Messgeber, addiert.	Ja	Ja	Ja	Ja
Feinabgleich Getriebeübersetzung, Prozessregler	P-0-0694 A-0-2605	Änderungen von P-0-0694 werden sofort aktiv	Ja	Ja	Ja	Ja
Winkelverschiebung Tabellenanfang, Prozessregler	P-0-0695 A-0-2620	Änderungen von P-0-0695 werden über PT1-Filter mittels P-0-0696 (Filterzeitkonstante) aktiv. Wenn P-0-0696=0, werden Änderungen an P-0-0695 sofort aktiv.	Nein	Nein	Ja	Ja

Abb. 5-33: Unterstützte Regelparameter und unterstütztes Verhalten

Die folgenden "ControlledValueIDN"-Werte werden integriert:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Initialisierung Während der Initialisierung müssen die Eingänge "Setpoint" und "Recorded-Position" über gültige Daten verfügen.

PresetValue Für einen fehlerfreien Start der Reglerausgänge "ControlValue" und "ParameterControlValue" muss der "Preset"-Eingang gesetzt sein, bevor der "Enable"-Eingang aktiviert wird. Während dieser Zeit müssen die Eingänge "Setpoint" und "RecordedPosition" über gültige Daten verfügen. Zudem werden die oberen und unteren Grenzwerte überwacht, um eine Begrenzung

des Ausgangs "ControlValue" sicherzustellen, wenn der aktuelle Wert für "PresetValue" außerhalb des Gültigkeitsbereiches von "HighLimit" bzw. "LowLimit" liegt.

Der "PresetAck"-Ausgang ist aktiv wenn der Wert am "PresetValue"-Eingang in die Ausgänge "ControlValue" und "ParameterControlValue" kopiert wird. Der "Preset"-Eingang hat eine höhere Priorität als der "Pause"-Eingang. Der Wert in "PresetValue" wird in die entsprechenden Ausgänge bei steigender Flanke am "Preset"-Eingang kopiert wenn "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt.

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. In diesem Fall wird die Abweichung ("Setpoint" – "RecordedPosition") auf 0 gesetzt. Dadurch ändern sich die Ausgänge "ControlValue" und "ParameterControlValue" nicht mehr. Diese kann verwendet werden, um die Werte der Ausgänge "ControlValue" und "ParameterControlValue" beizubehalten. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Solange der Eingang "Pause" aktiv ist, ist der Eingang "InPause" TRUE.

Sollwertübernahme Einen Moduswechsel von Manuell zu Automatik wird durch das Einrasten beim nächsten erkannten Registermarker erreicht. Wenn zum Beispiel eine neue Papierbahn in Position gebracht und die Registermarke erkannt wird, wird der Eingang "SetpointLock" auf TRUE gesetzt. Der Wert im "Setpoint" wird mit dem Wert im "RecordedPosition" überschrieben sobald der "Step"-Eingang geändert wird.

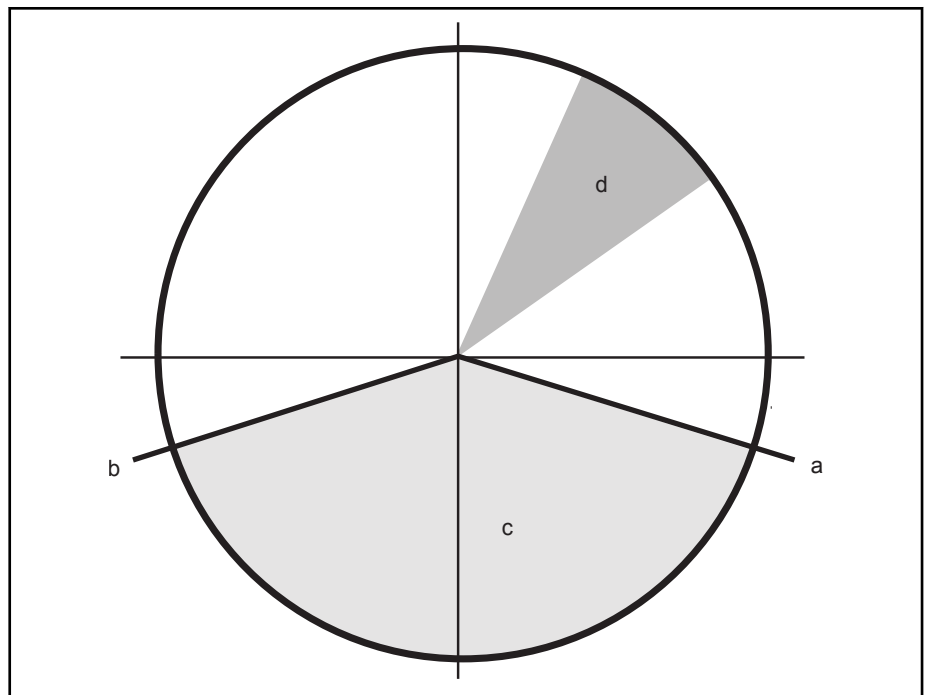
Korrekturfenster Das Korrekturfenster ist ein auf die Messachse bezogener Winkelbereich. Der Registerregler regelt die Regelungsabweichungen innerhalb dieses Fensters. Die Lage und Größe des Korrekturfensters werden mithilfe der folgenden Eingänge gesetzt:

Der Eingang "CorrectionMode" stellt das Korrekturfenster entweder auf absolut oder relativ ein. Wenn "CorrectionMode" auf CORR_WIN_DISABLED gesetzt wird, ist das Korrekturfenster inaktiv.

Der Eingang "CorrectionStart" legt den Anfang des Korrekturfensters fest.

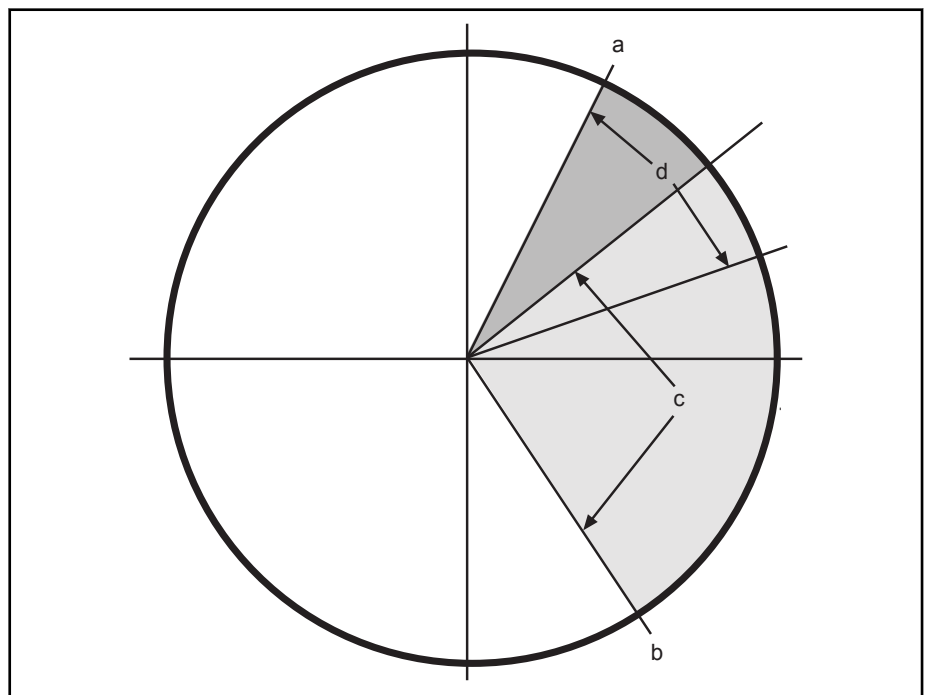
Der Eingang "CorrectionEnd" legt das Ende des Korrekturfensters fest.

Das Korrekturfenster unterstützt nur einen positiven Modulo-Bereich (z.B. 0...+360 Grad) der Messachse.



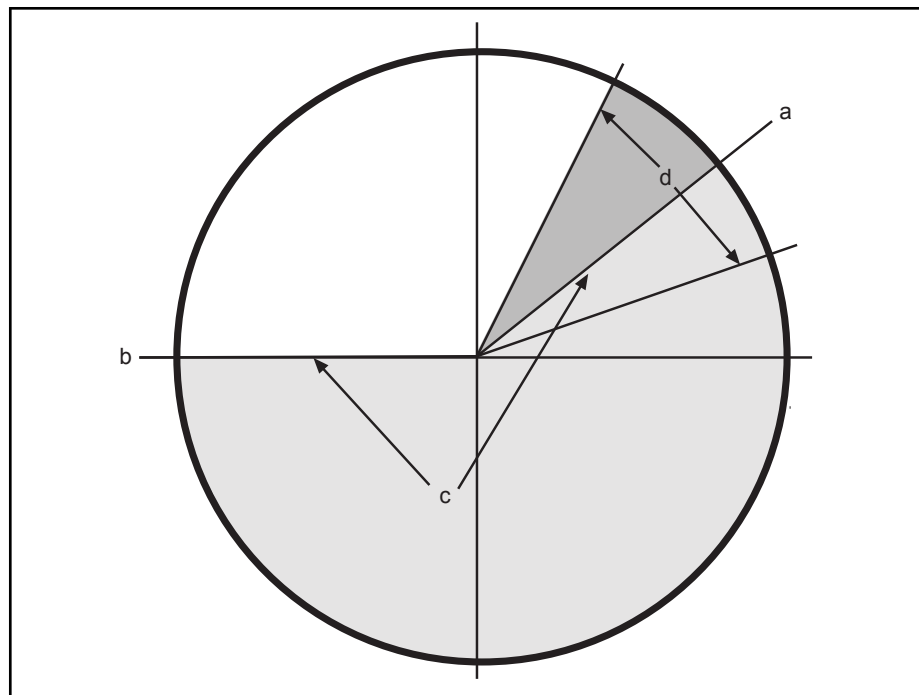
- a:** Beginn Korrekturfenster
b: Ende Korrekturfenster
c: Korrekturfenster
d: Erwartungsfenster

Abb. 5-34: Korrekturfenster absolut



- a:** Anfang Korrekturfenster = Messende
b: Relatives Ende des Korrekturfensters bezogen auf das Messende
c: Korrekturfenster
d: Erwartungsfenster

Abb. 5-35: Korrekturfenster relativ



- a:** Anfang Korrekturfenster = Messende
b: Ende des Korrekturfensters = perm. Winkel
c: Korrekturfenster
d: Erwartungsfenster

Abb. 5-36: Korrekturfenster, relativer Start und absolutes Ende

Normalbetrieb

Der RegisterController berechnet nur einen neuen Wert, wenn sich der Eingang "Step" verändert. Die Reglerparameter "PControl" und "IControl" können permanent geändert werden.

Wenn der Wert im "ControlValue"-Ausgang die Werte von "HighLimit" oder "LowLimit" über-, bzw. unterschreitet, dann wird "ControlValue" auf einen der folgenden Grenzwerte begrenzt.

- Ist der Wert in "ControlValue" positiv, wird der Wert in "HighLimit" verwendet.
- Ist der Wert in "ControlValue" negativ, wird der Wert in "LowLimit" verwendet.

Wenn der Registerregler eine Nachstellzeit hat, wird zusätzlich ein Anti-Reset-Windup durchgeführt und die Nachstellzeit wird nicht länger aufsummiert. In diesem Fall werden die Ausgänge "HighLimitActive" und "LowLimitActive" auf TRUE gesetzt.

Notwendige Parametrierung

MLC-Parametrierung

Wird eine **reale Achse mit Interpolation im Antrieb** als geregelte Achse ("ControlledAxis"-Eingang) benutzt, muss der verwendete Parameter im "ControlValueIDN"-Eingang einer der nachfolgenden unterstützten P-Parameter sein:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 und P-0-0695



Zusätzlich muss dieser P-Parameter im MDT des zyklischen Datenkanals im Antrieb konfiguriert sein.

Wird eine **reale Achse mit Interpolation in der Steuerung** oder eine **virtuelle Achse** als geregelte Achse benutzt, muss der verwendete Parameter im

"ControlledValueIDN"-Eingang einer der nachfolgenden unterstützten A-Parameter sein:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

MLD-Parametrierung

Der Parameter, der als geregelter Parameter für den "ControlledValueIDN"-Eingang (P-Parameter) verwendet wird, muss in der "AxisData"-Struktur konfiguriert werden. Daher muss die Achsdatenstruktur aktiviert werden. Folgende P-Parameter werden unterstützt:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

Ab MLD Firmware MPx20 (IL2G) werden am Eingang "MeasuredAxis" auch Achsreferenzen unterstützt, die nicht in der "AxisData"-Struktur enthalten sind (> VMA1).

In diesem Fall wird das Korrekturfenster nicht über die aktuelle Position der "MeasuredAxis" ermittelt, sondern über P-0-0761 "Leitachspannung für Folgeachse" der lokalen Achse gebildet.

Damit im P-0-0761 "Leitachspannung für Folgeachse" die benötigte Positionsinformation zur Verfügung steht, muss der Leitachsgenerator entsprechend konfiguriert werden.

Beteiligte Parameter:

- P-0-0760 "Virtuelle Leitachse, Positionssteuerwort"
- P-0-0916 "Leitachsformatwandler Signalauswahl"

Siehe "Virtueller Leitachsgenerator" in der Dokumentation IndraDrive MPx-20 Funktionen Anwendungsbeschreibung, DOK-INDRV*-MP*-20VRS**-APRS-DE-P.

Fehlerbehandlung

Der Funktionsbaustein gibt die Fehlermeldungen der folgenden unterlagerten Funktionsbausteine aus:

- [MB_RegiMeasuringType01, Seite 308](#)
- Fehlercodes Funktionsbaustein IL_PIType02, RIL_LoopControl
Bibliothek RIL_LoopControl (siehe Dokumentation "Rexroth IndraWorks 14VRS Basis-Bibliotheken IndraLogic 2G", DOK-IL*2G*-BASLIB**V14-LIRS-DE-P, R911343918)
- [MB_RegiRegulateType02, Seite 326](#)
- [MB_SetpointLockType01, Seite 350](#)

Er erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"Axis_Ref" liegt außerhalb des gültigen Wertebereichs (gemessen/ geregelt)
INPUT_RANGE_ERROR (16#0006)	16#0813	16#0001	"HighLimit" < "LowLimit"
INPUT_RANGE_ERROR (16#0006)	16#0813	16#0002	"PControl" < 0

Tab. 5-44: Fehlercodes Funktionsbaustein MB_RegisterControllerType02

5.6.3 MB_RegisterControllerType04

- Kurzbeschreibung

Der Registerregler-FB MB_RegisterControllerType04 unterstützt die folgende Funktionalität:
- Berechnung eines Korrekturwerts auf Grundlage eines gemessenen Werts und eines Sollwerts unter Verwendung eines P- oder PI-Regelkreises
 - Preset-Funktion
 - Pause-Funktion
 - Min/Max-Begrenzung des berechneten Stellwerts
 - Korrekturfenster-Funktion
 - Schieberegister
 - Totzone-Kp-Kennlinie
 - PT1-Element
 - Sollwert übernehmen
 - Markentakt-Funktion

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechRegi
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechRegi.lib (IL1G)
MLD mit MPx20 mit MA Funktionspaket	MX_TechRegi (IL2G)

Tab. 5-45: Referenztable MB_RegisterControllerType04

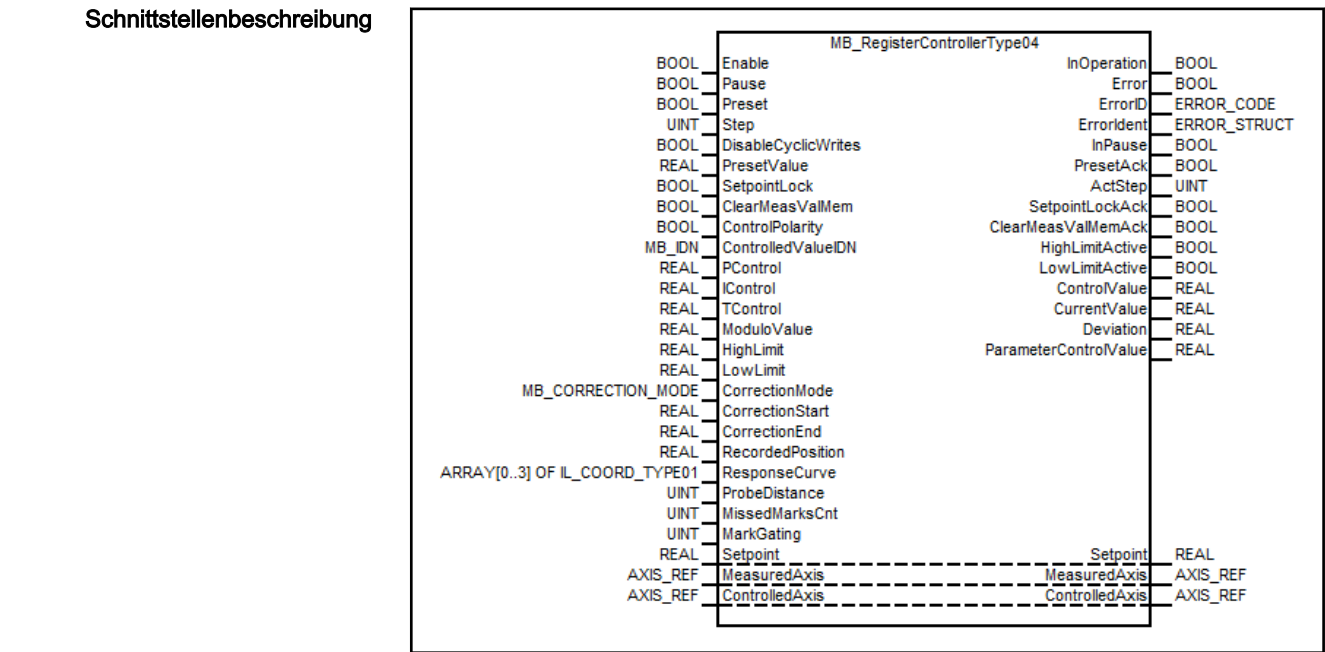


Abb. 5-37: Funktionsbaustein MB_RegisterControllerType04

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Aktiviert den Registerregler
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist. Die Ausgänge "ControlValue" und "ParameterControlValue" sind konstant, solange "Pause" aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt den Regelkreis zwangsweise auf "PresetValue". Zusätzlich setzt der "Preset"-Eingang "ParameterControlValue" auf den "PresetValue"-Wert
	Step	UINT	Verarbeitungsschritt
	DisableCyclicWrites	BOOL	Deaktiviert zyklisches Schreiben auf ausgewählten Parameter in "ControlValueIDN" wenn TRUE
	PresetValue	REAL	Wert an die Ausgänge "ControlValue" und "ParameterControlValue" bei steigender Flanke des "Preset"-Eingangs übergeben
	SetpointLock	BOOL	Aktiviert die Sollwertübernahme. Wenn TRUE, wird die nächste gemessene Marke gelatcht
	ClearMeasValMem	BOOL	Speicher für die Messwerte, die wird von der Funktion „Messtaster Abstand“ verwendet werden, löschen
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist
	ControlValueIDN	MB_IDN	Sercos IDN des geregelten Parameters (z.B. P-0-0691). Die folgenden IDNs werden unterstützt: MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 MLC: A-0-2730, P-0-0690, A-0-2615, P-0-0691, A-0-2610, P-0-0692, A-0-2600, P-0-0694, A-0-2605, P-0-0695, A-0-2620
	PControl	REAL	Proportionalverstärkung des PI-Reglers
	IControl	REAL	Nachstellzeit T_n des PI-Reglers
	TControl	REAL	Filterkonstante für den Eingang "RecordedPosition"
	ModuloValue	REAL	Modulo-Wert der gemessenen Achse
	HighLimit	REAL	Maximalwert für "ControlValue"
	LowLimit	REAL	Minimalwert für "ControlValue"
	CorrectionMode	MB_CORRECTION_MODE	Eigenschaften des Korrekturfensters MB_CORRECTION_MODE , Seite 349
	CorrectionStart	REAL	Anfang des Korrekturfensters
	CorrectionEnd	REAL	Ende des Korrekturfensters
	RecordedPosition	REAL	Lage, an der die Flanke auftrat (in technischen Einheiten)
	ResponseCurve	ARRAY[0..3] OF IL_COORD_TYPE01	Kennlinie

E/A-Typ	Name	Datentyp	Beschreibung
	ProbeDistance	UINT	Der Abstand zwischen dem Markensensor und der geregelten Achse in Formatlängen (Leitachsenumdrehungen, Formate)
	MissedMarksCnt	UNIT	Zähler für fehlende Marken bei Verwendung eines Erwartungsfensters
	MarkGating	UINT	Jede n-te Marke wird für den Register verwendet
VAR_OUTPUT	InOperation	BOOL	Registerregler ist in Betrieb
	Error	BOOL	Zeigt einen Fehler an. Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 374
	InPause	BOOL	"Pause" ist aktiv. Die Ausgänge "ControlValue" und "ParameterControlValue" behalten ihre Werte
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in die Ausgänge "ControlValue" und "ParameterControlValue" kopiert
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "ControlValue" hoch
	SetpointLockAck	BOOL	"Setpoint" übernommen. Der aktuelle Wert in "RecordedPosition" wird nach "Setpoint" kopiert
	ClearMeasValMemAck	BOOL	Löschen des Messwertspeichers ist abgeschlossen
	HighLimitActive	BOOL	Obergrenze ist aktiv. "ControlValue" wäre größer, ist aber auf "HighLimit" beschränkt
	LowLimitActive	BOOL	Untergrenze ist aktiv. "ControlValue" wäre niedriger, ist aber auf "LowLimit" beschränkt
	ControlValue	REAL	Vom Registerregler errechneter Stellwert
	CurrentValue	REAL	Im Registerregler momentan verwendeter Wert. Er ist gleich dem "RecordedPosition"-Eingang wenn "ProbeDistance" und "MarkGating" deaktiviert sind
	Deviation	REAL	"Deviation" = ("Setpoint" - "CurrentValue") des Registerreglers
	ParameterControlValue	REAL	Der Registerregler berechnet einen Wert basierend auf dem ausgewählten Parameter im "ControlledValueIDN"-Eingang und speichert das Ergebnis im Ausgang "ParameterControlValue". Dieser Wert wird in den ausgewählten Parametereingang "ControlledValueIDN" geschrieben wenn der Eingang "DisableCyclicWrites" = FALSE ist
VAR_IN_OUT	Setpoint	REAL	Sollwert. Dieser Wert wird mit "RecordedPosition" verglichen

E/A-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT CONSTANT	MeasuredAxis	AXIS_REF	Bezug zur gemessenen Achse
VAR_IN_OUT CONSTANT	ControlledAxis	AXIS_REF	Bezug zur geregelten Achse

Tab. 5-46: Schnittstellenvariablen Funktionsbaustein MB_RegisterController-Type04

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	"Enable" TRUE (kontinuierlich)
Step	UINT	0	65.535	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
SetpointLock	BOOL			FALSE	Kontinuierlich
ClearMeasValMem	BOOL			FALSE	Kontinuierlich
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
Setpoint	REAL	n.def	n.def	0.0	Änderung von "Step"
PControl	REAL	0.0	n.def	1.0	Kontinuierlich
IControl	REAL	0.0	n.def	0.0	Kontinuierlich
TControl	REAL	0.0	n.def	0.0	Kontinuierlich
ModuloValue	REAL	0.0	n.def	0.0	Steigende Flanke an "Enable"
HighLimit	REAL	LowLimit	n.def	0.0	Kontinuierlich
LowLimit	REAL	n.def	HighLimit	0.0	Kontinuierlich
CorrectionMode	MB_CORRECTION_MODE			CORR_REL	Kontinuierlich
CorrectionStart	REAL	0.0	ModuloValue	0.0	Kontinuierlich
CorrectionEnd	REAL	0.0	ModuloValue	0.0	Kontinuierlich
RecordedPosition	REAL	n.def	n.def	0.0	Änderung von "Step"

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ResponseCurve	ARRAY[0..3] OF IL_COORD_TY-PE_01	n.def	n.def	0.0	Steigende Flanke an "Enable"
ProbeDistance	UINT	0	30	0	Kontinuierlich
MissedMarksCnt	UINT	0	65.535	0	Kontinuierlich. Initialisierung bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
MarkGating	UINT	0	65.535	1	Steigende Flanke an "Enable". Ein Wert von "0" deaktiviert den Markentakt (identisch mit dem Wert "1")

Tab. 5-47: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm Das folgende Diagramm zeigt den Signal-Zeit-Verlauf von FB MB_RegisterControllerType04 einschließlich der Funktionen "Pause" und "Preset".

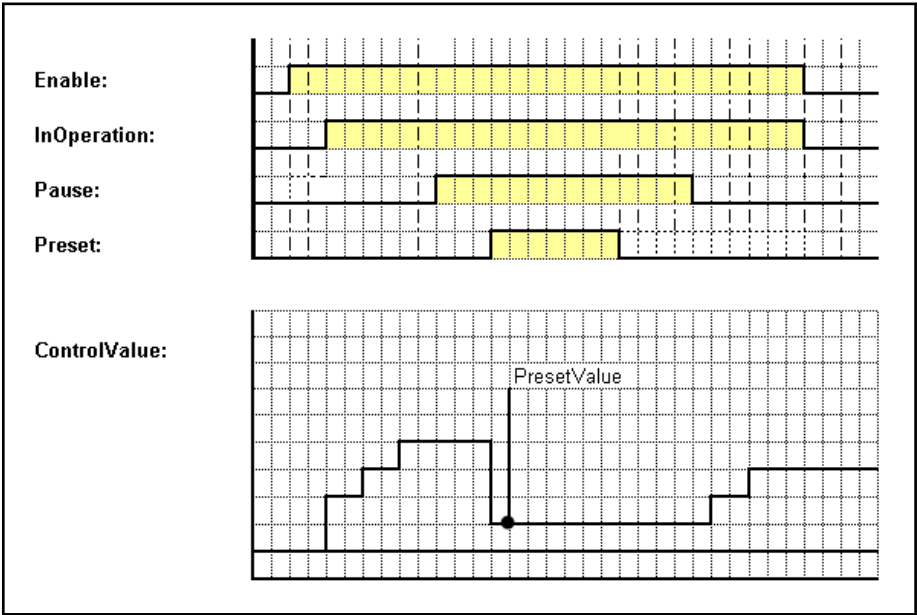


Abb. 5-38: Zeitdiagramm des FB MB_RegisterControllerType04 mit den Funktionen "Pause" und "Preset"

Funktionsbeschreibung Ein Sensor erkennt Marken, Perforationen, Schnitte oder einen geklebten Stoß auf dem Material und meldet dies dem Antrieb als (binäres) Signal. Die Messtasterfunktion des Antriebs bestimmt die Flanke des Sensorsignals und verriegelt, abhängig von der Signalauswahl des Messtasters, die folgenden Stellungen:

Antriebsistposition Parameter	S-0-0051, S-0-0053, P-0-0753
Leitachsisposition Parameter	P-0-0052, P-0-0227, P-0-0775, P-0-0776, P-0-0778

Die Messtasterfunktion zeichnet Positionsdaten mit einer Auflösung von 0,5µs auf. Der Sensor muss ein 24V-Signal mit einer Anstiegszeit im µs-Bereich bereitstellen. Die sensorspezifische Verzögerungszeit kann mithilfe der

Eingänge "DeadTimePosEdge" oder "DeadTimeNegEdge" des Funktionsbausteins MB_InitTouchProbe kompensiert werden.

Der Registerregler berechnet die Regelungsabweichung zwischen der gemessenen und der Sollwertposition und ermittelt den erforderlichen Korrekturwert, sobald eine neue Flanke des Sensorsignals ermittelt wird.

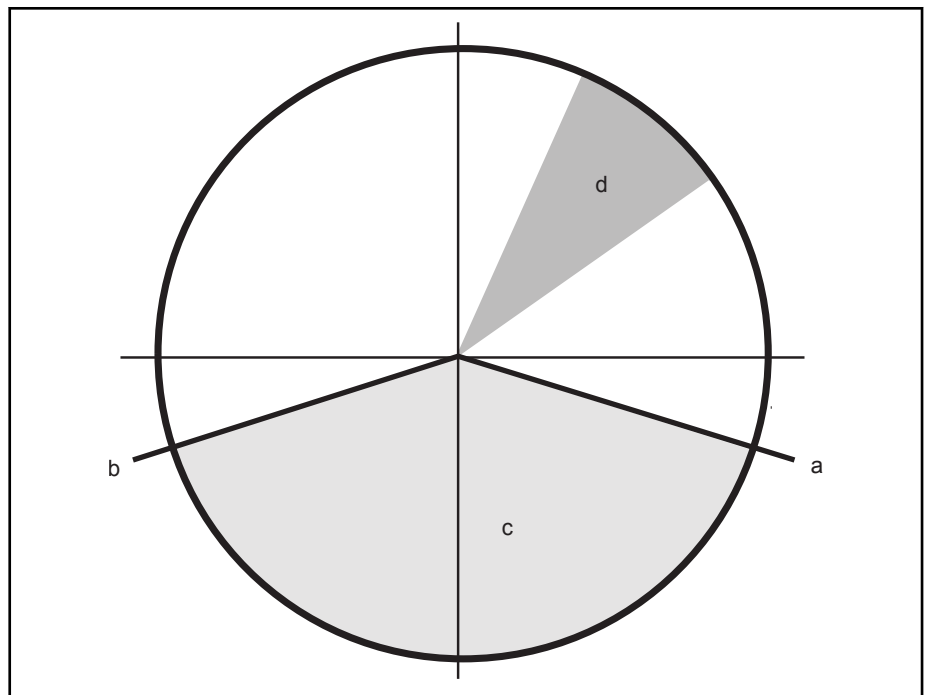
Änderungen des Korrekturwertes werden über das trapezförmiges Profil, die Geschwindigkeitsrampe, den PT1-Filter oder ohne Verzögerung aktiv, abhängig vom ausgewählten Regelungsparametereingang ("ControlledValueIDN") des Funktionsbausteins. Die nachstehende Tabelle führt die verfügbaren Regelungsparameter ("ControlledValueIDN") sowie das daraus resultierende MotionProfile aller verfügbaren Antriebsbetriebsarten mit Synchronisation auf.

Parametername	Controlled-ValueIDN	Bewegungsverhalten, wenn sich ein Parameterwert ändert	Verfügbar in den folgenden Betriebsmodus			
			Velocity-Synchronization	Phase-Synchronization	Cam Shaft	Electronic Motion Profile
Winkelverschiebung Anfang des Profils	P-0-0061 A-0-2730	Änderungen von P-0-0061 werden über Rampe mithilfe P-0-0158 (Geschwindigkeit) aktiv. Wenn P-0-0158=0, werden Änderungen an P-0-0061 sofort aktiv.	Nein	Nein	Ja	Ja
Geschwindigkeits-Sollwert additiv	P-0-0690 A-0-2615	Änderungen von P-0-0690 werden sofort aktiv	Ja	Nein	Nein	Nein
Lagesollwert additiv, Prozessregler	P-0-0691 A-0-2610	Wenn P-0-0155 Bit0 = 0: Änderungen von P-0-0691 werden über PT1-Filter mittels P-0-0060 (Filterzeitkonstante) aktiv. Wenn P-0-0060=0, werden Änderungen an P-0-0691 sofort aktiv. Wenn P-0-0155 Bit0 = 1: Änderungen von P-0-0691 werden sofort aktiv	Ja	Ja	Ja	Ja
Leitachspannung additiv, Prozessregler	P-0-0692 A-0-2600	Änderungen von P-0-0692 werden mit einem Filter 1. Ordnung geglättet bei Verwendung von P-0-0693 (Filterzeitkonstante) und dann zur Summe aus P-0-0054, Leitachspannung additiv, und P-0-0053, Leitachspannung, oder P-0-0052, Lageistwert Messgeber, addiert.	Ja	Ja	Ja	Ja
Feinabgleich Getriebeübersetzung, Prozessregler	P-0-0694 A-0-2605	Änderungen von P-0-0694 werden sofort aktiv	Ja	Ja	Ja	Ja
Winkelverschiebung Tabellenanfang, Prozessregler	P-0-0695 A-0-2620	Änderungen von P-0-0695 werden über PT1-Filter mittels P-0-0696 (Filterzeitkonstante) aktiv. Wenn P-0-0696=0, werden Änderungen an P-0-0695 sofort aktiv.	Nein	Nein	Ja	Ja

Abb. 5-39: Unterstützte Regelparameter und unterstütztes Verhalten

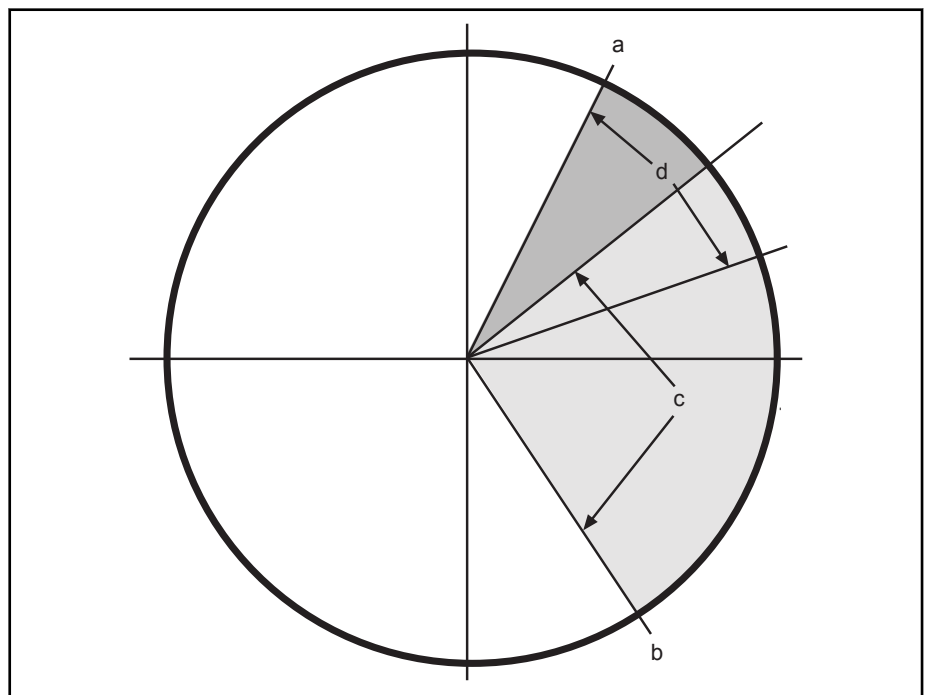
Die folgenden "ControlledValueIDN"-Werte werden integriert:

	• P-0-0061, A-0-2730 • P-0-0691, A-0-2610 • P-0-0692, A-0-2600 • P-0-0695, A-0-2620
Initialisierung	Während der Initialisierung müssen die Eingänge „Setpoint“ und „RecordedPosition“ gültige Daten enthalten.
PresetValue	Für einen fehlerfreien Start der Reglerausgänge "ControlValue" und "ParameterControlValue" muss der "Preset"-Eingang gesetzt sein, bevor der "Enable"-Eingang aktiviert wird. Während dieser Zeit müssen die Eingänge "Setpoint" und "RecordedPosition" über gültige Daten verfügen. Zudem werden die oberen und unteren Grenzwerte überwacht, um eine Begrenzung des Ausgangs "ControlValue" sicherzustellen, wenn der aktuelle Wert für "PresetValue" außerhalb des Gültigkeitsbereiches von "HighLimit" bzw. "LowLimit" liegt. Der "PresetAck"-Ausgang ist aktiv wenn der Wert am "PresetValue"-Eingang in die Ausgänge "ControlValue" und "ParameterControlValue" kopiert wird. Der "Preset"-Eingang hat eine höhere Priorität als der "Pause"-Eingang. Der Wert in "PresetValue" wird in die entsprechenden Ausgänge bei steigender Flanke am "Preset"-Eingang kopiert wenn "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt.
Pause	Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. In diesem Fall wird die Abweichung ("Setpoint" – "RecordedPosition") auf 0 gesetzt. Dadurch ändern sich die Ausgänge "ControlValue" und "ParameterControlValue" nicht mehr. Diese kann verwendet werden, um die Werte der Ausgänge "ControlValue" und "ParameterControlValue" beizubehalten. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Solange der Eingang "Pause" aktiv ist, ist der Eingang "InPause" TRUE.
Sollwertübernahme	Einen Moduswechsel von Manuell zu Automatik wird durch das Einrasten beim nächsten erkannten Registermarker erreicht. Wenn zum Beispiel eine neue Papierbahn in Position gebracht und die Registermarke erkannt wird, wird der Eingang "SetpointLock" auf TRUE gesetzt. Der Wert im "Setpoint" wird mit dem Wert im "RecordedPosition" überschrieben sobald der "Step"-Eingang geändert wird.
Korrekturfenster	Das Korrekturfenster ist ein auf die Messachse bezogener Winkelbereich. Der Registerregler regelt die Regelungsabweichungen innerhalb dieses Fensters. Die Lage und Größe des Korrekturfensters werden mithilfe der folgenden Eingänge gesetzt: Der Eingang "CorrectionMode" stellt das Korrekturfenster entweder auf absolut oder relativ ein. Wenn "CorrectionMode" auf CORR_WIN_DISABLED gesetzt wird, ist das Korrekturfenster inaktiv. Der Eingang "CorrectionStart" legt den Anfang des Korrekturfensters fest. Der Eingang "CorrectionEnd" legt das Ende des Korrekturfensters fest. Das Korrekturfenster unterstützt nur einen positiven Modulo-Bereich (z.B. 0...+360 Grad) der Messachse.



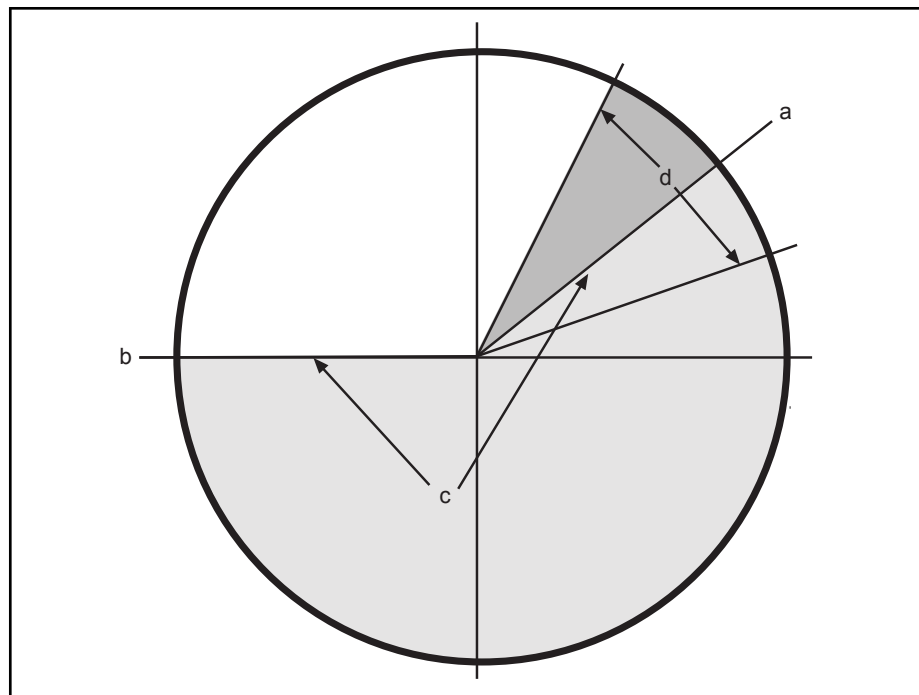
- a:** Beginn Korrekturfenster
b: Ende Korrekturfenster
c: Korrekturfenster
d: Erwartungsfenster

Abb. 5-40: Korrekturfenster absolut



- a:** Anfang Korrekturfenster = Messende
b: Relatives Ende des Korrekturfensters bezogen auf das Messende
c: Korrekturfenster
d: Erwartungsfenster

Abb. 5-41: Korrekturfenster relativ



- a:** Anfang Korrekturfenster = Messende
b: Ende des Korrekturfensters = perm. Winkel
c: Korrekturfenster
d: Erwartungsfenster

Abb. 5-42: Korrekturfenster, relativer Start und absolutes Ende

Glättung Der Eingang "TControl" ist die Dämpfungskonstante für den Eingang "RecordedPosition". Wenn er auf 0 gesetzt wird, ist die Glättung inaktiv.

ProbeDistance Im einfachsten Fall ist die Messachse ein Formatzylinder (= geregelte Achse) und der Abstand des Sensors zur Messachse (= geregelten Achse) ist kleiner als die Formatlänge. Messung und Korrektur beziehen sich auf dasselbe Produkt. Oft muss der Markenleser (Sensor) weiter weg von der Messachse angebracht werden, z.B. bei Gefahrbereichen in Druckanlagen. Wenn der Abstand größer als die Formatlänge ist, kann der gemessene Registerfehler erst korrigiert werden wenn das Produkt den Formatzylinder erreicht hat. Die Korrektur muss um so viele Formate verzögert werden, wie es Formate zwischen der geregelten Achse und dem Sensor gibt. Der Eingang "ProbeDistance" ist der Abstand zwischen dem Markenleser (Sensor) und der geregelten Achse, in Formatlängen (Leitachsumdrehungen, Formate). Wenn er auf 0 gesetzt ist, ist der Abstand des Sensors zur Messachse kleiner als die Formatlänge.

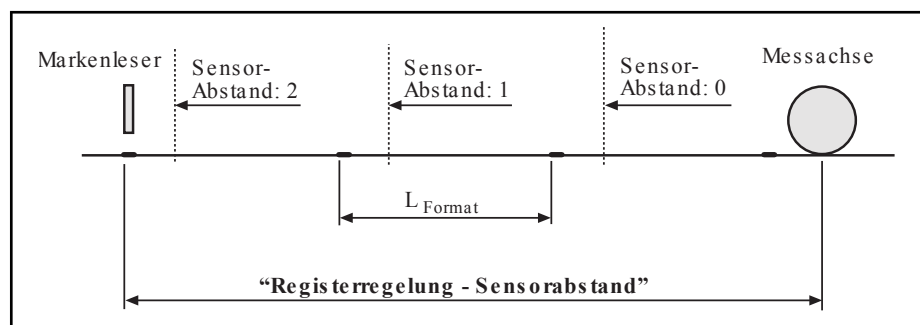


Abb. 5-43: Messtasterabstand



bei Verwendung der ProbeDistance wird vorausgesetzt, dass die zuvor ins Schieberegister eingetragenen Abweichungen jeweils zu 100% ausgeregelt werden ($P\_Control = 1$ / $I\_Control = 0$).

Das heißt: Als neue Abweichung wird jeweils nur das Delta zur Summe der bereits im Schieberegister enthaltenen Abweichungen eingetragen.

Löschen des internen Messwertspeichers

Die Funktion „Messtasterabstand“ enthält einen internen Speicher, der Messwerte (wie ein Schieberegister) speichert. Dieser interne Speicher kann mit dem Eingang "ClearMeasValMem" gelöscht (auf "0" gesetzt) werden.

ResponseCurve

Mit Hilfe des Eingangs "ResponseCurve" kann die Regelabweichung gewichtet werden. Durch diese Funktionalität kann beispielsweise bei kleinen Regelabweichungen eine geringere Gewichtung der Regelabweichung eingestellt werden. Dies ergibt quasi eine geringere Proportionalverstärkung und dadurch ein ruhigeres Regelverhalten im eingeschwungenen Zustand. Bei der Parametrierung des Eingangs sollten die Reglerparameter zunächst ohne Bewertung eingestellt werden und erst danach eine geringere Gewichtung vorgegeben werden. Hierdurch wird erreicht, dass die Reglerparameter zunächst global anhand der Systemdynamik eingestellt werden und anschließend eine lokale Optimierung (z. B. für den eingeschwungenen Zustand) durchgeführt wird.

Parametriert wird diese lokale Optimierung mittels einer Kennlinie mit bis zu 4 Stützstellen. Negative Abweichungen werden durch Punktspiegelung der Kurve am Ursprung behandelt. Daraus ergeben sich insgesamt neun lineare Sektionen. Die Sektion s5 verläuft durch den Ursprung. Die Sektionen s1 und s9 haben einen Anstieg von 1 und laufen jeweils ins positive oder negative Unendliche. Wird der Eingang unbeschaltet gelassen, wird als Default-Wert eine 1-zu-1 Kennlinie verwendet, d. h. es wird keinerlei Gewichtung durchgeführt.

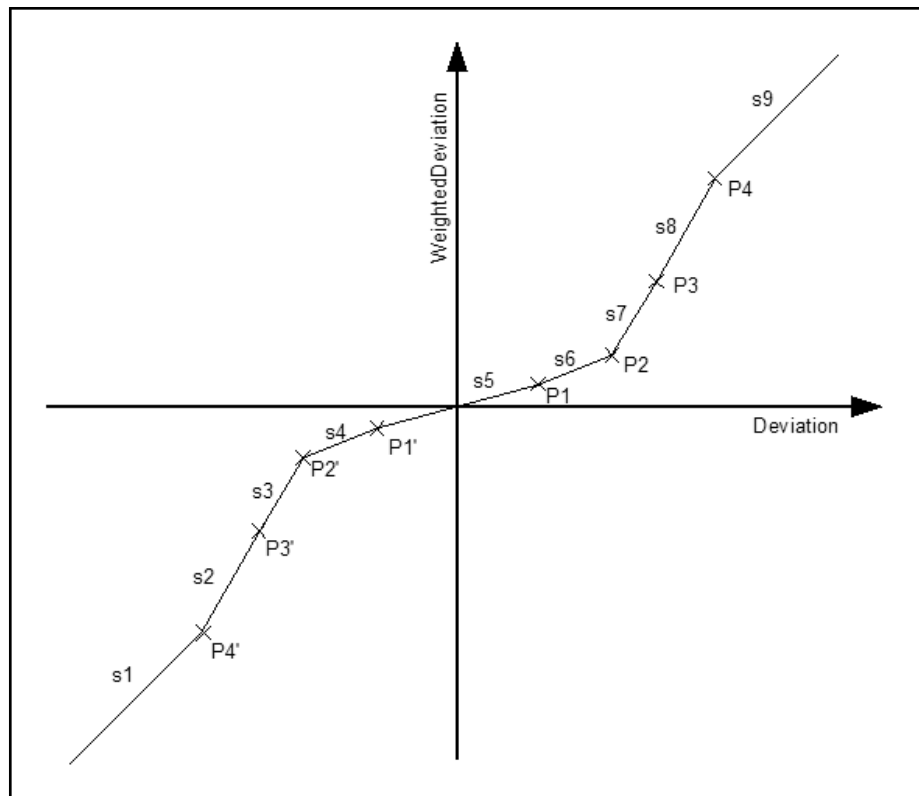


Abb. 5-44: Wichtung der Regelabweichung mit dem Eingang ResponseCurve

Die Punkte P1 bis P4 werden in einem Array aus IL_COORD_TYPE01 abgelegt. Werden weniger als vier Punkte benötigt, muss die X-Koordinate der nicht benötigten Punkte (vom letzten Punkt beginnend) auf 0 gesetzt werden. Die X-Werte müssen positiv sein und in positiver Richtung aufsteigen. Es ist erlaubt, denselben Punkt zweimal abzulegen.

MarkGating

Mit diesem Feature besteht die Möglichkeit, nur bei jeder n-ten Marke die Registerabweichung (Deviation) zu ermitteln, aber dennoch bei jeder Marke einen Regelungsschritt auszuführen. Man kann das z.B. verwenden, wenn die Marken auf dem Druckmedium mit einem Klischee gedruckt wurden, auf dem das gedruckte Bild mehrfach erscheint. In dieser Situation sind die Markenabstände zwischen aufeinander folgenden Marken nicht konstant. Die Markenabstände beispielsweise zwischen jeder n-ten aufeinanderfolgenden Marke sind jedoch konstant. Der Registerregler verwendet dann nur jede n-te Marke zur Ermittlung der Registerabweichung, führt aber bei jeder Marke einen Regelschritt aus, um die Registerabweichung über "n" Regelschritte verteilt auszuregeln.

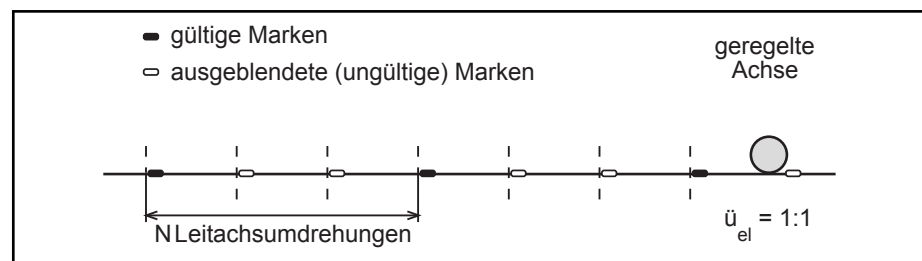


Abb. 5-45: MarkGating



Da die Registerabweichung über "n" Regelschritte verteilt ausge-regelt wird, sollte die Proportionalverstärkung des Reglers mit dem Wert 1/n beschaltet werden.

Normalbetrieb

Der RegisterController berechnet nur einen neuen Wert, wenn sich der Eingang "Step" verändert. Die Reglerparameter "PControl" und "IControl" können permanent geändert werden.

Wenn der Wert im "ControlValue"-Ausgang die Werte von "HighLimit" oder "LowLimit" über-, bzw. unterschreitet, dann wird "ControlValue" auf einen der folgenden Grenzwerte begrenzt.

- Ist der Wert in "ControlValue" positiv, wird der Wert in "HighLimit" verwendet.
- Ist der Wert in "ControlValue" negativ, wird der Wert in "LowLimit" verwendet.

Wenn der Registerregler eine Nachstellzeit hat, wird zusätzlich ein Anti-Reset-Windup durchgeführt und die Nachstellzeit wird nicht länger aufsummiert. In diesem Fall werden die Ausgänge "HighLimitActive" und "LowLimitActive" auf TRUE gesetzt.

Notwendige Parametrierung

MLC-Parametrierung

Wird eine **reale Achse mit Interpolation im Antrieb** als geregelte Achse ("ControlledAxis"-Eingang) benutzt, muss der verwendete Parameter im "ControlValueIDN"-Eingang einer der nachfolgenden unterstützten P-Parameter sein:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 und P-0-0695



Zusätzlich muss dieser P-Parameter im MDT des zyklischen Datenkanals im Antrieb konfiguriert sein.

Wird eine **reale Achse mit Interpolation in der Steuerung** oder eine **virtuelle Achse** als geregelte Achse benutzt, muss der verwendete Parameter im "ControlValueIDN"-Eingang einer der nachfolgenden unterstützten A-Parameter sein:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

MLD-Parametrierung

Der Parameter, der als geregelter Parameter für den "ControlValueIDN"-Eingang (P-Parameter) verwendet wird, muss in der "AxisData"-Struktur konfiguriert werden. Daher muss die Achsdatenstruktur aktiviert werden. Folgende P-Parameter werden unterstützt:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

Ab MLD Firmware MPx20 (IL2G) werden am Eingang "MeasuredAxis" auch Achsreferenzen unterstützt, die nicht in der "AxisData"-Struktur enthalten sind (> VMA1).

In diesem Fall wird das Korrekturfenster nicht über die aktuelle Position der "MeasuredAxis" ermittelt, sondern über P-0-0761 "Leitachseposition für Folgeachse" der lokalen Achse gebildet.

Damit im P-0-0761 "Leitachseposition für Folgeachse" die benötigte Positionsinformation zur Verfügung steht, muss der Leitachsgenerator entsprechend konfiguriert werden.

Beteiligte Parameter:

- P-0-0760 "Virtuelle Leitachse, Positionssteuerwort"

- P-0-0916 "Leitachsformatwandler Signalauswahl"

Siehe "Virtueller Leitachsgenerator" in der Dokumentation IndraDrive MPx-20 Funktionen Anwendungsbeschreibung, DOK-INDRV*-MP*-20VRS**-APRS-DE-P.

Fehlerbehandlung Der Funktionsbaustein gibt die Fehlermeldungen der folgenden unterlagerten Funktionsbausteine aus:

- [MB_RegiMeasuringType03, Seite 314](#)
- Fehlercodes Funktionsbaustein IL_PIType02, RIL_LoopControl
Bibliothek RIL_LoopControl (siehe Dokumentation "Rexroth IndraWorks 14VRS Basis-Bibliotheken IndraLogic 2G", DOK-IL*2G*-BASLIB**V14-LIRS-DE-P, R911343918)
- [MB_RegiRegulateType02, Seite 326](#)
- [MB_SetpointLockType01, Seite 350](#)

Er erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"Axis_Ref" liegt außerhalb des gültigen Wertebereichs (gemessen)
INPUT_RANGE_ERROR (16#0006)	16#0813	16#0001	"HighLimit" < "LowLimit"
INPUT_RANGE_ERROR (16#0006)	16#0813	16#0002	"PControl" < 0

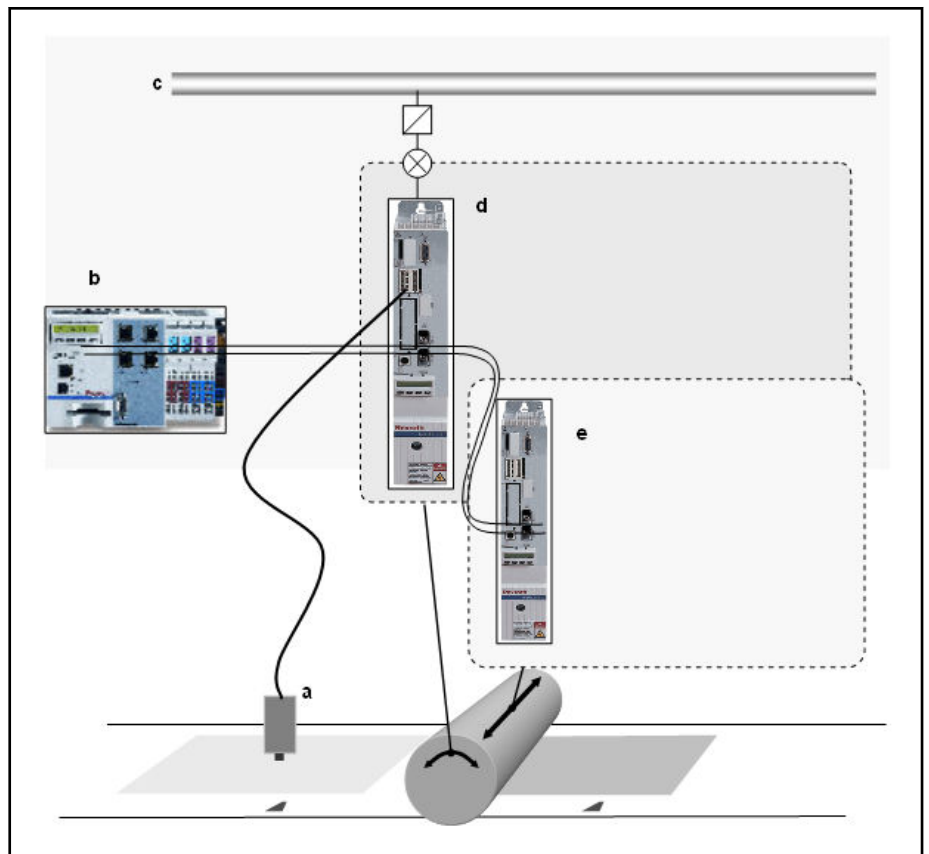
Tab. 5-48: Fehlercodes Funktionsbaustein MB_RegisterControllerType04

5.6.4 Konfiguration des Registerreglers

Überblick

Der Registerregler-FB MB_RegisterControllerType04 wird in Papier-, Druck-, Verpackungs- und Folienapplikationen eingesetzt, bei denen synchronisierte Antriebe (z.B. Winkel- oder Kurvenscheibensynchronisation), Material durch eine Maschine transportieren.

In diesen Applikationen beeinträchtigen Schwankungen in der Materialbeschaffenheit, rutschendes Material und der Produktionsprozess die Genauigkeit der Materialposition. Ein Sensor bestimmt die aktuelle Lage der Registermarken, die auf dem Material angebracht oder gedruckt sind und der Registerregler misst die Abweichung zum Sollwert. Er berechnet die notwendige Korrektur mithilfe eines Regelkreises. Durch diese Methode bleiben die Registermarken genau bei ihrem Sollwert und verschieben sich nicht. Die folgende Abbildung zeigt die Implementierung eines Registerreglers anhand eines Beispiels:



- a** Sensor
b: MLC-Regelung
c Leitachse (virtuelle Achse)
d Registerregelung, Messtasterfunktion, Winkelsynchronisation

Abb. 5-46: Beispiel eine Registerreglers bei einem MLC-basierten System

Einsatz der Registerregelung

Überblick

Die schnellste Möglichkeit, die Registerregelungsfunktionalität in einer Anwendung einzusetzen, ist die Kombination des Registerregler-FB mit einem Messtaster-FB. Der Messtaster-FB liefert den Istwert an den Registerregler. Der Registerregler berechnet den Korrekturwert auf Basis der aktuellen Abweichung zwischen dem Sollwert und dem Istwert.

Das Beispiel in diesem Abschnitt zeigt den Registerregler-FB MB_RegisterControllerType04 mit dem Funktionsbaustein MB_TouchProbeContinuous.

Konfiguration des Messtaster-FB

Die folgenden Messtastersignale müssen mithilfe des Funktionsbausteins MB_InitTouchProbe konfiguriert werden:

1. Antriebsmesstaster 1
2. Positiver Messtaster 1
3. Markenausfall aktiviert
4. Erwartungsfenster aktiviert



Die Beschreibung für den Messtaster-FB finden Sie in [MB_InitTouchProbe](#), Seite 127.

Konfiguration des Messtaster-FB und des RegisterControllerType04-FB

Die folgende Abbildung zeigt die Verbindung der Funktionsbausteine [MB_TouchProbeContinuous](#), Seite 138, und [MB_RegisterControllerType04](#), Seite 362, wenn die Messtasterfunktion durch eine positive Flanke ausgelöst wird.

Teil der Konfiguration des Funktionsbausteins [MB_InitTouchProbe](#), Seite 127, sind der aktivierte Markenausfall und das aktivierte Erwartungsfenster. Daher muss der Zähler für Markenausfälle auch verbunden werden.

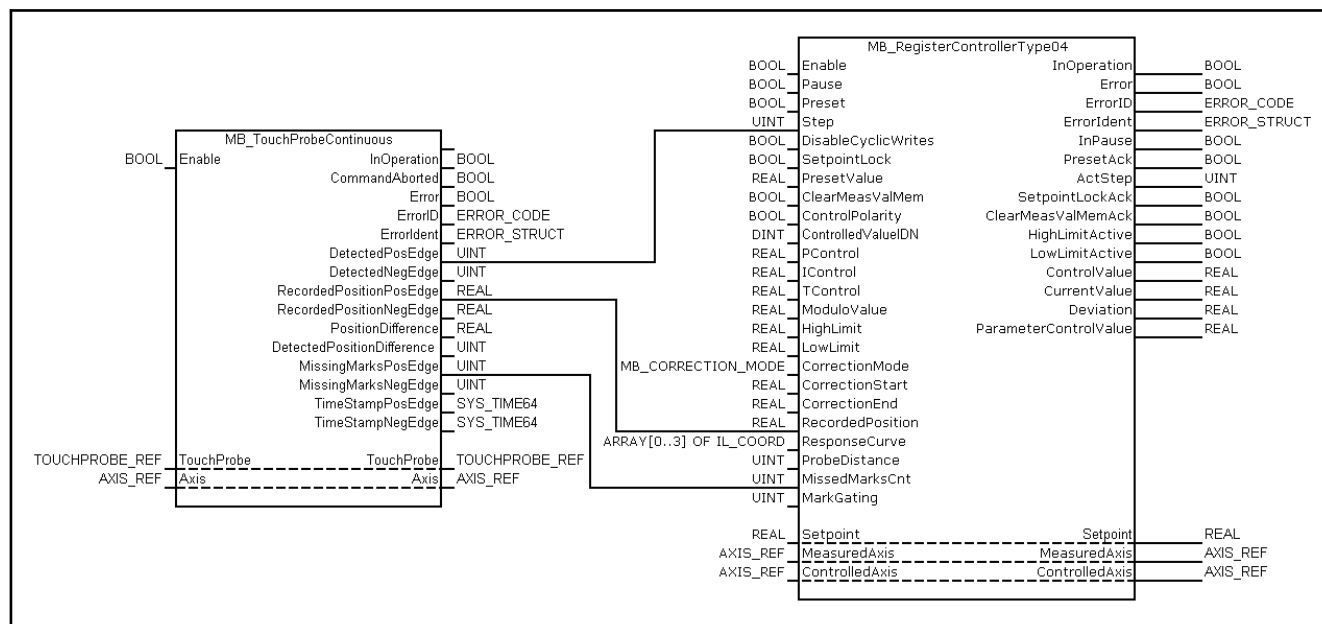


Abb. 5-47: Die Schnittstelle vom **MB_TouchProbeContinuous** und **MB_RegisterControllerType04**

Eine vollständige Beschreibung der Eingänge und Ausgänge finden Sie bei den einzelnen Beschreibungen der Funktionsbausteine.

In der obenstehenden Abbildung sind die folgenden Signalverbindungen beschrieben:

1. "DetectedPosEdge" zu "Step"
2. "RecordedPositionPosEdge" zu "RecordedPosition"
3. "MissingMarksPosEdge" zu "MissedMarksCnt"

Diese Verbindung ist nur notwendig wenn das Erwartungsfenster und der Zähler für Markenausfälle verwendet wird.

Parametrierung des Registerreglers

Achswichtung

Um die Registerregelungsfunktionalität zu verwenden, müssen beide Achsen (gemessene Achse, geregelte Achse) ins rotatorische Modulo-Format gesetzt werden. Die Menüauswahl bei IndraWorks wie folgt:

- Für die MLD, klicken Sie auf **IndraWorks ▶ Achse ▶ Wichtung / Mechanisches System ▶ Wichtung / Maßeinheiten**.
- Für die MLC, klicken Sie auf **IndraWorks ▶ Motion ▶ Virtuelle Achsen / Reale Achsen ▶ Achse ▶ Wichtung / Mechanisches System ▶ Wichtung / Maßeinheiten**.

Siehe als Beispiel folgende Abbildung:

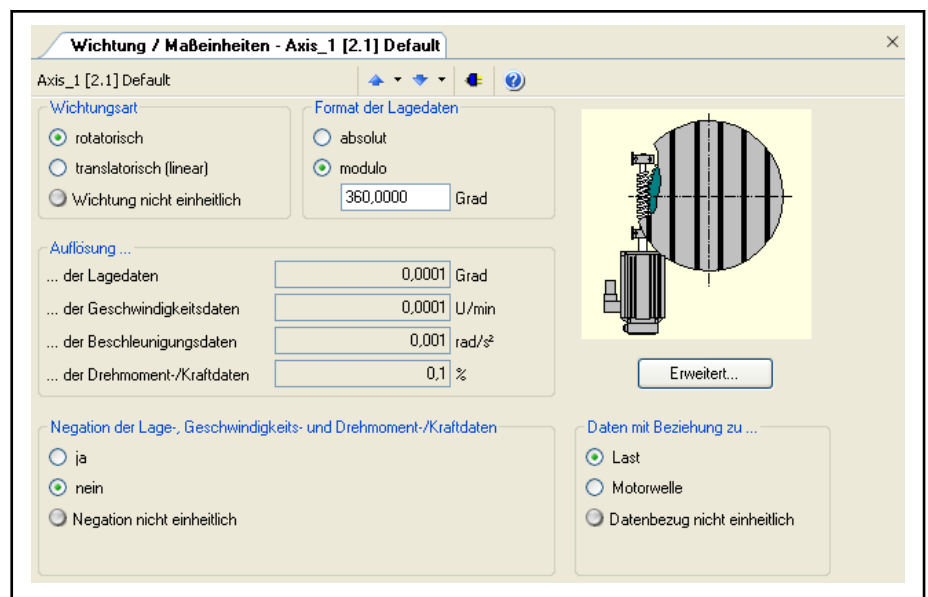


Abb. 5-48: Einstellen der Achswichtung

Konfiguration des MLC Master-Datentelegramms (MDT)

Der verwendete Parameter für den "ControlledValueIDN"-Eingang muss im MDT der geregelten Achse konfiguriert werden. Wählen Sie aus dem entsprechenden IndraWorks-Projekt **IndraWorks ▶ Motion**, dann Rechtsklick auf der geregelten Achse und wählen Sie **Kommunikation ▶ Zyklischer sercos Kanal**.

Siehe als Beispiel folgende Abbildung:

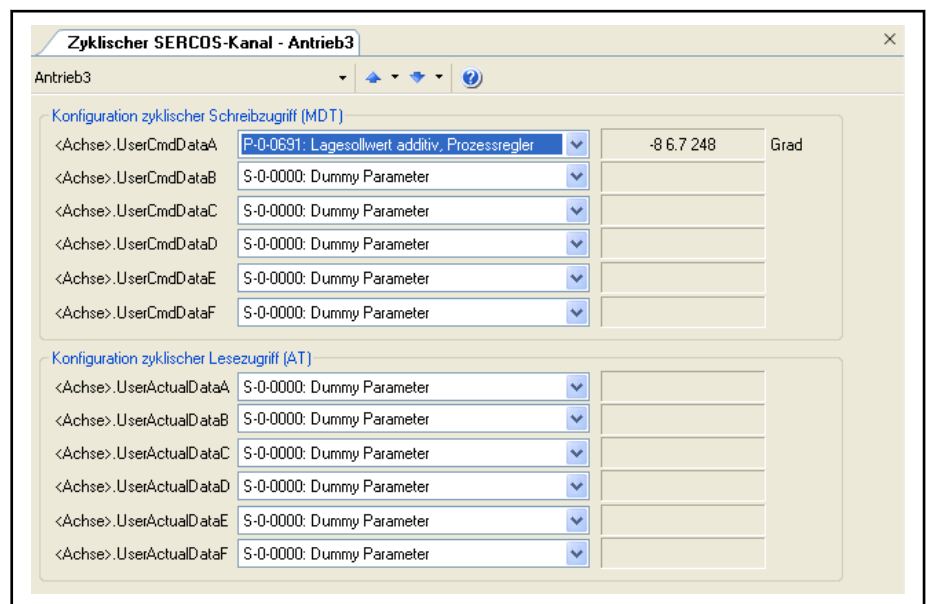


Abb. 5-49: MDT-Konfiguration der geregelten Achse (MLC)

MLD-Achsdatenkonfiguration

Der verwendete Parameter für den "ControlledValueIDN"-Eingang muss in der AxisData-Struktur der geregelten Achse konfiguriert werden. Wählen Sie aus dem entsprechenden IndraWorks-Projekt **IndraWorks ▶ MLD ▶ Konfiguration** und klicken Sie auf **Unterstützung der AxisData-Struktur**.

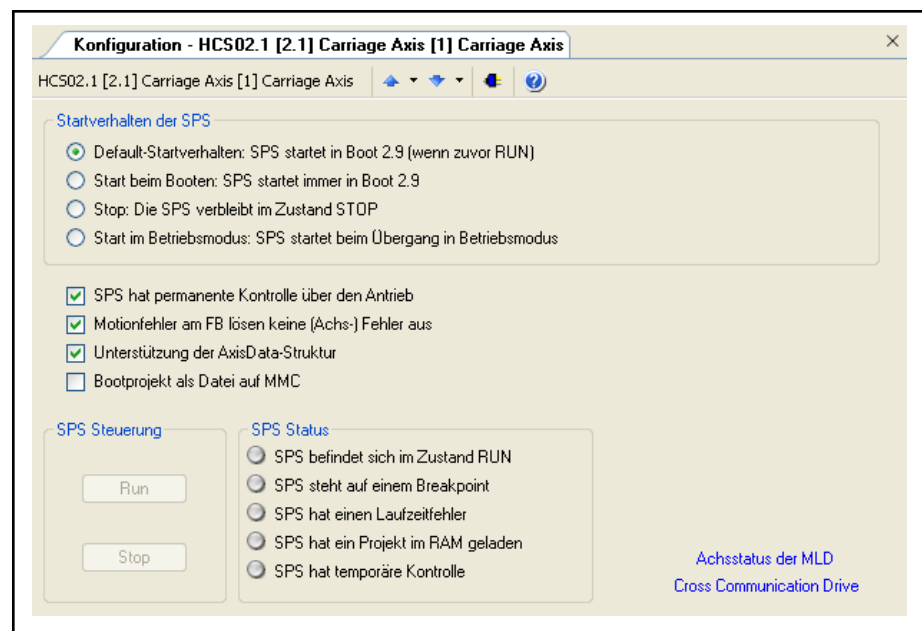


Abb. 5-50: MLD-Konfiguration

Anschließend klicken Sie auf **IndraWorks ► MLD ► Konfiguration ► AxisData** und wählen Sie den verwendeten Parameter für den "ControlledValueIDN"-Eingang aus.

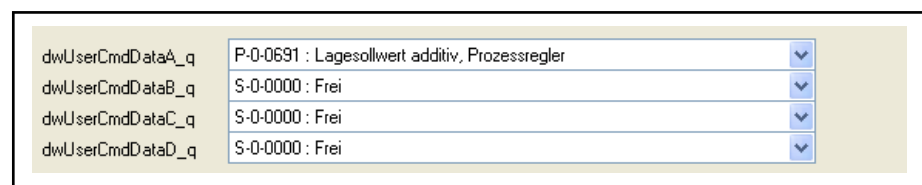


Abb. 5-51: AxisData-Konfiguration

5.6.5 Registerregler - Anwendungsbeispiele

Stanzen beim Etikettendruck

Das folgende Beispiel stellt einen Stanzvorgang an einer Etikettendruckmaschine dar:

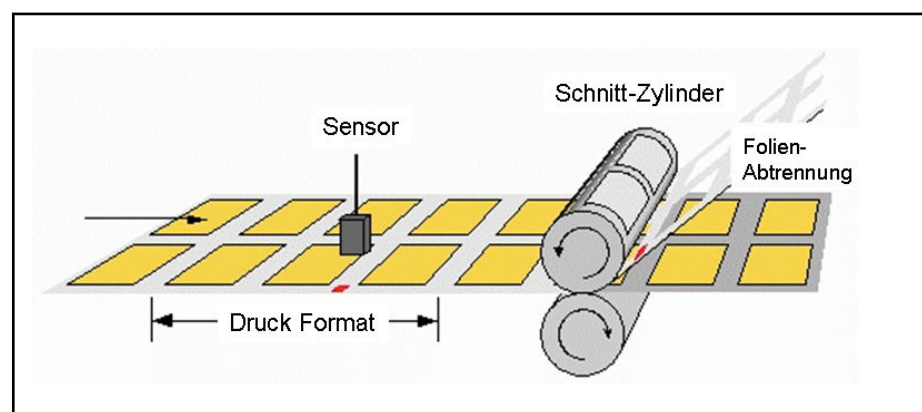


Abb. 5-52: Registerregelung einer Stanze

Etiketten werden auf einen Materialverbund einschließlich einer Verstärkungsfolie und einer selbstklebenden Decklage gedruckt. Das unbedruckte Raster wird eingeschlagen und umgewickelt.

Die Aufgabe des Registerreglers ist es, den Stanzzylinder auf die bedruckte Papierbahn einzustellen.

Der Stanzzylinder läuft im winkelsynchronen Betrieb. Die Ausrichtung auf die bedruckte Papierbahn erfolgt durch Anpassen des Winkelversatzes. Der Sensor ist am Antrieb des Stanzzylinders angeschlossen. Die Winkelposition (der Lageistwert) des Stanzzylinders wird über das Sensorsignal erfasst.

Der Positionierbefehl für den Registerregler entspricht der Zylinderposition, die exakt auf das Produkt ausgerichtet ist. Der Korrekturwert wird anhand der Differenz zwischen Referenz- und Istwert mit anschließender Addition zum aktuellen Winkelversatz berechnet.

Die wichtigsten Einstellungen für den Messtaster, den Registerregler und die Antriebsbetriebsart dieser Applikation sind:

- Gemessener Wert (Messtasterfunktion) = Lageistwert Geber 1 (S-0-0051)
- Geregelter Wert (ControlledValueIDN) = Lagesollwert additiv (P-0-0691)
- Antriebsbetriebsart = Winkelsynchronisation

Insetter-Regelung

Die folgende Abbildung stellt eine Insetter-Regelung in einer Materialzuführungsapplikation dar:

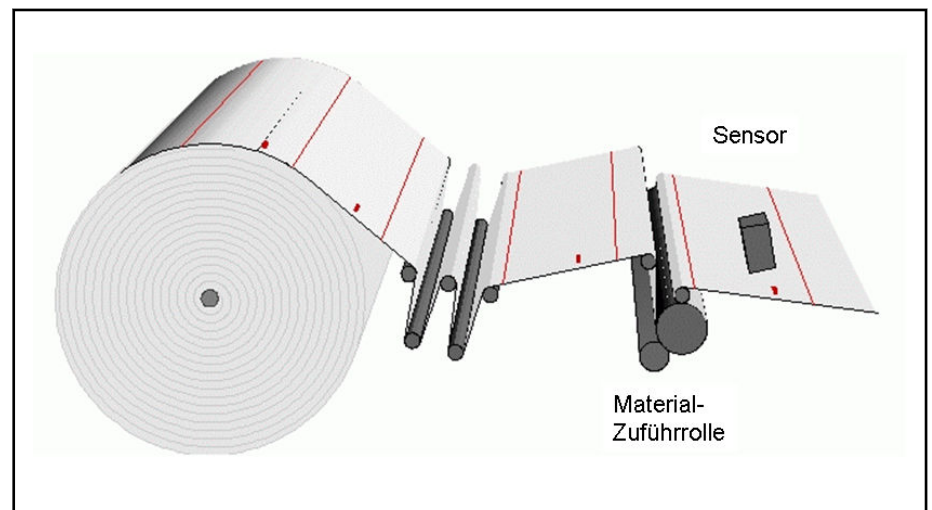


Abb. 5-53: Insetter-Regelung der Materialzuführung

Die Materialzuführungseinheit transportiert vorgedrucktes Material zu einer Druckmaschine. Nach dem Vordrucken und Trocknen ist die Papierbahn geschrumpft, d. h. die Formatlänge stimmt nicht exakt mit dem Nennformat überein. Die Bahn muss zur weiteren Bearbeitung wieder ausgedehnt werden.

Die Aufgabe des Registerreglers besteht darin, die Geschwindigkeit der Materialzuführungsrolle so zu regeln, dass die vorgedruckte Papierbahn auf das nominale Papierformat gedehnt wird. Dies setzt voraus, dass bei jeder Umdrehung der Leitachse jeweils ein Papierformat weitertransportiert wird.

Der Positionierbefehl des Registerregler bezieht sich auf die Leitachse.

Die Materialzuführungsrolle läuft im Geschwindigkeitssynchronisationsmodus. Die Geschwindigkeit der Zuführungsrolle wird über den Getriebeübersetzungs-Feinabgleich geregelt. Die Papierbahnausdehnung ist dann richtig, wenn die Druckmarke immer in der gleichen Leitachseposition unter dem Markenleser erkannt wird. Die Leitachseposition wird über das Sensorsignal erfasst. Der Markenleser ist am Messtastereingang des Materialzuführungsrollen-Antriebs angeschlossen.

Die wichtigsten Einstellungen für den Messtaster, den Registerregler und die Antriebsbetriebsart dieser Applikation sind:

- Gemessener Wert (Messtasterfunktion) = resultierende Leitachsisposition (P-0-0775)
- Geregelter Wert (ControlledValueIDN) = Getriebeübersetzungs-Feinabgleich (P-0-0694)
- Antriebsbetriebsart = Drehzahlsynchronisation

Schlauchbeutelmaschine

Die folgende Abbildung stellt eine Sideseal-Achse (Folieneinzugsachse) einer Schlauchbeutel-Applikation dar:

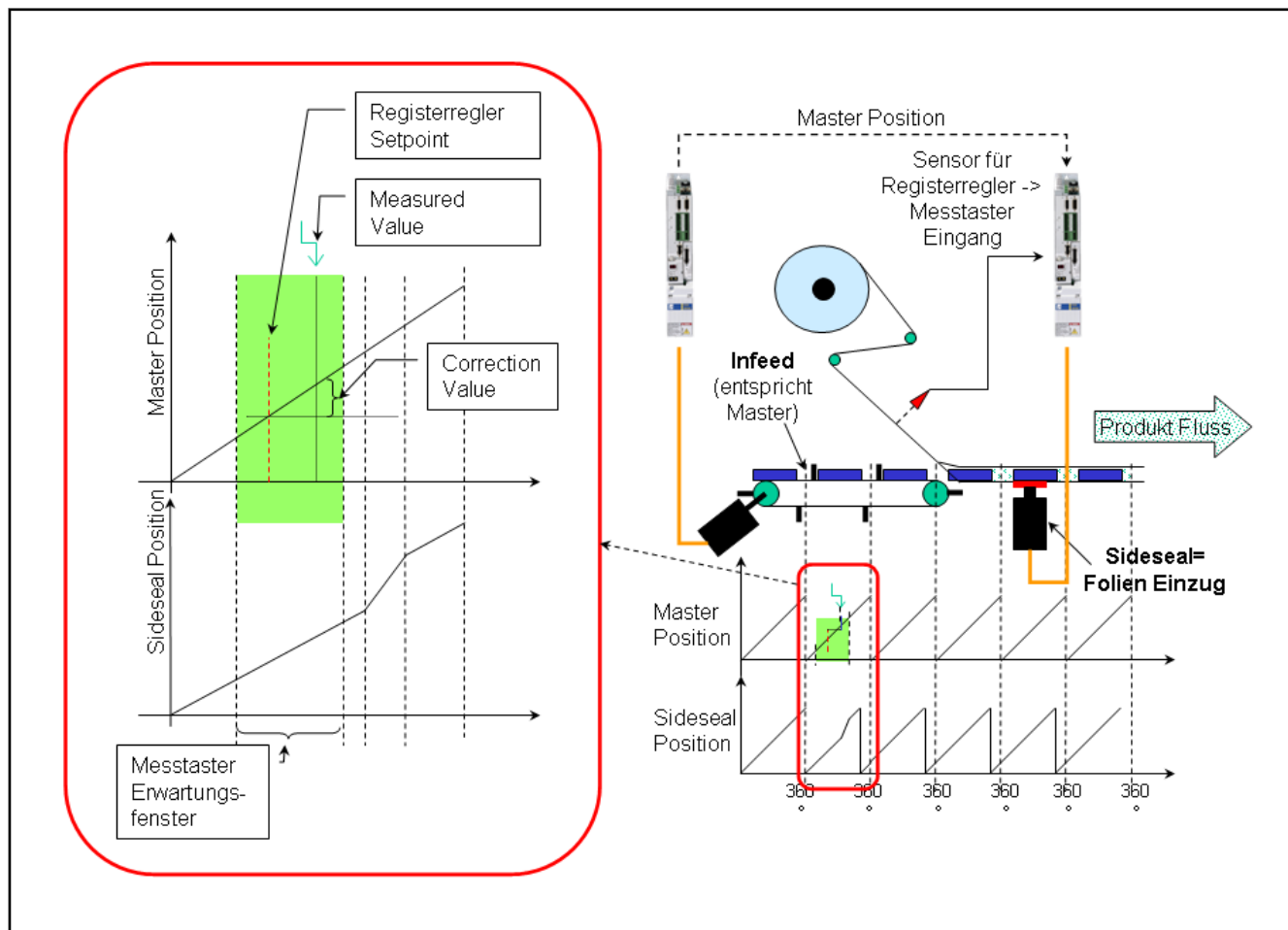


Abb. 5-54: Folieneinzugsachse einer Schlauchbeutel-Applikation

In dieser Applikation wird das Produkt von einem Eintaktband in die Verpackungsmaschine eingeführt. Die Folieneinzugsachse (Sideseal-Achse) läuft synchron mit dem Eintaktband und zieht die Folie ein. Dabei muss die korrekte Ausrichtung der Folie auf das Produkt und eine gleichbleibende Produktlänge gewährleistet sein.

Schlupf zwischen dem Material und den Einzugsrollen sowie Ungenauigkeiten im Material können Positionierfehler zwischen dem Servomotor und der Folie hervorrufen. Daher sind Registermarken auf das Material aufgedruckt, die einen Ausgleich von Positionierfehlern ermöglichen.

Bedruckung, Bilder und Schmutz auf dem Produkt können Störsignale am Druckmarkensensoreingang hervorrufen. Deshalb ist ein Erwartungsfenster für das Druckmarkensignal erforderlich. Die Antriebsmesstasterfunktion ist

nur innerhalb des Erwartungsfensters aktiv um die Druckmarkenposition zu erfassen. Des weiteren ist es notwendig, dass auch das Fehlen von Marken innerhalb des Erwartungsfensters erkannt wird.

Die Korrekturbewegung verhält sich proportional zur Differenz zwischen dem gemessenen und dem Sollwert. Aufgrund mechanischer Begrenzungen der Maschine muss die Korrekturbewegung auf einen nutzerdefinierten Wert begrenzt werden.

Der Registerregler muss die Folieneinzugsachse an die bedruckte Folie anpassen, um diese an die Produktposition anzupassen.

Die Folieneinzugsachse läuft im Winkelsynchronisationsmodus. Der Registerregler errechnet einen additiven Lagesollwert für die Folieneinzugsachse, durch den die Folie wieder korrekt ausgerichtet wird.

Die wichtigsten Einstellungen für den Messtaster, den Registerregler und die Antriebsbetriebsart dieser Applikation sind:

- Gemessener Wert (Messtasterfunktion) = resultierende Leitachsposition (P-0-0775)
- Geregelter Wert (ControlledValueIDN) = Lagesollwert additiv (P-0-0691)
- Antriebsbetriebsart = Winkelsynchronisation

5.7 Seitenregisterregler-FB

5.7.1 Einführung und Übersicht

Der Seitenregisterregler-FB ist in den Bibliotheken ML_TechRegi (MLC) und MX_TechRegi.lib (MLD) enthalten.

5.7.2 MB_RegisterControllerSideType01

Kurzbeschreibung Der Registerregler-FB MB_RegisterControllerSideType01 unterstützt die folgende Funktionalität:

- Berechnung eines Korrekturwerts auf der Grundlage sowohl gemessener Werte als auch Sollwerte unter Verwendung eines P- oder PI-Regelkreises
- Preset-Funktion
- Pause-Funktion
- Min/Max-Begrenzung des berechneten Stellwerts
- Sollwertübernahme-Funktion
- Totzone K_p Kennlinie
- PT1-Element

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechRegi
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechRegi.lib

Tab. 5-49: Referenztable MB_RegisterControllerSideType01

Schnittstellenbeschreibung

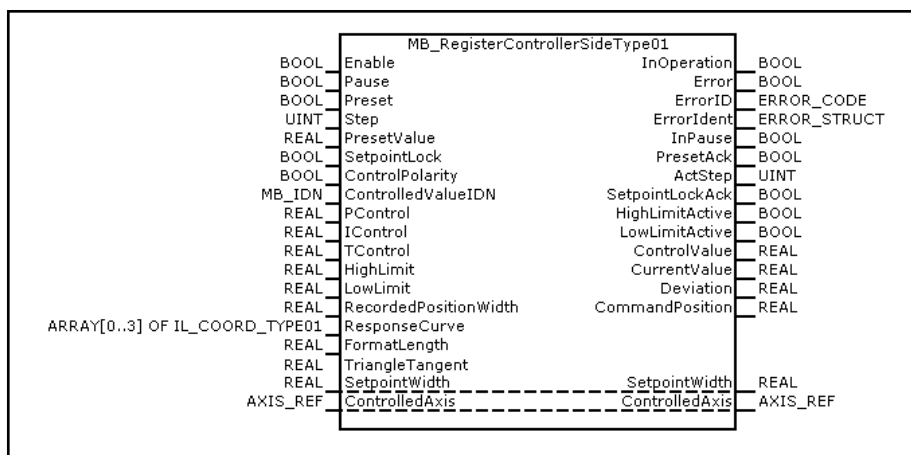


Abb. 5-55: Funktionsbaustein MB_RegisterControllerSideType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Der Funktionsbaustein ist aktiv, solange der Eingang TRUE ist
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, solange der "Enable"-Eingang aktiv ist. Die Ausgänge "ControlValue" und "CommandPosition" sind konstant, solange "Pause" aktiv ist
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, solange "Enable" TRUE ist. Eine positive Flanke dieses Eingangs setzt den Regelkreis zwangsweise auf "Preset-Value"
	Step	UINT	Verarbeitungsschritt
	PresetValue	REAL	Wert an den Ausgang "ControlValue" bei steigender Flanke am "Preset"-Eingang übergeben
	SetpointLock	BOOL	Aktiviert die Sollwertübernahme. Wenn TRUE, wird die nächste gemessene Marke gelatcht
	ControlPolarity	BOOL	Die Polarität des Stellwerts wird invertiert, solange dieser Eingang TRUE ist
	ControlledValueIDN	MB_IDN	Der folgende IDN-Regelung wird unterstützt: S-0-0258
	PControl	REAL	Proportionalverstärkung des PI-Reglers. Bei Einstellung 0 wird der proportionale Anteil des Reglers deaktiviert und die proportionale Verstärkung intern auf 1 gesetzt, sodass er als I-Regler arbeitet
	IControl	REAL	Nachstellzeit T_n des PI-Reglers. Bei der Einstellung auf 0 wird der Nachstellanteil des PI-Reglers deaktiviert
	TControl	REAL	Filterkonstante für den Eingang "RecordedPosition-Width"
	HighLimit	REAL	Maximalwert für "ControlValue"
	LowLimit	REAL	Minimalwert für "ControlValue"
	RecordedPosition-Width	REAL	Breite der gemessenen Marke (in technischen Einheiten)

E/A-Typ	Name	Datentyp	Beschreibung
	ResponseCurve	ARRAY[0..3] OF IL_COORD_TYPE01	Kennlinie
	FormatLength	REAL	Formatlänge in mm
	TriangleTangent	REAL	Wert von $\tan \alpha = (\text{Länge der Gegenkathete} / \text{Länge der Ankathete})$
VAR_OUTPUT	InOperation	BOOL	Seitenregisterregler ist in Betrieb
	Error	BOOL	Zeigt einen Fehler an Fehler mit "Enable" = FALSE löschen
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 387
	InPause	BOOL	"Pause" ist aktiv. Die Ausgänge "ControlValue" und "Command" behalten ihre Werte
	PresetAck	BOOL	"Preset" ist abgeschlossen. Der Preset-Wert wird in den Ausgang "ControlValue" kopiert
	ActStep	UINT	Schrittwert; zählt mit jedem neuen Wert am Ausgang "ControlValue" hoch. Hinweis: Der Prozessregler wird angehalten und der Ausgang "ActStep" zählt immer noch mit jedem neuen Wert am Eingang "Step" hoch
	SetpointLockAck	BOOL	Sollwert wird übernommen. Der aktuelle Wert in "RecordedPositionWidth" wird nach "SetpointWidth" kopiert
	HighLimitActive	BOOL	Obergrenze ist aktiv. "ControlValue" wäre größer, ist aber auf "HighLimit" beschränkt
	LowLimitActive	BOOL	Untergrenze ist aktiv. ControlValue wäre niedriger, ist aber auf "LowLimit" beschränkt
	ControlValue	REAL	Vom Registrierregler errechneter Stellwert
	CurrentValue	REAL	Aufgezeichnete Lage, die aktuell in der Registerreglerseite verwendet wird. Hinweis: Umgewandelter Wert „RecordedPositionWidth“
	Deviation	REAL	Differenz der Sollwertseite (umgewandelt "SetpointWidth") - "CurrentValue" des Seitenregisterreglers (mm)
	CommandPosition	REAL	Berechnete Sollposition für den Antrieb
VAR_IN_OUT	SetpointWidth	REAL	Wert der Sollwertbreite (in technischen Einheiten)
VAR_IN_OUT CONSTANT	ControlledAxis	AXIS_REF	Bezug zur geregelten Achse

Tab. 5-50: Schnittstellenvariablen Funktionsbaustein MB_RegisterControllerSideType01

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	"Enable" TRUE (kontinuierlich)
Step	UINT	0	65335	0	Kontinuierlich. Beginnt bei steigender Flanke an "Enable" und fallender Flanke an "Pause"
PresetValue	REAL	n.def	n.def	0.0	Steigende Flanke an "Preset"
SetpointLock	BOOL	n.def	n.def	FALSE	Kontinuierlich
ControlPolarity	BOOL			FALSE	Kontinuierlich
ControlledValueIDN	MB_IDN	n.def	n.def	0	Steigende Flanke an "Enable"
PControl	REAL	0.0	n.def	1.0	Kontinuierlich
IControl	REAL	0.0	n.def	0.0	Kontinuierlich
TControl	REAL	0	n.def	0.0	Kontinuierlich
HighLimit	REAL	LowLimit	n.def	0.0	Kontinuierlich
LowLimit	REAL	n.def	HighLimit	0.0	Kontinuierlich
RecordedPositionWidth	REAL	n.def	n.def	0.0	Änderung von "Step"
ResponseCurve	ARRAY[0..3] OF IL_COORD	n.def	n.def	0.0	Kontinuierlich
FormatLength	REAL	0.0	n.def	0.0	Steigende Flanke an "Enable"
TriangleTangent	REAL	0.0	n.def	1.0	Steigende Flanke an "Enable"

Tab. 5-51: Min.- Max.- und Default-Werte der Eingänge

Zeitdiagramm

Das folgende Diagramm zeigt den Signal-Zeit-Verlauf des Funktionsbausteins MB_RegisterControllerSideType01 einschließlich der Funktionen "Pause" und "Preset".

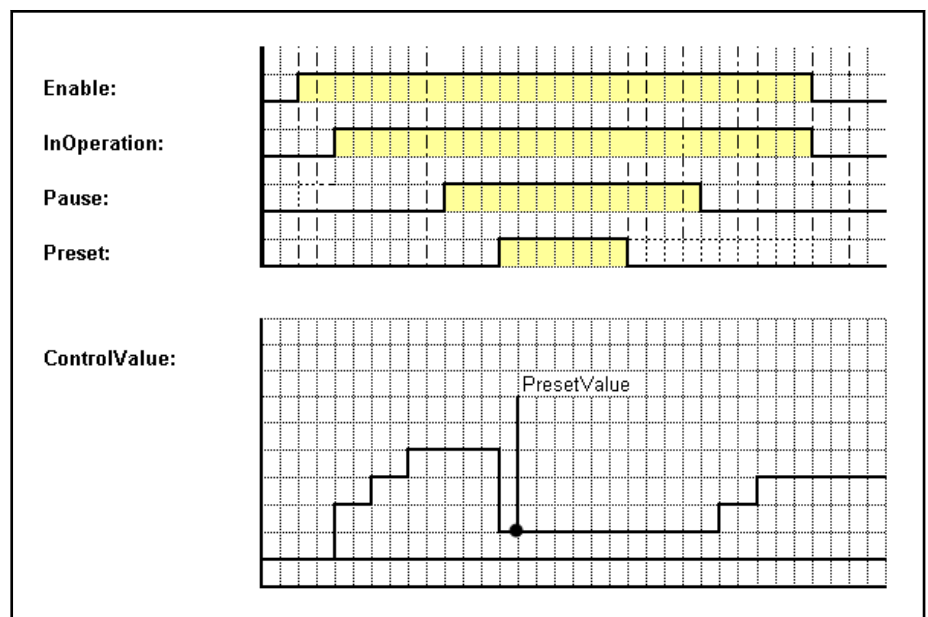


Abb. 5-56: Zeitdiagramm des MB_RegisterControllerSideType01 mit den Funktionen "Pause" und "Preset"

Funktionsbeschreibung

Initialisierung Während der Initialisierung müssen die Eingänge "SetpointWidth" und "RecordedPositionWidth" über gültige Daten verfügen.

PresetValue Für einen fehlerfreien Start der Reglerausgänge "ControlValue" und "CommandPosition", muss der "Preset"-Eingang gesetzt sein, bevor der "Enable"-Eingang aktiviert wird. Während dieser Zeit müssen die Eingänge "SetpointWidth" und "RecordedPositionWidth" über gültige Daten verfügen. Zudem werden die oberen und unteren Grenzwerte überwacht, um eine Begrenzung des Ausgangs "ControlValue" sicherzustellen, wenn der aktuelle Wert für "PresetValue" außerhalb des Gültigkeitsbereiches von "HighLimit" bzw. "LowLimit" liegt.

Der "PresetAck"-Ausgang ist aktiv wenn der Wert im Eingang "PresetValue" in den Ausgang "ControlValue" kopiert wird. Der "Preset"-Eingang hat eine höher Priorität als der "Pause"-Eingang. Der Wert in "PresetValue" wird in den entsprechenden Ausgang bei steigender Flanke am "Preset"-Eingang kopiert wenn "Enable" aktiv ist. Wenn dies erfolgreich ist, wird der Ausgang "PresetAck" auf TRUE gesetzt.

Pause Der Eingang "Pause" wird nur ausgewertet, solange der Eingang "Enable" aktiv ist. In diesem Fall wird die Abweichung ("SetpointWidth" – "RecordedPositionWidth") auf 0 gesetzt. Dadurch ändern sich die Ausgänge "ControlValue" und "CommandPosition" nicht mehr. Diese kann verwendet werden, um die Werte der Ausgänge "ControlValue" und "CommandPosition" beizubehalten. Der Ausgang "ActStep" wird selbst dann aktualisiert wenn der Eingang "Pause" aktiv ist. Solange der Eingang "Pause" aktiv ist, ist der Eingang "InPause" TRUE.

Glättung Der Eingang "TControl" ist die Dämpfungskonstante für den Eingang "RecordedPositionWidth". Wenn er auf 0 gesetzt wird, ist die Glättung inaktiv.

ResponseCurve Mit Hilfe des Eingang "ResponseCurve" kann die Regelabweichung gewichtet werden. Durch diese Funktionalität kann beispielsweise bei kleinen Regelabweichungen eine geringere Gewichtung der Regelabweichung eingestellt werden. Dies ergibt quasi eine geringere Proportionalverstärkung und dadurch ein ruhigeres Regelverhalten im eingeschwungenen Zustand. Bei der Parametrierung des Eingangs sollten die Reglerparameter zunächst ohne

Bewertung eingestellt werden und erst danach eine geringere Gewichtung vorgegeben werden. Hierdurch wird erreicht, dass die Reglerparameter zunächst global anhand der Systemdynamik eingestellt werden und anschließend eine lokale Optimierung (z. B. für den eingeschwungenen Zustand) durchgeführt wird.

Parametriert wird diese lokale Optimierung mittels einer Kennlinie mit bis zu 4 Stützstellen. Negative Abweichungen werden durch Punktspiegelung der Kurve am Ursprung behandelt. Daraus ergeben sich insgesamt neun lineare Sektionen. Die Sektion s5 verläuft durch den Ursprung. Die Sektionen s1 und s9 haben einen Anstieg von 1 und laufen jeweils ins positive oder negative Unendliche. Wird der Eingang unbeschaltet gelassen, wird als Default-Wert eine 1-zu-1 Kennlinie verwendet, d. h. es wird keinerlei Gewichtung durchgeführt.

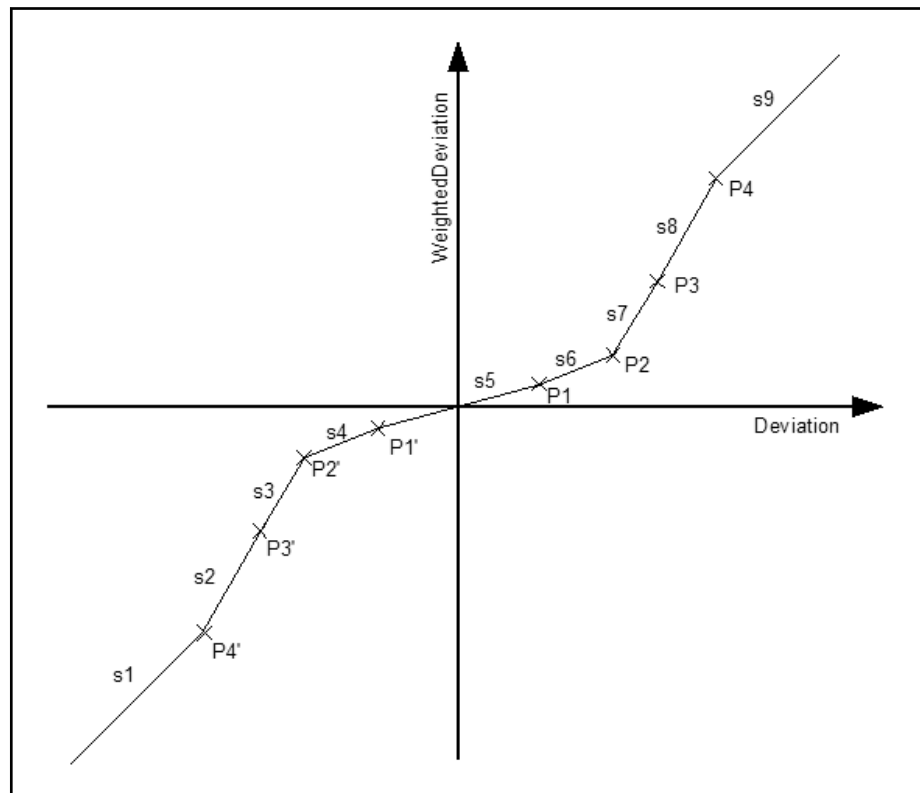


Abb. 5-57: Wichtung der Regelabweichung mit dem Eingang ResponseCurve

Die Punkte P1 bis P4 werden in einem Array aus IL_COORD_TYPE01 abgelegt. Werden weniger als vier Punkte benötigt, muss die X-Koordinate der nicht benötigten Punkte (vom letzten Punkt beginnend) auf 0 gesetzt werden. Die X-Werte müssen positiv sein und in positiver Richtung aufsteigen. Es ist erlaubt, denselben Punkt zweimal abzulegen.

FormatLength	Formatlänge (Leitachsenumdrehungen, Formate) in mm.
TriangleTangent	Die Marken zwischen Gegenkathete und Ankathete ($\tan \alpha$).

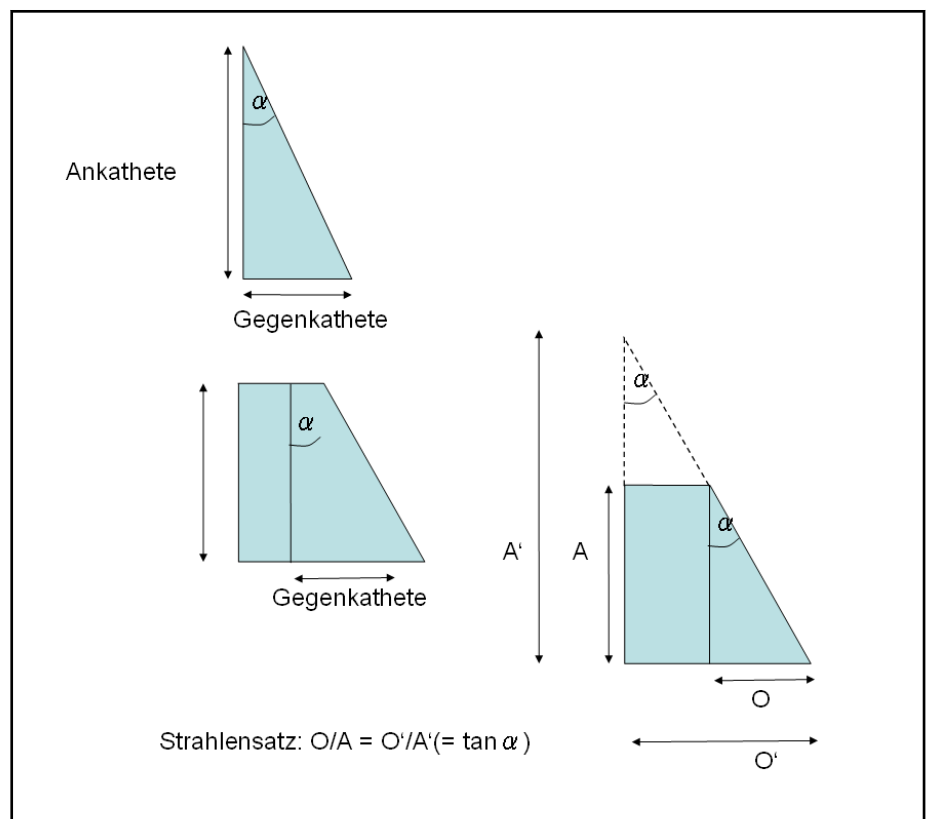


Abb. 5-58: Rechtwinklige Dreiecksmarke und konisch abgestumpfte Kegelmärke

Normalbetrieb

Die Registerreglerseite berechnet nur einen neuen Wert wenn sich der Eingang "Step" verändert. Die Reglerparameter "PControl", "IControl" und "TControl" können permanent geändert werden.

Wenn der Wert im "ControlValue"-Ausgang die Werte von "HighLimit" oder "LowLimit" über-, bzw. unterschreitet, dann wird "ControlValue" auf einen der folgenden Grenzwerte begrenzt.

- Ist der Wert in "ControlValue" positiv, wird der Wert in "HighLimit" verwendet
- Ist der Wert in "ControlValue" negativ, wird der Wert in "LowLimit" verwendet

Wenn die Registerreglerseite eine Nachstellzeit hat, wird zusätzlich ein Anti-Reset-Windup durchgeführt und die Nachstellzeit wird nicht länger aufsummiert. In diesem Fall werden die Ausgänge "HighLimitActive" und "LowLimitActive" auf TRUE gesetzt.

Fehlerbehandlung

Der Funktionsbaustein gibt die Fehlermeldungen der folgenden unterlagerten Funktionsbausteine aus.

- [Fehlercodes Funktionsbaustein MB_RegiMeasuringType01, Seite 310](#)
- IL_PIType02, RIL_LoopControl
Bibliothek RIL_LoopControl (siehe Dokumentation "Rexroth IndraWorks 14VRS Basis-Bibliotheken IndraLogic 2G", DOK-IL*2G*-BASLIB**V14-LIRS-DE-P, R911343918)
- [Fehlercodes Funktionsbaustein MB_RegiRegulateType01, Seite 325](#)
- [Fehlercodes Funktionsbaustein MB_RegiSetpointLockType01, Seite 352](#)

Er erzeugt in Additional1/Additional2 für die Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

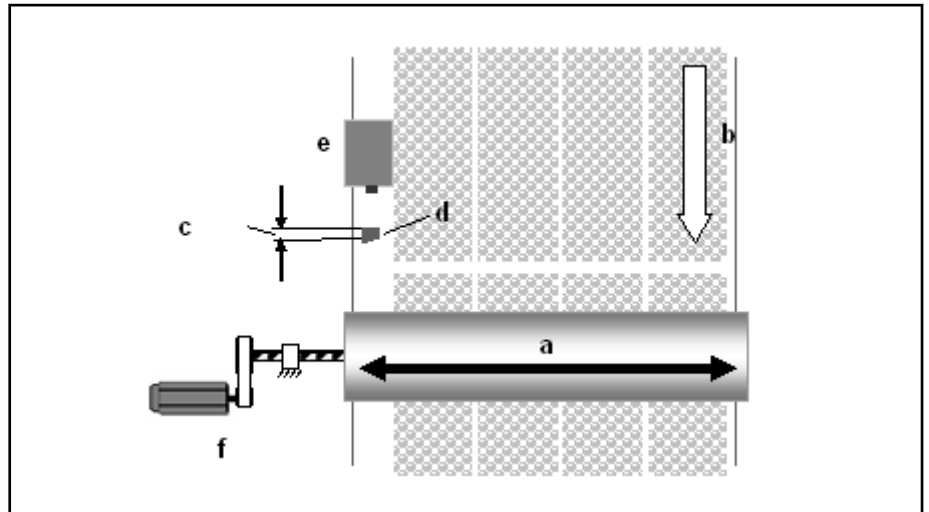
ErrorID	Additional1	Additional2	Beschreibung
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Die Antriebsfirmware wird nicht unterstützt
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo von "ControlledAxis" ist außerhalb des gültigen Wertebereichs
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Achse hat keine Stromversorgung "ControlledAxis"
RESOURCE_ERROR (16#0003)	16#0009	16#0000	"Axis_Ref" hat "ControlledAxis" geändert
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0001	Ausgewählte "ControlledValueIDN" wird nicht unterstützt
INPUT_RANGE_ERROR (16#0006)	16#0813	16#0001	"HighLimit" < "LowLimit"
INPUT_RANGE_ERROR (16#0006)	16#0813	16#0002	"PControl" < 0
INPUT_RANGE_ERROR (16#0006)	16#0816	16#0001	Eingang "SetpointWidth" < 0
INPUT_RANGE_ERROR (16#0006)	16#0816	16#0002	Eingang "RecordedPositionWidth" < 0
INPUT_RANGE_ERROR (16#0006)	16#0816	16#0003	"TriangleTangent"-Eingang ist außerhalb des gültigen Wertebereichs
INPUT_RANGE_ERROR (16#0006)	16#0816	16#0004	"FormatLength"-Eingang ist außerhalb des gültigen Wertebereichs
INPUT_INVALID_ERROR (16#0001)	16#0C01	16#0001	"LowLimit" > "HighLimit"
INPUT_INVALID_ERROR (16#0001)	16#0C01	16#0002	Auswahl des Reglertyps wird nicht unterstützt. Eingänge "IControl" und "PControl" überprüfen
INPUT_RANGE_ERROR (16#0006)	16#0C02	16#0001	Eingang "TControl" ist ungültig
INPUT_RANGE_ERROR (16#0006)	16#0C02	16#0003	Eingang "IControl" ist ungültig
INPUT_RANGE_ERROR (16#0006)	16#0C02	16#0004	Eingang "PControl" ist ungültig
INPUT_INVALID_ERROR (16#0001)	16#0C05	16#0001	X-Werte in "ResponseCurve" nicht aufsteigend
INPUT_INVALID_ERROR (16#0001)	16#0C05	16#0002	Mehr als einen Wert für eine X-Position in "ResponseCurve"
INPUT_INVALID_ERROR (16#0001)	16#0C05	16#0003	Negativer X-Wert in "ResponseCurve"
ACCESS_ERROR (16#0004)	16#0815	16#0008	Die geregelte Achse ist eine MLD-M Folgeachse, aber der Parameter P-0-1367, Bit 6 ist nicht TRUE

Tab. 5-52: Fehlercodes Funktionsbaustein MB_RegisterControllerSideType01

5.7.3 Konfiguration des Seitenregisters

Überblick Der Seitenregister ist eine Funktion zur axialen Anpassung der rotierenden Zylinder. Typisches Beispiel ist die Anpassung eines Druckzylinders für eine vorgedruckte Papierbahn.

Die Dreiecksmarke ist der axiale Bezug auf dem gedruckten Image. Ein optischer Sensor registriert die eingehende und ausgehende Flanke der Marke. Der Unterschied zwischen beiden Flanken ist die Breite der Marke. Der Wert ist der Istwert des Seitenregisters. Der Unterschied zwischen der Istbreite der Marke und der Sollbreite wird in die Zielposition des Servomotors überführt.

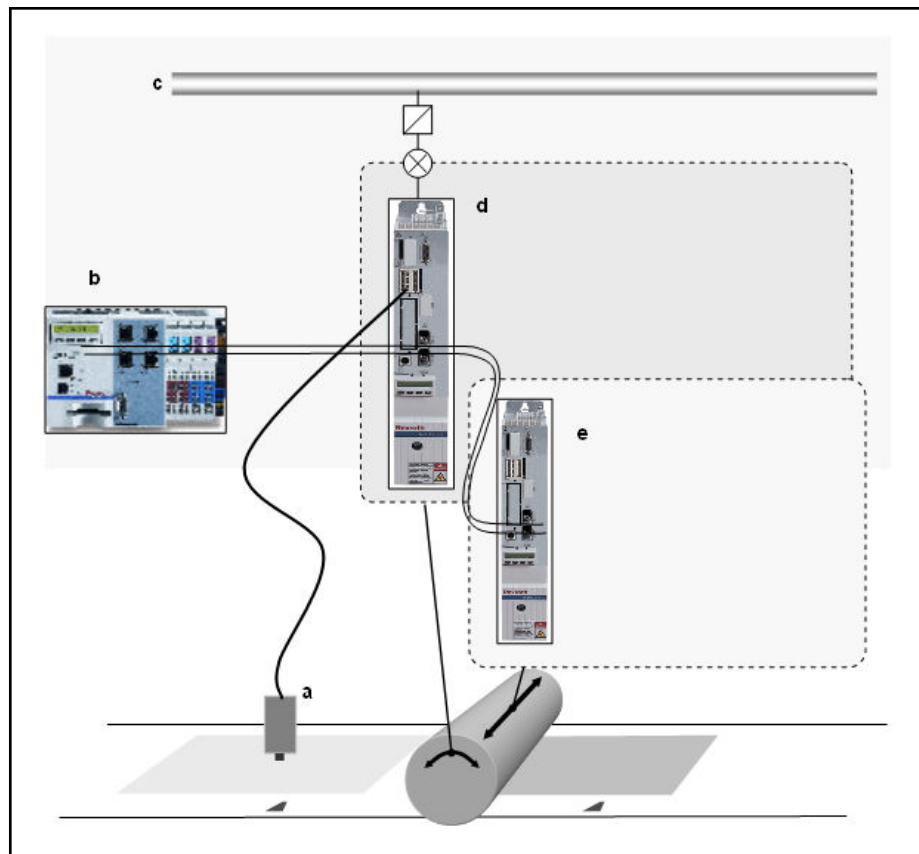


- a: Seitenregister
- b: Bahnaufrichtung
- c: Gemessener Wert
- d: Registermarke
- e: Sensor
- f: Servomotor

Abb. 5-59: Seitenregisterkonfiguration

Verwendung der Seitenregisterregelung

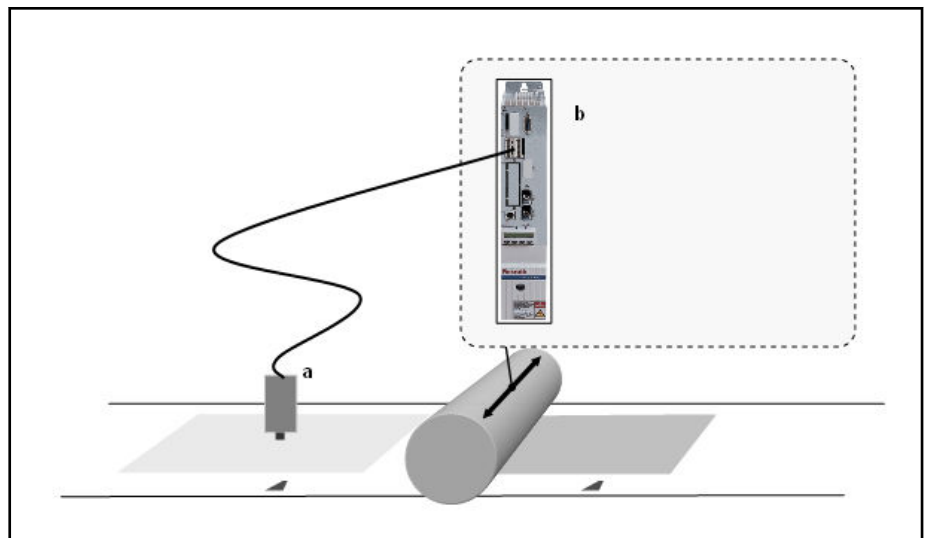
Die nachfolgende Abbildung zeigt eine mögliche Konfiguration eines Seitenregisterreglers für eine MLC.



- a:** Sensor
b: MLC
c: Leitachse
d: Registerregler, Messtasterfunktion, Winkelsynchronisation
e: Seitenregisterregler, absolutes Positionieren

Abb. 5-60: Überblick über Registerregler und Seitenregisterregler auf der MLC

Die nachfolgende Abbildung zeigt die Konfiguration eines Seitenregisterreglers für eine MLD.



a: Sensor

b: MLD: Messtasterfunktion, Seitenregisterregler, absolutes Positionieren

Abb. 5-61: Überblick über den Seitenregisterregler in einer MLD



Die axiale Achse muss eine absolute Linearachse sein.

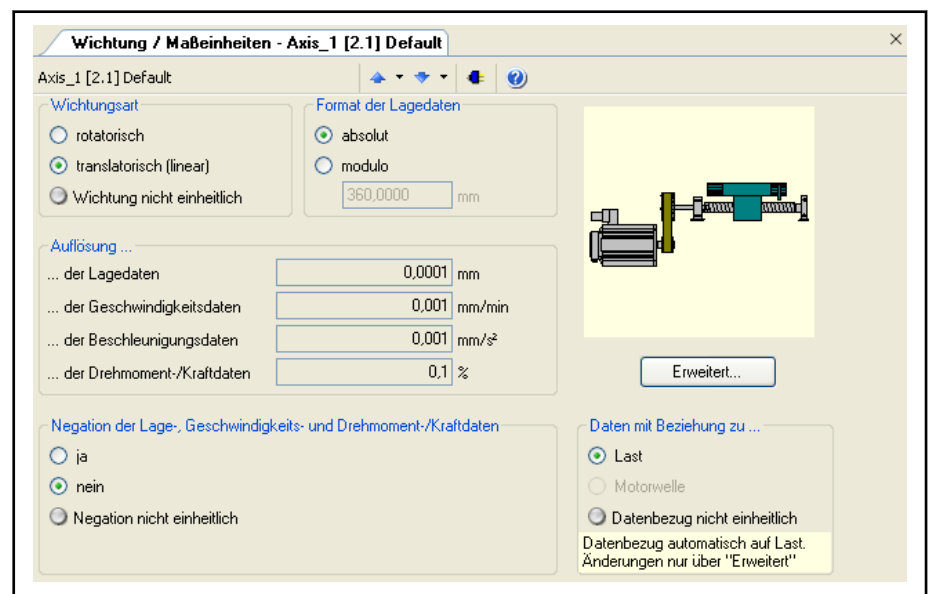


Abb. 5-62: Wichtung der geregelten Achse des Seitenregisterreglers

Beispielkonfiguration

Die schnellste Möglichkeit eine Seitenregisterreglerfunktionalität in einer Anwendung einzusetzen, ist die Kombination des Messtaster-FB mit dem AchsInterface. Der Messtaster-FB liefert den aktuellen Wert an den Seitenregister. Der Seitenregisterregler berechnet die Zielposition für die Achse. Der Seitenregister verfügt über keine interne Funktionalität um die Lage auf einen Sollwertlageparameter der Achse zu schreiben. Daher wird das AchsInterface verwendet, um die Sollage auf den entsprechenden Achsparameter zu schreiben.

Dieses Beispiel zeigt den Seitenregisterregler [MB_RegisterControllerSide-Type01](#), Seite 381, mit dem Funktionsbaustein MB_TouchProbeContinuous,

ML_TechMotionBase.library, und den Funktionsbausteinen des AchsInterface.

1. Ausgang "PositionDifference" zu Eingang "RecordedPositionWidth"
2. Ausgang "DetectedPositionDifference" zu Eingang "Step"

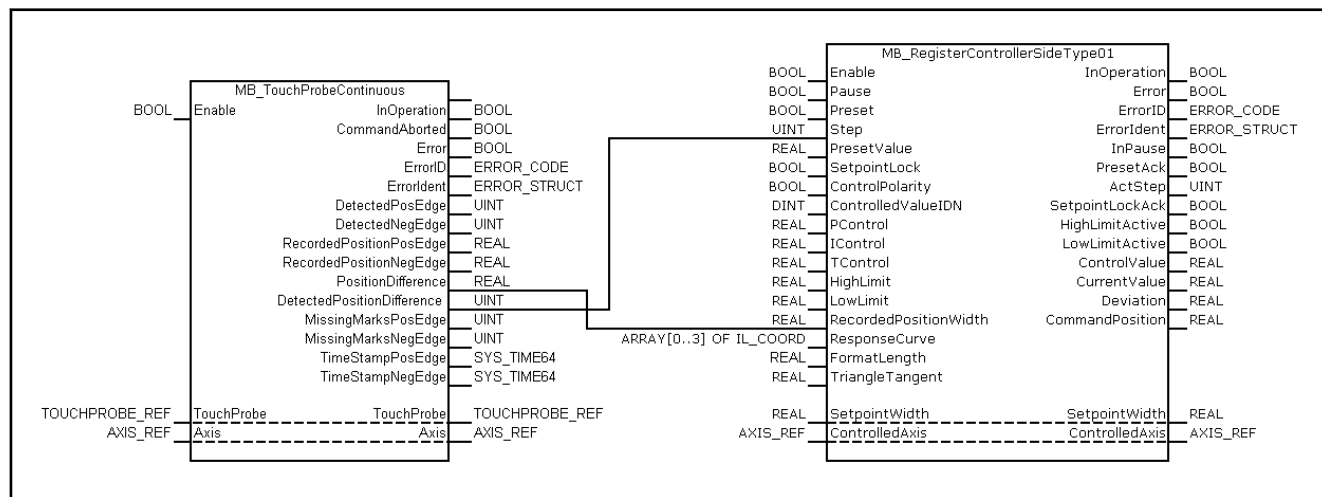


Abb. 5-63: Schematische Darstellung der FB MB_TouchProbeContinuous und MB_RegisterControllerSideType01

3. Weisen Sie "CommandPosition" dem Array arAxisCtrl[ControlledAxis].PosMode.Position zu



AchsInterface-Einstellungen

- Betriebsmodus: Absolutes Positionieren
- Folgendes Kommando aktiviert den Betriebsmodus:
`arAxisCtrl[ ].Admin._OpMode := ModePosAbs; bzw. arAxisCtrl[ ].Admin.MODE_POS_ABS := TRUE;`
- EnableCyclicScanning muss aktiviert werden

Mehr Informationen finden Sie in der Dokumentation "XLC/MLC xxVRS Generic Application Template" in Kapitel "ML_TechInterface.library".

xx - Version

Programm:

```
fbMB_RegisterControllerSideType01(
    Enable:= bEnableRegi,
    Pause:= bPauseRegi,
    Preset:= bPresetRegi,
    Step:= uiDetectedPositionDifference,
    PresetValue:= bPresetValueRegi,
    SetpointLock:= bSetpointLock,
    ControlPolarity:= bControlPolarity,
    ControlledValueIDN:= diControlledValueIDN,
```



```
PControl:= rPControl,
IControl:= rIControl,
TControl:= rTControl,
HighLimit:= rHighLimit,
LowLimit:= rLowLimit,
RecordedPositionWidth:= rPositionDifference,
ResponseCurve:= stResponseCurve,
FormatLength:= rForamtLength,
TriangleTangent:= rTriangleTangent,
SetpointWidth:= rSetpointWidth,
ControlledAxis:= Drive5,
InOperation=> bInOperationRegi,
Error=> bErrorRegi,
ErrorID=> stErrorIDRegi,
ErrorIdent=> stErrorIdentRegi,
InPause=> bInPauseRegi,
PresetAck=> bPresetAckRegi,
ActStep=> uiActStepRegi,
SetpointLockAck=> bSetpointLockAck,
HighLimitActive=> bHighLimitActive,
LowLimitActive=> bLowLimitActive,
ControlValue=> rControlValue,
CurrentValue=> rCurrentValue,
Deviation=> rDeviation,
CommandPosition=>CommandPosition );

fbMB_AxisInitType01(
  Execute:= bExecuteAxisInit,
  ReadModuloValue:= FALSE,
  EnableCyclicScanning:= TRUE,
  AdminCtrl:= arAxisCtrl[Drive5.AxisNo].Admin,
  PosModeCtrl:= arAxisCtrl[Drive5.AxisNo].PosMode,
  VelModeCtrl:= arAxisCtrl[Drive5.AxisNo].VelMode,
  SyncModeCtrl:= arAxisCtrl[Drive5.AxisNo].SyncMode,
  AxisDataPtrStatus:= arAxisStatus[Drive5.AxisNo].PtrAxisData,
  AdminStatus:= arAxisStatus[Drive5.AxisNo].Admin,
  DiagStatus:= arAxisStatus[Drive5.AxisNo].Diag,
  Axis:= Drive5,
  Done=> bDoneInit,
  Active=> bActiveInit,
  Error=> bErrorInit,
```

```
ErrorID=> stErrorIDInit,  
ErrorIdent=> stErrorIdent);  
  
(* Lagebefehl zum Antrieb der Seitenregisterregelung *)  
arAxisCtrl[Drive5.AxisNo].PosMode.Position := CommandPosition;  
fbMB_AxisInterfaceType01(  
    AdminCtrl:= arAxisCtrl[Drive5.AxisNo].Admin,  
    PosModeCtrl:= arAxisCtrl[Drive5.AxisNo].PosMode,  
    VelModeCtrl:= arAxisCtrl[Drive5.AxisNo].VelMode,  
    SyncModeCtrl:= arAxisCtrl[Drive5.AxisNo].SyncMode,  
    AdminStatus:= arAxisStatus[Drive5.AxisNo].Admin,  
    DiagStatus:= arAxisStatus[Drive5.AxisNo].Diag,  
    PosStatus:= arAxisStatus[Drive5.AxisNo].PosMode,  
    VelStatus:= arAxisStatus[Drive5.AxisNo].VelMode,  
    SyncModeStatus:= arAxisStatus[Drive5.AxisNo].SyncMode );
```

6 ML_TechCrank.library

6.1 Einführung und Übersicht

Allgemeines Technologie-Funktionsbausteine (Tech-FB) erweitern die Grundfunktionalität des Zielsystems MLC und stellen applikationsspezifische Funktionalitäten zur Verfügung. Die hier beschriebene Bibliothek beinhaltet Funktionalitäten für Kurbel- und Kniehebelkinematiken.

Die hier beschriebenen Funktionsbausteine werden über die Bibliothek "ML_TechCrank" für das Zielsystem MLC zur Verfügung gestellt.

Diese Dokumentation beschreibt die Funktionalität, sowie die Ein- und Ausgänge der Technologie-Funktionsbausteine.



Die Bibliothek benötigt das TEC (Technologie) Funktionspaket der Steuerung.

Voraussetzungen Technologie-Funktionsbausteine der Bibliothek "ML_TechCrank" benötigen die Firmware-Unterstützung des Zielsystems MLC. Spezifische Voraussetzungen der Technologie-Funktionsbausteine sind im Kapitel der entsprechenden Funktionsbausteine dokumentiert.

Übersicht über die Funktionsbausteine Die folgende Tabelle gibt einen Überblick über die verfügbaren Funktionsbausteine.

Anwendungsgebiete und Funktionsbausteine von ML_TechCrank

Anwendungsgebiet (Ordner in der Bibliothek)	
Funktionsbausteine	Beschreibung
Kniehebelkinematik (BellCrank/Basics)	
MB_BellCrankData, Seite 400	Berechnet die Y-Positionen und die Kurbelwinkel Phi an der unteren und oberen Polstelle
MB_YvirtToYmech, Seite 402	Wandelt eine virtuelle translatorische Position in die mechanische Position um
Kniehebelkinematik (BellCrank/CamCalculation)	
MB_CamTableBellCrank, Seite 404	Berechnet eine Kurvenscheibe zur Transformation für einen Kniehebelantrieb
MB_CamTableBellCrankSuperimposed, Seite 408	Überlagert eine Anwenderkurvenscheibe mit der Transformationskurvenscheibe zu einer überlagerten Kurvenscheibe
Kniehebelkinematik (BellCrank/Trafos)	
MB_PhiToYvirt, Seite 413	Wandelt den Kurbelwinkel Phi in die mechanische und virtuelle translatorische Position um
MB_YvirtToPhi, Seite 415	Wandelt die virtuelle translatorische Position in den Kurbelwinkel Phi um
MB_YvirtToPhiPoly5, Seite 417	Wandelt die virtuelle translatorische Position in den Kurbelwinkel Phi mittels eines Ersatzpolynoms 5. Ordnung um
MB_YVirtToWorkRange, Seite 419	Gibt den Arbeitsbereich der Kurbel 1 zurück
Kurbelkinematik (Crank/Crank_Cam)	
MB_CamTableCrank, Seite 425	Berechnet eine Kurvenscheibe zur Transformation einer translatorischen virtuellen Leitachslageposition in rotatorische Werte (Kurbelwinkel)

MB_CamTableCrankSuperimposed, Seite 430	Überlagert eine Anwenderkurvenscheibe mit der Transformationskurvenscheibe zu einer überlagerten Kurvenscheibe
Kurbelkinematik (Crank/Trafos)	
MB_MasterToPhi, Seite 436	Gibt Kurbelwinkel aus, die anhand der Leitachposition und der überlagerten Kurvenscheibe berechnet worden sind
MB_PhiToXvirt, Seite 435	Wandelt Kurbelwinkel (Phi) in virtuelle translatorische Positionswerte (Xvirt) zur Verwendung durch das SPS-Programm um
MB_XvirtToXmech, Seite 437	Wandelt virtuelle translatorische Positionswerte (Xvirt) in mechanische translatorische Positionswerte (Xmech) um

Tab. 6-1: ML_TechCrank-Bibliothek

6.2 Kniehebelkinematik (Bell Crank Kinematik)

6.2.1 Einführung und Übersicht

Kniehebelkinematiken werden zum Beispiel zum Antrieb von Formwerkzeugen bei Thermoformmaschinen oder Pressen eingesetzt. Die nachfolgende Abbildung zeigt den mechanischen Aufbau der im folgenden verwendeten Kniehebelkinematik (Bell Crank Kinematic).

Bei dieser Kinematik bewirkt die Drehung der Kurbel 1 eine translatorische Bewegung in y-Richtung. Die Kurbel wird durch einen Servomotor angetrieben. Das Messsystem ist am Servomotor angebracht und arbeitet damit in rotatorischen Einheiten.

Bei der gezeigten Kniehebelkinematik werden Soll- und Istwerte (wie z.B. Position, Geschwindigkeit...) in translatorischen Einheiten vorgegeben bzw. angezeigt, obwohl das Messsystem rotatorisch arbeitet. Die im folgenden beschriebenen Tech-FBs führen entsprechende Transformationen durch.

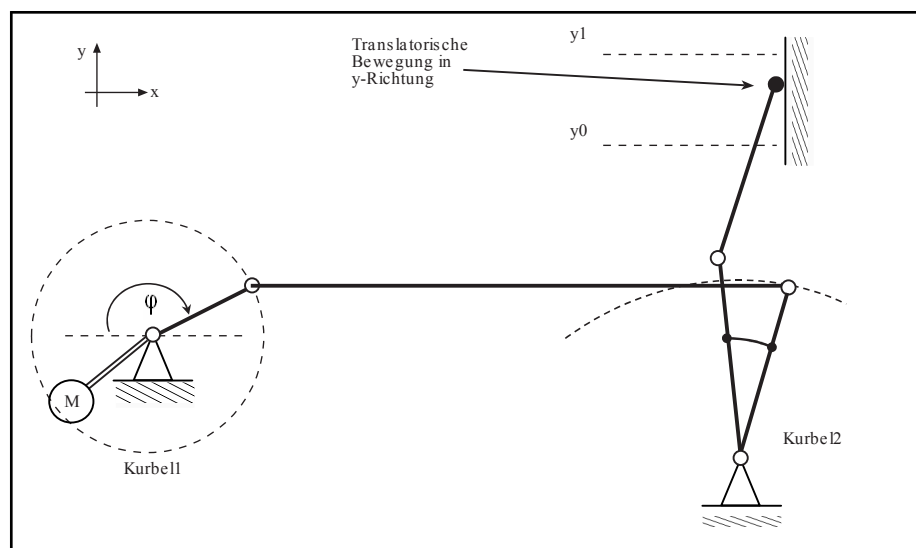


Abb. 6-1: Kniehebelkinematik (Bell Crank Kinematik)

6.2.2 Allgemeine Definitionen

Grundlegenden Variablen

Die nachfolgende Abbildung zeigt den Zusammenhang zwischen Kurbel- und Translatorischen Koordinaten, sowie die Bedeutung der einzelnen Mechanikparameter.

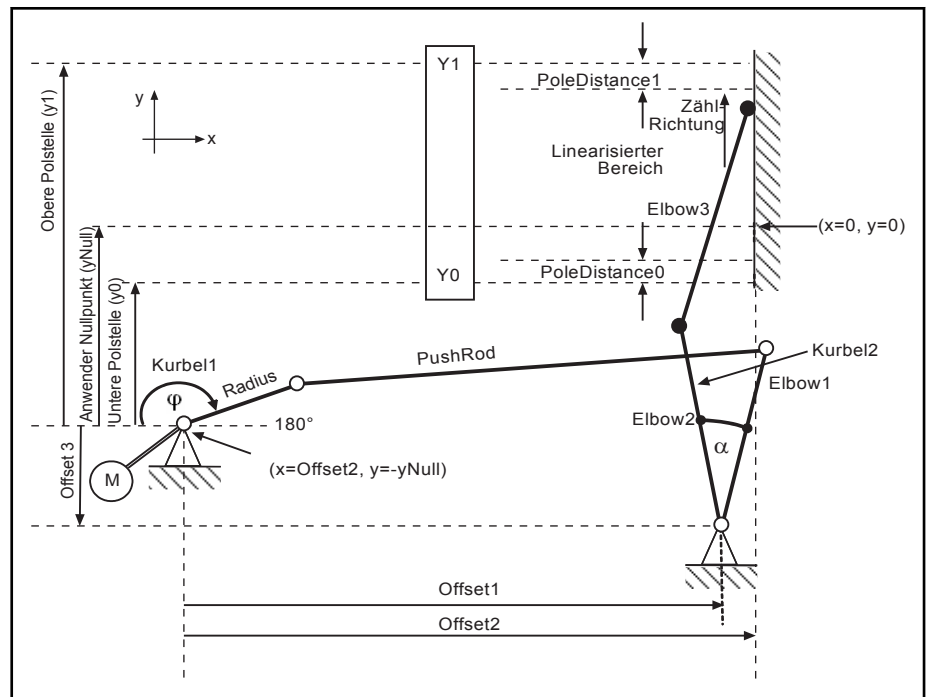


Abb. 6-2: Zusammenhang von Kurbel- und translatorischen Koordinaten an der Kniehebelkinematik

Zählrichtung

Allgemein gelten die folgenden Zählrichtungen für die in gezeigte Anordnung (Bild oben):

- Zählrichtung des Kurbel-Winkels:
Im Uhrzeigersinn, links beginnend mit 0° und endend mit 360°
- Zählweise der translatorischen y-Position:
Nach oben grösser werdende Zahlenwerte

Mechanische (Y_{mech}) und virtuelle (Y_{virt}) translatorische Position

Eine mechanische translatorische Position (Y_{mech}) kann typischerweise über zwei verschiedene Winkelstellungen der Kurbel 1 erreicht werden. Damit ist die mechanische translatorische Position (Y_{mech}) nicht eindeutig einem Kurbelwinkel zuzuordnen. Um diese Mehrdeutigkeit zu lösen, wird die virtuelle translatorische Position (Y_{virt}) eingeführt, welche auch bei Richtungsumkehr der translatorischen Achse weiter in positive Richtung zählt.

Der Anwender kann über den Nullpunkt (Y_{null}) festlegen, wo auf dem Verfahrbereich der mechanische Nullpunkt und damit auch der Moduloüberlauf liegt. Der Modulowert von Y_{virt} muss hierbei dem doppelten Verfahrbereich der translatorischen Achse entsprechen. Der Verfahrbereich (S_v) sei hierbei als Strecke zwischen der oberen (Y_1) und unteren (Y_0) Polstelle definiert.

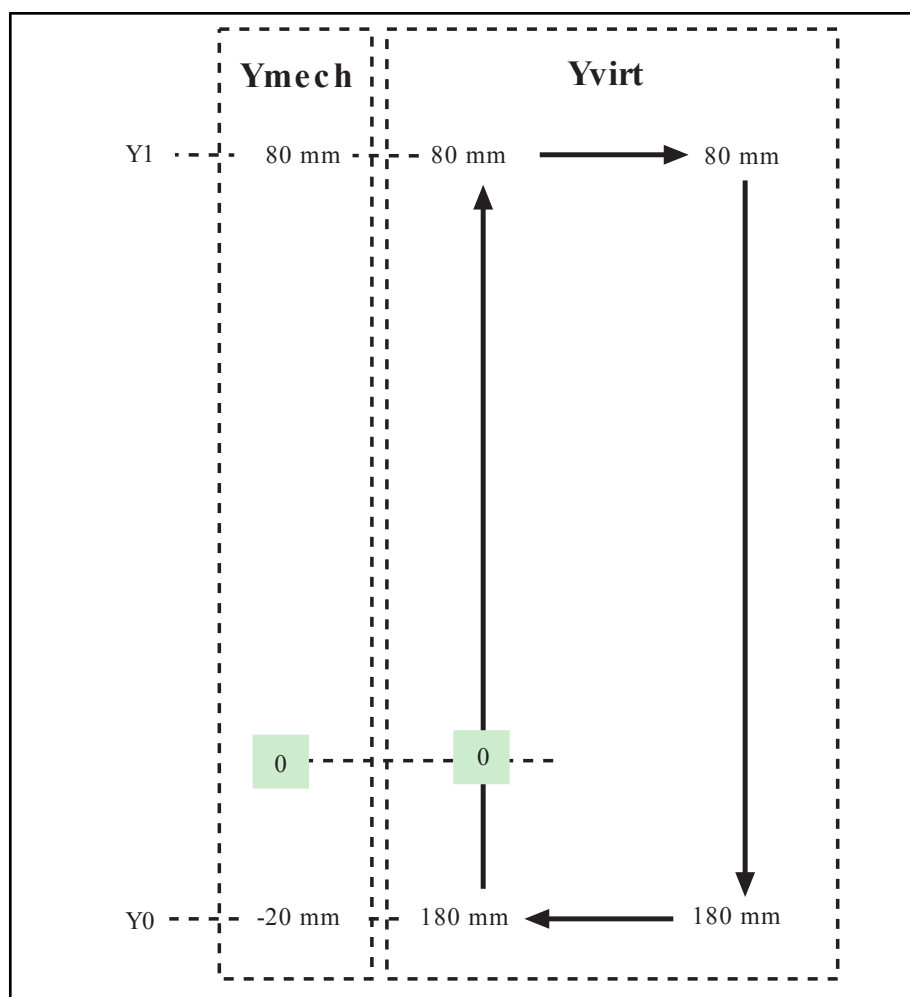


Abb. 6-3: Zusammenhang zwischen Y_{virt} und Y_{mech}

Grundprinzip der Kurvenscheiben- transformationen

Die von den Kniehebelkinematik FB berechneten Kurvenscheiben transformieren eine Leitachsposition in einen Kurbelwinkel des Kniehebelantriebes. Grundsätzlich werden dabei die folgenden zwei Anwendungsfälle unterschieden:

- Im ersten Fall wird die reale Kurbelachse über eine Transformationskurvenscheibe an eine translatorisch gewichtete virtuelle Achse gekoppelt. Die virtuelle Achse kann damit in anwenderfreundlichen translatorischen Einheiten (z.B. mm, mm/s...) in allen verfügbaren Betriebsarten betrieben werden und die Kurbel folgt gemäß der Transformationskurvenscheibe. Die notwendige Transformationskurvenscheibe wird hierbei durch den FB "MB_CamTableBellCrank" berechnet. Beim Zielsystem MLD unterstützt die virtuelle Achse keine Synchronbetriebsarten, wodurch dieser Anwendungsfall dort nur eingeschränkt sinnvoll ist.

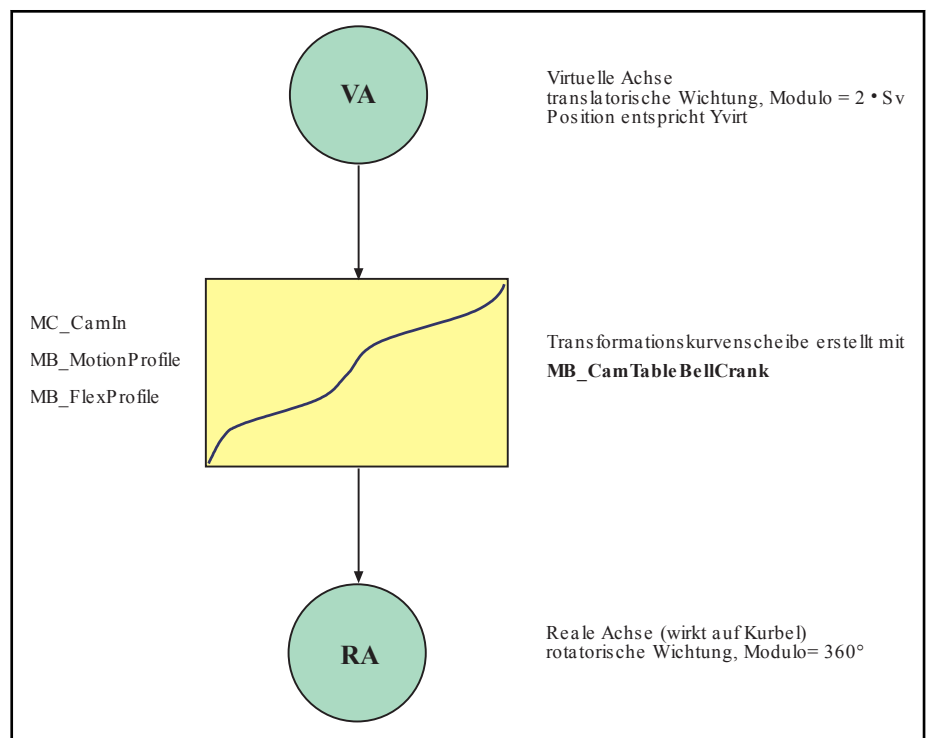


Abb. 6-4: Funktionsweise der Transformationskurvenscheibe mit einer translatorischen virtuellen Achse

- Im zweiten Fall wird die reale Kurbelachse über eine überlagerte Anwenderkurvenscheibe an eine beliebige Leitachse mit frei wählbarer Modulowichtung gekoppelt.

Die Anwenderkurvenscheibe beschreibt hierbei die Bewegung der translatorischen Achse im Bezug zur Leitachse ohne das nichtlineare Übersetzungsverhalten der Kinematik zu berücksichtigen. Der FB "MB_CamTableBellCrankSuperimposed" überlagert dafür die Anwenderkurvenscheibe mit den Transformationsgleichungen der Kinematik und berechnet damit die überlagerte Kurvenscheibe. Damit kann der Kniehebelantrieb mit einer Anwenderkurvenscheibe an eine beliebige Leitachse mit frei wählbarem Modulowert gekoppelt werden. Dieser Anwendungsfall ist speziell für das Zielsystem MLD sinnvoll, da der Kniehebelantrieb damit an eine Leitachse gekoppelt werden kann. Beim Zielsystem MLC hat dieser Anwendungsfall geringere Bedeutung, es kann jedoch damit eine zusätzliche virtuelle Achse eingespart werden.

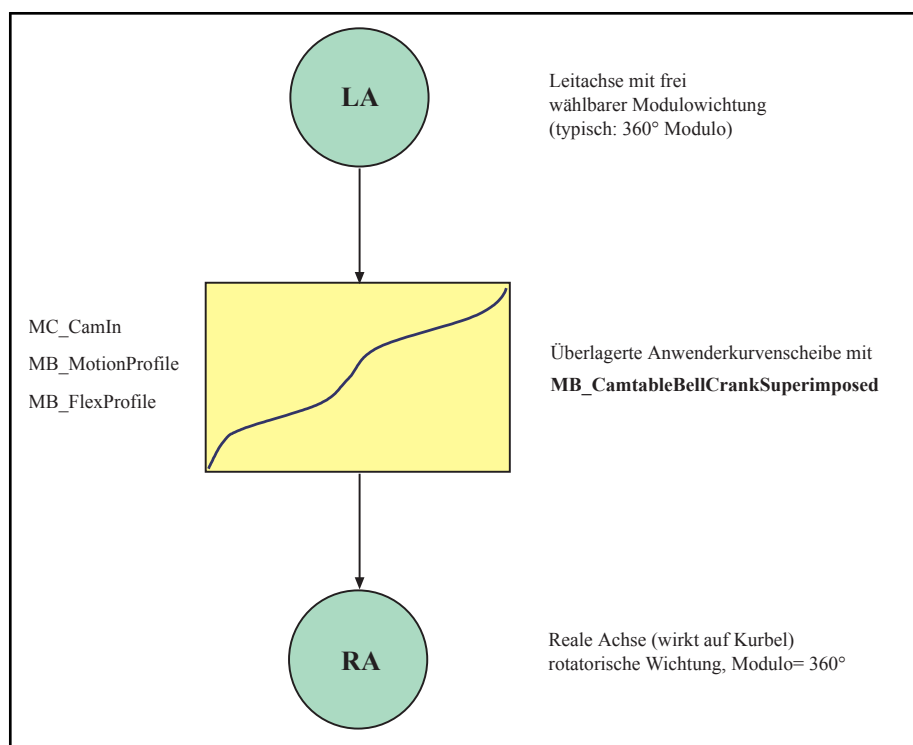


Abb. 6-5: Funktionsweise der überlagerten Kurvenscheibe an einer beliebigen Leitachse

Vereinfachungen und Randbedingungen

Die folgenden Vereinfachungen werden festgelegt und müssen erfüllt sein. Die mechanischen Kinematikdaten (wie z.B. Radius, Pushrod...) sind entsprechend zu wählen.

- An der oberen Polstelle (Y1) erreichen beide Kurbeln gleichzeitig ihre Totpunkte. An der oberen Polstelle bildet damit Radius und Pushrod, sowie Elbow2 und Elbow3 eine Gerade
- Die zweite Kurbel bewegt sich nur im Arbeitsbereich 1. Im Arbeitsbereich 1 bewegt sich die mechanische Position der y-Achse in positive Richtung wenn sich die zweite Kurbel im Uhrzeigersinn bewegt. Die erste Kurbel darf sich hingegen in beiden Arbeitsbereichen bewegen und auch endlos in eine Richtung drehen

6.2.3 MB_BellCrankData

Funktionsbeschreibung

Der Funktionsbaustein MB_BellCrankData berechnet die y-Positionen (Y0, Y1) und die Kurbelwinkel Phi (Phi0, Phi1) an der unteren und oberen Polstelle, sowie den maximalen Verfahrbereich (Sv) der y-Achse.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx08 und MLD mit MPx17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-2: Referenztable MB_BellCrankData

Schnittstellenbeschreibung

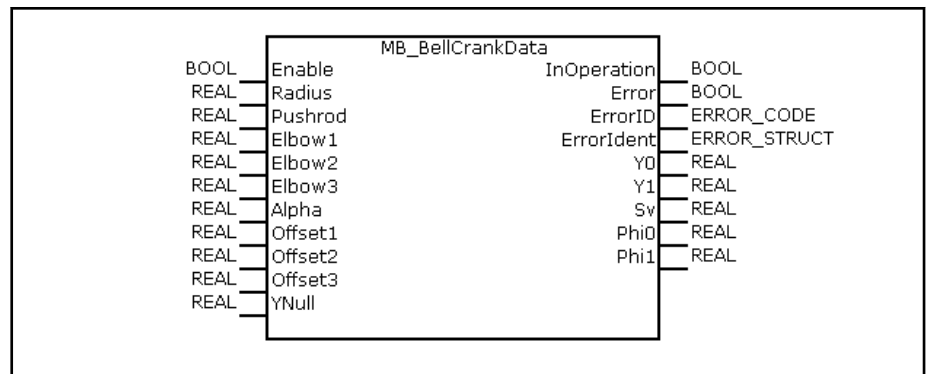


Abb. 6-6: Funktionsbaustein MB_BellCrankData

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Berechnung der Daten in jedem Zyklus solange Enable=TRUE
	Radius	REAL	Länge der Kurbel1 in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
VAR_OUTPUT	InOperation	BOOL	Fehlerfreie Bearbeitung, Ausgangsdaten sind gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 401
	Y0	REAL	Mechanische y-Position der unteren Polstelle [mm]
	Y1	REAL	Mechanische y-Position der oberen Polstelle [mm]
	Sv	REAL	Maximaler Verfahrbereich der y-Achse = ABS(Y1-Y0) [mm]
	Phi0	REAL	Winkel Phi der Kurbel 1 an der unteren Polstelle [°]
	Phi1	REAL	Winkel Phi der Kurbel 1 an der oberen Polstelle [°]

Tab. 6-3: Schnittstellenvariablen Funktionsbaustein MB_BellCrankData

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle „F_RELATED_TABLE“, 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Tab. 6-4: Fehlercodes Funktionsbaustein MB_BellCrankData

6.2.4 MB_YvirtToYmech

Funktionsbeschreibung

Der Funktionsbaustein MB_YvirtToYmech wandelt die virtuelle translatorische Position (Yvirt) in die mechanische translatorische Position (Ymech) um. Ymech kann z.B. für Anzeigezwecke genutzt werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-5: Referenztable MB_YvirtToYmech

Schnittstellenbeschreibung

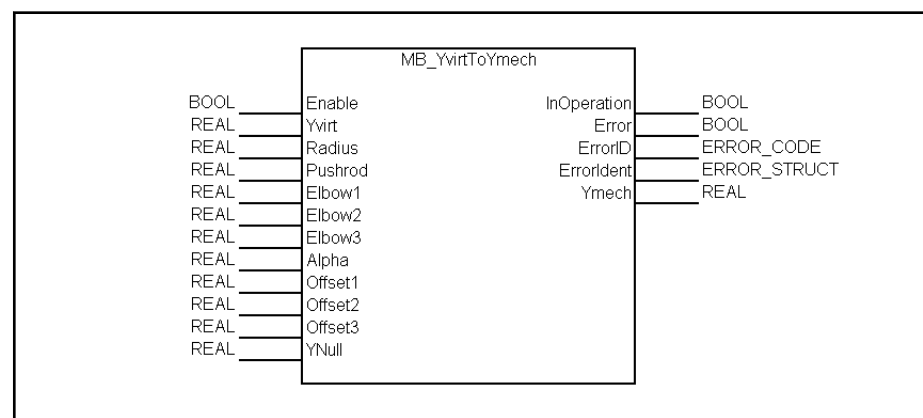


Abb. 6-7: Funktionsbaustein MB_YvirtToYmech

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	FB führt Berechnung aus, solange Enable=TRUE ist
	Yvirt	REAL	Virtuelle Translatorische Position in [mm]
	Radius	REAL	Länge der Kurbel in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
VAR_OUTPUT	InOperation	BOOL	Ausgangswert (Ymech) ist gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 403
	Ymech	REAL	Mechanische translatorische Position (Ymech) [mm]

Tab. 6-6: Schnittstellenvariablen Funktionsbaustein MB_YvirtToYmech

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Tab. 6-7: Fehlercodes Funktionsbaustein MB_YvirtToYmech

6.2.5 MB_CamTableBellCrank

Funktionsbeschreibung

Der Funktionsbaustein MB_CamTableBellCrank berechnet die notwendige Transformationskurvenscheibe (mit 1024 Stützstellen) für einen Kniehebelantrieb. Die Transformationskurvenscheibe dient zur Umrechnung von translatorischen Einheiten in den Kurbelwinkel Phi. Damit kann der rotatorische Kurbelantrieb über die Transformationskurvenscheibe an eine translatorisch gewichtete virtuelle Achse gekoppelt werden.

Die virtuelle Achse kann dann in translatorischen Einheiten bewegt werden, und der Kurbelantrieb folgt entsprechend der Transformationskurvenscheibe.

Um auch die Polstellen zu durchfahren, kann der Abstand zu den Polstellen (PoleDistance0 und PoleDistance1) definiert werden. Innerhalb der PoleDistance wird eine Ausgleichsfunktion (Ersatzpolynom 5ter Ordnung) verwendet, welche den Kniehebelantrieb approximiert und die resultierende Antriebsdynamik begrenzt. Bei dem Übergang in und aus dem linearisierten Bereich sind Position, Geschwindigkeit und Beschleunigung stetig.

Der FB stellt die berechnete Kurvenscheibe über den Ein/Ausgang "CamTable" dem SPS Programm zur Verfügung.



Betrifft Anwendung der Punktetabelle mit dem **FlexProfile**:

Je nach FB-Eingangsdaten ist der Startwert der berechneten Punktetabelle $\neq 0\%$.

Bei Aktivierung des FlexProfiles wird jedoch der Startwert = 0 wirksam.

Aktivieren Sie daher das FlexProfile mit dem Eingang:

SlaveOffset = (360°/100%)* erster Punkt der Punktetabelle [%]

Damit wird der korrekte Startwert und damit Kurbelwinkel aktiv.

Die Tabellenwerte liegen im Format "ARRAY OF DINT" vor und sind für 6 Nachkommastellen berechnet. Bei der o.g. Berechnung muss daher noch durch 1.000.000 dividiert werden.

Codebeispiel:

```
fbFlexProfile.SlaveOffset:=
    3.6*DINT_TO_REAL(arCamTable[0])/1000000;
```



Die Steigung der berechneten Punktetabelle wird im Bereich der Polstellen umso größer, je kleiner die Eingänge PoleDistance0 und PoleDistance1 gewählt werden.

Im Bereich der Polstelle wird damit die Geschwindigkeit höher. Dadurch kann der Steuerungsfehler F5113002, Sollgeschwindigkeit zu hoch für Zykluszeit, resultieren.

Sie können den F5113002-Fehler wie folgt beheben

- Motion-Zykluszeit verringern (wenn möglich)
- Leitachsgeschwindigkeit reduzieren (wenn möglich)
- PoleDistance0 und PoleDistance1 Eingang vergrößern

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx08 und MLD mit MPx17 ohne MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-8: Referenztable MB_CamTableBellCrank

Schnittstellenbeschreibung

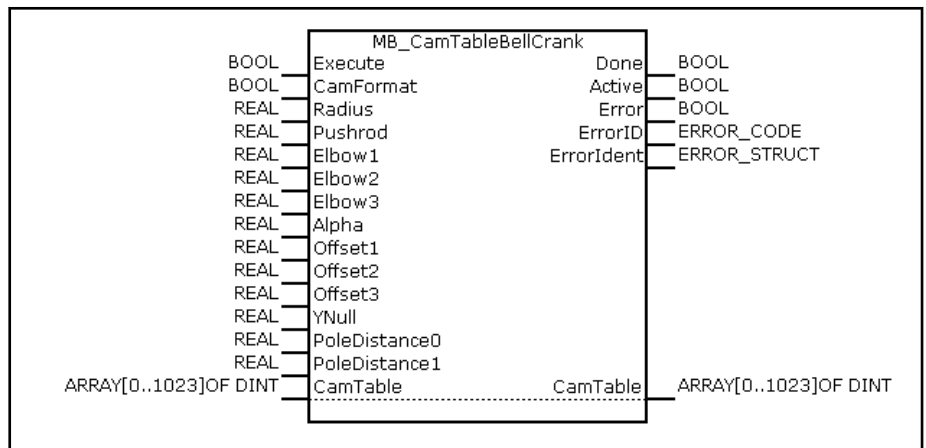


Abb. 6-8: Funktionsbaustein MB_CamTableBellCrank

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Positive Flanke startet die Berechnung der Transformationskurvenscheibe
	CamFormat	BOOL	TRUE: Neues Kurvenscheiben Format -> letzter Tabellenpunkt entspricht erstem Tabellenpunkt
	Radius	REAL	Länge der Kurbel1 in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]

E/A-Typ	Name	Datentyp	Beschreibung
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
	PoleDistance0	REAL	Entfernung von der unteren Polstelle Y0 (in [mm]), um mit begrenzter Antriebsdynamik die Polstelle zu durchfahren. Innerhalb dieses Bereiches wird ein Polynom 5ter Ordnung verwendet, welches die Kniehebelkinematik approximiert und die notwendige Antriebsdynamik begrenzt
	PoleDistance1	REAL	Entfernung von der oberen Polstelle Y1 (in [mm]), um mit begrenzter Antriebsdynamik die Polstelle zu durchfahren. Innerhalb dieses Bereiches wird ein Polynom 5ter Ordnung verwendet, welches die Kniehebelkinematik approximiert und die notwendige Antriebsdynamik begrenzt
VAR_OUTPUT	Done	BOOL	Berechnung abgeschlossen -> Kurvenscheibentabelle ist gültig
	Active	BOOL	FB ist in Bearbeitung
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_ CODE	Fehlerbeschreibung
	ErrorStruct	ERROR_ STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 407
VAR_IN_OUT	CamTable	ARRAY [0..1023] OF DINT	Array mit den Daten der berechneten Transformations-Kurvenscheiben

Tab. 6-9: Schnittstellenvariablen Funktionsbaustein MB_CamTableBellCrank

Notwendige Randbedingungen und Voraussetzungen

- An der oberen Polstelle (Y1) erreichen beide Kurbeln gleichzeitig ihre Totpunkte. An der oberen Polstelle bildet damit Radius und Pushrod, sowie Elbow2 und Elbow3 eine Gerade
- Die zweite Kurbel bewegt sich nur im Arbeitsbereich 1. Im Arbeitsbereich 1 bewegt sich die mechanische Position der y-Achse in positive Richtung wenn sich die zweite Kurbel im Uhrzeigersinn bewegt. Die erste Kurbel darf sich hingegen in beiden Arbeitsbereichen bewegen und auch endlos in eine Richtung drehen
- Die für die Kurvenscheibe (als Master) verwendete virtuelle Achse muss translatorisch, modulo gewichtet sein.
Der Modulowert soll hierbei wie folgt eingestellt werden:
Modulowert = 2 * Verfahrbereich (Sv).
Der Verfahrbereich (Sv) sei hierbei als Strecke zwischen der oberen (Y1) und unteren (Y0) Polstelle definiert.

**Anwendungsbeispiel von
MB_CamTableBellCrank**

Der Verfahrbereich Sv kann über den FB [MB_BellCrankData](#), Seite 400, ermittelt werden

- Der Kurbelantrieb muss rotatorisch, modulo 360° gewichtet sein

Mit der folgenden Sequenz kann die translatorische Achse im Einzelachsbetrieb verfahren werden:

- Berechnung der Transformationskurvenscheibe durch den Baustein MB_CamTableBellCrank
- Berechnete Transformationskurvenscheibe auf den Kurbelantrieb laden (z.B. durch MB_WriteListParameter, falls diese nicht bereits geladen ist)
- Zuschalten der Leistung (MC_Power)
- Herstellen des absoluten Maßbezuges, falls kein Absolutwertgeber verwendet wird. Hierzu wird das Antriebskommando "Antriebsgeführtes Referenzieren" C0600 verwendet
- Virtuelle Achse auf die aus dem Kurbelwinkel resultierende translatorische Position positionieren. Die resultierende translatorische Position "Yvirt" kann aus dem Kurbelwinkel (Phi) mit dem FB [MB_PhiToYvirt](#), Seite 413, ermittelt werden. Die virtuelle Achse wird dann auf die ermittelte Position positioniert
- Zuschalten der Kurvenscheibenbetriebsart am Kurbelantrieb (z.B. über MC_CamIn, mit CamShaftDistance=360,1:1-Getriebe und Auswahl der zuvor berechneten und geladenen Transformationskurvenscheibe). Der Kurbelantrieb wird keine Aufsynchronisier-Bewegung ausführen, da die virtuelle Achse bereits an der "Synchronposition" steht
- Die virtuelle Achse kann jetzt über die Bausteine MC_MoveAbsolute und MC_MoveVelocity oder AxisInterface im Einzelachsbetrieb verfahren werden (die Kurbelachse folgt dann der Transformationskurvenscheibe). Die virtuelle Achse kann auch in einer Synchronbetriebsart betrieben werden



Der Kurbelantrieb führt eine dynamische Aufsynchronisierbewegung bei Zuschalten der Kurvenscheibenbetriebsart aus, wenn die virtuelle Achse vorher nicht auf die Kurbelposition positioniert wurde.

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0006	PoleDistance0 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0007	PoleDistance1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0008	PoleDistance0+PoleDistance1 > Verfahrbereich
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	$Y_{null} > ((Y_0 + Y_{null}) + 2 \cdot \text{Verfahrbereich})$
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Tab. 6-10: Fehlercodes Funktionsbaustein MB_CamTableBellCrank

6.2.6 MB_CamTableBellCrankSuperimposed

Funktionsbeschreibung

Dieser Baustein überlagert die vorgegebene Anwenderkurvenscheibe (CamInput) mit der Transformationskurvenscheibe und liefert das Resultat über den Ein/Ausgang "CamOutput". Die Transformationskurvenscheibe wird innerhalb des FB nach den gleichen Gesetzmässigkeiten, wie im FB [MB_CamTableBellCrank](#), Seite 404, berechnet.

Das Prinzip der Überlagerung ist in [Prinzip der Kurvenscheiben Überlagerung](#), Seite 409, dargestellt. Die Anwenderkurvenscheibe muss 1024 Stützpunkte enthalten und definiert hierbei die Bewegung der translatorischen Achse im Bezug zur Leitachse (ohne Berücksichtigung der Kniehebelkinematik).

Der Tabellenwert 100% entspricht dabei der Bewegung

$2 \cdot \text{Verfahrbereich} = \text{Modulwert der virtuellen translatorischen Achse.}$

Die berechnete überlagerte Kurvenscheibe muss zur Nutzung von dem SPS Programm auf den Antrieb geladen werden, und über die Kurvenscheiben Betriebsart genutzt werden.



Je nach vorgegebenem Endpunkt der Anwenderkurvenscheibe werden folgende Fälle unterschieden:

- Ist der Endpunkt der Anwenderkurvenscheibe nahe 100%, so führt die Kurbel 1 keine Richtungsumkehr aus (Kurbel 1 "dreht durch") -> Energieoptimales Verfahren, da natürliche Bewegung der Kurbel genutzt wird
- Ist der Endpunkt der Anwenderkurvenscheibe 0%, entsteht eine Vor-Rück-Bewegung der translatorischen Achse mit Richtungsumkehr der Kurbel 1 (wie in [Prinzip der Kurvenscheiben Überlagerung](#), Seite 409, dargestellt)

Weiterführender Hinweis

Ab MLC03VRS unterstützt die verwendete virtuelle Achse Synchron-Betriebsarten. Dadurch ist das Überlagern der Anwenderkurvenscheibe mit der Transformationskurvenscheibe typischerweise für MLC03 Anwendungen nur dann notwendig, wenn eine zusätzliche virtuelle Achse eingespart werden soll.

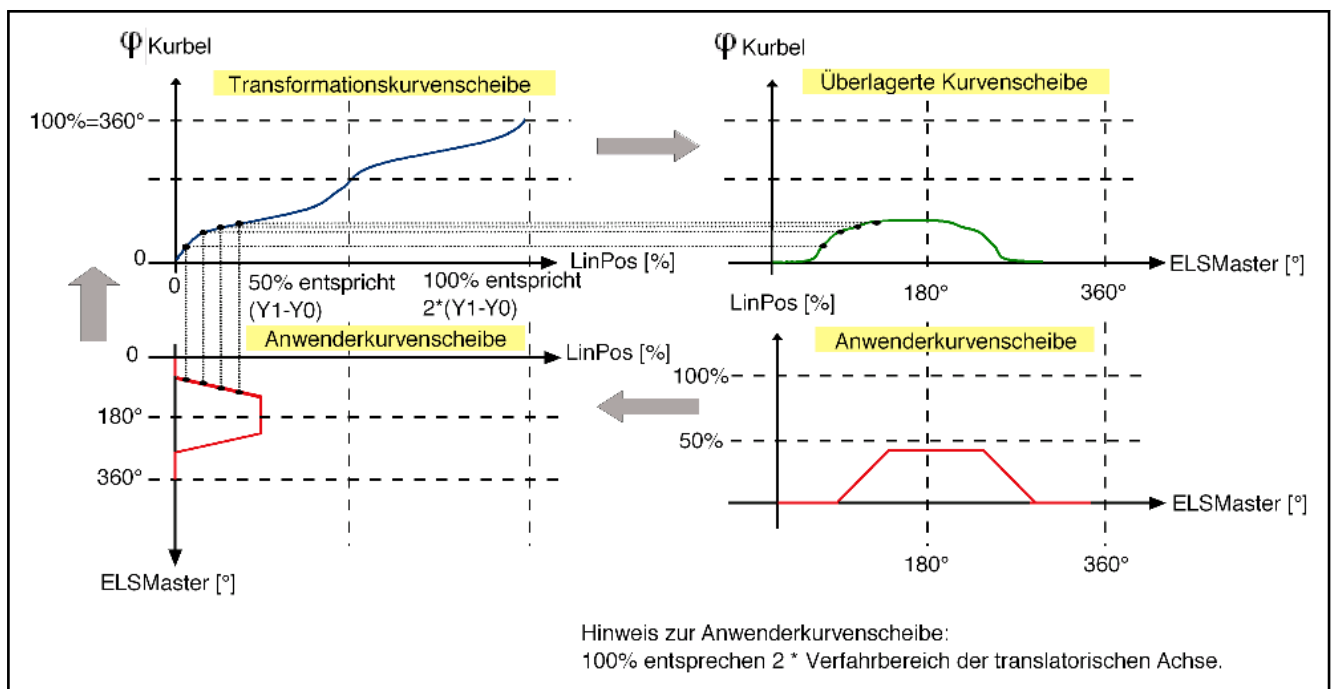


Abb. 6-9: Prinzip der Kurvenscheiben Überlagerung



Betrifft Anwendung der Punktetabelle mit dem **FlexProfile**:

Je nach FB-Eingangsdaten ist der Startwert der berechneten Punktetabelle $\neq 0\%$.

Bei Aktivierung des FlexProfiles wird jedoch der Startwert = 0 wirksam.

Aktivieren Sie daher das FlexProfile mit dem Eingang:

SlaveOffset = (360°/100%) * erster Punkt der Punktetabelle [%]

Damit wird der korrekte Startwert und damit Kurbelwinkel aktiv.

Die Tabellenwerte liegen im Format "ARRAY OF DINT" vor und sind für 6 Nachkommastellen berechnet. Bei der o.g. Berechnung muss daher noch durch 1.000.000 dividiert werden.

Codebeispiel:

```
fbFlexProfile.SlaveOffset:=
    3.6*DINT_TO_REAL(arCamTable[0])/1000000;
```



Die Steigung der berechneten Punktetabelle wird im Bereich der Polstellen umso größer, je kleiner die Eingänge PoleDistance0 und PoleDistance1 gewählt werden.

Im Bereich der Polstelle wird damit die Geschwindigkeit höher. Dadurch kann der Steuerungsfehler F5113002, Sollgeschwindigkeit zu hoch für Zykluszeit, resultieren.

Sie können den F5113002-Fehler wie folgt beheben

- Motion-Zykluszeit verringern (wenn möglich)
- Leitachsgeschwindigkeit reduzieren (wenn möglich)
- PoleDistance0 und PoleDistance1 Eingang vergrößern

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx08 und MLD mit MPx17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-11: Referenztable MB_CamTableBellCrankSuperimposed

Schnittstellenbeschreibung

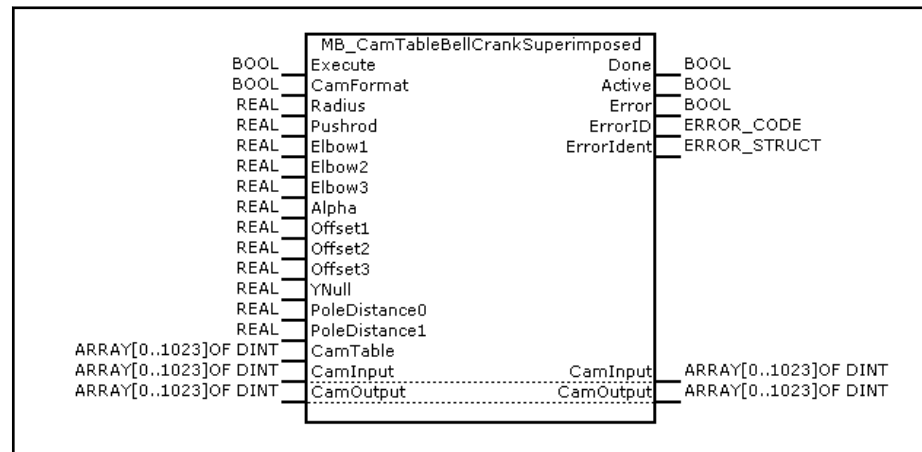


Abb. 6-10: Funktionsbaustein MB_CamTableBellCrankSuperimposed

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	Positive Flanke startet die Berechnung der überlagerten Kurvenscheibe
	CamFormat	BOOL	TRUE: Neues Kurvenscheiben Format -> letzter Tabellenpunkt entspricht erstem Tabellenpunkt
	Radius	REAL	Länge der Kurbel1 in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]

E/A-Typ	Name	Typ	Kommentar
	PoleDistance0	REAL	Entfernung von der unteren Polstelle Y0 (in [mm]), um mit begrenzter Antriebsdynamik die Polstelle zu durchfahren. Innerhalb dieses Bereiches wird ein Polynom 5ter Ordnung verwendet, welches die Kniehebelkinematik approximiert und die notwendige Antriebsdynamik begrenzt
	PoleDistance1	REAL	Entfernung von der oberen Polstelle Y1 (in [mm]), um mit begrenzter Antriebsdynamik die Polstelle zu durchfahren. Innerhalb dieses Bereiches wird ein Polynom 5ter Ordnung verwendet, welches die Kniehebelkinematik approximiert und die notwendige Antriebsdynamik begrenzt
	CamTable	ARRAY [0..1023] OF DINT	
VAR_OUTPUT	Done	BOOL	Berechnung abgeschlossen, Ausgangs-Kurvenscheibentabelle ist gültig
	Active	BOOL	FB ist in Bearbeitung
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_ CODE	Fehlerbeschreibung
	ErrorIdent	ERROR_ STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 413
VAR_IN_OUT	CamInput	ARRAY [0..1023] OF DINT	Array mit den Daten der Anwenderkurvenscheibe zur Vorgabe der translatorischen Bewegung
	CamOutput	ARRAY [0..1023] OF DINT	Array mit den Daten der berechneten überlagerten Kurvenscheibe zur Ansteuerung der Kurbelachse

Tab. 6-12: Schnittstellenvariablen Funktionsbaustein MB_CamTableBellCrank-Superimposed



zur Anwendung der überlagerten Kurvenscheibe

Soll die translatorische Achse auf die überlagerte Kurvenscheibe synchronisiert werden, ist wie folgt vorzugehen:

- Anwenderkurvenscheibe wird über den Ein/Ausgang "CamInput" dem FB zur Verfügung gestellt (z.B. durch Auslesen einer Antriebskurvenscheibe, oder Auslesen eines Files...)
- Die Berechnung der überlagerten Kurvenscheibe wird durch den Baustein MB_CamTableBellCrankSuperimposed ausgeführt
- Nach der Kurvenscheiben-Überlagerung lädt das Anwenderprogramm die überlagerte Kurvenscheibe "CamOutput" auf den Antrieb oder die Steuerung (MB_WriteListParameter)
- Zuschaltung der Leistung (MC_Power) am Kurbelantrieb
- Herstellen des absoluten Maßbezuges der Kurbelachse, falls kein Absolutwertgeber verwendet wird. Hierzu muss das Antriebskommando C0600 "Antriebsgeführtes Referenzieren" verwendet werden
- Die aktuelle Kurbelposition und der Lagesollwert aus der überlagerten Kurvenscheibe können auseinander liegen. Ein Umschalten der Kurvenscheibenbetriebsart (z.B. über MC_CamIn, mit CamShaftDistance= 360,1:1 Getriebe und Auswahl der überlagerten Kurvenscheibe) kann damit über die folgenden 2 Optionen durchgeführt werden:
 - Umschalten der Kurvenscheibenbetriebsart am Kurbelantrieb ohne vorherigen Positions-Abgleich -> Der Kurbelantrieb wird ein dynamisches Aufsynchronisieren ausführen
 - Kurbelachse wird vor Umschalten der Kurvenscheibenbetriebsart auf die "Synchronposition" positioniert. Hierzu kann über die Funktion "MB_MasterToPhi" der Kurbelwinkel aus der überlagerten Kurvenscheibe ermittelt werden. Vor Umschalten der Synchronbetriebsart kann dann der Kurbelantrieb auf diese Position positioniert werden. Im Anschluss kann dann die überlagerte Kurvenscheibe ohne Aufsynchronisierungsbewegung aktiviert werden



zur Wichtung der Achsen

- Die verwendete Leitachse muss Modulo gewichtet werden (translatorisch oder rotatorisch)
- Die Achse zum Antrieb der Kurbel muss rotatorisch, modulo 360° gewichtet sein

Notwendige Randbedingungen und Voraussetzungen

- An der oberen Polstelle (Y1) erreichen beide Kurbeln gleichzeitig ihre Totpunkte. An der oberen Polstelle bildet damit Radius und Pushrod, sowie Elbow2 und Elbow3 eine Gerade
- Die zweite Kurbel bewegt sich nur im Arbeitsbereich 1. Im Arbeitsbereich 1 bewegt sich die mechanische Position der y-Achse in positive Richtung wenn sich die zweite Kurbel im Uhrzeigersinn bewegt. Die erste Kurbel darf sich hingegen in beiden Arbeitsbereichen bewegen und auch endlos in eine Richtung drehen

- Die für die Kurvenscheibe (als Master) verwendete virtuelle Achse muss translatorisch, modulo gewichtet sein.
Der Modulwert soll hierbei wie folgt eingestellt werden:
 $\text{Modulwert} = 2 \cdot \text{Verfahrbereich (Sv)}$
Der Verfahrbereich (Sv) sei hierbei als Strecke zwischen der oberen (Y1) und unteren (Y0) Polstelle definiert.
Der Verfahrbereich Sv kann über den FB [MB_BellCrankData](#), Seite 400, ermittelt werden
- Der Kurbelantrieb muss rotatorisch, modulo 360° gewichtet sein

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius ≤ 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod ≤ 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0006	PoleDistance0 ≤ 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0007	PoleDistance1 ≤ 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0008	PoleDistance0+PoleDistance1 > Verfahrbereich
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 ≤ 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 ≤ 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 ≤ 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Tab. 6-13: Fehlercodes Funktionsbaustein MB_CamTableBellCrankSuperimposed

6.2.7 MB_PhiToYvirt

Funktionsbeschreibung

Der Funktionsbaustein MB_PhiToYvirt wandelt den Kurbelwinkel (Phi) in die mechanische - und virtuelle translatorische Position (Ymech, Yvirt) um.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-14: Referenztablelle MB_PhiToYvirt

Schnittstellenbeschreibung

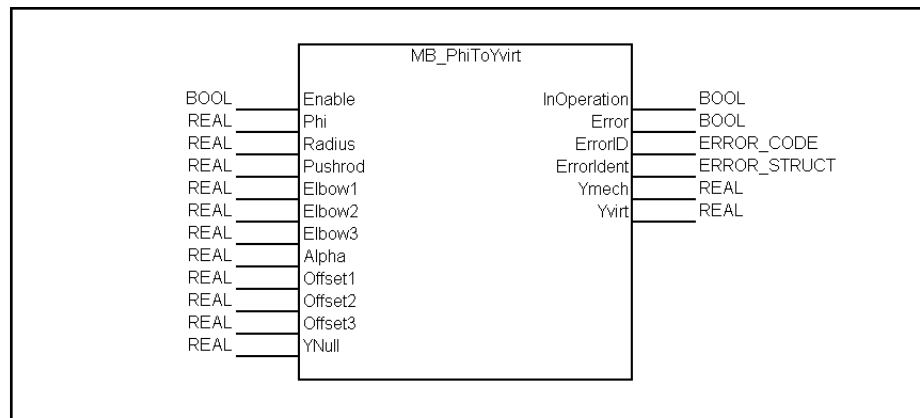


Abb. 6-11: Funktionsbaustein MB_PhiToYvirt

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	Berechnung von Yvirt in jedem Zyklus solange Enable=TRUE
	Phi	REAL	Kurbelposition in [°]
	Radius	REAL	Länge der Kurbel in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
VAR_OUTPUT	InOperation	BOOL	Berechnung ist fertig -> Yvirt ist gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 414
	Ymech	REAL	Mechanische translatorische Position [mm]
	Yvirt	REAL	Virtuelle translatorische Position [mm]

Tab. 6-15: Schnittstellenvariablen Funktionsbaustein MB_PhiToYvirt

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Tab. 6-16: Fehlercodes Funktionsbaustein MB_PhiToYvirt

6.2.8 MB_YvirtToPhi

Funktionsbeschreibung

Der Funktionsbaustein MB_YvirtToPhi wandelt die virtuelle translatorische Position (Yvirt) in den Kurbelwinkel (Phi) um.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx08 und MLD mit MPx17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-17: Referenztablelle MB_YvirtToPhi

Schnittstellenbeschreibung

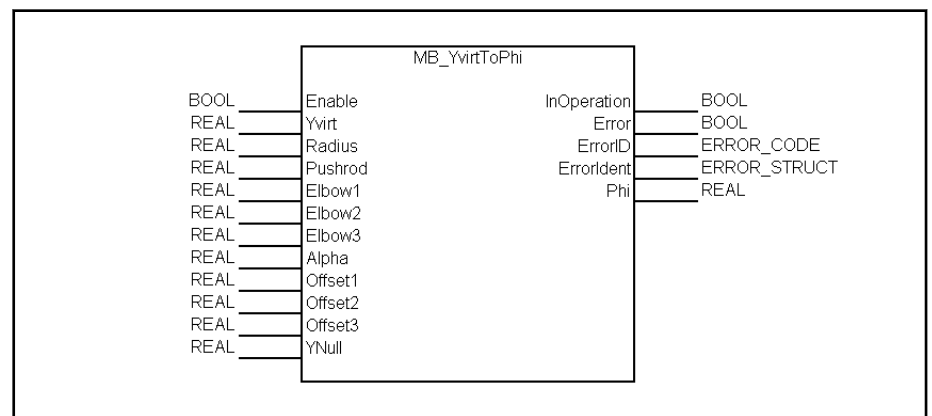


Abb. 6-12: Funktionsbaustein MB_YvirtToPhi

E/A-Typ	Name	Typ	Beschreibung
VAR_INPUT	Enable	BOOL	Berechnung von Phi in jedem Zyklus solange Enable=TRUE
	Yvirt	REAL	Virtuelle translatorische Position [mm]
	Radius	REAL	Länge der Kurbel in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
VAR_OUTPUT	InOperation	BOOL	Berechnung ist fertig -> Phi ist gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 416
	Phi	REAL	Kurbelwinkel in [°]

Tab. 6-18: Schnittstellenvariablen Funktionsbaustein MB_YvirtToPhi

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xyyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Tab. 6-19: Fehlercodes Funktionsbaustein MB_YvirtToPhi

6.2.9 MB_YvirtToPhiPoly5

Funktionsbeschreibung

Der Funktionsbaustein MB_YvirtToPhiPoly5 wandelt die virtuelle translatorische Position (Yvirt) in den Kurbelwinkel (Phi) um. Innerhalb der PoleDistance wird eine Ausgleichsfunktion (Ersatzpolynom 5ter Ordnung) verwendet, welche den Kniehebelantrieb approximiert und die resultierende Antriebsdynamik begrenzt. Bei dem Übergang in und aus dem linearisierten Bereich sind Position, Geschwindigkeit und Beschleunigung an der Kurbel 1 stetig.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx08 und MLD mit MPx17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-20: Referenztable MB_YvirtToPhiPoly5

Schnittstellenbeschreibung

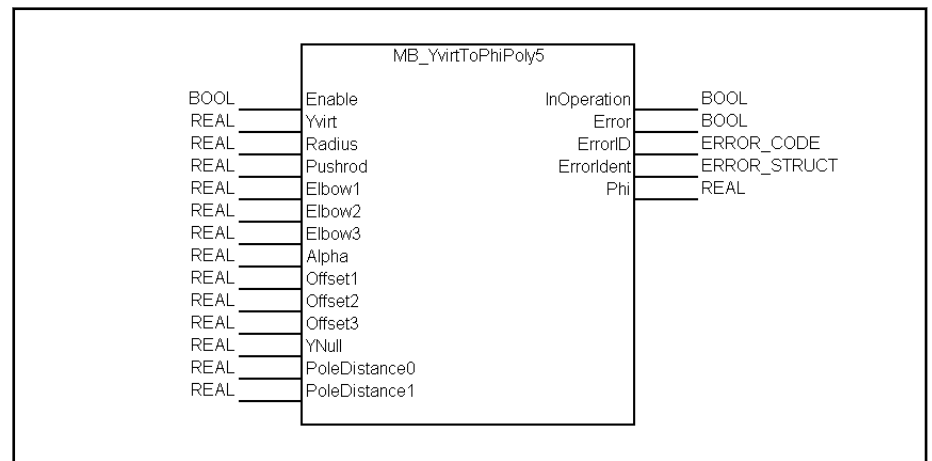


Abb. 6-13: Funktionsbaustein MB_YvirtToPhiPoly5

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Berechnung von Phi in jedem Zyklus solange Enable=TRUE
	Yvirt	REAL	Virtuelle translatorische Position [mm]
	Radius	REAL	Länge der Kurbel in [mm]
	Pushrod	REAL	Länge des Pleuel in [mm]
	Elbow1	REAL	Länge des Kniehebels 1 [mm]
	Elbow2	REAL	Länge des Kniehebels 2 [mm]
	Elbow3	REAL	Länge des Kniehebels 3 [mm]
	Alpha	REAL	Winkel zwischen Kniehebel 1 und 2 [°]
	Offset1	REAL	X-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	Offset2	REAL	X-Achsen Versatz zwischen Kurbeldrehpunkt und translatorischer Bewegung [mm]
	Offset3	REAL	Y-Achsen Versatz zwischen Kurbel- und Kniehebel-Drehpunkt [mm]
	YNull	REAL	Abstand Kurbeldrehpunkt zur Nullposition der translatorischen Bewegung [mm]
	PoleDistance0	REAL	Entfernung von der unteren Polstelle Y0 (in [mm]) um mit begrenzter Antriebsdynamik die Polstelle zu durchfahren. Innerhalb dieses Bereiches wird ein Polynom 5ter Ordnung verwendet, welches die Kniehebelkinematik approximiert und die notwendige Antriebsdynamik begrenzt
	PoleDistance1	REAL	Entfernung von der oberen Polstelle Y1 (in [mm]) um mit begrenzter Antriebsdynamik die Polstelle zu durchfahren. Innerhalb dieses Bereiches wird ein Polynom 5ter Ordnung verwendet, welches die Kniehebelkinematik approximiert und die notwendige Antriebsdynamik begrenzt
VAR_OUTPUT	InOperation	BOOL	Berechnung ist fertig -> Phi ist gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 418
	Phi	REAL	Kurbelwinkel in [°]

Tab. 6-21: Schnittstellenvariablen Funktionsbaustein MB_YvirtToPhiPoly5

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0006	PoleDistance0 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0007	PoleDistance1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0008	PoleDistance0+PoleDistance1 > Verfahrbereich
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0009	Elbow1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000A	Elbow2 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000B	Elbow3 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000C	Alpha > 360° OR Alpha < -360°
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000D	Ynull < (Y0+Ynull)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Ynull > ((Y0+Ynull)+2*Verfahrbereich)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Interner Mathematik-Fehler, siehe Additional 2: xx = 16#01: Error caused by MB_BellCrankData xx = 16#02: Error caused by MB_PhiToYvirt xx = 16#03: Error caused by MB_YvirtToPhi xx = 16#04: Error caused by MB_YvirtToPhiPoly5

Tab. 6-22: Fehlercodes Funktionsbaustein MB_YvirtToPhiPoly5

6.2.10 MB_YVirtToWorkRange

Funktionsbeschreibung

Die Funktion MB_YVirtToWorkRange liefert als Rückgabewert den Arbeitsbereich (1 oder 2) der Kurbel 1, welche sich aus der virtuellen translatorischen Position ergibt. Im Arbeitsbereich 1 bewegt sich die mechanische Position der y-Achse in positive Richtung wenn sich die Kurbel 1 im Uhrzeigersinn bewegt. Im Arbeitsbereich 2 bewegt sich die mechanische Position der y-Achse in negative Richtung wenn sich die Kurbel 1 im Uhrzeigersinn bewegt. Diese Funktion wird intern von anderen Funktionsbausteinen verwendet.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx08 und MLD mit MPx17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-23: Referenztable MB_YVirtToWorkRange

Schnittstellenbeschreibung

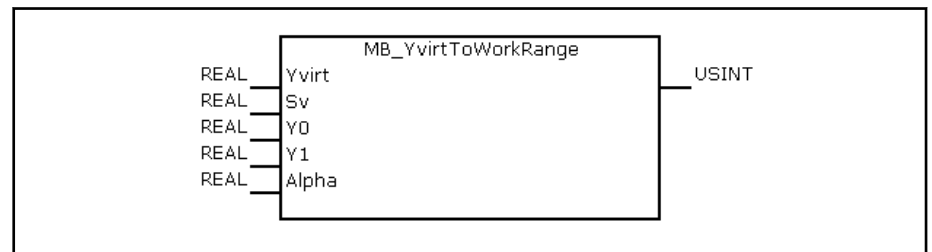


Abb. 6-14: Funktion MB_YVirtToWorkRange

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Yvirt	REAL	Virtuelle Translatorische Position [mm]
	Sv	REAL	Maximaler Verfahrbereich der y-Achse = $ABS(Y1-Y0)$ [mm]
	Y0	REAL	Mechanische y-Position der unteren Polstelle [mm]
	Y1	REAL	Mechanische y-Position der oberen Polstelle [mm]
	Alpha	REAL	Winkel zwischen dem ersten und dem zweiten Ellbogen [°]
Rückgabewert		USINT	Liefert den Arbeitsbereich (1 oder 2) der Kurbel 1

Tab. 6-24: Schnittstellenvariablen Funktion MB_YVirtToWorkRange

Fehlerbehandlung Die Funktion generiert keine Fehler.

6.3 Kurbelkinematik

6.3.1 Einführung und Übersicht

Die in diesem Abschnitt beschriebenen Funktionen und Funktionsbausteine dienen zur Konvertierung von translatorischer in rotatorische Bewegung (Kurbelwinkel) und umgekehrt und sind für Kurbelkinematik-Applikationen bestimmt.

Kurbelkinematiken werden häufig zum Antrieb von Quersiegelbacken in Schlauchbeutelmaschinen oder von Form- und Stanzwerkzeugen in Thermoformmaschinen eingesetzt. Der translatorische Schlitten in dieser Kinematik wird durch die Drehbewegung einer Kurbel bewegt. Die Achse des Servomotors, der die Kurbel antreibt, ist gegenüber der Achse des translatorischen Schlittens versetzt angeordnet.

Die unten dargestellte Kurbelkinematik gibt Soll- und Istwerte (Lage, Geschwindigkeit usw.) in translatorischen Einheiten aus, obwohl die Ausgabe des Messsystems in rotatorischen Einheiten erfolgt. Deshalb müssen die Soll- und Istwerte von translatorischen in rotatorische Einheiten umgewandelt werden und umgekehrt.

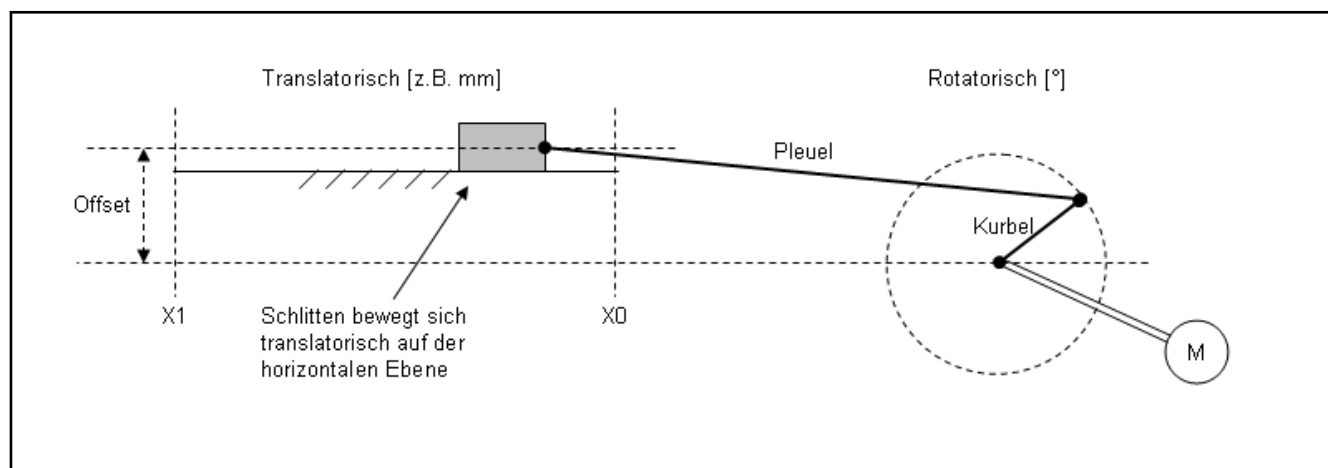


Abb. 6-15: Einzelachs-Kurbelkinematik mit versetzter Kurbelachse

Die folgenden Funktionen und Funktionsbausteine werden unterstützt:

Funktion	Beschreibung
MB_MasterToPhi, Seite 436	Gibt Kurbelwinkel aus, die anhand der Leitachsposition und der überlagerten Kurvenscheibe berechnet worden sind

Tab. 6-25: Kurbelkinematik-Funktion

Funktionsbaustein	Beschreibung
MB_CamTableCrank, Seite 425	Berechnet eine Kurvenscheibe zur Transformation einer translatorischen virtuellen Leitachsposition in rotatorische Werte (Kurbelwinkel)
MB_CamTableCrankSuperimposed, Seite 430	Überlagert eine Anwenderkurvenscheibe mit der Transformationskurvenscheibe zu einer überlagerten Kurvenscheibe
MB_PhiToXvirt, Seite 435	Wandelt Kurbelwinkel (Phi) in virtuelle translatorische Positionswerte (Xvirt) zur Verwendung durch das SPS-Programm um
MB_XvirtToXmech, Seite 437	Wandelt virtuelle translatorische Positionswerte (Xvirt) in mechanische translatorische Positionswerte (Xmech) um

Tab. 6-26: Kurbelkinematik-Funktionsbausteine

Typische Anwendungsbeispiele sind:**Betriebsart Einzelachse**

Der Begriff Einzelachsbetrieb bezieht sich auf den lage- oder geschwindigkeitsgeregelten Betrieb der translatorischen Achse. In der vorliegenden Applikation wird eine virtuelle Leitachse (mit translatorischer Wichtung) durch Lage- oder Geschwindigkeitsregelungs-Funktionsbausteine bewegt (z. B. MC_MoveAbsolute, MC_MoveRelative oder MC_MoveVelocity).

Die virtuelle Leitachsposition wird durch die Transformationskurvenscheibe von translatorischen Positionswerten in Kurbelwinkelwerte umgerechnet. Die Transformationskurvenscheibe wird durch den Funktionsbaustein [MB_CamTableCrank, Seite 425](#), berechnet und mit dem Funktionsbaustein MC_CamIn aktiviert. Die Kurbelachse folgt dann der virtuellen Leitachse (die translatorische Einheiten ausgibt), wobei die Transformationskurvenscheibe die translatorischen Positionswerte in Kurbelwinkelwerte umwandelt. Einzelheiten entnehmen Sie bitte .

Synchronbetrieb mit Kurvenscheibe

In dieser Applikation folgt eine translatorische Achse mit einem nutzerdefinierten Kurvenscheibenprofil einer Leitachse. Die anwenderdefinierte Kurvenscheibe bezieht sich nur auf die translatorische Achse. Das Übertragungsverhalten der Kurbel muss daher nicht in der anwenderdefinierten Kurvenscheibe berücksichtigt werden.

Statt dessen wird die anwenderdefinierte Kurvenscheibe durch den Funktionsbaustein [MB_CamTableCrankSuperimposed, Seite 430](#), mit einer Transformationskurvenscheibe überlagert, und die resultierende Kurvenscheibe wird durch das SPS-Programm an den Antrieb übertragen.

Umgekehrte Transformation

In den beiden oben beschriebenen Anwendungsbeispielen werden translatorische in rotatorische Einheiten umgewandelt (Sollwerterstellung). Damit die SPS jedoch anschließend Lage- bzw. Geschwindigkeitsbefehle verarbeiten kann, müssen die rotatorischen Einheiten wieder in translatorische Einheiten umgewandelt werden. Hierzu dient der Funktionsbaustein [MB_PhiToXvirt, Seite 435](#).

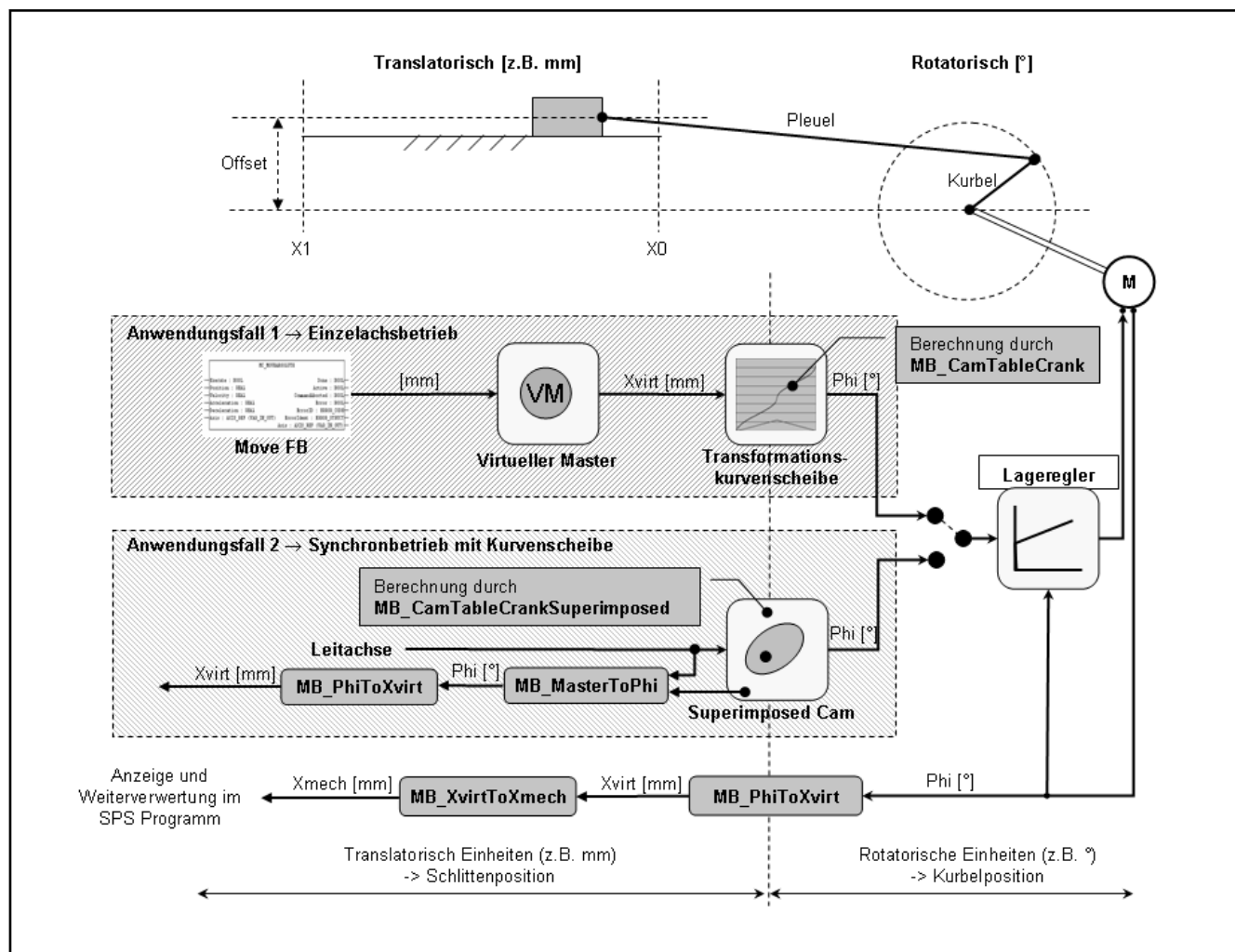


Abb. 6-16: Schematische Darstellung der Kurbelkinematik-Technologie-Funktionsbausteine

6.3.2 Allgemeine Definitionen

Definition der grundlegenden Variablen bei der Kurbelkinematik

Die nachstehenden Abbildungen stellen die Zusammenhänge zwischen Kurbel- und translatorischen Koordinaten sowie die Bedeutung verschiedener mechanischer Parameter dar.

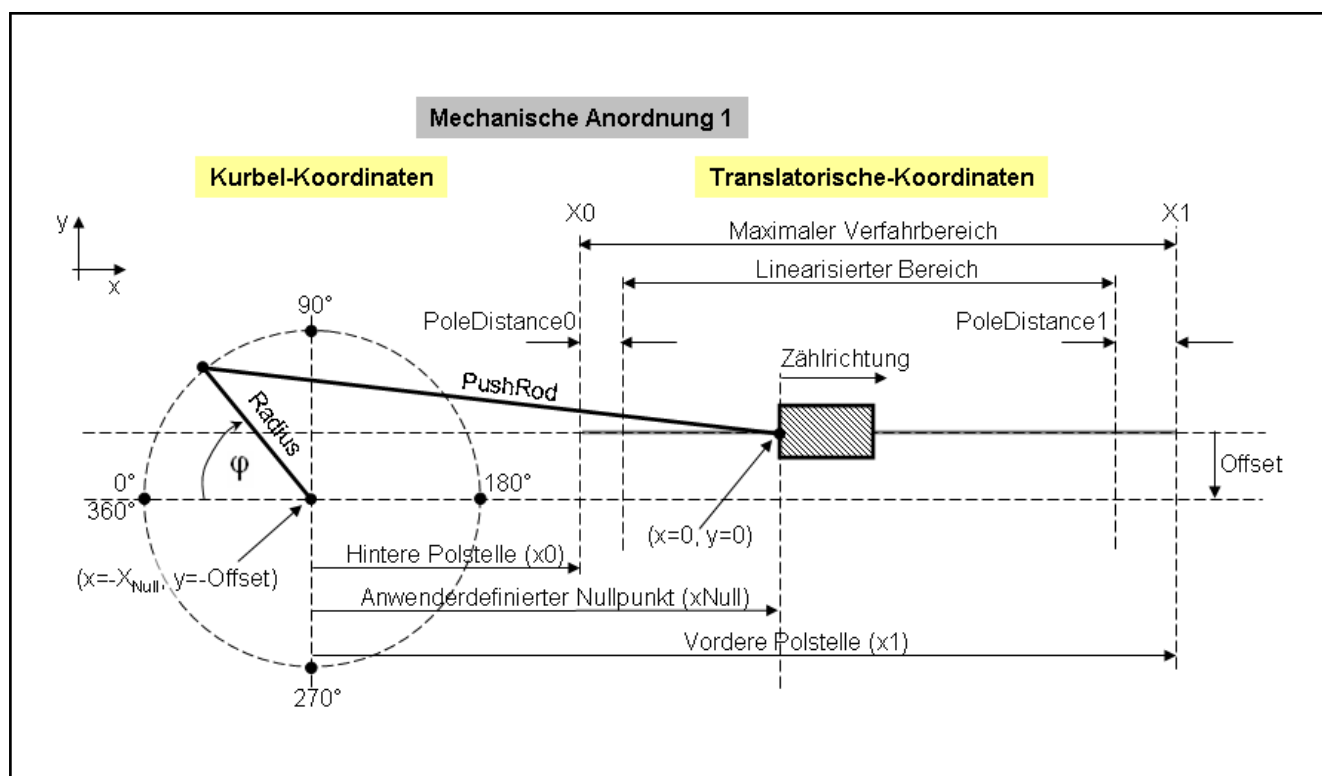


Abb. 6-17: Mechanische Anordnung 1: Zusammenhang von Kurbel- und translatorischen Koordinaten bei der Kurbelkinematik

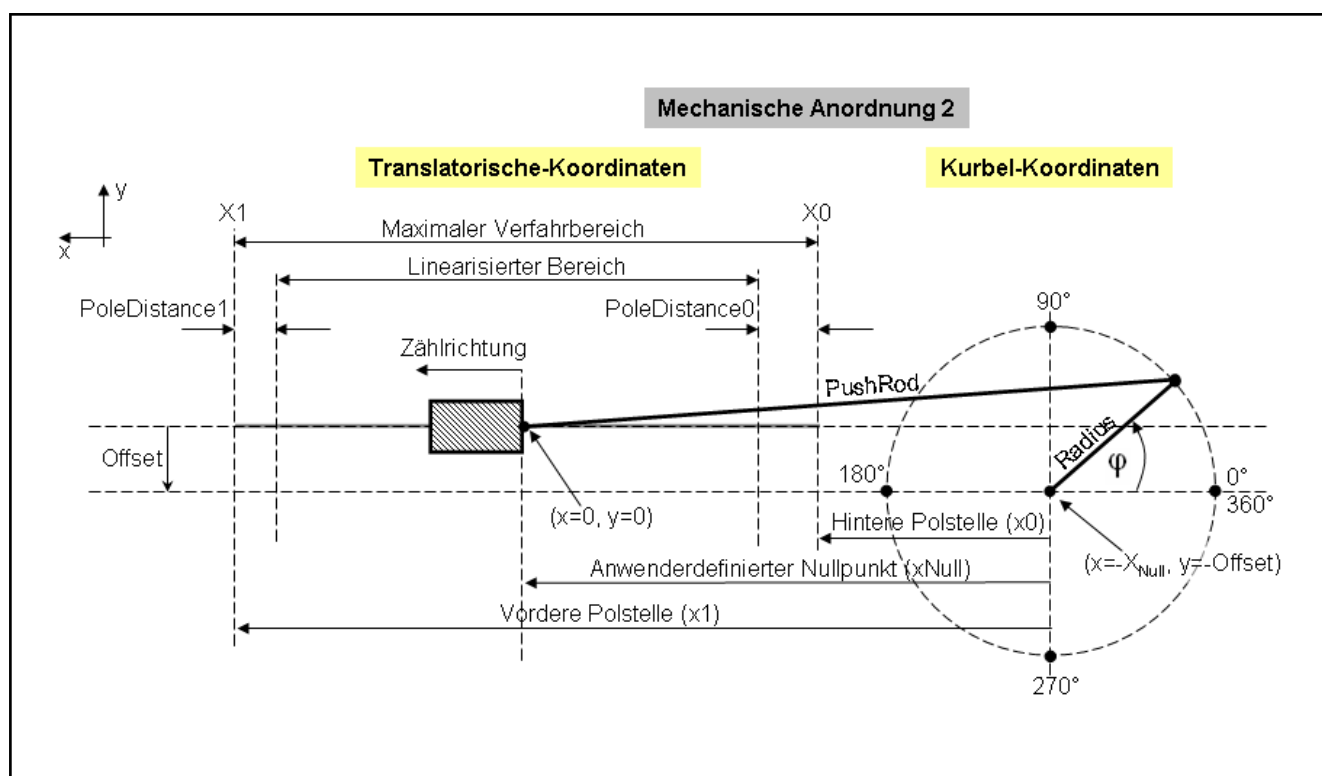


Abb. 6-18: Mechanische Anordnung 2: Zusammenhang von Kurbel- und translatorischen Koordinaten bei der Kurbelkinematik

Variable/Begriff	Definition
XNull	Abstand zwischen Kurbelachse und Schlitten-Nullpunkt
PoleDistance0 PoleDistance1	Abstand von der Polstelle X0 / X1 (jenseits des linearisierten Bereichs wird eine Kompensationsfunktion angewendet)
Maximaler Verfahrbereich	Verfahrbereich des translatorischen Koordinatensystems = X1-X0
Linearisierter Bereich	Im linearisierten Bereich ist kein Ersatzpolynom wirksam
Transformationskurvenscheibe	Wandelt translatorische (mm) Koordinaten in Kurbelwinkel (Grad) um

Tab. 6-27: Variable und Begriffe

Zählrichtung Für die in [Abb. 6-17 "Mechanische Anordnung 1: Zusammenhang von Kurbel- und translatorischen Koordinaten bei der Kurbelkinematik"](#) auf Seite 423 und [Abb. 6-18 "Mechanische Anordnung 2: Zusammenhang von Kurbel- und translatorischen Koordinaten bei der Kurbelkinematik"](#) auf Seite 423 abgebildeten mechanischen Anordnungen gelten allgemein die folgenden Zählrichtungen:

- Zählrichtung des Kurbelwinkels:
 - Mechanische Anordnung 1: Im Uhrzeigersinn (Rechtsdrehung), beginnend links mit 0° und endend mit 360°
 - Mechanische Anordnung 2: Im Gegenuhrzeigersinn (Linksdrehung), beginnend rechts mit 0° und endend mit 360°
- Zählrichtung des translatorischen Schlittens:
 - Mechanische Anordnung 1: Nach rechts hin zunehmende numerische Werte
 - Mechanische Anordnung 2: Nach links hin zunehmende numerische Werte

Mechanische (Xmech) und virtuelle (Xvirt) translatorische Position

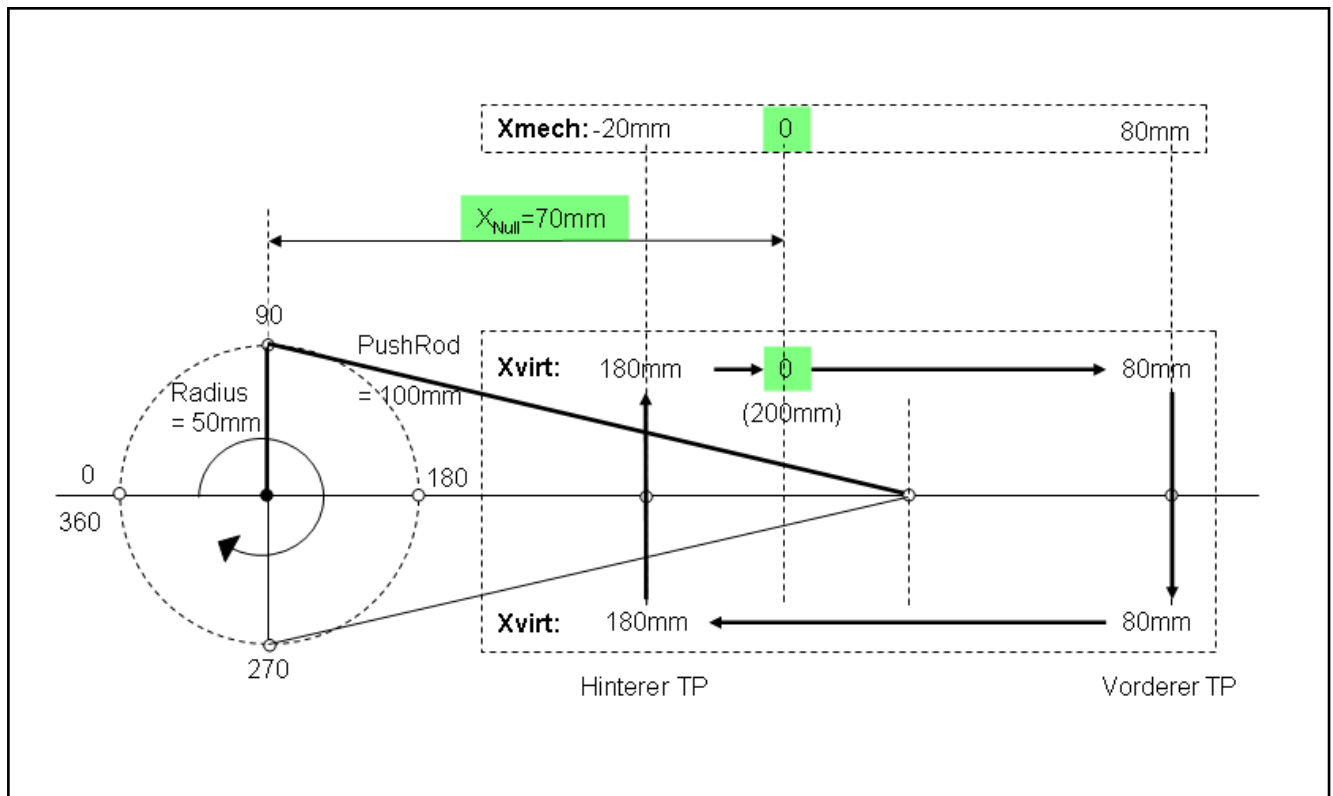
Mechanisch kann sich der Schlitten (translatorische Achse) zwischen der hinteren (X0) und der vorderen (X1) Polstelle bewegen. Zwischen diesen beiden Begrenzungen liegt der "maximale Verfahrbereich". Jeder mechanischen translatorischen Position (Xmech) entsprechen zwei verschiedene Kurbelwinkel. Deshalb dient die virtuelle translatorische Position (Xvirt) dazu, jedem Kurbelwinkel eine eindeutige translatorische Position zuzuordnen. Die virtuelle translatorische Position bewegt sich daher in positive Richtung, auch wenn die mechanische Position (Xmech) durch Überfahren der Polstelle die Richtung umkehrt. Der Moduloüberlauf von Xvirt wird vom Nutzer über den Nullpunkt (Xnull) definiert. Des Weiteren entspricht der Modulowert von Xvirt dem doppelten Verfahrbereich der Kurbelkinematik. Der Verfahrbereich wird wie folgt berechnet:

$$\text{Travelrange} = (\text{FrontPolePoint} - \text{RearPolePoint})$$

$$\text{Travelrange} = \sqrt{(\text{Radius} + \text{PushRod})^2 - \text{Offset}^2} - \sqrt{(\text{Radius} - \text{PushRod})^2 - \text{Offset}^2}$$

Abb. 6-19: Verfahrbereichsgleichung für die Kurbelkinematik

Die folgende Abbildung stellt Xvirt und Xmech an einem Beispiel dar:

Abb. 6-20: Zählrichtung der Schlittenpositionen (X_{virt} und X_{mech})

6.3.3 MB_CamTableCrank

Kurzbeschreibung

Der Funktionsbaustein MB_CamTableCrank berechnet anhand der kurbel-spezifischen Eingangswerte eine Transformationskurvenscheibe (mit 1024 Datenpunkten). Diese Transformationskurvenscheibe wandelt translatorische Positionswerte in rotatorische Positionswerte (Kurbelwinkel) zur Verwendung in einer Kurbelkinematik um. Sie ermöglicht die Kopplung einer translatorischen virtuellen Leitachse mit einem rotatorischen Kurbelantrieb. Einzelheiten entnehmen Sie bitte [Einführung und Übersicht, Seite 420](#).

Im Einzelachsbetrieb bewegt sich die virtuelle Achse in translatorischen Einheiten, während der Antrieb der Transformationskurvenscheibe folgt.

Eine Bewegung über den linearisierten Bereich hinaus kann durch Vorgabe je eines Wertes PoleDistance0 und PoleDistance1 realisiert werden. Innerhalb des Bereiches "PoleDistance" wird auf die Position eine Ausgleichsfunktion (Ersatzpolynom 5. Ordnung) angewendet, die den Kurbelantrieb approximiert und die resultierende Antriebsdynamik begrenzt. Beim Übergang in den und aus dem linearisierten Lagebereich sind Geschwindigkeit und Beschleunigung konstant.

Der Funktionsbaustein stellt die berechnete Transformationskurvenscheibe über den VAR_IN_OUT "CamTable" zur Verfügung. Das SPS-Programm muss diese Kurvenscheibentabelle in den Antrieb laden (z. B. über MB_WriteListeParameter), damit sie verwendet werden kann (z. B. von MC_CamIn Funktionsbaustein).



Betrifft Anwendung der Punktetabelle mit dem **FlexProfile**:

Je nach FB-Eingangsdaten ist der Startwert der berechneten Punktetabelle $\neq 0\%$.

Bei Aktivierung des FlexProfiles wird jedoch der Startwert = 0 wirksam.

Aktivieren Sie daher das FlexProfile mit dem Eingang:

SlaveOffset = (360°/100%)* erster Punkt der Punktetabelle [%]

Damit wird der korrekte Startwert und damit Kurbelwinkel aktiv.

Die Tabellenwerte liegen im Format "ARRAY OF DINT" vor und sind für 6 Nachkommastellen berechnet. Bei der o.g. Berechnung muss daher noch durch 1.000.000 dividiert werden.

Codebeispiel:

```
fbFlexProfile.SlaveOffset:=
    3.6*DINT_TO_REAL(arCamTable[0])/1000000;
```



Die Steigung der berechneten Punktetabelle wird im Bereich der Polstellen umso größer, je kleiner die Eingänge PoleDistance0 und PoleDistance1 gewählt werden.

Im Bereich der Polstelle wird damit die Geschwindigkeit höher. Dadurch kann der Steuerungsfehler F5113002, Sollgeschwindigkeit zu hoch für Zykluszeit, resultieren.

Sie können den F5113002-Fehler wie folgt beheben

- Motion-Zykluszeit verringern (wenn möglich)
- Leitachsgeschwindigkeit reduzieren (wenn möglich)
- PoleDistance0 und PoleDistance1 Eingang vergrößern

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-28: Referenztable MB_CamTableCrank

Schnittstellenbeschreibung

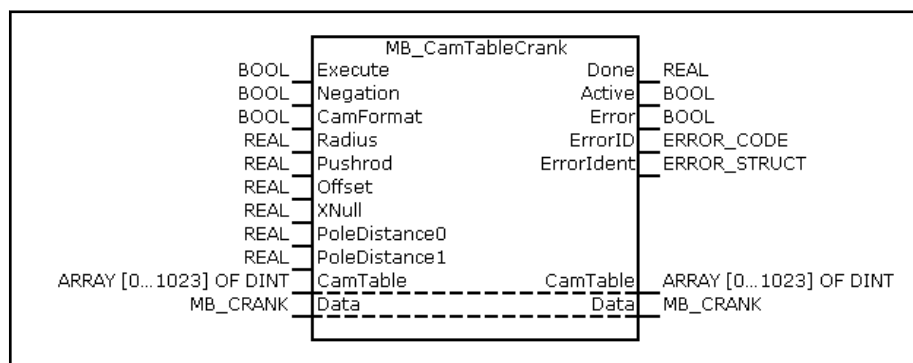


Abb. 6-21: Funktionsbaustein MB_CamTableCrank

E/A-Typ	Name	Datentyp	Erläuterung
VAR_INPUT	Execute	BOOL	Durch eine positive Flanke wird die Berechnung der Transformationskurvenscheibe ausgelöst
	Negation	BOOL	Invertiert die Zählrichtung einer translatorischen Achse
	CamFormat	BOOL	TRUE = Neues Format (letzter Punkt = erster Punkt = 360°) FALSE = altes Format (letzter Punkt = 360° - d)
	Radius	REAL	Kurbellänge in [mm]
	Pushrod	REAL	Länge des Pleuels in [mm]
	Offset	REAL	Versatz der Schlittenebene gegenüber dem Kurbelmittelpunkt [mm]
	XNull	REAL	Abstand zwischen Kurbelmittelpunkt und Schlitten-Nullpunkt [mm]
	PoleDistance0	REAL	Abstand zwischen der hinteren Polstelle X0 [mm] und dem linearisierten Bereich, in dem die Bewegung durch die begrenzte Antriebsdynamik beeinflusst wird
	PoleDistance1	REAL	Abstand zwischen der vorderen Polstelle X1 [mm] und dem linearisierten Bereich, in dem die Bewegung durch die begrenzte Antriebsdynamik beeinflusst wird
VAR_OUTPUT	Done	BOOL	Berechnung abgeschlossen, Kurvenscheibentabellen- und Ausgangsdaten (Data) sind gültig
	Active	BOOL	Funktionsbaustein wird gerade ausgeführt
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorStruct	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 429
VAR_IN_OUT	CamTable	ARRAY[0..1023] OF DINT	Array mit den Daten der errechneten Transformationskurvenscheibe
	Data	MB_CRANK	Struktur mit intern berechneten Daten der Kurbelkinematik. Die Daten werden in diesem Funktionsbaustein berechnet und an andere Funktionsbausteine übergeben MB_CRANK, Seite 429

Tab. 6-29: Schnittstellenvariablen Funktionsbaustein MB_CamTableCrank

Randbedingung

Die folgenden Randbedingungen müssen erfüllt werden, um SPS-Fehlermeldungen zu vermeiden:

Randbedingung	Grund
$Pleuel \geq \text{Radius} + \text{Offset} $	Wird diese Randbedingung nicht erfüllt, kann der translatorische Schlitten mechanisch nicht auf einer konstanten y-Koordinate verfahren
$(XNull \geq \text{hinterer TP}) \ \& \ (XNull \leq \text{hinterer TP} + 2 \times \text{Verfahrbereich})$	Der Nullpunkt muss innerhalb des Verfahrbereichs liegen

Tab. 6-30: Randbedingungen

Um den translatorischen Schlitten im Einzelachsbetrieb verfahren zu können, ist die folgende Konfiguration in IndraWorks durchzuführen:

- Translatorische Wichtung der virtuellen Achse mit einem Modulwert von $(2 \times \text{Verfahrbereich})$.
- Rotatorische Wichtung des Kurbelantriebs (reale Achse) mit einem Modulwert von 360° .



Zur Konfiguration der Skalierungsfaktoren muss sich IndraWorks im Onlinebetrieb und der Antrieb im Parametriermodus befinden.

Anwendungsbeispiel

Das folgende Ablaufschema gilt für Einzelachsbetrieb:

1. Der Funktionsbaustein MB_CamTableCrank berechnet die Transformationskurvenscheibe.
2. Die berechnete Transformationskurvenscheibe wird durch den Funktionsbaustein MB_WriteListParameter auf den Kurbelantrieb geschrieben.

Zur Übertragung erwartet der Funktionsbaustein MB_WriteListParameter ein ARRAY OF REAL, der Funktionsbaustein MB_CamTableCrank erzeugt jedoch ein ARRAY OF DINT. Vor der Nutzung des MB_WriteListParameter muss daher das ARRAY OF DINT in ein ARRAY OF REAL konvertiert werden.

Das nachfolgende Codebeispiel zeigt die Konvertierung:

```
FOR i := 0 TO 1023 DO
  arRealCamTable[i] := DINT_TO_REAL(arDintCamTable[i])/1000000;
END_FOR
```

3. Über den Funktionsbaustein MC_Power wird der Kurbelantrieb eingeschaltet.
4. Mit dem Funktionsbaustein MC_Home den Kurbelantrieb referenzieren. Dieser Schritt ist erforderlich, wenn sich am Kurbelantriebsmotor kein Absolutgeber befindet.
5. Eingabe der Kurbelantriebs-Istposition in den Funktionsbaustein [MB_PhiToXvirt](#), [Seite 435](#).
Die Kurbel-Istposition wird von Kurbelwinkelgraden in einen virtuellen translatorischen Wert in mm umgerechnet.
6. Um die virtuelle Leitachse in die Istposition des Kurbelantriebs zu bewegen, muss nun der Ausgangswert Xvirt des Funktionsbausteins MB_PhiToXvirt an den Funktionsbaustein MC_MoveAbsolute für die virtuelle Achse übergeben werden. Nach dieser Abfolge befindet sich die virtuelle Leitachse in der Istposition des Kurbelantriebs, und der Kurbelantrieb ist – ohne eine Aufsynchronisierungsbewegung ausgeführt zu haben – mit der virtuellen Leitachse synchronisiert.

7. Umschaltung des Kurbelantriebs in den Kurvenscheiben-Betriebsmodus über MC_CamIn. Einstellen von CamShaftDistance = 360, Getriebeübersetzung auf 1:1, Auswählen der Transformationskurvenscheibe und der virtuellen Leitachse als Leitachse. Der Kurbelantrieb führt zur Synchronisation mit der virtuellen Leitachse keine Bewegung aus, da die virtuelle Leitachse in Schritt 6 auf die Kurbelposition eingestellt wurde. Wurde Schritt 6 nicht ausgeführt, führt der Kurbelantrieb eine dynamische Synchronisationsbewegung aus.
8. Nun kann die virtuelle Achse mit den Funktionsbausteinen MC_MoveAbsolute und MC_MoveVelocity im Einzelachsbetrieb bewegt werden (die Kurbelachse folgt nun der Leitachse und wendet dabei die Transformationskurvenscheibe an).

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0001	Pushrod-Länge zu klein. Folgende Bedingung muss erfüllt sein: $\text{Pushrod} > \text{Radius} + \text{ABS}(\text{Offset})$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0002	Xnull ist zu klein. Folgende Bedingung muss erfüllt sein: $\text{Xnull} > \text{SQRT}((\text{Pushrod}-\text{Radius})^2 - \text{Offset}^2)$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0003	Xnull ist zu groß. Folgende Bedingung muss erfüllt sein: $\text{Xnull} < \text{SQRT}((\text{Pushrod}-\text{Radius})^2 - \text{Offset}^2) + 2 \cdot \text{Verfahrbereich}$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	$\text{Radius} \leq 0$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	$\text{Pushrod} \leq 0$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0006	$\text{PoleDistance0} \leq 0$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0007	$\text{PoleDistance1} \leq 0$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0008	$\text{PoleDistance0} + \text{PoleDistance1} > \text{Verfahrbereich}$
CALCULATION_ERROR (16#0007)	16#1001	16#0005	Interner Mathematikfehler: $(\text{Xmech} + \text{Xnull})^2 + \text{Offset}^2 < 0$
CALCULATION_ERROR (16#0007)	16#1001	16#0006	Interner Mathematikfehler: $(\text{Radius}^2 + (\text{Xmech} + \text{Xnull})^2 + \text{Offset}^2 + \text{Pushrod}^2) / (2 \cdot \text{Radius} \cdot \text{SQRT}(\text{rVar1})) > 1$
CALCULATION_ERROR (16#0007)	16#1001	16#0007	Interner Mathematikfehler: $\text{Xmech} + \text{Xnull} = 0$

Tab. 6-31: Fehlercodes Funktionsbaustein MB_CamTableCrank

6.3.4 MB_CRANK

Die Struktur MB_Crank ist für den internen Datenaustausch zwischen Funktionsbausteinen und Funktionen zuständig.

Der Inhalt der Struktur MB_CRANK wird von den Funktionsbausteinen [MB_CamTableCrank](#), [Seite 425](#), und [MB_CamTableCrankSuperimposed](#), [Seite 430](#), berechnet und gilt für alle anderen relevanten Funktionsbausteine und Funktionen.

Typvereinbarung

```

TYPE MB_CRANK:
STRUCT
  X0: REAL;      (* Hintere Polstelle des linearen Schlittens in linearen Einheiten *)
  X1: REAL;      (* Vordere Polstelle des linearen Schlittens in linearen Einheiten *)

```

```

X01: REAL;    (* Eintrittsposition in den linearisierten Bereich an der hinteren Polstelle *)
X02: REAL;    (* Austrittsposition aus dem linearisierten Bereich an der hinteren Polstelle *)
X11: REAL;    (* Eintrittsposition in den linearisierten Bereich an der vorderen Polstelle *)
X12: REAL;    (* Austrittsposition aus dem linearisierten Bereich an der vorderen Polstelle *)

Sv: REAL;     (* Fahrbereich der translatorischen Achse = ABS(X1-X0) *)

Phi0: REAL;   (* Hintere Polstelle der Kurbel in rotatorischen Einheiten *)
Phi1: REAL;   (* Vordere Polstelle der Kurbel in rotatorischen Einheiten *)

Phi01: REAL;  (* Eintrittswinkel in den linearisierten Bereich an der hinteren Polstelle *)
Phi02: REAL;  (* Austrittswinkel aus dem linearisierten Bereich an der hinteren Polstelle *)
Phi11: REAL;  (* Eintrittswinkel in den linearisierten Bereich an der vorderen Polstelle *)
Phi12: REAL;  (* Austrittswinkel aus dem linearisierten Bereich an der vorderen Polstelle *)
END_STRUCT
END_TYPE

```

6.3.5 MB_CamTableCrankSuperimposed

Kurzbeschreibung Der Funktionsbaustein MB_CamTableSuperimposed überlagert die nutzerdefinierte Kurvenscheibe (CamInput) auf die Transformationskurvenscheibe und gibt die resultierende überlagerte Kurvenscheibe über den Ausgang CamOutput aus.

Die Transformationskurvenscheibe wird innerhalb des Funktionsbausteins berechnet, vergleichbar mit dem Funktionsbaustein [MB_CamTableCrank](#), Seite 425.

Das Überlagerungsprinzip ist in [Abb. 6-22 "Prinzip der Kurvenscheiben-Überlagerung"](#) auf Seite 431 gezeigt. Die nutzerdefinierte Kurvenscheibe muss 1024 Datenpunkte enthalten und die Bewegung der translatorischen Achse in Bezug auf die Leitachse definieren (ungeachtet der Kurbelkinematik).

Der 100-Prozent-Wert der Tabelle entspricht der Bewegung

(2 x Verfahrbereich = Modulowert der virtuellen translatorischen Achse).

Die berechnete überlagerte Kurvenscheibe muss mit Hilfe des Funktionsbausteins MB_WriteListParameter auf den Kurbelantrieb geschrieben und vom SPS-Programm über den Funktionsbaustein MC_CamIn aktiviert werden.

Hinweise für den Anwender

Abhängig vom jeweiligen Endpunkt der nutzerdefinierten Kurvenscheibe werden die folgenden Fälle unterschieden:

- Liegt der Endpunkt der nutzerdefinierten Kurvenscheibe nahe bei 100%, führt die Kurbel keine Richtungsumkehr durch (sondern dreht sich kontinuierlich in die gleiche Richtung). Dies ist energetisch der günstigste Ablauf, da die natürliche Bewegung der Kurbel ausgenutzt wird.
- Beträgt der Endpunkt der nutzerdefinierten Kurvenscheibe 0%, findet eine Hin- und Herbewegung der translatorischen Achse statt, bei der die Kurbel fortwährend ihre Drehrichtung ändert (siehe Abbildung unten).

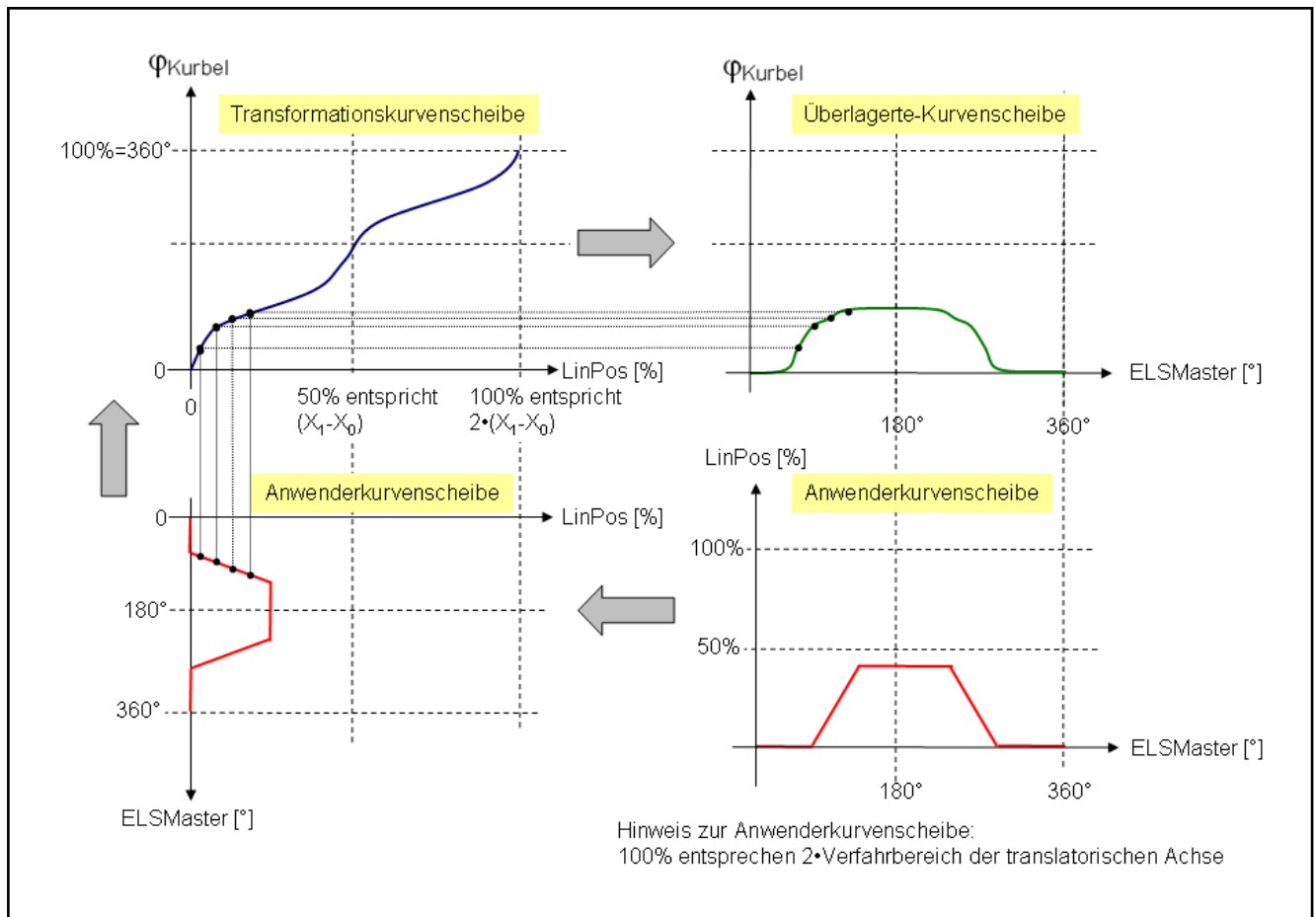


Abb. 6-22: Prinzip der Kurvenscheiben-Überlagerung



Betrifft Anwendung der Punktetabelle mit dem **FlexProfile**:

Je nach FB-Eingangsdaten ist der Startwert der berechneten Punktetabelle $\leq 0\%$.

Bei Aktivierung des FlexProfiles wird jedoch der Startwert = 0 wirksam.

Aktivieren Sie daher das FlexProfile mit dem Eingang:

SlaveOffset = $(360^\circ/100\%) \cdot \text{erster Punkt der Punktetabelle } [\%]$

Damit wird der korrekte Startwert und damit Kurbelwinkel aktiv.

Die Tabellenwerte liegen im Format "ARRAY OF DINT" vor und sind für 6 Nachkommastellen berechnet. Bei der o.g. Berechnung muss daher noch durch 1.000.000 dividiert werden.

Codebeispiel:

```
fbFlexProfile.SlaveOffset:=
    3.6*DINT_TO_REAL(arCamTable[0])/1000000;
```



Die Steigung der berechneten Punktetabelle wird im Bereich der Polstellen größer, je kleiner die Eingänge PoleDistance0 und PoleDistance1 gewählt werden. Im Bereich der Polstelle wird damit die Geschwindigkeit höher. Dadurch kann der Steuerungsfehler "F5113002, Sollgeschwindigkeit zu hoch für Zykluszeit" resultieren.

Mit folgenden Maßnahmen können Sie den F5113002 Fehler beheben:

- Motion Zykluszeit verringern (wenn möglich)
- Leitachsgeschwindigkeit reduzieren (wenn möglich)
- PoleDistance0 und PoleDistance1 Eingang vergrößern

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-32: Referenztabelle MB_CamTableCrankSuperimposed

Schnittstellenbeschreibung

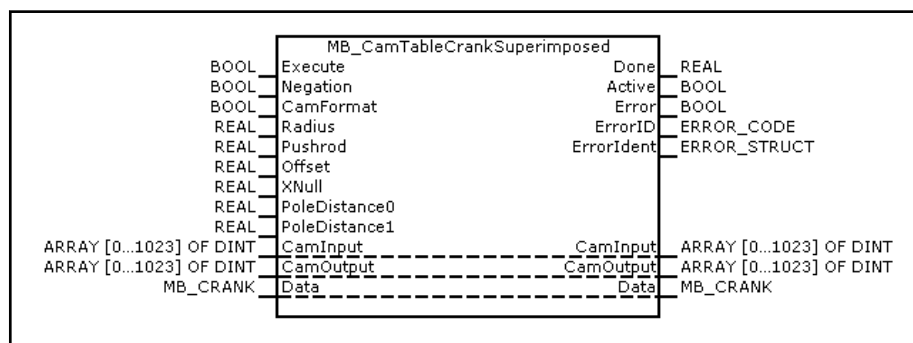


Abb. 6-23: Funktionsbaustein MB_CamTableCrankSuperimposed

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Durch eine positive Flanke wird die Berechnung der überlagerten Kurvenscheibe ausgelöst
	Negation	BOOL	Invertiert die Zählrichtung einer translatorischen Achse
	CamFormat	BOOL	TRUE = Neues Format (letzter Punkt = erster Punkt = 360°) FALSE = altes Format (letzter Punkt = 360° - d)
	Radius	REAL	Kurbellänge in [mm]
	Pushrod	REAL	Länge des Pleuels in [mm]
	Offset	REAL	Versatz der Schlittenebene gegenüber dem Kurbelmittelpunkt [mm]
	XNull	REAL	Abstand zwischen Kurbelmittelpunkt und Schlitten-Nullpunkt [mm]
	PoleDistance0	REAL	Abstand zwischen der hinteren Polstelle X0 [mm] und dem linearisierten Bereich, in dem die Bewegung durch die begrenzte Antriebsdynamik beeinflusst wird

E/A-Typ	Name	Datentyp	Beschreibung
	PoleDistance1	REAL	Abstand zwischen der hinteren Polstelle X1 [mm] und dem linearisierten Bereich, in dem die Bewegung durch die begrenzte Antriebsdynamik beeinflusst wird
VAR_OUTPUT	Done	BOOL	Berechnung abgeschlossen, Kurvenscheibentabellen- und Ausgangsdaten (Data) sind gültig
	Active	BOOL	Funktionsbaustein wird gerade ausgeführt
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorStruct	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 434
VAR_IN_OUT	CamInput	ARRAY[0..1023] OF DINT	Array mit den Daten der nutzerdefinierten Kurvenscheibe zur Voreinstellung des translatorischen Bewegungsprofils
	CamOutput	ARRAY[0..1023] OF DINT	Array mit den Daten der berechneten überlagerten Kurvenscheibe zur Aktivierung der Kurbel
	Data	MB_CRANK	Struktur mit intern berechneten Daten der Kurbelkinematik. Die Daten werden in diesem Funktionsbaustein berechnet und dann von anderen Funktionsbausteinen verwendet MB_CRANK, Seite 429

Tab. 6-33: Schnittstellenvariablen Funktionsbaustein MB_CamTableCrankSuperimposed

Randbedingung Die folgenden Randbedingungen müssen erfüllt werden, um SPS-Fehlermeldungen zu vermeiden:

Randbedingung	Grund
$Pleuel \geq Radius + Offset $	Wird diese Randbedingung nicht erfüllt, kann der translatorische Schlitten mechanisch nicht auf einer konstanten y-Koordinate verfahren
$(XNull \geq \text{hinterer TP}) \ \& \ (XNull \leq \text{hinterer TP} + 2 \times \text{Verfahrbereich})$	Der Nullpunkt muss innerhalb des Verfahrbereichs liegen

Tab. 6-34: Randbedingungen

Um den translatorischen Schlitten im Einzelachsbetrieb verfahren zu können, ist die folgende Konfiguration in IndraWorks durchzuführen:

- Translatorische Wichtung der virtuellen Achse mit einem Modulowert von (2 x Verfahrbereich).
- Rotatorische Wichtung des Kurbelantriebs (reale Achse) mit einem Modulowert von 360°.

Um den translatorischen Schlitten im Synchronbetrieb mit einer vorgegebenen, nutzerdefinierten Kurvenscheibe verfahren zu können, ist die folgende Konfiguration in IndraWorks durchzuführen:

- Rotatorische Wichtung des Kurbelantriebs (reale Achse) mit einem Modulowert von 360°.



Zur Konfiguration der Skalierungsfaktoren muss sich IndraWorks im Onlinebetrieb und der Antrieb im Parametriermodus befinden.

Anwendungsbeispiel überlagerte Kurvenscheibe

Das folgende Ablaufschema gilt für eine kurvenscheibensynchrone Applikation:

1. Eingabe einer nutzerdefinierten Kurvenscheibe über den Eingang CamInput des Funktionsbausteins (z. B. durch Einlesen einer Antriebs-Kurvenscheibe oder Auslesen aus einer Datei usw.)
2. Ausgabe einer überlagerten Kurvenscheibe durch den Funktionsbaustein MB_CamTableCrankSuperimposed, die durch Überlagerung einer nutzerdefinierten Kurvenscheibe in eine Transformationskurvenscheibe berechnet wurde (siehe [Abb. 6-22 "Prinzip der Kurvenscheiben-Überlagerung" auf Seite 431](#)).
3. Die berechnete überlagerte Kurvenscheibe wird durch den Funktionsbaustein MB_WriteListParameter auf den Kurbelantrieb geschrieben.
4. Über den Funktionsbaustein MC_Power den Kurbelantrieb einschalten.
5. Mit dem Funktionsbaustein MC_Home den Kurbelantrieb referenzieren. Dieser Schritt ist erforderlich, wenn kein Absolutgeber verwendet wird.
6. Der Lagesollwert der überlagerten Kurvenscheibe und die aktuelle Kurbelantriebsposition können auseinander liegen. Zur Zuschaltung der Kurvenscheibenbetriebsart (über MC_CamIn, mit den Einstellwerten CamShaftDistance=360, Getriebeübersetzung 1:1 und Auswahl der überlagerten Kurvenscheibe) bestehen zwei Möglichkeiten:
 - Zuschaltung der Kurvenscheibenbetriebsart am Kurbelantrieb ohne vorherige Positionskalibrierung. Der Kurbelantrieb führt dann eine dynamische Aufsynchroisation durch.
 - Synchronisieren des Kurbelantriebs mit der Leitachse durch Eingabe der Leitachsposition und der überlagerten Kurvenscheibe in den Funktionsbaustein MB_MasterToPhi. Anschließend – vor Umschalten auf Synchronbetrieb – muss der Kurbelantrieb mit dem Funktionsbaustein MC_MoveAbsolute so positioniert werden, dass er der Ausgabeposition des Funktionsbausteins MB_MasterToPhi entspricht.

Fehlerbehandlung

Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0001	Pushrod-Länge zu klein. Folgende Bedingung muss erfüllt sein: Pushrod > Radius + ABS(Offset)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0002	Xnull ist zu klein. Folgende Bedingung muss erfüllt sein: Xnull > SQRT((Pushrod-Radius)^2-Offset^2)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0003	Xnull ist zu groß. Folgende Bedingung muss erfüllt sein: Xnull < SQRT((Pushrod-Radius)^2-Offset^2)+2*Verfahrbereich
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0005	Pushrod <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0006	PoleDistance0 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0007	PoleDistance1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0008	PoleDistance0+PoleDistance1 > Verfahrbereich
CALCULATION_ERROR (16#0007)	16#1001	16#0005	Interner Mathematikfehler: (Xmech+Xnull)^2+Offset^2 < 0

ErrorID	Additional1	Additional2	Beschreibung
CALCULATION_ERROR (16#0007)	16#1001	16#0006	Interner Mathematikfehler: $(\text{Radius}^2 + (\text{Xmech} + \text{Xnull})^2 + \text{Offset}^2 + \text{Pushrod}^2) / 2 * \text{Radius} * \text{SQRT}(\text{rVar1}) > 1$
CALCULATION_ERROR (16#0007)	16#1001	16#0007	Interner Mathematikfehler: $\text{Xmech} + \text{Xnull} = 0$

Tab. 6-35: Fehlercodes Funktionsbaustein MB_CamTableCrankSuperimposed

6.3.6 MB_PhiToXvirt

Kurzbeschreibung

Der Funktionsbaustein MB_PhiToXvirt wandelt Kurbelwinkelwerte (Phi) in virtuelle translatorische Positionswerte (Xvirt) um.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx08 und MLD mit MPx17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-36: Referenztabelle MB_PhiToXvirt

Schnittstellenbeschreibung

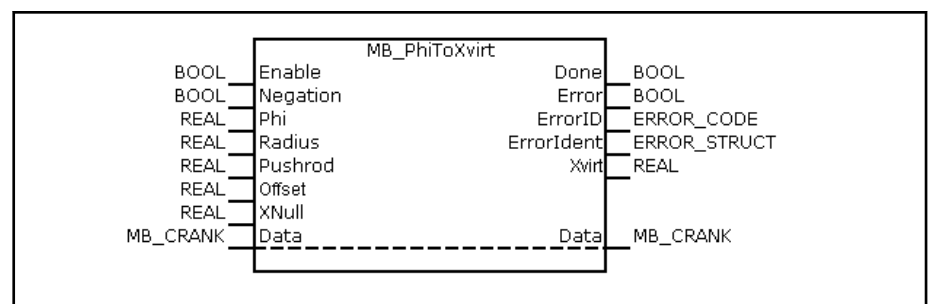


Abb. 6-24: Funktionsbaustein MB_PhiToXvirt

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Berechnung von Xvirt in jedem Zyklus, während Enable=TRUE
	Negation	BOOL	Invertiert die Zählrichtung der translatorischen Achse
	Phi	REAL	Kurbelposition in [°]
	Radius	REAL	Kurbellänge in [mm]
	Pushrod	REAL	Länge des Pleuels in [mm]
	Offset	REAL	Versatz der Schlittenebene gegenüber dem Kurbelmittelpunkt [mm]
	XNull	REAL	Abstand zwischen Kurbelmittelpunkt und Schlitten-Nullpunkt [mm]
VAR_OUTPUT	Done	BOOL	Berechnung abgeschlossen, Xvirt ist gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorStruct	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 436

E/A-Typ	Name	Datentyp	Beschreibung
	Xvirt	REAL	Virtuelle translatorische Position (Xvirt)
VAR_IN_OUT	Data	MB_CRANK	Struktur mit den internen Daten der Kurbelkinematik (zu berechnen vor MB_CamTableCrank bzw. MB_CamTableCrankSuperimposed) MB_CRANK, Seite 429

Tab. 6-37: Schnittstellenvariablen Funktionsbaustein MB_PhiToXvirt

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0001	Pushrod-Länge zu klein. Folgende Bedingung muss erfüllt sein: $\text{Pushrod} > \text{Radius} + \text{ABS}(\text{Offset})$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0002	Xnull ist zu klein. Folgende Bedingung muss erfüllt sein: $\text{Xnull} > \text{SQRT}((\text{Pushrod}-\text{Radius})^2 - \text{Offset}^2)$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0003	Xnull ist zu groß. Folgende Bedingung muss erfüllt sein: $\text{Xnull} < \text{SQRT}((\text{Pushrod}-\text{Radius})^2 - \text{Offset}^2) + 2 \cdot \text{Verfahrbereich}$
CALCULATION_ERROR (16#0007)	16#1001	16#0008	Interner Mathematikfehler: $\text{Pushrod}^2 - (\text{Radius} \cdot \sin(\text{Phi}) - \text{Offset})^2 < 0$

Tab. 6-38: Fehlercodes Funktionsbaustein MB_PhiToXvirt

6.3.7 MB_MasterToPhi

Kurzbeschreibung Die Funktion MB_MasterToPhi liefert den Kurbelwinkel (Phi), der anhand der Leitachspolition (Leitwert) und der (Überlagerungs-) Kurvenscheibe berechnet wurde. Die überlagerte Kurvenscheibe muss vor dem Aufruf dieser Funktion berechnet werden (durch [MB_CamTableCrankSuperimposed, Seite 430](#)).

Anhand des Ausgabewertes der Funktion MB_MasterToPhi kann der Kurbeltrieb vor der Umschaltung in den Synchronbetrieb so positioniert werden, dass er der Position der Leitachse entspricht.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-39: Referenztabelle MB_MasterToPhi

Schnittstellenbeschreibung

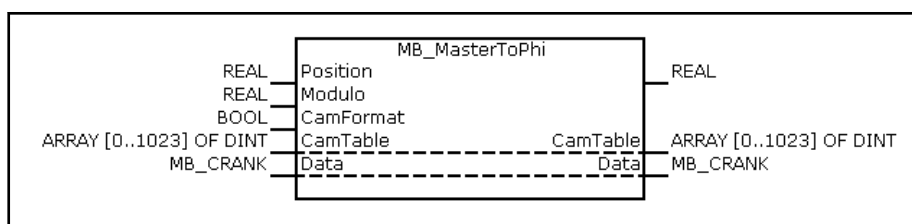


Abb. 6-25: Funktion MB_MasterToPhi

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Position	REAL	Leitachsposition
	Modulo	REAL	Modulowert der Leitachse
	CamFormat	BOOL	TRUE = Neues Format (letzter Punkt = erster Punkt = 360°) FALSE = altes Format (letzter Punkt = 360° - d)
Rückgabewert		REAL	Kurbelwinkel (Phi)
VAR_IN_OUT	CamTable	ARRAY[0..1023] OF DINT	Array mit den Daten der überlagerten Kurvenscheibe
	Data	MB_CRANK	Struktur mit den internen Daten der Kurbelkinematik (zu berechnen vor MB_CamTableCrank bzw. MB_CamTableCrankSuperimposed) MB_CRANK, Seite 429

Tab. 6-40: Schnittstellenvariablen Funktion MB_MasterToPhi

6.3.8 MB_XvirtToXmech

Kurzbeschreibung

Der Funktionsbaustein MB_XvirtToXmech wandelt die virtuelle translatorische Position (Xvirt) in die mechanische translatorische Position (Xmech) des Schlittens um. Xmech-Werte können zu Anzeigezwecken verwendet werden (z. B. auf einem HMI-Gerät).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechCrank
MLD mit MPx08 und MLD mit MPx17 mit MA Funktionspaket	MX_Packaging_07.lib

Tab. 6-41: Referenztafel MB_XvirtToXmech

Schnittstellenbeschreibung

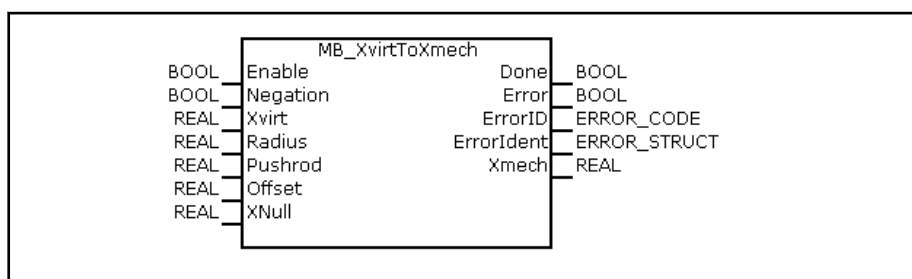


Abb. 6-26: Funktionsbaustein MB_XvirtToXmech

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Funktionsbaustein führt die Berechnung durch, solange Enable=TRUE
	Negation	BOOL	Invertiert die Zählrichtung einer translatorischen Achse
	Xvirt	REAL	Virtuelle translatorische Position in [mm]
	Radius	REAL	Kurbellänge in [mm]
	Pushrod	REAL	Länge des Pleuels in [mm]
	Offset	REAL	Versatz der Schlittenebene gegenüber dem Kurbelmittelpunkt [mm]
	XNull	REAL	Abstand zwischen Kurbelmittelpunkt und Schlitten-Nullpunkt [mm]
VAR_OUTPUT	Done	BOOL	Ausgangswert "Phi" ist gültig
	Error	BOOL	Fehler (siehe ErrorID und ErrorStruct)
	ErrorID	ERROR_CODE	Fehlerbeschreibung
	ErrorStruct	ERROR_STRUCT	Detaillierte Fehlerbeschreibung Fehlerbehandlung, Seite 438
	Xmech	REAL	Mechanische translatorische Position (Xmech) des Schlittens

Tab. 6-42: Schnittstellenvariablen Funktionsbaustein MB_XvirtToXmech

Fehlerbehandlung Der Funktionsbaustein erzeugt in Additional1/Additional2 der Tabelle "F_RELATED_TABLE", 16#0170, die folgenden Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0001	Pushrod-Länge zu klein. Folgende Bedingung muss erfüllt sein: Pushrod > Radius + ABS(Offset)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0002	Xnull ist zu klein. Folgende Bedingung muss erfüllt sein: Xnull > SQRT((Pushrod-Radius)^2-Offset^2)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0003	Xnull ist zu groß. Folgende Bedingung muss erfüllt sein: Xnull < SQRT((Pushrod-Radius)^2-Offset^2)+2*Verfahrbereich

Tab. 6-43: Fehlercodes Funktionsbaustein MB_XvirtToXmech

7 ML_TechTensionAdvanced.library

7.1 Einführung und Übersicht

Die Bibliothek ML_TechTensionAdvanced stellt Funktionsbausteine für Multiachs-Zugkraftregler mit Entkopplungsnetzwerk zur Verfügung.



Die Verwendung der Funktionsbausteine der Bibliothek ML_TechTensionAdvanced unterliegen einer **Lizenzpflicht**.

Die entsprechende Lizenz FWS-IM*MLC*-TEC-NNVRS-NN-TENS*ADV muss zur Nutzung der Funktionsbausteine erworben werden.

Pro Steuerung, auf der die Funktionsbausteine verwendet werden, ist jeweils eine Lizenz erforderlich (Runtime-Lizenz), die nur zur Verwendung auf diesem Steuerungssystem berechtigt.

Die Funktionsbausteine von ML_TechTensionAdvanced

Anwendungsgebiet (Ordner in der Bibliothek)	
Funktionsbaustein	Beschreibung
Parameteridentifikation für Zugkraftregler mit einfacher Parametrierung	
MB_TensionIdentModelParamType01, Seite 439	Funktionsbaustein zur online Identifikation der relevanten Regelstreckenparameter
Multiachs-Zugkraftregler	
MB_TensionControlLoadCellType02, Seite 447	Multiachs-Zugkraftregler-Funktionsbaustein mit dynamischer Vorsteuerung für bis zu 8 Achsen
MB_TensionControlLoadCellMatParType02, Seite 463	Multiachs-Zugkraftregler-Funktionsbaustein mit einfacher Parametrierung und dynamischer Vorsteuerung für bis zu 8 Achsen
Entkopplungsnetzwerk für Multiachs-Zugkraftregler	
MB_TensionDecouplingNetworkType01, Seite 479	Entkopplungsnetzwerk-Funktionsbaustein für Multiachs-Zugkraftregler

Tab. 7-1: Funktionsbausteine der Bibliothek ML_TechTensionAdvanced (lizenzpflichtig)

7.2 MB_TensionIdentModelParamType01

Kurzbeschreibung

Der Identifikationsbaustein MB_TensionIdentModelParamType01 ermittelt anhand der resultierenden Bahnzugkraft, infolge einer Systemanregung, die individuellen Streckenparameter einer Bahnzugkraftregelstrecke.

Diese online identifizierten Parameter (Streckenverstärkung K_s , Verzugszeit T_u und Ausgleichszeit T_g) können anschließend über die Struktur MB_Tension_CONTROL_MATERIAL (K_s), und die Parameter DelayFeedbackValue (T_u) und DistanceRollToRoll ($\sim T_g$) den Funktionsbausteinen MB_TensionControlLoadCellMatParType01, ML_TechMotion.library und [MB_TensionControlLoadCellMatParType02, Seite 463](#), übergeben werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechTensionAdvanced

Tab. 7-2: Referenztable MB_TensionIdentModelParamType01



Die Verwendung des Funktionsbausteines unterliegt einer Lizenzpflicht (Details siehe [Einführung und Übersicht, Seite 439](#)).

Schnittstellenbeschreibung

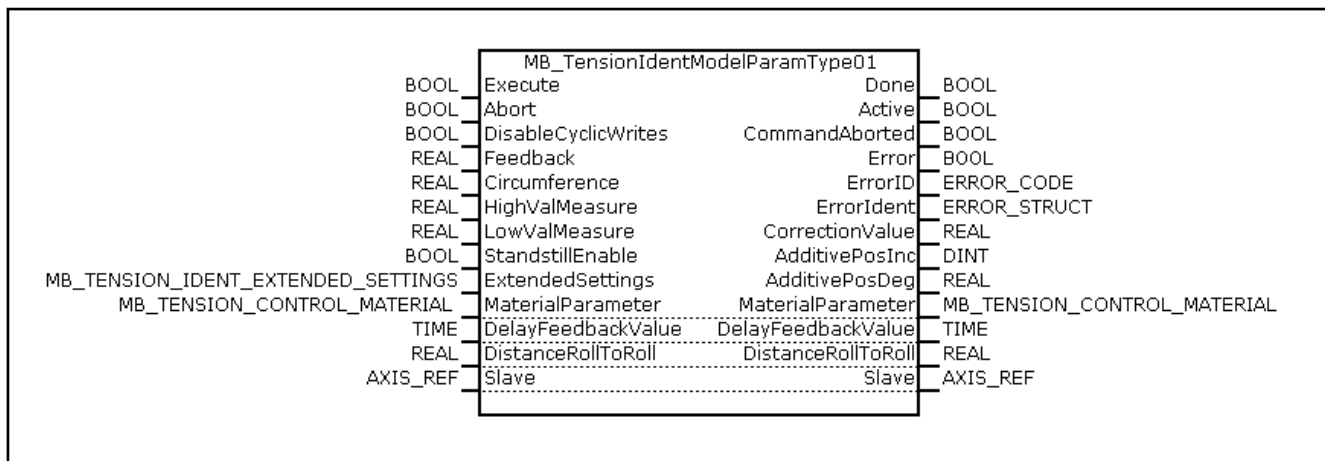


Abb. 7-1: Funktionsbaustein MB_TensionIdentModelParamType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert).
	Abort	BOOL	Abbruch der laufenden Identifikation.
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben.
	Feedback	REAL	Aktueller Istwert der Zugkraft [N]*.
	Circumference	REAL	Achsumfang der Slaveachse [mm]*.
	HighValMeasure	REAL	Maximalwert den die Bahnzugkraft während der Identifikation erreichen soll [N]*.
	LowValMeasure	REAL	Minimalwert den die Bahnzugkraft während der Identifikation erreichen soll [N]*.
	StandstillEnable	BOOL	StandstillEnable = FALSE: Identifikation bei Bahntransport, Stellgröße: Getriebefeinabgleich. StandstillEnable = TRUE: Identifikation im Stillstand, Stellgröße: Leitachspannung additiv.
	ExtendedSettings	MB_TENSION_IDENT_EXTENDED_SETTINGS	Struktur, die individuelle Einstellungsmöglichkeiten der Identifikation beinhaltet. MB_TENSION_IDENT_EXTENDED_SETTINGS, Seite 447

E/A-Typ	Name	Datentyp	Beschreibung
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig.
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig.
	CommandAborted	BOOL	Bearbeitung durch Eingang Abort abgebrochen.
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen. Mit "Enable" = FALSE wird der Fehler gelöscht.
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung.
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung gemäß Fehlertabelle (siehe Fehlercodes , Seite 445).
	CorrectionValue	REAL	Stellsignal bei Bahntransport (StandstillEnable = FALSE) [%].
	AdditivePosInc	DINT	Stellsignal bei Stillstand (Standstill = TRUE). Additive Leitachseposition in Inkrementen, genutzt für reale Slave-Achse mit Interpolation auf dem Antrieb (Zielparame-ter: P-0-0692) [-].
	AdditivePosDeg	REAL	Stellsignal bei Stillstand (Standstill = TRUE). Additive Leitachseposition in Grad, genutzt für Slave-Achse mit Interpolation auf der Steuerung (z.B. für virtuelle Achsen oder reale Achse mit Interpolation auf der Steuerung, Zielparame-ter: A-0-2600) [°].
VAR_IN_OUT	MaterialParameter	MB_TENSION_CONTROL_MATERIAL	Struktur mit materialspezifischen Daten (Typ, Breite, Dicke, Grammatur und E-Modul). MB_TENSION_CONTROL_MATERIAL , Seite 305
	DelayFeedbackValue	TIME	Verzugszeit T_u der Bahnzugkraftregelstrecke. Summe aller peripherer Zeitkonstanten. U.a. Motordynamik, Filterzeiten, Messaufnehmer, Motionzykluszeit.
	DistanceRollToRoll	REAL	Materialbahnlänge zwischen den begrenzenden Klemmstellen [mm]*.
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Referenz zur Folgeachse.

*

Bei der Verwendung von benutzerdefinierten Elastizitätsmoduln können beliebige Einheiten wie z.B. [N], [kg] und / oder [mm], [ft] verwendet werden. Es ist darauf zu achten, dass die gewählten Einheiten einheitlich verwendet werden.

Tab. 7-3: Schnittstellenvariablen Funktionsbaustein MB_TensionIdentModel-ParamType01

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL	FALSE	TRUE	FALSE	kontinuierlich
Abort	BOOL	FALSE	TRUE	FALSE	kontinuierlich
DisableCyclicWrites	BOOL	FALSE	TRUE	FALSE	steigende Flanke an "Execute"
Feedback	REAL	n.def	n.def	0.0	kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Circumference	REAL	>0.0	n.def	100.0	steigende Flanke an "Execute"
HighValMeasure	REAL	>LowVal-Measure	n.def.	0.0	steigende Flanke an "Execute"
LowValMeasure	REAL	0.0	<HighVal-Measure	0.0	steigende Flanke an "Execute"
StandstillEnable	BOOL	FALSE	TRUE	TRUE	steigende Flanke an "Execute"
ExtendedSettings	MB_TENSION_IDENT_EXTENDED_SETTINGS	Siehe Datenstruktur	Siehe Datenstruktur	Siehe Datenstruktur	steigende Flanke an "Execute"
MaterialParameter	MB_TENSION_CONTROL_MATERIAL	Siehe Datenstruktur	Siehe Datenstruktur	Siehe Datenstruktur	steigende Flanke an "Execute"
DelayFeedbackValue	TIME	T#25ms	n.def.	Benutzerdefiniert	steigende Flanke an "Execute"
DistanceRollToRoll	REAL	>0.0	n.def	Benutzerdefiniert	steigende Flanke an "Execute"
Slave	AXIS_REF	siehe RIL_CommonTypes	siehe RIL_CommonTypes	siehe RIL_CommonTypes	steigende Flanke an "Execute"

Tab. 7-4: Eingangsverhalten MB_TensionIdentModelParamType01

Funktionsbeschreibung

Ein Bahnabschnitt wird immer durch zwei Klemmstellen begrenzt.

Für die Regelung der Bahnzugkraft wird jedoch typischer Weise ausschließlich eine dieser Klemmstellen verwendet.

Als Systemanregung für eine Identifikation schaltet der Funktionsbaustein MB_TensionIdentModelParamType01 entweder im Stillstand eine Lage ("StandstillEnable" := TRUE) bzw. während des Bahntransportes eine Umfangsgeschwindigkeit ("StandstillEnable" := FALSE) auf das gewählte Stellsignal auf. Anhand der auf diese Anregungen resultierenden Zugkraftänderung, kann der Funktionsbaustein die individuellen Streckenparameter ermitteln.

Je nach Achstyp und Betriebsmodus ("StandstillEnable") werden hierbei, falls das zyklische Schreiben nicht ausgeschaltet ist ("DisableCyclicWrites" := FALSE), entsprechende Parameter beschrieben.

Während des Bahntransportes wird als Stellsignal der Getriebefeinabgleich verwendet (P-0-0694 bei Interpolation auf dem Antrieb bzw. A-0-2605 bei Interpolation auf der Steuerung). Bei stehender Leitachse wird die Zugwalze über die additive Leitachsisposition bewegt, da in diesem Fall der Getriebefeinabgleich keinen Einfluss auf die Zugkraft hat. Je nach Achstyp wird hierbei der entsprechende Parameter beschrieben (P-0-0692 bei Interpolation auf dem Antrieb bzw. A-0-2600 bei Interpolation auf der Steuerung).

Inbetriebnahme

Ist "DisableCyclicWrites" inaktiv, müssen die beiden Parameter P-0-0694 (Getriebefeinabgleich) und P-0-0692 (additive Leitachsisposition) im zyklischen Sollwertkanal der Folgeachsen aufgelegt werden, wenn eine reale Slave-Achse mit Interpolation auf dem Antrieb verwendet wird. Bei steigender Flanke an "Execute" werden die zyklischen Datenkanäle überprüft. Wird einer der beiden Parameter nicht gefunden, erfolgt eine entsprechende Fehlermeldung.

an den Fehlerausgängen des Funktionsbausteins. Ist "DisableCyclicWrites" aktiv, wird die Konfiguration der zyklischen Sollwertdatenkanäle nicht überprüft.

Bei Slave-Achsen mit Interpolation auf der Steuerung (z.B. virtuelle Achse) ist die Konfiguration der zyklischen Datenkanäle nicht notwendig.



- Die verwendete Slave-Achse muss vor dem Aktivieren des "Execute"-Eingangs in eine synchrone Betriebsart gebracht werden. Dies kann durch das AchsInterface oder die PLCopen-Funktionsbausteine "MC_GearIn" oder "MB_GearInPos" geschehen.
- Die Master Achse muss Modulo 360 Grad gewichtet sein

Im Folgenden werden die einzelnen Funktionen der Eingänge des Funktionsbausteins näher erläutert. Allgemein gilt für alle Eingänge, dass sie ausschließlich bei einer steigenden Flanke am "Execute"-Eingang ausgewertet werden.

Ausnahme dieser Regel bildet der "Abort"- und "Feedback"-Eingang welche kontinuierlich übernommen werden. Das heißt, Änderungen dieser Eingänge sind sofort ohne erneutes Toggeln des "Execute"-Eingangs wirksam.

Abort Mittels einer steigenden Flanke am "Abort"-Eingang ist es möglich eine bereits gestartete Identifikation vorzeitig abubrechen. Dies ist beispielhaft sinnvoll, falls aus technischen oder applikativen Gründen die aus der Anregung resultierende Bahnzugkraft nicht am "Feedback"-Eingang des Funktionsbausteins ankommt.

DisableCyclicWrites Der Eingang "DisableCyclicWrites" bestimmt, ob die vom Funktionsbaustein berechneten Werte direkt über den zyklischen Kanal geschrieben werden (FALSE) oder nur am Bausteinenausgang verfügbar sind (TRUE).

Ist "DisableCyclicWrites" auf TRUE gesetzt, wird außerdem die Prüfung, ob die optional zyklischen Parameter auch konfiguriert sind, nicht durchgeführt.

Feedback Dieser Eingang stellt den Istwert der Zugkraft dar. Dieser kann z.B. durch einen Analogeingang am Antrieb oder den Inline I/O eingelesen werden. Der Anwender ist dabei für eine sinnvolle Skalierung verantwortlich.

Circumference Der Eingang "Circumference" bezieht sich auf den Umfang der "Slave"-Achse und dient zur internen Umrechnung zwischen einer Winkelverstellung und einer Wegänderung, beziehungsweise einer Winkelgeschwindigkeit und einer Umfangsgeschwindigkeit.

In Stillstand der Anlage ("StandstillEnable" := TRUE) wird mit diesem Umfang und der Bahnlänge ("DistanceRollToRoll") eine direkt proportionale Beziehung zwischen einer Winkeländerung und einer Dehnungsänderung hergestellt.

$$\Delta \varepsilon(t) = \frac{\text{Circumference}}{\text{DistanceRollToRoll}} \cdot \frac{A-0-2720}{A-0-2721} \cdot \frac{\Delta \varphi(t)}{360^\circ}$$

Abb. 7-2: Umrechnung zwischen einer Winkel- und Dehnungsänderung

HighValMeasure, LowValMeasure

Mit den Eingänge "HighValMeasure" und "LowValMeasure" kann der Anwender die minimal und maximal zulässige Bahnzugkraft während der Identifikation vorgeben. Infolge einer symmetrischen Anregung folgt - unter der Voraussetzung eines linearen Systems - auch eine symmetrische Reaktion der Bahnzugkraft um den stationären Zustand. Demzufolge wird in der Initialisierungsphase des Funktionsbausteins die maximal zulässige Auslenkung in

Abhängigkeit der stationären Bahnzugkraft und der vorgegebenen Grenzen anhand nachfolgender Formel berechnet.

$$\Delta F = \min [HighValMeasure - F_{\text{stat}} , F_{\text{stat}} - LowValMeasure]$$

Abb. 7-3: Maximale Auslenkung der Bahnzugkraft um den stationären Zustand

Während der Identifikation werden infolge dessen ausschließlich bei einer symmetrischen Vorgabe der Grenzen um den stationären Wert beide Grenzen tangiert. Ansonsten wird die Bahnzugkraft lediglich die Grenze mit der kleineren Distanz zum stationären Wert erreichen.

StandstillEnable Durch Setzen des Eingangs "StandstillEnable" kann zwischen einer Identifikation im Stillstand ("StandstillEnable":=TRUE) und einer Identifikation während des Materialbahntransportes ("StandstillEnable":=FALSE) unterschieden werden.

Im Stillstand der Anlage können hierbei sowohl die Streckenverstärkung Ks (MaterialEModul*WebThickness*WebWidth) als auch die Verzugszeit Tu ("DelayFeedbackValue") ermittelt werden.

Anschließend können im Arbeitspunkt die Streckenausgleichszeit Tg bzw. die hierzu proportionale effektive Bahnabschnittslänge ("DistanceRollToRoll"), und die aufgrund der Bearbeitung / klimatischer Einflüsse resultierende Streckenverstärkung Ks ermittelt werden.

ExtendedSettings Durch die Eingangsstruktur "ExtendedSettings" kann die Identifikation an die individuellen Bedürfnisse des Anwenders angepasst werden.

1. Der Eingang "ExtendedSettings.NumMeasureData" gibt die Anzahl der abschließenden redundanten Messungen für die Mittelwertbildung an.
2. Der Eingang "ExtendedSettings.AccuracyFactor" gibt die minimale Genauigkeit an, ab der die abschließende Mittelwertbildung beginnt. Zusätzlich wird die Mittelwertbildung wiederholt, falls ein Wert diese Genauigkeit überschreitet.
3. Der Eingang "ExtendedSettings.TimeFactor" gibt die Relation zwischen Anregungsfrequenz und Streckenreaktion an. Für eine adäquate Anregung der Systemzustände wird die Anregefrequenz um diesen Faktor langsamer gewählt als die zu erwartende Streckenzeitkonstante. Für eine Reduzierung der Identifikationszeit kann diese Relation verringert werden.
4. Der Eingang "ExtendedSettings.ExcitationFactor" gibt die maximale Erhöhung der Anregungsamplitude pro Iterationszyklus an. Falls eine Identifikation ohne a priori Informationen gestartet wird, kann durch eine Erhöhung dieses Wertes die Identifikationszeit reduziert werden.
5. Der Eingang "ExtendedSettings.EModuleSaftyFactor" gibt die Reduzierung der initialen Anregung, welche auf Basis des vorgegebenen E-Moduls berechnet wird, an. Wird Beispielhaft eine Identifikation mit den exakten Werten gestartet, so würde theoretisch die Anregungsamplitude ohne Sicherheitsfaktor dazu führen, dass die Systemreaktion genau die Grenzen "HighValMeasure" und/oder "LowValMeasure" tangiert. Erhöht sich nun das effektive E-Modul aufgrund von thermischen oder hygroskopischen Einflüssen, würden ohne Reduzierung bereits bei der initialen Anregung die vorgegebenen Grenzen über- bzw. unterschritten.
6. Der Eingang "ExtendedSettings.ErrorFactor" gibt den zusätzlichen Zugkraftbereich an, welcher ergänzend zu den Grenzen "HighValMeasure" und "LowValMeasure" maximal auftreten darf, ohne dass der Funktionsbaustein mit einer Fehlermeldung abbricht. Die maximal zulässigen

Grenzen können entsprechend nachfolgender Formel berechnet werden.

$$\begin{aligned} \text{MaxValue} &= (\text{HighValMeasure} - F_{\text{stat}}) \cdot (1 + \text{ErrorFactor} / 100\%) + F_{\text{stat}} \\ \text{MinValue} &= (\text{LowValMeasure} - F_{\text{stat}}) \cdot (1 + \text{ErrorFactor} / 100\%) + F_{\text{stat}} \end{aligned}$$

Abb. 7-4: Grenzen der Bahnzugkraft bevor der Funktionsbaustein mit einer Fehlermeldung abbricht

MaterialParameter

Die Struktur "MaterialParameter" beinhaltet materialspezifische Informationen und dient den Bahnzugkraftreglern MB_TensionControlLoadCellMatParType01 und MB_TensionControlLoadCellMatParType02 zur individuellen Berechnung der Reglerparameter.

([MB_TensionControlLoadCellMatParType01](#), Seite 292, und [MB_TensionControlLoadCellMatParType02](#), Seite 463)

Die einzelnen Elemente dieser Struktur beschreiben den Typ, die Breite, die Dicke bzw. die Grammatur und den Elastizitätsmodul des Bahnmaterials.

Bei Bahnmaterialien aus Papier werden hierbei die Grammatur "WebGrammage" und bei anderen Bahnmaterialien die Materialdicke "WebThickness" berücksichtigt.

Bei Auswahl eines definierten Bahnmaterials, wird das initiale Elastizitätsmodul aus der internen Datenbank bezogen. Bei Vorgabe eines benutzerdefinierten Elastizitätsmodul wird der Eingang "MaterialParameter.MaterialEModule" als initialer Wert herangezogen.

Nach erfolgreicher Identifikation schaltet der Funktionsbaustein bei Bedarf auf den entsprechenden benutzerdefinierten Materialtyp "CUSTOM_PAPER" oder "CUSTOM_NON_PAPER" um, und gibt das identifizierte Elastizitätsmodul über die Variable "MaterialParameter.MaterialEModule" aus.

DelayFeedbackValue

Der Ein-/ Ausgang "DelayFeedbackValue" dient den Bahnzugkraftreglern MB_TensionControlLoadCellMatParType01 und MB_TensionControlLoadCellMatParType02 zur internen Berechnung der Reglerparameter.

Der Ein-/ Ausgang beschreibt die Verzugszeit T_u der Bahnzugkraftregelstrecke. Somit entspricht dieser Wert der Summe aller peripherer Zeitkonstanten wie Motordynamik, Filterzeiten, Messaufnehmer, Motionzykluszeit.

DistanceRollToRoll

Der Ein-/ Ausgang "DistanceRollToRoll" dient den Bahnzugkraftreglern MB_TensionControlLoadCellMatParType01 und MB_TensionControlLoadCellMatParType02 zur internen Berechnung der Reglerparameter.

Der Ein-/ Ausgang ist proportional zur Ausgleichszeit T_g der Bahnzugkraftregelstrecke. Er beschreibt die individuelle Materialbahnlänge zwischen den beiden begrenzenden Klemmstellen.

Fehlercodes

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle F_RELATED_TABLE, 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0004)	16#0A00	16#0001	Der Parameter P-0-0694 ist nicht im optional zyklischen Telegramm der Folgeachse parametrier
ACCESS_ERROR (16#0004)	16#0A00	16#0002	Der Parameter P-0-692 ist nicht im optional zyklischen Telegramm der Folgeachse parametrier
DEVICE_ERROR (16#0008)	16#0A02	16#0004	Der Eingang StandstillEnable ist false, und die Geschwindigkeit der Slaveachse ist 0.

ErrorID	Additional1	Additional2	Beschreibung
DEVICE_ERROR (16#0008)	16#0A02	16#0005	Die Motionszykluszeit (C-0-0400) ist kleiner oder gleich 0
DEVICE_ERROR (16#0008)	16#0A02	16#0006	Die Getriebeübersetzung der geregelten Achse ist unzulässig (A-0-2720, A-0-2721)
INPUT_INVALID_ERROR (16#0001)	16#0A04	16#0003	Der am Eingang "MaterialParameter.Material-Type" angegebene Wert ist nicht im Datentyp "MB_TENSION_CONTROL_MATERIAL_TYPE" enthalten.
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0009	Der Eingang "MaterialParameter.WebWidth" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0010	Der Eingang "MaterialParameter.WebThickness" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0011	Der Eingang "MaterialParameter.Material-EModul" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0012	Der Eingang "MaterialParameter.WebGrammage" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0014	Der Eingang "Circumference" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0015	Der Eingang "DistanceRollToRoll" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0016	Der Eingang "LowValMeasure" hat einen größeren oder gleichen Wert wie der Eingang "HighValMeasure"
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0017	Der Eingang "LowValMeasure" ist kleiner 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0018	Der Eingang "ExtendedSettings.NumMeasureData" ist 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0019	Der Eingang "ExtendedSettings.AccuracyFactor" ist kleiner als 1 oder größer als 100
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0020	Der Eingang "ExtendedSettings.TimeFactor" ist kleiner als 10 oder größer als 100
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0021	Der Eingang "ExtendedSettings.ExcitationFactor" ist kleiner als 10 oder größer als 1000
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0022	Der Eingang "ExtendedSettings.EModuleSafetyFactor" ist kleiner als 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0023	Der Eingang "ExtendedSettings.ErrorFactor" ist kleiner als 10 oder größer als 100
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0024	Der Eingang Feedback ist entweder kleiner als "LowValMeasure" oder größer als "HighValMeasure"

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0025	Der Eingang Feedback hat eine absolute Grenze "LowValMeasure" oder "HighValMeasure" einschließlich ExtendedSettings.ErrorFactor über- oder unterschritten
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Zustandsmaschine des Funktionsbausteins befindet sich in einen ungültigen Status

Tab. 7-5: Fehlercodes Funktionsbaustein MB_TensionIdentModelParam-Type01

7.3 MB_TENSION_IDENT_EXTENDED_SETTINGS

In der folgenden Tabelle sind die Elemente der Struktur MB_TENSION_IDENT_EXTENDED_SETTINGS aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Beschreibung
NumMeasureData	USINT	1	n.def	3	Anzahl Iterationen der abschließenden Mittelwertbildung [-].
AccuracyFactor	REAL	1.0	100.0	20.0	Die abschließenden Messwerte dürfen maximal in diesem prozentualen Wertebereich variieren [%].
TimeFactor	USINT	10	100	25	Relation zwischen Anregungsfrequenz und Streckendynamik [-].
ExcitationFactor	UINT	10	1000	100	Maximale Erhöhung der Anregungsamplitude pro Iterationszyklus [%].
EModuleSaftyFactor	UINT	0	n.def	100	Prozentuale Reduktion der initialen Anregung infolge des vorgegebenen E-Moduls [%].
ErrorFactor	UINT	10	100	50	Zugkraftbereich welcher ergänzend zu den Grenzen "HighValMeasure" und "LowValMeasure" maximal auftreten darf [%].

Tab. 7-6: Strukturbeschreibung MB_TENSION_IDENT_EXTENDED_SETTINGS

7.4 MB_TensionControlLoadCellType02

Kurzbeschreibung

Der Multiachs-Zugkraftregler regelt, abhängig vom Messwert einer Kraftmessdose, die Zugkraft von bis zu 8 Achsen auf den gewünschten Sollwert.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechTensionAdvanced

Tab. 7-7: Referenztable MB_TensionControlLoadCellType02



Die Verwendung des Funktionsbausteines unterliegt einer Lizenzpflicht (Details siehe [Einführung und Übersicht, Seite 439](#)).



Ab MLC 12VRS werden zusätzlich auch Slave-Achsen mit Interpolation auf der Steuerung unterstützt.



Der Eingang "ResponseCurve" ist in der Bibliothek RIL_LoopControl deklariert. Daher muss die Bibliothek RIL_LoopControl zusätzlich im Bibliotheksverwalter aufgenommen werden, wenn der Eingang "ResponseCurve" genutzt wird.

Schnittstellenbeschreibung

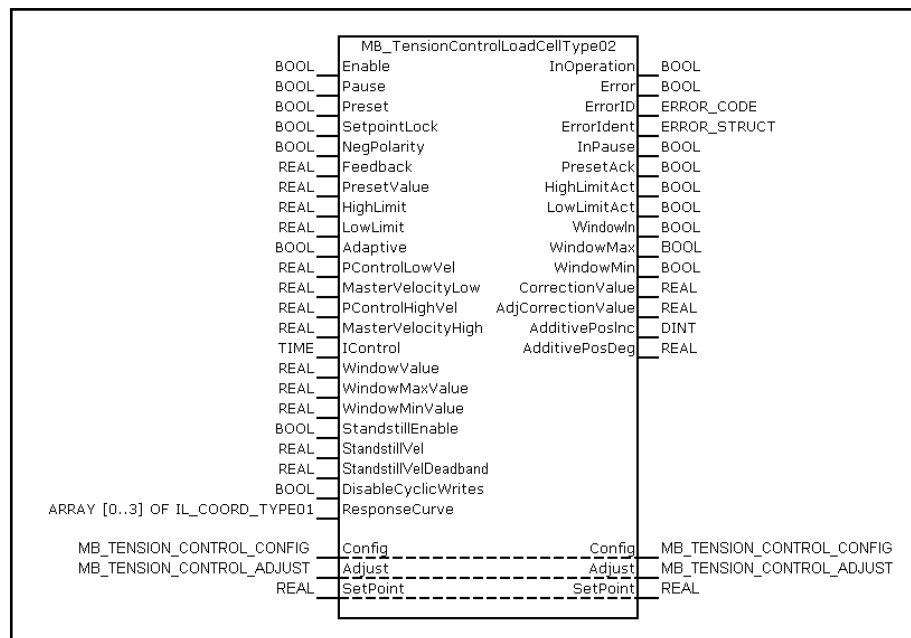


Abb. 7-5: Funktionsbaustein MB_TensionControlLoadCellType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Wird dieser Eingang gesetzt, arbeitet der Funktionsbaustein
	Pause	BOOL	Bei gesetztem Eingang wird der Zugkraftregler angehalten und der I-Anteil des Ausgangswertes eingefroren. Der Ausgang "InPause" signalisiert ein aktives "Pause"
	Preset	BOOL	Bei der steigenden Flanke dieses Eingangs wird der Reglerausgang auf den in PresetValue parametrierbaren Wert gesetzt und bei Erfolg der Ausgang "PresetAck" gesetzt
	SetpointLock	BOOL	Bei steigender Flanke am Eingang wird der aktuelle Zugkraftistwert an "Feedback" als neuer Sollwert für den Zugkraftregler übernommen
	NegPolarity	BOOL	Bei gesetztem Eingang wird die Stellgröße "CorrectionValue" und "AdditivePosInc"/"AdditivePosDeg" mit einem negativen Vorzeichen versehen
	Feedback	REAL	aktueller Istwert der Zugkraft
	PresetValue	REAL	Wert auf den der Reglerausgang gesetzt wird, wenn der Eingang "Preset" gesetzt ist

E/A-Typ	Name	Datentyp	Beschreibung
	HighLimit	REAL	Maximalwert den der Reglerausgang annehmen kann. Wird die vom Regler berechnete Stellgröße größer als dieser Wert, wird die Stellgröße begrenzt und der I-Anteil des Reglers nicht weiter aufsummiert
	LowLimit	REAL	Minimalwert den der Reglerausgang annehmen kann. Wird die vom Regler berechnete Stellgröße kleiner als dieser Wert, wird die Stellgröße begrenzt und der I-Anteil des Reglers nicht weiter aufsummiert
	Adaptive	BOOL	Bei gesetztem Eingang wird eine adaptive P-Verstärkung für die Regelung verwendet
	PControlLowVel	REAL	P-Verstärkung des Reglers wenn "Adaptive" = FALSE. Ansonsten entspricht der Wert der P-Verstärkung bei niedrigen Leitachsgeschwindigkeiten
	MasterVelocityLow	REAL	Nur wirksam wenn "Adaptive" = TRUE. Leitachsgeschwindigkeit bis zu der "PControlLowVel" verwendet wird
	PControlHighVel	REAL	P-Verstärkung des Reglers bei hohen Leitachsgeschwindigkeiten. Wenn "Adaptive" = FALSE hat dieser Eingang keine Wirkung
	MasterVelocityHigh	REAL	Nur wirksam wenn "Adaptive" = TRUE. Leitachsgeschwindigkeit ab der "PControlHighVel" verwendet wird
	IControl	TIME	Nachstellzeit des I-Reglers. Wird der Wert auf 0 gesetzt, so wird der I-Regler unwirksam
	WindowValue	REAL	Prozessgrößenfenster in [%]
	WindowMaxValue	REAL	Maximalwert der Prozessgröße in [%]
	WindowMinValue	REAL	Minimalwert der Prozessgröße in [%]
	StandstillEnable	BOOL	StandstillEnable = FALSE: Zweig "InVelocity Control" ist aktiv, Stellgröße: Getriebefeinabgleich StandstillEnable = TRUE: Zweig "Regelung im Stillstand" ist aktiv, Stellgröße: Leitachspannung additiv Die in "StandstillVel" angegebenen additive Geschwindigkeit ist wirksam
	StandstillVel	REAL	Additive Geschwindigkeit für die Regelung im Stillstand
	StandstillVelDeadband	REAL	Dieser Wert bestimmt den Wert der Regelabweichung unterhalb dessen die Spanngeschwindigkeit bei der Regelung im Stillstand reduziert wird
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben. Der Wert wird nur bei steigender Flanke von "Enable" ausgewertet

E/A-Typ	Name	Datentyp	Beschreibung
	ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	Kennlinie zur Gewichtung der Regelabweichung im Zweig "InVelocity Control" IL_COORD_TYPE01, RIL_LoopControl.library
VAR_OUTPUT	InOperation	BOOL	Der Zugkraftregler arbeitet und die Ausgänge sind gültig
	Error	BOOL	Signalisiert einen Fehler. Mit "Enable" = FALSE wird der Fehler gelöscht
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung gemäß Fehlertabelle (siehe Fehlercodes , Seite 462)
	InPause	BOOL	"Pause" ist aktiv
	PresetAck	BOOL	Ist dieser Ausgang aktiv, wurde der "Preset" Eingang ausgewertet und der Ausgang "CorrectionValue" hat den Wert "PresetValue" angenommen
	HighLimitAct	BOOL	Dieser Ausgang wird gesetzt, wenn die vom Regler berechnete Stellgröße größer als der in "HighLimit" angegebenen Wert ist
	LowLimitAct	BOOL	Dieser Ausgang wird gesetzt, wenn die vom Regler berechnete Stellgröße kleiner als der in "LowLimit" angegebenen Wert ist
	WindowIn	BOOL	Wenn TRUE, befindet sich die Prozessgröße innerhalb des parametrisierten Fensters
	WindowMax	BOOL	Wenn TRUE befindet sich die Prozessgröße oberhalb des parametrisierten Maximalwertes
	WindowMin	BOOL	Wenn TRUE, befindet sich die Prozessgröße unterhalb des parametrisierten Minimalwertes
	CorrectionValue	REAL	Stellgröße des Zugkraftreglers
	AdjCorrectionValue	REAL	Stellgröße des Zugkraftreglers mit dynamischer Vorsteuerung
	AdditivePosInc	DINT	Additive Leitachsposition in Inkrementen je sercos Zyklus, genutzt für reale Slave-Achse mit Interpolation auf dem Antrieb (Zielparameter: P-0-0692)
	AdditivePosDeg	REAL	Ab Version 12VRS: Additive Leitachsposition in Grad je sercos Zyklus, genutzt für Slave-Achse mit Interpolation auf der Steuerung (z.B. für virtuelle Achsen oder reale Achse mit Interpolation auf der Steuerung, Zielparameter: A-0-2600)
VAR_IN_OUT	Config	MB_TENSION_CONTROL_CONFIG	Konfiguration des Zugkraftreglers (Achsen, Zugkraftverteilung) MB_TENSION_CONTROL_CONFIG , Seite 478

E/A-Typ	Name	Datentyp	Beschreibung
	Adjust	MB_TENSION_CONTROL_ADJUST	Struktur für die Folgeachseinstellungen im laufenden Betrieb MB_TENSION_CONTROL_ADJUST , Seite 478
	Setpoint	REAL	Sollwert der Zugkraft Diese wird in der Regel durch den Anwender gesetzt, kann aber auch vom Funktionsbaustein selbst geändert werden (SetpointLock , Seite 458)

Tab. 7-8: Schnittstellenvariablen Funktionsbaustein MB_TensionControlLoadCellType02

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
Pause	BOOL			FALSE	kontinuierlich
Preset	BOOL			FALSE	kontinuierlich
SetPointLock	BOOL			FALSE	kontinuierlich
NegPolarity	BOOL			FALSE	kontinuierlich
Feedback	REAL	n.def.	n.def.	0.0	kontinuierlich
PresetValue	REAL	n.def.	n.def.	0.0	steigende Flanke an "Preset"
HighLimit	REAL	>LowLimit	n.def.	0.00001	kontinuierlich
LowLimit	REAL	-95.0	<HighLimit	-0.00001	kontinuierlich
Adaptive	BOOL			FALSE	kontinuierlich
PControlLowVel	REAL	>0.0	n.def.	0.0001	kontinuierlich
MasterVelocity-Low	REAL	>0.0	<=MasterVelocity-High	0.0	kontinuierlich
PControlHighVel	REAL	>0.0	n.def.	0.0	kontinuierlich
MasterVelocity-High	REAL	>=MasterVelocity-Low	n.def.	1.0	kontinuierlich
IControl	TIME	0s	n.def.	20ms	kontinuierlich
WindowValue	REAL	0.0	100.0	0.0	kontinuierlich
WindowMaxValue	REAL	>=WindowMinValue	n.def.	0.0	kontinuierlich
WindowMinValue	REAL	0.0	<=WindowMaxValue	0.0	kontinuierlich
StandstillEnable	BOOL			FALSE	kontinuierlich
StandstillVel	REAL	0.0	n.def.	0.0	kontinuierlich
StandstillDeadband	REAL	0.0	n.def.	1.0	kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
DisableCyclicWrites	BOOL			FALSE	steigende Flanke an "Enable"
ResponseCurve	ARRAY [0..3] OF IL_COORD_TY-PE01	siehe RIL_Loop-Control	siehe RIL_Loop-Control	siehe Datenstruktur	steigende Flanke an "Enable"
Config	MB_TENSION_CONTROL_CONFIG	siehe Datenstruktur	siehe Datenstruktur	siehe Datenstruktur	steigende Flanke an "Enable"
Adjust	MB_TENSION_CONTROL_ADJUST	siehe Datenstruktur	siehe Datenstruktur	siehe Datenstruktur	kontinuierlich
Setpoint	REAL	n.def.	n.def.	0.0	kontinuierlich

Tab. 7-9: Min- Max- und Default-Werte Funktionsbaustein MB_TensionControlLoadCellType02

Funktionsbeschreibung

Wie in der nachfolgenden Abbildung dargestellt, werden mehrere Zugwalzen durch den Funktionsbaustein MB_TensionControlLoadCellType02 gesteuert. Die Messwerte der Kraftmessdose werden auf den "Feedback"-Eingang des Funktionsbausteins gelegt. Die Sollzugkraft wird über den Eingang "SetPoint" eingestellt. Im normalen Betrieb (Zweig "InVelocity Control" ist aktiv) wird die Zugkraft durch den Getriebefeinabgleich der Zugwalzen geregelt. Je nach Achstyp wird hierbei der entsprechende Parameter beschrieben (P-0-0694 bei Interpolation auf dem Antrieb bzw. A-0-2605 bei Interpolation auf der Steuerung). Bei stehender Leitachse werden die Zugwalzen über die additive Leitachsisposition bewegt, da in diesem Fall der Getriebefeinabgleich keinen Einfluss auf die Zugkraft hat. Je nach Achstyp wird hierbei der entsprechende Parameter beschrieben (P-0-0692 bei Interpolation auf dem Antrieb bzw. A-0-2600 bei Interpolation auf der Steuerung).

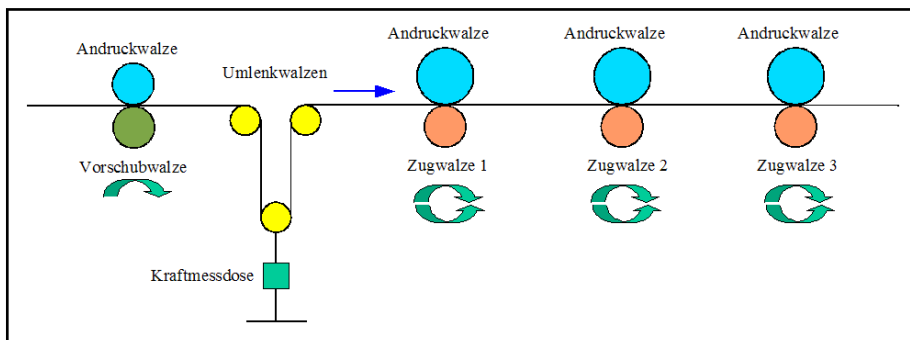


Abb. 7-6: Multiachs-Zugkraftregler

Im Folgenden werden die einzelnen Funktionen der Eingänge des Funktionsbausteins näher erläutert. Allgemein gilt für alle Eingänge, dass sie kontinuierlich bei gesetztem "Enable"-Eingang ausgewertet werden. Das heißt, Änderungen der Eingänge werden sofort ohne erneutes Toggeln des "Enable"-Eingangs wirksam.

Ausnahme dieser Regel bildet der "PresetValue"-Eingang, der nur bei steigender Flanke am "Preset"-Eingang übernommen wird. Ebenso wird der Eingang "DisableCyclicWrites" nur bei steigender Flanke am "Enable"-Eingang übernommen.

Die gesamte Wirkungsweise des Zugkraftreglers ist in den folgenden Flussdiagrammen dargestellt. Je nach Eingang "StandstillEnable" ist entweder der Pfad "InVelocity" oder "Regelung im Stillstand" (Standstill Control) wirksam.

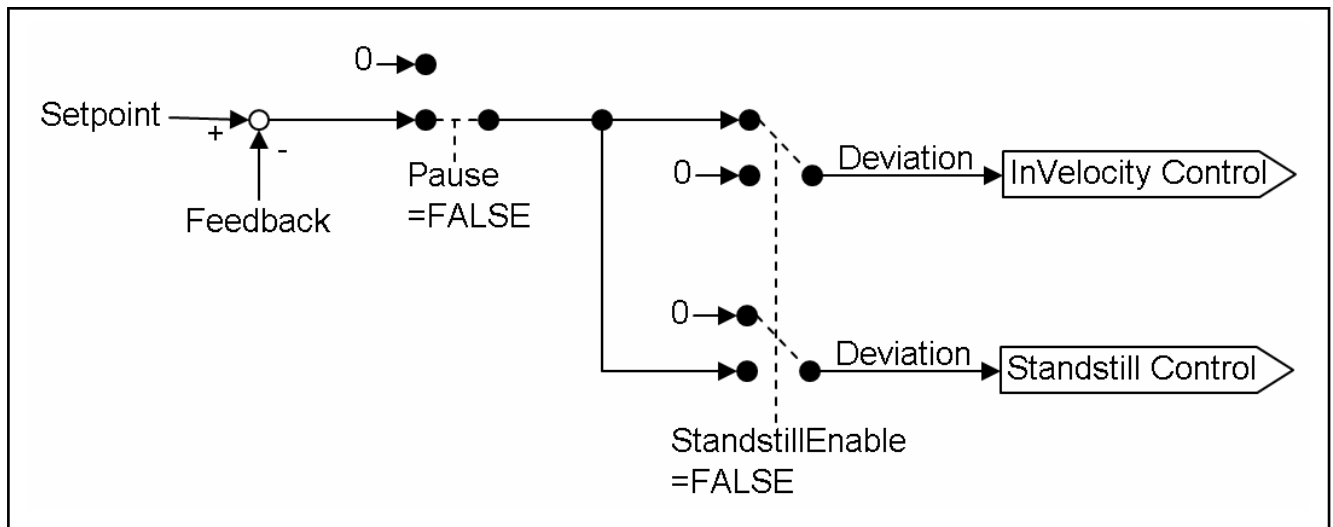


Abb. 7-7: Flussdiagramm des Zugkraftreglers mit Abzweig für Regelung im Stillstand / In Geschwindigkeit

Der Pfad "InVelocity Control" ist wirksam, wenn der Eingang "Standstill-Enable" nicht gesetzt ist. Die Zugkraft wird hierbei über den Parameter "Getriebefineinabgleich" P-0-0694 (Slave-Achse mit Interpolation auf dem Antrieb / IPO Drive) bzw. A-0-2605 (Slave-Achse mit Interpolation auf der Steuerung/IPO Control) geregelt. Es wird direkt auf die Parameter geschrieben, wenn der Eingang "DisableCyclicWrites" nicht gesetzt ist. Bei realen Achsen mit Interpolation auf dem Antrieb ist daher auch die Konfiguration des Parameters P-0-0694 in den optionalen zyklischen sercos Daten (MDT) erforderlich.

Die Stellgröße wird zusätzlich auch über den Ausgang "CorrectionValue" ausgegeben.

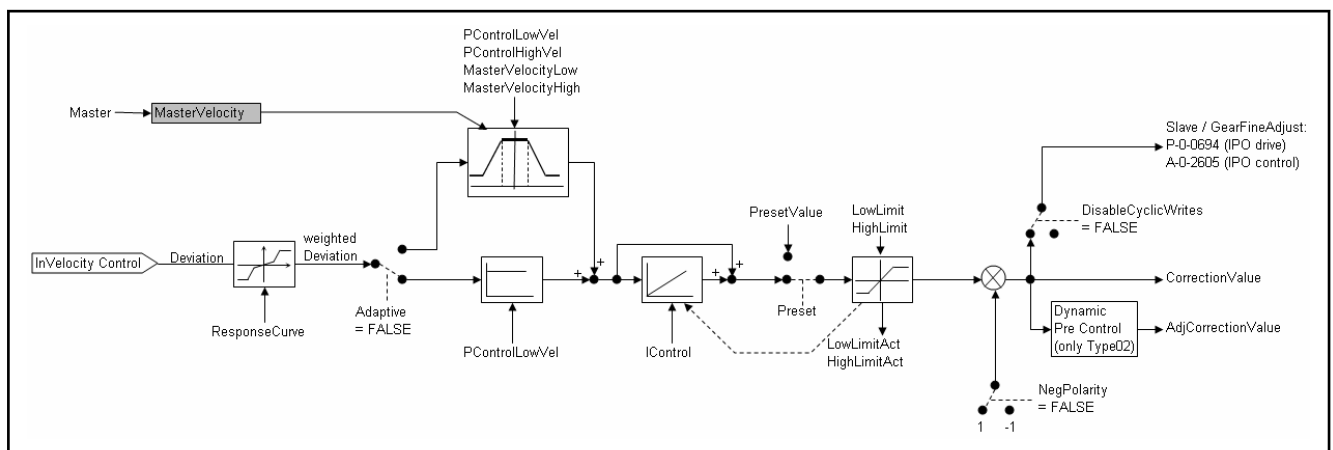


Abb. 7-8: Flussdiagramm des Zugkraftreglers für den Zweig "InVelocity Control"

Der Pfad "Regelung bei Stillstand der Leitachse" (Standstill Control) ist wirksam, wenn der Eingang "StandstillEnable" gesetzt ist. Dieser Pfad wirkt auf die additive Leitachseposition der Slave-Achse, um damit eine Geschwindigkeit vorzugeben. Es wird direkt auf die Parameter geschrieben, wenn der Eingang "DisableCyclicWrites" nicht gesetzt ist. Bei Slave-Achsen mit Interpolation auf dem Antrieb wird als additive Leitachseposition der Zielparameter P-0-0692 beschrieben. Bei Slave-Achsen mit Interpolation auf der Steuerung wird als additive Leitachseposition der Zielparameter A-0-2600 beschrieben. Bei realen Achsen mit Interpolation auf dem Antrieb ist daher auch die Konfi-

guration des Parameters P-0-0692 in den optionalen zyklischen sercos Daten (MDT) erforderlich.

Gleichzeitig steht die additive Leitachsposition immer über den Ausgang "AdditivePosInc" (bei Interpolation auf dem Antrieb) oder "AdditivePosDeg" (bei Interpolation auf der Steuerung) zur Verfügung.

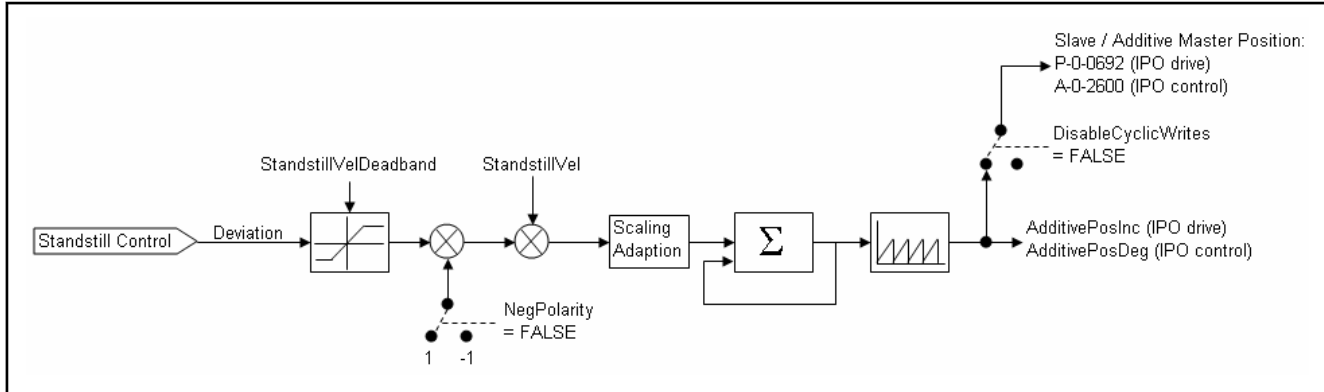


Abb. 7-9: Flussdiagramm des Zugkraftreglers für die Regelung im Stillstand (Standstill Control)

Das Flussdiagramm enthält aus Gründen der Übersichtlichkeit nicht die Begrenzungen "WindowValue", "WindowMaxValue", "WindowMinValue" sowie deren Statusanzeigen.

Inbetriebnahme

Ist "DisableCyclicWrites" inaktiv, müssen die beiden Parameter P-0-0694 (Getriebefeinabgleich) und P-0-0692 (additive Leitachsposition) im zyklischen Sollwertkanal der Folgeachsen aufgelegt werden, wenn reale Slave-Achsen (Config.Slave[1..8]) mit Interpolation auf dem Antrieb verwendet wird. Bei steigender Flanke an "Enable" werden zyklische Datenkanäle überprüft. Wird einer der beiden Parameter nicht gefunden, erfolgt eine entsprechende Fehlermeldung an den Fehlerausgängen des Funktionsbausteins. Ist "DisableCyclicWrites" aktiv, wird die Konfiguration der zyklischen Sollwertdatenkanäle nicht überprüft.

Bei Slave-Achsen mit Interpolation auf der Steuerung (z.B. virtuelle Achse) ist die Konfiguration der zyklischen Datenkanäle nicht notwendig.



- Die verwendeten Slave-Achsen müssen vor dem Aktivieren des "Enable"-Eingangs in eine synchrone Betriebsart gebracht werden. Dies kann durch das AchsInterface oder die PLCopen-Funktionsbausteine "GearIn" oder "GearInPos" geschehen. Ausnahme: Die synchrone Betriebsart ist nicht erforderlich, wenn der "Pause" Eingang gesetzt ist
- Die Master Achse muss Modulo 360 Grad gewichtet sein
- Es ist möglich die Slave-Achsen gemischt zu konfigurieren (z.B. Slave-Achse 1 mit Interpolation auf der Steuerung, Slave-Achse 2 mit Interpolation im Antrieb). **Bei einer gemischten Konfiguration muss der Parameter P-0-0750 jeder realen Slave-Achse mit Interpolation auf dem Antrieb auf dem Defaultwert = 1 belassen werden**

Config

Über die Struktur "MB_TENSION_CONTROL_CONFIG" werden statische Konfigurationsparameter wie die Achskonfiguration und die Geometrie festgelegt. Der Eintrag "Master" enthält die Achsreferenz auf die Leitachse. Alle Zugwalzen werden im Arrayeintrag "Slave" angegeben. Eine nicht vorhandene Achse wird durch die Konstante "NoAxis" repräsentiert. Für die dynamische Vorsteuerung wird der Umfang der ersten Zugwalze im Struktureintrag

"Circumference" sowie die Bahnlänge zwischen Vorschubwalze und erster Zugwalze im Struktureintrag "DistanceRoll1To2" benötigt.

Adjust

Die Struktur "MB_TENSION_CONTROL_ADJUST" enthält zur Laufzeit veränderbare Einstellungen. Im Array "SlavePreCtrl" wird die Art der Zugwalzenansteuerung vorgegeben. Wird für eine Achse die Konstante "NO_PRE_CTRL" eingetragen, erfolgt keine Regelung der Achse. Der Getriebefeinabgleich (P-0-0694 bzw. A-0-2605) wird fest auf "0" eingestellt.

Die Konstante "PRE_CTRL_100" bewirkt, dass der Wert des Ausgangs "CorrectionValue" in den Getriebefeinabgleich geschrieben wird.

In der Einstellung "PRE_CTRL_DYNAMIC" wird der Ausgang "AdjCorrection-Value" auf den Getriebefeinabgleich der Achse geschrieben. Die dynamische Zugwalzensteuerung wird für alle Walzen hinter der ersten Zugwalze verwendet. Hierfür muss die Konstante "PRE_CTRL_DYNAMIC" in "SlavePreCtrl" der nachfolgenden Zugwalzen eingetragen werden. Die erste Zugwalze muss auf "PRE_CTRL_100" stehen. Diese Einstellung bewirkt, dass die Zugkraft der Folgezugwalzen bei Geschwindigkeitsänderung nicht beeinflusst wird.

Die Einstellungen in "SlavePreCtrl" haben keinen Einfluss auf die ausgeführte Bewegung im Standstill-Modus.

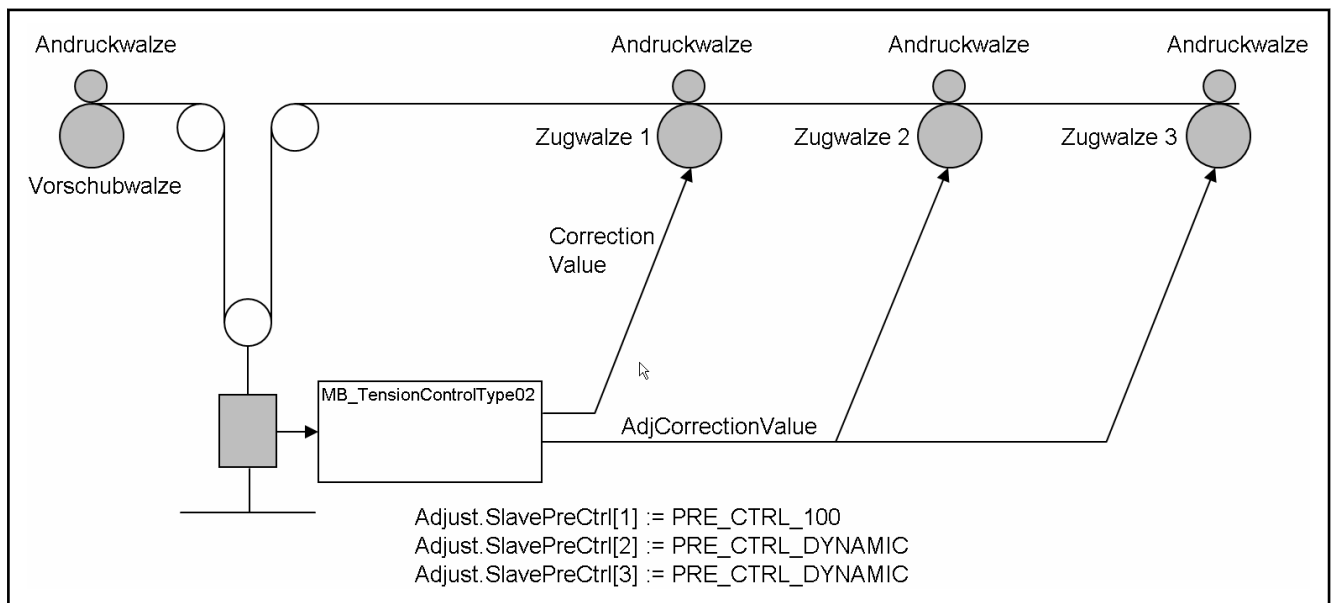


Abb. 7-10: MB TensionControlType02 Beispielkaskade mit 3 Zugwalzen

Preset/PresetValid

Durch eine positive Flanke am Eingang "Preset" wird ein Anfangswert über die Variable "PresetValue" für die Stellgröße vorgegeben. Gleichzeitig werden interne Variablen des Reglers so gesetzt, dass ein ruckfreies Anfahren des Reglers erfolgt. Der "Preset" wirkt nur auf den Getriebefeinabgleich im Pfad "InVelocity Control".

Der "Preset" Eingang wird nur ausgewertet, wenn "Enable" gesetzt ist. Bei steigender Flanke an "Preset" wird der neue Wert an "PresetValue" übernommen und "PresetAck" gesetzt. Eine fallende Flanke an "Preset" oder "Enable" löscht "PresetAck". Der "Preset"-Eingang ist dominant über "Pause", d.h. der Wert an "PresetValue" erscheint am Ausgang, auch wenn "Pause" gesetzt ist. Ist bei steigender Flanke an "Enable" "Preset" gesetzt, so wird "PresetValue" sofort als Stellgröße übernommen.

	 Die Preset Funktion sollte im Stillstand ausgeführt werden (StandstillEnable=TRUE), da ansonsten Geschwindigkeitssprünge durch das Setzen des ControlValue auf den PresetValue entstehen können. Der Betrag der Geschwindigkeitssprünge ist proportional zu der Geschwindigkeit. Der "Preset"-Eingang wirkt nur bei der steigenden Flanke.
DisableCyclicWrites	Der Eingang "DisableCyclicWrites" bestimmt, ob die vom Funktionsbaustein berechneten Werte direkt über den zyklischen Kanal geschrieben werden (FALSE) oder nur am Bausteinausgang verfügbar sind (TRUE). Damit ist es möglich, mehrere Bausteine zu kaskadieren, indem man "DisableCyclicWrites" auf TRUE setzt. Dann kann man lediglich die Ausgänge ("CorrectionValue" und "AdditivePosInc" bzw. "AdditivePosDeg") verwenden und weiter modifizieren. Allerdings ist es dann noch notwendig, dass der Anwender die Werte selbst in den Zielparameter schreibt. Ist "DisableCyclicWrites" auf TRUE gesetzt, wird außerdem die Prüfung, ob die optional zyklischen Parameter auch konfiguriert sind, nicht durchgeführt.
ResponseCurve	Mit Hilfe des Eingang "ResponseCurve" kann die Regelabweichung gewichtet werden. Durch diese Funktionalität kann beispielsweise bei kleinen Regelabweichungen eine geringere Gewichtung der Regelabweichung eingestellt werden. Dies ergibt quasi eine geringere Proportionalverstärkung und dadurch ein ruhigeres Regelverhalten im eingeschwungenen Zustand. Bei der Parametrierung des Eingangs sollten die Reglerparameter zunächst ohne Bewertung eingestellt werden und erst danach eine geringere Gewichtung vorgegeben werden. Hierdurch wird erreicht, dass die Reglerparameter zunächst global anhand der Systemdynamik eingestellt werden und anschließend eine lokale Optimierung (z. B. für den eingeschwungenen Zustand) durchgeführt wird. Parametriert wird diese lokale Optimierung mittels einer Kennlinie mit bis zu 4 Stützstellen. Negative Abweichungen werden durch Punktspiegelung der Kurve am Ursprung behandelt. Daraus ergeben sich insgesamt neun lineare Sektionen. Die Sektion s5 verläuft durch den Ursprung. Die Sektionen s1 und s9 haben einen Anstieg von 1 und laufen jeweils ins positive oder negative Unendliche. Wird der Eingang unbeschaltet gelassen, wird als Default-Wert eine 1-zu-1 Kennlinie verwendet, d. h. es wird keinerlei Gewichtung durchgeführt.

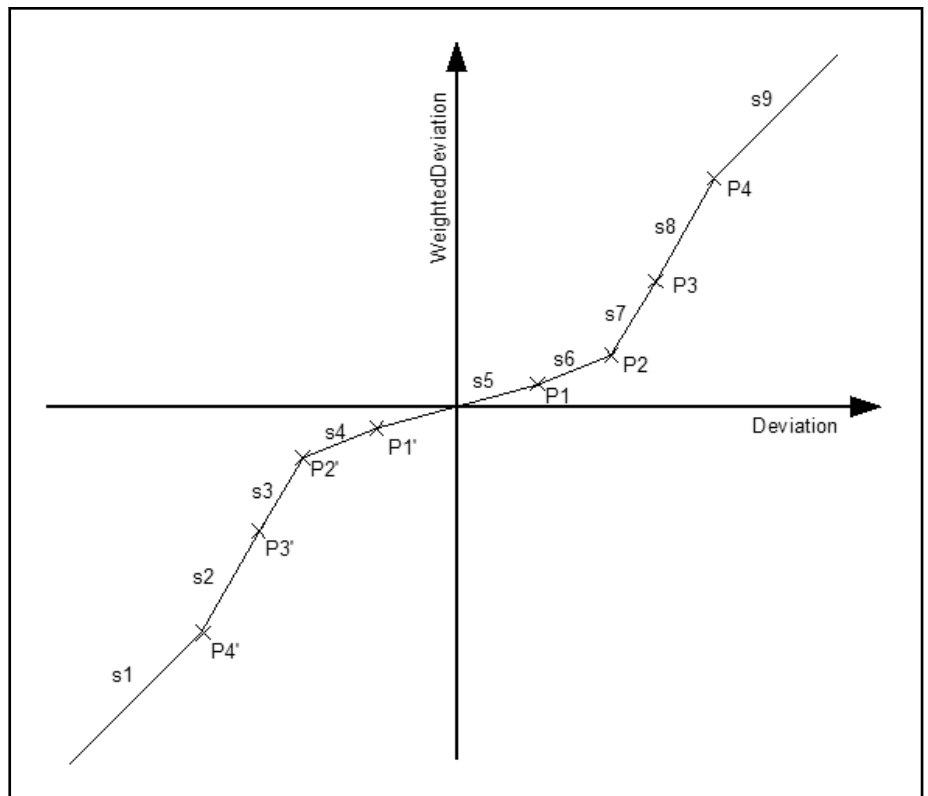


Abb. 7-11: Wichtung der Regelabweichung mit dem Eingang ResponseCurve

Die Punkte P1 bis P4 werden in einem Array aus IL_COORD_TYPE01 abgelegt. Werden weniger als vier Punkte benötigt, muss die X-Koordinate der nicht benötigten Punkte (vom letzten Punkt beginnend) auf 0 gesetzt werden. Die X-Werte müssen positiv sein und in positiver Richtung aufsteigen. Es ist erlaubt, denselben Punkt zweimal abzulegen.

Adaptive, PControlLowVel, PControlHighVel, MasterVelocityLow, MasterVelocityHigh

Die P-Verstärkung kann wahlweise als konstante P-Verstärkung oder als adaptive P-Verstärkung vorgegeben werden. Über den Eingang "Adaptive" wird zwischen adaptiver oder konstanter P-Verstärkung gewählt. Durch Setzen von "Adaptive" auf TRUE wird die adaptive P-Verstärkung ausgewählt.

Ist eine konstante P-Verstärkung gewählt, ist nur der Wert in "PControlLowVel" für die P-Verstärkung wirksam.

Wird die adaptive P-Verstärkung gewählt, müssen die Eingänge "PControlLowVel", "PControlHighVel", "MasterVelocityLow" und "MasterVelocityHigh" beschaltet werden. Die Kennlinie sieht dann folgendermaßen aus:

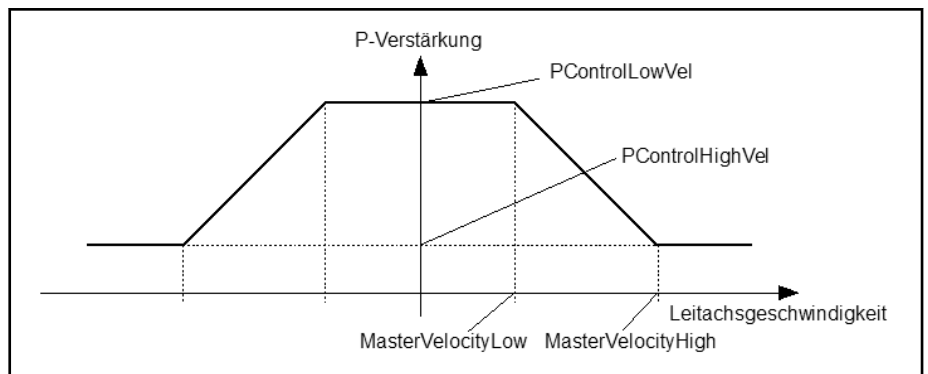


Abb. 7-12: Kennlinie der adaptiven P-Verstärkung

Die einzelnen Eingänge beschreiben die Kennlinie für positive Leitachsgeschwindigkeiten. Die Kennlinie für negative Leitachsgeschwindigkeiten ergibt sich aus der Spiegelung an der Achse der P-Verstärkung. D.h. es sind nur symmetrische Verläufe möglich. Die P-Verstärkung zwischen den Leitachsgeschwindigkeiten "MasterVelocityLow" und "MasterVelocityHigh" wird linear interpoliert. Dadurch ergeben sich einige Bedingungen für die Wertebereiche der Eingaben:

- Die Werte für "MasterVelocityLow" und "MasterVelocityHigh" müssen positiv sein
- Die Werte für "MasterVelocityLow" und "MasterVelocityHigh" dürfen nicht gleich sein
- Der Wert von "MasterVelocityLow" muss kleiner als der Wert von "MasterVelocityHigh" sein

Ist eine dieser Bedingungen verletzt, wird ein Fehler am Ausgang des Bausteines generiert.

SetpointLock Durch Setzen des Eingangs "SetpointLock" wird der aktuelle Zugkraftistwert als neuer Sollwert für den Zugkraftregler übernommen. Die Funktion erlaubt es, z.B. für den Sollwert der Zugkraft einen sinnvollen Startwert zu erhalten, der den momentanen Zustand der Maschine widerspiegelt.

Pause Der Eingang "Pause" ermöglicht das Anhalten des Zugkraftreglers. Wird der Eingang auf TRUE gesetzt, wird die Regelabweichung auf "0" gesetzt, d.h. der Reglerausgang wird auf dem aktuellen I-Anteil eingefroren. Ist "Pause" aktiv, wird der Ausgang "InPause" gesetzt. Bei aktiver Pause ist die Synchronbetriebsart nicht erforderlich, und es kann die Leistung (MC_Power) abgeschaltet werden.



Wenn keine Zugkraftregelung im Stillstand genutzt wird (StandstillEnable ist immer FALSE), muss im Maschinenstillstand der Eingang "Pause" gesetzt werden.

NegPolarity Der Eingang "NegPolarity" beeinflusst den Regelsinn. Abhängig von der Lage der Zugwalze, bezüglich der Durchlaufrichtung des Materials und der Lage der Vorschubwalze, muss der Regelsinn entweder positiv oder negativ sein. Ist dieser Eingang FALSE, wird der Regelsinn positiv und eine positive Regelabweichung führt zu einer positiven Stellbewegung. Der Eingang wirkt auch auf die Stillstandsregelung.

Feedback Dieser Eingang stellt den Istwert der Zugkraft dar. Dieser kann z.B. durch einen Analogeingang am Antrieb oder den Inline I/O eingelesen werden. Der Anwender ist dabei für eine sinnvolle Skalierung verantwortlich.

PresetValue Durch Setzen des Eingangs "Preset" wird ein Anfangswert über die Variable "PresetValue" für die Stellgröße vorgegeben. Gleichzeitig werden interne Variablen des Reglers so gesetzt, dass ein ruckfreies Anfahren des Reglers erfolgt.

HighLimit, HighLimitAct Mit diesem Eingang wird die Stellgröße des Reglers auf einen Maximalwert begrenzt. Tritt die Begrenzung ein, wird der I-Anteil des Reglers nicht weiter aufsummiert ("Anti-reset-windup"). Der maximale Wert wird durch den im Antrieb definierten Maximalwert für den Getriebefeinabgleich bestimmt. Tritt eine Begrenzung ein, wird der Ausgang "HighLimitAct" gesetzt.

LowLimit, LowLimitAct Mit diesem Eingang wird die Stellgröße des Reglers auf einen Minimalwert begrenzt. Tritt die Begrenzung ein, wird der I-Anteil des Reglers nicht weiter aufsummiert ("Anti-reset-windup"). Der minimale Wert liegt bei -95 unabhängig von ggf. kleineren Werten im Antrieb (siehe dazu auch den Abschnitt "Getriebefeinabgleich"). Tritt eine Begrenzung ein, wird der Ausgang "LowLimitAct" gesetzt.

WindowValue, WindowIn

Mit dem Wert "WindowValue" kann der Anwender ein symmetrisches Fenster definieren innerhalb dessen sich der aktuelle Zugkraftistwert befinden soll. Das Fenster ist wie folgt definiert:

$$\left(\text{SetPoint} - |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right) \leq \text{Feedback} \leq \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right)$$

Abb. 7-13: WindowValue, WindowIn

Ist diese Bedingung erfüllt, wird der Ausgang "WindowIn" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt. Eine Begrenzung wird damit nicht durchgeführt.

WindowMaxValue, WindowMax

Mit diesem Wert kann der Anwender einen oberen Wert für den Zugkraftistwert festlegen. Der Wert wird wie folgt ausgewertet:

$$\text{Feedback} > \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowMaxValue} - 100}{100} \right)$$

Abb. 7-14: WindowMaxValue, WindowMax

Ist diese Bedingung erfüllt, wird der Ausgang "WindowMax" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt. Eine Begrenzung wird damit nicht durchgeführt.

WindowMinValue, WindowMin

Mit diesem Wert kann der Anwender einen unteren Wert für den Zugkraftistwert festlegen. Der Wert wird wie folgt ausgewertet:

$$\text{Feedback} < \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowMinValue} - 100}{100} \right)$$

Abb. 7-15: WindowMinValue, WindowMin

Ist diese Bedingung erfüllt, wird der Ausgang "WindowMin" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt. Eine Begrenzung wird damit nicht durchgeführt.

StandstillEnable

Durch Setzen des Eingangs "StandstillEnable" wird die in "StandstillVel" angegebene Geschwindigkeit additiv auf die zu regelnden Achsen aufgeschaltet. Damit ist eine Zugkraftregelung auch im Stillstand möglich.



Der Getriebefeinabgleich verliert bei niedrigen Geschwindigkeiten immer mehr an Einfluss. Bei Stillstand der Leitachse besteht sogar ein offener Regelkreis, da der Getriebefeinabgleich dann keinen Einfluss mehr hat. Daher muss der Eingang "Standstill-Enable" bei Erreichen des Stillstandsfensters gesetzt werden, damit im Pfad "InVelocity Control" der I-Anteil durch den offenen Regelkreis nicht bis zur Grenze ansteigt.



Bei Achsen mit Interpolation auf der Steuerung wird der Ausgang "AdditivePosInc" im Moduloformat 2²⁰ ausgegeben (ab Version 12).



Wird dieser Eingang verwendet, muss sichergestellt sein, dass der Zugkraftregler nur in der sercos synchronen Task verwendet wird. Ist dies nicht der Fall, kann es zu unruhigem Lauf des Motors, oder Oszillieren um die Stillstands-lage führen.

StandstillVel Dieser Wert gibt die additive Geschwindigkeit an, die beim Setzen des Eingangs "StandstillEnable" an den Antrieb gesendet wird. Die Einheiten der Geschwindigkeit entsprechen dabei der im Antrieb eingestellten Wichtung.



Im Stillstand wird die Geschwindigkeit sprunghaft erzeugt. Da die Geschwindigkeit innerhalb eines sercos Zyklus erreicht werden muss, können keine beliebig hohen Werte vorgegeben werden. Wird der Wert der Stillstandsgeschwindigkeit zu hoch gewählt, kann dies zur Oszillation des Antriebes führen. Der Maximalwert richtet sich dabei nach der Wichtung der Geschwindigkeit, sowie der eingestellten maximalen Beschleunigung und angeschlossenen Masse am Antrieb. Um zu große Geschwindigkeitssprünge zu vermeiden, kann eine Filterzeitkonstante im Antriebsparameter P-0-0693 definiert werden, wodurch der Sprung geglättet wird.

Hintergrundinformationen

Getriebefeinabgleich

Der Zugkraftregler arbeitet in der Regel mit Geschwindigkeiten größer "0". Dabei ist der Getriebefeinabgleich ein probates Mittel um die Zugkraft aufrechtzuerhalten. Allerdings gibt es dabei auch einige Randbedingungen, die hier erläutert werden.

Der Getriebefeinabgleich wird intern auf ein Minimum von -95% begrenzt. Wird ein kleinerer Wert eingegeben, erzeugt der Baustein einen Fehler.

Hintergrund dieser Begrenzung ist der errechnete Ausgangswert der bei gegebenem Eingangswert wie folgt definiert ist:

$$\text{Ausgangswert} = \text{Eingangswert} * \left(1 + \frac{\text{Getriebefeinabgleich}}{100} \right)$$

Abb. 7-16: Ausgangswert

Anhand der Gleichung kann man erkennen, dass - wenn der Getriebefeinabgleich gegen -100% geht - sich der Ausgangswert gegen "0" bewegt. Damit würde der Regelkreis praktisch geöffnet werden, da eine Änderung des Eingangswertes keinerlei Reaktion am Ausgang bewirkt. Schlimmer noch, bei einem Getriebefeinabgleich der kleiner als -100% wäre, kehrt sich das Vorzeichen des Ausgangswertes um. Im Regelkreis würde eine Mitkopplung entstehen, die das System instabil machen kann.

StandstillEnable, StandstillVel, StandstillVelDeadband

Die Regelung im Stillstand kann durch Verwendung der Eingänge "StandstillEnable" und "StandstillVel" erfolgen. Dabei wird eine additive Geschwindigkeit aufgeschaltet. Um eine bessere Regelgüte zu erreichen, wird diese um den Sollwert herum verringert. Dies geschieht mit Hilfe einer abgewandelten Signum-Funktion, die nachfolgende Übertragungsfunktion besitzt:

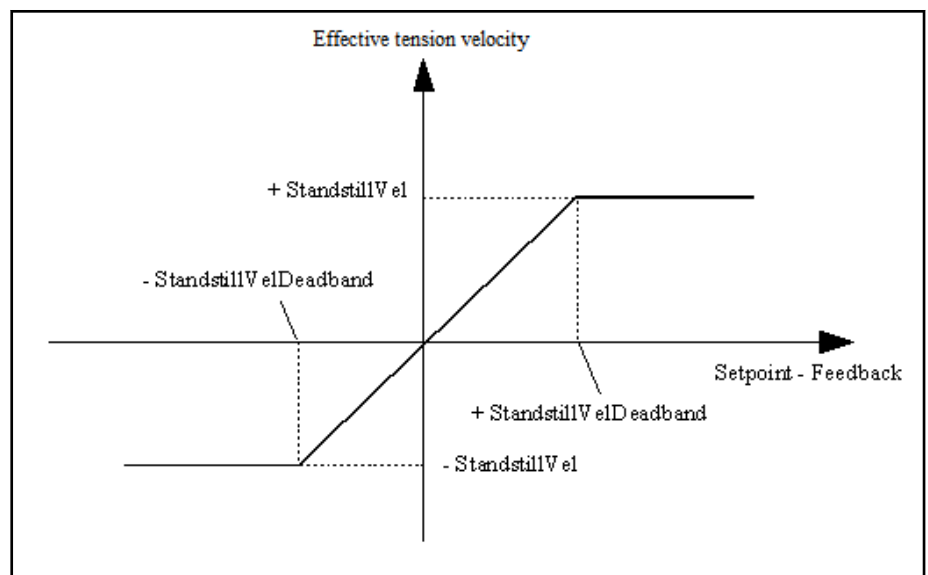


Abb. 7-17: Abgewandelte Signum-Funktion für die Begrenzung der aufgeschalteten StandstillVel

Die Regelabweichung wird dadurch im Bereich des "Setpoint" relativiert. Nähert sich der "Feedback" dem "Setpoint", wird die Regelung "feinfühlicher" als ein reiner Zweipunktregler. Der Wert bei dem diese Begrenzung stattfindet kann man am Eingang "StandstillVelDeadband" einstellen. Die Werte besitzen dabei dieselben Einheiten wie der "Setpoint", bzw. "Feedback" Eingang. Zwischen den Extremwerten "+StandstillVel" und "-StandstillVel" wird linear interpoliert, so dass sich obige Übertragungsfunktion für Regelabweichung und Spanngeschwindigkeit ergibt.

Anschließend wird der ermittelte Faktor mit der vorgegebenen "StandstillVel" multipliziert. Die Geschwindigkeit wird dabei als Drehzahl, in der im Antrieb eingestellten Wichtung, verarbeitet. Möchte man anstatt der Drehzahl eine Bahngeschwindigkeit eingeben, ist diese vorher selbst zu berechnen.

Um eine umgekehrte Bewegungsrichtung für die Regelung im Stillstand zu erreichen, ist der Eingang "NegPolarity" zu setzen.

Nachfolgend wird noch mit einem Korrekturfaktor multipliziert. Dieser Korrekturfaktor setzt sich aus dem Getriebeübersetzungsverhältnis zusammen. Jedoch ist der Getriebefeinabgleich nicht berücksichtigt. In der Praxis ist der Getriebefeinabgleich in der Regel sehr klein, so dass dieser nicht ins Gewicht fällt. Das Getriebeübersetzungsverhältnis wird nur bei einer steigenden Flanke des "Enable"-Eingangs übernommen.

Im Flussdiagramm ist ersichtlich, dass die angegebene Geschwindigkeit in die additive Leitachsposition in Inkrementen bzw. Grad umgerechnet wird. Dabei sind einige Regeln zu beachten:

- Die angegebene Geschwindigkeit kann nicht beliebig hoch gewählt werden, denn der Antrieb muss die Geschwindigkeit innerhalb eines sercos Zyklus erreichen können. Damit sind nur sehr kleine Werte zulässig. Die Größe hängt dabei von der verwendeten Wichtung und der Masse am Motor ab. Die errechneten Inkremente/Position werden über den Parameter P-0-0692 (Slave-Achse mit Interpolation auf dem Antrieb) bzw. A-0-2600 (Slave-Achse mit Interpolation auf der Steuerung) vorgegeben, dieser kann zudem über Filter P-0-0693 (Slave-Achse mit Interpolation auf dem Antrieb) bzw. A-0-2601 (Slave-Achse mit Interpolation auf der Steuerung) geglättet werden, so dass auch größere Sprünge möglich sind

- Die errechneten Inkremente werden noch vor dem Getriebe aufaddiert. Das Getriebeübersetzungsverhältnis (A-0-2720 und A-0-2721 bzw. P-0-0156 und P-0-0157) wird dabei mit eingerechnet. Der Getriebefeinabgleich (P-0-0083/A-0-2722 und P-0-0694/A-0-2605) wird jedoch nicht berücksichtigt, da dieser in der Regel sehr klein ist. Das kann dazu führen, dass die Achse sich mit einer geringfügig anderen, als der eingestellten Geschwindigkeit bewegt
- Es wird eine Skalierung der Einheiten durchgeführt. Außerdem wird der eingegebene Geschwindigkeitswert nicht nach Maximum- und Minimumwerten überprüft. Dadurch können sich sehr große Werte für die Anzahl der Inkremente/Position je sercos Zyklus ergeben. Überschreitet der errechnete Zwischenwert die Maximalwerte von $\pm$ halber Modulowert, werden diese automatisch begrenzt. Die Folge ist eine niedrigere additive Geschwindigkeit, als am Eingang "StandstillVel" vorgegeben wird
- Bei sehr kleinen Geschwindigkeiten ist außerdem zu beachten, dass die aktuelle Geschwindigkeit von der angegebenen Geschwindigkeit abweichen kann. Dies ist durch die Granulierung der Inkremente je sercos Zyklus zu erklären: entspricht z.B. 1 Inkrement einer Geschwindigkeit von 0,03 U/min, so können nur Geschwindigkeiten die ein Vielfaches von 0,03 U/min sind, wirksam werden. Ist dies nicht der Fall, wird zum nächstgelegenen Wert gerundet.
- Die errechneten Inkremente werden mit jedem sercos Zyklus verarbeitet. Deshalb muss bei Verwendung der Regelung im Stillstand der Baustein in der Motion/sercos synchronen Task abgearbeitet werden. Ist dies nicht der Fall, wird sich der Antrieb einen sercos Zyklus um die errechnete Anzahl an Inkrementen bewegen, und in den folgenden sercos Zyklen wieder still stehen, was eine ruckende Bewegung zur Folge hat. Der Einsatz des Filters in P-0-0693 kann auch hier Abhilfe schaffen.



Bei Slave-Achsen mit Interpolation auf dem Antrieb wird empfohlen, die Filterzeitkonstante des P-0-0693 mit der 4-fachen sercos Zykluszeit einzustellen. Ist die Filterzeitkonstante = "0", so wird der Antrieb bei Spannen im Stillstand die Warnung E2070 "Beschleunigungsbegrenzung aktiviert" ausgeben, da die im sercos Takt übertragene additive Leitachspannung nicht innerhalb des Lageregeltaktes im Antrieb feininterpoliert wird.

Fehlercodes

Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle **F_RELATED_TABLE**, 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0004)	16#00000A00	16#00000001	Der Parameter P-0-0694 ist nicht im optional zyklischen Telegramm der Folgeachse parametrierbar
ACCESS_ERROR (16#0004)	16#00000A00	16#00000002	Der Parameter P-0-692 ist nicht im optional zyklischen Telegramm der Folgeachse parametrierbar
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000001	Die Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") ist < 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000002	Die Leitachsgeschwindigkeit im oberen Betriebspunkt ("MasterVelocityHigh") ist < 0

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000003	Die Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") und oberen Betriebspunkt ("MasterVelocityHigh") besitzen die gleichen Werte
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000004	Der Wert der Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") ist größer als der Wert im oberen Betriebspunkt ("MasterVelocityHigh")
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000005	Der Wert des Eingangs "LowLimit" ist kleiner als der minimal zulässige (-95.0)
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000006	Der Wert der Eingänge "PGainLowVel" oder "PGainHighVel" ist kleiner oder gleich 0
DEVICE_ERROR (16#0008)	16#00000A02	16#00000007	Bei dem in "Slave" angegebenen Antrieb ist keine Leistung zugeschaltet. Eine Zugkraftregelung ist daher nicht möglich
DEVICE_ERROR (16#0008)	16#00000A02	16#00000008	Der in "Slave" angegebene Antrieb befindet sich nicht in einer synchronen Betriebsart
INPUT_INVALID_ERROR (16#0001)	16#00000A03	16#00000000	Ein SlavePreCtrl-Eingang hat einen ungültigen Wert
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000000	"Circumference" <= 0.0
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000001	"Distance" <= 0.0

Tab. 7-10: Fehlercodes Funktionsbaustein MB_TensionControlLoadCellType02

7.5 MB_TensionControlLoadCellMatParType02

Kurzbeschreibung

Der Multiachs-Zugkraftregler MB_TensionControlLoadCellMatParType02 regelt abhängig vom Messwert einer Kraftmessdose die Bahnzugkraft eines führenden Bahnabschnitts, entkoppelt von den nachfolgenden Bahnzugkräften, auf einen gewünschten Sollwert.

In Vergleich zu dem Funktionsbaustein [MB_TensionControlLoadCellType02, Seite 447](#), bietet diese Variante nachfolgende Vorzüge.

- Reglerparametrierung durch Maschinen- und Materialdaten
- Individuelle Reglerauslegung (Führungs- oder Störverhalten)
- Individuelle Regelkreisdynamik (aperiodisch- oder periodisch)
- Automatische Adaption der Reglerparameter

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechTensionAdvanced

Tab. 7-11: Referenztable MB_TensionControlLoadCellMatParType02



Die Verwendung des Funktionsbausteines unterliegt einer Lizenzpflicht (Details siehe [Einführung und Übersicht, Seite 439](#)).



Der Eingang "ResponseCurve" ist in der Bibliothek RIL_LoopControl deklariert. Daher muss die Bibliothek RIL_LoopControl zusätzlich im Bibliotheksverwalter aufgenommen werden, wenn der Eingang "ResponseCurve" genutzt wird.

Schnittstellenbeschreibung

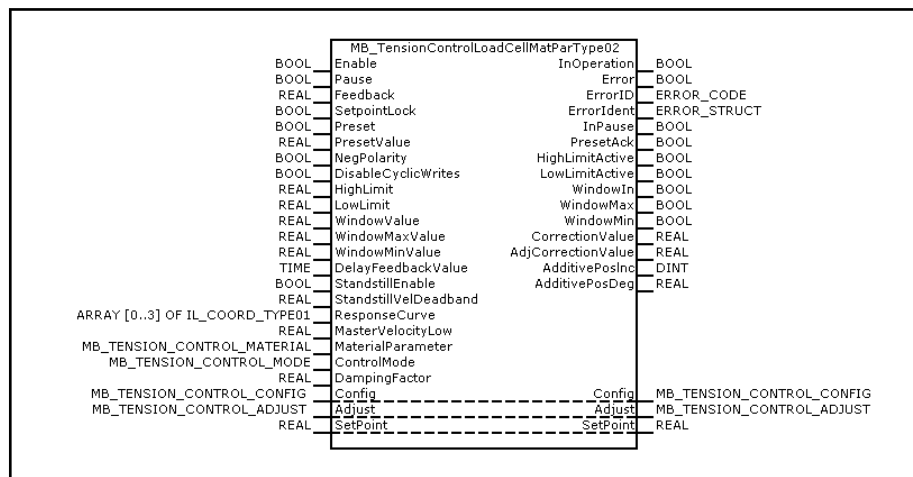


Abb. 7-18: Funktionsbaustein MB_TensionControlLoadCellMatParType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Wird dieser Eingang gesetzt, arbeitet der Funktionsbaustein
	Pause	BOOL	Bei gesetztem Eingang wird der Zugkraftregler angehalten und der Ausgangswert eingefroren. Der Ausgang "InPause" signalisiert ein aktives "Pause"
	Feedback	REAL	Aktueller Istwert der Zugkraft [N]*
	SetpointLock	BOOL	Bei einer steigenden Flanke an diesem Eingang wird der aktuelle Zugkraftistwert "Feedback" als neuer Sollwert "Setpoint" für den Zugkraftregler übernommen
	Preset	BOOL	Bei einer steigenden Flanke an diesem Eingang wird der Reglerausgang auf den in "PresetValue" parametrierbaren Wert gesetzt und bei Erfolg der Ausgang "PresetAck" gesetzt
	PresetValue	REAL	Wert auf den der Reglerausgang "CorrectionValue" gesetzt wird, wenn der Eingang "Preset" gesetzt wird [%]
	NegPolarity	BOOL	Bei gesetztem Eingang wird der Regelsinn invertiert
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben

E/A-Typ	Name	Datentyp	Beschreibung
	HighLimit	REAL	Maximalwert den der Reglerausgang annehmen kann [%]. Wird die vom Regler berechnete Stellgröße größer als dieser Wert, wird die Stellgröße begrenzt und der I-Anteil des Reglers nicht weiter aufsummiert
	LowLimit	REAL	Minimalwert den der Reglerausgang annehmen kann [%]. Wird die vom Regler berechnete Stellgröße kleiner als dieser Wert, wird die Stellgröße begrenzt und der I-Anteil des Reglers nicht weiter aufsummiert
	WindowValue	REAL	Prozessgrößenfenster (relativ) [%]
	WindowMaxValue	REAL	Maximalwert der Prozessgröße (absolut) [%]
	WindowMinValue	REAL	Minimalwert der Prozessgröße (absolut) [%]
	DelayFeedbackValue	TIME	Verzugszeit der Bahnzugkraftermittlung (ausschließlich der sercos Zykluszeit)
	StandstillEnable	BOOL	StandstillEnable = FALSE: Zweig "InVelocity Control" ist aktiv, Stellgröße: Getriebefeinabgleich StandstillEnable = TRUE: Zweig "Regelung im Stillstand" ist aktiv, Stellgröße: Leitachsposition additiv
	StandstillVelDeadband	REAL	Maximal wirksame Regelabweichung in der Stillstandsregelung [N]*. Entsprechend dieser Regelabweichung wird die Spanngeschwindigkeit begrenzt
	ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	Kennlinie zur Gewichtung der Regelabweichung im Zweig "InVelocity Control" IL_COORD_TYPE01, RIL_LoopControl.library
	MasterVelocityLow	REAL	Leitachsgeschwindigkeit unterhalb derer die Regleradaption begrenzt wird [1/min]
	MaterialParameter	MB_TENSION_CONTROL_MATERIAL	Struktur mit materialspezifischen Daten (Typ, Breite, Dicke, Grammatur und E-Modul) MB_TENSION_CONTROL_MATERIAL, Seite 305
	ControlMode	MB_TENSION_CONTROL_MODE	Auswahl des Regelkreisverhaltens. Optimiert auf Führungs- oder Störverhalten MB_TENSION_CONTROL_MODE, Seite 306
	DampingFactor	REAL	Dieser Eingang wirkt auf das Regelkreisverhalten [%]. Es kann stufenlos zwischen einem langsamen aperiodischen und einem schnellen periodischen Verhalten gewählt werden
VAR_OUTPUT	InOperation	BOOL	Der Zugkraftregler arbeitet und die Ausgänge sind gültig
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Fehlerkurzbeschreibung
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerbeschreibung gemäß Fehlertabelle (siehe Fehlercodes, Seite 477)

E/A-Typ	Name	Datentyp	Beschreibung
	InPause	BOOL	"Pause" ist aktiv
	PresetAck	BOOL	Ist dieser Ausgang aktiv, wurde der "Preset" Eingang ausgewertet und der Ausgang "CorrectionValue" hat den Wert "PresetValue" angenommen
	HighLimitAct	BOOL	Dieser Ausgang wird gesetzt, wenn die vom Regler berechnete Stellgröße größer als der in "HighLimit" angegebenen Wert ist
	LowLimitAct	BOOL	Dieser Ausgang wird gesetzt, wenn die vom Regler berechnete Stellgröße kleiner als der in "LowLimit" angegebenen Wert ist
	WindowIn	BOOL	Wenn TRUE, befindet sich die Prozessgröße innerhalb des parametrisierten Fensters
	WindowMax	BOOL	Wenn TRUE, befindet sich die Prozessgröße oberhalb des parametrisierten Maximalwertes
	WindowMin	BOOL	Wenn TRUE, befindet sich die Prozessgröße unterhalb des parametrisierten Minimalwertes
	CorrectionValue	REAL	Stellgröße des Zugkraftreglers [%]
	AdjCorrectionValue	REAL	Stellgröße des Zugkraftreglers mit dynamischer Vorsteuerung [%]
	AdditivePosInc	DINT	Additive Leitachspannung in Inkrementen, genutzt für reale Slave-Achse mit Interpolation auf dem Antrieb (Zielparameter: P-0-0692) [-]
	AdditivePosDeg	REAL	Additive Leitachspannung in Grad, genutzt für Slave-Achse mit Interpolation auf der Steuerung (z.B. für virtuelle Achsen oder reale Achse mit Interpolation auf der Steuerung, Zielparameter: A-0-2600) [°]
VAR_IN_OUT	Config	MB_TENSION_CONTROL_CONFIG	Konfiguration des Zugkraftreglers (Achsen, Zugkraftverteilung) MB_TENSION_CONTROL_CONFIG, Seite 478
	Adjust	MB_TENSION_CONTROL_ADJUST	Struktur für die Folgeachseinstellungen im laufenden Betrieb MB_TENSION_CONTROL_ADJUST, Seite 478
	Setpoint	REAL	Sollwert der Zugkraft [N]*. Diese wird in der Regel durch den Anwender gesetzt, kann aber auch vom Funktionsbaustein selbst geändert werden (siehe SetpointLock, Seite 470)

*

Die Einheit [N] ist bei Verwendung der internen Materialdatenbank (E [N/mm²]) obligatorisch. Bei der Verwendung von benutzerdefinierten Elastizitätsmoduln kann eine beliebige Einheit wie z.B. [kg] verwendet werden. Es ist darauf zu achten, dass die gewählte Einheit einheitlich verwendet wird.

Tab. 7-12: Schnittstellenvariablen Funktionsbaustein MB_TensionControlLoad-CellMatParType02

Min.- Max.- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Default-Werte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL	FALSE	TRUE	FALSE	kontinuierlich
Pause	BOOL	FALSE	TRUE	FALSE	kontinuierlich
Feedback	REAL	n.def.	n.def.	0.0	kontinuierlich
SetpointLock	BOOL	FALSE	TRUE	FALSE	kontinuierlich
Preset	BOOL	FALSE	TRUE	FALSE	kontinuierlich
PresetValue	REAL	-95	n.def.	0.0	steigende Flanke an "Preset"
NegPolarity	BOOL	FALSE	TRUE	FALSE	kontinuierlich
DisableCyclicWrites	BOOL	FALSE	TRUE	FALSE	steigende Flanke an "Enable"
HighLimit	REAL	>LowLimit	n.def.	200	kontinuierlich
LowLimit	REAL	-95.0	<HighLimit	-50	kontinuierlich
WindowValue	REAL	n.def.	n.def.	0.0	kontinuierlich
WindowMaxValue	REAL	>=WindowMinValue	n.def.	0.0	kontinuierlich
WindowMinValue	REAL	n.def.	<=WindowMaxValue	0.0	kontinuierlich
DelayFeedbackValue	TIME	n.def.	n.def.	T#100ms	kontinuierlich
StandstillEnable	BOOL	FALSE	TRUE	FALSE	kontinuierlich
StandstillVelDeadband	REAL	>0.0	n.def.	1.0	kontinuierlich
ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	siehe RIL_Loop-Control	siehe RIL_Loop-Control	siehe RIL_Loop-Control	steigende Flanke an "Enable"
MasterVelocityLow	REAL	>0.0	n.def.	1.0	kontinuierlich
MaterialParameter	MB_TENSION_CONTROL_MATERIAL	siehe Datenstruktur	siehe Datenstruktur	siehe Datenstruktur	kontinuierlich
ControlMode	MB_TENSION_CONTROL_MODE	siehe Datenstruktur	siehe Datenstruktur	siehe Datenstruktur	kontinuierlich
DampingFactor	REAL	20.0	200.0	100.0	kontinuierlich
Config	MB_TENSION_CONTROL_CONFIG	siehe Datenstruktur	siehe Datenstruktur	siehe Datenstruktur	steigende Flanke an "Enable"
Adjust	MB_TENSION_CONTROL_ADJUST	siehe Datenstruktur	siehe Datenstruktur	siehe Datenstruktur	kontinuierlich
Setpoint	REAL	n.def.	n.def.	0.0	kontinuierlich

Tab. 7-13: Eingangsverhalten MB_TensionControlLoadCellMatParType02

Funktionsbeschreibung Wie in der nachfolgenden Abbildung dargestellt, werden mehrere Zugwalzen durch den Funktionsbaustein MB_TensionControlLoadCellMatParType02 gesteuert. Der Messwert der Kraftmessdose wird auf den "Feedback"-Eingang des Funktionsbausteins gelegt.

Die Sollzugkraft wird über den Eingang "SetPoint" eingestellt. Das resultierende Reglerstellsignal wird auf bis zu 8 Zugwalzen aufgeschaltet. Hierdurch ist es diesem Multiachs-Bahnzugkraftregler möglich, die Bahnzugkraft des führenden Bahnabschnitts entkoppelt von den nachfolgenden Bahnzugkräften zu regeln.

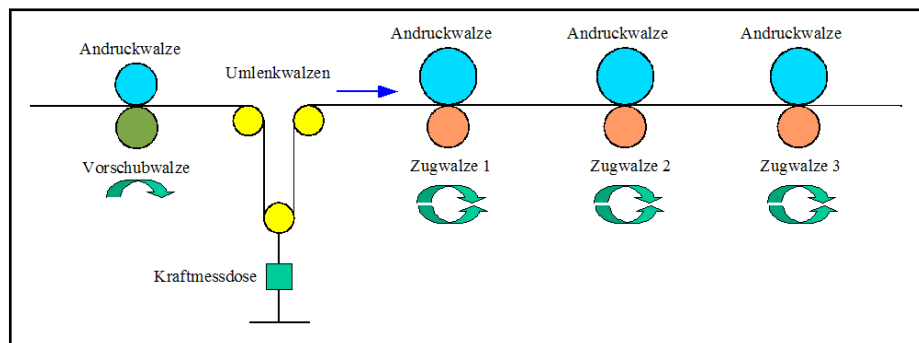


Abb. 7-19: Multiachs-Zugkraftregler

Die Wirkungsweise des Zugkraftreglers ist in den folgenden Flussdiagrammen dargestellt. Je nach Eingang "StandstillEnable" ist entweder der Pfad "InVelocity Control" oder "Standstill Control" wirksam.

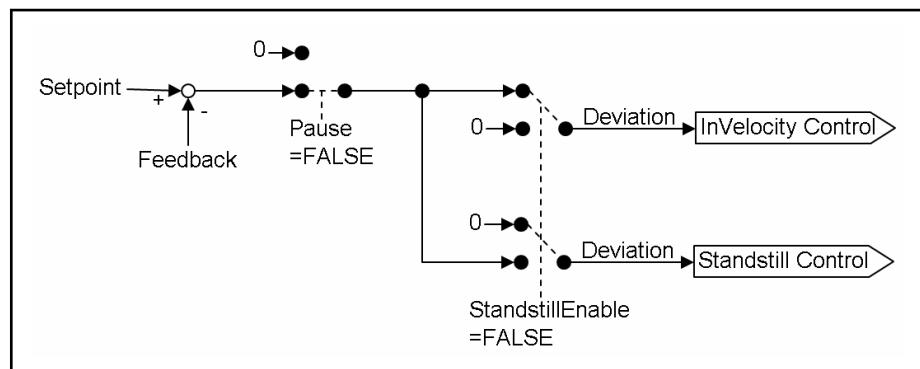


Abb. 7-20: Flussdiagramm des Zugkraftreglers mit Abzweig für Regelung im Stillstand / In Geschwindigkeit

Der Pfad "InVelocity Control" ist wirksam, wenn der Eingang "Standstill-Enable" nicht gesetzt ist. Die Zugkraft wird hierbei über den Parameter "Getriebefeinabgleich" P-0-0694 (Slave-Achse mit Interpolation auf dem Antrieb / IPO Drive) bzw. A-0-2605 (Slave-Achse mit Interpolation auf der Steuerung / IPO Control) geregelt.

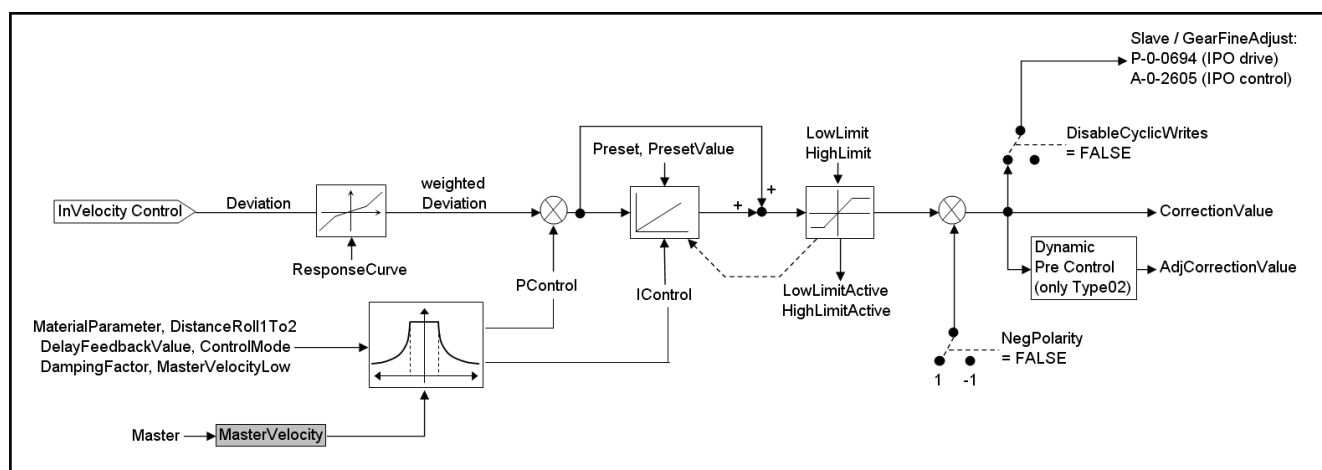


Abb. 7-21: Flussdiagramm des Zugkraftreglers für den Zweig (InVelocity Control)

Für die Regelung im Stillstand kann der Pfad "Standstill Control" aktiv geschaltet werden. Dieser Pfad ist wirksam, wenn der Eingang "Standstill-Enable" gesetzt ist. Die Zugkraft wird hierbei über den Parameter "additive Leitachsposition" P-0-0692 (IPO Drive) bzw. A-0-2600 (IPO Control) geregelt.

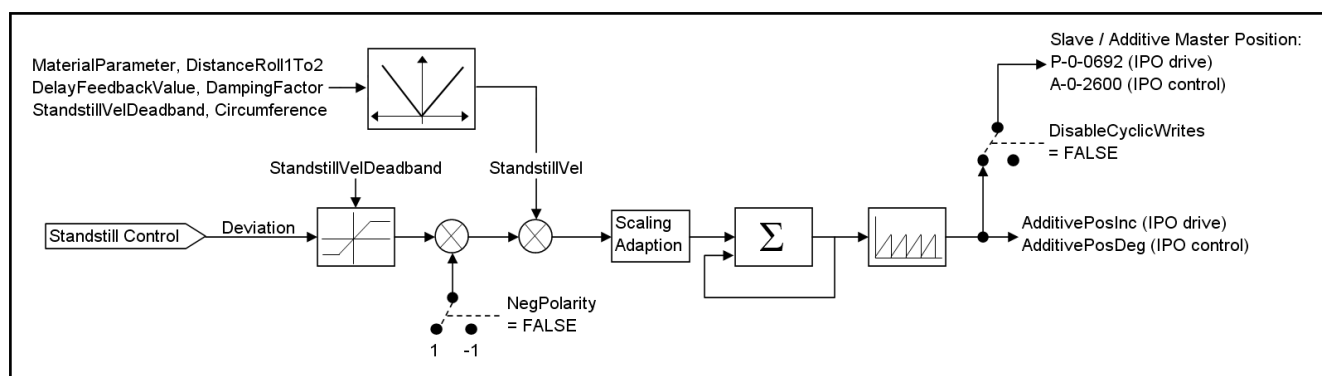


Abb. 7-22: Flussdiagramm des Zugkraftreglers für die Regelung im Stillstand (Standstill Control)

Die Flussdiagramme enthalten aus Gründen der Übersichtlichkeit nicht die Begrenzungen "WindowValue", "WindowMaxValue", "WindowMinValue" sowie deren Statusanzeigen.

Inbetriebnahme

Ist "DisableCyclicWrites" inaktiv, und es werden reale Slave-Achsen (Config.Slave[1..8]) mit Interpolation auf dem Antrieb verwendet, müssen die Parameter P-0-0694 ("InVelocity Control") und P-0-0692 ("Standstill Control") im zyklischen Sollwertkanal der Slave-Achsen aufgelegt werden. Bei steigender Flanke an "Enable" werden die zyklischen Datenkanäle überprüft. Wird einer der beiden Parameter nicht gefunden, erfolgt eine entsprechende Fehlermeldung an den Fehlerausgängen des Funktionsbausteins. Ist "DisableCyclicWrites" aktiv, werden die Konfigurationen der zyklischen Sollwertdatenkanäle nicht überprüft.



Es ist möglich die Slave-Achsen gemischt zu konfigurieren (z.B. Slave-Achse 1 mit Interpolation auf der Steuerung und Slave-Achse 2 mit Interpolation im Antrieb). **Bei einer gemischten Konfiguration muss der Parameter P-0-0750 jeder realen Slave-Achse mit Interpolation auf dem Antrieb auf dem Defaultwert = 1 belassen werden.**

Im Folgenden werden die einzelnen Eingänge des Funktionsbausteins näher erläutert.

Allgemein gilt für alle Eingänge, dass sie kontinuierlich bei gesetztem "Enable"-Eingang ausgewertet werden. Das heißt, Änderungen der Eingänge werden sofort ohne erneutes Toggeln des "Enable"-Eingangs wirksam. Ausnahme dieser Regel bilden die Eingänge "PresetValue", "DisableCyclicWrites", "ResponseCurve" und "Config". Der "PresetValue"-Eingang wird hierbei nur bei einer steigenden Flanke am "Preset"-Eingang, und die Eingänge "DisableCyclicWrites", "ResponseCurve" und "Config" nur bei einer steigenden Flanke am "Enable"-Eingang, übernommen.

Pause, InPause

Der Eingang "Pause" ermöglicht das Anhalten des Zugkraftreglers. Wird der Eingang auf TRUE gesetzt, wird die Regelabweichung auf "0" gesetzt, d.h. der Reglerausgang wird auf dem aktuellen I-Anteil eingefroren. Ist "Pause" aktiv, wird der Ausgang "InPause" gesetzt.



Wenn keine Zugkraftregelung im Stillstand genutzt wird (StandstillEnable ist immer FALSE), muss im Maschinenstillstand der Eingang "Pause" gesetzt werden.

Feedback

Dieser Eingang stellt den Istwert der Zugkraft dar. Dieser kann z.B. durch einen Analogeingang am Antrieb oder den Inline I/O eingelesen werden. Der Anwender ist dabei für eine sinnvolle Skalierung verantwortlich.

SetpointLock

Durch Setzen des Eingangs "SetpointLock" wird der aktuelle Zugkraftistwert als neuer Sollwert "Setpoint" für den Zugkraftregler übernommen. Die Funktion erlaubt es, z.B. für den Sollwert der Zugkraft einen sinnvollen Startwert zu erhalten, der den momentanen Zustand der Maschine widerspiegelt.

Preset, PresetValue, PresetAck

Durch eine positive Flanke am Eingang "Preset" wird über die Variable "PresetValue" ein Anfangswert für die Stellgröße "CorrectionValue" vorgegeben. Gleichzeitig werden interne Variablen des Reglers so gesetzt, dass ein ruckfreies Anfahren des Reglers erfolgt. Der "Preset" wirkt nur auf den Getriebe-feinabgleich im Pfad "InVelocity Control".

Der "Preset" Eingang wird nur ausgewertet, wenn "Enable" gesetzt ist. Bei steigender Flanke an "Preset" wird der neue Wert an "PresetValue" übernommen und "PresetAck" gesetzt. Eine fallende Flanke an "Preset" oder "Enable" löscht "PresetAck". Der "Preset"-Eingang ist dominant über "Pause", d.h. der Wert an "PresetValue" erscheint am Ausgang, auch wenn "Pause" gesetzt ist. Ist bei steigender Flanke an "Enable" "Preset" gesetzt, so wird "PresetValue" sofort als Stellgröße übernommen.



Die Preset Funktion sollte im Stillstand ausgeführt werden (StandstillEnable=TRUE), da ansonsten Geschwindigkeitssprünge durch das Setzen des "CorrectionValue" auf den "PresetValue" entstehen können. Der Betrag der Geschwindigkeitssprünge ist proportional zu der Geschwindigkeit.

Der "Preset"-Eingang wirkt nur bei der steigenden Flanke.

NegPolarity

Der Eingang "NegPolarity" beeinflusst den Regelsinn. Abhängig von der Lage der Zugwalze, bezüglich der Durchlaufrichtung des Materials und der Lage der Vorschubwalze, muss der Regelsinn entweder positiv oder negativ sein. Ist dieser Eingang FALSE (positiver Regelsinn), führt eine positive Regelabweichung zu einer positiven Stellbewegung. Der Eingang wirkt auch auf die Stillstandsregelung.

DisableCyclicWrites

Der Eingang "DisableCyclicWrites" bestimmt, ob die vom Funktionsbaustein berechneten Werte direkt über den zyklischen Kanal geschrieben werden (FALSE) oder nur am Bausteinausgang verfügbar sind (TRUE).

Ist "DisableCyclicWrites" auf TRUE gesetzt, ist es möglich die Reglerausgänge "CorrectionValue", "AdjCorrectionValue", "AdditivePosInc" und "AdditivePosDeg" abzugreifen und zu modifizieren, ohne dass diese zyklisch auf den Sollwertkanal der Folgeachse aufgelegt werden. Des Weiteren erfolgt in diesem Fall keine Prüfung, ob die optional zyklischen Parameter konfiguriert sind. Diese Einstellung ist beispielsweise bei der Verwendung des Entkopplungsnetzwerks "MB_TensionControlDecouplingNetworkType01" notwendig, da hierbei erst die Reglerwerte modifiziert und anschließend geschrieben werden.

HighLimit, HighLimitAct

Mit diesem Eingang wird die Stellgröße "CorrectionValue" des Reglers auf einen Maximalwert begrenzt. Tritt die Begrenzung ein, wird der I-Anteil des Reglers nicht weiter aufsummiert ("Anti-reset-windup"). Der maximale Wert wird durch den im Antrieb definierten Maximalwert für den Getriebefeinabgleich bestimmt. Tritt eine Begrenzung ein, wird der Ausgang "HighLimitAct" gesetzt.

LowLimit, LowLimitAct

Mit diesem Eingang wird die Stellgröße "CorrectionValue" des Reglers auf einen Minimalwert begrenzt. Tritt die Begrenzung ein, wird der I-Anteil des Reglers nicht weiter aufsummiert ("Anti-reset-windup"). Der minimale Wert liegt bei -95 unabhängig von ggf. kleineren Werten im Antrieb (siehe [Getriebefeinabgleich, Seite 476](#)). Tritt eine Begrenzung ein, wird der Ausgang "LowLimitAct" gesetzt.

WindowValue, WindowIn

Mit dem Wert "WindowValue" kann der Anwender ein symmetrisches Fenster definieren innerhalb dessen sich der aktuelle Zugkraftistwert befinden soll. Das Fenster ist wie folgt definiert:

$$\left(\text{SetPoint} - |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right) \leq \text{Feedback} \leq \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right)$$

Abb. 7-23: WindowValue, WindowIn

Ist diese Bedingung erfüllt, wird der Ausgang "WindowIn" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt. Es findet nur eine boolesche Ausgabe, aber keinerlei Begrenzung statt.

WindowMaxValue, WindowMax

Mit diesem Wert kann der Anwender einen oberen Wert für den Zugkraftistwert festlegen. Der Wert wird wie folgt ausgewertet:

$$\text{Feedback} > \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowMaxValue} - 100}{100} \right)$$

Abb. 7-24: WindowMaxValue, WindowMax

Ist diese Bedingung erfüllt, wird der Ausgang "WindowMax" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt. Es findet nur eine boolesche Ausgabe, aber keinerlei Begrenzung statt.

WindowMinValue, WindowMin

Mit diesem Wert kann der Anwender einen unteren Wert für den Zugkraftistwert festlegen. Der Wert wird wie folgt ausgewertet:

$$\text{Feedback} < \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowMinValue} - 100}{100} \right)$$

Abb. 7-25: WindowMinValue, WindowMin

Ist diese Bedingung erfüllt, wird der Ausgang "WindowMin" gesetzt. Ist die Bedingung nicht erfüllt, ist der Ausgang nicht gesetzt. Es findet nur eine boolesche Ausgabe, aber keinerlei Begrenzung statt.

DelayFeedbackValue

Der Eingang "DelayFeedbackValue" dient der internen Berechnung der Reglerparameter. Er beschreibt die Verzugszeit T_u der Bahnzugkraftregelstrecke. Somit entspricht dieser Wert der Summe aller peripherer Zeitkonstanten wie Motordynamik, Filterzeiten, Messaufnehmer, Motionzykluszeit.

StandstillEnable, StandstillVelDeadband

Durch Setzen des Eingangs "StandstillEnable" wird zur Regelung der Bahnzugkraft eine additive Geschwindigkeit auf die zu regelnde Achse aufgeschaltet. Diese Geschwindigkeit ist hierbei - solange die Begrenzung "StandstillVelDeadband" nicht aktiv ist - direkt proportional zur Regeldifferenz.



Der Getriebefeinabgleich verliert bei niedrigen Geschwindigkeiten immer mehr an Einfluss. Bei Stillstand der Leitachse besteht sogar ein offener Regelkreis, da der Getriebefeinabgleich dann keinen Einfluss mehr hat. Daher muss der Eingang "StandstillEnable" bei Erreichen des Stillstandsfensters gesetzt werden, damit im Pfad "InVelocity Control" der I-Anteil durch den offenen Regelkreis nicht bis zur Grenze ansteigt.



Bei Achsen mit Interpolation auf der Steuerung wird der Ausgang "AdditivePosInc" im Moduloformat 2^{20} ausgegeben.

Die Stellsignalbegrenzung "StandstillVelDeadband" entspricht einer Begrenzung der Regeldifferenz und somit - aufgrund des proportionalen Reglers - ebenfalls einer Begrenzung des Stellsignals. Der Wert besitzt dieselbe Einheiten wie der "Setpoint"- bzw. "Feedback"- Eingang.

Ist die Differenz aus "Setpoint" und "Feedback" größer als der am Eingang "StandstillVelDeadband" vorgegebene Wert, so wird die Spanngeschwindigkeit begrenzt. Diese Begrenzung der Spanngeschwindigkeit ist unter anderem bei einer neu eingelegten ungespannten Materialbahn relevant. Dieser Eingang sollte entsprechend so eingestellt werden, dass regelungstechnisch relevante Regeldifferenzen ohne Begrenzung ausgeglichen werden können, und prozesstechnisch bedingte Regeldifferenzen durch die begrenzte Spanngeschwindigkeit berücksichtigt werden.

Ein "StandstillVelDeadband" von 5% bis 25% des "Setpoint" sollte für die meisten Prozesse ausreichen.



Im Stillstand wird die Achsgeschwindigkeit sprunghaft erzeugt. Da die Geschwindigkeit innerhalb eines Sercos-Zyklus erreicht werden muss, können keine beliebig hohen Werte vorgegeben werden und der Zugkraftregler muss in der sercosynchronen Task verwendet werden. Der Maximalwert richtet sich dabei nach der maximalen Beschleunigung des Antriebs sowie der angeschlossenen Masse.

Um zu große Geschwindigkeitssprünge zu vermeiden, kann zum einen die Filterzeitkonstante im Antriebsparameter P-0-0693 vergrößert werden (es wird die 4-fache Sercos-Zykluszeit empfohlen) und/oder zum anderen das maximale Stellsignal durch das Stillstandsfenster "StandstillVelDeadband" begrenzt werden.

Um eine umgekehrte Bewegungsrichtung für die Regelung im Stillstand zu erreichen, ist der Eingang "NegPolarity" zu setzen.

ResponseCurve

Mit Hilfe des Eingang "ResponseCurve" kann die Regelabweichung gewichtet werden. Durch diese Funktionalität kann beispielsweise bei kleinen Regelabweichungen eine geringere Gewichtung der Regelabweichung eingestellt werden. Dies ergibt quasi eine geringere Proportionalverstärkung und dadurch ein ruhigeres Regelverhalten im eingeschwungenen Zustand. Bei der

Parametrierung des Eingangs sollten die Reglerparameter zunächst ohne Bewertung eingestellt werden und erst danach eine geringere Gewichtung vorgegeben werden. Hierdurch wird erreicht, dass die Reglerparameter zunächst global anhand der Systemdynamik eingestellt werden und anschließend eine lokale Optimierung (z. B. für den eingeschwungenen Zustand) durchgeführt wird.

Parametriert wird diese lokale Optimierung mittels einer Kennlinie mit bis zu 4 Stützstellen. Negative Abweichungen werden durch Punktspiegelung der Kurve am Ursprung behandelt. Daraus ergeben sich insgesamt neun lineare Sektionen. Die Sektion s5 verläuft durch den Ursprung. Die Sektionen s1 und s9 haben einen Anstieg von 1 und laufen jeweils ins positive oder negative Unendliche. Wird der Eingang unbeschaltet gelassen, wird als Default-Wert eine 1-zu-1 Kennlinie verwendet, d. h. es wird keinerlei Gewichtung durchgeführt.

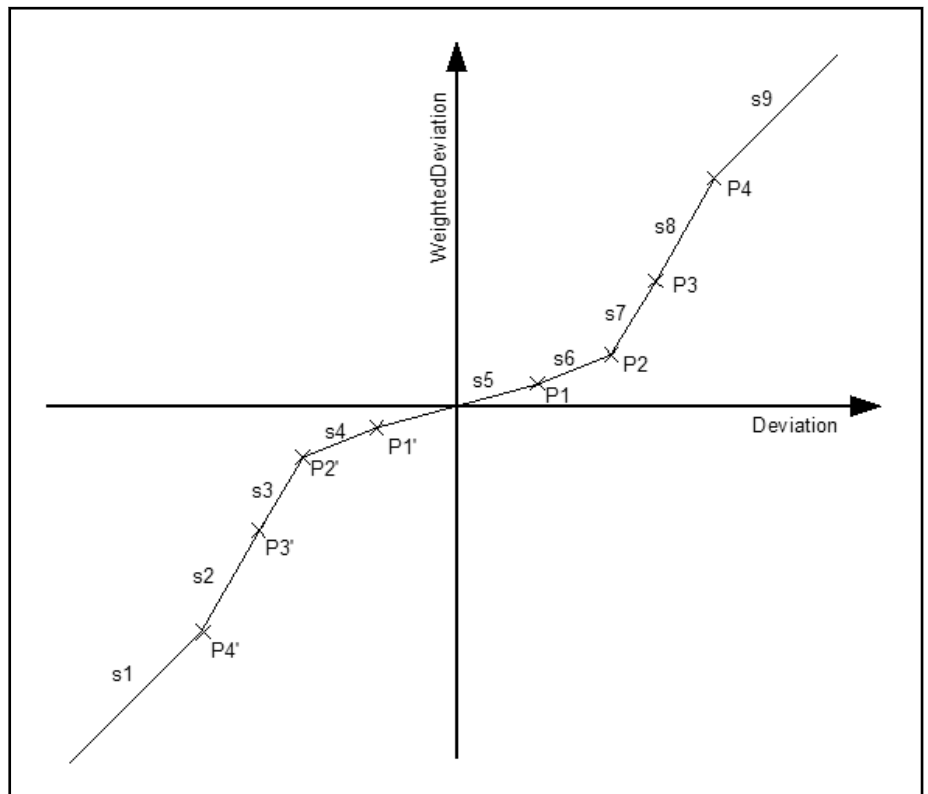


Abb. 7-26: Wichtung der Regelabweichung mit dem Eingang ResponseCurve

Die Punkte P1 bis P4 werden in einem Array aus IL_COORD_TYPE01 abgelegt. Werden weniger als vier Punkte benötigt, muss die X-Koordinate der nicht benötigten Punkte (vom letzten Punkt beginnend) auf 0 gesetzt werden. Die X-Werte müssen positiv sein und in positiver Richtung aufsteigen. Es ist erlaubt, denselben Punkt zweimal abzulegen.

MasterVelocityLow

Der Eingang "MasterVelocityLow" dient zur Begrenzung der Reglerparameter. Infolge des geschwindigkeitsabhängigen Streckenverhaltens ergeben sich geschwindigkeitsabhängige Reglerparameter. Abhängig vom gewünschten Regelkreisverhalten ist entweder die Reglerv Verstärkung und/oder die Reglernachstellzeit geschwindigkeitsabhängig. Nachfolgend ist stellvertretend die Abhängigkeit der Reglerv Verstärkung von der Bahngeschwindigkeit grafisch dargestellt.

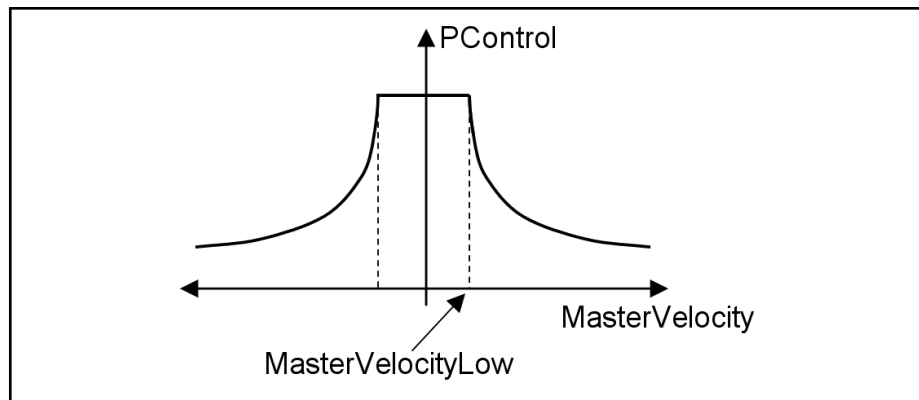


Abb. 7-27: Abhängigkeit der Reglerverstärkung von der Bahngeschwindigkeit

Wie bereits für "StandstillEnable" dargelegt wurde, verliert der Getriebefei-nabgleich bei niedrigen Geschwindigkeiten immer mehr an Einfluss. Dieses Verhalten spiegelt sich direkt in der Reglerverstärkung wieder. Um unreal gro-ßen Reglerparametern vorzubeugen, dient der Eingang "MasterVelocityLow" zur Begrenzung der Reglerparameter bei "kleinen" Bahngeschwindigkeiten. Der Wert dieses Eingangs sollte in den meisten Fällen unterhalb der klein-ten Produktionsgeschwindigkeit und oberhalb der Umschaltgeschwindigkeit in den Stillstandspfad liegen. Somit ist eine uneingeschränkte Regelkreisdyn-amik oberhalb der "MasterVelocityLow"-Geschwindigkeit gewährleistet, und gleichzeitig werden die Reglerparameter bei "kleinen" Bahngeschwindigkeiten begrenzt.

MaterialParameter

Die Eingangsstruktur "MaterialParameter" dient zur internen Berechnung der Reglerparameter.

Diese Struktur beinhaltet Informationen über das individuelle Bahnmaterial. Die einzelnen Elemente dieser Struktur beschreiben den Typ, die Breite, die Dicke bzw. die Grammatur und den Elastizitätsmodul des Bahnmaterials.

Bei Bahnmaterialien aus Papier wird hierbei die Grammatur "WebGrammage" und bei anderen Bahnmaterialien die Materialdicke "WebThickness" aus-gewertet. Bei Auswahl eines vorgegebenen Bahnmaterials "MaterialParame-ter.MaterialType", wird das Elastizitätsmodul aus der internen Datenbank be-zogen. Entsprechend wird der Eingang "MaterialParameter.MaterialEModule" ausschließlich ausgewertet, wenn ein benutzerdefinierter Materialtyp "CUS-TOM_PAPER" oder "CUSTOM_NON_PAPER" ausgewählt wird.

Zur Verdeutlichung sind nachfolgend zwei beispielhafte Beschaltungen dar-gestellt:

Beispiel:

Papier aus bausteininterner Datenbank

```
MaterialParameter.MaterialType := OFFSET_PAPER;
MaterialParameter.WebWidth := 500;
MaterialParameter.WebGrammage := 80;
```

Beispiel:

Benutzerdefiniertes Bahnmaterial (z.B. Folie)

```
MaterialParameter.MaterialType := CUSTOM_NON_PAPER;
MaterialParameter.WebWidth := 500;
MaterialParameter.WebThickness := 200;
```

MaterialParameter.MaterialEModule := 650;

Ist das verwendete Bahnmaterial nicht in der internen Datenbank aufgezählt, und es liegen keine Informationen über den Elastizitätsmodul vor, so kann dieses in einer Stillstandsmessung ermittelt werden.

$$\Delta F \approx A \cdot E \cdot \frac{\Delta L}{L} \Rightarrow E \approx \frac{L \cdot \Delta F}{A \cdot \Delta L}$$

Abb. 7-28: Hookesches Gesetz

Entsprechend dieser Hookeschen Formel ist ersichtlich, dass durch eine gemessene Kraftdifferenz, infolge einer vorgegebenen Streckung (Längenänderung infolge einer additiven Leitachsposition), das individuelle Elastizitätsmodul [N/mm²] im Stillstand ermittelt werden kann. Des Weiteren kann ebenfalls direkt das Produkt aus Elastizitätsmodul [N/mm²] und Querschnittsfläche [mm²] ermittelt werden, und durch Normierung der Querschnittsfläche ("CUSTOM_NON_PAPER", "WebWidth:=1[mm]", "WebThickness:=1000[μm]") auf A=1[mm²] dieses als modifiziertes Elastizitätsmodul [N] vorgegeben werden.

- ControlMode** Durch den Eingang "ControlMode" kann das primäre Regelkreisverhalten festgelegt werden. Der Bahnzugkraftregler kann entweder auf Führungsverhalten oder auf Störverhalten optimiert werden. Bei einer Festwertregelung der Bahnzugkraft sollte dieser Eingang mit "CONTROL_MODE_OPT_DISTURBANCE" belegt werden. Infolgedessen werden die internen Reglerparameter so bestimmt, dass Störungen optimal ausgeregelt werden. Entgegengesetzt sollte bei einer Folgeregelung dieser Eingang mit "CONTROL_MODE_OPT_COMMAND" belegt werden. Infolgedessen werden die internen Reglerparameter so bestimmt, dass die Regelgröße optimal der Führungsgröße folgt.
- DampingFactor** Durch den Eingang "DampingFactor" kann die Regelkreisdynamik festgelegt werden. Er wirkt auf beide Zweige, Stillstandsregelung und Regelung im Bahnlaufbetrieb. Die Regelkreisdynamik kann entweder aperiodisch (ohne Überschwingen) oder periodisch (mit Überschwingen) eingestellt werden. Hierbei entspricht ein "DampingFactor" von 100% - bei exakten Streckenparametern - genau dem aperiodischen Grenzfall, und ein „DampingFactor“ von 120% einem periodischen Einschwingen mit ca. 20% Überschwingen. Des Weiteren kann das Regelkreisverhalten bis zu einem "DampingFactor" von 20% verlangsamt, bzw. bis zu einem "DampingFactor" von 200% beschleunigt werden. Liegen vom realen Verhalten abweichende Streckenparameter ("Circumference", "DistanceRoll1To2", "DelayFeedbackValue", "MaterialParameter") oder Systemverhalten (z.B. orts- und zeitabhängige Feuchtigkeit und Temperaturen) vor, so kann des Weiteren über diesen Eingang eine Korrektur des gewünschten dynamischen Regelkreisverhaltens erzielt werden.
- Config** Über die Struktur "MB_TENSION_CONTROL_CONFIG" werden statische Konfigurationsparameter wie die Achskonfiguration und die Geometrie festgelegt. Der Eintrag "Master" enthält die Achsreferenz auf die Leitachse. Alle Zugwalzen werden im Arrayeintrag "Slave" angegeben. Eine nicht vorhandene Achse wird durch die Konstante "NoAxis" repräsentiert. Für die dynamische Vorsteuerung und die interne Reglerauslegung wird der Umfang der Vorschubwalze ("Master") im Struktureintrag "Circumference" sowie die Bahnlänge zwischen Vorschubwalze ("Master") und erster Zugwalze ("Slave[1]") im Struktureintrag "DistanceRoll1To2" benötigt.
- Adjust** Die Struktur "MB_TENSION_CONTROL_ADJUST" mit dem Array "Slave-PreCtrl" bestimmt die Art der Zugwalzenansteuerung.

Wird für eine Achse die Konstante "NO_PRE_CTRL" eingetragen, erfolgt keine Regelung der Achse. Der Getriebefeinabgleich (P-0-0694 bzw. A-0-2605) wird fest auf "0" eingestellt.

Die Konstante "PRE_CTRL_100" bewirkt, dass der Wert des Ausgangs "CorrectionValue" in den Getriebefeinabgleich geschrieben wird.

In der Einstellung "PRE_CTRL_DYNAMIC" wird der Ausgang "AdjCorrectionValue" auf den Getriebefeinabgleich der Achse geschrieben. Die dynamische Zugwalzensteuerung sollte für alle Walzen hinter der ersten Zugwalze verwendet werden. Hierfür muss die Konstante "PRE_CTRL_DYNAMIC" in "SlavePreCtrl" der nachfolgenden Zugwalzen (2, 3) eingetragen werden. Die erste Zugwalze (1) sollte auf "PRE_CTRL_100" stehen. Diese Einstellung bewirkt, dass die Zugkraft des führenden Bahnabschnitts entkoppelt von den nachfolgenden Bahnzugkräften geregelt wird.

Die Einstellungen in "SlavePreCtrl" haben keinen Einfluss auf die ausgeführte Bewegung im Standstill-Modus.

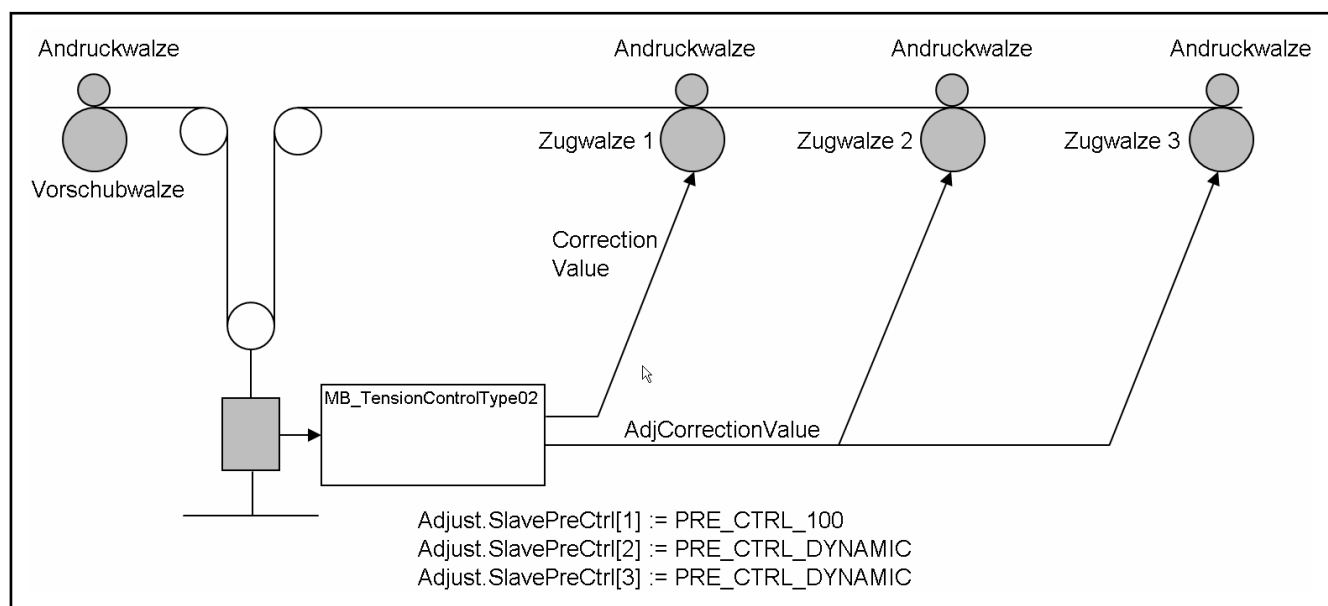


Abb. 7-29: MB_TensionControlLoadCellMatParType02 Beispielskaskade mit 3 Zugwalzen

Hintergrundinformationen

Getriebefeinabgleich

Der Zugkraftregler arbeitet in der Regel mit Geschwindigkeiten größer "0". Dabei ist der Getriebefeinabgleich ein probates Mittel um die Zugkraft zu regeln.

Der Getriebefeinabgleich wird intern auf ein Minimum von -95% begrenzt. Wird ein kleinerer Wert eingegeben, erzeugt der Baustein einen Fehler.

Hintergrund dieser Begrenzung ist der errechnete Ausgangswert der bei gegebenem Eingangswert wie folgt definiert ist:

$$\text{Ausgangswert} = \text{Eingangswert} * \left(1 + \frac{\text{Getriebefeinabgleich}}{100} \right)$$

Abb. 7-30: Ausgangswert

Entsprechend dieser formalen Beschreibung ist erkennbar, dass bei einem Getriebefeinabgleich kleiner "-100%" eine Umkehr der Drehrichtung resultiert.

Fehlercodes Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle **F_RELATED_TABLE**, 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0004)	16#00000A00	16#00000001	Der Parameter P-0-0694 ist nicht im optional zyklischen Telegramm der Folgeachse parametrier
ACCESS_ERROR (16#0004)	16#00000A00	16#00000002	Der Parameter P-0-692 ist nicht im optional zyklischen Telegramm der Folgeachse parametrier
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000001	Die Leitachsgeschwindigkeit im unteren Betriebspunkt ("MasterVelocityLow") ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000005	Der Wert des Eingangs "LowLimit" ist kleiner als der minimal zulässige (-95.0)
DEVICE_ERROR (16#0008)	16#00000A02	16#00000005	Die Sercoszykluszeit (C-0-0503) ist kleiner oder gleich 0
DEVICE_ERROR (16#0008)	16#00000A02	16#00000006	Die Getriebeübersetzung der geregelten Achse ist unzulässig (A-0-2720, A-0-2721)
INPUT_INVALID_ERROR (16#0001)	16#00000A03	16#00000000	Ein SlavePreCtrl-Eingang hat einen ungültigen Wert
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000000	Der Eingang Config.Circumference ist kleiner oder gleich 0.0
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000001	Der Eingang "Config.Distance-Roll1To2" ist kleiner oder gleich 0.0
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000002	Der am Eingang "ControlMode" angegebene Wert ist nicht im Datentyp "MB_TENSION_CONTROL_MODE" enthalten.
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000003	Der am Eingang "MaterialParameter.MaterialType" angegebene Wert ist nicht im Datentyp "MB_TENSION_CONTROL_MATERIAL_TYPE" enthalten.
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000007	Der Eingang "StandstillVelDeadband" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000009	Der Eingang "MaterialParameter.WebWidth" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000010	Der Eingang "MaterialParameter.WebThickness" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000011	Der Eingang "MaterialParameter.MaterialEModul" ist kleiner oder gleich 0

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000012	Der Eingang "MaterialParameter.WebGrammage" ist kleiner oder gleich 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000013	Der Eingang "DampingFactor" ist kleiner als 20 oder größer als 200
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Zustandsmaschine des Funktionsbausteins befindet sich in einen ungültigen Status
RESOURCE_ERROR (16#0003)	16#00001503	16#00000000	AxisData ist in der Steuerung nicht aktiv. Dieser Fehler tritt nur bei der MLD auf

Tab. 7-14: Fehlercodes Funktionsbaustein MB_TensionControlLoadCellMatParType02

7.6 MB_TENSION_CONTROL_CONFIG

Die Struktur MB_TENSION_CONTROL_CONFIG dient der Konfiguration des Zugkraftreglers (Achsen, Zugkraftverteilung).

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Beschreibung
Master	AXIS_REF			NoAxis	Leitachse
Slave[1]	AXIS_REF			NoAxis	Zugwalze 1
Slave[2]	AXIS_REF			NoAxis	Zugwalze 2
Slave[3]	AXIS_REF			NoAxis	Zugwalze 3
Slave[4]	AXIS_REF			NoAxis	Zugwalze 4
Slave[5]	AXIS_REF			NoAxis	Zugwalze 5
Slave[6]	AXIS_REF			NoAxis	Zugwalze 6
Slave[7]	AXIS_REF			NoAxis	Zugwalze 7
Slave[8]	AXIS_REF			NoAxis	Zugwalze 8
Circumference	REAL	0.0	n.def.	0.0	Achsumfang der Vorschubwalze (Master) [mm]*
DistanceRoll1To2	REAL	0.0	n.def.	0.0	Bahnlänge zwischen Vorschubwalze (Master) und erster Zugwalze (Slave[1]) [mm]*

*

Die Einheit [mm] ist bei Verwendung der internen Materialdatenbank (E [N/mm²]) obligatorisch. Bei der Verwendung von benutzerdefinierten Elastizitätsmoduln kann eine beliebige Einheit wie z.B. [ft] verwendet werden. Es ist darauf zu achten, dass die gewählte Einheit einheitlich verwendet wird.

Tab. 7-15: Strukturbeschreibung MB_TENSION_CONTROL_CONFIG

7.7 MB_TENSION_CONTROL_ADJUST

Die Struktur MB_TENSION_CONTROL_ADJUST enthält die Folgeachseinstellungen für die einzelnen Achsen im laufenden Betrieb.

Die Struktur verwendet den Aufzählungstyp [MB_TENSION_PRE_CTRL](#), Seite 479.

Strukturelement	Typ	Default-Wert	Beschreibung
SlavePreCtrl[1]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 1
SlavePreCtrl[2]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 2
SlavePreCtrl[3]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 3
SlavePreCtrl[4]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 4
SlavePreCtrl[5]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 5
SlavePreCtrl[6]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 6
SlavePreCtrl[7]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 7
SlavePreCtrl[8]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Vorsteuerung für Folgeachse 8

Tab. 7-16: Strukturbeschreibung MB_TENSION_CONTROL_ADJUST

7.8 MB_TENSION_PRE_CTRL

Der Aufzählungstyp MB_TENSION_PRE_CTRL beschreibt die möglichen Varianten der Vorsteuerung einer Achse.

Der Aufzählungstyp wird von der Struktur [MB_TENSION_CONTROL_ADJUST](#), Seite 478, verwendet.

Element	Beschreibung
NO_PRE_CTRL	keine Vorsteuerung/Regelung der Achse
PRE_CTRL_100	Vorsteuerung der Achse ohne Zeitkonstante
PRE_CTRL_DYNAMIC	Dynamische Vorsteuerung

Tab. 7-17: Aufbau des Aufzählungstyps MB_TENSION_PRE_CTRL

7.9 Entkopplungsnetzwerk

7.9.1 MB_TensionDecouplingNetworkType01

Kurzbeschreibung Ein Zugkraftregler regelt abhängig vom Messwert einer Kraftmessdose die Zugkraft in einem Bahnabschnitt auf den gewünschten Sollwert (MB_TensionControlLoadCellType01, ML_TechMotion / [MB_TensionControlLoadCellType02](#), Seite 447).

Über das Entkopplungsnetzwerk MB_TensionDecouplingNetworkType01 besteht nun die Möglichkeit, eine Stellsignaländerung eines beliebigen Bahnzugkraftreglers der kontinuierlichen Fertigungsanlage an alle zur Entkopplung benötigten Achsen weiterzuleiten.

Durch diese entkoppelnden Vorsteuerungen wird eine temporäre und stationäre Änderung aller in der Maschine vorhandenen Bahnzugkräfte, ausgenommen der lokal zu regelnden, verhindert.

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	ML_TechTensionAdvanced

Tab. 7-18: Referenztabelle MB_TensionDecouplingNetworkType01



Die Verwendung des Funktionsbausteines unterliegt einer Lizenzpflicht (Details siehe [Einführung und Übersicht](#), Seite 439).

Schnittstellenbeschreibung

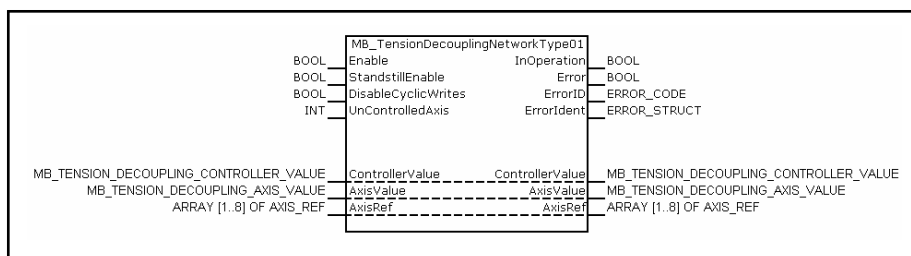


Abb. 7-31: Funktionsbaustein MB_TensionDecouplingNetworkType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Freigabe des Funktionsbausteins (einmalig, flanken-gesteuert)
	StandstillEnable	BOOL	Bei gesetztem Eingang ist die Stillstandsregelung ak-tiv
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	UnControlledAxis	INT	Achse im System, die nicht für die Regelung und Entkopplung der Bahnzugkräfte genutzt wird
VAR_OUTPUT	InOperation	BOOL	Das Entkopplungsnetzwerk arbeitet und die Ausgänge sind gültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose Fehlerbehandlung, Seite 487
VAR_IN_OUT	ControllerValue	MB_TENSION_DECOUPLING_CONTROLLER_VALUE	Struktur der Reglerausgänge (CorrectionValue, Adj-CorrectionValue, AdditivePosInc und AdditivePosDeg) MB_TENSION_DECOUPLING_CONTROLLER_VALUE, Seite 487
VAR_IN_OUT	AxisValue	MB_TENSION_DECOUPLING_AXIS_VALUE	Struktur der Achseingänge (CorrectionValue, AdditivePosInc und AdditivePosDeg) MB_TENSION_DECOUPLING_AXIS_VALUE, Seite 488
VAR_IN_OUT CONSTANT	AxisRef	ARRAY [1..8] OF AXIS_REF	Stellachsen des Systems (ohne "UnControlledAxis")

Tab. 7-19: Schnittstellenbeschreibung MB_TensionDecouplingNetworkType01

Min-/Max- und Defaultwerte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
ControllerValue	MB_TENSION_DECOUPLING_CONTROLLER_VALUE			0	kontinuierlich
AxisValue	MB_TENSION_DECOUPLING_AXIS_VALUE			0	kontinuierlich

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
AxisRef	ARRAY [1..8] OF AXIS_REF			0	steigende Flanke an "Enable"
Enable	BOOL			FALSE	kontinuierlich
StandstillEnable	BOOL			FALSE	kontinuierlich
DisableCyclicWrites	BOOL			FALSE	steigende Flanke an "Enable"
UnControlledAxis	INT	1	9	1	steigende Flanke an "Enable"

Tab. 7-20: Eingangsverhalten MB_TensionDecouplingNetworkType01

Funktionsbeschreibung

Im Folgenden werden die einzelnen Funktionen der Eingänge des Funktionsbausteins näher erläutert. Allgemein gilt für alle Eingänge, dass sie kontinuierlich bei gesetztem "Enable"-Eingang ausgewertet werden. D.h. Änderungen der Eingänge werden sofort wirksam ohne erneutes Toggeln dieses Eingangs. Ausnahme dieser Regel bilden die Eingänge "UnControlledAxis", "DisableCyclicWrites" und das Ein- Ausgangsarray "AxisRef" die nur bei steigender Flanke des "Enable"-Eingangs ausgewertet werden. Alle Eingabewerte bekommen intern den Wert "0" bzw. FALSE zugewiesen, wenn deren Eingänge nicht angeschlossen werden. Ausnahme dieser Regel ist der Eingang "UnControlledAxis", dieser wird bei nicht Belegung auf den Wert 1 gesetzt. Wie schon beim Bahnzugkraftregler MB_Tension_ControlLoadCell-Type02 wird das Stellsignal im normalen Betrieb auf den Getriebefeinabgleich geschrieben. Je nach Achstyp wird hierbei der entsprechende Parameter (P-0-0694 bei Interpolation auf dem Antrieb bzw. A-0-2605 bei Interpolation auf der Steuerung), falls "DisableCyclicWrites" inaktiv ist, direkt beschrieben. Außerdem stehen die Stellsignale aller Achsen immer über die Struktur "MB_TENSION_DECOUPLING_AXIS_VALUE" am Ausgang des Entkopplungsnetzwerks zur Verfügung. Liegen einheitliche Achstypen zugrunde, ist es ausreichend, lediglich das entsprechende Stillstandsstellsignal (AdditivePosInc bei Interpolation im Antrieb oder AdditivePosDeg bei Interpolation in der Steuerung) von den Reglern auf das Entkopplungsnetzwerk aufzuschalten. Liegen gemischte Achstypen (Interpolation auf dem Antrieb und Interpolation in der Steuerung) zugrunde, ist es zwingend notwendig, bei allen Reglern beide Stillstandsstellsignale (AdditivePosInc und AdditivePosDeg) auf des Entkopplungsnetzwerk aufzuschalten. Da die Ausgänge "CorrectionValue", "AdjCorrectionValue", "AdditivePosInc" und "AdditivePosDeg" der Bahnspannungsregler bei der Nutzung des Entkopplungsnetzwerk nicht mehr direkt auf die Achsen geschrieben werden dürfen, ist es zwingend notwendig in allen beteiligten Bahnzugkraftreglern das zyklische Schreiben der Werte aufzuheben (DisableCyclicWrites:=TRUE). Die gesamte Wirkungsweise des Entkopplungsnetzwerks ist in den folgenden Diagrammen dargestellt.

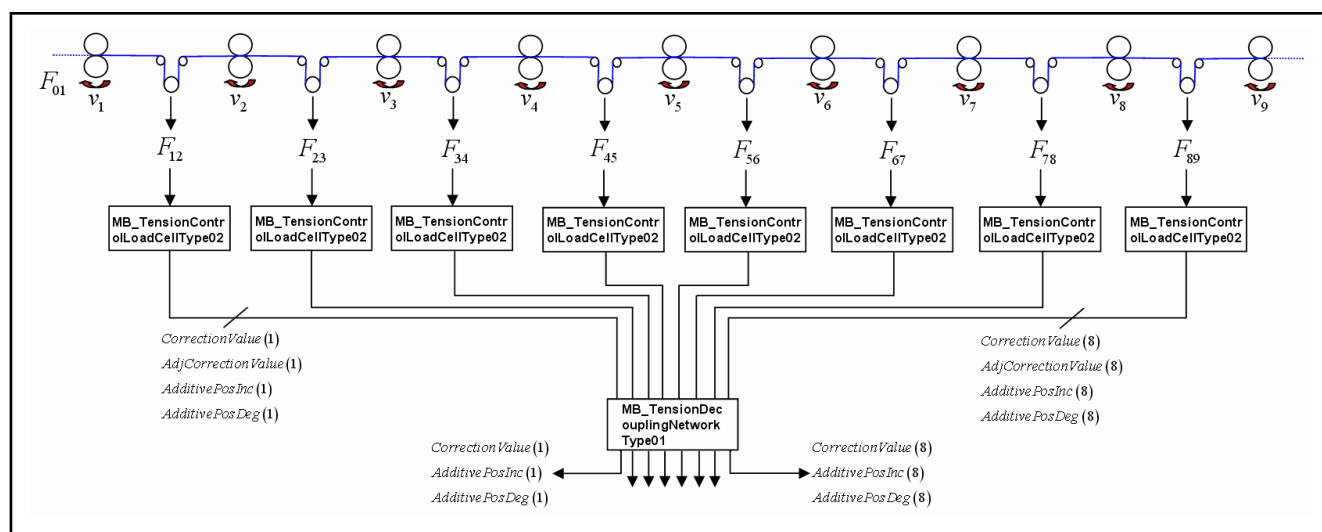


Abb. 7-32: Allgemeines Diagramm des Funktionsbausteins MB_TensionDecouplingNetworkType01

Es ist ersichtlich, dass jeder zu regelnde Bahnabschnitt mit einem Bahnzugkraftregler MB_TensionControlLoadCellType02 ausgestattet werden muss. Hierbei ist zu beachten, dass alle Bahnzugregler im positiven Sinn arbeiten müssen (NegPolarity:=FALSE). Die Reglerausgänge "CorrectionValue", "AdjCorrectionValue", "AdditivePosInc" und "AdditivePosDeg" werden dann dem Entkopplungsnetzwerk MB_TensionDecouplingNetworkType01 zugeführt. In dem dargestellten Beispiel soll in acht Bahnabschnitten die Bahnzugkraft unabhängig voneinander geregelt werden. Somit sind mindestens acht Stellglieder nötig. Da n zu regelnde Bahnabschnitte immer durch n+1 Achsen begrenzt werden, kann eine beliebige Achse gewählt werden, die ausschließlich mit Masterachsgeschwindigkeit betrieben wird und keine zusätzlichen Stellbewegungen ("CorrectionValue", "AdditivePosInc" oder "AdditivePosDeg") ausführen muss. In nachfolgender Darstellung wurde die erste Achse des Systems für die entkoppelte Regelung der Bahnzugkräfte ausgeschlossen (UnControlledAxis:=1). Diese Entkopplungsvariante entspricht zugleich der Grundfunktion des MB_TensionControlLoadCellType02, wenn nur der erste Bahnzugkraftregler aktiv ist.

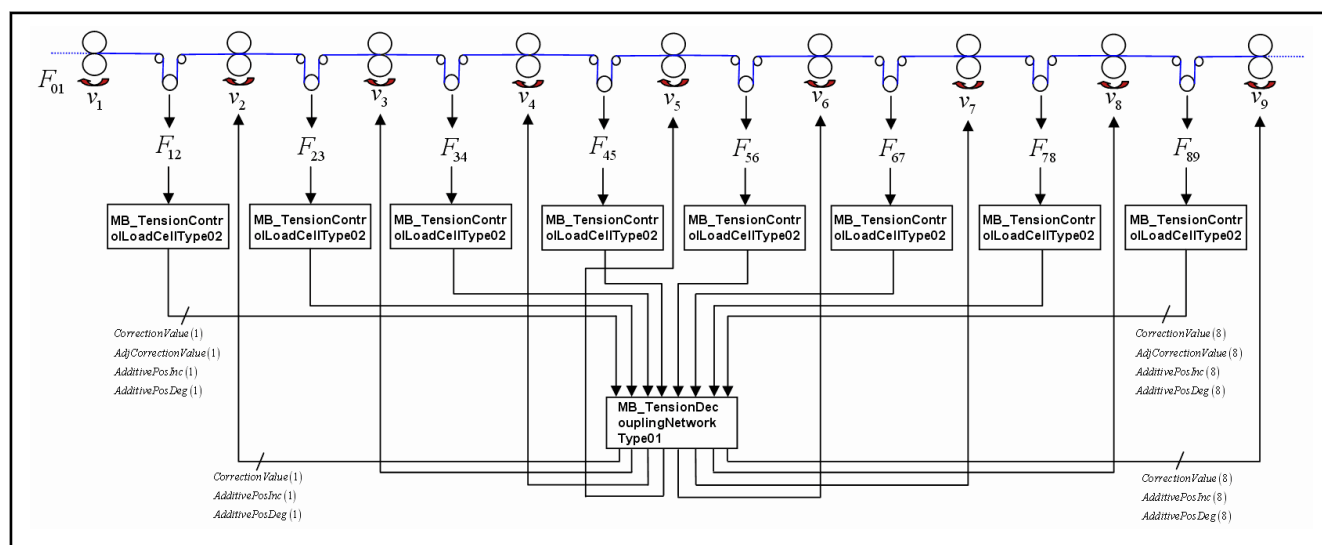


Abb. 7-33: Entkopplung der Bahnzugkräfte durch Vorsteuerung ausschließlich in Bahnaufrichtung (führende Klemmstellen werden nicht für die Regelung und Entkopplung benötigt)

Wird diese Variante für ein System mit weniger als acht zu regelnden Bahnabschnitten gewählt, so können einfach die hinteren in Bahnlaufrichtung befindlichen Bahnzugkraftregler unberücksichtigt bleiben.

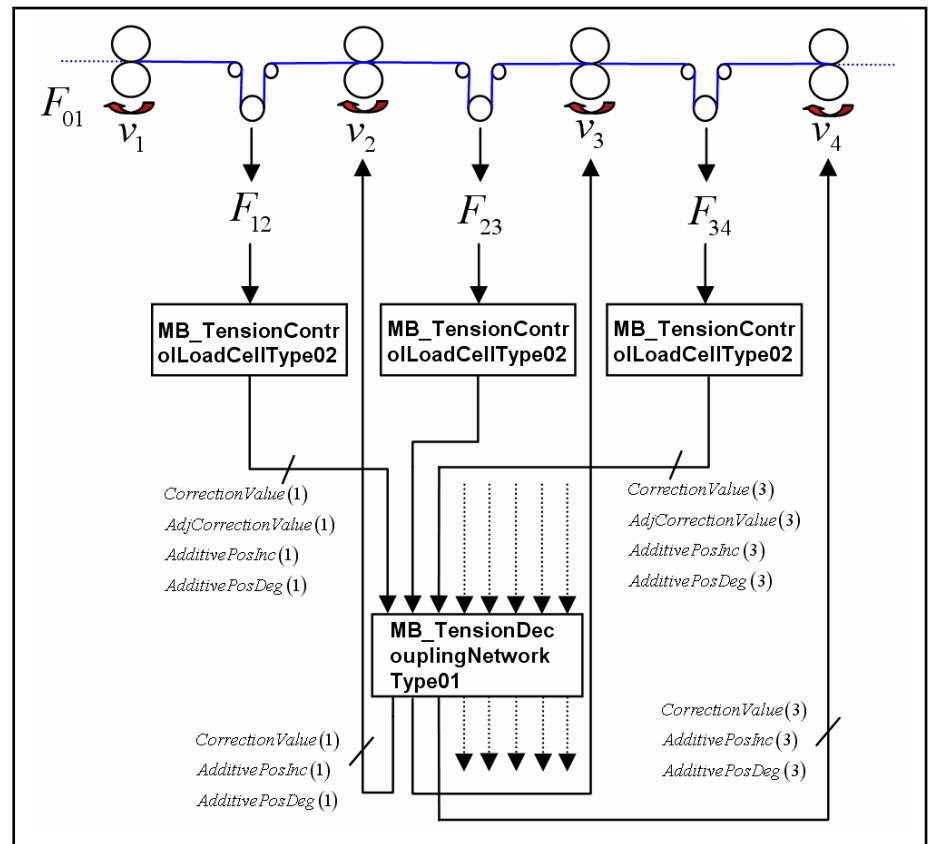


Abb. 7-34: Entkopplung der Bahnzugkräfte durch Vorsteuerung ausschließlich in Bahnlaufrichtung (Darstellung bei drei zu regelnden Bahnabschnitten)

Eine weitere Möglichkeit ist nachfolgend dargestellt. Hierbei wird die letzte Achse des Systems für die Regelung der Bahnzugkräfte ausgeschlossen (UnControlledAxis:=9).

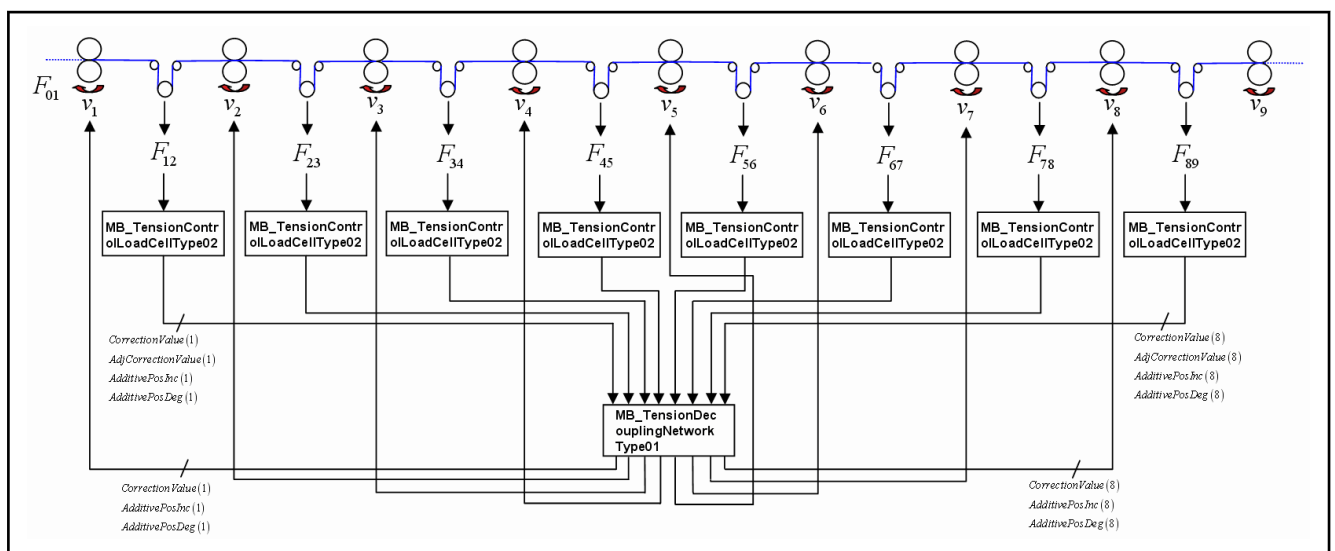


Abb. 7-35: Entkopplung der Bahnzugkräfte durch Vorsteuerung ausschließlich gegen die Bahnlaufrichtung (nachfolgende Klemmstellen werden nicht für die Regelung und Entkopplung benötigt)

Wird diese Variante für ein System mit weniger als acht zu regelnden Bahnabschnitten gewählt, so können einfach die führenden gegen die Bahnlauf- richtung befindlichen Bahnzugkraftregler unberücksichtigt bleiben.

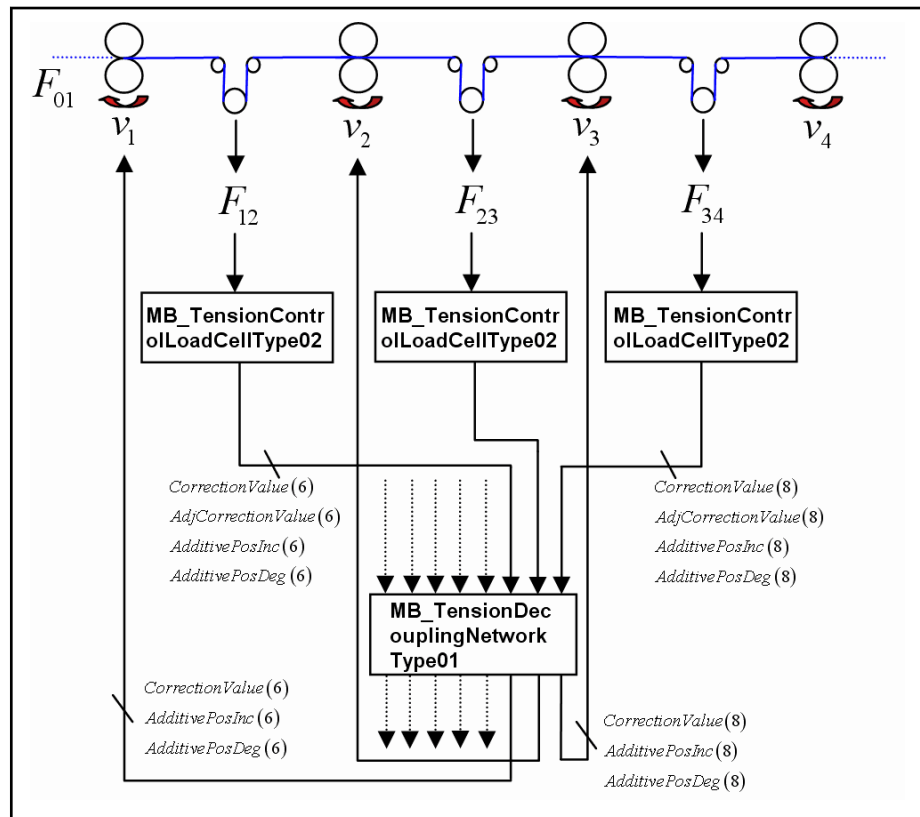


Abb. 7-36: Entkopplung der Bahnzugkräfte durch Vorsteuerung ausschließlich gegen die Bahnlaufrichtung (Darstellung bei drei zu regelnden Bahnabschnitten)

Da jede beliebige Achse des Systems für die Regelung der Bahnzugkräfte ausgeschlossen werden kann, besteht weiter die Möglichkeit, eine mittlere Achse auszuschließen. Dies entspricht den Varianten "UnControlledAxis":=2-8 und ist beispielhaft für die fünfte Achse (UnControlledAxis:=5) dargestellt.

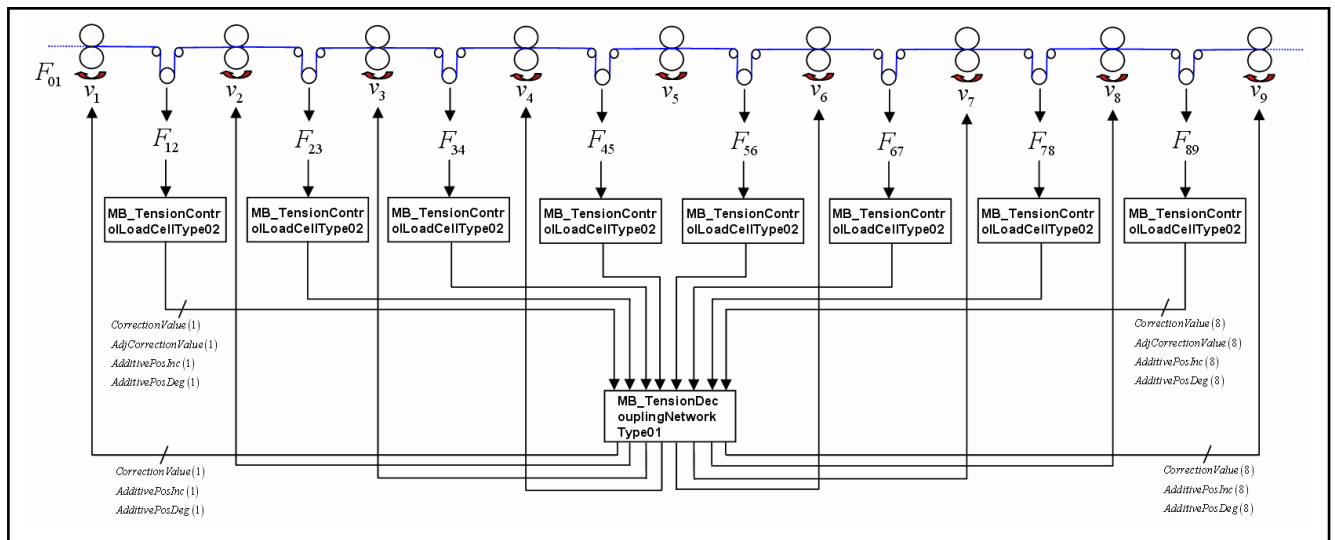


Abb. 7-37: Entkopplung der Bahnzugkräfte durch Vorsteuerung in und gegen die Bahnaufrichtung (eine mittlere Klemmstellen, in Abb. "v5", wird nicht für die Regelung und Entkopplung benötigt)

Wird diese Variante für ein System mit weniger als acht zu regelnden Bahnabschnitten gewählt, so können entsprechend den beiden vorhergehenden Varianten einfach die führenden und/oder die in Bahnaufrichtung folgenden Bahnzugkraftregler unberücksichtigt bleiben. Wählt man zweckmäßig die nicht zu regelnde Achse entsprechend des realen Systems in Bahnaufrichtung, so ist es lediglich erforderlich, die folgenden real nicht vorhandenen Bahnzugkraftregler unberücksichtigt zu lassen.

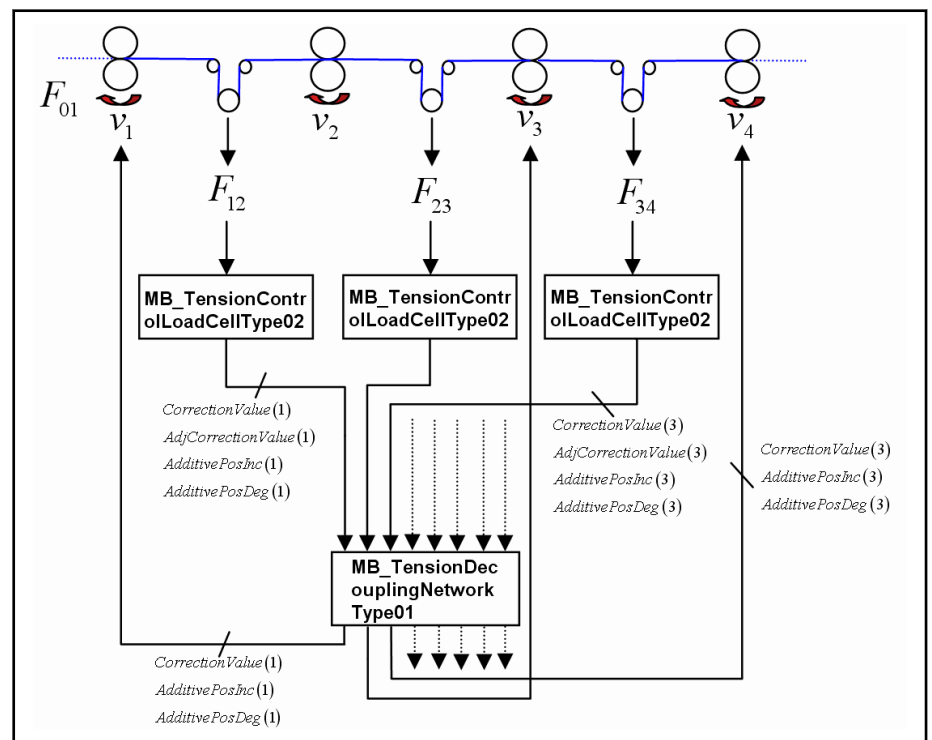


Abb. 7-38: Entkopplung der Bahnzugkräfte durch Vorsteuerung in und gegen die Bahnaufrichtung (Darstellung bei drei zu regelnden Bahnabschnitten)

Neben diesen drei Varianten der Entkopplung, bei denen jeweils eine Achse nicht zur Regelung und Entkopplung benötigt wird, kann das Entkopplungsnetzwerk weiter nahezu beliebig eingesetzt werden. Beispielhaft sei der Fall

einer Anlage mit bis zu 15 zu regelnden Bahnabschnitten und der Forderung nach einem minimalen Stellaufwand erwähnt.

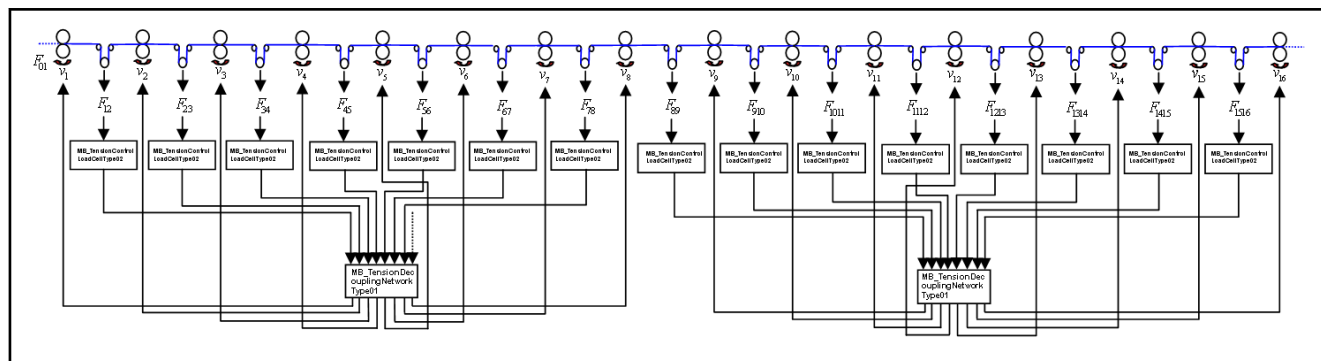


Abb. 7-39: Expansion des Entkopplungsnetzwerks auf 15 zu regelnde Bahnabschnitte bei zusätzlicher Forderung nach minimalem Stellaufwand

Es ist in dieser Darstellung zu erkennen, dass zwei Funktionsbausteine MB_TensionDecouplingNetworkType01 eingesetzt werden, und eine Stellsignaländerung eines Bahnzugkraftreglers maximal acht ((geregelter Abschnitte +1)/2) Stellbewegungen verursacht. Weiter ist zu erkennen, dass in dem führenden Entkopplungsnetzwerk eine Upstream-Entkopplung (UnControlledAxis:=9) und dem hinteren Entkopplungsnetzwerk eine Downstream-Entkopplung (UnControlledAxis:=1) eingesetzt wird. Lediglich der letzte Bahnzugkraftregler des führenden Entkopplungsnetzwerks wurde freigelassen. Da der letzte Bahnabschnitt des führenden Entkopplungsnetzwerks komplett entkoppelt von allen davorliegenden Bahnabschnitten ist, kann diese Kombination der beiden Entkopplungsnetzwerke eingesetzt werden, ohne dass eine Querkommunikation bzw. Quervorsteuerung benötigt wird. Setzt man zwei oder mehr Entkopplungsnetzwerke ein, so kann man diese in zahllosen Varianten miteinander kombinieren. Ist für eine spezielle Anlage eine weitere Variante mit mehreren Entkopplungsnetzwerken zweckmäßig, so können die Ein- und Ausgänge der Entkopplungsnetzwerke extern kombiniert werden. Je nach gewünschtem Verhalten kann es hierbei außerdem nötig sein, das zyklische Schreiben (DisableCyclicWrites:=TRUE) zu unterbinden. Beispielfür diese zahllosen Varianten sind Anlagen mit mehr als acht zu regelnden Bahnabschnitten und der Forderung, dass die erste, letzte oder eine beliebige Achse nicht geregelt und/oder vorgesteuert werden darf. Hierbei ist es dann sinnvoll, ein oder mehrere Ausgänge eines Entkopplungsnetzwerks mit einem oder mehreren Eingängen eines anderen Entkopplungsnetzwerks zu kombinieren. Diese Umsetzungen setzen jedoch detaillierte Kenntnisse voraus.

- Inbetriebnahme** Ist "DisableCyclicWrites" inaktiv, müssen die beiden Parameter P-0-0694 (Getriebefeinabgleich) und P-0-0692 (additive Leitachsisposition) im zyklischen Sollwertkanal der Folgeachse aufgelegt werden. Bei steigender Flanke an "Enable" werden zyklische Datenkanäle überprüft. Wird einer der beiden Parameter nicht gefunden, erfolgt eine entsprechende Fehlermeldung an den Fehlerausgängen des Funktionsbausteins. Ist "DisableCyclicWrites" aktiv, wird die Konfiguration der zyklischen Sollwertdatenkanäle nicht überprüft.
- Enable** Der Eingang „Enable“ aktiviert den Funktionsbaustein.



Beim Einschalten des Funktionsbausteins werden sämtliche Reglerstellsignale und Achspositionen ausgewertet, um Positionssprünge an den Achsen zu vermeiden.

Infolgedessen sollte dieser Enable-Eingang erst aktiviert werden, wenn sämtliche Bahnzugkraftregler einen aktiven "InOperation"-Ausgang erreicht haben.

StandstillEnable	Durch Setzen des Eingangs „StandstillEnable“ wird die Zugkraftregelung im Stillstand aktiviert. Somit werden die Eingänge „AdditivePosInc“ und „AdditivePosDeg“ ausgewertet und über diese die gesamten Stellsignale "AdditivePosInc" und "AdditivePosDeg" der einzelnen Achsen berechnet und ausgegeben.
DisableCyclicWrites	Der Eingang "DisableCyclicWrites" bestimmt, ob die vom Funktionsbaustein berechneten Werte direkt über den zyklischen Kanal geschrieben werden (FALSE) oder nur am Baustein Ausgang verfügbar sind (TRUE). Mit "DisableCyclicWrites":=TRUE ist es somit möglich, mehrere Entkopplungsnetzwerke miteinander zu kombinieren, indem man die Ausgänge mehrerer Entkopplungswerke manuell verknüpft.
UnControlledAxis	Der Eingang "UnControlledAxis" bestimmt die Entkopplungsvariante sowie die Position der nicht zu regelnden und vorzusteuern Achse. Standardmäßig wird dieser Eingang auf 1 gesetzt, was für das Entkopplungsnetzwerk bedeutet, dass ausschließlich in Bahnlaufichtung vorgesteuert werden soll (Downstream-Entkopplung). Wird dieser Eingang auf 9 gesetzt, so wird die letzte Achse des Systems für die Regelung und Entkopplung ausgeschlossen (Upstream-Entkopplung). Eine Vorsteuerung wird in diesem Falle ausschließlich gegen die Bahnlaufichtung erfolgen. Des Weiteren kann eine beliebige Achse des Systems für die Bahnzugkraftregelung und Entkopplung ausgeschlossen werden (UnControlledAxis:=2-8).
Fehlerbehandlung	Der Funktionsbaustein erzeugt in Additional1/Additional2 für die Tabelle F_RELATED_TABLE , 16#0170, folgende Fehlermeldungen:

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR, 16#0004	16#00000A00	16#00000001	Der Parameter P-0-0694 ist nicht im optional zyklischen Telegramm der vorzusteuern Achse parametrier
ACCESS_ERROR, 16#0004	16#00000A00	16#00000002	Der Parameter P-0-0692 ist nicht im optional zyklischen Telegramm der vorzusteuern Achse parametrier
INPUT_RANGE_ERROR, 16#0006	16#00000A01	16#00000007	Der Eingang „UnControlledAxis“ ist außerhalb des gültigen Bereichs
DEVICE_ERROR, 16#0008	16#00000A02	16#00000007	Bei einem Antrieb ist keine Leistung zugeschaltet. Eine Zugkraftregelung ist daher nicht möglich
DEVICE_ERROR, 16#0008	16#00000A02	16#00000008	Ein Antrieb befindet sich nicht in einer synchronen Betriebsart

Tab. 7-21: Fehlercodes von MB_TensionDecouplingNetworkType01

7.9.2 MB_TENSION_DECOUPLING_CONTROLLER_VALUE

Die Struktur MB_TENSION_DECOUPLING_CONTROLLER_VALUE enthält die Reglerausgänge (CorrectionValue, AdjCorrectionValue, AdditivePosInc und AdditivePosDeg) des Entkopplungsnetzwerkes.

Die Struktur wird vom Funktionsbaustein "MB_TensionDecouplingNetworkType01" verwendet.

Name	Typ	Min.-Wert	Max.-Wert	Beschreibung
ControllerCorrectionValue [1]	REAL			CorrectionValue des ersten Bahnzugkraftreglers
ControllerCorrectionValue [2]	REAL			CorrectionValue des zweiten Bahnzugkraftreglers

Name	Typ	Min.-Wert	Max.-Wert	Beschreibung
ControllerCorrectionValue [3]	REAL			CorrectionValue des dritten Bahnzugkraftreglers
ControllerCorrectionValue [4]	REAL			CorrectionValue des vierten Bahnzugkraftreglers
ControllerCorrectionValue [5]	REAL			CorrectionValue des fünften Bahnzugkraftreglers
ControllerCorrectionValue [6]	REAL			CorrectionValue des sechsten Bahnzugkraftreglers
ControllerCorrectionValue [7]	REAL			CorrectionValue des siebten Bahnzugkraftreglers
ControllerCorrectionValue [8]	REAL			CorrectionValue des achten Bahnzugkraftreglers
ControllerAdjCorrectionValue [1]	REAL			AdjCorrectionValue des ersten Bahnzugkraftreglers
ControllerAdjCorrectionValue [2]	REAL			AdjCorrectionValue des zweiten Bahnzugkraftreglers
ControllerAdjCorrectionValue [3]	REAL			AdjCorrectionValue des dritten Bahnzugkraftreglers
ControllerAdjCorrectionValue [4]	REAL			AdjCorrectionValue des vierten Bahnzugkraftreglers
ControllerAdjCorrectionValue [5]	REAL			AdjCorrectionValue des fünften Bahnzugkraftreglers
ControllerAdjCorrectionValue [6]	REAL			AdjCorrectionValue des sechsten Bahnzugkraftreglers
ControllerAdjCorrectionValue [7]	REAL			AdjCorrectionValue des siebten Bahnzugkraftreglers
ControllerAdjCorrectionValue [8]	REAL			AdjCorrectionValue des achten Bahnzugkraftreglers
ControllerAdditivePosInc [1]	DINT			AdditivePosInc des ersten Bahnzugkraftreglers
ControllerAdditivePosInc [2]	DINT			AdditivePosInc des zweiten Bahnzugkraftreglers
ControllerAdditivePosInc [3]	DINT			AdditivePosInc des dritten Bahnzugkraftreglers
ControllerAdditivePosInc [4]	DINT			AdditivePosInc des vierten Bahnzugkraftreglers
ControllerAdditivePosInc [5]	DINT			AdditivePosInc des fünften Bahnzugkraftreglers
ControllerAdditivePosInc [6]	DINT			AdditivePosInc des sechsten Bahnzugkraftreglers
ControllerAdditivePosInc [7]	DINT			AdditivePosInc des siebten Bahnzugkraftreglers
ControllerAdditivePosInc [8]	DINT			AdditivePosInc des achten Bahnzugkraftreglers
ControllerAdditivePosDeg [1]	REAL			AdditivePosDeg des ersten Bahnzugkraftreglers
ControllerAdditivePosDeg [2]	REAL			AdditivePosDeg des zweiten Bahnzugkraftreglers
ControllerAdditivePosDeg [3]	REAL			AdditivePosDeg des dritten Bahnzugkraftreglers
ControllerAdditivePosDeg [4]	REAL			AdditivePosDeg des vierten Bahnzugkraftreglers
ControllerAdditivePosDeg [5]	REAL			AdditivePosDeg des fünften Bahnzugkraftreglers
ControllerAdditivePosDeg [6]	REAL			AdditivePosDeg des sechsten Bahnzugkraftreglers
ControllerAdditivePosDeg [7]	REAL			AdditivePosDeg des siebten Bahnzugkraftreglers
ControllerAdditivePosDeg [8]	REAL			AdditivePosDeg des achten Bahnzugkraftreglers

Tab. 7-22: Strukturbeschreibung MB_TENSION_DECOUPLING_CONTROL-
LER_VALUE

7.9.3 MB_TENSION_DECOUPLING_AXIS_VALUE

Die Struktur MB_TENSION_DECOUPLING_AXIS_VALUE enthält die Werte der Achseingänge (CorrectionValue, AdditivePosInc und AdditivePosDeg).

Die Struktur wird vom Funktionsbaustein "MB_TensionDecouplingNetwork-Type01" verwendet.

Name	Typ	Min.-Wert	Max.-Wert	Beschreibung
AxisCorrectionValue [1]	REAL			CorrectionValue der ersten Achse
AxisCorrectionValue [2]	REAL			CorrectionValue der zweiten Achse
AxisCorrectionValue [3]	REAL			CorrectionValue der dritten Achse
AxisCorrectionValue [4]	REAL			CorrectionValue der vierten Achse
AxisCorrectionValue [5]	REAL			CorrectionValue der fünften Achse
AxisCorrectionValue [6]	REAL			CorrectionValue der sechsten Achse
AxisCorrectionValue [7]	REAL			CorrectionValue der siebten Achse
AxisCorrectionValue [8]	REAL			CorrectionValue der achten Achse
AxisAdditivePosInc [1]	DINT			AdditivePosInc der ersten Achse
AxisAdditivePosInc [2]	DINT			AdditivePosInc der zweiten Achse
AxisAdditivePosInc [3]	DINT			AdditivePosInc der dritten Achse
AxisAdditivePosInc [4]	DINT			AdditivePosInc der vierten Achse
AxisAdditivePosInc [5]	DINT			AdditivePosInc der fünften Achse
AxisAdditivePosInc [6]	DINT			AdditivePosInc der sechsten Achse
AxisAdditivePosInc [7]	DINT			AdditivePosInc der siebten Achse
AxisAdditivePosInc [8]	DINT			AdditivePosInc der achten Achse
AxisAdditivePosDeg [1]	REAL			AdditivePosDeg der ersten Achse
AxisAdditivePosDeg [2]	REAL			AdditivePosDeg der zweiten Achse
AxisAdditivePosDeg [3]	REAL			AdditivePosDeg der dritten Achse
AxisAdditivePosDeg [4]	REAL			AdditivePosDeg der vierten Achse
AxisAdditivePosDeg [5]	REAL			AdditivePosDeg der fünften Achse
AxisAdditivePosDeg [6]	REAL			AdditivePosDeg der sechsten Achse
AxisAdditivePosDeg [7]	REAL			AdditivePosDeg der siebten Achse
AxisAdditivePosDeg [8]	REAL			AdditivePosDeg der achten Achse

Tab. 7-23: Strukturbeschreibung MB_TENSION_DECOUPLING_AXIS_VALUE

8 RMB_TechWinder.library // MX_TechWinder

8.1 Einführung und Übersicht

In den Bibliotheken RMB_TechWinder (MLC) und MX_TechWinder.lib (MLD) sind verschiedene Funktionsbausteine zusammengefasst, die Funktionalitäten für das Auf- und Abwickeln von Stoffbahnen (Wicklern) bereitstellen.



Die Bibliothek RMB_TechWinder (MLC) benötigt das **TEC (Technologie) Funktionspaket** der Steuerung.

Bibliotheksabhängigkeiten

Steuerung MLC - RMB_TechWinder

Die benötigten Bibliotheken werden automatisch mit eingebunden.

Steuerung MLD - MX_TechWinder

IndraLogic 1.x (Antriebsfirmware < MPx18):

Folgende Bibliotheken werden benötigt:

- MX_Technology0x (MPX0x-Antriebsfirmware)
- RIL_LoopControl

IndraLogic 2G (Antriebsfirmware ab MPX20):

Die benötigten Bibliotheken werden automatisch mit eingebunden.

Funktionspakete für MLD

Für MLD wird für die Bausteine der MX_TechWinder-Bibliothek das Antriebsfunktionspaket MA "MLD Advanced" benötigt.

Dieses Funktionspaket ist mit der entsprechenden Antriebsfirmware separat zu bestellen.



Die Bibliothek MX_TechWinder unterscheidet sich im Funktionsumfang von der Bibliothek RMB_TechWinder für MLC. Mit Hilfe der Zielsystemtabellen für die jeweiligen Funktionsbausteine kann überprüft werden, welche Funktionsbausteine für welches Target zur Verfügung stehen.

Umgesetzt werden können mit Hilfe dieser Funktionsbausteine die Reglerfunktionen eines

- Tänzerlagereglers
- Wickelrechners mit Tänzer
- Wickelrechners mit Zugkraftregelung
- Wickelrechners mit Zugkraftsteuerung

Im Folgenden werden die jeweils zur Umsetzung dieser Funktionalitäten erforderlichen Funktionsbausteine beschrieben.

In den folgenden Tabellen wird eine Übersicht über alle dokumentierten Funktionsbausteine und ihre jeweilige Anwendung gegeben.

Vorab verschaltete komplette Wickler-Funktionsbausteine

Funktionsbaustein	Beschreibung
MB_WinderDancerCtrlType01, Seite 525	Funktionsbaustein für Wickler mit Tänzer beinhaltet unter anderem: MB_DancerControlType03, Seite 495 MB_DiameterCalculatorType03, Seite 509 MB_DiameterMeasurementType01, Seite 519 (Die Stellgröße des Tänzerlagereglers wird mit der Formatlänge der Referenzachse und dem aktuellen Wickeldurchmesser normiert.)
MB_WinderDancerCtrlType02, Seite 532	erweiterter Funktionsbaustein für Wickler mit Tänzer beinhaltet unter anderem: MB_DancerControlType04, Seite 501 MB_DiameterCalculatorType03, Seite 509 MB_DiameterMeasurementType01, Seite 519 1. Normierung der Tänzerstellgröße unabhängig von der Formatlänge der Referenzachse 2. Zusätzliche Diagnoseausgänge für Tänzerregelung
MB_WinderTensionCtrlType02, Seite 542	Funktionsbaustein für Wickler mit Zugkraftregelung bzw. Zugkraftsteuerung (sensorlos) beinhaltet unter anderem: MB_DiameterCalculatorType03, Seite 509 MB_DiameterMeasurementType01, Seite 519
MB_WinderTensionCtrlSpeedType01, Seite 560	Funktionsbaustein für Wickler mit Zugkraftregelung und Drehzahlkorrektur beinhaltet unter anderem: MB_DiameterCalculatorType03, Seite 509 MB_DiameterMeasurementType01, Seite 519

Tab. 8-1: Vorab verschaltete Wickler-FB



Die vorab verschalteten Funktionsbausteine beinhalten alle Funktionen der einzelnen Bausteine und nehmen dem Anwender die richtige Zusammenschaltung der einzelnen Funktionsbausteine ab.

Detailinformationen zur Inbetriebnahme sind in "MLC 14VRS Anwendung Wicklerfunktionen DOK-MLC***-TF*WIND*V14-APRS-DE-P, R911341512" im Medienverzeichnis enthalten.

Funktionsbausteine für einzelne Funktionalitäten

Funktionsbaustein	Beschreibung
MB_DancerControlType03, Seite 495	Funktionsbaustein zur Berechnung einer Tänzerlageregelung für eine zugkraftgeregelter Achse mit Tänzer bzw. Wickelachse (Die Stellgröße des Tänzerlagereglers wird mit der Formatlänge der Referenzachse und dem Referenzdurchmesser normiert.)
MB_DancerControlType04, Seite 501	Erweiterter Funktionsbaustein zur Berechnung einer Tänzerlageregelung für eine zugkraftgeregelter Achse mit Tänzer bzw. Wickelachse 1. Normierung der Tänzerstellgröße unabhängig von der Formatlänge der Referenzachse 2. Zusätzliche Diagnoseausgänge für Tänzerregelung
MB_DiameterCalculatorType03, Seite 509	Funktionsbaustein zur Berechnung des aktuellen Durchmessers für Wickelachsen mit Geschwindigkeitsnachführung
MB_DiameterMeasurementType01, Seite 519	Funktionsbaustein zum Einlesen eines gemessenen Durchmessers für Wickelachsen mit Geschwindigkeitsnachführung

Tab. 8-2: Einzelfunktionsbausteine

Funktionsbausteine für zusätzliche Wicklerfunktionen

Funktionsbaustein	Beschreibung
MB_CalcInertialLimitType02, Seite 577	Funktionsbaustein zur Berechnung des maximalen Trägheitsmomentes des Wickelgutes auf einem Wickel
MB_UnwindMaterialType02, Seite 579	Funktionsbaustein zur Berechnung der Restlänge, Restlaufzeit, ... der Stoffbahn für einen Abwickler
MB_WindTaperProfileType01, Seite 585	Funktionsbaustein zur Berechnung eines Wickelhärteprofils für einen Aufwickler mit Zugkraftregelung bzw. -steuerung
MB_WindSpeedControlAdaptionType01, Seite 588	Funktionsbaustein zur automatischen Anpassung der K_p -Verstärkung des Drehzahlreglers
MB_TraverseControlType01, Seite 592	Funktionsbaustein steuert eine translatorische Verlegerachse, die im Zusammenhang mit einem Wickler (rotatorische Achse), die Verlegung von unterschiedlichen Materialien (z. B. Draht) ermöglicht.
MB_DancerCalcControlParamType01, Seite 608	Funktionsbaustein berechnet aus physikalischen Daten des Tänzersystems (Durchmesser der Zugwalze bzw. Hülsendurchmesser beim Wickler sowie Verzögerungszeit des Systems) die Reglerparameter des Tänzerlagereglers

Tab. 8-3: Funktionsbausteine für zusätzliche Funktionalitäten

8.2 Tänzerlageregelung

8.2.1 Einführung und Übersicht

Der Tänzerlageregler wird z. B. an zugkraftgeregelter Achsen oder Wicklern mit Tänzern eingesetzt. Der Tänzerlageregler regelt die Lage eines Tänzers. Die Zugkraft wird durch eine Kraft vorgegeben, die am frei beweglichen Teil des Tänzers angreift. Diese Kraft kann durch eine Gewichtskraft, eine Feder

oder durch Pneumatik aufgebracht werden. Der Istwert der Tänzerlage wird z. B. über einen Analogkanal erfasst, an die Steuerung übergeben und im Prozessregler verarbeitet.

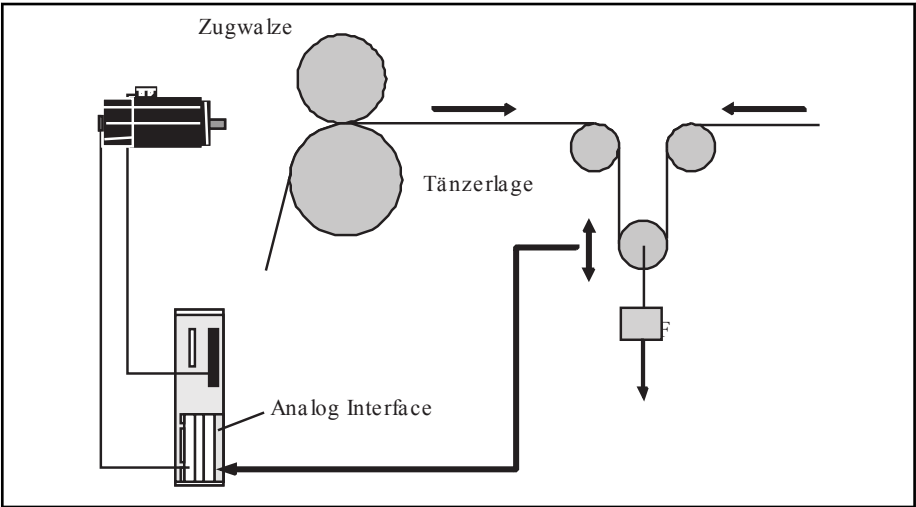


Abb. 8-1: Prinzip der Tänzerlageregelung

Der Tänzerlageregler ist im Funktionsbaustein [MB_DancerControlType03](#), Seite 495, implementiert und als PID-Regler ausgeführt. Durch Setzen der Reglerparameter können die einzelnen Reglerfunktionen (P-Anteil: Proportionalanteil, I-Anteil: Integralanteil, D-Anteil: differentieller Anteil) angesteuert werden. Wird ein Parameter zu Null gesetzt, ist der entsprechende Regleranteil ebenfalls nicht wirksam.

Nachfolgende Tabelle zeigt die möglichen und erlaubten Reglerkonfigurationen.

Nr.	P-Anteil	I-Anteil	D-Anteil	erlaubt	Bemerkung
1	0	0	0	ja	Regler unwirksam, Ausgang = 0.0
2	0	0	1	nein	D-Regler nicht definiert, Fehler
3	0	1	0	ja	reiner I-Regler
4	0	1	1	nein	DI-Regler nicht definiert, Fehler
5	1	0	0	ja	reiner P-Regler
6	1	0	1	ja	PD-Regler
7	1	1	0	ja	PI-Regler
8	1	1	1	ja	PID-Regler

Tab. 8-4: Erlaubte und nicht erlaubte Reglerkonfigurationen (1 entspricht Anteil gesetzt (Wert ≥ 0.0), 0 entspricht Anteil nicht gesetzt (Wert = 0.0))

Funktionsdiagramm des Tänzerlagereglers

Anhand des folgenden Funktionsdiagramms wird die Funktionsweise des Tänzerlagereglers schematisch erläutert.

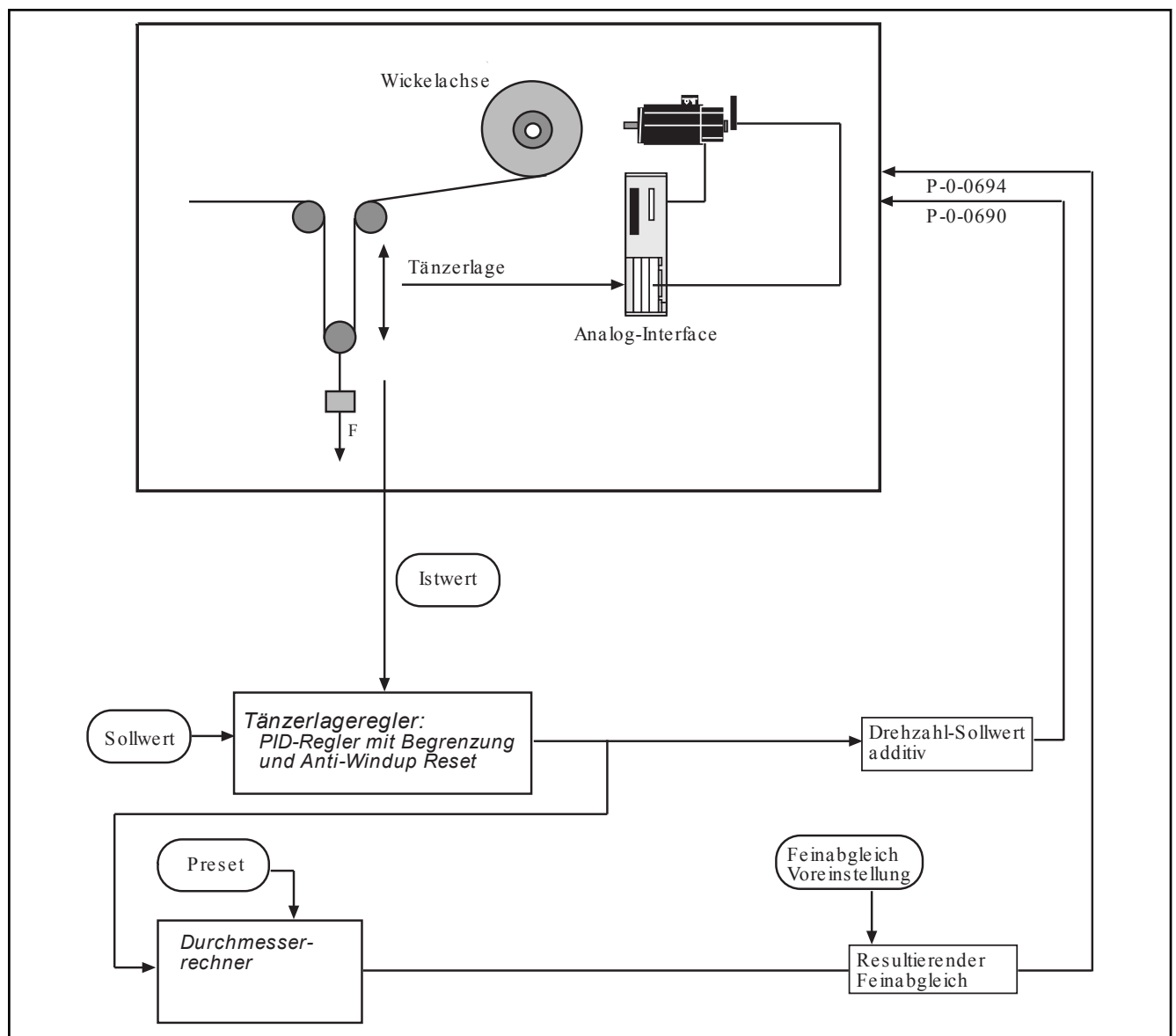


Abb. 8-2: Funktionsdiagramm des Tänzerlageregers

8.2.2 MB_DancerControlType03

Kurzbeschreibung

Der Funktionsbaustein berechnet zyklisch eine Tänzerlageregelung für eine Zugwalze mit Tänzer oder für eine Wickelachse mit Tänzer. Zyklische Ausgangsgröße des Tänzerlageregers ist der Drehzahlversatz für den jeweiligen Antrieb einer Zugwalze oder Wickelachse. Des Weiteren wird die aktuelle Regelabweichung des Tänzerlageregers ausgegeben.

Der Ausgang "AddVelocityDancer" des Funktionsbausteins wirkt auf den Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" (Interpolation im Antrieb) bzw. auf den Parameter "A-0-2615 ELS, Geschwindigkeits-Sollwert additiv, Prozessregler" (Interpolation in der Steuerung). Ist dies nicht gewünscht, kann dies mit dem Eingang "DisableCyclicW-rites" verhindert werden. Es wird dann nur auf den Ausgang "AddVelocityDancer" geschrieben.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechWinder
MLD mit MPx07/08/17VRS mit MA Funktionspaket	MX_TechWinder.lib

Tab. 8-5: Referenztable MB_DancerControlType03

Schnittstellenbeschreibung

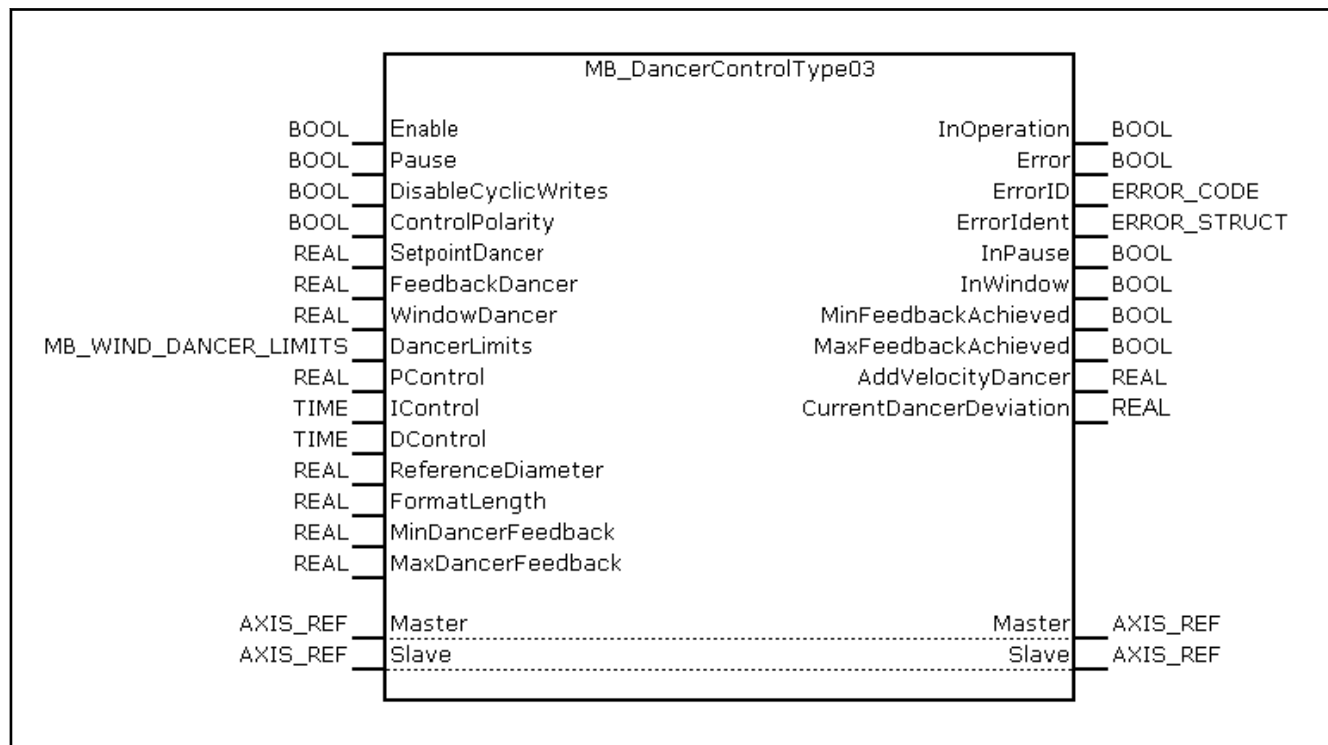


Abb. 8-3: Funktionsbaustein MB_DancerControlType03

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	Pause	BOOL	Tänzerregler pausiert, keine Geschwindigkeitsanpassung (pegelgesteuert)
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	ControlPolarity	BOOL	Polarität der Tänzerregelung wird invertiert (Übernahme bei pos. Flanke von Enable)
	SetpointDancer	REAL	Sollwert für die Tänzerregelung
	FeedbackDancer	REAL	Istwert der Tänzerlage
	WindowDancer	REAL	Fenster zur Überwachung des Tänzerlageistwertes [%], bezogen auf den Tänzerlagesollwert. Entfernt sich der Tänzer vom Lagesollwert um mehr als den angegebenen Wert, meldet dies der Baustein durch Rücksetzen des Ausgangs "InWindow"

E/A-Typ	Name	Typ	Kommentar
	DancerLimits	MB_WIND_DANCER_LIMITS	Struktur für die automatische Stellgrößenbegrenzung des Tänzers MB_WIND_DANCER_LIMITS, Seite 615
	PControl	REAL	Wert für die Kp-Verstärkung der Tänzerregelung
	IControl	TIME	Wert für die Nachstellzeit der Tänzerregelung
	DControl	TIME	Wert für die Vorhaltezeit der Tänzerregelung
	ReferenceDiameter	REAL	Bezugsdurchmesser, (Wert dient zur Normierung von PControl und der Stellgrößenbegrenzung DancerLimits, d. h. diese Werte werden intern angepasst) Bei reinem Tänzerlagereger: Durchmesser der Zugwalze [mm] Bei Wickler mit Tänzer: Aktueller Wickeldurchmesser [mm] vom Durchmesserrechner Baustein zuweisen
	FormatLength	REAL	Formatlänge [mm], (Wert dient zur Normierung von PControl) (1 Umdrehung der Referenzachse abgewickelt)
	MinDancerFeedback	REAL	Untere Schwelle für den Tänzeristwert (nur für Anzeige mit binärem Ausgang)
	MaxDancerFeedback	REAL	Obere Schwelle für den Tänzeristwert (nur für Anzeige mit binärem Ausgang)
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose Fehlerbehandlung, Seite 500
	InPause	BOOL	Tänzerregler pausiert
	InWindow	BOOL	Tänzeristwert innerhalb des Tänzerfensters
	MinFeedbackAchieved	BOOL	Istwert der Tänzerlage ist unterhalb des Eingang MinDancerFeedback
	MaxFeedbackAchieved	BOOL	Istwert der Tänzerlage ist oberhalb des Eingang MaxDancerFeedback
	AddVelocityDancer	REAL	Drehzahloffset für den Wickel- oder Zugwalzenantrieb [U/min]
	CurrentDancerDeviation	REAL	Aktuelle Regelabweichung des Tänzerreglers
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (zugkraftgeregelte Achse)

Tab. 8-6: Schnittstellenvariablen Funktionsbaustein MB_DancerControlType03

Min-/Max und Standardwerte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Standardwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
ControlPolarity	BOOL			FALSE	Steigende Flanke an Enable
SetpointDancer	REAL	0.0	n. def.	0.0	Kontinuierlich
FeedbackDancer	REAL	n. def.	n. def.	0.0	Kontinuierlich
WindowDancer	REAL	0.0	n. def.	0.0	Kontinuierlich
DancerLimits	STRUCT				
PControl	REAL	0.0	n. def.	0.0	Kontinuierlich
IControl	TIME	0s	n. def.	0s	Kontinuierlich
DControl	TIME	0s	n. def.	0s	Kontinuierlich
ReferenceDiameter	REAL	0.0	n. def.	100	Kontinuierlich
FormatLength	REAL	0.0	n. def.	0.0	Steigende Flanke an "Enable"
MinDancerFeedback	REAL	0.0	MaxDancerFeedback	0.0	Kontinuierlich
MaxDancerFeedback	REAL	MinDancerFeedback	n. def.	0.0	Kontinuierlich

Tab. 8-7: Min-/Max- und Default-Werte der Eingänge MB_DancerControl-Type03

Stellgrößenbegrenzung

Die Stellgrößenbegrenzung des Tänzerlagereglers wird in Abhängigkeit der Referenzachsgeschwindigkeit und dem Durchmesser Verhältnis ($\text{Formatlänge} / \pi * \text{ReferenceDiameter}$) dynamisch angepasst. D. h. bei höherer Geschwindigkeit ist die Begrenzung der additiven Geschwindigkeit vom Tänzerlageregler automatisch größer als bei niedriger Geschwindigkeit. Die Berücksichtigung des Durchmesser Verhältnisses führt dazu, dass die Begrenzung bei Wickelanwendungen automatisch einem sich ändernden "ReferenceDiameter" anpasst (mit steigendem Durchmesser wird die Begrenzung automatisch kleiner).

Signal-Zeit-Diagramm

Im folgenden Signal-Zeit-Diagramm sind ausgewählte Signale des Funktionsbausteins dargestellt.

Zusammenhang der Signale

Signaleingang	Signalausgang	Beschreibung
Enable	InOperation	Baustein in Betrieb nehmen
	Error	Betriebsbereitschaft wurde unterbrochen
Pause	InPause	Internen Reglerprozess anhalten

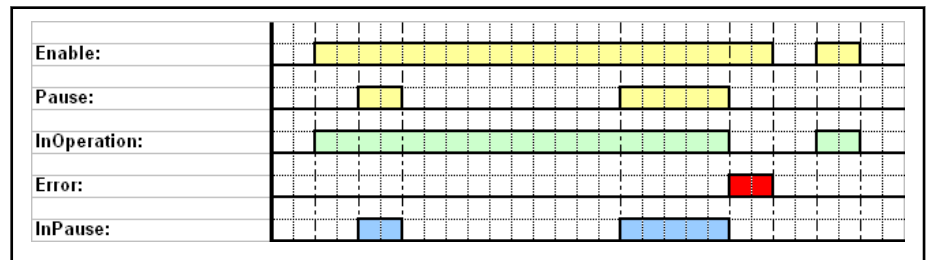


Abb. 8-4: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_DancerControlType03

Funktionsbeschreibung

Der Funktionsbaustein berechnet zyklisch eine Tänzerlageregelung für eine Zugwalze mit Tänzer oder für eine Wickelachse mit Tänzer. Es wird der Funktionsbaustein IL_PIDType01 aus der Bibliothek RIL_LoopControl als Regler verwendet. Der Regler ist als PID-Regler ausgeführt. Zyklische Ausgangsgrößen des Tänzerlagereglers sind der Drehzahlversatz des jeweiligen Antriebs für die Zugwalze oder Wickelachse, sowie die aktuelle Regelabweichung des Tänzerlagereglers. Der Ausgang "AddVelocityDancer" des Funktionsbausteins sollte zur Tänzerlageregelung auf den Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" wirken (externe Verschaltung). Die aktuelle Regelabweichung "CurrentDancerDeviation" steht zu Diagnosezwecken zur Verfügung. Die Proportionalverstärkung (Faktor K_P) und die Low- und High-Werte für die Stellgrößenbegrenzung "DancerLimits" werden automatisch mit dem Eingang "ReferenceDiameter", bezogen auf die Formatlänge "FormatLength", angepasst. Für den Fall einer Tänzerlageregelung für eine Zugwalze ist der Durchmesser der Zugwalze "ReferenceDiameter" anzugeben. Bei einem Wickler mit Tänzerlageregelung ist "ReferenceDiameter" der aktuelle Wickeldurchmesser (mit steigendem Durchmesser wird die Proportionalverstärkung des Tänzerlagereglers automatisch kleiner). Sind die Werte für den integralen Anteil (Nachstellzeit T_n , "IControl") und das Vorhaltglied (Vorhaltezeit T_D , "DControl") gleich Null, sind die internen Speichergrößen des PID-Reglers gelöscht. Am Eingang "FeedBackDancer" muss ein Wert anliegen, der proportional zur Tänzerwalzenverstellung ist (z. B. ein analoger Spannungswert). Sollwert und Istwert müssen die gleiche Einheit besitzen.



Der Funktionsbaustein verarbeitet nur positive Sollwerte und Istwerte.

Negative Sollwerte führen zu einem Fehler.

Negative Istwerte werden auf Null begrenzt.

Tänzerlageregelung

Zur Erfassung der Tänzerposition wird ein Messglied benötigt, das die Tänzerlage in ein proportionales Signal umsetzt. Die Ist-Position des Tänzers wird von einem gegebenen Sollwert abgezogen und als Regelabweichung auf einen intern eingebundenen PID-Regler gegeben. Die Stellgröße des Reglers wirkt auf den Drehzahlsollwert mittels Addition. Somit dreht sich die Wickelachse im Vergleich zur Leitachse schneller oder langsamer. Hiermit wird die Lage des Tänzers geregelt.

Unter normalen Bedingungen (z. B. keine hohen Drehzahlsollwertsprünge) erfolgt die Grobeinstellung der Umfangsgeschwindigkeit einer Wickelachse über den Getriebefinabgleich, die Grobeinstellung einer zugkraftgeregelten Achse über das elektronische Getriebe. Der Tänzerlageregler sollte somit lediglich kleine Abweichungen ausregeln.

Drehrichtung der Zug- bzw. Wickelachse, Regelsinn der Tänzerregelung

Die Drehrichtung der Zug- bzw. Wickelachse kann mit Hilfe des Antriebsparameters "P-0-0108 Polarität Leitantrieb" bzw. des Achsparameters "A-0-2798 Polarität der Leitachseposition" eingestellt werden. Dreht sich die Wickelachse nicht in die gewünschte Richtung, ist der Parameter P-0-0108 bzw. der Achsparameter A-0-2798 entsprechend umzustellen. Bei einer Invertierung der Drehrichtung des Antriebes ist auch der Regelsinn (ControlPolarity) zu invertieren.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_DancerControlType03 aufgelistet und beschrieben.

Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

Zusätzlich können die Fehlercodes des Bausteines IL_PIDType01, RIL_LoopControl, sowie MB_GetCyclicParameterHandle, ML_TechMotionBase, auftreten.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1160	16#0020	Fehler in der State machine
INPUT_RANGE_ERROR	16#1161	16#0002	HighLimit < LowLimit
		16#0003	FormatLength < 0.0
		16#0004	ReferenceDiameter < 0.0
		16#0005	ProcessWindow < 0.0
		16#0006	HighLimit < 0.0
		16#0007	LowLimit < 0.0
		16#0008	VelocityHighLimit < 0.0
		16#0009	P-Gain < 0.0
		16#000A	Setpoint < 0.0
		16#000B	MinFeedback < 0.0
		16#000C	MaxFeedback < 0.0
		16#000D	MaxFeedback < MinFeedback
		16#000E	Nummer der Slaveachse falsch
		16#000F	Nummer der Masterachse falsch
DEVICE_ERROR	16#1162	16#0001	Achse nicht im Betriebsmodus
RESSOURCE_ERROR	16#0004	16#0000	Falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert

Tab. 8-8: Fehlercodes FB MB_DancerControlType03

Notwendige Hardware

- CMLx, XMx, VPx
- IndraDrive

Notwendige Firmware

- MLC Firmware FWA-CMLXX*-MLC-14VRS
XX - Gerätevariante, z. B. CML65
- IndraDrive Antriebsfirmware

Notwendige Software

- IndraWorks 14VRS für MLC oder MLD

Besonderheiten für MLC

Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

Interpolation im Antrieb

Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss der Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus.

Interpolation in der Steuerung

Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss **kein** Parameter eingetragen sein. Die Zielgröße ist der Steuerungsparameter "A-0-2615 ELS, Geschwindigkeits-Sollwert additiv, Prozessregler".

Besonderheiten für MLD

Bei Verwendung der Steuerung MLD-S mit übergeordneter Steuerung MLC muss für die Tänzerreglerachse "Interpolation im Antrieb" aktiviert sein (**Reale Achsen ▶ entspr. Achse ▶ Eigenschaften**). Bei Verwendung der Steuerung MLD-M gibt es diese Einstellung nicht.

Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

Im zyklischen Kanal zwischen Steuerung und Antrieb des entsprechenden Antriebs muss der Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" eingetragen sein ("UserCmdDataX")

Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. (Der Dialog befindet sich im Projektbaum unterhalb des MLD-Knotens **AxisData**.)

Die "AxisData-Struktur" des Antriebes muss parametrierbar sein ("P-0-1367 SPS Konfiguration, Bit 6 ist gesetzt", bzw. Dialog SPS-Konfiguration)

Inbetriebnahme Tänzerlageregelung

Zur Inbetriebnahme der Tänzerlageregelung siehe [Kap. 8.2.4 "Inbetriebnahme Tänzerlageregelung"](#) auf Seite 508.

8.2.3 MB_DancerControlType04

Kurzbeschreibung

Der Funktionsbaustein berechnet zyklisch eine Tänzerlageregelung für eine Zugwalze mit Tänzer oder für eine Wickelachse mit Tänzer. Zyklische Ausgangsgröße des Tänzerlagereglers ist der Drehzahlversatz für den jeweiligen Antrieb einer Zugwalze oder Wickelachse. Des Weiteren wird die aktuelle Regelabweichung des Tänzerlagereglers ausgegeben.

Der Ausgang "AddVelocityDancer" des Funktionsbausteins wirkt auf den Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" (Interpolation im Antrieb) bzw. auf den Parameter "A-0-2615 ELS, Geschwindigkeits-Sollwert additiv, Prozessregler" (Interpolation in der Steuerung). Ist dies nicht gewünscht, kann dies mit dem Eingang "DisableCyclicWrites" verhindert werden. Es wird dann nur auf den Ausgang "AddVelocityDancer" geschrieben.

Die Normierung der Stellgrößenbegrenzung bzw. die Adaption der P-Verstärkung erfolgt über einen einstellbaren minimalen Bezugsdurchmesser (z. B. minimaler Hülsendurchmesser) und ist unabhängig von der Formatlänge, die beim Durchmesserrechner eingestellt wird.

Desweiteren sind Diagnoseeingänge für die Stellgrößenbegrenzung vorhanden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechWinder
MLD ab MPx20 mit MA Funktionspaket	MX_TechWinder

Tab. 8-9: Referenztable MB_DancerControlType04

Schnittstellenbeschreibung

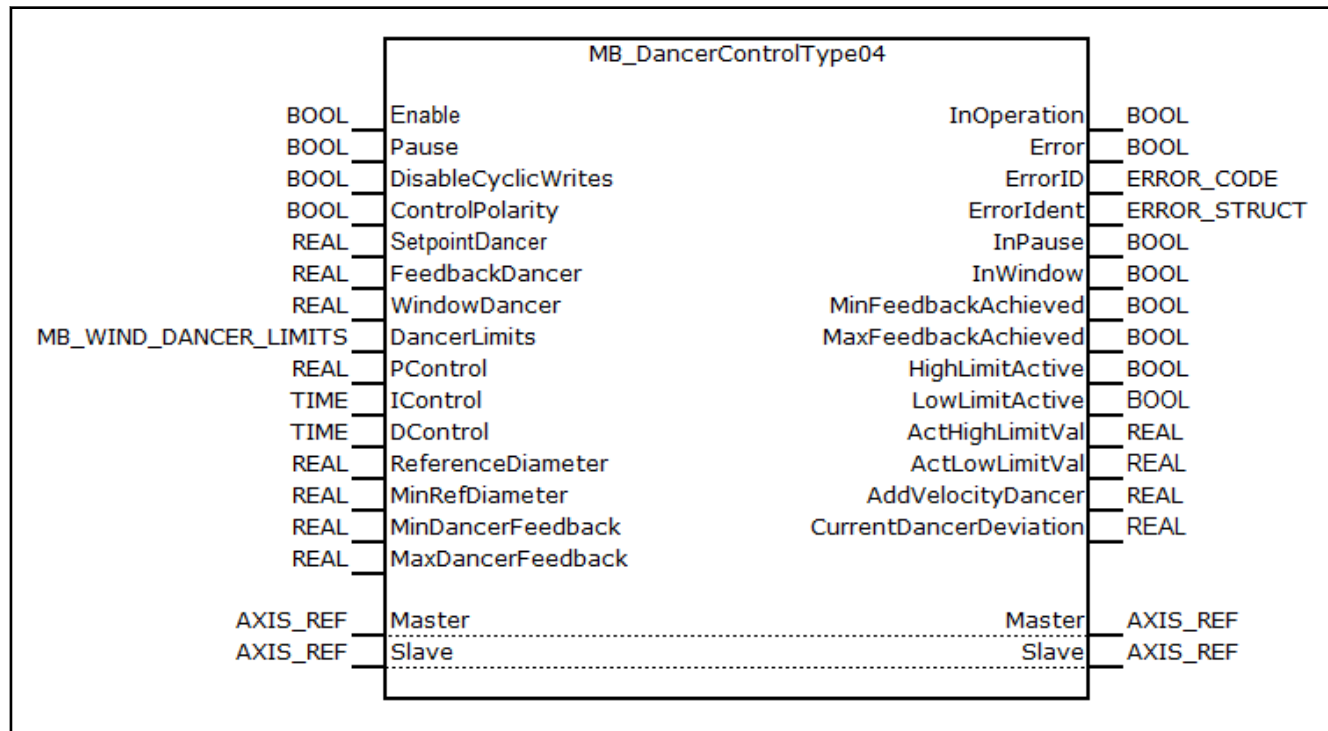


Abb. 8-5: Funktionsbaustein MB_DancerControlType04

I/O-TYP	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	Pause	BOOL	Tänzerregler pausiert, keine Geschwindigkeitsanpassung (pegelgesteuert)
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	ControlPolarity	BOOL	Polarität der Tänzerregelung wird invertiert (Übernahme bei pos. Flanke von Enable)
	SetpointDancer	REAL	Sollwert für die Tänzerregelung
	FeedbackDancer	REAL	Istwert der Tänzerlage
	WindowDancer	REAL	Fenster zur Überwachung des Tänzerlageistwertes [%], bezogen auf den Tänzerlagesollwert. Entfernt sich der Tänzer vom Lagesollwert um mehr als den angegebenen Wert, meldet dies der Baustein durch Rücksetzen des Ausganges "InWindow"
	DancerLimits	MB_WIND_DANCER_LIMITS	Struktur für die automatische Stellgrößenbegrenzung des Tänzers MB_WIND_DANCER_LIMITS, Seite 615
	PControl	REAL	Wert für die Kp-Verstärkung der Tänzerregelung
	IControl	TIME	Wert für die Nachstellzeit der Tänzerregelung
	DControl	TIME	Wert für die Vorhaltezeit der Tänzerregelung

I/O-TYP	Name	Typ	Kommentar
	ReferenceDiameter	REAL	Bezugsdurchmesser, (Wert dient zur Normierung von PControl und der Stellgrößenbegrenzung DancerLimits, d. h. diese Werte werden intern angepasst) <i>Bei reinem Tänzerlageregler:</i> - Durchmesser der Zugwalze [mm] <i>Bei Wickler mit Tänzer:</i> - Aktueller Wickeldurchmesser [mm] vom Durchmesserrechner Baustein zuweisen
	MinRefDiameter	REAL	Minimaler Durchmesser [mm], (Wert dient zur Normierung von PControl) <i>Bei reinem Tänzerlageregler:</i> - identisch zu ReferenceDiameter <i>Bei Wickler mit Tänzer:</i> - kleinster möglicher Hülsendurchmesser
	MinDancerFeedback	REAL	Untere Schwelle für den Tänzeristwert (nur für Anzeige mit binärem Ausgang)
	MaxDancerFeedback	REAL	Obere Schwelle für den Tänzeristwert (nur für Anzeige mit binärem Ausgang)
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose Fehlerbehandlung, Seite 506
	InPause	BOOL	Tänzerregler pausiert
	InWindow	BOOL	Tänzeristwert innerhalb des Tänzerfensters
	MinFeedbackAchieved	BOOL	Istwert der Tänzerlage ist unterhalb des Eingang MinDancerFeedback
	MaxFeedbackAchieved	BOOL	Istwert der Tänzerlage ist oberhalb des Eingang MaxDancerFeedback
	HighLimitActive	BOOL	Begrenzung des Drehzahloffsets "HighLimit" ist aktiv
	LowLimitActive	BOOL	Begrenzung Drehzahloffsets "LowLimit" ist aktiv
	ActHighLimitVal	REAL	Aktueller Wert für die obere Begrenzung des Drehzahloffsets
	ActLowLimitVal	REAL	Aktueller Wert für die untere Begrenzung des Drehzahloffsets
	AddVelocityDancer	REAL	Drehzahloffset für den Wickel- oder Zugwalzenantrieb [U/min]
	CurrentDancerDeviation	REAL	Aktuelle Regelabweichung des Tänzerreglers

I/O-TYP	Name	Typ	Kommentar
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (zugkraftgeregelter Achse)

Tab. 8-10: Schnittstellenvariablen FB MB_DancerControlType04

Min-/Max- und Standardwerte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Standardwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
ControlPolarity	BOOL			FALSE	Steigende Flanke an Enable
SetpointDancer	REAL	0.0	n. def.	0.0	Kontinuierlich
FeedbackDancer	REAL	n. def.	n. def.	0.0	Kontinuierlich
WindowDancer	REAL	0.0	n. def.	0.0	Kontinuierlich
DancerLimits	STRUCT				
PControl	REAL	0.0	n. def.	0.0	Kontinuierlich
IControl	TIME	0s	n. def.	0s	Kontinuierlich
DControl	TIME	0s	n. def.	0s	Kontinuierlich
ReferenceDiameter	REAL	0.0	n. def.	100	Kontinuierlich
MinRefDiameter	REAL	0.0	n. def.	0.0	Steigende Flanke an "Enable"
MinDancerFeedback	REAL	0.0	MaxDancerFeedback	0.0	Kontinuierlich
MaxDancerFeedback	REAL	MinDancerFeedback	n. def.	0.0	Kontinuierlich

Tab. 8-11: Min-/Max- und Default-Werte der Eingänge MB_DancerControlType04

Stellgrößenbegrenzung

Die Stellgrößenbegrenzung des Tänzerlagereglers wird in Abhängigkeit der Referenzachsgeschwindigkeit und dem Durchmesser Verhältnis ("MinRefDiameter" / "ReferenceDiameter") dynamisch angepasst. D. h. bei höherer Geschwindigkeit ist die Begrenzung der additiven Geschwindigkeit vom Tänzerlageregler automatisch größer als bei niedriger Geschwindigkeit. Die Berücksichtigung des Durchmesser Verhältnisses führt dazu, dass die Begrenzung bei Wickelanwendungen automatisch einem sich ändernden "ReferenceDiameter" anpasst wird (mit steigendem Durchmesser wird die Begrenzung automatisch kleiner).

Wird die Stellgröße des Tänzerlagereglers begrenzt, wird dies über die binären Ausgänge "HighLimitActive" bzw. "LowLimitActive" angezeigt. Die binären Ausgänge "ActHighLimitVal" und "ActLowLimitVal" zeigen die aktuellen Werte für die Begrenzung an.

Signal-Zeit-Diagramm

Im folgenden Signal-Zeit-Diagramm sind ausgewählte Signale des Funktionsbausteins dargestellt.

Zusammenhang der Signale

Signaleingang	Signalausgang	Beschreibung
Enable	InOperation	Baustein in Betrieb nehmen
	Error	Betriebsbereitschaft wurde unterbrochen
Pause	InPause	Internen Reglerprozess anhalten

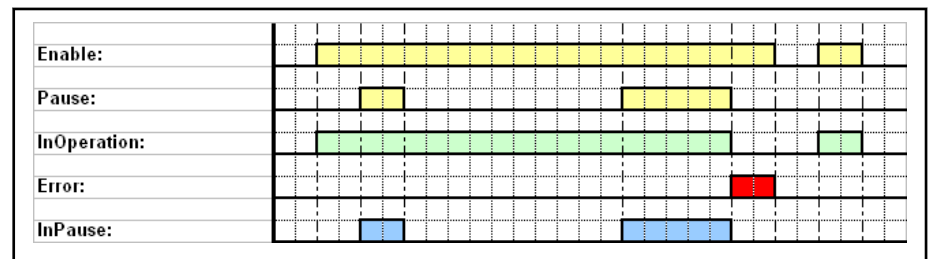


Abb. 8-6: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_DancerControlType04

Funktionsbeschreibung

Der Funktionsbaustein berechnet zyklisch eine Tänzerlageregelung für eine Zugwalze mit Tänzer oder für eine Wickelachse mit Tänzer.

Es wird der Funktionsbaustein IL_PIDType01, RIL_LoopControl, als Regler verwendet. Der Regler ist als PID-Regler ausgeführt. Zyklische Ausgangsgrößen des Tänzerlagereglers sind der Drehzahlversatz des jeweiligen Antriebs für die Zugwalze oder Wickelachse, sowie die aktuelle Regelabweichung des Tänzerlagereglers. Der Ausgang "AddVelocityDancer" des Funktionsbausteins wirkt für die Tänzerlageregelung auf den Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler". Die aktuelle Regelabweichung "CurrentDancerDeviation" steht zu Diagnosezwecken zur Verfügung. Die Proportionalverstärkung (Faktor K_p) und die Low- und High-Werte für die Stellgrößenbegrenzung "DancerLimits" werden automatisch mit dem Eingang "ReferenceDiameter", angepasst, bezogen auf den Eingang "MinRefDiameter". Für den Fall einer Tänzerlageregelung für eine Zugwalze ist der Durchmesser der Zugwalze für "ReferenceDiameter" und "MinRefDiameter" anzugeben. Bei einem Wickler mit Tänzerlageregelung ist "ReferenceDiameter" der aktuelle Wickeldurchmesser (mit steigendem Durchmesser wird die Proportionalverstärkung des Tänzerlagereglers automatisch kleiner) und "MinRefDiameter" der kleinste mögliche Hülsendurchmesser. Sind die Werte für den integralen Anteil (Nachstellzeit T_n , "IControl") und das Vorhaltglied (Vorhaltezeit T_D , "DControl") gleich Null, sind die internen Speichergrößen des PID-Reglers gelöscht. Am Eingang "FeedBackDancer" muss ein Wert anliegen, der proportional zur Tänzerwalzenverstellung ist (z. B. ein analoger Spannungswert). Sollwert und Istwert müssen die gleiche Einheit besitzen.



Der Funktionsbaustein verarbeitet nur positive Sollwerte und Istwerte.

Negative Sollwerte führen zu einem Fehler.

Negative Istwerte werden auf Null begrenzt.

Tänzerlageregelung

Zur Erfassung der Tänzerposition wird ein Messglied benötigt, das die Tänzerlage in ein proportionales Signal umsetzt. Die Ist-Position des Tänzers

wird von einem gegebenen Sollwert abgezogen und als Regelabweichung auf einen intern eingebundenen PID-Regler gegeben. Die Stellgröße des Reglers wirkt auf den Drehzahlsollwert mittels Addition. Somit dreht sich die Wickelachse im Vergleich zur Leitachse schneller oder langsamer. Hiermit wird die Lage des Tänzers geregelt.

Unter normalen Bedingungen (z. B. keine hohen Drehzahlsollwertsprünge) erfolgt die Grobeinstellung der Umfangsgeschwindigkeit einer Wickelachse über den Getriebefeinabgleich. Die Grobeinstellung einer zugkraftgeregelten Achse erfolgt über das elektronische Getriebe. Der Tänzerlagereger sollte somit lediglich kleine Abweichungen ausregeln.

Drehrichtung der Zug- bzw. Wickelachse, Regelsinn der Tänzerregelung

Die Drehrichtung der Zug- bzw. Wickelachse kann mit Hilfe des Antriebsparameters "P-0-0108 Polarität Leitantrieb" bzw. des Achsparameters "A-0-2798 Polarität der Leitachseposition" eingestellt werden. Dreht sich die Wickelachse nicht in die gewünschte Richtung, ist der Parameter P-0-0108 bzw. der Achsparameter A-0-2798 entsprechend umzustellen. Bei einer Invertierung der Drehrichtung des Antriebes ist auch der Regelsinn (ControlPolarity) zu invertieren.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_DancerControlType04 aufgelistet und beschrieben.

Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

Zusätzlich können die Fehlercodes des Bausteines IL_PIDType01, RIL_LoopControl, sowie MB_GetCyclicParameterHandle, ML_TechMotionBase, auftreten.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1160	16#0020	Fehler in der State machine
INPUT_RANGE_ERROR	16#1161	16#0002	HighLimit < LowLimit
		16#0003	FormatLength < 0.0
		16#0004	ReferenceDiameter < 0.0
		16#0005	ProcessWindow < 0.0
		16#0006	HighLimit < 0.0
		16#0007	LowLimit < 0.0
		16#0008	VelocityHighLimit < 0.0
		16#0009	P-Gain < 0.0
		16#000A	Setpoint < 0.0
		16#000B	MinFeedback < 0.0
		16#000C	MaxFeedback < 0.0
		16#000D	MaxFeedback < MinFeedback
		16#000E	Nummer der Slaveachse falsch
		16#000F	Nummer der Masterachse falsch
DEVICE_ERROR	16#1162	16#0001	Achse nicht im Betriebsmodus

ErrorID	Additional1	Additional2	Text
RESSOURCE_ERROR	16#0004	16#0000	Falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert

Tab. 8-12: Fehlercodes FB MB_DancerControlType04

Notwendige Hardware	• CMLx, XMx, VPx • IndraDrive
Notwendige Firmware	• MLC Firmware FWA-CMLXX*-MLC-14VRS XX - Gerätevariante, z. B. CML65 • IndraDrive Antriebsfirmware
Notwendige Software	• IndraWorks 14VRS für MLC
Besonderheiten für MLC	Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Interpolation im Antrieb: Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss der Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. Interpolation in der Steuerung: Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss kein Parameter eingetragen sein. Die Zielgröße ist der Steuerungsparameter "A-0-2615 ELS, Geschwindigkeits-Sollwert additiv, Prozessregler".
Besonderheiten für MLD	Bei Verwendung der Steuerung MLD-S mit übergeordneter Steuerung MLC muss für die Tänzerreglerachse "Interpolation im Antrieb" aktiviert sein (Reale Achsen ▶ entspr. Achse ▶ Eigenschaften). Bei Verwendung der Steuerung MLD-M gibt es diese Einstellung nicht. Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Im zyklischen Kanal zwischen Steuerung und Antrieb des entsprechenden Antriebs muss der Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" eingetragen sein ("UserCmdDataX") Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. (Der Dialog ist im Projektbaum unterhalb des MLD-Knotens Axis-Data.) Die "AxisData-Struktur" des Antriebes muss parametrierbar sein ("P-0-1367 SPS Konfiguration, Bit 6 ist gesetzt", bzw. Dialog SPS-Konfiguration). Achsreferenz für die Masterachse: Für die Technologiebausteine wird als Achsreferenz für die Masterachse empfohlen: • Für eine virtuelle Leitachse die Achsreferenz "VmAxis1" verwenden. Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 20 (VMA_1) eingestellt werden. • Für eine reale Leitachse die Achsreferenz "RmAxis1" verwenden. Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 23 (RMA_1) eingestellt werden. Zusätzlich muss in diesem Fall noch die logische Adresse des Messgebers im Parameter "P-0-1820.0.2 CCD: Quelle des Messgeber-Lageistwertes" eingetragen werden. Detaillierte Informationen können der Funktionsbeschreibung MLD entnommen werden.

Inbetriebnahme Tänzerlageregelung Zur Inbetriebnahme der Tänzerlageregelung siehe [Kap. 8.2.4 "Inbetriebnahme Tänzerlageregelung"](#) auf Seite 508.

8.2.4 Inbetriebnahme Tänzerlageregelung

Antriebsregelkreis	Es wird vorausgesetzt, dass der Drehzahlregelkreis des Zugwalzenantriebs eingestellt wurde. D. h. die Parameter "S-0-0100 Geschwindigkeitsregler-Proportionalverstärkung" und "S-0-0101 Geschwindigkeitsregler-Nachstellzeit" müssen der Applikation entsprechend eingestellt sein.
Kommandierung der Tänzerreglerachse	Für die Tänzerreglerachse muss die richtige Betriebsart des Antriebes sowie die Reglerfreigabe des Antriebes gesetzt werden. <i>Das Setzen von Betriebsart und Reglerfreigabe kann auf zwei unterschiedliche Arten erfolgen:</i> • Setzen der Betriebsart "ModeSyncVel" mit Hilfe des Achsinterfaces (dies wird empfohlen) • Benutzung der PLC-Open Funktionsbausteine "MC_Power", "MC_GearIn", "MC_Stop" zur Kommandierung der Achse
Grundsätzliche Einstellungen	Zunächst wird der Regelsinn des Tänzerlagereglers, abhängig von der Drehrichtung der Zugkraftachse, festgelegt. Abhängig hiervon wirkt der Regler in entgegengesetzten Richtungen. Der Regelsinn wird durch Setzen des Eingangs "ControlPolarity" festgelegt. Als weitere wichtige Grundeinstellung gilt die Festlegung des Bezugsdurchmessers (Eingang "ReferenceDiameter"). Beim Wickler mit Tänzer (siehe Wickelrechner mit Tänzer 524) wird hier der aktuell berechnete Wickeldurchmesser angeschlossen. Beim Tänzerlageregler wird der Durchmesser der zugkraftgeregelten Achse eingetragen. Mit den Eingängen "MinDancerFeedback" und "MaxDancerFeedback" können Werte für den Verstellbereich des Tänzers angegeben werden, die dann mittels der Ausgänge "MinFeedbackAchieved" und "MaxFeedbackAchieved" angezeigt werden. Diese sollten innerhalb der Endlagen des Tänzers liegen. Als nächstes wird das Prozessgrößenfenster ("WindowDancer") definiert. Mit dem Fenster wird die Überwachung der Tänzersollposition parametrierbar. Entfernt sich der Tänzerlageistwert vom Lagesollwert ("SetpointDancer"), bezogen auf den Sollwert um mehr als den angegebenen Prozentsatz, wird der binäre Ausgang "InWindow" zurückgesetzt ("InWindow": = FALSE). Dies ermöglicht der übergeordneten SPS entsprechend auf den Vorgang zu reagieren. Ein Verlassen des parametrierbaren Fensters hat zunächst keinen Einfluss auf die Funktion des Tänzerlagereglers. Beträgt beispielsweise der Tänzersollwert "SetpointDancer" = 500.0 und die Fenstergröße ist auf "WindowDancer" = 10% festgelegt, liegen die Grenzen des Fensters bei 450.0 und 550.0. Innerhalb dieses Wertebereichs ist der Ausgang "InWindow" = TRUE. Für den Baustein MB_DancerControlType03, Seite 495: Schließlich muss die Formatlänge angegeben werden. Die Formatlänge entspricht einer abgewickelten Umdrehung der Referenzachse (Eingang "FormatLength"). Für den Baustein MB_DancerControlType04, Seite 501: Für den Eingang "MinRefDiameter" wird beim reinen Tänzerlageregler der identische Wert wie am Eingang "ReferenceDiameter" eingetragen. Für einen Wickler mit Tänzer wird der kleinste mögliche Hülsendurchmesser eingetragen. Parametrierung des Tänzerlagereglers

In einem nächsten Schritt werden die Reglerparameter eingestellt:

- Proportionalverstärkung ("PControl")
- Nachstellzeit ("IControl")
- Vorhaltezeit ("DControl")

Zur Einstellung der Größen kann z. B. die Methode von Ziegler und Nichols eingesetzt werden. Der Regelkreis wird geschlossen und die Proportionalverstärkung ("PControl") solange erhöht, bis der Ausgang des Regelkreises bei konstantem Eingang eine Dauerschwingung mit der Periode T_{krit} bei der Proportionalverstärkung $K_{p,krit}$ ausführt. Die Einstellregeln für die Verstärkung K_p ("PControl"), die Nachstellzeit T_n ("IControl") und die Vorhaltezeit T_v ("DControl") lauten, wie in folgender Tabelle angegeben.

Regler	PControl	IControl	DControl
P-Regler	$0,5 \cdot K_{p,krit}$		
PI-Regler	$0,45 \cdot K_{p,krit}$	$0,85 \cdot T_{krit}$	
PD-Regler	$0,55 \cdot K_{p,krit}$		$0,15 \cdot T_{krit}$
PID-Regler	$0,6 \cdot K_{p,krit}$	$0,5 \cdot T_{krit}$	$0,12 \cdot T_{krit}$

Tab. 8-13: Reglerparametrierung nach Ziegler und Nichols

Schließlich werden die Eingänge der Struktur MB_WIND_DANCER_LIMITS belegt. Hier wird der Arbeitsbereich des Tänzerlagereglers durch Angabe der Unter- und Obergrenze ("LowLimitControl", "HighLimitControl") des internen Begrenzers festgelegt.

Erster Anlauf des Tänzerlagereglers

Zunächst wird am Eingang "SetpointDancer" ein Wert als Sollwert vorgegeben. Es ist in der Regel zweckmäßig, dass dieser Wert in etwa der Mittellage des Tänzers entspricht. In diesem Fall hat der Tänzer in beide Richtungen einen vergleichsweise großen Bahnweg.

Nach dem Einschalten des Tänzerlagereglers ("Enable" = TRUE) fährt der Tänzer in die vorgegebene Sollwertlage. Die Schwankungen der Bahngeschwindigkeit verursachen Schwankungen in der Bahnspannung, die auf die Tänzerlage und somit auf die Regelabweichung des Tänzerlagereglers wirken. Dies versucht der Tänzer über den Tänzerlageregler durch Anpassung der Antriebsgeschwindigkeit der zugkraftgeregelten Achse bzw. Wickelachse auszugleichen. An den Ausgängen "CurrentDancerDeviation" und "AddVelocityDancer" werden diese Größen angezeigt und können hiermit überwacht werden.

8.3 Durchmesserberechnung im Wickelrechner

8.3.1 Einführung und Übersicht

Der Funktionsbaustein MB_DiameterCalculatorType03 stellt einen vielseitigen Funktionsbaustein für die Umsetzung von Wickelrechnern dar. Er kommt in dieser Dokumentation in allen beschriebenen Wickelrechnern zum Einsatz. Aus diesem Grund wird er zuerst beschrieben. Im Folgenden wird der Funktionsbaustein erläutert.

8.3.2 MB_DiameterCalculatorType03

Kurzbeschreibung

Der Baustein berechnet zyklisch den aktuellen Durchmesser, die Geschwindigkeitsnachführung für die Wickelachse und das aktuelle Trägheitsmoment des Wickelgutes auf einem Wickel. Der Funktionsbaustein wird für jede

Wickelachse aufgerufen. Zyklische Ausgangsgrößen sind der aktuelle Wickeldurchmesser "CurrentDiameter", der resultierende Getriebefineabgleich für den zugehörigen Wickelantrieb "GearRatioFineAdjust" und das aktuelle Trägheitsmoment des Wickelgutes "CurrentInertia".

Der Getriebefineabgleich "GearRatioFineAdjust" wirkt auf den Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" (Interpolation im Antrieb) bzw. auf den Parameter "A-0-2605 ELS, Feinabgleich der Getriebeübersetzung, Prozessregler" (Interpolation auf der Steuerung). Dieser Parameter wird im entsprechenden zyklischen Sercos-Kanal eingetragen. Die Parametrierung erfolgt im Parametriermodus.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechWinder
MLD mit MPx07/08/17VRS mit MA Funktionspaket	MX_TechWinder.lib (Indralogic 1.x)
MLD ab MPx20 mit MA Funktionspaket	MX_TechWinder

Tab. 8-14: Referenztabelle MB_DiameterCalculatorType03

Schnittstellenbeschreibung

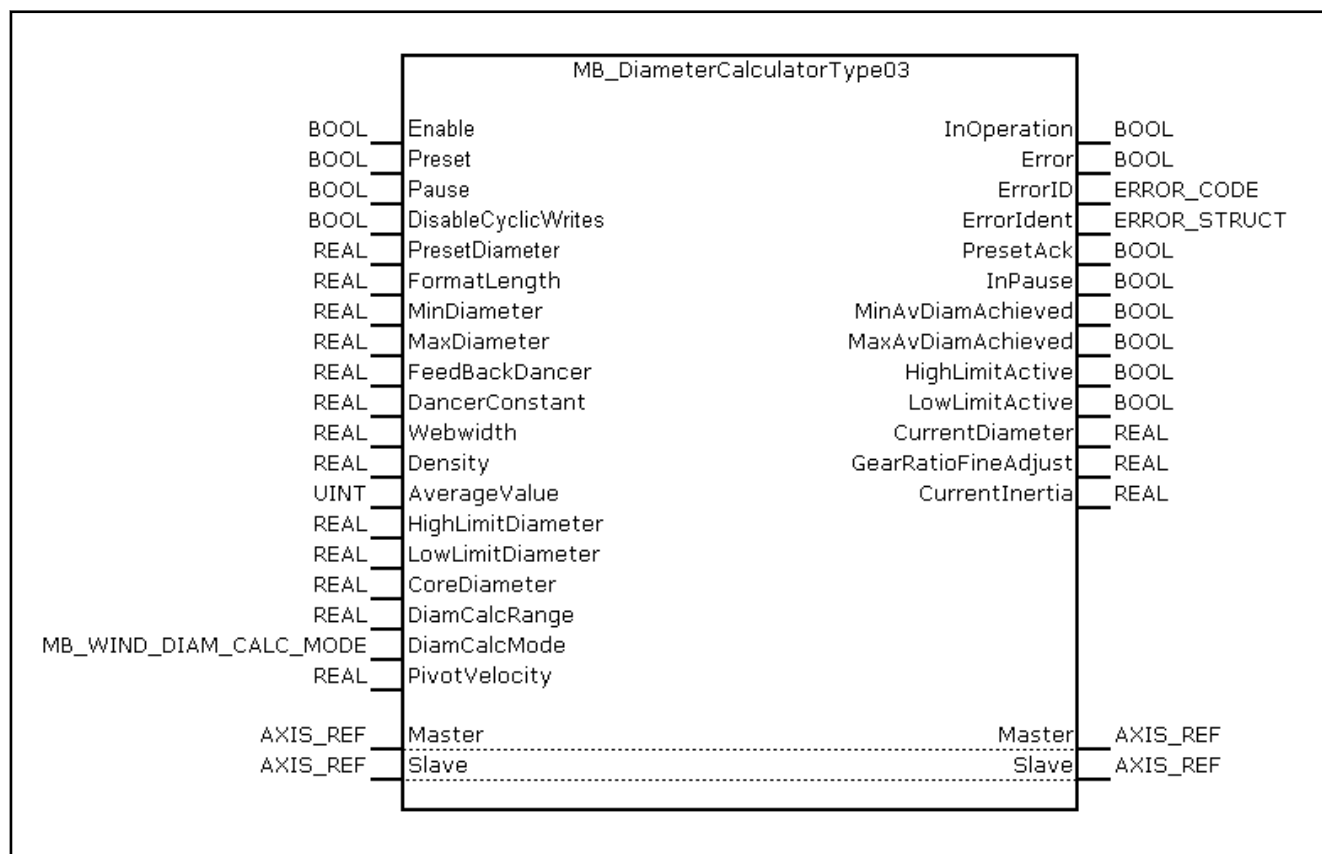


Abb. 8-7: Funktionsbaustein MB_DiameterCalculatorType03

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	Preset	BOOL	Preset des aktuellen Wickeldurchmessers

E/A-Typ	Name	Typ	Kommentar
	Pause	BOOL	Durchmesserrechner pausiert (pegelgesteuert)
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	PresetDiameter	REAL	CurrentDiameter wird auf PresetVal gesetzt, GearRatioFineAdjust entsprechend angepasst (flanken-gesteuert)
	FormatLength	REAL	Formatlänge [mm] (entspricht einer abgewickelten Umdrehung der Referenzachse)
	MinDiameter	REAL	Minimaler Durchmesser [mm] Wird lediglich für die Anzeige durch den Ausgang "MinAvDiamAchieved" verwendet
	MaxDiameter	REAL	Maximaler Durchmesser [mm] Wird lediglich für die Anzeige durch den Ausgang "MaxAvDiamAchieved" verwendet
	FeedbackDancer	REAL	Tänzerlageistwert (nur für die Funktionalität "Wickler mit Tänzer")
	DancerConstant	REAL	Bahnweg bei Verstellung der Tänzerwalze, bezogen auf die Einheit der Tänzerverstellung [mm] Vorzeichen ist abhängig vom Aufbau des Tänzers einzustellen! (nur für die Funktionalität "Wickler mit Tänzer")
	Webwidth	REAL	Bahnbreite [mm]
	Density	REAL	Materialdichte [kg/dm ³]
	AverageValue	UINT	Anzahl der Durchmesserwerte, die gemittelt werden. Eingang wird auf maximal 30 begrenzt
	HighLimitDiameter	REAL	Obere Grenze für den Durchmesser [mm]. Auf diesen Wert wird der Durchmesser begrenzt
	LowLimitDiameter	REAL	Untere Grenze für den Durchmesser [mm]. Der Durchmesser unterschreitet diesen Wert nicht
	CoreDiameter	REAL	Hülsendurchmesser des Wickels [mm]
	DiamCalcRange	REAL	Bereich für die Durchmesserberechnung [Grad]. Gibt an, nach wieviel Grad der Durchmesser regelmäßig berechnet werden soll. Bezieht sich auf die Wickelachse (lastseitig)
	DiamCalcMode	MB_WIND_DI-AM_CALC_MODE	Modus für die Durchmesserberechnung MB_WIND_DIAM_CALC_MODE , Seite 616
	PivotVelocity	REAL	Schwenkgeschwindigkeit des Wickels (Relativgeschwindigkeit zur Wickelbewegung) [mm/s] Damit wird die Durchmesserberechnung bei einer Relativbewegung der Wickelachse korrigiert
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet

E/A-Typ	Name	Typ	Kommentar
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose Fehlerbehandlung, Seite 515
	PresetAck	BOOL	Presetdurchmesser wurde übernommen (ist gesetzt, solange Preset ansteht)
	InPause	BOOL	Durchmesserrechner pausiert
	MinAvDiamAchieved	BOOL	Minimaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	MaxAvDiamAchieved	BOOL	Maximaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	HighLimitActive	BOOL	Aktuell berechneter Durchmesser hat obere Grenze erreicht (Durchmesserwert wird verworfen)
	LowLimitActive	BOOL	Aktuell berechneter Durchmesser hat untere Grenze erreicht. (Durchmesserwert wird verworfen)
	CurrentDiameter	REAL	Aktueller Wickeldurchmesser
	GearRatioFineAdjust	REAL	Getriebefeinabgleich zur Drehzahlachsführung des Antriebes [%]
	CurrentInertia	REAL	Aktuelles gesamtes Trägheitsmoment (bezogen auf die Motorseite) [kg/m ²]
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (zugkraftgeregelter Achse)

Tab. 8-15: Schnittstellenvariablen FB MB_DiameterCalculatorType03

Min-/Max- und Standardwerte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Standardwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an Enable
PresetDiameter	REAL	> 0.0	n. def.	100.0	Steigende Flanke an Preset
FormatLength	REAL	> 0.0	n. def.	100.0	Steigende Flanke an Enable
MinDiameter	REAL	> 0.0	n. def.	100.0	Kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
MaxDiameter	REAL	> 0.0	n. def.	2000.0	Kontinuierlich
FeedbackDancer	REAL	0.0	n. def.	0.0	Kontinuierlich
DancerConstant	REAL	n. def.	n. def.	0.0	Kontinuierlich
Webwidth	REAL	0.0	n. def.	0.0	Steigende Flanke an Enable
Density	REAL	0.0	n. def.	1.0	Steigende Flanke an Enable
AverageValue	UINT	1	30	1	Kontinuierlich
HighLimitDiameter	REAL	0.0	n. def.	2000.0	Kontinuierlich
LowLimitDiameter	REAL	0.0	n. def.	100.0	Kontinuierlich
CoreDiameter	REAL	0.0	n. def.	100.0	Steigende Flanke an Enable
DiamCalcRange	REAL	0.0	n. def.	360.0	Kontinuierlich
DiamCalcMode	ENUM			0	Steigende Flanke an Enable
PivotVelocity	REAL	0.0	n. def.	0.0	Kontinuierlich

Tab. 8-16: Min-/Max- und Default-Werte MB_DiameterCalculatorType03

Signal-Zeit-Diagramm

Im folgenden Signal-Zeit-Diagramm sind die Ein- bzw. Ausgänge des Funktionsbausteins dargestellt.

Zusammenhang der Signale

Signaleingang	Signalausgang	Beschreibung
Enable	InOperation	Baustein in Betrieb nehmen
	Error	Betriebsbereitschaft wurde unterbrochen
Pause	InPause	Interne Berechnung anhalten
Preset	PresetAck	Mit positiver Flanke am Eingang "Preset" wird der Durchmesserrechner gestartet. Startwert ist der am Eingang "PresetDiameter" anstehende Wert
PresetDiameter		

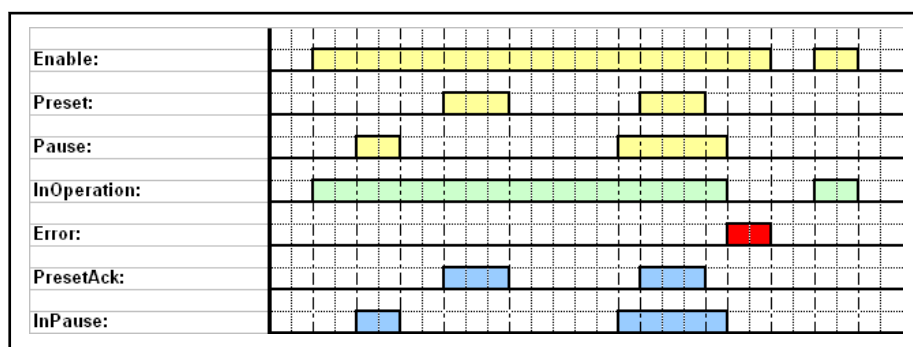


Abb. 8-8: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_DiameterCalculatorType03

Funktionsbeschreibung**Preset des Durchmessers**

Da der berechnete Durchmesser über den resultierenden Feinabgleich direkt auf die Wickelachsgeschwindigkeit wirkt, muss zu Beginn der Berechnung bei Aktivierung des Wickelrechners ein sinnvoller Wert für den Durchmesser vorgegeben werden. Daher sollte am Eingang "PresetDiameter" ein Wert für den Durchmesser angegeben werden, der nahe dem tatsächlichen Durchmessers liegt und ein Preset durchgeführt werden. Im Falle eines Aufwicklers sollte der Wert etwas kleiner und für Abwickler etwas größer gewählt werden. Damit wird jeweils ein leichter Anlauf ermöglicht.

Mit jeder neuen Rolle sollte einmalig zu Beginn ein Preset des Wickeldurchmessers durchgeführt werden. Dies erfolgt über den Eingang "Preset". Der Durchmesser wird auf den Wert des Eingangs "PresetDiameter" gesetzt. Das Setzen des Durchmessers mit Hilfe von "Preset" ist zu jeder Zeit ohne Setzen des Eingangs "Pause" möglich.

Nach dem Einschalten der Steuerung ist der Wickeldurchmesser, der Getriebefeinabgleich und das Trägheitsmoment gleich Null. Dies bedeutet, dass nach jedem Einschalten der Steuerung ein Preset durchgeführt werden sollte, um die entsprechenden Größen mit sinnvollen Werten vorzubelegen. Erfolgt dies nicht, kann ein nicht sinnvoller Wert für den aktuellen Wickeldurchmesser berechnet werden. Dies kann zur Fehlfunktion des Wickelrechners und damit zum Halt des Wicklers führen.

Ist es nicht möglich einen korrekten Presetwert für den Anfangswickels zu bestimmen, wird der "auto-calc-mode" für die Durchmesserberechnung empfohlen (FIRST_REV_AUTO_CALC). Hierbei wird für die erste Wickelachsumdrehung nach dem Einschalten der Bereich für die Durchmesserberechnung (DiamCalcRange) automatisch angepasst, um schneller einen vernünftigen Anfangsdurchmesser zu bekommen.



Soll der aktuelle Durchmesser, nach dem Ausschalten der Steuerung erhalten bleiben, muss der Ausgang "CurrentDiameter" auf eine remanente Variable kopiert werden. Mit diesem Wert kann nach dem Einschalten wieder ein Preset ausgeführt werden.

Durchmesserberechnung

Die Durchmesserberechnung ist proportional zum Quotienten aus der Referenzachsgeschwindigkeit n_{Ref} und der Wickelachsgeschwindigkeit n_{Wickel} und zur Formatlänge F (entspricht einer abgewickelten Umdrehung der Referenzachse).

$$d \sim \frac{n_{Ref} \cdot F}{n_{Wickel}}$$

Abb. 8-9: Proportionalität bei der Durchmesserberechnung



Das Verhältnis "Formatlänge" / "CoreDiameter" darf nicht größer als 25 sein.

Ansonsten reicht die Getriebeanpassung über den Getriebefeinabgleich nicht aus!



Eine Berechnung des Durchmessers erfolgt nur bei drehender Referenzachse.

Das Stillstandsfenster der Referenzachse muss passend eingestellt sein.

Der berechnete Wickeldurchmesser wirkt direkt auf den resultierenden Feinabgleich des Getriebes. Damit wird eine durchmesserabhängige Geschwindigkeitsanpassung des Wickels umgesetzt.

Mittelwertbildung

Der berechnete aktuelle Durchmesser "CurrentDiameter" kann mit Setzen des Eingangs "AverageValue" gemittelt werden.

An "AverageValue" wird die Anzahl der Durchmesserwerte eingetragen, über die gemittelt werden soll. Zur Mittelwertbildung tragen nur die intern berechneten aktuellen Durchmesser bei, die innerhalb der Grenzen von "HighLimitDiameter" und "LowLimitDiameter" liegen. Außerhalb liegende Werte werden verworfen. Für "AverageValue" sind Werte zwischen 1 und 10 zweckmäßig. Aus diesem Grund wird "AverageValue" auf maximal 30 begrenzt. Wird ein größerer Wert eingegeben, wird "AverageValue" auf 30 begrenzt. Je höher die Zahl gewählt wird, desto glatter wird der berechnete Durchmesser. Allerdings nimmt mit zunehmenden Werten für "AverageValue" die Genauigkeit des Durchmessers ab, da dieser aufgrund der "Filterung" nachläuft. Daher ist der Wert für "AverageValue" geeignet zu wählen. Bei kleineren Wickeldurchmessern ist die Änderung der berechneten Durchmesserwerte höher. Aus diesem Grund darf der Wert des Filters nicht zu groß gewählt werden. Die durch die Mittelung verursachte Verzögerung des berechneten Durchmessers hätte sonst ein deutliches "Hinterherlaufen" des berechneten Durchmessers zur Folge. Bei einem Wickler mit Tänzer beispielsweise kann dies zu unerwünschten Auslenkungen des Tänzers führen.

Berechnung des Trägheitsmoments

Mit Hilfe der Eingänge "Webwidth" und "Density" wird intern im Durchmesserrechner das Trägheitsmoment des Wickels berechnet und am Ausgang "CurrentInertia" angezeigt. Mit Hilfe des, vom Durchmesser abhängigen, Trägheitsmoments kann, falls erforderlich, eine Adaption der K_p -Verstärkung des Drehzahlreglers (S-0-0100) im SPS-Projekt in IndraLogic implementiert werden.

Vorbereitung Rollenwechsel

Schließlich können die Eingänge "MinDiameter" und "MaxDiameter" belegt werden. Die hier angegebenen Größen wirken auf die Ausgänge "MinAvDiamAchieved" und "MaxAvDiamAchieved" und melden das Überschreiten der angegebenen Grenzen durch den berechneten aktuellen Wickeldurchmesser. Die jeweiligen Werte sollten innerhalb des erlaubten Durchmesserbereichs liegen, der durch die Eingänge "HighLimitDiameter" und "LowLimitDiameter" definiert wird. So können z. B. für einen Rollenwechsel oder die Anpassung des Eingangs "DiamCalcRange" anhand einer Vorgabe des minimalen/maximalen Durchmessers der Zeitpunkt für bestimmte Prozessschritte in der SPS detektiert und angezeigt werden.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_DiameterCalculatorType03 aufgelistet und beschrieben.

Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

Zusätzlich können die Fehlercodes des Bausteins MB_GetCyclicParameterHandle, ML_TechMotionBase, auftreten.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1100	16#0020	Fehler in der StateMachine
INPUT_RANGE_ERROR	16#1101	16#0001	HighLimitDiameter < LowLimitdiameter
		16#0002	FormatLength < 0.0
		16#0003	Webwidth < 0.0
		16#0004	Density < 0.0
		16#0005	Core diameter < 0.0
		16#0006	DiamCalcRange < 0.0
		16#0007	Preset diameter < 0.0
		16#0008	Minimum diameter < 0.0
		16#0009	Maximum diameter < 0.0
		16#000A	HighLimitDiameter < 0.0
		16#000B	LowLimitDiameter < 0.0
		16#000C	DiamCalcMode ungültig
		16#000D	Nummer der Slaveachse falsch
		16#000E	Nummer der Masterachse falsch
DEVICE_ERROR	16#1102	16#0001	kein Betriebsmodus (keine sercos Phase 4)
RESSOURCE_ERROR	16#0004	16#0000	falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert

Tab. 8-17: Fehlercodes FB MB_DiameterCalculatorType03

Notwendige Hardware

- CMLx, XMx, VPx
- IndraDrive

Notwendige Firmware

- MLC Firmware FWA-CMLXX*-MLC-14VRS
XX - Gerätevariante, z. B. CML65
- IndraDrive Antriebsfirmware

Notwendige Software

- IndraWorks 14VRS für MLC bzw. MLD

Besonderheiten für MLC Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

Eine real vorhandene mechanische Getriebeübersetzung von der Antriebsachse zur Wickelachse muss in die Parameter "S-0-0121 Lastgetriebe-Eingangsumdrehungen" und "S-0-0122 Lastgetriebe-Ausgangsumdrehungen" eingetragen werden.

Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z. B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!

Interpolation im Antrieb: Im zyklischen sercos Kanal des entsprechenden Antriebs muss der Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus.

Interpolation in der Steuerung: Im zyklischen sercos Kanal des entsprechenden Antriebs muss **kein**Parameter eingetragen sein. Die Zielgröße ist der

- Steuerungsparameter "A-0-2605 ELS, Feinabgleich der Getriebeübersetzung, Prozessregler".
- Besonderheiten für MLD** Bei Verwendung der Steuerung MLD-S mit übergeordneter Steuerung MLC muss für die Wickelachse Interpolation im Antrieb aktiviert sein (**reale Achsen ▶ entspr. Achse ▶ Eigenschaften**). Bei Verwendung der Steuerung MLD-M gibt es diese Einstellung nicht.
- Die Wichtungsart für die Master-Achse und Slave-Achse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.
- Im zyklischen Kanal zwischen Steuerung und Antrieb des entsprechenden Antriebs muss der Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" eingetragen sein ("UserCmdDataX")
- Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus (der Dialog befindet sich im Projektbaum unterhalb des MLD-Knotens → AxisData).
- Die "AxisData-Struktur" des Antriebes muss parametriert sein ("P-0-1367 SPS Konfiguration, Bit 6 ist gesetzt", bzw. Dialog SPS-Konfiguration).
- Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter "S-0-0121 Lastgetriebe-Eingangsumdrehungen" und "S-0-0122 Lastgetriebe-Ausgangsumdrehungen" eingetragen werden.
- Stillstandsfenster Referenzachse:**
- Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z. B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!
- Für die Referenzachse kann der Funktionsbaustein auf keinen Wert für das Stillstandsfenster zugreifen. Intern wird dann ein Wert von 1 U/min angenommen. Ist dieser Wert für die Applikation nicht geeignet, kann dieser mittels folgender SPS-Codezeilen beeinflusst werden:

Deklaration

```
prMaStandstillWdw: POINTER TO REAL; // for StandstillWindow master MLD
rTestMaStandstillWdw: REAL := 0.5; // for StandstillWindow master MLD
```

Implementation

```
prMaStandstillWdw := ADR(BausteinInstanznameDurchmesserrechner.rMaStandstillWindow);
prMaStandstillWdw^ := rTestMaStandstillWdw;
```

Der Wert für diese Variable bleibt nicht gespeichert, der Wert muss beim Einschalten der SPS also neu geschrieben werden.

Achsreferenz für die Masterachse:

Für die Technologiebausteine wird als Achsreferenz für die Masterachse empfohlen:

- Für eine virtuelle Leitachse die Achsreferenz "**VmAxis1**" verwenden.
Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 20 (VMA_1) eingestellt werden.
- Für eine reale Leitachse die Achsreferenz "**RmAxis1**" verwenden.
Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 23 (RMA_1) eingestellt werden.
Zusätzlich muss in diesem Fall noch die logische Adresse des Messgebers im Parameter "P-0-1820.0.2 CCD: Quelle des Messgeber-Lageistwertes" eingetragen werden.

Inbetriebnahme des Durchmesserrechners

Detaillierte Informationen können der Funktionsbeschreibung MLD entnommen werden.

Zur Inbetriebnahme des Durchmesserrechners siehe [Kap. 8.3.3 "Inbetriebnahme des Durchmesserrechners"](#) auf Seite 518.

8.3.3 Inbetriebnahme des Durchmesserrechners**Antriebsregelkreis**

Es wird vorausgesetzt, dass der Drehzahlregelkreis des Wickelantriebes eingestellt wurde. D. h. die Parameter "S-0-0100 Geschwindigkeitsregler-Proportionalverstärkung" und "S-0-0101 Geschwindigkeitsregler-Nachstellzeit" müssen der Applikation entsprechend eingestellt sein.

Für die Inbetriebnahme des Durchmesserrechners sind die folgenden Schritte zweckmäßig:

- Festlegung der Stoffbahngrößen
- Festlegung der Wickelgrößen
- Festlegung der zyklisch veränderlichen Eingänge
- Festlegung der Eingänge zur Manipulation der Durchmesserberechnung und
- weitere Festlegungen

Festlegung der Stoffbahngrößen

Als Stoffbahngrößen werden die Eingänge und Werte bezeichnet, die sich auf die Abmaße der Stoffbahn beziehen und üblicherweise während des Betriebs als konstant angenommen werden können. Ausnahme bilden Stoffbahnen mit veränderlichem Material oder variablen Abmessungen.

Folgende Eingänge beschreiben Stoffbahngrößen:

- FormatLength
- Webwidth
- Density

Da diese Größen nicht zyklisch eingelesen werden, müssen sie vor der Aktivierung des Bausteins als Eingangswerte an den Eingängen anliegen.

Festlegung der Wickelgrößen

Wickelgrößen beschreiben die räumlichen, allgemeinen Abmaße des Wickels. Diese sind der Hülsendurchmesser "CoreDiameter", die maximal und minimal zulässigen Wickeldurchmesser "LowLimitDiameter" und "HighLimitDiameter". Über das Verhältnis von Referenz- und Wickelachsdrehzahl wird mit dem Hülsendurchmesser der aktuelle Durchmesser bestimmt. Der aktuelle Durchmesser "CurrentDiameter" wird auf diese angegebenen Grenzen begrenzt.

Festlegung zyklisch veränderlicher Eingänge

Zu Beginn und während der aktiven Durchmesserberechnung, kann es erforderlich sein, den Wert des Durchmesser manuell zu setzen. Hierzu wird der Eingang "PresetDiameter" beschrieben. Er wird zyklisch eingelesen und somit innerhalb eines Zyklus wirksam.

Festlegung der Eingänge zur Manipulation der Durchmesserberechnung

Die Methode zur Berechnung des Durchmessers ist so implementiert, dass mittels zweier Eingänge des Funktionsbaustein auf das Resultat der Berechnung Einfluss genommen werden kann. "AverageValue" gibt an, über wie viele berechnete Durchmesserwerte gleitend gemittelt werden soll. "DiamCalcRange" erlaubt es, die Häufigkeit der Berechnung zu beeinflussen, was wiederum Einfluss auf die Genauigkeit des Durchmesserwertes hat. Beide

Größen können, da sie zyklisch eingelesen und verarbeitet werden, während des Betriebs manipuliert und angepasst werden.

Weitere Festlegungen

Abhängig von der jeweiligen Anwendung werden die Eingänge "DancerConstant", "FeedBackDancer" (beide nur für den Fall der Funktionalität eines Wicklers mit Tänzer), "MinDiameter" und "MaxDiameter" beschaltet. Sollten diese Eingänge für die zu implementierenden Funktionen im SPS-Projekt nicht benötigt werden, ist es empfehlenswert, die Eingänge "MinDiameter" und "MaxDiameter" mit den gleichen Werten, wie "LowLimitDiameter" und "HighLimitDiameter" zu belegen. Hierdurch wird eine entsprechende Fehlermeldung vermieden.

Kommandierung der Wicklerachse

Für die Wicklerachse muss die richtige Betriebsart des Antriebes, sowie die Reglerfreigabe des Antriebes gesetzt werden.

Das Setzen von Betriebsart und Reglerfreigabe kann auf zwei unterschiedliche Arten erfolgen:

- Setzen der Betriebsart "ModeSyncVel" mit Hilfe des Achsinterfaces (dies wird empfohlen)
- Benutzung der PLCopen-Funktionsbausteine "MC_Power", "MC_GearIn", "MC_Stop" zur Kommandierung der Achse

8.4 Erfassung eines extern gemessenen Durchmessers

8.4.1 Einführung und Übersicht

Der Funktionsbaustein [MB_DiameterMeasurementType01](#), Seite 519, liest einen gemessenen Durchmesser ein und führt die Geschwindigkeit des Wickelantriebes nach. In diesem Fall ersetzt er dann den Baustein [MB_DiameterCalculatorType03](#), Seite 509.

Er wird in den Bausteinen [MB_WinderDancerCtrlType01](#), Seite 525, und [MB_WinderTensionCtrlType02](#), Seite 542, verwendet.

8.4.2 MB_DiameterMeasurementType01

Kurzbeschreibung

Der Baustein liest einen gemessenen Durchmesser ein und berechnet die Geschwindigkeitsnachführung für die Wickelachse und das aktuelle Trägheitsmoment des Wickelgutes auf einem Wickel. Der Funktionsbaustein wird für jede Wickelachse aufgerufen, bei der der Durchmesser gemessen wird. Zyklische Ausgangsgrößen sind der resultierende Getriebefeinabgleich für den zugehörigen Wickelantrieb "GearRatioFineAdjust" und das aktuelle Trägheitsmoment des Wickelgutes "CurrentInertia".

Der Getriebefeinabgleich "GearRatioFineAdjust" wirkt auf den Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" (Interpolation im Antrieb) bzw. auf den Parameter "A-0-2605 ELS, Feinabgleich der Getriebeübersetzung, Prozessregler" (Interpolation auf der Steuerung). Dieser Parameter wird im entsprechenden zyklischen Sercos-Kanal eingetragen. Die Parametrierung erfolgt im Parametriermodus.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS	RMB_TechWinder
Funktionspaket TEC	

MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechWinder.lib (Indralogic 1.x)
MLD ab MPx20 mit MA Funktionspaket	MX_TechWinder

Tab. 8-18: Referenztabelle MB_DiameterMeasurementType01

Schnittstellenbeschreibung

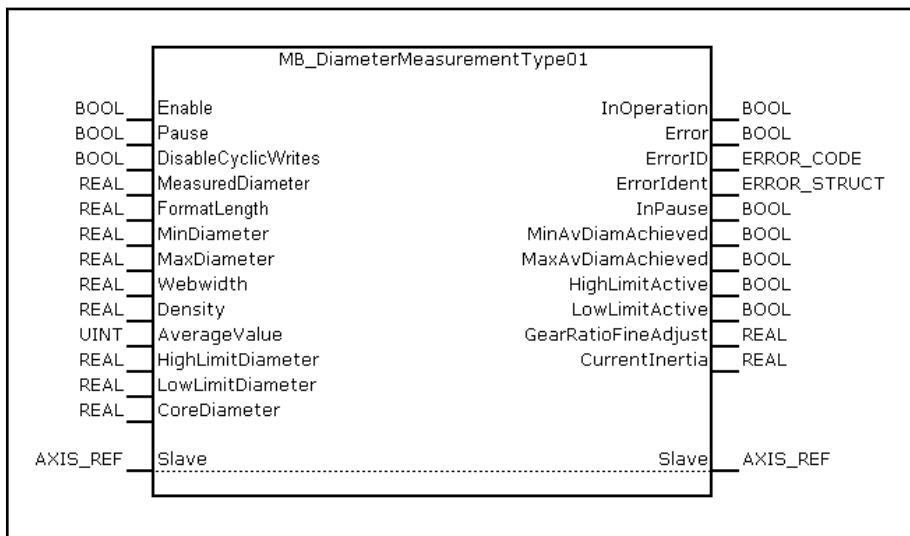


Abb. 8-10: Funktionsbaustein MB_DiameterMeasurementType01

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	Pause	BOOL	FB pausiert, Getriebefeinabgleich bleibt "eingefroren"
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	MeasuredDiameter	REAL	Gemessener Durchmesser [mm]
	FormatLength	REAL	Formatlänge [mm] (1 Umdrehung der Masterachse abgewickelt)
	MinDiameter	REAL	Minimaler Durchmesser [mm], wird lediglich für die Anzeige durch den Ausgang "MinAvDiamAchieved" verwendet
	MaxDiameter	REAL	Maximaler Durchmesser [mm], wird lediglich für die Anzeige durch den Ausgang "MaxAvDiamAchieved" verwendet
	WebWidth	REAL	Bahnbreite [mm]
	Density	REAL	Materialdichte [kg/dm³]
	AverageValue	UINT	Anzahl der Durchmesserwerte, die gemittelt werden. Eingang wird auf maximal 30 begrenzt
	HighLimitDiameter	REAL	Obere Grenze für den Durchmesser [mm] Auf diesen Wert wird der Durchmesser begrenzt

E/A-Typ	Name	Typ	Kommentar
	LowLimitDiameter	REAL	Untere Grenze für den Durchmesser [mm] Auf diesen Wert wird der Durchmesser begrenzt
	CoreDiameter	REAL	Hülsendurchmesser des Wickels [mm]
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose "Fehlerbehandlung" auf Seite 522
	InPause	BOOL	Tänzerregler pausiert
	MinAvDiamAchieved	BOOL	Minimaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	MaxAvDiamAchieved	BOOL	Maximaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	HighLimitActive	BOOL	Aktuell berechneter Durchmesser hat obere Grenze erreicht (Durchmesserwert wird verworfen)
	LowLimitActive	BOOL	Aktuell berechneter Durchmesser hat untere Grenze erreicht (Durchmesserwert wird verworfen)
	GearRatioFineAdjust	REAL	Getriebefeinabgleich zur Drehzahlachsführung des Antriebes [%]
	CurrentInertia	REAL	Aktuelles gesamtes Trägheitsmoment (bezogen auf die Motorseite) [kg/m ²]
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Referenz zur Slaveachse

Tab. 8-19: Schnittstellenvariablen FB MB_DiameterMeasurementType01

Min-/Max- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an Enable
MeasuredDiameter	REAL	> 0.0	n. def.	100.0	Kontinuierlich
FormatLength	REAL	> 0.0	n. def.	100.0	Steigende Flanke an Enable
MinDiameter	REAL	> 0.0	n. def.	100.0	Kontinuierlich
MaxDiameter	REAL	> 0.0	n. def.	2000.0	Kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Webwidth	REAL	0.0	n. def.	0.0	Steigende Flanke an Enable
Density	REAL	0.0	n. def.	1.0	Steigende Flanke an Enable
AverageValue	UINT	0	30	1	Kontinuierlich
HighLimitDiameter	REAL	0.0	n. def.	2000.0	Kontinuierlich
LowLimitDiameter	REAL	0.0	n. def.	100.0	Kontinuierlich
CoreDiameter	REAL	0.0	n. def.	100.0	Steigende Flanke an Enable

Tab. 8-20: Min-/Max- und Default-Werte der Eingänge MB_DiameterMeasurementType01

Signal-Zeitdiagramm

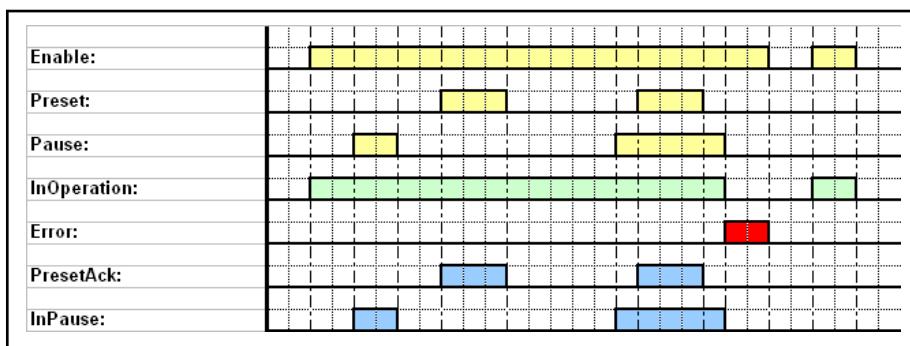


Abb. 8-11: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_DiameterMeasurementType01

Funktionsbeschreibung

Der Funktionsbaustein MB_DiameterMeasurementType01 liest einen gemessenen Durchmesserwert (z. B. mit einem Ultraschallsensor) ein und sorgt für die richtige Geschwindigkeitsanpassung der jeweiligen Wickelsachse bezogen auf den gemessenen Durchmesser.

Der Baustein wird in den Bausteinen [MB_WinderDancerCtrlType01](#), Seite 525, und [MB_WinderTensionCtrlType02](#), Seite 542, verwendet.



Das Verhältnis "Formatlänge"/"CoreDiameter" darf nicht größer als 25 sein, ansonsten reicht die Getriebenanpassung über den Getriebefeinabgleich nicht aus!

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_DiameterMeasurementType01 aufgelistet und beschrieben.

Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

Zusätzlich können die Fehlercodes des Bausteins MB_GetCyclicParameterHandle, ML_TechMotionBase, auftreten.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1180	16#0020	Fehler in der Statemachine
INPUT_RANGE_ERROR	16#1181	16#0001	HighLimit < LowLimit
INPUT_RANGE_ERROR	16#1181	16#0002	FormatLength < 0.0
INPUT_RANGE_ERROR	16#1181	16#0003	Webwidth < 0.0

ErrorID	Additional1	Additional2	Text
INPUT_RANGE_ERROR	16#1181	16#0004	Density < 0.0
INPUT_RANGE_ERROR	16#1181	16#0005	CoreDiameter < 0.0
INPUT_RANGE_ERROR	16#1181	16#0008	MinDiameter < 0.0
INPUT_RANGE_ERROR	16#1181	16#0009	MaxDiameter < 0.0
INPUT_RANGE_ERROR	16#1181	16#000A	HighLimit < 0.0
INPUT_RANGE_ERROR	16#1181	16#000B	Lowlimit < 0.0
INPUT_RANGE_ERROR	16#1181	16#000C	falsche Slave-Achs Nummer
DEVICE_ERROR	16#1182	16#0001	kein Betriebsmodus (sercos Phase 4)
RESSOURCE_ERROR	16#0004	16#0000	falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert

Tab. 8-21: Fehlercodes FB MB_DiameterMeasurementType01

Notwendige Hardware	• CMLx, XMx, VPx • IndraDrive
Notwendige Firmware	• MLC Firmware FWA-CMLXX*-MLC-14VRS XX - Gerätevariante, z. B. CML65 • IndraDrive Antriebsfirmware
Notwendige Software	• IndraWorks 14VRS für MLC bzw. MLD
Besonderheiten für MLC	Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter "S-0-0121 Lastgetriebe-Eingangsumdrehungen" und "S-0-0122 Lastgetriebe-Ausgangsumdrehungen" eingetragen werden. Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z. B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht! Interpolation im Antrieb: Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss der Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. Interpolation in der Steuerung: Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss kein Parameter eingetragen sein. Die Zielgröße ist der Steuerungsparameter "A-0-2605 ELS, Feinabgleich der Getriebeübersetzung, Prozessregler".
Besonderheiten für MLD	Bei Verwendung der Steuerung MLD-S mit übergeordneter Steuerung MLC muss für die Wickelachse Interpolation im Antrieb aktiviert sein (reale Achsen → entspr. Achse → Eigenschaften). Bei Verwendung der Steuerung MLD-M gibt es diese Einstellung nicht. Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Im zyklischen Kanal zwischen Steuerung und Antrieb des entsprechenden Antriebs muss der Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" eingetragen sein ("UserCmdDataX").

Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. (Der Dialog ist im Projektbaum unterhalb des MLD-Knotens → AxisData.)

Die "AxisData-Struktur" des Antriebes muss parametrierbar sein ("P-0-1367 SPS Konfiguration, Bit 6 ist gesetzt", bzw. Dialog SPS-Konfiguration)

Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter "S-0-0121 Lastgetriebe-Eingangsumdrehungen" und "S-0-0122 Lastgetriebe-Ausgangsumdrehungen" eingetragen werden.

Achsreferenz für die Masterachse:

Für die Technologiebausteine wird als Achsreferenz für die Masterachse empfohlen:

- Für eine virtuelle Leitachse die Achsreferenz "**VmAxis1**" verwenden.
Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 20 (VMA_1) eingestellt werden.
- Für eine reale Leitachse die Achsreferenz "**RmAxis1**" verwenden.
Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 23 (RMA_1) eingestellt werden.
Zusätzlich muss in diesem Fall noch die logische Adresse des Messgebers im Parameter "P-0-1820.0.2 CCD: Quelle des Messgeber-Lageistwertes" eingetragen werden.

Detaillierte Informationen können der Funktionsbeschreibung MLD entnommen werden.

8.5 Wickelrechner mit Tänzer

8.5.1 Einführung und Übersicht

Der Wickelrechner mit Tänzer wird an Zentrumswicklern eingesetzt. Die Zugkraft wird durch eine Kraft vorgegeben, die am frei beweglichen Teil des Tänzers angreift. Diese Kraft kann durch eine Gewichtskraft, eine Feder oder durch Pneumatik aufgebracht werden.

Die Regelung der Tänzerlage erfolgt im Tänzerlageregler. Der Istwert der Tänzerlage wird z.B. über einen Analogkanal erfasst und der Regelung zugeführt.

Der aktuelle Wickeldurchmesser wird vom intern eingebundenen Wickelrechner berechnet. Mit Hilfe des berechneten Wickeldurchmessers wird die Wickelachsgeschwindigkeit nachgeführt. Somit ersetzt der berechnete Wickeldurchmesser einen über eine Sensorik erfassten realen Istdurchmesser.

Der Wickelrechner mit Tänzer ist somit eine Kombination aus einem Wickler, dargestellt durch die Wickelachse und den Durchmesserrechner, und einem Tänzerlageregler. Der Durchmesserrechner wirkt auf den Getriebefeinabgleich und der Tänzerlageregler auf den additiven Drehzahlsollwert.

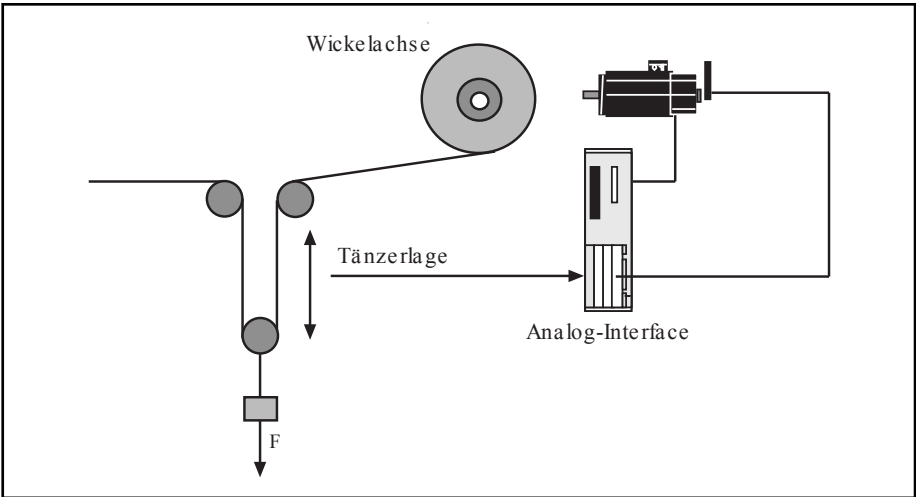


Abb. 8-12: Prinzip des Wickelrechners mit Tänzerlageregler

8.5.2 MB_WinderDancerCtrlType01

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechWinder
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechWinder.lib

Tab. 8-22: Referenztablelle MB_WinderDancerCtrlType01

Schnittstellenbeschreibung

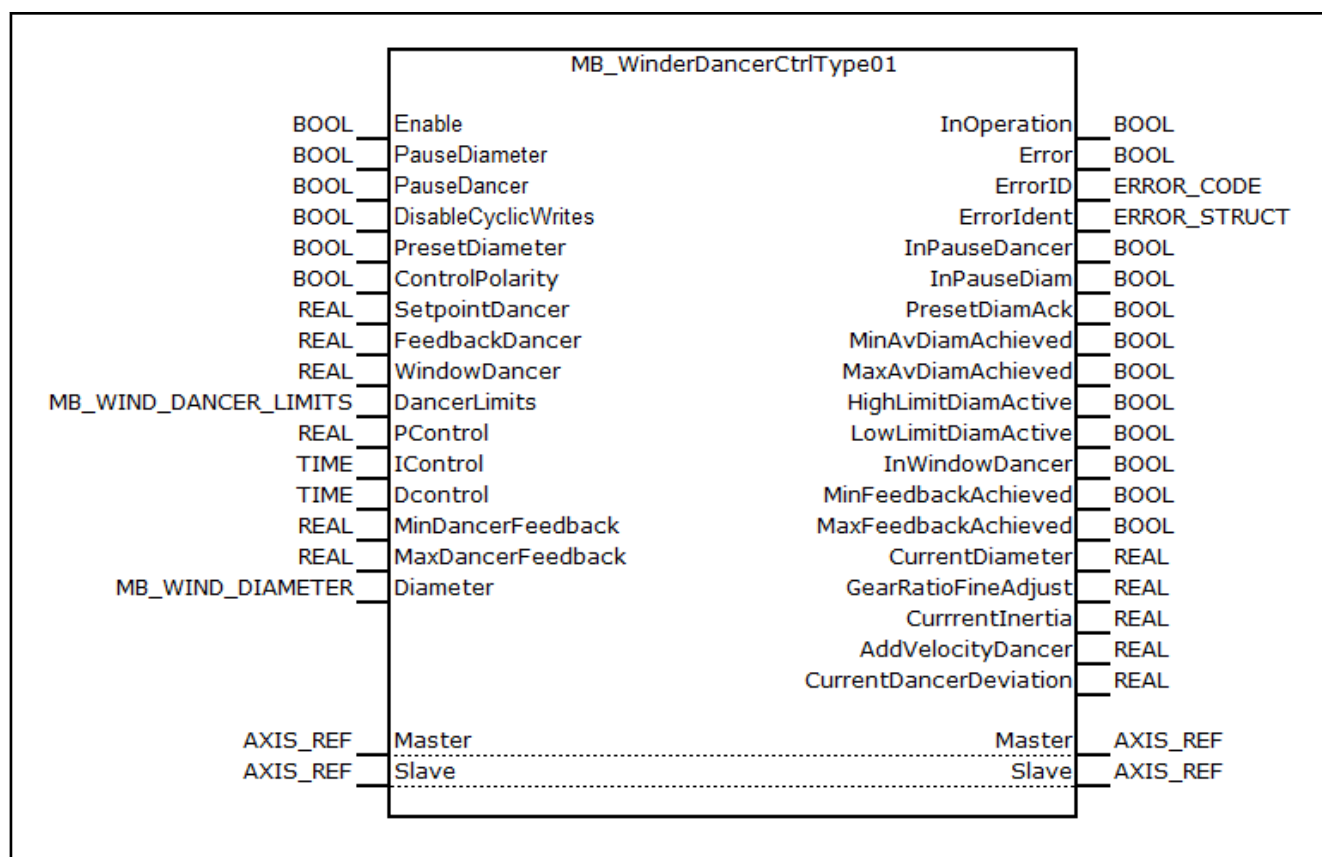


Abb. 8-13: Funktionsbaustein MB_WinderDancerCtrlType01

I/O-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	PauseDiameter	BOOL	Durchmesserrechner pausiert, keine weitere Durchmesserberechnung und Geschwindigkeitsanpassung (pegelgesteuert)
	PauseDancer	BOOL	Tänzerregler pausiert (pegelgesteuert)
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	PresetDiameter	BOOL	CurrentDiameter wird auf PresetVal gesetzt, GearRatioFineAdj entsprechend angepasst (flankengesteuert)
	ControlPolarity	BOOL	Polarität der Tänzerregelung wird invertiert (Übernahme bei pos. Flanke von Enable)
	SetpointDancer	REAL	Sollwert für die Tänzerregelung
	FeedbackDancer	REAL	Istwert der Tänzerlage
	WindowDancer	REAL	Fenster zur Überwachung des Tänzerlageistwertes [%] bezogen auf den Tänzerlagesollwert. Entfernt sich der Tänzer vom Lagesollwert um mehr als den angegebenen Wert, meldet dies der Baustein durch Rücksetzen des Ausganges "InWindow"

I/O-Typ	Name	Typ	Kommentar
	DancerLimits	MB_WIND_DANCER_LIMITS	Struktur für die automatische Stellgrößenbegrenzung des Tänzers MB_WIND_DANCER_LIMITS, Seite 615
	PControl	REAL	Wert für die Kp-Verstärkung der Tänzerregelung
	IControl	TIME	Wert für die Nachstellzeit der Tänzerregelung
	DControl	TIME	Wert für die Vorhaltezeit der Tänzerregelung
	MinDancerFeedback	REAL	Untere Schwelle für den Tänzeristwert (nur für Anzeige mit binärem Ausgang)
	MaxDancerFeedback	REAL	Obere Schwelle für den Tänzeristwert (nur für Anzeige mit binärem Ausgang)
	Diameter	MB_WIND_DIAMETER	Struktur für Durchmesserrechner MB_WIND_DIAMETER, Seite 616
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose Fehlerbehandlung, Seite 530
	InPauseDancer	BOOL	Tänzerregler pausiert
	InPauseDiam	BOOL	Durchmesserrechner pausiert
	PresetDiamAck	BOOL	Presetdurchmesser wurde übernommen (ist gesetzt, solange Preset ansteht)
	MinAvDiamAchieved	BOOL	Minimaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	MaxAvDiamAchieved	BOOL	Maximaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	HighLimitDiamActive	BOOL	Aktuell berechneter Durchmesser hat obere Grenze erreicht (Durchmesserwert wird verworfen)
	LowLimitDiamActive	BOOL	Aktuell berechneter Durchmesser hat untere Grenze erreicht (Durchmesserwert wird verworfen)
	InWindowDancer	BOOL	Tänzeristwert innerhalb des Tänzerfensters
	MinFeedbackAchieved	BOOL	Istwert der Tänzerlage ist unterhalb des Eingang Min-DancerFeedback
	MaxFeedbackAchieved	BOOL	Istwert der Tänzerlage ist oberhalb des Eingang Max-DancerFeedback
	CurrentDiameter	REAL	Aktueller Wickeldurchmesser
	GearRatioFineAdjust	REAL	Getriebefeinabgleich zur Drehzahlachsführung des Antriebes [%]

I/O-Typ	Name	Typ	Kommentar
	CurrentInertia	REAL	Aktuelles Trägheitsmoment des Wickelgutes + Mechanik (bezogen auf die Motorseite) [kg m²]
	AddVelocityDancer	REAL	Drehzahloffset für den Wickelantrieb [U/min]
	CurrentDancerDeviation	REAL	Aktuelle Regelabweichung des Tänzerreglers
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (Wickelachse)

Tab. 8-23: Schnittstellenvariablen des FBs MB_WinderDancerCtrlType01

Min-/Max- und Standardwerte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Standardwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standardwert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
PauseDiameter	BOOL			FALSE	Kontinuierlich
PauseDancer	BOOL			FALSE	Kontinuierlich
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
PresetDiameter	BOOL			FALSE	Kontinuierlich
ControlPolarity	BOOL			FALSE	Steigende Flanke an "Enable"
SetpointDancer	REAL	0.0	n. def.	0.0	Kontinuierlich
FeedbackDancer	REAL	n. def.	n. def.	0.0	Kontinuierlich
WindowDancer	REAL	0.0	n. def.	0.0	Kontinuierlich
DancerLimits	STRUCT				
PControl	REAL	0.0	n. def.	0.0	Kontinuierlich
IControl	TIME	0s	n. def.	0s	Kontinuierlich
DControl	TIME	0s	n. def.	0s	Kontinuierlich
MinDancerFeedback	REAL	0.0	MaxDancerFeedback	0.0	Kontinuierlich
MaxDancerFeedback	REAL	MinDancerFeedback	n. def.	0.0	Kontinuierlich
Diameter	STRUCT				Je nach Strukturelement

Tab. 8-24: Min-/Max- und Standardwerte der Eingänge MB_WinderDancerCtrlType01

Funktionsbeschreibung**Durchmesserberechnung**

Die Durchmesserberechnung stellt über den zyklisch aufgerufenen Funktionsbaustein [MB_DiameterCalculatorType03, Seite 509](#), den aktuellen, berechneten Durchmesser zur Verfügung. Abhängig vom Durchmesser wird der additive Feinabgleich der Wickelachse eingestellt. Diese wird im Verhältnis

zur Referenzachse schneller oder langsamer bewegt, um eine konstante Zugkraft der Stoffbahn zu erzielen.

Eine Relativbewegung der Tänzerwalze wird bei der Durchmesserberechnung mit Hilfe des Einganges "DancerConstant" berücksichtigt. Hierzu ist diese Tänzerkonstante einmalig zu ermitteln.

Folgende Vorgehensweise wird empfohlen:

1. Bei Tänzerauslenkung von der einen Endlage in die andere Endlage, den Bahnweg ermitteln (Einheit in mm).
2. Differenz zwischen den beiden Tänzeristwerten der jeweiligen Endlage ermitteln.
3. Quotienten der beiden Werte berechnen.
4. Vorzeichen der Tänzerkonstante durch praktischen Versuch bestimmen. Hierzu wird bei langsam drehender Wickel- u. Referenzachse die Durchmesserberechnung beobachtet. Sollwertsprünge auf den Tänzer Sollwert sollten bei richtigem Vorzeichen keinen signifikanten Einfluss auf die Durchmesserberechnung haben.
5. Berechneten Wert für die Tänzerkonstante wie im Schritt 4, wenn nötig, iterativ "nachoptimieren".



Soll der aktuelle Durchmesser, nach dem Ausschalten der Steuerung erhalten bleiben, muss der Ausgang "CurrentDiameter" auf eine remanente Variable kopiert werden. Mit diesem Wert kann nach dem Einschalten wieder ein Preset ausgeführt werden.

Rollenwechsel

Nach einem Rollenwechsel ändert sich in der Regel der Wickeldurchmesser schlagartig. Der zuletzt berechnete interne Wert für den Wickeldurchmesser entspricht dann nicht mehr dem aktuellen Durchmesser. Damit die Durchmesserberechnung korrekt aufsetzen kann, ist ein Preset des Wickeldurchmessers sinnvoll. Der Eingangswert für den Preset sollte annähernd dem tatsächlichen Wert des Wickeldurchmessers entsprechen, um einen möglichst genauen resultierenden Feinabgleich zu erzielen.



Wird der Preset nicht durchgeführt, kann es zu Fehlfunktionen kommen. Daher sollte der Preset durchgeführt werden, bevor der Durchmesserrechner wieder aktiv wird.

Bei einem fliegenden Rollenwechsel muss sichergestellt werden, dass die Umfangsgeschwindigkeit der neuen Rolle, der aktuellen Rolle entspricht. Dies ist durch ein entsprechendes SPS-Projekt in IndraLogic sicherzustellen. Beim Zuschalten des Durchmesserrechners für die neue Rolle können Sie mit dem Eingang "Pause" arbeiten und nach dem Abschlagen der alten Rolle "Pause" wieder wegnehmen.

Tänzerlageregelung

Zur Erfassung der Tänzerposition wird ein Messglied benötigt, das die Tänzerlage in ein proportionales Signal umsetzt. Die Ist-Position des Tänzers wird von einem gegebenen Sollwert abgezogen und als Regelabweichung auf einen internen PID-Regler gegeben. Die Stellgröße des Reglers wirkt auf den Drehzahlsollwert mittels Addition. Somit dreht sich die Wickelachse im Vergleich zur Leitachse schneller oder langsamer. Hiermit wird die Zugkraft des Tänzers geregelt ([Wickelrechner mit Zugkraftsteuerung bzw. Zugkraftregelung, Seite 541](#)).

Unter normalen Bedingungen (z. B. keine hohen Drehzahlsollwertsprünge oder plötzliche Durchmesseränderungen des Wickels) erfolgt die Grobeinstellung der Umfangsgeschwindigkeit der Wickelachse über den Getriebefeinabgleich. Hierzu wird der zuvor beschriebene Durchmesserrechner einge-

**Drehrichtung der Wickelachse,
Regelsinn der Tänzerregelung**

setzt. Der Tänzerlagereger muss somit lediglich kleine Abweichungen ausregeln.

Die Drehrichtung der Zug- bzw. Wickelachse kann mit Hilfe des Antriebsparameters "P-0-0108 Polarität Leittrieb" bzw. des Achsparameters "A-0-2798 Polarität der Leitachsposition" eingestellt werden. Dreht sich die Wickelachse nicht in die gewünschte Richtung, ist der Parameter P-0-0108 bzw. der Achsparameter A-0-2798 entsprechend umzustellen. Bei einer Invertierung der Drehrichtung des Antriebes ist auch der Regelsinn (ControlPolarity) zu invertieren.

Funktionsdiagramm des Tänzerlageregers

Anhand des folgenden Funktionsdiagramms wird die Funktionsweise des Wickelrechners mit Tänzer schematisch erläutert.

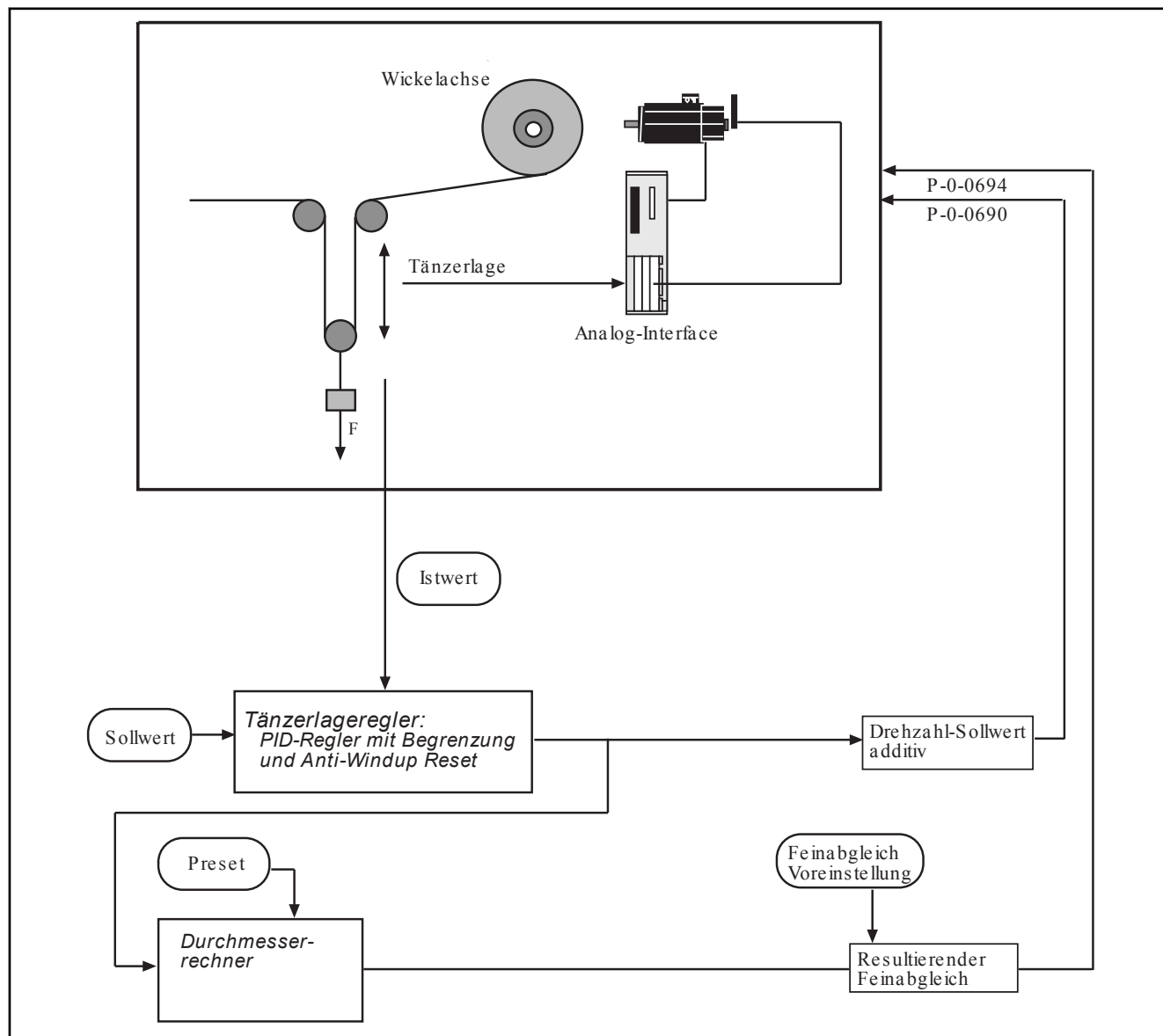


Abb. 8-14: Funktionsdiagramm des Wickelrechners mit Tänzer

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_WinderDancerCtrlType01 aufgelistet und beschrieben.

Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

Zusätzlich können die Fehlercodes der Bausteine MB_DancerControlType03, MB_DiameterCalculatorType03, MB_DiameterMeasurementType01, IL_PID-Type01, RIL_LoopControl, sowie MB_GetCyclicParameterHandle, ML_Tech-MotionBase, auftreten.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1190	16#0020	Fehler in der Statemachine

Tab. 8-25: Fehlercodes FB MB_WinderDancerCtrlType01

Notwendige Hardware	• CMLx, XMx, VPx • IndraDrive
Notwendige Firmware	• MLC Firmware FWA-CMLXX*-MLC-14VRS XX - Gerätevariante, z. B. CML65 • IndraDrive Antriebsfirmware
Notwendige Software	• IndraWorks 14VRS für MLC bzw. MLD
Besonderheiten für MLC	Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter ("S-0-0121 Lastgetriebe-Eingangsumdrehungen") und ("S-0-0122 Lastgetriebe-Ausgangsumdrehungen") eingetragen werden. Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z. B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht! Interpolation im Antrieb: Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss der Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" und Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. Interpolation in der Steuerung: Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss kein Parameter eingetragen sein. Die Zielgrößen sind die Steuerungsparameter "A-0-2605 ELS, Feinabgleich der Getriebeübersetzung, Prozessregler" und "A-0-2615 ELS, Geschwindigkeits-Sollwert additiv, Prozessregler".
Besonderheiten für MLD	Bei Verwendung der Steuerung MLD-S mit übergeordneter Steuerung MLC muss für die Wickelachse Interpolation im Antrieb aktiviert sein (reale Achsen ▶ entspr. Achse ▶ Eigenschaften). Bei Verwendung der Steuerung MLD-M gibt es diese Einstellung nicht. Die Wichtungsart für die Master-Achse und Slave-Achse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Im zyklischen Sercos-Kanal des entsprechenden Antriebs müssen der Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" und "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus (der Dialog ist im Projektbaum unterhalb des MLD-Knotens Axis-Data). Die "AxisData-Struktur" des Antriebes muss parametrierbar sein ("P-0-1367 SPS Konfiguration, Bit 6 ist gesetzt", bzw. Dialog SPS-Konfiguration).

Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter "S-0-0121 Lastgetriebe-Eingangsumdrehungen" und "S-0-0122 Lastgetriebe-Ausgangsumdrehungen" eingetragen werden.

Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z. B. 0.1 U/min).

Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!

Inbetriebnahme des Wickelrechners mit Tänzer

Zur Inbetriebnahme des Wickelrechners mit Tänzer siehe [Kap. 8.5.4 "Inbetriebnahme des Wickelrechners mit Tänzer"](#) auf Seite 539.

8.5.3 MB_WinderDancerCtrlType02

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechWinder
MLD ab MPx20 mit MA Funktionspaket	MX_TechWinder

Tab. 8-26: Referenztable MB_WinderDancerCtrlType02

Schnittstellenbeschreibung

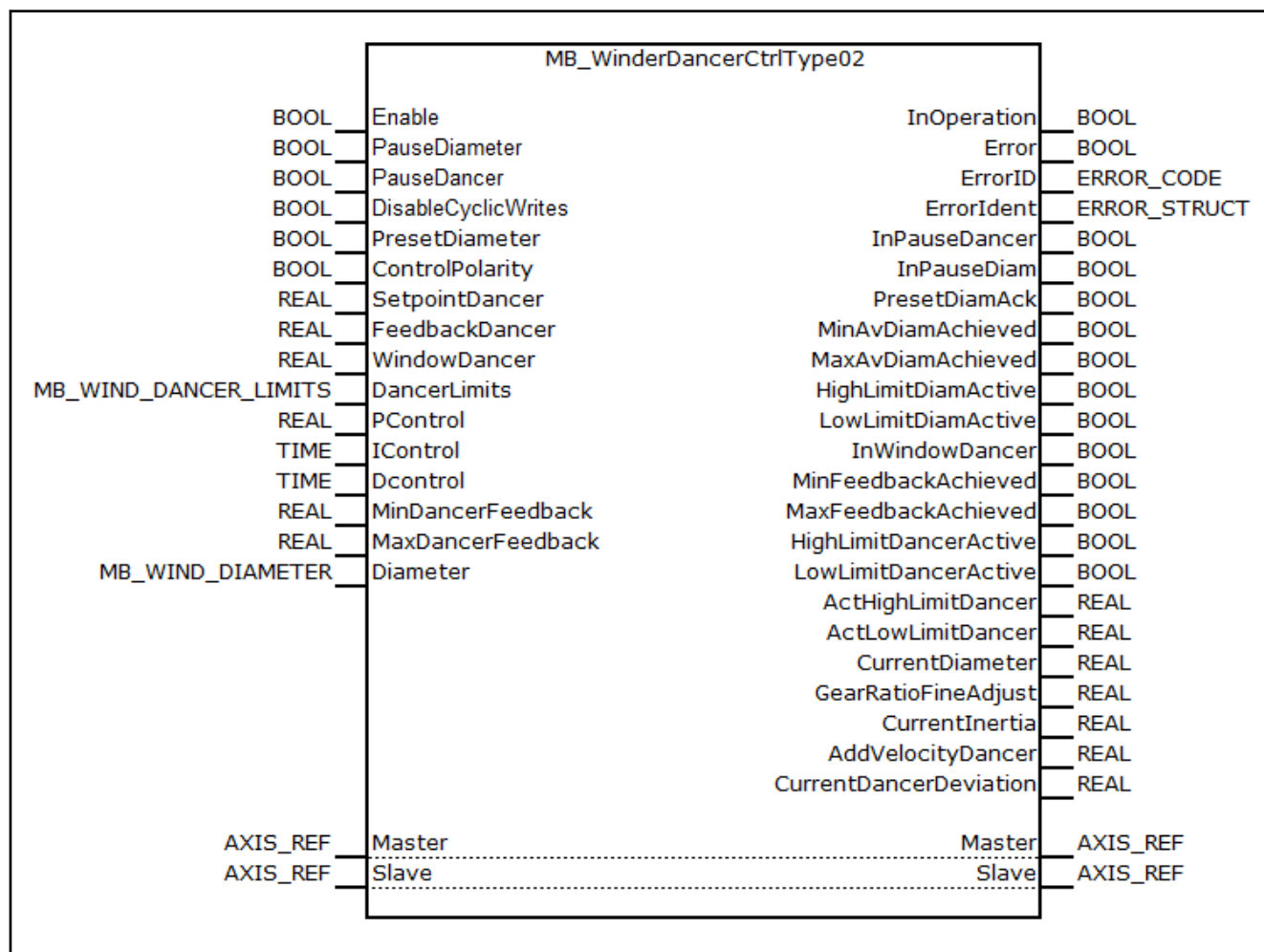


Abb. 8-15: Funktionsbaustein MB_WinderDancerCtrlType02

I/O-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	PauseDiameter	BOOL	Durchmesserrechner pausiert, keine weitere Durchmesserberechnung und Geschwindigkeitsanpassung (pegelgesteuert)
	PauseDancer	BOOL	Tänzerregler pausiert (pegelgesteuert)
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	PresetDiameter	BOOL	CurrentDiameter wird auf PresetVal gesetzt, GearRatioFineAdj entsprechend angepasst (flankengesteuert)
	ControlPolarity	BOOL	Polarität der Tänzerregelung wird invertiert (Übernahme bei pos. Flanke von Enable)
	SetpointDancer	REAL	Sollwert für die Tänzerregelung
	FeedbackDancer	REAL	Istwert der Tänzerlage
	WindowDancer	REAL	Fenster zur Überwachung des Tänzerlageistwertes [%] bezogen auf den Tänzerlagesollwert. Entfernt sich der Tänzer vom Lagesollwert um mehr als den angegebenen Wert, meldet dies der Baustein durch Rücksetzen des Ausganges "InWindow"
	DancerLimits	MB_WIND_DANCER_LIMITS	Struktur für die automatische Stellgrößenbegrenzung des Tänzers MB_WIND_DANCER_LIMITS, Seite 615
	PControl	REAL	Wert für die Kp-Verstärkung der Tänzerregelung
	IControl	TIME	Wert für die Nachstellzeit der Tänzerregelung
	DControl	TIME	Wert für die Vorhaltezeit der Tänzerregelung
	MinDancerFeedback	REAL	Untere Schwelle für den Tänzeristwert (nur für Anzeige mit binärem Ausgang)
	MaxDancerFeedback	REAL	Obere Schwelle für den Tänzeristwert (nur für Anzeige mit binärem Ausgang)
	Diameter	MB_WIND_DIAMETER	Struktur für Durchmesserrechner MB_WIND_DIAMETER, Seite 616
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose Fehlerbehandlung, Seite 537
	InPauseDancer	BOOL	Tänzerregler pausiert
	InPauseDiam	BOOL	Durchmesserrechner pausiert
	PresetDiamAck	BOOL	Presetdurchmesser wurde übernommen (ist gesetzt, solange Preset ansteht)

I/O-Typ	Name	Typ	Kommentar
	MinAvDiamAchieved	BOOL	Minimaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	MaxAvDiamAchieved	BOOL	Maximaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	HighLimitDiamActive	BOOL	Aktuell berechneter Durchmesser hat obere Grenze erreicht (Durchmesserwert wird verworfen)
	LowLimitDiamActive	BOOL	Aktuell berechneter Durchmesser hat untere Grenze erreicht (Durchmesserwert wird verworfen)
	InWindowDancer	BOOL	Tänzeristwert innerhalb des Tänzerfensters
	MinFeedbackAchieved	BOOL	Istwert der Tänzerlage ist unterhalb des Eingang Min-DancerFeedback
	MaxFeedbackAchieved	BOOL	Istwert der Tänzerlage ist oberhalb des Eingang Max-DancerFeedback
	HighLimitDancerActive	BOOL	Begrenzung des Drehzahloffsets "HighLimit " ist aktiv
	LowLimitDancerActive	BOOL	Begrenzung Drehzahloffsets "LowLimit" ist aktiv
	ActHighLimitDancer	REAL	Aktueller Wert für die obere Begrenzung des Drehzahloffsets
	ActLowLimitDancer	REAL	Aktueller Wert für die untere Begrenzung des Drehzahloffsets
	CurrentDiameter	REAL	Aktueller Wickeldurchmesser
	GearRatioFineAdjust	REAL	Getriebefeinabgleich zur Drehzahlachsführung des Antriebes [%]
	CurrentInertia	REAL	Aktuelles Trägheitsmoment des Wickelgutes + Mechanik (bezogen auf die Motorseite) [kg m ²]
	AddVelocityDancer	REAL	Drehzahloffset für den Wickelantrieb [U/min]
	CurrentDancerDeviation	REAL	Aktuelle Regelabweichung des Tänzerreglers
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (Wickelachse)

Tab. 8-27: Schnittstellenvariablen FB MB_WinderDancerCtrlType02

Min-/Max- und Standardwerte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Standardwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standardwert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
PauseDiameter	BOOL			FALSE	Kontinuierlich
PauseDancer	BOOL			FALSE	Kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Standardwert	Übernahme
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
PresetDiameter	BOOL			FALSE	Kontinuierlich
ControlPolarity	BOOL			FALSE	Steigende Flanke an "Enable"
SetpointDancer	REAL	0.0	n. def.	0.0	Kontinuierlich
FeedbackDancer	REAL	n. def.	n. def.	0.0	Kontinuierlich
WindowDancer	REAL	0.0	n. def.	0.0	Kontinuierlich
DancerLimits	STRUCT				
PControl	REAL	0.0	n. def.	0.0	Kontinuierlich
IControl	TIME	0s	n. def.	0s	Kontinuierlich
DControl	TIME	0s	n. def.	0s	Kontinuierlich
MinDancerFeed-back	REAL	0.0	MaxDancerFeed-back	0.0	Kontinuierlich
MaxDancerFeed-back	REAL	MinDancerFeed-back	n. def.	0.0	Kontinuierlich
Diameter	STRUCT				Je nach Strukturelement

Tab. 8-28: Min-/Max- und Standardwerte der Eingänge MB_WinderDancerCtrl-Type02

Funktionsbeschreibung

Durchmesserberechnung

Die Durchmesserberechnung stellt über den zyklisch aufgerufenen Funktionsbaustein MB_DiameterCalculatorType03 den aktuellen, berechneten Durchmesser zur Verfügung. Abhängig vom Durchmesser wird der additive Feinabgleich der Wickelachse eingestellt. Diese wird im Verhältnis zur Referenzachse schneller oder langsamer bewegt, um eine konstante Zugkraft der Stoffbahn zu erzielen.

Eine Relativbewegung der Tänzerwalze wird bei der Durchmesserberechnung mit Hilfe des Einganges "DancerConstant" berücksichtigt. Hierzu ist diese Tänzerkonstante einmalig zu ermitteln.

Folgende Vorgehensweise wird empfohlen:

1. Bei TänzerAuslenkung von der einen Endlage in die andere Endlage, den Bahnweg ermitteln (Einheit in mm).
2. Differenz zwischen den beiden Tänzeristwerten der jeweiligen Endlage ermitteln.
3. Quotienten der beiden Werte berechnen.
4. Vorzeichen der Tänzerkonstante durch praktischen Versuch bestimmen. Hierzu wird bei langsam drehender Wickel- u. Referenzachse die Durchmesserberechnung beobachtet. Sollwertsprünge auf den Tänzer-sollwert sollten bei richtigem Vorzeichen keinen signifikanten Einfluss auf die Durchmesserberechnung haben.
5. Berechneten Wert für die Tänzerkonstante wie im Schritt 4, wenn nötig, iterativ "nachoptimieren".



Soll der aktuelle Durchmesser, nach dem Ausschalten der Steuerung erhalten bleiben, muss der Ausgang "CurrentDiameter" auf eine remanente Variable kopiert werden. Mit diesem Wert kann nach dem Einschalten wieder ein Preset ausgeführt werden.

Rollenwechsel

Nach einem Rollenwechsel ändert sich in der Regel der Wickeldurchmesser schlagartig. Der zuletzt berechnete interne Wert für den Wickeldurchmesser entspricht dann nicht mehr dem aktuellen Durchmesser. Damit die Durchmesserberechnung korrekt aufsetzen kann, ist ein Preset des Wickeldurchmessers sinnvoll. Der Eingangswert für den Preset sollte annähernd dem tatsächlichen Wert des Wickeldurchmessers entsprechen, um einen möglichst genauen resultierenden Feinabgleich zu erzielen.



Wird der Preset nicht durchgeführt, kann es zu Fehlfunktionen kommen. Daher sollte der Preset durchgeführt werden, bevor der Durchmesserrechner wieder aktiv wird.

Bei einem fliegenden Rollenwechsel muss sichergestellt werden, dass die Umfangsgeschwindigkeit der neuen Rolle, der aktuellen Rolle entspricht. Dies ist durch ein entsprechendes SPS-Projekt in IndraLogic sicherzustellen. Beim Zuschalten des Durchmesserrechners für die neue Rolle kann mit dem Eingang "Pause" gearbeitet werden und nach dem Abschlagen der alten Rolle "Pause" wieder weggenommen werden.

Tänzerlageregelung

Zur Erfassung der Tänzerposition wird ein Messglied benötigt, das die Tänzerlage in ein proportionales Signal umsetzt. Die Ist-Position des Tänzers wird von einem gegebenen Sollwert abgezogen und als Regelabweichung auf einen internen PID-Regler gegeben. Die Stellgröße des Reglers wirkt auf den Drehzahlsollwert mittels Addition. Somit dreht sich die Wickelachse im Vergleich zur Leitachse schneller oder langsamer. Hiermit wird die Zugkraft des Tänzers geregelt (vgl. [Wickelrechner mit Zugkraftsteuerung bzw. Zugkraftregelung, Seite 541](#)).

Unter normalen Bedingungen (z. B. keine hohen Drehzahlsollwertsprünge oder plötzliche Durchmesseränderungen des Wickels) erfolgt die Grobeinstellung der Umfangsgeschwindigkeit der Wickelachse über den Getriebefeinabgleich. Hierzu wird der zuvor beschriebene Durchmesserrechner eingesetzt. Der Tänzerlageregler muss somit lediglich kleine Abweichungen ausregeln.

Drehrichtung der Wickelachse, Regelsinn der Tänzerregelung

Die Drehrichtung der Zug- bzw. Wickelachse kann mit Hilfe des Antriebsparameters "P-0-0108 Polarität Leitantrieb" bzw. des Achsparameters "A-0-2798 Polarität der Leitachseposition" eingestellt werden. Dreht sich die Wickelachse nicht in die gewünschte Richtung, ist der Parameter P-0-0108 bzw. der Achsparameter A-0-2798 entsprechend umzustellen. Bei einer Invertierung der Drehrichtung des Antriebes ist auch der Regelsinn (ControlPolarity) zu invertieren.

Funktionsdiagramm des Tänzerlagereglers

Anhand des folgenden Funktionsdiagramms wird die Funktionsweise des Wickelrechners mit Tänzer schematisch erläutert.

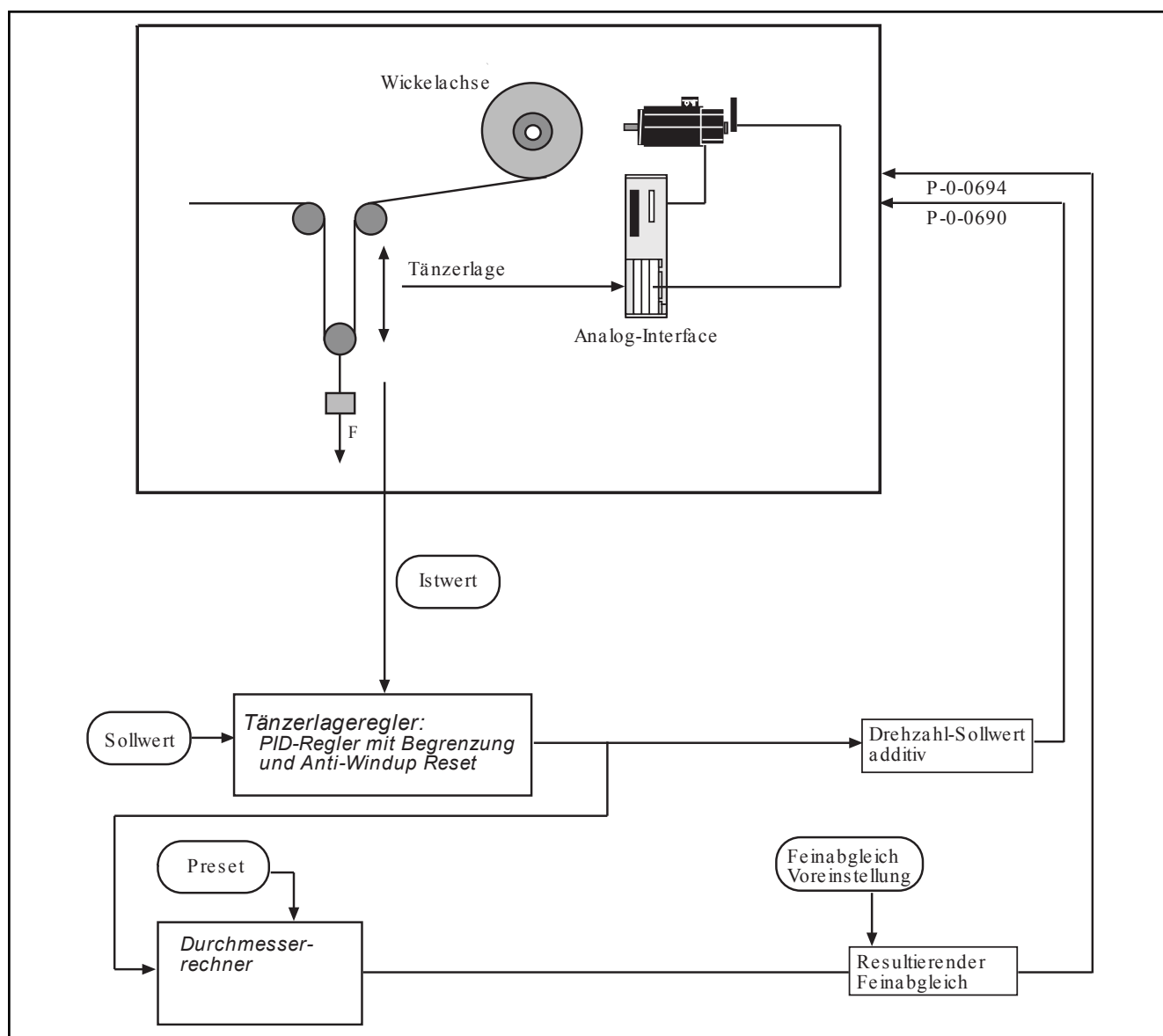


Abb. 8-16: Funktionsdiagramm des Wickelrechners mit Tänzer

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_WinderDancerCtrlType02 aufgelistet und beschrieben.

Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

Zusätzlich können die Fehlercodes der Bausteine MB_DancerControlType03, MB_DiameterCalculatorType03, MB_DiameterMeasurementType01, IL_PID-Type01, RIL_LoopControl, sowie MB_GetCyclicParameterHandle, ML_TechMotionBase, auftreten.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1190	16#0020	Fehler in der Statemachine

Tab. 8-29: Fehlercodes FB MB_WinderDancerCtrlType02

Notwendige Hardware

- CMLx, XMx, VPx
- IndraDrive

Notwendige Firmware	• MLC Firmware FWA-CMLXX*-MLC-14VRS XX - Gerätevariante, z. B. CML65 • IndraDrive Antriebsfirmware
Notwendige Software	• IndraWorks 14VRS für MLC
Besonderheiten für MLC	Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter ("S-0-0121 Lastgetriebe-Eingangsumdrehungen") und ("S-0-0122 Lastgetriebe-Ausgangsumdrehungen") eingetragen werden. Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z. B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht! Interpolation im Antrieb: Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss der Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" und Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. Interpolation in der Steuerung: Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss kein Parameter eingetragen sein. Die Zielgrößen sind die Steuerungsparameter "A-0-2605 ELS, Feinabgleich der Getriebeübersetzung, Prozessregler" und "A-0-2615 ELS, Geschwindigkeits-Sollwert additiv, Prozessregler". Besonderheiten für MLD Bei Verwendung der Steuerung MLD-S mit übergeordneter Steuerung MLC muss für die Wickelachse Interpolation im Antrieb aktiviert sein (reale Achsen ► entspr. Achse ► Eigenschaften). Bei Verwendung der Steuerung MLD-M gibt es diese Einstellung nicht. Die Wichtungsart für die Master-Achse und Slave-Achse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen. Im zyklischen Sercos-Kanal des entsprechenden Antriebs müssen der Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" und "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus. Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus (der Dialog befindet sich im Projektbaum unterhalb des MLD-Knotens → AxisData). Die "AxisData-Struktur" des Antriebes muss parametrierbar sein ("P-0-1367 SPS Konfiguration, Bit 6 ist gesetzt", bzw. Dialog SPS-Konfiguration). Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter "S-0-0121 Lastgetriebe-Eingangsumdrehungen" und "S-0-0122 Lastgetriebe-Ausgangsumdrehungen" eingetragen werden. Stillstandsfenster Referenzachse: Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z. B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht! Für die Referenzachse kann der Funktionsbaustein auf keinen Wert für das Stillstandsfenster zugreifen. Intern wird dann ein Wert von 1 U/min angenommen.

men. Ist dieser Wert für die Applikation nicht geeignet, kann dieser mittels folgender SPS-Codezeilen beeinflusst werden:

Deklaration

```
prMaStandstillWdw:    POINTER TO REAL; // for StandstillWindow master MLD
rTestMaStandstillWdw: REAL := 0.5;    // for StandstillWindow master MLD
```

Implementation

```
prMaStandstillWdw := ADR(BausteinInstanznameWickler.fbDiameterCalculator.rMaStandstillWindow);
prMaStandstillWdw^ := rTestMaStandstillWdw;
```

Der Wert für diese Variable bleibt nicht gespeichert, der Wert muss beim Einschalten der SPS also neu geschrieben werden.

Achsreferenz für die Masterachse:

Für die Technologiebausteine wird als Achsreferenz für die Masterachse empfohlen:

- Für eine virtuelle Leitachse die Achsreferenz "**VmAxis1**" verwenden.
Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 20 (VMA_1) eingestellt werden.
- Für eine reale Leitachse die Achsreferenz "**RmAxis1**" verwenden.
Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 23 (RMA_1) eingestellt werden.
Zusätzlich muss in diesem Fall noch die logische Adresse des Messgebers im Parameter "P-0-1820.0.2 CCD: Quelle des Messgeber-Lageistwertes" eingetragen werden.

Detaillierte Informationen können der Funktionsbeschreibung MLD entnommen werden.

Inbetriebnahme des Wickelrechners mit Tänzer

Zur Inbetriebnahme des Wickelrechners mit Tänzer siehe [Kap. 8.5.4 "Inbetriebnahme des Wickelrechners mit Tänzer"](#) auf Seite 539.

8.5.4 Inbetriebnahme des Wickelrechners mit Tänzer

Antriebsregelkreis

Es wird vorausgesetzt, dass der Drehzahlregelkreis des Wickelantriebes eingestellt wurde. D. h. die Parameter "S-0-0100 Geschwindigkeitsregler-Proportionalverstärkung" und "S-0-0101 Geschwindigkeitsregler-Nachstellzeit" müssen der Applikation entsprechend eingestellt sein.

Es wird eine Inbetriebnahme eines Wickelrechners mit Tänzer erläutert. Hierzu sind die folgenden Schritte zweckmäßig:

- Parametrierung des Tänzerlagereglers
- Parametrierung des Durchmesserrechners
- Erster Anlauf der Wickelachse

Kommandierung der Wicklerachse

Für die Wicklerachse muss die richtige Betriebsart des Antriebes, sowie die Reglerfreigabe des Antriebes gesetzt werden.

Das Setzen von Betriebsart und Reglerfreigabe kann auf zwei unterschiedliche Arten erfolgen:

- Setzen der Betriebsart "ModeSyncVel" mit Hilfe des Achsinterfaces (dies wird empfohlen)
- Benutzung der PLC-Open Funktionsbausteine "MC_Power", "MC_GearIn", "MC_Stop" zur Kommandierung der Achse

Parametrierung des Tänzerlagereglers

Da die Funktion des Wickelrechners mit Tänzer auf die Funktion des Tänzerlagereglers aufbaut, kann die Parametrierung des Tänzerlagereglers wie in [Tänzerlageregelung, Seite 493](#), beschrieben, durchgeführt werden.

Parametrierung des Durchmesserrechners

Es werden, wie zuvor beschrieben, die einzelnen Eingänge des Durchmesserrechners MB_DiameterCalculatorType03 beschaltet. Hierzu sind für die Inbetriebnahme des Wickelrechners mit Tänzer einige Besonderheiten zu beachten.

Aus der Parametrierung des Tänzerlagereglers sind bereits die Eingänge "FormatLength" und "FeedBackDancer" des Durchmesserrechners vorbelegt. Eine weitere, den Tänzer betreffende Größe, ist die Tänzerkonstante "DancerConstant". Diese gibt an, welchen Bahnweg die volle Auslenkung von negativen bis positiven Anschlag, bezogen auf die Einheit der Tänzerverstellung, des Tänzers beschreibt. Der Eingang "ReferenceDiameter" sollte mit dem aktuell berechneten Durchmesser "CurrentDiameter" verknüpft sein.

Da der berechnete Durchmesser über den resultierenden Feinabgleich direkt auf die Wickelachsgeschwindigkeit wirkt, ist zu Beginn der Berechnung bei Aktivierung des Wickelrechners ein sinnvoller Wert für den Durchmesser vorzugeben. Daher ist am Eingang "PresetDiameter" ein Wert für den Durchmesser anzugeben, der nahe dem des tatsächlichen Durchmessers liegt. Im Falle eines Aufwicklers kann der Wert etwas kleiner und für Abwickler etwas größer gewählt werden.

Schließlich können die Eingänge "MinDiameter" und "MaxDiameter" belegt werden. Die hier angegebenen Größen wirken auf die Ausgänge "MinAvDiamAchieved" und "MaxAvDiamAchieved". Die jeweiligen Werte sollten innerhalb des erlaubten Durchmesserbereichs liegen, der durch die Eingänge "HighLimitDiameter" und "LowLimitDiameter" definiert wird. So können z. B. für einen Rollenwechsel oder die Anpassung des "DiamCalcRange" anhand einer Vorgabe des minimalen/maximalen Durchmessers der Zeitpunkt für bestimmte Prozessschritte in der SPS detektiert und angezeigt werden.



Das Verhältnis "Formatlänge" / "CoreDiameter" darf nicht größer als 25 sein.

Ansonsten reicht die Getriebeanpassung über den Getriebefeinabgleich nicht aus!

Erster Anlauf der Wickelachse

Der Tänzerlageregler und der Wickelrechner werden in dieser Reihenfolge nacheinander aktiviert. Nach der Aktivierung der Tänzerlageregelung läuft der Tänzer in seine Sollposition. Wie zuvor beschrieben, sollte diese in etwa der Mittellage des Tänzers entsprechen.

Die Winkellagen von Wickel- und Referenzachse werden aufsummiert und abhängig vom Wert des Eingangs "DiamCalcRange" wird regelmäßig der Durchmesser berechnet und als aktueller Durchmesserwert am Ausgang "CurrentDiameter" angezeigt. Nach dem Einschalten des Durchmesserrechners erscheint an "CurrentDiameter" zunächst der Wert Null, bis das erste Mal ein Durchmesser berechnet worden ist. Soll ein anderer Wert als Null verwendet werden, muss ein Preset durchgeführt werden. Der Wert "PresetDiameter" wird bei einer positiven Flanke an "Preset" als aktueller Durchmesserwert übernommen.

Die Dauer zwischen zwei Berechnungen ist abhängig von der Wickelachsdrehzahl und dem Wert im Eingang "DiamCalcRange".

Der Wickelrechner wirkt auf den Getriebefeinabgleich. Der entsprechende Wert wird am Ausgang "GearRatioFineAdjust" ausgegeben.

8.6 Wickelrechner mit Zugkraftsteuerung bzw. Zugkraftregelung

8.6.1 Einführung und Übersicht

Der Funktionsbaustein stellt zwei verschiedene Wickelrechner zur Verfügung:

- Wickelrechner mit Zugkraftsteuerung
 - entspricht einem sensorlosen Wickelrechner
- Wickelrechner mit Zugkraftregelung
 - entspricht einem Wickelrechner mit Kraftmessdose

Die Wickelrechner sind im Funktionsbaustein MB_WinderTensionCtrlType02 implementiert.

Der Funktionsbaustein berechnet zyklisch den aktuellen Wickeldurchmesser, das aktuelle Trägheitsmoment des Wickels, die entsprechende Geschwindigkeitsnachführung über den Getriebefeinabgleich, einen additiven Drehzahloffset zum Spannen der Stoffbahn und die erforderliche Drehmomentgrenze.



Bei Anwendung dieses Wickelkonzeptes sollte die mechanische Reibung (Motorankopplung, Getriebe...) so gering wie möglich gehalten werden. Die mechanische Reibung sollte wenn möglich immer kleiner sein als das kleinste Zugmoment, das sich bei der minimalen Zugkraft und dem minimalen Wickeldurchmesser ergibt.

Wickelrechner mit Zugkraftsteuerung

Der Wickelrechner mit Zugkraftsteuerung wird an Zentrumswicklern eingesetzt. Der Antrieb der Wickelachse arbeitet generell an seiner Drehmomentgrenze. Die Drehmomentgrenze ist abhängig vom aktuellen Wickeldurchmesser und der eingestellten Zugkraft.

Die Zugkraft wird durch Einstellung des Drehmoments der Wickelachse erzeugt. Zusätzliche Messglieder werden nicht benötigt.

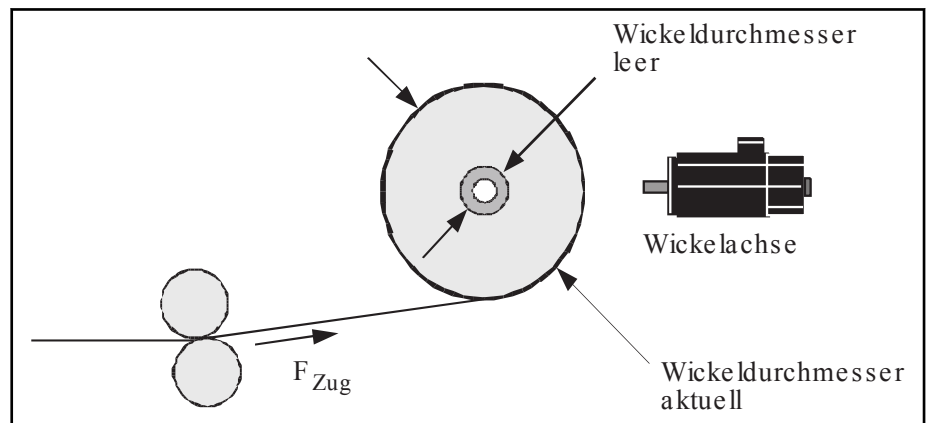


Abb. 8-17: Wickelrechner mit Zugkraftsteuerung

Wickelrechner mit Zugkraftregelung

Der Wickelrechner mit Zugkraftregelung wird an Zentrumswicklern eingesetzt. Der Antrieb der Wickelachse arbeitet generell an seiner Drehmomentgrenze. Die Drehmomentgrenze ist abhängig vom aktuellen Wickeldurchmesser und der eingestellten Zugkraft. Die Drehmomentgrenze entspricht dem Wert des Motormoments, das benötigt wird, um den Wickel mit der vorgege-

benen Drehzahl zu drehen. Es setzt sich aus dem Last-, dem Reib- und dem Beschleunigungsmoment zusammen.

Der Zugkraftregler wird als PI-Regler realisiert. Dieser beinhaltet den Funktionsbaustein IL_PIDType01 aus der Bibliothek RIL_LoopControl. Die Stellgröße des Reglers und der aktuelle Wickeldurchmesser bestimmen den Drehmomentgrenzwert des Antriebs. Der Sollwert der Zugkraft sollte zusätzlich gewichtet auf den Reglerausgang gegeben werden (PreControlWeighting ungleich Null). Bei einer Wichtung ungleich Null, besteht die Zugkraft für die Drehmomentbildung aus Reglerausgangsgröße addiert mit dem gewichteten Sollwert. Wichtung Null bedeutet, dass nur der Zugkraftregler wirkt. Der Zugkraftregler sollte daher nur additiv arbeiten (z. B. mit der Wichtung 1.0).

Der Zugkraftistwert wird direkt mittels einer Kraftmessdose gemessen und über einen Analogkanal oder über einen Feldbus dem Zugkraftregler als Istwert zugeführt. Bei gespannter Bahn kann der Durchmesserrechner den aktuellen Wickeldurchmesser berechnen.

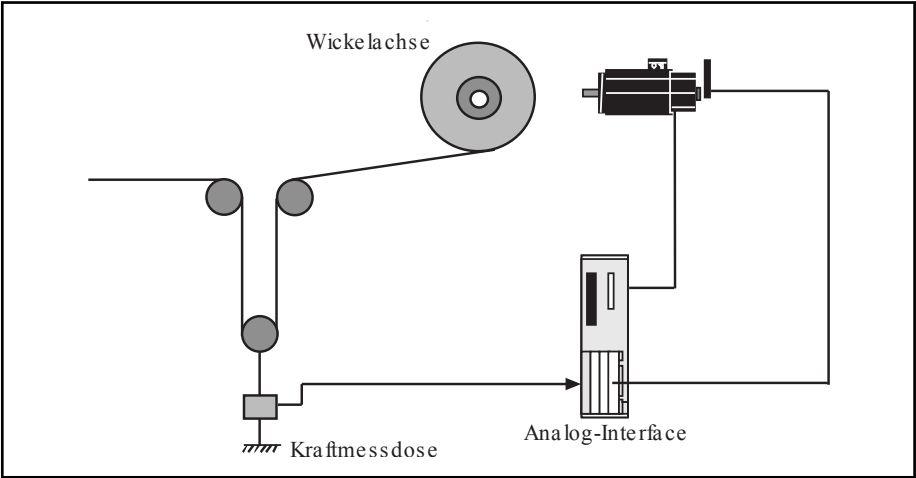


Abb. 8-18: Wickelrechner mit Zugkraftregelung

8.6.2

MB_WinderTensionCtrlType02

Kurzbeschreibung

Der Funktionsbaustein berechnet zyklisch den aktuellen Wickeldurchmesser, das aktuelle Trägheitsmoment des Wickels, die entsprechende Geschwindigkeitsnachführung über den Getriebefeinabgleich, einen additiven Drehzahloffset zum Spannen der Bahn und die erforderliche Drehmomentgrenze.

Zielgrößen des Funktionsbausteins sind bei Interpolation im Antrieb die Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler", "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" und "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar" bzw. diese Parameter müssen in den zyklischen Datencontainer gelegt werden.

Bei Interpolation in der Steuerung sind die Zielgrößen die Parameter "A-0-2605 ELS, Feinabgleich der Getriebeübersetzung, Prozessregler", "A-0-2615 ELS, Geschwindigkeits-Sollwert additiv, Prozessregler" und "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar".

Dann muss nur der Parameter S-0-0092 in den zyklischen Datencontainer gelegt werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechWinder

MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechWinder.lib
MLD ab MPx20 mit MA Funktionspaket	MX_TechWinder

Tab. 8-30: Referenztable MB_WinderTensionCtrlType02

Schnittstellenbeschreibung

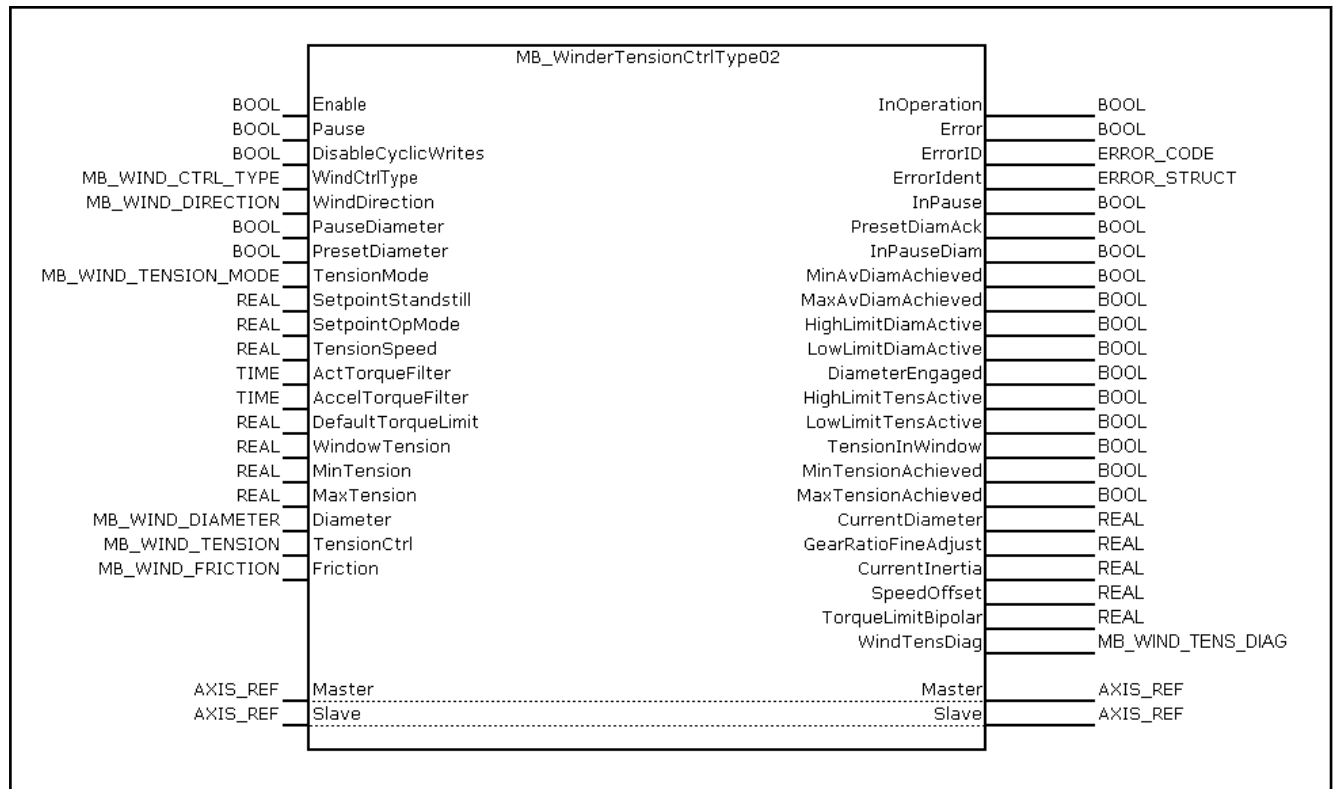


Abb. 8-19: Funktionsbaustein MB_WinderTensionCtrlType02

I/O-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	Pause	BOOL	Pausieren des Zugkraftreglers und des Wickelrechners (Werte werden eingefroren)
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	WindCtrlType	MB_WIND_CTRL_TYPE	<i>Wickelstruktur</i> - Zugkraftsteuerung (sensorlos): OPEN_LOOP (16#0000) - Zugkraftregelung (Messdose): CLOSED_LOOP (16#0001) MB_WIND_CTRL_TYPE , Seite 615

I/O-Typ	Name	Typ	Kommentar
	WindDirection	MB_WIND_DIRECTION	<i>Wickelrichtung</i> - Aufwickler REWIND (16#0000) - Abwickler UNWIND (16#0001) MB_WIND_DIRECTION , Seite 618
	PauseDiameter	BOOL	Anhalten der Durchmesserberechnung Regelung/Steuerung der Drehmomentgrenze bleibt in Betrieb
	PresetDiameter	BOOL	CurrentDiameter wird auf PresetVal gesetzt, GearRatioFineAdj entsprechend angepasst (flankengesteuert)
	TensionMode	MB_WIND_TENSION_MODE	<i>Modus zum Spannen der Materialbahn</i> - Spannen mit konstanter Drehzahl CONST_SPEED - Spannen mit konstanter Bahngeschwindigkeit CONST_WEB_VEL MB_WIND_TENSION_MODE , Seite 619
	SetpointStandstill	REAL	Vorgabe der Stillstandszugkraft [N] (Umschaltung zwischen Stillstands- und Laufzugkraft anhand der Stillstandsmeldung des Antriebes)
	SetpointOpMode	REAL	Vorgabe der Laufzugkraft [N] (Umschaltung zwischen Stillstands- und Laufzugkraft anhand der Stillstandsmeldung des Antriebes)
	TensionSpeed	REAL	Spanndrehzahl bei leerem Wickel [min ⁻¹]
	ActTorqueFilter	TIME	Filterzeitkonstante für das Istdrehmoment S-0-0084 "Drehmoment-/Kraft-Istwert"
	AccelTorqueFilter	TIME	Filterzeitkonstante für das Beschleunigungsmoment
	DefaultTorqueLimit	REAL	Wert für die bipolare Drehmomentgrenze nach Wegnahme von "Enable" [%]
	WindowTension	REAL	Fenster für den Zugkraftistwert [%], bezogen auf den Zugkraftsollwert (Ausgang "TensionInWindow")
	MinTension	REAL	Untere Schwelle für den Zugkraftistwert (nur für die Anzeige mit binärem Ausgang "MinTensionAck")
	MaxTension	REAL	Obere Schwelle für den Zugkraftistwert (nur für die Anzeige mit binärem Ausgang "MaxTensionAck")
	Diameter	MB_WIND_DIAMETER	Struktur für Durchmesserrechner MB_WIND_DIAMETER , Seite 616
	TensionCtrl	MB_WIND_TENSION	Struktur für Zugkraftregler MB_WIND_TENSION , Seite 619
	Friction	MB_WIND_FRICTION	Struktur für Reibmoment MB_WIND_FRICTION , Seite 618
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen

I/O-Typ	Name	Typ	Kommentar
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose Fehlerbehandlung, Seite 553
	InPause	BOOL	Zugkraftregler und Wickelrechner pausieren
	PresetDiamAck	BOOL	Presetdurchmesser wurde übernommen (ist gesetzt, solange Preset ansteht)
	InPauseDiam	BOOL	Pause Durchmesserrechner
	MinAvDiamAchieved	BOOL	Minimaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	MaxAvDiamAchieved	BOOL	Maximaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	HighLimitDiamActive	BOOL	Aktuell berechneter Durchmesser hat obere Grenze erreicht (Durchmesserwert wird verworfen)
	LowLimitDiamActive	BOOL	Aktuell berechneter Durchmesser hat untere Grenze erreicht (Durchmesserwert wird verworfen)
	DiameterEngaged	BOOL	Durchmesser "eingerastet" TRUE: eingerastet FALSE: ausgerastet
	HighLimitTensActive	BOOL	Ausgang des Zugkraftreglers hat obere Grenze erreicht
	LowLimitTensActive	BOOL	Ausgang des Zugkraftreglers hat untere Grenze erreicht
	TensionInWindow	BOOL	Zugkraftistwert ist innerhalb des Zugkraftfensters
	MinTensionAchieved	BOOL	Zugkraft unterhalb des Einganges "MinTension"
	MaxTensionAchieved	BOOL	Zugkraft oberhalb des Einganges "MaxTension"
	CurrentDiameter	REAL	aktuell berechneter Durchmesser [mm]
	GearRatioFineAdjust	REAL	Resultierender Getriebefineabgleich zur Drehzahl-nachführung des Antriebes [%]
	CurrentInertia	REAL	Aktuelles Trägheitsmoment des Wickelgutes + Mechanik (bezogen auf die Motorseite) [kg m ²]
	SpeedOffset	REAL	Spanndrehzahl [U/min]
	TorqueLimitBipolar	REAL	Bipolare Drehmomentengrenze [%]
	WindTensDiag	MB_WIND_TENS_DIAG	Aktuelle Regelabweichung des Tänzerreglers MB_WIND_TENS_DIAG, Seite 618
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (Wickelachse)

Tab. 8-31: Schnittstellenvariablen FB MB_WinderTensionCtrlType02

Min-/Max- und Standardwerte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Standardwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standardwert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
WindCtrlType	MB_WIND_CTRL_TYPE				Steigende Flanke an Enable
WindDirection	MB_WIND_DIRECTION				Steigende Flanke an "Enable"
PauseDiameter	BOOL			FALSE	Kontinuierlich
PresetDiameter	REAL			FALSE	Steigende Flanke an "Preset"
TensionMode	MB_WIND_TENSION_MODE				Steigende Flanke an "Enable"
SetpointStandstill	REAL	0.0	n. def.	0.0	Kontinuierlich
SetpointOpMode	REAL	n. def.	n. def.	0.0	Kontinuierlich
TensionSpeed	REAL	0.0	n. def.	0.0	Kontinuierlich
ActTorqueFilter	TIME	0s	n. def.	0s	Kontinuierlich
AccelTorqueFilter	TIME	0s	n. def.	0s	Kontinuierlich
DefaultTorqueLimit	REAL	0.0	n. def.	100.0	Kontinuierlich
WindowTension	REAL	0.0	n. def.	0.0	Kontinuierlich
MinTension	REAL	0.0	n. def.	0.0	Kontinuierlich
MaxTension	REAL	0.0	n. def.	0.0	Kontinuierlich
Diameter	MB_WIND_DIAMETER				Je nach Strukturelement
TensionCtrl	MB_WIND_TENSION				Kontinuierlich
Friction	MB_WIND_FRICTION				Kontinuierlich

Tab. 8-32: Min-/Max- und Standardwerte der Eingänge des FBs MB_Winder-TensionCtrlType02

Signal-Zeit-Diagramm

Im folgenden Signal-Zeit-Diagramm sind ausgewählte Signale des Funktionsbausteins dargestellt. Mit einer positiven Flanke am Eingang "Enable", der am Ausgang "InOperation" signalisiert wird, wird der Baustein in Betrieb genommen. Die Betriebsbereitschaft kann durch einen Fehler, der am Ausgang "Error" bestätigt wird, unterbrochen werden. Der interne Reglerprozess kann mit dem Eingang "Pause" angehalten werden. "Pause" wird am Ausgang "In-Pause" signalisiert. Das Belegen des Ausgangs "CurrentDiameter" mit einem Vorgabewert (Eingang "PresetVal" in der Struktur MB_WIND_DIAMETER) wird durch eine positive Flanke am Eingang "PresetDiameter" unabhängig vom Eingang "Pause" vorgenommen. Es ist jedoch empfehlenswert, den Funktionsbaustein in "Pause" zu schalten. Dadurch lässt sich der Durchmesser-Preset kontrollierter und übersichtlicher durchführen. Das Setzen des Durchmessers wird am Ausgang "PresetDiamAck" angezeigt.

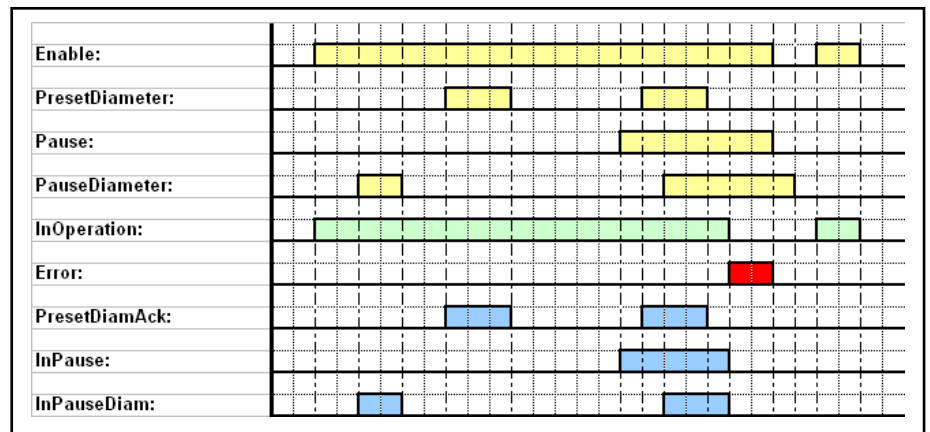


Abb. 8-20: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_WinderTensionCtrlType02

Funktionsbeschreibung

Wickelrechner mit Zugkraftsteuerung

Der Funktionsumfang des Wickelrechners mit Zugkraftsteuerung beinhaltet:

- Spannen des ungespannten Wickelgutes
- Drehzahlabhängige Reibkompensation
- Kompensation des Beschleunigungsmoments
- Getrennte Parametrierung der Laufzugkraft und der Stillstandszugkraft und
- Berechnung durchmesserabhängiger Größen
(mit Funktionsbaustein MB_DiameterCalculatorType03)

Funktionsdiagramm des Wickelrechners mit Zugkraftsteuerung

Anhand des folgenden Funktionsdiagramms wird die Funktionsweise des Wickelrechners mit Zugkraftsteuerung schematisch erläutert.

Anhand des folgenden Funktionsdiagramms wird die Funktionsweise des Wickelrechners mit Zugkraftregelung schematisch erläutert.

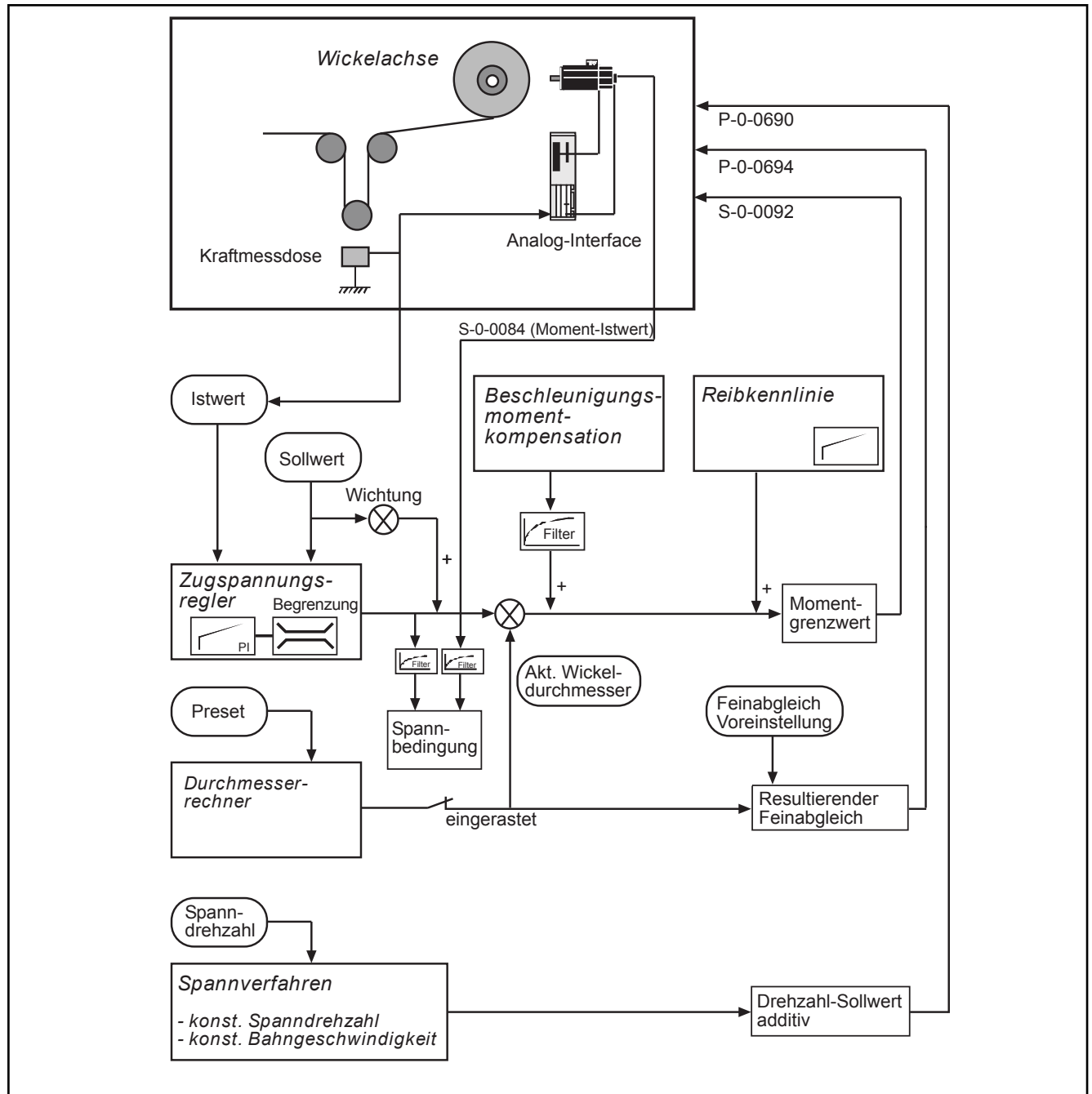


Abb. 8-22: Funktionsdiagramm des Wicklers mit Zugkraftregelung

Drehrichtung des Wickelantriebes, Auf-/Abwickler

Mit Hilfe eines Eingangs des Funktionsbausteins wird der Wicklertyp Auf- bzw. Abwickler eingestellt. Die Drehrichtung des Antriebs ist entweder über die Wichtungparameter oder über die Polarität vom Leitantrieb einzustellen.



Die Drehrichtung der Masterachse des Wickelantriebes (d. h. die Masterachse des Bausteins MC_GearIn bzw. Masterachse des Achs-Interface) muss positiv sein, d. h. positiver Drehzahlsollwert. Eine Invertierung des Wickelantriebes muss über die Polaritätenparameter erfolgen.

Durchmesserberechnung

Das Drehmoment für den Wickelantrieb ergibt sich aus dem Produkt von Zugkraft und Wickelradius. Daher muss der Radius bzw. der Durchmesser des Wickels laufend bestimmt werden. Im Funktionsbaustein ist daher ein interner Durchmesserrechner (Funktionsbaustein [MB_DiameterCalculator-Type03, Seite 509](#)) eingebunden.

Die Durchmesserberechnung wird erst bei gespanntem Wickelgut und keiner Stillstandsmeldung der Referenzachse aktiv und wird zyklisch in Abhängigkeit von der Wickelachsdrehzahl bzw. -lage durchgeführt. Zu Diagnosezwecken kann der aktuelle Durchmesser am Funktionsbaustein ausgelesen werden.

Der berechnete Wickeldurchmesser wirkt direkt auf den resultierenden Feinabgleich des Getriebes. Wird die Bahnspannung zu gering oder negativ (Schlaufenbildung), setzt der Durchmesserrechner aus, da dann keine sinnvolle Durchmesserberechnung mehr möglich ist. Liegt das Istmoment des Antriebs 25% unter dem Sollwert, rechnet der Durchmesserrechner nicht weiter (Einrastbedingung).



Soll der aktuelle Durchmesser, nach dem Ausschalten der Steuerung erhalten bleiben, muss der Ausgang "CurrentDiameter" auf eine remanente Variable kopiert werden. Mit diesem Wert kann nach dem Einschalten wieder ein Preset ausgeführt werden.



Das Verhältnis "Formatlänge" / "CoreDiameter" darf nicht größer als 25 sein.

Ansonsten reicht die Getriebeanpassung über den Getriebefeinabgleich nicht aus!

Spannen des Wickelgutes

Bei Freigabe des Wickelrechners ("Enable" = TRUE) wird das Wickelgut zunächst mit der Spanndrehzahl gespannt. Um eine Schlaufenbildung im Wickelgut zu vermeiden, sollte die Leitachse nicht drehen. Das Wickelgut gilt als gespannt, wenn der Drehmoment-Sollwert des Antriebs eine vorgegebene Grenze überschreitet und die relative Abweichung zwischen Drehmoment-Sollwert und -Istwert kleiner einer bestimmten Schranke ist.

Für das Spannen des Wickelgutes kann zwischen zwei verschiedenen Verfahren gewählt werden:

- Spannen mit konstanter Drehzahl und
- Spannen mit konstanter Bahngeschwindigkeit.

Beim Spannen mit konstanter Drehzahl wird die Spanndrehzahl konstant gehalten. Dadurch steigt mit zunehmendem Wickeldurchmesser die Bahngeschwindigkeit an.

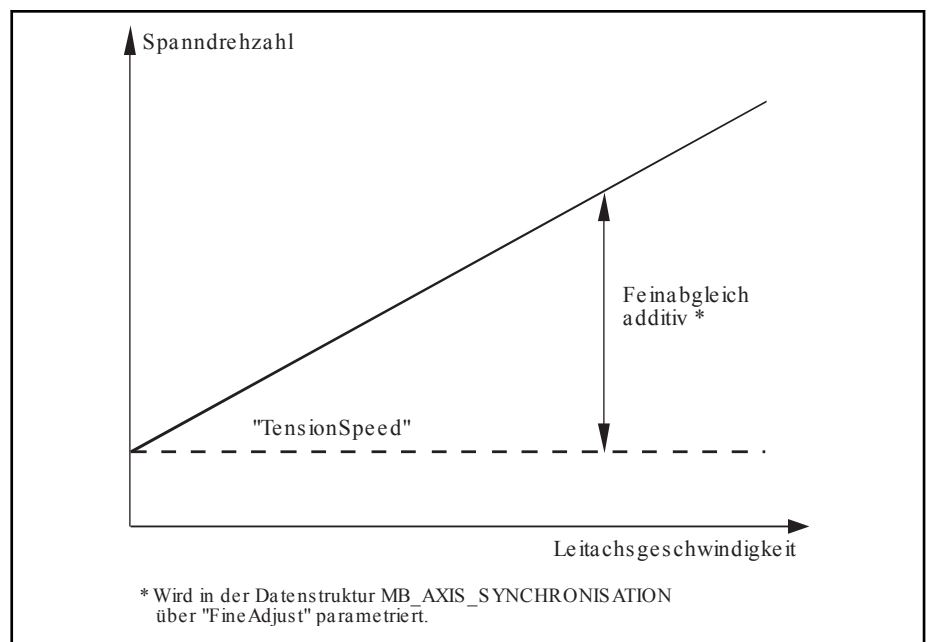


Abb. 8-23: *Spannen mit konstanter Spanndrehzahl*

Beim Spannen mit konstanter Bahngeschwindigkeit sinkt dagegen die Spanndrehzahl reziprok zum Wickeldurchmesser. Die Berechnung der Wickelachsdrehzahl bezieht sich auf die Spanndrehzahl bei leerem Wickel. Diese muss entsprechend dem Funktionsbaustein übergeben werden.

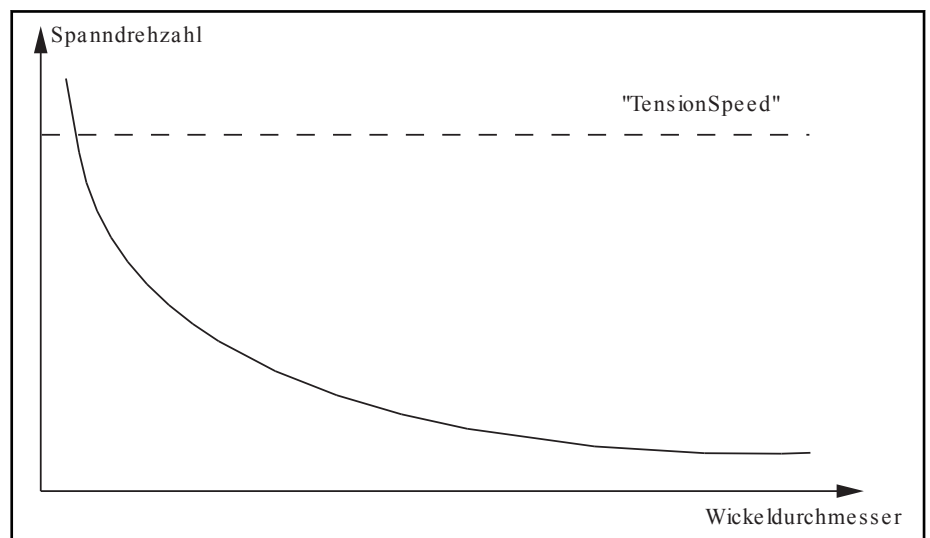


Abb. 8-24: *Spannen mit konstanter Bahngeschwindigkeit*

Reibkompensation

Mit Hilfe verschiedener Eingänge wird eine Reibmomentkompensation umgesetzt. Die Annäherung des Reibungsverhaltens ist wie folgt definiert:

Die Reibung nimmt vom Stillstandsreibmoment zunächst bis zu einer bestimmten Drehzahl ab. Danach steigt das Reibmoment linear mit der Wickelachsdrehzahl bis zur maximalen Drehzahl ("S-0-0091 Geschwindigkeitsgrenzwert bipolar", bzw. "A-0-0032 Geschwindigkeitsgrenzwert, positiv", "A-0-0033 Geschwindigkeitsgrenzwert, negativ") an. Für negative Drehzahlen gilt das Verhalten analog. Die Kurve wird an der y-Achse (Reibmoment) gespiegelt. Mit Hilfe dieses Reibmomentverhaltens, kann dem Effekt der Reibung durch eine Kompensation entgegen gewirkt werden.

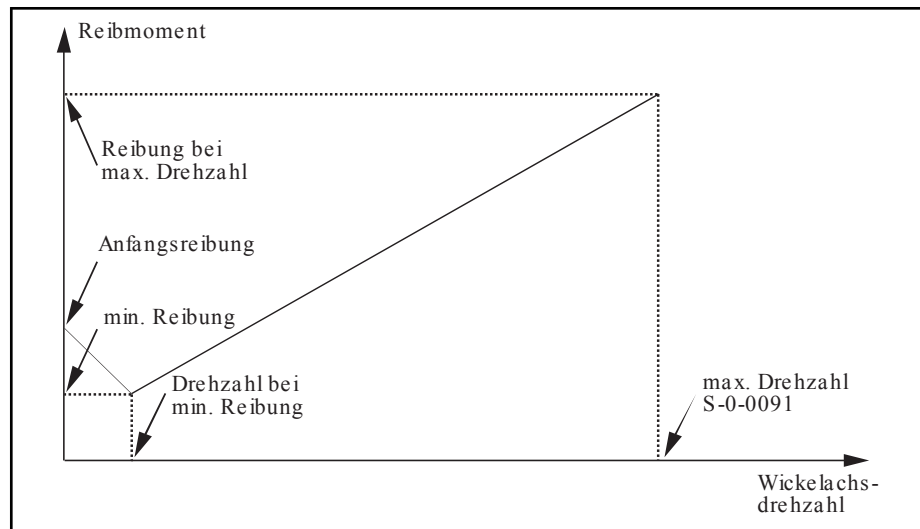


Abb. 8-25: Reibmomentkompensation

Kompensation des Beschleunigungsmoments

Für die Beschleunigung der Wickelachse ist ein zusätzliches Beschleunigungsmoment erforderlich. Dieses ist abhängig vom Wickeldurchmesser. Mit der Angabe von Material- und Stoffbahndaten wird eine durchmesserabhängige Trägheit des Wickels berechnet. Das berechnete Beschleunigungsmoment kann zudem gefiltert werden.

Rollenwechsel

Nach einem Rollenwechsel ändert sich in der Regel der Wickeldurchmesser schlagartig. Der zuletzt berechnete interne Wert für den Wickeldurchmesser entspricht dann nicht mehr dem aktuellen Durchmesser. Damit die Durchmesserberechnung korrekt aufsetzen kann, ist ein Preset des Wickeldurchmessers erforderlich. Der Eingangswert für den Preset sollte annähernd dem tatsächlichen Wert des Wickeldurchmessers entsprechen.



Wird der Preset nicht durchgeführt, kann es zu Fehlfunktionen kommen. Daher sollte der Preset abgeschlossen sein, bevor der Durchmesserrechner wieder aktiv wird.

Der Wert für den Preset sollte wie folgt gewählt werden:

- beim Aufwickler etwas kleiner bzw. gleich dem tatsächlichen Wickeldurchmesser und
- beim Abwickler etwas größer bzw. gleich dem tatsächlichen Wickeldurchmesser

Bei einem fliegenden Rollenwechsel muss sichergestellt werden, dass die Umfangsgeschwindigkeit der neuen Rolle, der aktuellen Rolle entspricht. Dies sollte entsprechend im SPS-Projekt in IndraLogic umgesetzt werden.

Filterung des Momentenistwerts

Bei niedrigen Zugkräften kann ein ungefilterter Momentenwert zu einem Ausrasten des Wickelrechners führen. Daher sollte eine Filterung vorgenommen und der entsprechende Eingang beschaltet werden. Hierzu werden Ist- und Sollmoment des Antriebs miteinander verglichen (z. B. 20ms).

Zusätzliche Vor- bzw. Nacheilung der Wickelachse

Eine zusätzliche Vor- bzw. Nacheilung über den Getriebefeinabgleich (Voreinstellung Feinabgleich "FineAdjust") kann über die Strukturvariable "qrAxisCtrl[AxisNummer].Admin.SyncMode.FineAdjust" der Datenstruktur

Aktivierung der Korrektur der Drehmomentkonstante bei Synchronmotoren

MB_AXIS_SYNCHRONISATION der Bibliothek ML_TechInterface eingestellt werden. Bei einem Aufwickler sollte ein kleiner positiver Wert (z. B. 5%), bei einem Abwickler ein kleiner negativer Wert eingestellt werden. Dies führt dann bei der jeweiligen Wickelrichtung zu einem besseren Anlaufverhalten.

Das von einem Motor entwickelte Drehmoment wird antriebsseitig anhand des gemessenen Motorstroms und einer sogenannten Drehmomentkonstante P-0-0051 "Drehmoment-/Kraft-Konstante" ermittelt. Diese sogenannte "Drehmoment- bzw. Kraftkonstante" ist jedoch physikalisch kein statischer Wert.

Er ist abhängig von:

1. der Höhe des aktuell fließenden Motorstroms.
(Die Konstante reduziert sich bei Strömen größer I_{Nenn}).
2. der Temperatur von Motorwicklung und Rotor/Läufer.
(Die Konstante reduziert sich bei steigender Temperatur).

Für den **Synchronmotor** kann eine korrigierte Drehmomentkonstante verwendet werden. Im Parameter **P-0-0556 "Achsregler-konfiguration" Bit 11 (gesetzt)** kann diese aktiviert werden.

Hierdurch werden folgende Einflüsse auf die Drehmoment-/Kraftkonstante berücksichtigt:

1. Magnetfeldschwächung durch magnetische Sättigung aufgrund erhöhtem Motorstrom gegenüber Nennstrom.
2. Magnetfeldschwächung durch erhöhte Motorwicklungstemperatur.
3. Magnetfeldschwächung durch erhöhte Rotortemperatur abhängig von der mittleren Drehzahl.

Die in diesem Fall aktuell wirkende Drehmomentkonstante kann im Parameter P-0-0450 "Drehmoment-/Kraft-konstante, aktuell" abgelesen werden.

Eine deutliche Abweichung von der Drehmomentkonstante P-0-0051 wird sich erst ab einer Strombelastung von 100 % (und mehr) des Nennstromes S-0-0111 "Stillstandsstrom Motor" ergeben. D.h. die Korrektur wirkt umso besser, je besser der Motor ausgelastet ist. Für einen schlecht ausgelasteten Motor wird sich nur eine minimale bzw. keine Korrektur ergeben.



Für einen Asynchronmotor ist eine Korrektur der Drehmomentkonstante automatisch aktiviert und muss nicht separat im Parameter P-0-0556 eingestellt werden.

Siehe auch Funktionsbeschreibung des entsprechenden Indradrive-Antriebs.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_WinderTensionCtrlType02 aufgelistet und beschrieben.

Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

Zusätzlich können die Fehlercodes der Funktionsbausteine MB_DiameterCalculatorType03 und IL_PIDType01, RIL_LoopControl, sowie MB_GetCyclicParameterHandle, ML_TechMotionBase, auftreten.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1120	16#20	Fehler in der Statemachine
INPUT_RANGE_ERROR	16#1121	16#01	Diameter.MechanicInertia < 0.0
INPUT_RANGE_ERROR	16#1121	16#02	TensionCtrl.PreControlWeighting < 0.0

ErrorID	Additional1	Additional2	Text
INPUT_RANGE_ERROR	16#1121	16#03	MaxTension < 0.0
INPUT_RANGE_ERROR	16#1121	16#04	TensionHighLimit < TensionLowLimit
INPUT_RANGE_ERROR	16#1121	16#05	Friction.StandstillTorque < 0.0
INPUT_RANGE_ERROR	16#1121	16#06	Friction.MaxTorque < 0.0
INPUT_RANGE_ERROR	16#1121	16#07	Friction.MinTorque < 0.0
INPUT_RANGE_ERROR	16#1121	16#08	Friction.MinVelocity < 0.0
INPUT_RANGE_ERROR	16#1121	16#09	WindowTension < 0.0
INPUT_RANGE_ERROR	16#1121	16#0A	MinTension < 0.0
INPUT_RANGE_ERROR	16#1121	16#0B	MaxTension < 0.0
INPUT_RANGE_ERROR	16#1121	16#0C	Wickelachse nicht in "BB"
INPUT_RANGE_ERROR	16#1121	16#0D	Drehmomentwichtung falsch (A-0-0050) Wichtung für Motormoment muss motorseitig be- zogen sein
INPUT_RANGE_ERROR	16#1121	16#0E	Achsnummer der Slaveachse falsch
INPUT_RANGE_ERROR	16#1121	16#0F	Achsnummer der Masterachse falsch
INPUT_RANGE_ERROR	16#1121	16#12	DefaultTorqueLimit < 0.0
INPUT_RANGE_ERROR	16#1121	16#14	TensionSpeed < 0.0
INPUT_RANGE_ERROR	16#1121	16#15	DefaultTorqueLimit maximum
RESOURCE_ERROR	16#0004	16#0000	falsche Antriebsfirmware
RESOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert
RESOURCE_ERROR	16#0012	16#00xx	wrong local cntrl number for local cntrl xx
RESOURCE_ERROR	16#001C	16#00xx	wrong Axistype for Axis xx

Tab. 8-33: Fehlercodes FB MB_WinderTensionCtrlType02

Notwendige Hardware

- CMLx, XMx, VPx
- IndraDrive

Notwendige Firmware

- MLC Firmware FWA-CMLXX*-MLC-14VRS
XX - Gerätevariante, z. B. CML65
- IndraDrive Antriebsfirmware

Notwendige Software

- IndraWorks 14VRS für MLC bzw. MLD

Besonderheiten für MLC

Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter ("S-0-0121 Lastgetriebe-Eingangsumdrehungen") und ("S-0-0122 Lastgetriebe-Ausgangsumdrehungen") eingetragen werden.

Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z. B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!

Interpolation im Antrieb

Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss der Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" und Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus.

Interpolation in der Steuerung

Die Zielgrößen sind die Steuerungsparameter "A-0-2605 ELS, Feinabgleich der Getriebeübersetzung, Prozessregler" und "A-0-2615 ELS, Geschwindigkeits-Sollwert additiv, Prozessregler" und "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar".

Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss der Parameter "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar" eingetragen sein.

Besonderheiten für MLD

Bei Verwendung der Steuerung MLD-S mit übergeordneter Steuerung MLC muss für die Wickelachse Interpolation im Antrieb aktiviert sein (reale Achsen entspr. Achse ► Eigenschaften). Bei Verwendung der Steuerung MLD-M gibt es diese Einstellung nicht.

Die Wichtungsart für die Master-Achse und Slave-Achse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

Im zyklischen Sercos-Kanal des entsprechenden Antriebs müssen die Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" und Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" und "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus.

Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus (der Dialog ist im Projektbaum unterhalb des MLD-Knotens **Axis-Data**).

Die "AxisData-Struktur" des Antriebes muss parametrierbar sein ("P-0-1367 SPS Konfiguration Bit 6 ist gesetzt", bzw. Dialog SPS-Konfiguration).

Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter "S-0-0121 Lastgetriebe-Eingangsumdrehungen" und "S-0-0122 Lastgetriebe-Ausgangsumdrehungen" eingetragen werden. Den Parameter "S-0-0091 Geschwindigkeits-Grenzwert bipolar" entsprechend der maximalen Drehzahl des Wickelantriebes setzen.

Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z. B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!

Stillstandsfenster Referenzachse:

Für die Referenzachse kann der Funktionsbaustein auf keinen Wert für das Stillstandsfenster zugreifen. Intern wird dann ein Wert von 1 U/min angenommen. Ist dieser Wert für die Applikation nicht geeignet, kann dieser mittels folgender SPS-Codezeilen beeinflusst werden:

Deklaration

```
prMaStandstillWdw:    POINTER TO REAL; // for StandstillWindow master MLD
rTestMaStandstillWdw: REAL := 0.5;    // for StandstillWindow master MLD
```

Implementation

```
prMaStandstillWdw := ADR(BausteinInstanznameWickler.fbDiameterCalculator.rMaStandstillWindow);
prMaStandstillWdw^ := rTestMaStandstillWdw;
```

Der Wert für diese Variable bleibt nicht gespeichert, der Wert muss beim Einschalten der SPS also neu geschrieben werden.

Achsreferenz für die Masterachse:

Für die Technologiebausteine wird als Achsreferenz für die Masterachse empfohlen:

- Für eine virtuelle Leitachse die Achsreferenz "**VmAxis1**" verwenden.
Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 20 (VMA_1) eingestellt werden.
- Für eine reale Leitachse die Achsreferenz "**RmAxis1**" verwenden.
Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 23 (RMA_1) eingestellt werden.
Zusätzlich muss in diesem Fall noch die logische Adresse des Messgebers im Parameter "P-0-1820.0.2 CCD: Quelle des Messgeber-Lageistwertes" eingetragen werden.

Detaillierte Informationen können der Funktionsbeschreibung MLD entnommen werden.

Inbetriebnahme des Wickelrechners mit Zugkraftsteuerung bzw. Zugkraftregelung

Zur Inbetriebnahme des Wickelrechners mit Zugkraftsteuerung bzw. Zugkraftregelung siehe [Kap. 8.6.3 "Inbetriebnahme des Wickelrechners mit Zugkraftsteuerung bzw. Zugkraftregelung"](#) auf Seite 556.

8.6.3 Inbetriebnahme des Wickelrechners mit Zugkraftsteuerung bzw. Zugkraftregelung

Antriebsregelkreis

Es wird vorausgesetzt dass der Drehzahlregelkreis des Wickelantriebes eingestellt wurde. D. h. die Parameter "S-0-0100 Geschwindigkeitsregler-Proportionalverstärkung" und "S-0-0101 Geschwindigkeitsregler-Nachstellzeit" müssen der Applikation entsprechend eingestellt sein.

Im Folgenden wird die Inbetriebnahme des Wickelrechners in sieben Schritten beschrieben:

1. Festlegung des Wicklertyps
2. Festlegung der Drehrichtung der Wickelachse
3. Ermittlung des Reibmoments im Stillstand
4. Ermittlung des Reibmoments bei maximaler Drehzahl
5. Ermittlung des minimalen Reibmoments
6. abschließende Einstellungen und
7. erster Anlauf der Wickelachse

Kommandierung der Wicklerachse

Für die Wicklerachse muss die richtige Betriebsart des Antriebes, sowie die Reglerfreigabe des Antriebes gesetzt werden.

Das Setzen von Betriebsart und Reglerfreigabe kann auf zwei unterschiedliche Arten erfolgen:

- Setzen der Betriebsart "ModeSyncVel" mit Hilfe des Achsinterfaces (dies wird empfohlen)
- Benutzung der PLC-Open Funktionsbausteine "MC_Power", "MC_GearIn", "MC_Stop" zur Kommandierung der Achse

Festlegung des Wicklertyps

Zunächst muss festgelegt werden, welcher Wicklertyp vorliegt bzw. eingesetzt werden soll. Dazu wird am Eingang "WindDirection" die Wicklerrichtung festgelegt (TRUE: Abwickeln, FALSE: Aufwickeln). Zudem wird der Wickelrechner mit dem Eingang "WindCtrlType" ausgewählt (TRUE: Wickelrechner mit Zugkraftregelung, FALSE: Wickelrechner mit Zugkraftsteuerung).

Festlegung der Drehrichtung der Wickelachse

Die Drehrichtung des Wicklers kann mit Hilfe des Antriebsparameters "P-0-0108 Polarität Leittrieb" bzw. des Achsparameters "A-0-2798 Polarität der Leitachspannung" eingestellt werden. Dreht sich die Wickelachse nicht in die gewünschte Richtung, ist der Parameter P-0-0108 bzw. der Achsparameter A-0-2798 entsprechend umzustellen. Die Polarität der Leitachspannung wird lediglich beim Einschalten des Funktionsbausteins überprüft ("Enable" = TRUE). Eine Änderung der genannten Parameter während des Betriebs des Funktionsbausteins zeigt somit keine Wirkung.

Alternativ kann die Drehrichtung über die Wichtungsart des Wickelantriebs geändert werden. Dies kann durch Drehen des Vorzeichens der Lage-Beschleunigungs- und der Drehmoment-Kraftdaten erfolgen (Parameter S-0-0055, A-0-0029, S-0-0043, S-0-0085).

Die folgende Abbildung zeigt die prinzipiellen Zusammenhänge bei Auf- bzw. Abwickeln.

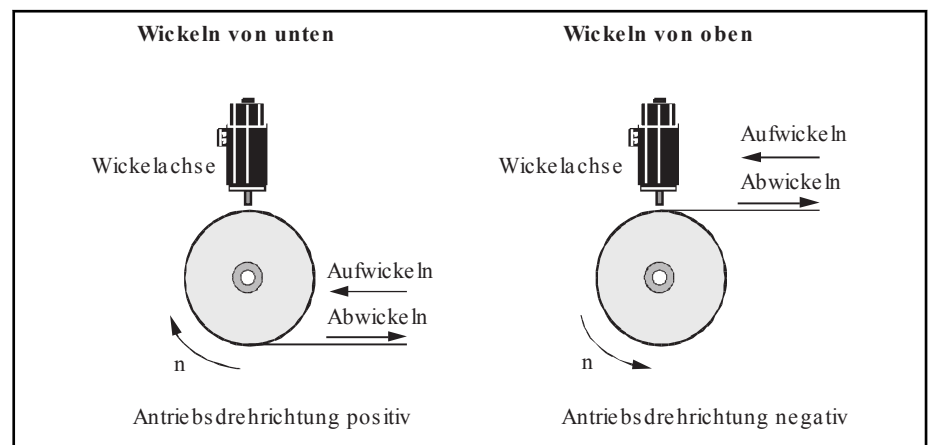


Abb. 8-26: Prinzipien von Auf- und Abwickeln

Ermittlung des Reibmoments im Stillstand

1. Das "Reibmoment im Stillstand" (Variable "StandstillTorque" in der Datenstruktur MB_WIND_TENSION) wird bei stehender Leitachse und mit leerem Wickel (nur Hülse) ermittelt. Der Wickelbaustein ist ausgeschaltet. Für die Wickelachse (Folgeachse) ist die Betriebsart "Geschwindigkeitsregelung" zu wählen (im Achs-Interface der Bibliothek ML_TechInterface "MODE_VEL" anwählen), alternativ kann die Achse mit dem PLCopen-Funktionsbaustein MC_MoveVelocity kommandiert werden. Als Sollwert kann z. B. 10 U/min eingestellt werden.
 2. Nun wird der Wert Parameter "P-0-0109 Spitzendrehmoment-/Kraft-Begrenzung" auf den Wert Null gesetzt. Die Achse muss nun stehen bleiben. Der Wert für diesen Parameter wird nun schrittweise erhöht, bis sich der Antrieb zu drehen beginnt. Der letzte Wert entspricht dem Reibmoment im Stillstand.
 3. Die Achsfreigabe "MODE_VEL" wird wieder zurückgenommen (alternativ mit dem PLCopen-Funktionsbaustein MC_Stop).
- Der Parameter "P-0-0109 Spitzendrehmoment-/Kraft-Begrenzung" wird wieder auf seinen alten Wert (z. B. 400%) gesetzt.
- Die Einstellung des Reibmoments im Stillstand ist somit abgeschlossen.

Ermittlung des Reibmoments bei maximaler Drehzahl

1. Das "Reibmoment bei maximaler Drehzahl" (Variable "MaxTorque" in der Datenstruktur MB_WIND_TENSION) wird bei stehender Leitachse

und mit leerem Wickel (nur Hülse) ermittelt. Der Funktionsbausteins ist ausgeschaltet.

2. Für die Wickelachse (Folgeachse) ist die Betriebsart "Geschwindigkeitsregelung" zu wählen (im Achs-Interface der Bibliothek ML_TechInterface "MODE_VEL" anwählen, alternativ PLCopen-FB).

Der Sollwert für die angewählte Betriebsart ist schrittweise bis zum Wert des Parameters "S-0-0091 Geschwindigkeits-Grenzwert bipolar" zu erhöhen. Der Wert für den Parameter "S-0-0091 Geschwindigkeits-Grenzwert bipolar" muss entsprechend Applikation und Motordaten eingetragen sein.

3. Im nächsten Schritt wird der Wert des Parameters "P-0-0109 Spitzendrehmoment-/Kraft-Begrenzung" schrittweise verringert (z. B. von 100% Richtung 0%), bis die Drehzahl des Antriebs ebenfalls zu sinken beginnt. Die Drehzahl des Antriebs kann mittels des Antriebsparameters "S-0-0040 Geschwindigkeitswert Geber 1" bzw. des Achsparameters "A-0-0102 Geschwindigkeitswert" verfolgt werden. Der letzte Wert von "P-0-0109 Spitzendrehmoment-/Kraft-Begrenzung" entspricht dem maximalen Reibmoment bei maximaler Drehzahl.

4. Die Achsfreigabe "MODE_VEL" wird wieder zurückgenommen (alternativ mit dem PLCopen-Funktionsbaustein MC_Stop).

Der Parameter "P-0-0109 Spitzendrehmoment-/Kraft-Begrenzung" wird wieder auf seinen alten Wert (z. B. 400%) gesetzt.

Die Einstellung des Reibmoments maximaler Drehzahl "MaxTorque" ist somit abgeschlossen.

Ermittlung des minimalen Reibmoments

Das minimale Reibmoment wird mit Hilfe eines Bremsversuchs mit leerem Wickel ermittelt. Der Wickelantrieb wird von einer konstanten Geschwindigkeit (z. B. 50 min⁻¹) bis zum Stillstand abgebremst. Gleichzeitig werden die Antriebsparameter "S-0-0080 Drehmoment-/Kraft-Sollwert" (Achsparameter "A-0-0038 Drehmoment-/Kraftgrenzwert, bipolar") und "S-0-0040 Geschwindigkeitswert Geber 1" (Achsparameter "A-0-0102 Geschwindigkeitswert") mit einem Oszilloskop bzw. mit der Tracefunktion in IndraWorks aufgezeichnet.

Aus dem Drehmomentverlauf wird der Minimalwert abgelesen und kann der Struktur MB_WIND_FRICTION als "MinTorque" übergeben werden. Die korrespondierende Drehzahl wird als "MinTorqueVelocity" in MB_WIND_FRICTION eingetragen.

Abschließende Einstellungen

Schließlich werden, wie in der folgenden Tabelle gezeigt, die verbleibenden Eingangsgrößen festgelegt.

Eingangsgröße	Bemerkung/Einstellung
SetpointOpMode	Einstellung der gewünschten Laufzugkraft
SetpointStandstill	Einstellung des Stillstandzugs
TensionMode	Auswahl der Spanndrehzahlcharakteristik: • TRUE: Spannen mit konstanter Bahngeschwindigkeit • FALSE: Spannen mit konstanter Spanndrehzahl

Eingangsgröße	Bemerkung/Einstellung
Offset im Feinabgleich	In der Datenstruktur MB_AXIS_SYNCHRONISATION aus der Bibliothek ML_TechInterface kann eine Voreinstellung des Feinabgleichs durch die Variable "FineAdjust" mit einem Wert abhängig von der Wickelrichtung belegt. Diese Variable wird zyklisch verarbeitet, wenn "EnableCyclicScanning" = TRUE gesetzt ist. Die Wickelrichtung wird, wie zuvor beschrieben, mit "WindDirection" festgelegt. Bei einem Abwickler wird ein positiver Wert für "FineAdjust" und für einen Aufwickler ein negativer Wert eingesetzt. Diese Werte sollten $\leq 5\%$ betragen
ActTorqueFilter	Soll das Ist Drehmoment (als Messwert) gefiltert werden, muss an diesem Eingang des Funktionsbaustein ein Wert ungleich Null als Zeitkonstante angelegt werden. Dieser Wert wirkt auf den Standardparameter "S-0-0084 Drehmoment-/Kraft-Istwert"
AccelTorqueFilter	Soll das Beschleunigungsmoment gefiltert werden, muss an diesem Eingang des Funktionsbausteins ein Wert ungleich Null als Zeitkonstante angelegt werden

Tab. 8-34: Abschließende Festlegung der verbleibenden Eingangsgrößen

Erster Anlauf der Wickelachse

Das Wickelgut wird ungespannt (leicht durchhängend) aufgelegt. Die Wickelachse muss entweder über das Achsinterface oder über die PLCopen Funktionsbausteine kommandiert werden. Danach wird der Wickelrechner mit Setzen des Eingangs "Enable" aktiviert. Jetzt sollte sich das Wickelgut mit der vorgegebenen Spanndrehzahl (Eingang "TensionSpeed") spannen. Der Ausgang "SpeedOffset" stellt die Spanndrehzahl zur Verfügung.

Das Prozessgrößenfenster "WindowTension" kann auf einen Wert von 10 - 20% eingestellt werden.

Die Leitachse wird mit einer vergleichsweise niedrigen Drehzahl (z. B. 10 min^{-1}) freigegeben. Nach einigen Umdrehungen muss der Ausgang "DiameterEngaged" gesetzt sein. Hiermit wird angezeigt, dass die Durchmesserberechnung aktiv ist ("eingerastet"). Diese ist abhängig von der Bahnspannung bzw. dem Istmoment an der Wickelachse und der Drehmomentgrenze (Antriebsparameter "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar" bzw. Achsparameter "A-0-0038 Drehmoment-/Kraftgrenzwert, bipolar"). Die Drehmoment/Kraft-Grenze wird am Ausgang "TorqueLimitBipolar" ausgegeben.

Bei eingerastetem Wickelrechner können Stillstands- und Laufzugkraft überprüft werden. Dazu wird der Wert "CurrentTension" aus der Datenstruktur ML_WIND_TENS_DIAG beobachtet. In dieser Datenstruktur können zudem der aktuelle Drehmomentgrenzwert "TorqueLimit", das aktuelle Beschleunigungsmoment "AccelerationTorque" und das aktuelle angenommene Reibmoment "FrictionTorque" betrachtet werden.

Schließlich stellt der Funktionsbaustein an seinen Ausgängen verschiedene Prozessgrößen zur Verfügung. Der berechnete aktuelle Durchmesser des Wickels (Ausgang "CurrentDiameter") sollte mit geringfügiger, vernachlässigbarer Abweichung dem tatsächlichen Durchmesser des Wickels entsprechen. Ferner werden der Getriebefinabgleich zur Drehzahlachsführung der Antriebsachse des Wickels (Ausgang "GearRatioFineAdjust"), das aktuelle Trägheitsmoment des Wickels (Ausgang "CurrentInertia") und die Spanndrehzahl (Ausgang "SpeedOffset") ausgegeben.

Momentengrenzwertüberwachungen

Beim ersten Hochlaufen der Steuerung kann es zu einer Warnung des Antriebes "E2056 Momenten-Grenzwert = 0" kommen, wenn der Wicklerbaustein noch nicht eingeschaltet ist.

Um dies zu verhindern, kann in der Anlaufphase des SPS-Programmes (z. B. Warmstart) der entsprechende zyklische Datencontainer, der den Wert für den Parameter S-0-0092 "Drehmoment/Kraftwert bipolar" enthält, mit einem Wert ungleich Null beschrieben werden, z. B. 100%.

Kommt es bei eingeschaltetem Baustein zur Warnung "E8260 Moment-/Kraft-Sollwertbegrenzung aktiv", kann diese Warnung durch Setzen des Bit 4 des Parameters P-0-0556 "Achsregler-Konfiguration" beseitigt werden.

In seltenen Fällen (z. B. mehrfach F8078 "Fehler im Drehzahlregelkreis") kann es notwendig sein, die Geschwindigkeitsreglerüberwachung zu deaktivieren (Setzen des Bit 1 des Parameters P-0-0556 "Achsregler-Konfiguration").

8.7 Wickelrechner mit Zugkraftregelung und Drehzahlkorrektur

8.7.1 Einführung und Übersicht

Der Wickelrechner mit Zugkraftregelung und Drehzahlkorrektur wird an Zentrumswicklern eingesetzt. Der Antrieb der Wickelachse arbeitet generell NICHT an seiner Drehmomentengrenze. Aus Sicherheitsgründen wird eine Drehmomentenbegrenzung berechnet die zusätzlich noch einstellbar ist (diese sollte oberhalb des benötigten Motormomentes liegen und den Antrieb im normalen Betrieb nicht begrenzen).

Dieser Wickler wird hauptsächlich für elastisches, dehnbares Material eingesetzt.

Der Zugkraftistwert wird direkt mittels einer Kraftmessdose gemessen und über einen Analogkanal oder über einen Feldbus dem Zugkraftregler als Istwert zugeführt. Bei gespannter Bahn kann der Durchmesserrechner den aktuellen Wickeldurchmesser berechnen.

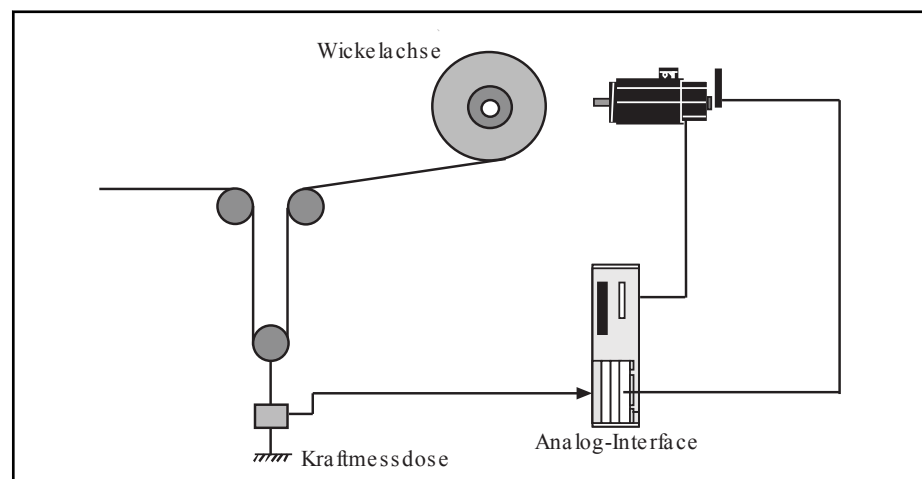


Abb. 8-27: Wickelrechner mit Zugkraftregelung

8.7.2 MB_WinderTensionCtrlSpeedType01

Kurzbeschreibung

Der Funktionsbaustein berechnet zyklisch den aktuellen Wickeldurchmesser, das aktuelle Trägheitsmoment des Wickels, die entsprechende Nachführung der Geschwindigkeit über den Getriebefeinabgleich, einen additiven Dreh-

zahloffset zum Spannen der Bahn und eine parametrierbare Drehmomentenbegrenzung.

Zielgrößen des Funktionsbausteines sind die Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler", "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler", "S-0-0081 Drehmoment-/Kraft-Sollwert additiv und "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar".

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechWinder
MLD ab MPx20 mit MA Funktionspaket	MX_TechWinder

Tab. 8-35: Referenztable MB_WinderTensionCtrlSpeedType01

Schnittstellenbeschreibung

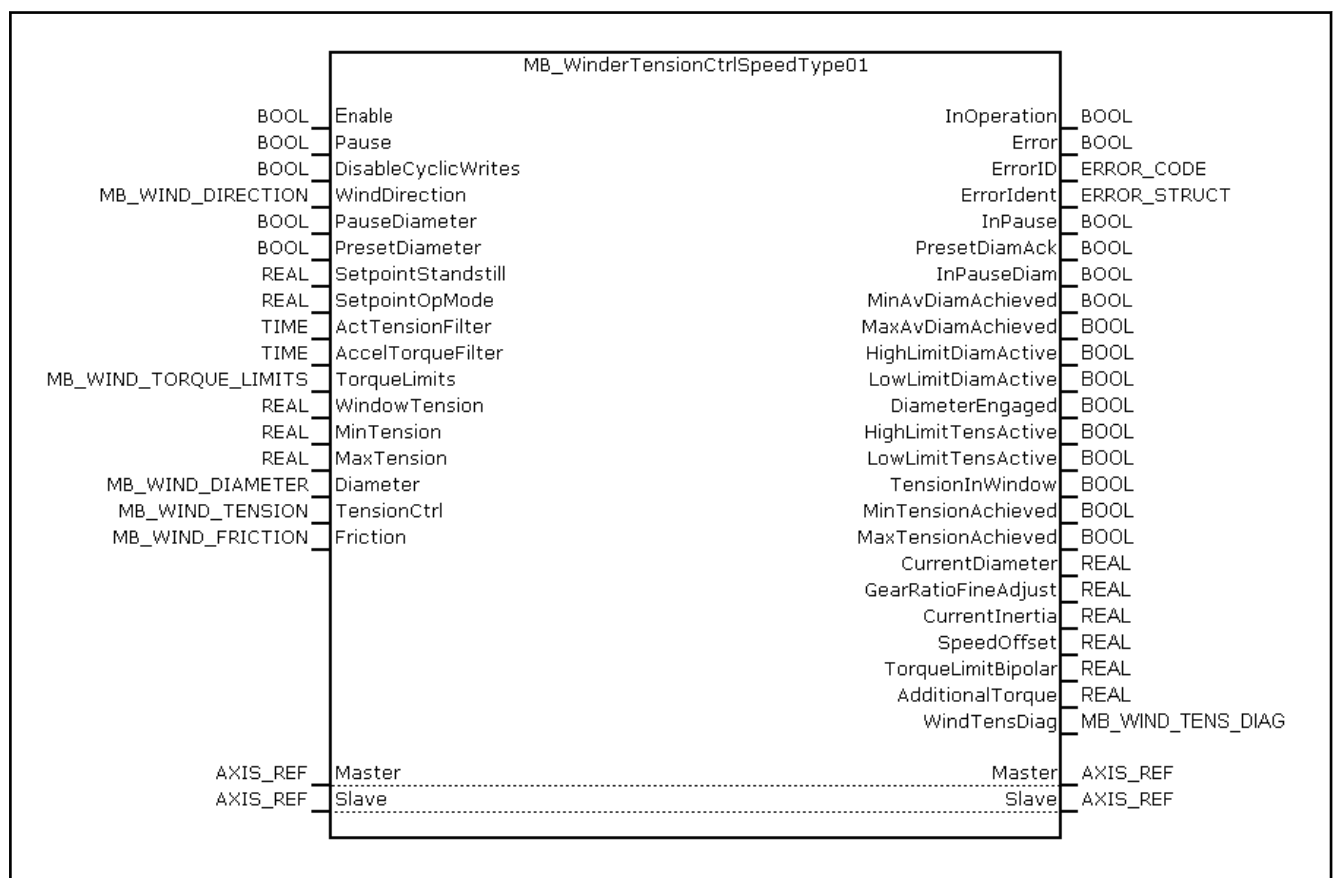


Abb. 8-28: Funktionsbaustein MB_WinderTensionCtrlSpeedType01

I/O-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegel-gesteuert)
	Pause	BOOL	Pausieren des Zugkraftreglers und des Wickelrech-ners (Werte werden eingefroren)

I/O-Typ	Name	Typ	Kommentar
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht direkt in die optional zyklischen Datencontainer geschrieben, sondern nur am Ausgang ausgegeben
	WindDirection	MB_WIND_DIRECTION	<i>Wickelrichtung</i> - Aufwickler REWIND (16#0000) - Abwickler UNWIND (16#0001) MB_WIND_DIRECTION , Seite 618
	PauseDiameter	BOOL	Anhalten der Durchmesserberechnung.
	PresetDiameter	BOOL	CurrentDiameter wird auf PresetVal gesetzt, GearRatioFineAdjust entsprechend angepasst (flankengesteuert)
	SetpointStandstill	REAL	Vorgabe der Stillstandszugkraft [N] (Umschaltung zwischen Stillstands- u. Laufzugkraft anhand der Stillstandsmeldung des Antriebes)
	SetpointOpMode	REAL	Vorgabe der Laufzugkraft [N] (Umschaltung zwischen Stillstands- und Laufzugkraft anhand der Stillstandsmeldung des Antriebes)
	ActTensionFilter	TIME	Filterzeitkonstante für die Istzugkraft
	AccelTorqueFilter	TIME	Filterzeitkonstante für das Beschleunigungsmoment
	TorqueLimits	MB_WIND_TORQUE_LIMITS	Struktur für die bipolare Drehmomentbegrenzung (S-0-0092) MB_WIND_TORQUE_LIMITS , Seite 619
	WindowTension	REAL	Fenster für den Zugkraftistwert [%] bezogen auf den Zugkraftsollwert("TensionInWindow")
	MinTension	REAL	Untere Schwelle für den Zugkraftistwert (nur für die Anzeige mit binärem Ausgang "MinTensionAck")
	MaxTension	REAL	Obere Schwelle für den Zugkraftistwert (nur für die Anzeige mit binärem Ausgang „MaxTensionAck“)
	Diameter	MB_WIND_DIAMETER	Struktur für Durchmesserrechner MB_WIND_DIAMETER , Seite 616
	TensionCtrl	MB_WIND_TENSION	Struktur für Zugkraftregler MB_WIND_TENSION , Seite 619
	Friction	MB_WIND_FRICTION	Struktur für Reibmoment MB_WIND_FRICTION , Seite 618
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detailldiagnose Fehlerbehandlung , Seite 570
	InPause	BOOL	Zugkraftregler und Wickelrechner pausieren

I/O-Typ	Name	Typ	Kommentar
	PresetDiamAck	BOOL	Presetdurchmesser wurde übernommen (ist gesetzt, solange Preset ansteht)
	InPauseDiam	BOOL	Pause Durchmesserrechner
	MinAvDiamAck	BOOL	Minimaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	MaxAvDiamAck	BOOL	Maximaler Durchmesser erreicht (bezieht sich auf gemittelten Durchmesser)
	HighLimitDiamAck	BOOL	Aktuell berechneter Durchmesser hat obere Grenze erreicht (Durchmesserwert wird verworfen)
	LowLimitDiamAck	BOOL	Aktuell berechneter Durchmesser hat untere Grenze erreicht (Durchmesserwert wird verworfen)
	DiameterEngaged	BOOL	Durchmesser "eingrastet" TRUE: eingerastet FALSE: ausgerastet
	HighLimitTensActive	BOOL	Ausgang des Zugkraftreglers hat obere Grenze erreicht
	LowLimitTensActive	BOOL	Ausgang des Zugkraftreglers hat untere Grenze erreicht
	TensionInWindow	BOOL	Zugkraftistwert ist innerhalb des Zugkraftfensters
	MinTensionAchieved	BOOL	Zugkraft unterhalb des Einganges "MinTension"
	MaxTensionAchieved	BOOL	Zugkraft oberhalb des Einganges "MaxTension"
	CurrentDiameter	REAL	Aktuell berechneter Durchmesser [mm]
	GearRatioFineAdjust	REAL	Resultierender Getriebefeinabgleich zur Drehzahl-nachführung des Antriebes [%]
	CurrentInertia	REAL	Aktuelles gesamtes Trägheitsmoment (bezogen auf die Motorseite) [kg/m ²]
	SpeedOffset	REAL	Spanndrehzahl [U/min]
	TorqueLimitBipolar	REAL	Bipolare Drehmomentengrenze [%]
	AdditionalTorque	REAL	Additives Drehmoment [%]
	WindTensDiag	MB_WIND_TENS_DIAG	Aktuelle Regelabweichung des Tänzerreglers MB_WIND_TENS_DIAG , Seite 618
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (Wickelachse)

Tab. 8-36: Schnittstellenvariablen FB MB_WinderTensionCtrlSpeedType01

Min-/Max- und Standardwerte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Standardwerte der Bausteineingänge aufgelistet:

Name	Typ	Min-Wert	Max-Wert	Standardwert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
WindDirection	MB_WIND_DIRECTION			FALSE	Steigende Flanke an "Enable"
PauseDiameter	BOOL			FALSE	Kontinuierlich
PresetDiameter	BOOL			FALSE	Kontinuierlich
SetpointStandstill	REAL	0.0	n.def.	0.0	Kontinuierlich
SetpointOpMode	REAL	0.0	n.def.	0.0	Kontinuierlich
ActTensionFilter	TIME	0s	n.def.	0s	Kontinuierlich
AccelTorqueFilter	TIME	0s	n.def.	0s	Kontinuierlich
TorqueLimits	MB_WIND_TORQUE_LIMITS				Kontinuierlich
WindowTension	REAL	0.0	n.def.	0.0	Kontinuierlich
MinTension	REAL	0.0	n.def.	0.0	Kontinuierlich
MaxTension	REAL	0.0	n.def.	0.0	Kontinuierlich
Diameter	MB_WIND_DIAMETER				Je nach Strukturelement
TensionCtrl	MB_WIND_TENSION				Kontinuierlich
Friction	MB_WIND_FRIC-TION				Kontinuierlich

Tab. 8-37: Min-/Max- und Standardwerte der Eingänge MB_WinderTensionCtrlSpeedType01

Signal-Zeitdiagramm

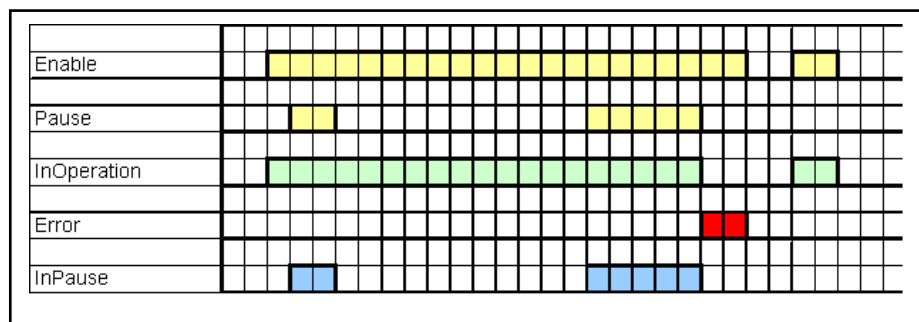


Abb. 8-29: Signal-Zeitdiagramm für MB_WinderTensionCtrlSpeedType01

Funktionsbeschreibung

Der Funktionsbaustein stellt einen Wickelrechner mit Zugkraftregelung dar, der normalerweise nicht an der Drehmomentengrenze arbeitet.

Der Funktionsbaustein berechnet zyklisch den aktuellen Wickeldurchmesser, das aktuelle Trägheitsmoment des Wickels, die entsprechende Nachführung der Geschwindigkeit über den Getriebefeinabgleich und einen additiven Drehzahloffset zum Spannen der Stoffbahn, sowie den benötigten Drehmomentsollwert zur Vorsteuerung.

Intern werden die Bausteine MB_DiameterCalculatorType03 und MB_DiameterMeasurementType01 verwendet.

Zielgrößen des Funktionsbausteins sind die Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler", "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler", "S-0-0081 Drehmoment-/Kraft-Sollwert additiv" und "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar".

Die Geschwindigkeitsanpassung, aufgrund des sich ändernden Wickeldurchmessers, erfolgt über den Getriebefeinabgleich. Der Zugkraftregler wirkt auf den Geschwindigkeitssollwert additiv. Das berechnete Drehmoment (Moment für die Zugkraft, Reibmoment, Beschleunigungsmoment) wird über den additiven Drehmomentsollwert vorgesteuert.



Eine Anpassung der Kp-Verstärkung des Zugkraftreglers kann durch externe Beschaltung des Eingangs "TensionCtrl.PControl" realisiert werden.

z.B. durchmesserabhängig: Mit zunehmendem Durchmesser eine Verringerung der Kp-Verstärkung (Multiplikation mit dem Verhältnis "MinRefDiameter"/"CurrentDiameter")

z.B. materialabhängig: Mit steiferem Wickelmaterial geringeres Kp

Aus Sicherheitsgründen wird der Drehmomentengrenzwert auf den aktuell berechneten Drehmomentensollwert multipliziert mit einem Wichtungsfaktor, sowie einem zusätzlichen additiven Anteil gesetzt. Im normalen Wickelbetrieb soll die Drehmomentenbegrenzung nicht wirken und der Wickelantrieb drehzahl geregelt betrieben werden.

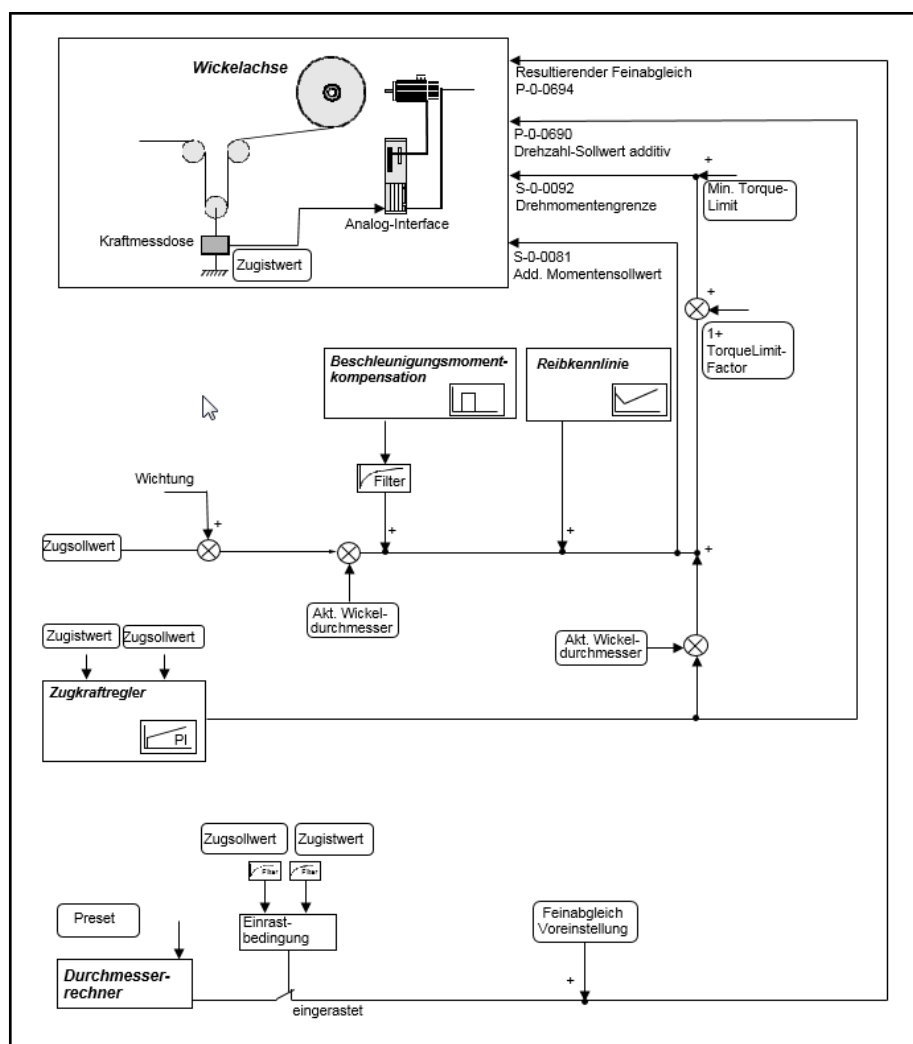


Abb. 8-30: Schematische Darstellung der Regelung bei einer Wickelachse

Drehrichtung des Wickelantriebes, Auf-/Abwickler

Mit Hilfe der Enumeration "MB_WindDirection" des Funktionsbausteins wird der Wicklertyp Auf- bzw. Abwickler eingestellt. Die Drehrichtung des Antriebs ist entweder über die Wichtungsparameter oder über die Polarität vom Leit-antrieb einzustellen.



Die Drehrichtung der Masterachse des Wickelantriebes (d. h. die Masterachse des Bausteins MC_GearIn bzw. Masterachse des AxisInterfaces) muss positiv sein, d. h. positiver Drehzahlsollwert. Eine Invertierung des Wickelantriebes muss über die Polaritätenparameter erfolgen.

Durchmesserberechnung

Das Drehmoment für den Wickelantrieb ergibt sich aus dem Produkt von Zugkraft und Wickelradius. Daher muss der Radius bzw. der Durchmesser des Wickels laufend bestimmt werden. Im Funktionsbaustein ist daher ein interner Durchmesserrechner ([MB_DiameterCalculatorType03](#), Seite 509) eingebunden.

Die Durchmesserberechnung wird erst bei gespanntem Wickelgut und keiner Stillstandsmeldung der Referenzachse aktiv und wird zyklisch in Abhängig-

keit von der Wickelachsdrehzahl bzw. -lage durchgeführt. Zu Diagnosezwecken kann der aktuelle Durchmesser am Funktionsbaustein ausgelesen werden.

Der berechnete Wickeldurchmesser wirkt direkt auf den resultierenden Feinabgleich des Getriebes. Wird die Bahnspannung zu gering oder negativ (Schlaufenbildung), setzt der Durchmesserrechner aus, da dann keine sinnvolle Durchmesserberechnung mehr möglich ist.

Liegt die Istzugkraft des Antriebs 25% unter dem Sollwert, rechnet der Durchmesserrechner nicht weiter (**Einrastbedingung**).



Soll der aktuelle Durchmesser nach dem Ausschalten der Steuerung erhalten bleiben, muss der Ausgang "CurrentDiameter" auf eine remanente Variable kopiert werden. Mit diesem Wert kann nach dem Einschalten wieder ein Preset ausgeführt werden.



Das Verhältnis "Formatlänge"/"CoreDiameter" darf nicht größer als 25 sein.

Ansonsten reicht die Getriebeanpassung über den Getriebefeinabgleich nicht aus!

Beeinflussung der Einrastbedingung

In Ausnahmefällen (z.B. starke Schwankungen der Istzugkraft) kann die Einrastbedingung (Istzugkraft liegt 25% unter der Sollzugkraft) beeinflusst werden. Hierzu wird mit einem Pointerzugriff der Wert für die entsprechende interne Variable verändert.

Deklaration

```
prTensionPercent:    POINTER TO REAL; // for tension condition
rTestTensionPercent: REAL := -0.5;    // example for tension condition 50%
```

Implementation

```
prTensionPercent := ADR(BausteinInstanznameWickler.fbDiameterCalculator.rCheckTensionPercent);
prTensionPercent^ := rTestTensionPercent;
```

Spannen des Wickelgutes

Bei Freigabe des Wickelrechners ("Enable" = TRUE) wird das Wickelgut zunächst über den additiven Drehzahloffset gespannt. Um eine Schlaufenbildung im Wickelgut zu vermeiden, sollte die Leitachse nicht drehen. Das Wickelgut gilt als gespannt, wenn der Zugkraftsollwert des Zugreglers einen bestimmten minimalen Wert überschreitet und die relative Abweichung zwischen Zugkraftsollwert und -Istwert kleiner einer bestimmten Schranke ist.

Reibkompensation

Mit Hilfe verschiedener Eingänge wird eine Reibmomentkompensation umgesetzt. Die Annäherung des Reibungsverhaltens ist wie folgt definiert:

Die Reibung nimmt vom Stillstandsreibmoment zunächst bis zu einer bestimmten Drehzahl ab. Danach steigt das Reibmoment linear mit der Wickelachsdrehzahl bis zur maximalen Drehzahl ("S-0-0091 Geschwindigkeits-Grenzwert bipolar", bzw. "A-0-0032 Geschwindigkeitsgrenzwert, positiv", "A-0-0033 Geschwindigkeitsgrenzwert, negativ") an.

Für negative Drehzahlen gilt das Verhalten analog. Die Kurve wird an der y-Achse (Reibmoment) gespiegelt. Mit Hilfe dieses Reibmomentverhaltens, kann dem Effekt der Reibung durch eine Kompensation entgegen gewirkt werden.

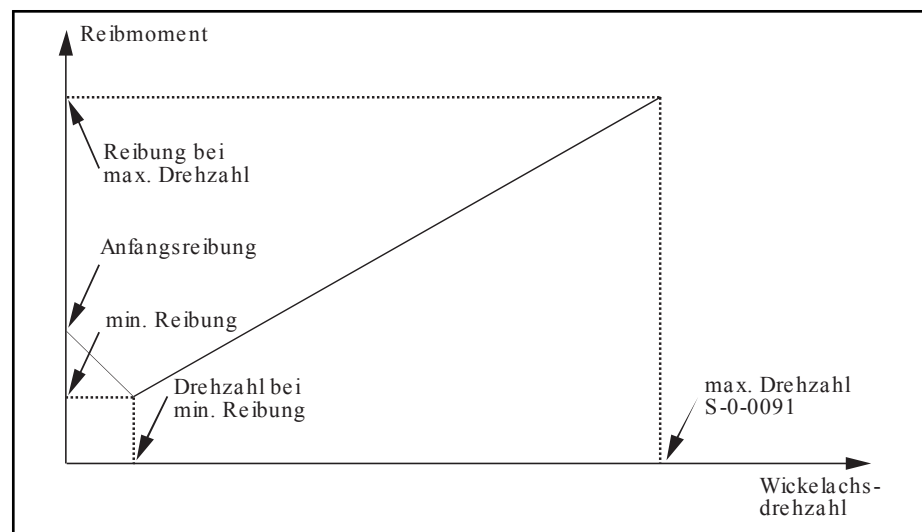


Abb. 8-31: Reibmomentkompensation

Kompensation des Beschleunigungsmoments

Für die Beschleunigung der Wickelachse ist ein zusätzliches Beschleunigungsmoment erforderlich. Dieses ist abhängig vom Wickeldurchmesser. Mit der Angabe von Material- und Stoffbahndaten wird eine durchmesserabhängige Trägheit des Wickels berechnet. Das berechnete Beschleunigungsmoment kann zudem gefiltert werden.

Drehmomentenbegrenzung und Vorsteuerung

Der Antrieb der Wickelachse arbeitet generell NICHT an seiner Drehmomentengrenze. Aus Sicherheitsgründen wird ein berechnetes Drehmoment zusätzlich eines einstellbaren additiven Anteils als Wert für die Drehmomentengrenze eingestellt. (dieser Wert sollte überhalb des benötigten Motormomentes liegen und den Antrieb im normalen Betrieb nicht begrenzen).

Eine Drehmomentengrenze, die proportional der Zugkraft und dem jeweiligen Wickeldurchmesser ist wird zyklisch berechnet.

$$M_{Grenz} \sim F_{Zug} * r_{Wickel} + M_{Beschl.} + M_{Reib}$$

Abb. 8-32: Drehmomentengrenze

Dieser zyklische Wert für die Drehmomentengrenze wird als additiver Drehmomentensollwert vorgesteuert.

Des Weiteren kann dieser Wert um einen proportionalen Anteil (TorqueLimitFactor) und einen additiven Anteil (MinTorqueLimit) erhöht werden, damit die Drehmomentengrenze im normalen Wickelbetrieb nicht wirkt.

$$M_{ResGrenz} \sim M_{Grenz} * (1 + TorqueLimitFactor) + MinTorqueLimit$$

Abb. 8-33: resultierende Drehmomentengrenze

Diese resultierende Drehmomentenbegrenzung entspricht dem Ausgang TorqueLimitbipolar. Standardmäßig wird die berechnete Drehmomentengrenze um den TorqueLimitFactor von 50% und einem minimalen TorqueLimit von 10% erhöht.

Während des Spannvorgangs ist es evtl. sinnvoll die Drehmomentenbegrenzung niedriger zu setzen (z. B. TorqueLimitFactor 10%) , wenn der Spann-

vorgang mit zuviel Drehmoment erfolgt. Nach erfolgtem Spannvorgang sollte der TorqueLimitFactor wieder erhöht werden.

Rollenwechsel

Nach einem Rollenwechsel ändert sich in der Regel der Wickeldurchmesser schlagartig. Der zuletzt berechnete interne Wert für den Wickeldurchmesser entspricht dann nicht mehr dem aktuellen Durchmesser. Damit die Durchmesserberechnung korrekt aufsetzen kann, ist ein Preset des Wickeldurchmessers erforderlich. Der Eingangswert für den Preset sollte annähernd dem tatsächlichen Wert des Wickeldurchmessers entsprechen.



Wird der Preset nicht durchgeführt, kann es zu Fehlfunktionen kommen. Daher sollte der Preset abgeschlossen sein, bevor der Durchmesserrechner wieder aktiv wird.

Der Wert für den Preset sollte wie folgt gewählt werden:

- Beim Aufwickler etwas kleiner bzw. gleich dem tatsächlichen Wickeldurchmesser
- Beim Abwickler etwas größer bzw. gleich dem tatsächlichen Wickeldurchmesser

Bei einem fliegenden Rollenwechsel muss sichergestellt werden, dass die Umfangsgeschwindigkeit der neuen Rolle, der aktuellen Rolle entspricht. Dies sollte entsprechend im SPS-Projekt in IndraLogic umgesetzt werden.

Filterung des Zugkraftistwerts

Bei niedrigen Zugkräften kann ein ungefilterter Zugkraftistwert zu einem Ausrasten des Wickelrechners führen. Daher sollte eine Filterung vorgenommen und der entsprechende Eingang beschaltet werden (z. B. 20ms).

Aktivierung der Korrektur der Drehmomentkonstante bei Synchronmotoren

Das von einem Motor entwickelte Drehmoment wird antriebsseitig anhand des gemessenen Motorstroms und einer sogenannten Drehmomentkonstante P-0-0051 "Drehmoment-/Kraft-Konstante" ermittelt. Diese sogenannte "Drehmoment- bzw. Kraftkonstante" ist jedoch physikalisch kein statischer Wert.

Er ist abhängig von:

1. der Höhe des aktuell fließenden Motorstroms.
(Die Konstante reduziert sich bei Strömen größer I_{Nenn}).
2. der Temperatur von Motorwicklung und Rotor/Läufer.
(Die Konstante reduziert sich bei steigender Temperatur).

Für den **Synchronmotor** kann eine korrigierte Drehmomentkonstante verwendet werden. Im Parameter **P-0-0556 "Achsregler-konfiguration" Bit 11 (gesetzt)** kann diese aktiviert werden.

Hierdurch werden folgende Einflüsse auf die Drehmoment-/Kraftkonstante berücksichtigt:

1. Magnetfeldschwächung durch magnetische Sättigung aufgrund erhöhtem Motorstrom gegenüber Nennstrom.
2. Magnetfeldschwächung durch erhöhte Motorwicklungstemperatur.
3. Magnetfeldschwächung durch erhöhte Rotortemperatur abhängig von der mittleren Drehzahl.

Die in diesem Fall aktuell wirkende Drehmomentkonstante kann im Parameter P-0-0450 "Drehmoment-/Kraft-konstante, aktuell" abgelesen werden.

Eine deutliche Abweichung von der Drehmomentkonstante P-0-0051 wird sich erst ab einer Strombelastung von 100 % (und mehr) des Nennstromes

S-0-0111 "Stillstandsstrom Motor" ergeben. D.h. die Korrektur wirkt umso besser, je besser der Motor ausgelastet ist. Für einen schlecht ausgelasteten Motor wird sich nur eine minimale bzw. keine Korrektur ergeben.



Für einen Asynchronmotor ist eine Korrektur der Drehmomentkonstante automatisch aktiviert und muss nicht separat im Parameter P-0-0556 eingestellt werden.

Siehe auch Funktionsbeschreibung des entsprechenden Indradrive-Antriebs.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_WinderTensionCtrlSpeedType01 aufgelistet und beschrieben.

Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

Zusätzlich können die Fehlercodes der Bausteine MB_DancerControlType03, MB_DiameterCalculatorType03, MB_DiameterMeasurementType01, IL_PID-Type01, RIL_LoopControl, sowie MB_GetCyclicParameterHandle, ML_Tech-MotionBase, auftreten.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#11B0	16#0020	Fehler in der State machine
INPUT_RANGE_ERROR	16#11B1	16#0001	Diameter.MechanicInertia < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0002	TensionCtrl.PreControlWeighting < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0003	MaxTension < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0004	TensionHighLimit < TensionLowLimit
INPUT_RANGE_ERROR	16#11B1	16#0005	Friction.StandstillTorque < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0006	Friction.MaxTorque < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0007	Friction.MinTorque < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0008	Friction.MinVelocity < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0009	WinderTension < 0.0
INPUT_RANGE_ERROR	16#11B1	16#000A	MinTension < 0.0
INPUT_RANGE_ERROR	16#11B1	16#000B	MaxTension < 0.0
INPUT_RANGE_ERROR	16#11B1	16#000C	Drehmomentwichtung falsch (A-0-0050 bzw. S-0-0086) Wichtung für Motormoment muss motorseitig sein
INPUT_RANGE_ERROR	16#11B1	16#000E	Achsnummer der Slaveachse falsch
INPUT_RANGE_ERROR	16#11B1	16#000F	Achsnummer der Masterachse falsch
INPUT_RANGE_ERROR	16#11B1	16#0010	SetpointStandstill < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0011	SetpointOpMode < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0012	DefaultTorqueLimit < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0013	MinTorqueLimit < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0014	TorqueLimitFactor < 0.0
INPUT_RANGE_ERROR	16#11B1	16#0015	DefaulTorqueLimit maximum
INPUT_RANGE_ERROR	16#11B1	16#0016	MinTorqueLimit maximum

ErrorID	Additional1	Additional2	Text
INPUT_RANGE_ERROR	16#11B1	16#0017	TorqueLimitFactor maximum
INPUT_RANGE_ERROR	16#11B2	16#0001	Wickelachse nicht in „BB“
RESOURCE_ERROR	16#0004	16#0000	Falsche Antriebsfirmware
RESOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert
RESOURCE_ERROR	16#0012	16#00xx	AxisRef.CntrlNo falsch
RESOURCE_ERROR	16#001C	16#00xx	Falsche Achstypen für Achse xx

Tab. 8-38: Fehlercodes FB MB_WinderTensionCtrlSpeedType01

Besonderheiten für MLC

Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter "S-0-0121 Lastgetriebe-Eingangsumdrehungen" und "S-0-0122 Lastgetriebe-Ausgangsumdrehungen" eingetragen werden.

Die Wichtungsart für die Master-Achse und Slave-Achse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

Den Parameter "A-0-0032 Geschwindigkeitsgrenzwert, positiv" bzw. "A-0-0033 Geschwindigkeitsgrenzwert, negativ" entsprechend der maximalen Drehzahl des Wickelantriebes setzen (dieser Wert kann im Parameter "S-0-0091 Geschwindigkeits-Grenzwert bipolar" ausgelesen werden).

Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z. B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!

Interpolation im Antrieb:

Im zyklischen Sercos-Kanal des entsprechenden Antriebs müssen der Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler", Parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler", "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar" und "S-0-0081 Drehmoment-/Kraft-Sollwert additiv" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus.

Interpolation in der Steuerung:

Die Zielgrößen sind die Steuerungsparameter "A-0-2605 ELS, Feinabgleich der Getriebeübersetzung, Prozessregler" und "A-0-2615 ELS, Geschwindigkeits-Sollwert additiv, Prozessregler", "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar" und "S-0-0081 Drehmoment-/Kraft-Sollwert additiv".

Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss der Parameter "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar" und "S-0-0081 Drehmoment-/Kraft-Sollwert additiv" eingetragen sein.

Besonderheiten für MLD

Bei Verwendung der Steuerung MLD-S mit übergeordneter Steuerung MLC muss für die Wickelachse Interpolation im Antrieb aktiviert sein (reale Achsen **entspr. Achse ► Eigenschaften**). Bei Verwendung der Steuerung MLD-M gibt es diese Einstellung nicht.

Die Wichtungsart für die Master-Achse und Slave-Achse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

Im zyklischen Sercos-Kanal des entsprechenden Antriebs müssen die Parameter "P-0-0694 Feinabgleich Getriebeübersetzung, Prozessregler" und Pa-

parameter "P-0-0690 Geschwindigkeits-Sollwert additiv, Prozessregler" und "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar" eingetragen sein ("UserCmdDataX").

Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus.

Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus (der Dialog ist im Projektbaum unterhalb des MLD-Knotens **Axis-Data**).

Die "AxisData-Struktur" des Antriebes muss parametrierbar sein ("P-0-1367 SPS Konfiguration Bit 6 ist gesetzt", bzw. Dialog SPS-Konfiguration).

Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssache zur Wickelachse muss in die Parameter "S-0-0121 Lastgetriebe-Eingangsumdrehungen" und "S-0-0122 Lastgetriebe-Ausgangsumdrehungen" eingetragen werden. Den Parameter "S-0-0091 Geschwindigkeits-Grenzwert bipolar" entsprechend der maximalen Drehzahl des Wickelantriebes setzen.

Stillstandsfenster Referenzachse:

Das Stillstandsfenster der Referenzachse ist auf einen möglichst kleinen Wert zu setzen (z. B. 0.1 U/min). Bei Stillstandsmeldung der Referenzachse rechnet der Durchmesserrechner nicht!

Für die Referenzachse kann der Funktionsbaustein auf keinen Wert für das Stillstandsfenster zugreifen. Intern wird dann ein Wert von 1 U/min angenommen. Ist dieser Wert für die Applikation nicht geeignet, kann dieser mittels folgender SPS-Codezeilen beeinflusst werden:

Deklaration

```
prMaStandstillWdw:    POINTER TO REAL; // for StandstillWindow master MLD
rTestMaStandstillWdw: REAL := 0.5;    // for StandstillWindow master MLD
```

Implementation

```
prMaStandstillWdw := ADR(BausteinInstanznameWickler.fbDiameterCalculator.rMaStandstillWindow);
prMaStandstillWdw^ := rTestMaStandstillWdw;
```

Der Wert für diese Variable bleibt nicht gespeichert, der Wert muss beim Einschalten der SPS also neu geschrieben werden.

Achsreferenz für die Masterachse:

Für die Technologiebausteine wird als Achsreferenz für die Masterachse empfohlen:

- Für eine virtuelle Leitachse die Achsreferenz "**VmAxis1**" verwenden.
Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 20 (VMA_1) eingestellt werden.
- Für eine reale Leitachse die Achsreferenz "**RmAxis1**" verwenden.
Der Parameter "P-0-1820.0.1 CCD: Konfigurationsliste Leitachsen" muss dann auf den Wert 23 (RMA_1) eingestellt werden.
Zusätzlich muss in diesem Fall noch die logische Adresse des Messgebers im Parameter "P-0-1820.0.2 CCD: Quelle des Messgeber-Lageistwertes" eingetragen werden.

Detaillierte Informationen können der Funktionsbeschreibung MLD entnommen werden.

Inbetriebnahme des Wicklers mit Zugkraftregelung und Drehzahlkorrektur

Zur Inbetriebnahme des Wicklers mit Zugkraftregelung und Drehzahlkorrektur siehe [Kap. 8.7.3 "Inbetriebnahme des Wicklers mit Zugkraftregelung und Drehzahlkorrektur"](#) auf Seite 573.

8.7.3 Inbetriebnahme des Wicklers mit Zugkraftregelung und Drehzahlkorrektur

Antriebsregelkreis

Es wird vorausgesetzt dass der Drehzahlregelkreis des Wickelantriebes eingestellt wurde. D. h. die Parameter "S-0-0100 Geschwindigkeitsregler-Proportionalverstärkung" und "S-0-0101 Geschwindigkeitsregler-Nachstellzeit" müssen der Applikation entsprechend eingestellt sein.

Im Folgenden wird die Inbetriebnahme des Wickelrechners in sieben Schritten beschrieben:

1. Festlegung des Wicklertyps
2. Festlegung der Drehrichtung der Wickelachse
3. Ermittlung des Reibmoments im Stillstand
4. Ermittlung des Reibmoments bei maximaler Drehzahl
5. Ermittlung des minimalen Reibmoments
6. abschließende Einstellungen und
7. erster Anlauf der Wickelachse

Kommandierung der Wicklerachse

Das Setzen von Betriebsart und Reglerfreigabe kann auf zwei unterschiedliche Arten erfolgen:

- Setzen der Betriebsart "ModeSyncVel" mit Hilfe des Achsinterfaces (dies wird empfohlen)
- Benutzung der PLCopen Funktionsbausteine "MC_Power", "MC_GearIn", "MC_Stop" zur Kommandierung der Achse

Festlegung des Wicklertyps

Zunächst muss festgelegt werden, welcher Wicklertyp vorliegt bzw. eingesetzt werden soll. Dazu wird am Eingang "WindDirection" die Wicklerrichtung festgelegt (TRUE: Abwickeln, FALSE: Aufwickeln).

Festlegung der Drehrichtung der Wickelachse

Die Drehrichtung des Wicklers kann mit Hilfe des Antriebsparameters "P-0-0108 Polarität Leitantrieb" bzw. des Achsparameters "A-0-2798 Polarität der Leitachspannung" eingestellt werden. Dreht sich die Wickelachse nicht in die gewünschte Richtung, ist der Parameter P-0-0108 bzw. der Achsparameter A-0-2798 entsprechend umzustellen. Die Polarität der Leitachspannung wird lediglich beim Einschalten des Funktionsbausteins überprüft ("Enable" = TRUE). Eine Änderung der genannten Parameter während des Betriebs des Funktionsbausteins zeigt somit keine Wirkung.

Alternativ kann die Drehrichtung über die Wichtungsart des Wickelantriebs geändert werden. Dies kann durch Drehen des Vorzeichens der Lage-Beschleunigungs- und der Drehmoment-Kraftdaten erfolgen (Parameter S-0-0055, A-0-0029, S-0-0043, S-0-0085).

Die folgende Abbildung zeigt die prinzipiellen Zusammenhänge bei Auf- bzw. Abwickeln.

rameters "A-0-0102 Geschwindigkeitsistwert" verfolgt werden. Der letzte Wert von "P-0-0109 Spitzendrehmoment-/Kraft-Begrenzung" entspricht dem maximalen Reibmoment bei maximaler Drehzahl.

4. Die Achsfreigabe "MODE_VEL" wird wieder zurückgenommen (alternativ mit dem PLCopen-Funktionsbaustein MC_Stop).

Der Parameter "P-0-0109 Spitzendrehmoment-/Kraft-Begrenzung" wird wieder auf seinen alten Wert (z. B. 400%) gesetzt.

Die Einstellung des Reibmoments maximaler Drehzahl "MaxTorque" ist somit abgeschlossen.

Ermittlung des minimalen Reibmoments

Das minimale Reibmoment wird mit Hilfe eines Bremsversuchs mit leerem Wickel ermittelt. Der Wickelantrieb wird von einer konstanten Geschwindigkeit (z. B. 50 min⁻¹) bis zum Stillstand abgebremst. Gleichzeitig werden die Antriebsparameter "S-0-0080 Drehmoment-/Kraft-Sollwert" (Achsparemeter A-0-0038 "Drehmoment/Kraft-Grenzwert bipolar") und "S-0-0040 Geschwindigkeitsistwert Geber 1" (Achsparemeter A-0-0102 "Geschwindigkeits-Istwert") mit einem Oszilloskop bzw. mit der Tracefunktion in IndraWorks aufgezeichnet.

Aus dem Drehmomentverlauf wird der Minimalwert abgelesen und kann der Struktur MB_WIND_FRICTION als "MinTorque" übergeben werden. Die korrespondierende Drehzahl wird als "MinTorqueVelocity" in MB_WIND_FRICTION eingetragen.

Abschließende Einstellungen

Schließlich werden die verbleibenden Eingangsgrößen festgelegt, die in der folgenden Tabelle zu sehen sind:

Eingangsgröße	Bemerkung/Einstellung
SetpointOpMode	Einstellung der gewünschten Laufzugkraft
SetpointStandstill	Einstellung des Stillstandzugs
ActTensionFilter	Soll die IstSpannung (als Messwert) gefiltert werden, muss an diesem Eingang des Funktionsbaustein ein Wert ungleich Null als Zeitkonstante angelegt werden
AccelTorqueFilter	Soll das Beschleunigungsmoment gefiltert werden, muss an diesem Eingang des Funktionsbausteins ein Wert ungleich Null als Zeitkonstante angelegt werden

Tab. 8-39: Abschließende Festlegung der verbleibenden Eingangsgrößen

Erster Anlauf der Wickelachse

Das Wickelgut wird ungespannt (leicht durchhängend) aufgelegt. Die Wickelachse muss entweder über das Achsinterface oder über die PLCopen Funktionsbausteine kommandiert werden. Danach sollte der Wickelrechner mit Setzen des Eingangs "Enable" aktiviert. Jetzt sollte das Wickelgut über die Bahnspannungsregelung gespannt werden. Der Ausgang "SpeedOffset" stellt die Geschwindigkeit dar, mit der gespannt wird.

Das Prozessgrößenfenster "WindowTension" kann z. B. auf einen Wert von 10 - 20% eingestellt werden.

Die Leitachse wird mit einer vergleichsweise niedrigen Drehzahl (z. B. 10 min⁻¹) freigegeben. Nach einigen Umdrehungen muss der Ausgang "DiameterEngaged" gesetzt sein. Hiermit wird angezeigt, dass die Durchmesserberechnung aktiv ist ("eingerastet"). Diese ist abhängig von der Bahnspannung bzw. dem Istmoment an der Wickelachse.

Bei eingerastetem Wickelrechner können Stillstands- und Laufzugkraft überprüft werden. Dazu wird der Wert "CurrentTension" aus der Datenstruktur

[MB_WIND_TENS_DIAG](#) , [Seite 618](#), beobachtet. In dieser Datenstruktur können zudem der aktuelle Drehmomentgrenzwert "TorqueLimit", das aktuelle Beschleunigungsmoment "AccelerationTorque" und das aktuelle angenommene Reibmoment "FrictionTorque" betrachtet werden.

Schließlich stellt der Funktionsbaustein an seinen Ausgängen verschiedene Prozessgrößen zur Verfügung. Der berechnete aktuelle Durchmesser des Wickels (Ausgang "CurrentDiameter") sollte mit geringfügiger, vernachlässigbarer Abweichung dem tatsächlichen Durchmesser des Wickels entsprechen. Ferner werden der Getriebefeinabgleich zur Drehzahlachsführung der Antriebsachse des Wickels (Ausgang "GearRatioFineAdjust"), das aktuelle Trägheitsmoment des Wickels (Ausgang "CurrentInertia") und der Drehzahloffset (Ausgang "SpeedOffset"), das additive Drehmoment "AddTorque" und der Drehmomentengrenzwert (Ausgang "TorqueLimitBipolar") ausgegeben.

Momentengrenzwertüberwachungen

Beim ersten Hochlaufen der Steuerung kann es zu einer Warnung des Antriebes "E2056 Momenten-Grenzwert = 0" kommen, wenn der Wicklerbaustein noch nicht eingeschaltet ist.

Um dies zu verhindern, kann in der Anlaufphase des SPS-Programmes (z. B. Warmstart) der entsprechende zyklische Datencontainer, indem sich der Wert für den Parameter "S-0-0092 Drehmoment/Kraft-Grenzwert bipolar" befindet, mit einem Wert ungleich Null beschrieben werden, z. B. 100%.

Kommt es bei eingeschaltetem Baustein zur Warnung "E8260 Moment-/Kraft-Sollwertbegrenzung aktiv", kann diese Warnung durch Setzen des Bit 4 des Parameters "P-0-0556 Achsregler-Konfiguration" beseitigt werden.

In seltenen Fällen (z. B. mehrfach F8078 "Fehler im Drehzahlregelkreis") kann es notwendig sein, die Geschwindigkeitsreglerüberwachung zu deaktivieren (Setzen des Bit 1 des Parameters P-0-0556 " Achsregler-Konfiguration").

8.8 Ergänzende Funktionsbausteine

8.8.1 Einführung und Übersicht

In diesem Abschnitt sind Funktionsbausteine zusammengefasst, die Funktionalitäten der verschiedenen Wicklerrechner und des Tänzerlagereger ergänzen können. Sie können zusätzlich in das SPS-Projekt in IndraLogic eingebunden werden.

Für zusätzliche Funktionalitäten (z. B. Restlaufzeit für einen Rollenwechsel) stehen weitere folgende Bausteine zur Verfügung:

Bezeichnung	Typ	Beschreibung
MB_CalcInertiaLimitType02, Seite 577	FB	Berechnet das maximale Trägheitsmoment des Wickelgutes auf dem Wickel
MB_UnwindMaterialType02, Seite 579	FB	Berechnet für einen Abwickler Prozessgrößen, wie die abgewickelte - und die restliche Länge der Stoffbahn, Materialstärke und Restlaufzeit für einen Rollenwechsel.
MB_WindTaperProfileType01, Seite 585	FB	Berechnet eine Wickelhärtekennlinie für einen Aufwickler
MB_WindSpeedControlAdaptionType01, Seite 588	FB	Berechnet für einen Wickler eine Kp-Adaption für den Drehzahlregler

Bezeichnung	Typ	Beschreibung
MB_TraverseControlType01, Seite 592	FB	Steuert eine translatorische Verlegerachse zum Verlegen von Wickelmaterial, das kleiner als die Breite des Wickels ist (z. B. Draht)
MB_DancerCalcControlParamType01, Seite 608	FB	Funktionsbaustein berechnet aus physikalischen Daten des Tänzersystems (Durchmesser der Zugwalze bzw. Hülsendurchmesser beim Wickler sowie Verzögerungszeit des Systems) die Reglerparameter des Tänzerlagereglers

Tab. 8-40: Kurzbeschreibung zusätzlicher Funktionsbausteine für verschiedene Wicklertypen aus der Bibliothek RMB_TechWinder

8.8.2 MB_CalcInertiaLimitType02

Kurzbeschreibung

Der Funktionsbaustein stellt, im Gegensatz zu den anderen in dieser Dokumentation beschriebenen Funktionsbausteinen, einen nicht zyklischen Baustein dar. Er wird gezielt, z. B. zur Initialisierung eines anderen Funktionsbausteins aufgerufen.

In diesem Funktionsbaustein werden die Materialeigenschaften und Abmessungen der Stoffbahn ausgewertet und das maximale Trägheitsmoment beim maximalen Durchmesser "MaxDiameter" für das Wickelgut auf dem Wickel berechnet. Das berechnete maximale Trägheitsmoment "MaxInertia" kann z. B. für eine Drehzahlregler-Adaption (Adaption des Parameters "S-0-0100 Geschwindigkeitsregler-Proportionalverstärkung") verwendet werden. Hierbei kann die Proportionalverstärkung zwischen einem Minimal- und Maximalwert angepasst werden. Es kann beispielsweise eine lineare Zunahme des Faktors der Proportionalverstärkung eingestellt werden.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechWinder
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_TechWinder.lib

Tab. 8-41: Referenztable MB_CalcInertiaLimitType02

Schnittstellenbeschreibung

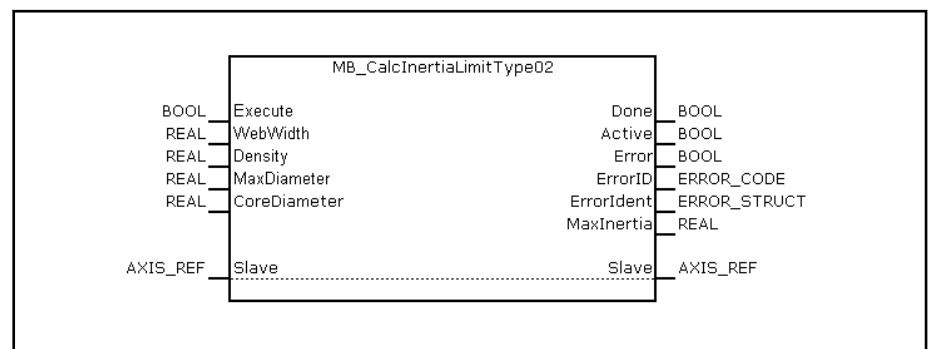


Abb. 8-35: Funktionsbaustein MB_CalcInertiaLimitType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Einschalten des Bausteins (flankengesteuert)
	WebWidth	REAL	Breite der Stoffbahn [mm]
	Density	REAL	Materialdichte der Stoffbahn [kg/dm³]

E/A-Typ	Name	Datentyp	Beschreibung
	MaxDiameter	REAL	Maximaler Durchmesser des Wickels [mm]
	CoreDiameter	REAL	Hülsendurchmesser des Wickels [mm]
VAR_OUTPUT	Done	BOOL	Maximales Trägheitsmoment "MaxInertia" berechnet
	Active	BOOL	Verarbeitung, Baustein aktiv
	Error	BOOL	Fehler aufgetreten
	ErrorID	ERROR_CODE	Fehlercode
	ErrorIdent	ERROR_STRUCT	Erweiterte Fehleranzeige Fehlerbehandlung, Seite 579
	MaxInertia	REAL	Maximales Trägheitsmoment [kg/m ²] bezogen auf die Motorseite
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Achsstruktur der Wickelachse

Tab. 8-42: Schnittstellenvariablen FB MB_CalcInertiaLimitType02

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
Webwidth	REAL	0.0	n. def.	0.0	Steigende Flanke an "Execute"
Density	REAL	0.0	n. def.	1.0	Steigende Flanke an Execute
MaxDiameter	REAL	> 0.0	n. def	2000.0	Steigende Flanke an "Execute"
CoreDiameter	REAL	> 0.0	n. def.	100.0	Steigende Flanke an "Execute"

Tab. 8-43: Min-/Max- und Default-Werte MB_CalcInertiaLimitType02

Signal-Zeit-Diagramm

Das nachfolgend dargestellte Signal-Zeit-Diagramm für die Betriebsein- und -ausgänge zeigt das flankengesteuerte Verhalten. Erfolgt am Eingang "Execute" eine positive Flanke, wird der Baustein aktiv. Die Verarbeitung, die hier beispielhaft mit einer Dauer von zwei Zyklen (zwei Kästchen in der Darstellung) angesetzt wird, wird durch den Ausgang "Active" angezeigt. Nach der Verarbeitung wird der Ausgang "Done" gesetzt. Dieser Zustand bleibt erhalten, wenn "Execute" gesetzt bleibt und kein Fehler ("Error") auftritt. Ist "Execute" nicht gesetzt, wird "Done" nach einem Zyklus gelöscht. Der Ausgang "Error" wird im Falle eines Fehlers gesetzt. "Error" bleibt gesetzt, bis "Execute" zurückgesetzt wird. Ist "Execute" nicht gesetzt, wird die Fehlermeldung am Ausgang "Error" nach einem Zyklus gelöscht.

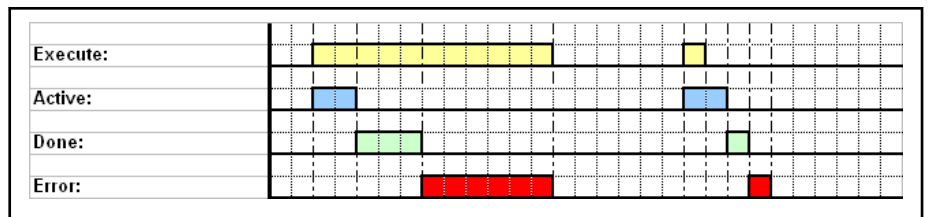


Abb. 8-36: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_CalcInertiaLimitType02

Funktionsbeschreibung

Mit Hilfe des Hülsendurchmessers des Wickels "CoreDiameter", der Bahnbreite "WebWidth" und der Dichte der Stoffbahn "Density" wird für den angegebenen maximalen Durchmesser "MaxDiameter" das Trägheitsmoment "MaxInertia" berechnet.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_CalcInertiaLimitType02 aufgelistet und beschrieben.

Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1140	16#0020	Fehler in der Statemachine
INPUT_RANGE_ERROR	16#1141	16#0001	Webwidth < 0.0
INPUT_RANGE_ERROR	16#1141	16#0002	Density < 0.0
INPUT_RANGE_ERROR	16#1141	16#0003	MaxDiameter < 0.0
INPUT_RANGE_ERROR	16#1141	16#0004	CoreDiameter < 0.0
INPUT_RANGE_ERROR	16#1141	16#0005	MaxDiameter < CoreDiameter
INPUT_RANGE_ERROR	16#1141	16#0006	Nummer der Slaveachse falsch
RESSOURCE_ERROR	16#0004	16#0000	Falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert

Tab. 8-44: Fehlercodes FB MB_CalcInertiaLimitType02

Notwendige Hardware

- CMLx, XMx, VPx
- IndraDrive

Notwendige Firmware

- MLC Firmware FWA-CMLXX*-MLC-14VRS
XX - Gerätevariante, z. B. CML65
- IndraDrive Antriebsfirmware

Notwendige Software

- IndraWorks 14VRS für MLC bzw. MLD

Notwendige Parametrierung

Eine real vorhandene mechanische Getriebeübersetzung von der Antriebssachse zur Wickelachse muss in die Parameter "S-0-0121 Lastgetriebe-Eingangsumdrehungen" und "S-0-0122 Lastgetriebe-Ausgangsumdrehungen" eingetragen sein.

8.8.3 MB_UnwindMaterialType02

Kurzbeschreibung

Der Funktionsbaustein stellt einen Materialzähler dar. Er berechnet für einen Abwickler die zyklischen Prozessgrößen abgewickelte Länge der Stoffbahn "UnrolledLength", restliche Länge der Stoffbahn "RestLength", die Materialstärke "MaterialThickness" sowie die Restlaufzeit "RemainRunTime".

Mit Hilfe dieser Daten kann ein Rollenwechsel vorbereitet werden.

Als Eingangsgrößen dienen der aktuelle Wickeldurchmesser "CurrentDiameter" und der Restrollendurchmesser "RestRollDiameter".

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechWinder

Tab. 8-45: Referenztable MB_UnwindMaterialType02

Schnittstellenbeschreibung

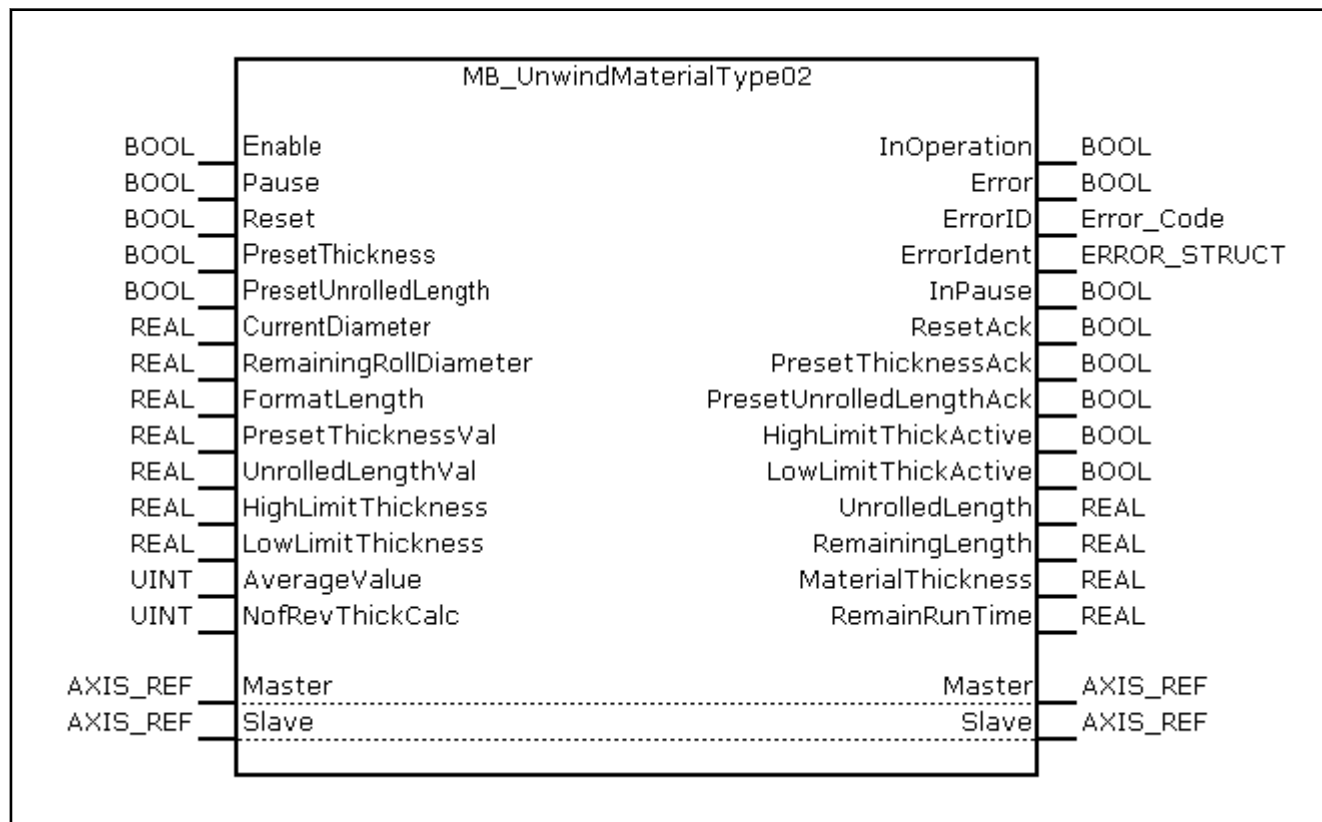


Abb. 8-37: Funktionsbaustein MB_UnwindMaterialType02

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Einschalten des Bausteins (pegelgesteuert)
	Pause	BOOL	Materialzählung angehalten
	Reset	BOOL	Zurücksetzen der Werte für abgewickelte Länge der Stoffbahn, restliche Länge der Stoffbahn
	PresetThickness	BOOL	Preset für die Materialstärke flankengesteuert
	PresetUnrolledLength	BOOL	Preset für die abgewickelte Länge der Stoffbahn flankengesteuert
	CurrentDiameter	REAL	Aktueller Wickeldurchmesser [mm] vom Durchmesserrechner-Baustein
	RemainingRollDiameter	REAL	Restrollendurchmesser [mm]
	FormatLength	REAL	Formatlänge [mm] (entspricht einer abgewickelten Umdrehung der Referenzachse)

E/A-Typ	Name	Datentyp	Beschreibung
	PresetThicknessVal	REAL	Preset-Wert für die Materialstärke [µm]
	UnrolledLengthVal	REAL	Preset-Wert für die abgewickelte Länge der Stoffbahn [mm]
	HighLimitThickness	REAL	obere Grenze für die Materialstärke [µm]
	LowLimitThickness	REAL	untere Grenze für die Materialstärke [µm]
	AverageValue	UINT	Anzahl der Materialstärkewerte, die gemittelt werden. Eingang wird auf maximal 30 begrenzt
	NofRevThickCalc	UINT	Anzahl der Umdrehungen, nach der eine Materialstärkeberechnung erfolgt
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Fehler aufgetreten
	ErrorID	ERROR_CODE	Fehlercode
	ErrorIdent	ERROR_STRUCT	Erweiterte Fehleranzeige Fehlerbehandlung, Seite 584
	InPause	BOOL	Materialberechnung angehalten
	ResetAck	BOOL	Reset ausgeführt
	PresetThicknessAck	BOOL	Materialstärke vorbesetzt
	PresetUnrolledLengthAck	BOOL	Abgewickelte Länge vorbesetzt
	HighLimitThickAchieved	BOOL	Aktuell berechnete Materialstärke hat obere Grenze erreicht (Wert wird verworfen)
	LowLimitThickAchieved	BOOL	Aktuell berechnete Materialstärke hat untere Grenze erreicht (Wert wird verworfen)
	RemainingLength	REAL	Restlänge der Stoffbahn auf dem Wickel [mm]
	MaterialThickness	REAL	Materialstärke der Stoffbahn [µm]
	UnrolledLength	REAL	abgewickelte Länge der Stoffbahn [mm]
	RemainRunTime	REAL	Restlaufzeit bis zum Rollenwechsel (bezogen auf den Restrollendurchmesser) [s]
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Achsstruktur der Referenzachse
	Slave	AXIS_REF	Achsstruktur der Wickelachse

Tab. 8-46: Schnittstellenvariablen FB MB_UnwindMaterialType02

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Reset	BOOL			FALSE	Kontinuierlich
PresetThickness	BOOL			FALSE	Kontinuierlich

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
PresetUnrolledLength	BOOL			FALSE	Kontinuierlich
CurrentDiameter	REAL	0.0	n. def.	0.0	Kontinuierlich
RestRollDiameter	REAL	0.0	n. def.	0.0	Steigende Flanke an "Enable"
FormatLength	REAL	0.0	n. def.	0.0	Steigende Flanke an "Enable"
PresetThicknessVal	REAL	> 0.0	n. def.	0.0	Kontinuierlich
UnrolledLengthVal	REAL	0.0	n. def.	0.0	Kontinuierlich
HighLimitThickness	REAL	0.0	n. def.	500.0	Kontinuierlich
LowLimitThickness	REAL	0.0	n. def.	10.0	Kontinuierlich
AverageValue	UINT	0	30	1.0	Kontinuierlich
NofRevThickCalc	UINT	0	n. def.	50	Kontinuierlich

Tab. 8-47: Min-/Max- und Default-Werte des Funktionsbausteins MB_Unwind-MaterialType02

Signal-Zeit-Diagramm

Im Folgenden wird das Signal-Zeit-Diagramm erläutert.

Mit dem Eingang "Enable" wird der Funktionsbaustein in Betrieb genommen. Dies wird am Ausgang "InOperation" angezeigt. Tritt ein Fehler auf, wird der Ausgang "Error" gesetzt und die Betriebsbereitschaft unterbrochen. Die Ausführung des Funktionsbausteins kann durch den Eingang "Pause" unterbrochen werden. Dies wird am Ausgang "InPause" signalisiert. Das Zurücksetzen des Bausteins (Eingang "Reset") oder ein Setzen von Materialparametern (Eingänge "PresetThickness" und "PresetUnrolledLength") erfolgt unabhängig vom Eingang "Pause". Jedoch ist es empfehlenswert, den Funktionsbaustein in "Pause" zu schalten. Dadurch lassen sich Preset und Reset kontrollierter und übersichtlicher durchführen. Das Setzen und Zurücksetzen wird an den Ausgängen "ResetAck", "PresetThicknessAck" und "PresetUnrolledLengthAck" angezeigt.

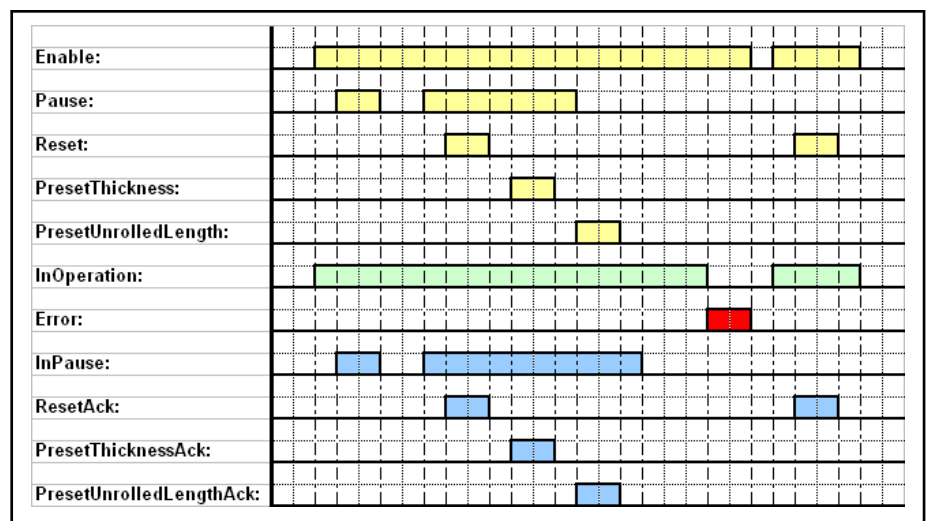


Abb. 8-38: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des Funktionsbausteins MB_UnwindMaterialType02

Funktionsbeschreibung

Berechnung der Materialgrößen

Die restliche Länge der Stoffbahn "RestLength" wird mit Hilfe des aktuellen Durchmessers berechnet. D. h., Schwankungen im aktuellen Durchmesser sind auch in der Restlängenberechnung ("RestLength") zu verfolgen. Gleiches gilt für die Materialstärkenberechnung ("MaterialThickness"), die wiederum mit dem Wert der Restlänge arbeitet, sowie für die Restlaufzeitberechnung ("RemainRunTime"), die Materialstärke und Geschwindigkeit der Referenzachse verwendet. Je genauer der aktuelle Durchmesser ermittelt wird, desto genauer sind die zu berechnenden Ausgänge des Bausteines "UnrolledLength", "RestLength", "MaterialThickness" und "RemainRunTime".

Weitere Eingangsgrößen werden verwendet. Um physikalisch sinnvolle Werte für die Materialstärke zu erhalten, wird diese mit "HighLimitThickness" und "LowLimitThickness" begrenzt. Diese Begrenzung wirkt bei der Mittelung der Materialstärke. Eine feste Anzahl von Werten "AverageValue" wird gemittelt. Eine Materialstärkeberechnung wird alle "NofRevThickCalc" Umdrehungen der Wickelachse berechnen.

Preset der Materialgrößen

Zu Beginn des Materialzählers werden die Ausgänge initialisiert. Dies erfolgt mit den Werten "PresetThicknessVal" und "UnrolledLengthVal". Diese Werte können zusätzlich während des Betriebs als Preset-Werte eingesetzt werden. Durch die Eingänge "PresetThickness" und "PresetUnrolledLength" werden diese gesetzt.

Reset des Materialrechners

Ferner kann der Materialzähler durch den Eingang "Reset" zurückgesetzt werden. Das Zurücksetzen ("Reset") als auch das Vorbesetzen der Ausgänge (Preset) kann jederzeit erfolgen, wenn der Funktionsbaustein aktiv ist.

Mittelwertbildung der Materialstärke

Die berechnete Materialstärke kann mit Setzen des Eingangs "AverageValue" gemittelt werden. An "AverageValue" wird die Anzahl der Materialstärkewerte eingetragen, über die gemittelt werden soll.

Zur Mittelwertbildung tragen nur die intern berechneten aktuellen Durchmesser bei, die innerhalb der Grenzen von "HighLimitThickness" und "LowLimitThickness" liegen. Außerhalb liegende Werte werden verworfen. Für "AverageValue" sind Werte zwischen 1 und 10 zweckmäßig. Aus diesem Grund wird "AverageValue" auf maximal 30 begrenzt. Wird ein größerer Wert eingege-

ben, wird "AverageValue" auf 30 gesetzt. Es wird allerdings keine Fehlermeldung ausgegeben, da es sich nicht um einen kritischen Vorgang handelt. Je höher die Zahl gewählt wird, desto glatter wird der berechnete Wert der Materialstärke. Allerdings nimmt mit zunehmenden Werten für "AverageValue" die Genauigkeit ab, da der berechnete Wert aufgrund der "Filterung" hinterläuft. Daher sollte der Wert für "AverageValue" geeignet gewählt werden.



Nach dem Ausschalten der Steuerung werden die Speicher gelöscht. Somit sind die Werte für "UnrolledLength", "RestLength", "Materialthickness" und "RemainRunTime" beim Wiedereinschalten der Steuerung gleich Null.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_UnwindMaterialType02 aufgelistet und beschrieben.

Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#11A0	16#20	Fehler in der Statemachine
INPUT_RANGE_ERROR	16#11A1	16#01	CurrentDiameter < 0.0
INPUT_RANGE_ERROR	16#11A1	16#02	RestRollDiameter < 0.0
INPUT_RANGE_ERROR	16#11A1	16#03	FormatLength <= 0.0
INPUT_RANGE_ERROR	16#11A1	16#04	PresetThicknessVal < 0.0
INPUT_RANGE_ERROR	16#11A1	16#05	UnrolledLengthVal < 0.0
INPUT_RANGE_ERROR	16#11A1	16#06	HighLimitThickness < 0.0
INPUT_RANGE_ERROR	16#11A1	16#07	LowLimitThickness < 0.0
INPUT_RANGE_ERROR	16#11A1	16#08	HighLimitThickness < LowLimitThickness
INPUT_RANGE_ERROR	16#11A1	16#09	AverageValue < 0
INPUT_RANGE_ERROR	16#11A1	16#0A	NofRevThickCalc < 0
INPUT_RANGE_ERROR	16#11A1	16#0B	Nummer der Slaveachse falsch
INPUT_RANGE_ERROR	16#11A1	16#0C	Nummer der Masterachse falsch
RESSOURCE_ERROR	16#0004	16#0000	falsche Antriebsfirmware
RESSOURCE_ERROR	16#000F	16#0003	Antriebsfunktionspaket "MA" nicht aktiviert

Tab. 8-48: Fehlercodes FB MB_UnwindMaterialType02

Notwendige Hardware

- CMLx, XMx, VPx
- IndraDrive

Notwendige Firmware

- MLC Firmware FWA-CMLXX*-MLC-14VRS
XX - Gerätevariante, z. B. CML65
- IndraDrive Antriebsfirmware

Notwendige Software

- IndraWorks 14VRS für MLC bzw. MLD

Notwendige Parametrierung

Für die Wicklerachse muss Interpolation im Antrieb aktiviert sein.

(Reale Achsen ► entspr. Achse ► Eigenschaften)

Die Wichtungsart für die Masterachse und Slaveachse muss gleich sein. Es wird rotatorisch und Vorzugswichtung empfohlen.

8.8.4 MB_WindTaperProfileType01

Kurzbeschreibung Dieser Funktionsbaustein realisiert eine durchmesserabhängige Anpassung der Wickelhärte für einen Aufwickler.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechWinder
MLD ab MPx20 mit MA Funktionspaket	MX_TechWinder

Tab. 8-49: Referenztabelle MB_WindTaperProfileType01

Schnittstellenbeschreibung

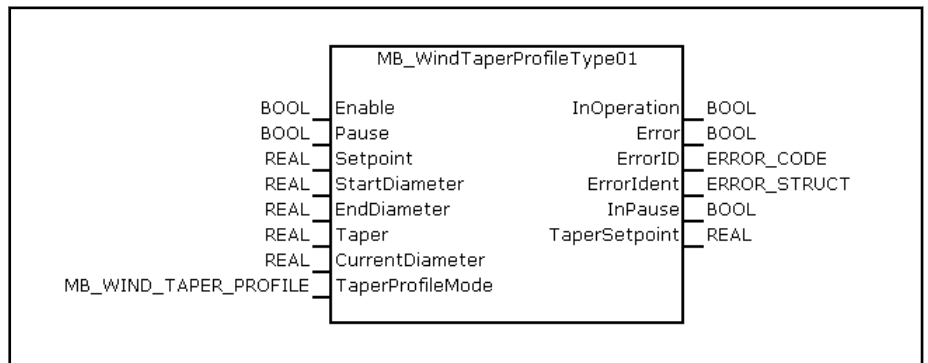


Abb. 8-39: Funktionsbaustein MB_WindTaperProfileType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegel-gesteuert)
	Pause	BOOL	FB pausiert, TaperTensionSetpoint bleibt "eingefroren"
	Setpoint	REAL	Sollwert [-] Zugkraftsollwert für einen Wickler mit Drehmomentenbegrenzung, Sollwert für pneumatische Kraft für einen Wickler mit Tänzer
	StartDiameter	REAL	Startdurchmesser [mm], ab dem die Zugkraft angepasst werden soll
	EndDiameter	REAL	Durchmesser [mm], ab dem die Zugkraft nicht mehr angepasst wird (konstant bleibt). Gilt nur für die lineare Kennlinie
	Taper	REAL	Wickelhärte [%] bestimmt den Verlauf der Wickelhärtekennlinie. Ist der Wert zwischen 0 und 100% nimmt die Zugkraft mit steigendem Durchmesser ab. Ist der Wert zwischen 0 und -100% so nimmt die Zugkraft mit steigendem Durchmesser zu
	CurrentDiameter	REAL	Aktueller Durchmesser [mm] des Aufwicklers (z.B. vom FB MB_WinderTensionCtrlType02)
	TaperProfileMode	MB_WIND_TAPER_PROFILE	Auswahl des Wickelhärteprofils MB_WIND_TAPER_PROFILE , Seite 618
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen

E/A-Typ	Name	Datentyp	Beschreibung
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose Fehlerbehandlung, Seite 588
	InPause	BOOL	Wickelhärte pausiert (Zugkraftsollwert wird "eingefroren")
	TaperSetpoint	REAL	Sollwert mit Berücksichtigung der Wickelhärtekennlinie

Tab. 8-50: Schnittstellenvariablen FB MB_WindTaperProfileType01

Min-/Max- und Default-Werte In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Setpoint	REAL	0.0	n.def.	0.0	Kontinuierlich
StartDiameter	REAL	> 0.0	n.def.	100.0	Kontinuierlich
EndDiameter	REAL	StartDiameter	n.def.	1000.0	Kontinuierlich
Taper	REAL	-100.0	100.0	0.0	Kontinuierlich
CurrentDiameter	REAL	0.0	n.def.	0.0	Kontinuierlich
TaperProfileMode	MB_WIND_TAPER_PROFILE			NO_PROFILE	Steigende Flanke an "Enable"

Tab. 8-51: Min-/Max- und Default-Werte des FBs MB_WindTaperProfileType01

Signal-Zeit-Diagramm Im folgenden Signal-Zeit-Diagramm sind ausgewählte Signale des Funktionsbausteins dargestellt.

Zusammenhang der Signale

Signaleingang	Signalausgang	Beschreibung
Enable	InOperation	Baustein in Betrieb nehmen
	Error	Betriebsbereitschaft wurde unterbrochen
Pause	InPause	Internen Reglerprozess anhalten

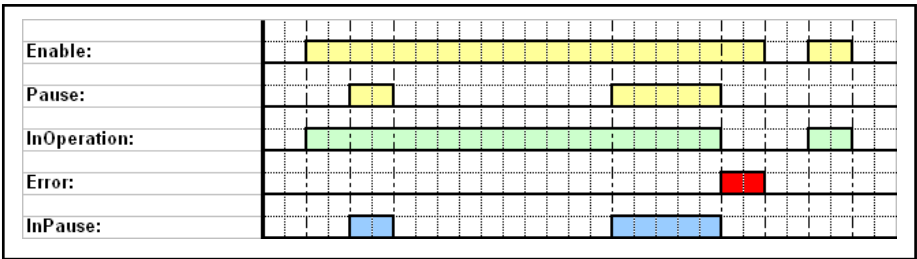


Abb. 8-40: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des FB MB_WindTaperProfileType01

Funktionsbeschreibung Beim Aufwickeln von Materialien ist es häufig wünschenswert, die Zugkraft mit der aufgewickelt wird, in Abhängigkeit des Wickeldurchmesser zu beeinflussen. Sehr häufig soll bei einem Aufwickelvorgang zu Beginn mit höherer

Zugkraft aufgewickelt werden und mit zunehmenden Wickeldurchmesser soll die Zugkraft dann reduziert werden.

Der Baustein MB_WindTaperProfileType01 beeinflusst einen vorgebbaren Sollwert nach einer auswählbaren Kennlinie und einem vorgebbaren Wickeldurchmesser. Die Wickelhärtekennlinie kann über den Eingang "Taper" (Wickelhärte) noch beeinflusst werden. Ist der Wert für die Wickelhärte zwischen 0% und 100%, so nimmt die Zugkraft mit steigendem Durchmesser ab, ist der Wert zwischen 0% und -100% so nimmt die Zugkraft mit steigendem Durchmesser zu. Je größer der absolute Wert für die Wickelhärte, desto stärker ist die Ab- bzw. Zunahme der Zugkraft.

Der Baustein kann für einen Wickler mit Drehmomentbegrenzung verwendet werden. Er wird dann dem Baustein MB_WinderTensionCtrlType02, Seite 542 vorgeschaltet, d. h. der Eingang "SetpointOpMode" des Winder-Funktionsbausteins MB_WinderTensionCtrlType02 wird mit dem Ausgang "TaperSetpoint" verschaltet.

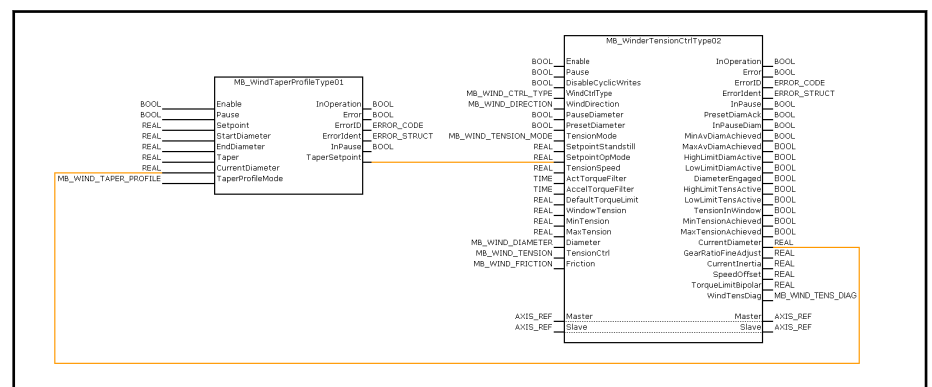


Abb. 8-41: MB_WindTaperProfileType01, Verschaltung

Wird der Baustein für einen Wickler mit Tänzer verwendet, so muss der Ausgang "TaperSetpoint" auf die pneumatische Kraft, mit der die Tänzerwalze beaufschlagt wird, wirken (z. B. analoger Spannungswert). Der Eingang "Setpoint" entspricht dann auch einem Wert für die pneumatische Kraft.

Lineare Wickelhärtekennlinie:

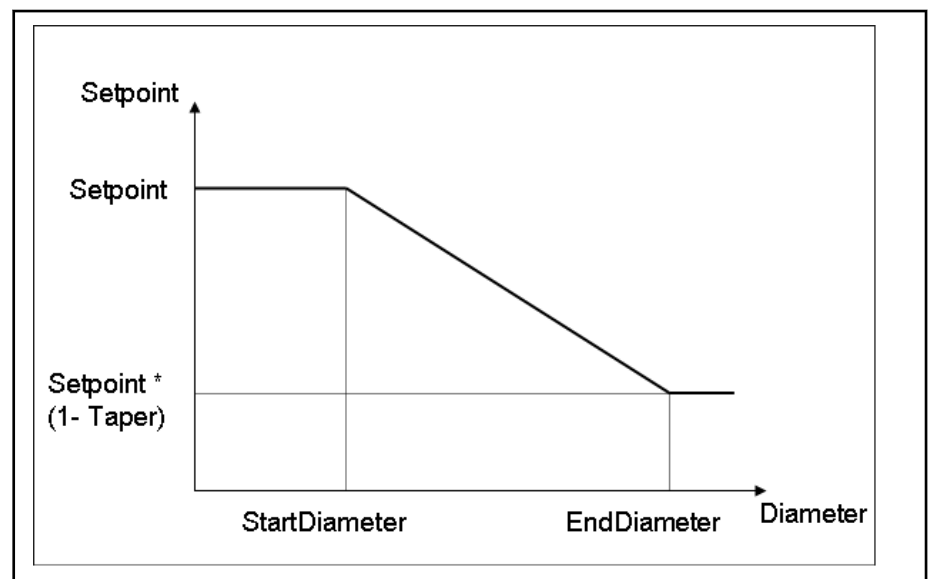


Abb. 8-42: MB_WindTaperProfileType01, Lineare Wickelhärtekennlinie

Hyperbolische Wickelhärtekennlinie:

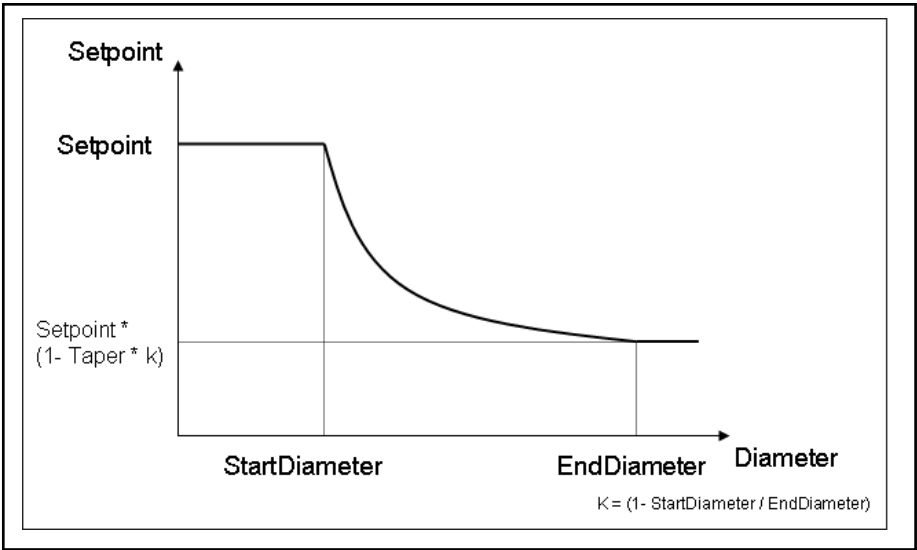


Abb. 8-43: MB_WindTaperProfileType01, Hyperbolische Wickelhärtekennlinie

Fehlerbehandlung In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_WindTaperProfileType01 aufgelistet und beschrieben.
Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1150	16#0020	Fehler in der Zustandsmaschine
INPUT_RANGE_ERROR	16#1151	16#0001	TensionSetpoint < 0.0
INPUT_RANGE_ERROR	16#1151	16#0002	StartDiameter <= 0.0
INPUT_RANGE_ERROR	16#1151	16#0003	EndDiameter <= 0.0
INPUT_RANGE_ERROR	16#1151	16#0004	EndDiameter <= StartDiameter
INPUT_RANGE_ERROR	16#1151	16#0005	Taper < -100.0 bzw. Taper > 100.0
INPUT_RANGE_ERROR	16#1151	16#0006	CurrentDiameter <= 0.0
INPUT_RANGE_ERROR	16#1151	16#0007	falscher TaperProfile-Mode

Tab. 8-52: Fehlercodes FB MB_WindTaperProfileType01

- Notwendige Hardware

Notwendige Firmware

Notwendige Software

Besonderheiten für MLC

Besonderheiten für MLD
- CMLx, XMx, VPx
 - IndraDrive
 - MLC Firmware FWA-CMLXX*-MLC-14VRS
XX - Gerätevariante, z. B. CML65
 - IndraDrive Antriebsfirmware
 - IndraWorks 14VRS für MLC

Keine besondere Parametrierung erforderlich.

Keine besondere Parametrierung erforderlich.

8.8.5 MB_WindSpeedControlAdaptionType01

Kurzbeschreibung Der Baustein führt eine Anpassung der Kp-Verstärkung des Drehzahlreglers des jeweiligen Wickelantriebes in Abhängigkeit des aktuellen Trägheitsmoments des Wickels und einem minimalen und maximalen Kp-Wert durch.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechWinder

Tab. 8-53: Referenztable MB_WindSpeedControlAdaptionType01

Schnittstellenbeschreibung

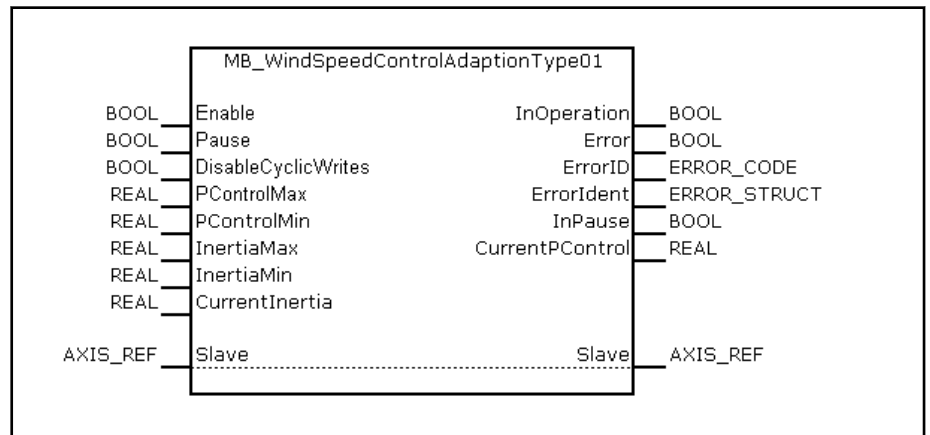


Abb. 8-44: Funktionsbaustein MB_WindSpeedControlAdaptionType01

E/A-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	Pause	BOOL	FB pausiert, CurrentPControl wird "eingefroren"
	DisableCyclicWrites	BOOL	Schreiben auf den Antriebsparameter S-0-0100 kann abgeschaltet werden
	PControlMax	REAL	Kp-Verstärkung für den maximalen Durchmesser [Nm/(rad/s)]
	PControlMin	REAL	Kp-Verstärkung für den minimalen Durchmesser [Nm/(rad/s)]
	InertiaMax	REAL	Trägheitsmoment beim maximalen Durchmesser bzw. Trägheitsmoment [kg*m²] (bezogen auf die Motorwelle)
	InertiaMin	REAL	Trägheitsmoment beim minimalen Durchmesser (Hülse) bzw. Trägheitsmoment [kg*m²] (bezogen auf die Motorwelle)
	CurrentInertia	REAL	Aktuelles Trägheitsmoment des Wickels [kg*m²] (bezogen auf die Motorwelle)
VAR_OUTPUT	InOperation	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose Fehlerbehandlung, Seite 591
	InPause	BOOL	FB pausiert (CurrentPControl "eingefroren")

E/A-Typ	Name	Datentyp	Beschreibung
	CurrentPControl	REAL	Aktueller Kp-Wert (Drehzahlregler) [Nm/(rad/s)]
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Achsstruktur Wickelachse

Tab. 8-54: Schnittstellenvariablen FB MB_WindSpeedControlAdaptionType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
DisableCyclicWrites	BOOL			FALSE	Steigende Flanke an "Enable"
PControlMax	REAL	PControlMin	n.def.	0.0	Steigende Flanke an "Enable"
PControlMin	REAL	0.0	n.def.	0.0	Steigende Flanke an "Enable"
InertiaMax	REAL	InertiaMin	n.def.	0.0	Steigende Flanke an "Enable"
InertiaMin	REAL	0.0	n.def.	0.0	Kontinuierlich
CurrentInertia	REAL	0.0	n.def.	0.0	Kontinuierlich

Tab. 8-55: Min-/Max- und Default-Werte des FB MB_WindSpeedControlAdaptionType01

Signal-Zeit-Diagramm

Im folgenden Signal-Zeit-Diagramm sind ausgewählte Signale des Funktionsbausteins dargestellt.

Zusammenhang der Signale

Signaleingang	Signalausgang	Beschreibung
Enable	InOperation	Baustein in Betrieb nehmen
	Error	Betriebsbereitschaft wurde unterbrochen
Pause	InPause	Internen Reglerprozess anhalten

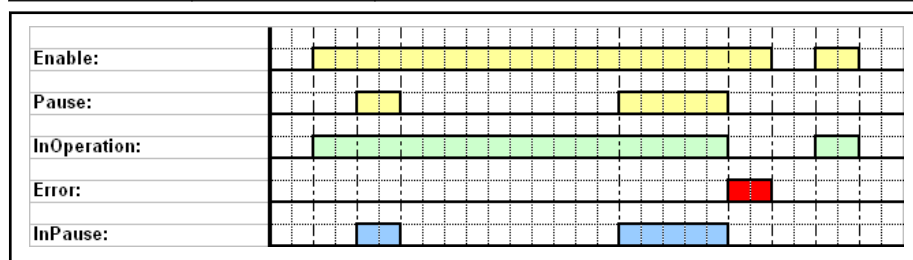


Abb. 8-45: Signal-Zeit-Diagramm für ausgewählte Ein- und Ausgänge des FB MB_WindSpeedControlAdaptionType01

Funktionsbeschreibung

Der Baustein berechnet zwischen einem minimalen und maximalen Kp-Wert für den Drehzahlregler linear den Wert für das Kp des Drehzahlreglers in Abhängigkeit des aktuellen Trägheitsmomentes. Das minimale und maximale Kp müssen jeweils bei leerer Hülse und bei vollem Wickeldurchmesser ermittelt werden. Des Weiteren werden der Werte für das minimale und maximale Trägheitsmoment (bei der Hülse bzw. bei vollem Wickeldurchmesser) benö-

tigt. Diese können mit dem Funktionsbaustein MB_CalcInertiaLimitType02 ermittelt werden.

Aus diesen Werten wird in Abhängigkeit des aktuellen Trägheitsmoments des Wicklers der aktuelle Kp-Wert des Drehzahlreglers berechnet.

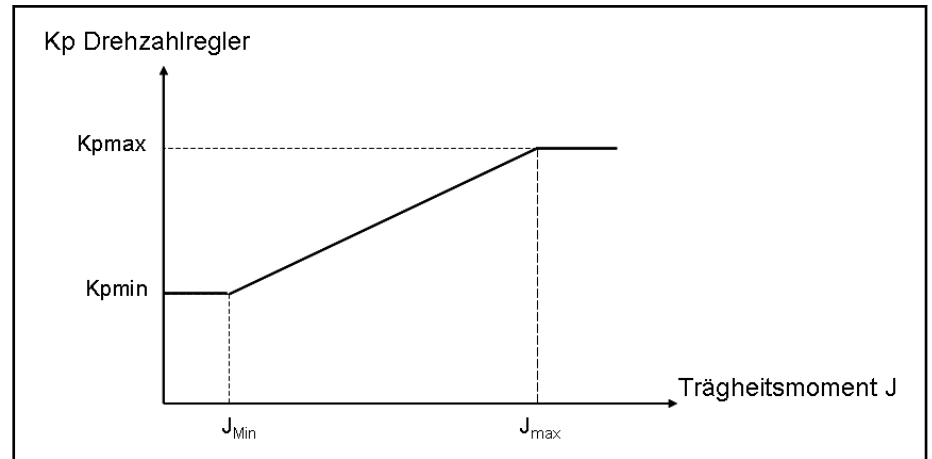


Abb. 8-46: MB_WindSpeedControlAdaptionType01

Dieser aktuelle Kp-Wert wird zyklisch in den Antriebsparameter "S-0-0100 Geschwindigkeitsregler-Proportionalverstärkung" geschrieben. Dazu muss der Parameter S-0-0100 in den zyklischen Datenkanälen zwischen Steuerung und Antrieb konfiguriert werden. Wird der Baustein ausgeschaltet, bleibt der letzte Wert von "CurrentPControl" anstehen.

Mit dem Eingang "DisableCyclicWrites" kann dieses Schreiben abgeschaltet werden. Der Ausgang "CurrentPControl" wird dann nicht vom Baustein intern geschrieben, sondern muss vom SPS-Programm in den Antriebsparameter transferiert werden.



Beim Einschalten der Steuerung werden die zyklischen Datenkanäle zwischen Steuerung und Antrieb immer mit Null initialisiert. Ist der Parameter "S-0-0100 Geschwindigkeitsregler-Proportionalverstärkung" zyklisch konfiguriert, muss nach dem Einschalten der Steuerung dafür gesorgt werden, dass der entsprechende zyklische Kanal mit einem sinnvollen Wert vorinitialisiert wird. Dieser gilt dann solange, bis der Baustein eingeschaltet wurde.

Fehlerbehandlung

In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_WindSpeedControlAdaptionType01 aufgelistet und beschrieben.

Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

ErrorID	Additional1	Additional2	Text
INPUT_RANGE_ERROR	16#11E0	16#0020	Fehler in der Statemachine
INPUT_RANGE_ERROR	16#11E1	16#01	PControlMax < 0.0
INPUT_RANGE_ERROR	16#11E1	16#02	PControlMin < 0.0
INPUT_RANGE_ERROR	16#11E1	16#03	InertiaMax < 0.0
INPUT_RANGE_ERROR	16#11E1	16#04	CurrentInertia < 0.0
INPUT_RANGE_ERROR	16#11E1	16#05	InertiaMin < 0.0
INPUT_RANGE_ERROR	16#11E1	16#06	PControlMax <= PControlMin

ErrorID	Additional1	Additional2	Text
INPUT_RANGE_ERROR	16#11E1	16#07	InertiaMax <= InertiaMin
DEVICE_ERROR	16#11E2	16#01	Achse nicht in BB

Tab. 8-56: Fehlercodes FB MB_WindSpeedControlAdaptionType01

Notwendige Hardware	• CMLx, XMx, VPx • IndraDrive
Notwendige Firmware	• MLC Firmware FWA-CMLXX*-MLC-14VRS XX - Gerätevariante, z. B. CML65 • IndraDrive Antriebsfirmware
Notwendige Software	• IndraWorks 14VRS für MLC
Besonderheiten für MLC	Im zyklischen Sercos-Kanal des entsprechenden Antriebs muss der "S-0-0100 Geschwindigkeitsregler-Proportionalverstärkung" eingetragen sein ("UserCmdDataX"). Diese Parametrierung der realen Achsen erfolgt in IndraWorks im Parametriermodus.



Die Initialwerte für Parameter im zyklischen Sercos-Kanal sind immer 0. D. h. beim Einschalten der Steuerung ist der Wert für den Parameter "S-0-0100 Geschwindigkeitsregler-Proportionalverstärkung" Null, wenn dieser im zyklischen Sercos-Kanal eingetragen ist und der Funktionsbaustein nicht enabled ist. Um dies zu verhindern, kann beim Einschalten der Steuerung der entsprechende Datencontainer einmalig mit einem Wert initialisiert werden, z. B.

```
AxisData[Antriebsname.AxisNo].dwUserCmdDataX_q.REAL_ := 10.0;
```

8.8.6 MB_TraverseControlType01

Kurzbeschreibung Der Baustein steuert eine translatorische Verlegerachse, die im Zusammenhang mit einem Wickler (rotatorische Achse), die Verlegung von unterschiedlichen Materialien ermöglicht. Dabei ist der Materialdurchmesser viel kleiner als die Breite des Wickels, so dass mehrere Material-Windungen nebeneinander gelegt werden. Es wird zwischen einem Auf- und Abwickelvorgang unterschieden.

Beim Aufwickeln ist die Geschwindigkeit der Verlegerachse an die Geschwindigkeit des Wicklers gekoppelt und sorgt so für eine gleichbleibende eng aneinander liegende Verlegung des Materials.

Beim Abwickeln sind mehrere Verfahren verfügbar, die abhängig von den gegebenen Voraussetzungen (z. B. vorhandene Hardware/Kraftmessdosen) ausgewählt werden können.

Ausgangsgrößen des Bausteins sind eine Zielposition und eine Positioniergeschwindigkeit für einen Positionierbaustein MC_MoveAbsolute bzw. "Position" und "Velocity" der Betriebsart "ModePosAbs" des Achsinterface für eine translatorische Achse.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechWinder

Tab. 8-57: Referenztablelle Funktionsbaustein MB_TraverseControl01

Schnittstellenbeschreibung

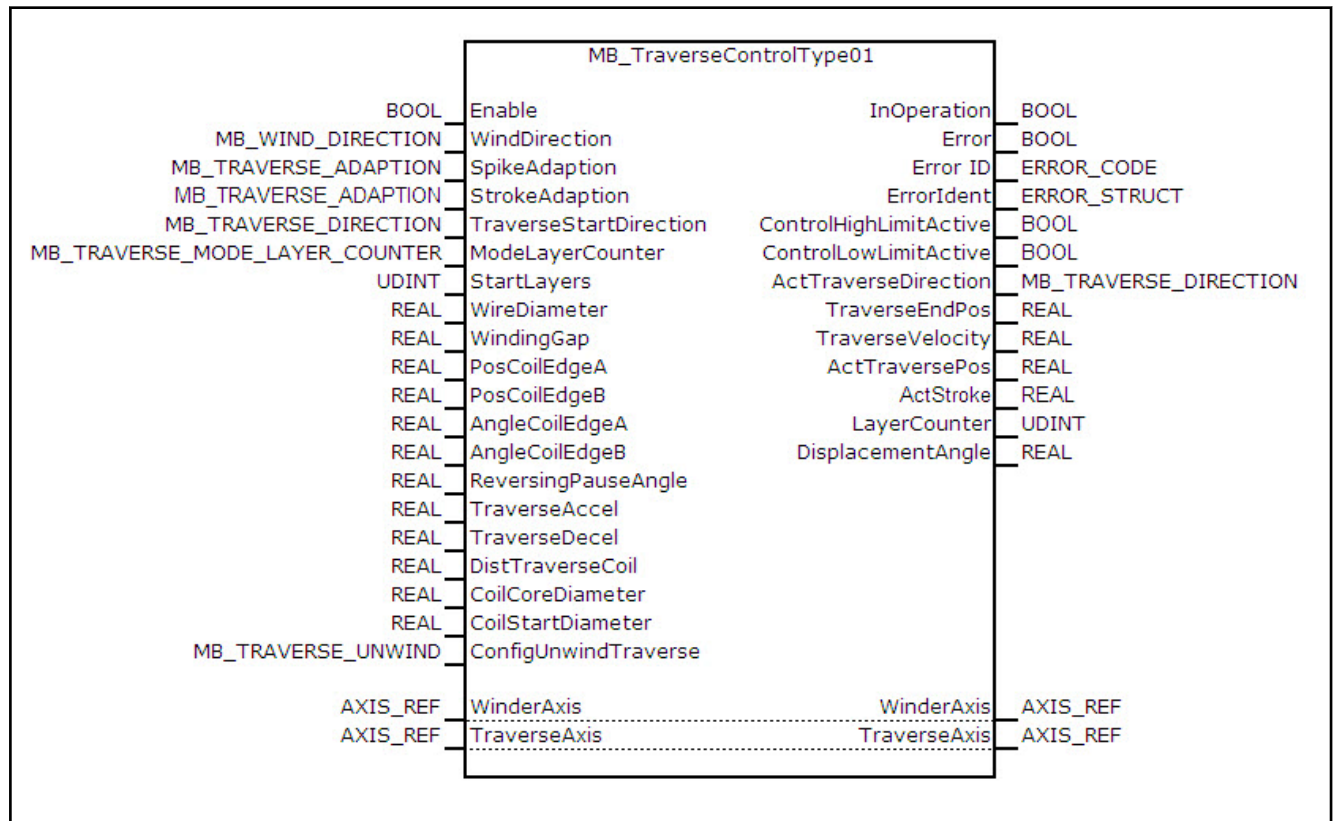


Abb. 8-47: Funktionsbaustein MB_TraverseControlType01

I/O-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins
	WindDirection	MB_WIND_DIRECTION	<i>Wickelrichtung</i> - Aufwickler REWIND (16#0000) - Abwickler UNWIND (16#0001) MB_WIND_DIRECTION , Seite 618
	SpikeAdaption	MB_TRAVERSE_ADAPTION	Ein- und Ausschalten der Geschwindigkeitsadaption um den fehlenden Weg während der Beschleunigungs- bzw. Verzögerungsphase zu kompensieren. MB_TRAVERSE_ADAPTION , Seite 612
	StrokeAdaption	MB_TRAVERSE_ADAPTION	Ein- und Ausschalten der Vorlaufadaption zum Spulenrand. Im Eingeschalteten Zustand fährt der Verleger um den Wert "ActStroke" über den Spulenrand hinaus, um zu gewährleisten, dass der Draht bzw. das Material das Ende der Spule erreicht. Im ausgeschalteten Zustand endet die Verlegerbewegung am Spulenrand. MB_TRAVERSE_ADAPTION , Seite 612

I/O-Typ	Name	Datentyp	Beschreibung
	TraverseStartDirection	MB_TRAVERSE_DIRECTION	Richtung, in die zu Beginn verlegt wird, also in Richtung von Spulenrand A zu Spulenrand B (A_TO_B) bzw. entgegengesetzt (B_TO_A). MB_TRAVERSE_DIRECTION, Seite 613
	ModeLayerCounter	MB_TRAVERSE_MODE_LAYER_COUNTER	<i>Auswahl, wie der Startwert für den Lagenzähler bei pos. Flanke an Enable ermittelt werden soll:</i> - Über Vorgabe an "StartLayers", dann wird der Startdurchmesser berechnet und der Eingang "CoilStartDiameter" nicht berücksichtigt. - Über den Startdurchmesser ("CoilStartDiameter") berechnen, dann wird der Eingang "StartLayers" nicht berücksichtigt. MB_TRAVERSE_MODE_LAYER_COUNTER, Seite 613
	StartLayers	UDINT	Anzahl der bereits aufgewickelten Lagen beim Start. Wird nur bei "INPUT_START_LAYERS" an „ModeLayerCounter“ ausgewertet.[-]
	WireDiameter	REAL	Durchmesser des Wickelguts (Draht etc.). [mm]
	WindingGap	REAL	Spalt zwischen den einzelnen Wicklungen (Drähten etc.) einer Lage. [% von WireDiameter]
	PosCoilEdgeA	REAL	Absolute Position des Spulenrands A. [mm]
	PosCoilEdgeB	REAL	Absolute Position des Spulenrands B. [mm]
	AngleCoilEdgeA	REAL	Winkel des Spulenrands gegen die Vertikale. (0: Spulenrand gerade, positiver Wert: Spulenrand nach außen geneigt d. h. Spule wird breiter) [Grad]
	AngleCoilEdgeB	REAL	Winkel des Spulenrands gegen die Vertikale. (0: Spulenrand gerade, positiver Wert: Spulenrand nach außen geneigt d. h. Spule wird breiter) [Grad]
	ReversingPauseAngle	REAL	Winkel, den der Wickler weiterdreht, während der Verleger am Spulenrand verweilt. [Grad]
	TraverseAccel	REAL	Beschleunigung der Verlegerachse. Entspricht dem Sollwert für MC_MoveAbsolute (bzw. Betriebsart "ModePosAbs" Techinterface) der Verlegerachse. [mm/s²]
	TraverseDecel	REAL	Verzögerung der Verlegerachse. Entspricht dem Sollwert für MC_MoveAbsolute der Verlegerachse. [mm/s²]
	DistTraverseCoil	REAL	Abstand zwischen Verleger und Spule, entspricht der Drahtlänge, die senkrecht vom Verleger bis zur Spule frei durch die Luft verläuft [mm]
	CoilCoreDiameter	REAL	Durchmesser der leeren Spule. [mm]
	CoilStartDiameter	REAL	Durchmesser der Spule zu Beginn (inkl. Draht). Dieser wird nur bei "CalculateStartLayers" am Eingang "ModeLayerCounter" ausgewertet. [mm]

I/O-Typ	Name	Datentyp	Beschreibung
	ConfigUnwindTraverse	MB_TRAVERSE_UNWIND	Struktur zur Konfiguration des Abwickelvorgangs MB_TRAVERSE_UNWIND, Seite 613
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose Fehlerbehandlung, Seite 607
	ControlHighLimitActive	BOOL	Ausgang des PID-Reglers, beim Abwickeln, hat oberen Grenzwert erreicht
	ControlLowLimitActive	BOOL	Ausgang des PID-Reglers, beim Abwickeln, hat unteren Grenzwert erreicht
	ActTraverseDirection	MB_TRAVERSE_DIRECTION	Aktuelle Verlegerichtung, von Spulenrand A zu Spulenrand B (A_TO_B) bzw. entgegengesetzt (B_TO_A) MB_TRAVERSE_DIRECTION, Seite 613
	TraverseEndPos	REAL	Ziel-Position (Umkehrpunkt) des Verlegers. Entspricht Positionssollwert des MC_MoveAbsolute der Verlegerachse (inkl. Stroke und Offset durch schräge Spulenränder). [mm]
	TraverseVelocity	REAL	Sollgeschwindigkeit des Verlegers. Entspricht Geschwindigkeitssollwert des MC_MoveAbsolute der Verlegerachse. [mm/min]
	ActTraversePos	REAL	Aktuelle Position des Verlegers. [mm]
	ActStroke	REAL	Aktueller Wert für den Vorlauf (Weg der über den Spulenrand hinaus gefahren wird). [mm]
	LayerCounter	UDINT	Anzahl der Lagen auf der Spule. [-]
	DisplacementAngle	REAL	Winkeldifferenz am Spulenende zum Startwinkel nach einer verlegten Lage. [Grad]
VAR_IN_OUT CONSTANT	WinderAxis	AXIS_REF	Achsreferenz auf die Wicklerachse
	TraverseAxis	AXIS_REF	Achsreferenz auf die Verlegerachse

Tab. 8-58: Schnittstellenvariablen FB MB_TraverseControlType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Standardwerte der Bausteineingänge aufgelistet.

Name	Typ	Min.-Wert	Max.-Wert	Standardwert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
WindDirection	MB_WIND_DIRECTION			REWIND	Kontinuierlich
SpikeAdaption	MB_TRAVERSE_ADAPTION			ADAPTION_OFF	Steigende Flanke an Enable
StrokeAdaption	MB_TRAVERSE_ADAPTION			ADAPTION_OFF	Steigende Flanke an Enable

Name	Typ	Min.-Wert	Max.-Wert	Standardwert	Übernahme
TraverseStartDirection	MB_TRAVERSE_DIRECTION			A_TO_B	Steigende Flanke an Enable
ModeLayerCounter	MB_TRAVERSE_MODE_LAYER_COUNTER			INPUT_START_LAYERS	Steigende Flanke an Enable
StartLayers	UDINT	0	n.def.	0	Steigende Flanke an Enable
WireDiameter	REAL	>0.0	n.def.	1.0	Kontinuierlich
WindingGap	REAL	0.0	n.def.	0.0	Kontinuierlich
PosCoilEdgeA	REAL	0.0	n.def.	0.0	Kontinuierlich
PosCoilEdgeB	REAL	> PosCoilEdgeA	n.def.	100	Kontinuierlich
AngleCoilEdgeA	REAL	>0.0	< 90.0	0.0	Kontinuierlich
AngleCoilEdgeB	REAL	>0.0	< 90.0	0.0	Kontinuierlich
ReversingPauseAngle	REAL	0.0	n.def.	60	Kontinuierlich
TraverseAccel	REAL	>0.0	n.def.	1000	Steigende Flanke an Enable
TraverseDecel	REAL	>0.0	n.def.	1000	Steigende Flanke an Enable
DistTraverseCoil	REAL	>0.0	n.def.	100	Steigende Flanke an Enable
CoilCoreDiameter	REAL	>0.0	n.def.	100	Steigende Flanke an Enable
CoilStartDiameter	REAL	>= CoilCoreDiameter	n.def.	100	Steigende Flanke an Enable
ConfigUnwindTraverse	STRUCT				Kontinuierlich

Tab. 8-59: Min-/Max und Default-Werte der Eingänge MB_TraverseControlType01

Signal-Zeit-Diagramm

Signal-Zeitdiagramm

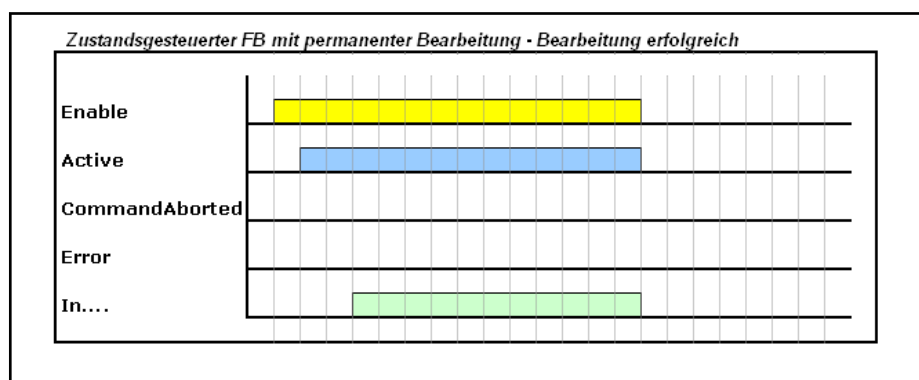


Abb. 8-48: Signal-Zeitdiagramm für MB_TraverseControlType01

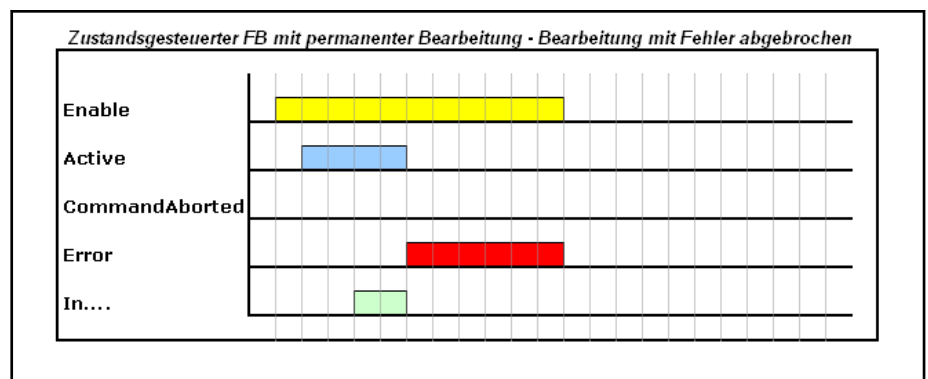


Abb. 8-49: Signal-Zeitdiagramm für MB_TraverseControlType01 bei Fehler

Orientierungshilfen:[Funktionsbeschreibung, Seite 597](#)[Aufwickelfunktionalität, Seite 598](#)[Abwickelfunktionalität, Seite 601](#)[Wechsel zwischen Auf- und Abwickeln, Seite 606](#)[Prinzipielle Anwendungsfälle, Seite 607](#)**Funktionsbeschreibung**

Der Baustein MB_TraverseControl erlaubt, zusammen mit einem Wicklerbaustein, das Auf- und Abwickeln von Drähten, Fäden etc. auf Spulen. Er wird immer mit einer translatorischen Verlegerachse verwendet.

Ausgangsgrößen des Bausteins sind eine Zielposition und eine Positioniergeschwindigkeit für einen Positionierbaustein MC_MoveAbsolute bzw. "Position" und "Velocity" der Betriebsart "ModePosAbs" des Achsinterfaces für eine translatorische Achse.

Kommandierung der Verlegerachse

Für die Verlegerachse muss die richtige Betriebsart des Antriebes sowie die Reglerfreigabe des Antriebes gesetzt werden.

Dies kann auf zwei unterschiedliche Arten erfolgen:

- Setzen der Betriebsart "ModePosAbs" mit Hilfe des Achsinterfaces (dies wird empfohlen)
- Benutzung der PLC-Open Funktionsbausteine "MC_Power", "MC_MoveAbsolute", "MC_Stop" zur Kommandierung der Achse

Des Weiteren muss auch die entsprechende Wickelachse kommandiert werden.

Es empfiehlt sich folgende Ein- und Ausschaltrihenfolge für Verlegerachse und Wickelachse:

Einschaltrihenfolge:

1. Einschalten Betriebsart Wicklerachse
2. Einschalten Betriebsart Verlegerachse
3. Evtl. Positionieren des Verlegers
4. Einschalten Funktionsbaustein Verleger
5. Einschalten Funktionsbaustein Wickler
6. Einschalten Betriebsart Leitachse

Ausschaltrihenfolge:

1. Ausschalten Betriebsart Leitachse

2. Ausschalten Funktionsbaustein Verleger
3. Ausschalten Betriebsart Verleger
4. Ausschalten Funktionsbaustein Wickler
5. Ausschalten Betriebsart Wickler

Verschalten mit dem Kommandierungsbaustein MC_MoveAbsolute bzw. Achsinterface

Die Werte für Acceleration und Deceleration des Kommandierungsbaustein bzw. Betriebsart des Achsinterfaces müssen auch an den Eingängen des Verlegerbausteins "TraverseAccel" und "TraverseDecel" angeschlossen werden.

Die Ausgänge des Verlegerbausteins "TraverseEndpos" und "TraverseVelocity" müssen an die Eingänge des Kommandierungsbausteins bzw. Achsinterface "Position" und "Velocity" angeschlossen werden.

Unterschiede Aufwickeln/Abwickeln

Generell wird für die Verlegerachse zwischen Aufwickeln und Abwickeln unterschieden. Die Verlegerachse ist translatorisch gewichtet.

- Beim Aufwickeln ist die Geschwindigkeit der Verlegerachse an die Geschwindigkeit des Wicklers (einer Wicklerachse) gekoppelt und sorgt so für eine gleichbleibende eng aneinander liegende Verlegung des Materials. Es wird alternierend von Spulenrand A zu Spulenrand B.
- Beim Abwickeln sind mehrere Verfahren verfügbar, die abhängig von den gegebenen Voraussetzungen (z. B. vorhandene Hardware/Kraftmessdosen) ausgewählt werden können.

Die Unterscheidung zwischen Auf- und Abwickeln erfolgt mit dem Eingang "WindDirection".

Aufwickelfunktionalität

Beim Aufwickeln fährt der Verleger alternierend von Spulenrand A zu Spulenrand B. Es wird mit jeder Windung automatisch um die Materialstärke und Lücke weiter verfahren.

Geschwindigkeit ("TraverseVelocity"):

Die Verlegergeschwindigkeit ist an die Drehzahl des Wicklers gekoppelt. Er verfährt pro Wicklerumdrehung um einen Wickelschritt, der sich aus Materialstärke und Spalt zwischen den Windungen ergibt.

Startrichtung ("TraverseStartDirection"):

Beim Starten des Wickelvorgangs ist es nötig, dass der Anwender die Richtung, in die der Verleger losfahren soll vorgibt, also bei "TraverseStartDirection" die Richtung (z. B. A_TO_B, also von Spulenrand A zu Spulenrand B) einträgt.

Die aktuelle Richtung (z. B. A_TO_B) und Position wird ausgegeben.

Spulenränder:

Die Spulenränder sind als absolute Positionen vorzugeben, dabei muss "PosCoilEdgeA" einen kleineren Wert haben als "PosCoilEdgeB". Bei Spulen mit schrägen Rändern sind die Endpunkte am Spulenkörper anzugeben.

Sind die Spulenränder nicht gerade, sondern um einen Winkel gegen die Senkrechte geneigt, muss dieser Winkel in "AngleCoilEdgeA" bzw. "AngleCoilEdgeB" angegeben werden. Ein positiver Winkel entspricht einer breiter werdenden Spule. Die Randposition und damit Zielposition des Verlegers (dies entspricht der Zielposition des zugehörigen Kommandierungsbausteins "MC_MoveAbsolute" bzw. der Betriebsart "ModePosAbs" des TechInterfaces) wird dann automatisch, je Wickellage, angepasst.

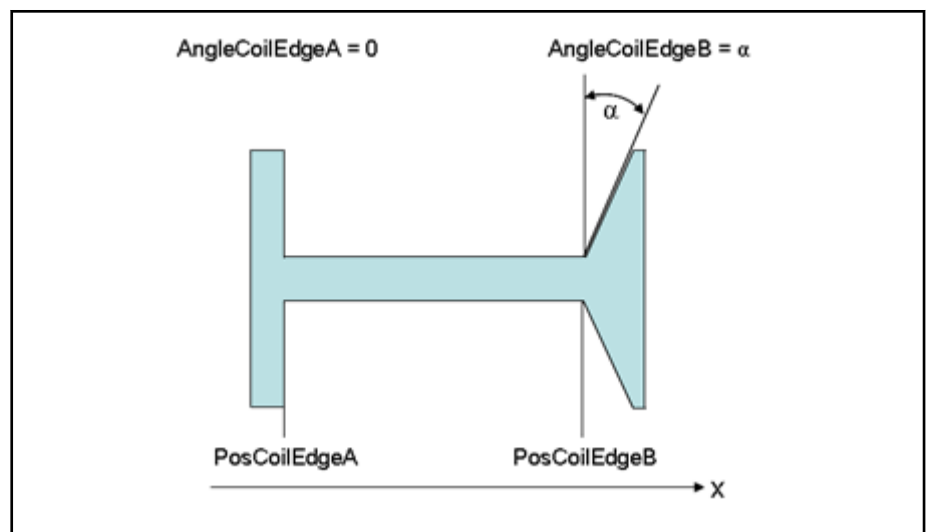


Abb. 8-50: Spulenränder

Vorlauf ("StrokeAdaption"):

Durch den Abstand zwischen Verleger und Spule folgt, dass der Verleger gegenüber der Position an der der Draht auf die Spule gewickelt wird, etwas voreilen muss. Dieser Vorlauf ist umso größer, je größer der Wickelschritt (Materialbreite plus Spalt) und der Abstand von Verleger zu Spule sind. Außerdem geht der aktuelle (veränderliche) Spulendurchmesser ein.

Um sicherzustellen, dass der Draht auch das Spulenende erreicht, kann es nötig sein über den Spulenrand hinauszufahren. Dies kann mit "StrokeAdaption" ein- und ausgeschaltet werden.

Ist "StrokeAdaption" eingeschaltet, fährt der Verleger also um den aktuellen Vorlauf ("ActStroke") über den eigentlichen Spulenrand hinaus, d. h. die Endposition berechnet sich aus SpulenrandA – Vorlauf bzw. SpulenrandB + Vorlauf. Dabei ist zu beachten, dass bei geraden Spulenrändern der Draht oben am Rand schleifen kann.

Ist "StrokeAdaption" ausgeschaltet, kehrt der Verleger am Spulenrand um.

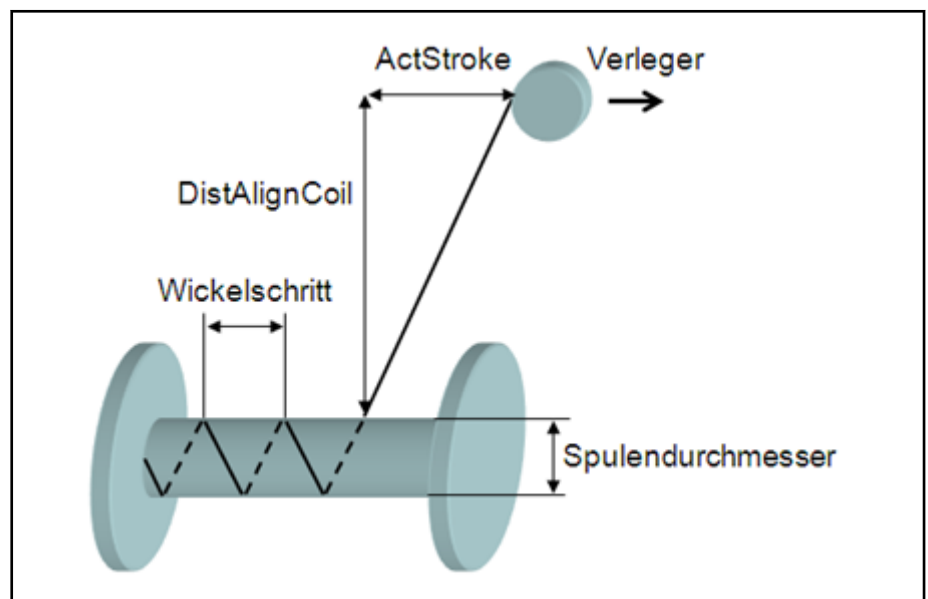


Abb. 8-51: Wickelvorgang

Lagenzähler ("LayerCounter"):

Die Anzahl der gewickelten Lagen wird gezählt und ausgegeben.

Soll eine bereits gefüllte Spule weiter gewickelt werden, muss der Anfangswert für den Lagenzähler ermittelt werden. Dies kann auf zwei Arten geschehen, die mit dem Modus Lagenzähler ("ModeLayerCounter") ausgewählt werden können.

Im ersten Fall gibt der Anwender die Anzahl der Lagen ("StartLayers") vor. Der Eingang "CoilStartDiameter" wird nicht ausgewertet, sondern aus der Lagenanzahl berechnet.

Im zweiten Fall wird die Anzahl der Lagen aus dem Startdurchmesser der Spule ("CoilStartDiameter") berechnet. Der Eingang "StartLayers" wird nicht ausgewertet.

Wartewinkel ("ReversingPauseAngle"):

Erreicht der Verleger seine Zielposition (Spulenrand), steht er dort jeweils so lange, bis der Wickler um den Wartewinkel weitergedreht hat. Das Warten ermöglicht dem Draht den Wechsel in die nächste (höhere) Lage.

Geschwindigkeitsadaption ("SpikeAdaption"):

Beim Beschleunigen und Verzögern am Umkehrpunkt (Spulenrand), also am Anfang und Ende jeder Lage (reversieren), wird der gewünschte Verfahrensweg (Wickelschritt) pro Wicklerumdrehung nicht ganz erreicht. Wird diese Abweichung zu groß, kann diese durch Geschwindigkeitsspitzen ausgeglichen werden, die mittels "SpikeAdaption" zuschaltbar sind. Ist dies eingeschaltet, wird während der ersten und letzten Wicklerumdrehung einer Lage die Geschwindigkeit des Verlegers so erhöht, dass der Verlust ausgeglichen wird. Dies kann z. B. bei dicken Drähten, die auf Spulen mit relativ kleinem Durchmesser gewickelt werden sollen, erforderlich sein.

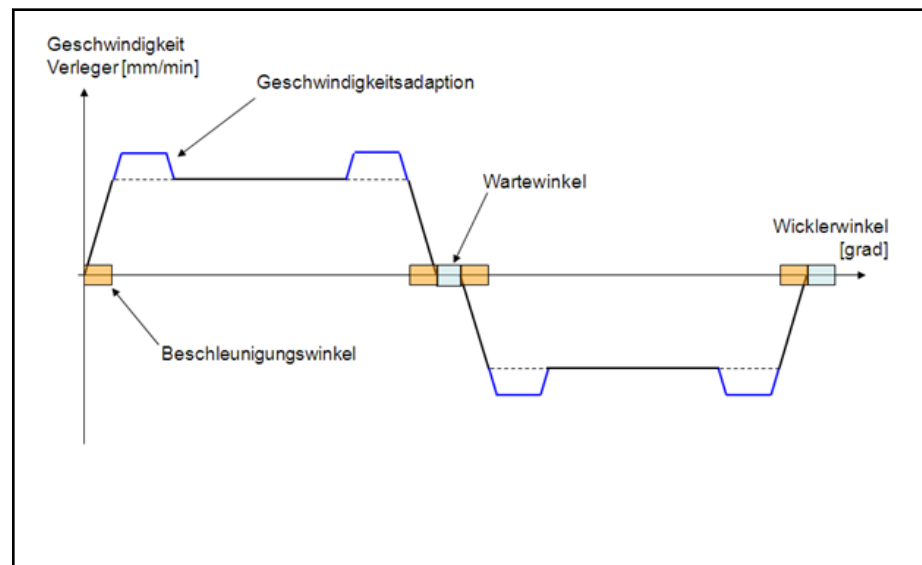


Abb. 8-52: Geschwindigkeitsverlauf mit SpikeAdaption

Versatzwinkel ("DisplacementAngle"):

Die Drähte einer Lage liegen, über den größten Teil des Spulenumfangs, in den Rillen, zwischen zwei Drähten, der unteren Lage. Allerdings ergeben sich zwangsläufig in jeder Lage Drahtkreuzungen, in deren Bereich die Höhe der Wicklung zunimmt. Um eine Unwucht der Spule zu vermeiden, sollten diese Drahtkreuzungen sich gleichmäßig über den Spulenumfang verteilen. Als Kontrolle hierzu wird der Versatzwinkel ausgegeben. Er gibt die Winkeldifferenz (Drehwinkel der Wickelachse) vom Beginn einer Lage des einen Spulenendes zum Beginn der nächsten Lage des anderen Spulenendes an.

Er sollte also möglichst kein Teiler von 180° sein. Der Versatzwinkel kann mit dem Spalt zwischen den Windungen und dem Wartewinkel beeinflusst werden.

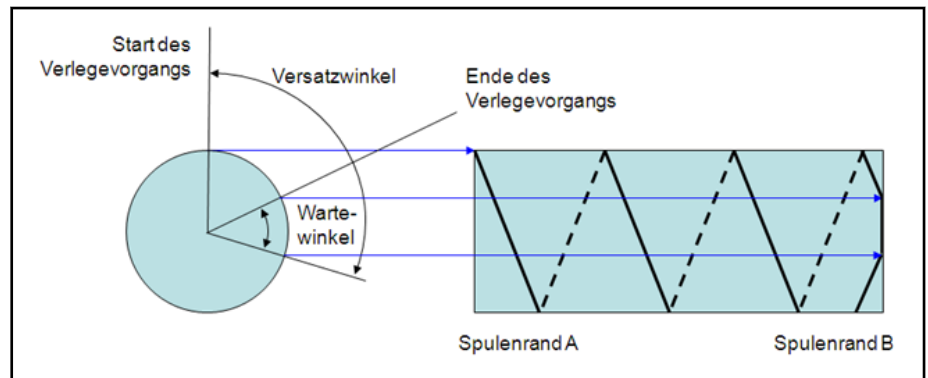


Abb. 8-53: Versatzwinkel

Abwickelfunktionalität

Beim Abwickeln sind vier verschiedene Methoden möglich. Sie werden mit dem Eingang "ConfigUnwindTraverse.Mode" ausgewählt.

Ist die Verlegung des Drahts auf der Spule unbekannt, können somit auch Sprünge oder Richtungswechsel enthalten sein. Hierfür sind die Methoden "WIRE_TENSION_TRIM", "WIRE_TENSION_TRIM_DELTA" oder "WINDING_STEP_TENSION_TRIM" zu empfehlen, da die Geschwindigkeitsregelung solche Effekte ausgleicht.

1. "MIDDLE_FIXED" Verlegerachse steht immer fest in Mittelstellung
2. "WINDING_STEP" Verlegerachse wird immer um Materialstärke verfahren (wie beim Aufwickeln)
3. "WIRE_TENSION_TRIM" Verlegerachse wird in Abhängigkeit der Differenz von Verlegerzugistwert und Zugistwert außerhalb des Verlegers gewichtet mit einem PID-Regler verfahren (Es werden 2 Kraftmessdosen benötigt)
4. "WINDING_STEP_TENSION_TRIM" Kombination von Methode 2 und 3, Verfahren wird um Materialstärke und Zugspannungsdifferenz (Es werden 2 Kraftmessdosen benötigt)

Wenn keine Messeinrichtungen vorhanden sind, können die Methoden "MIDDLE_FIXED" und "WINDING_STEP" verwendet werden. "MIDDLE_FIXED" als einfachste Möglichkeit, "WINDING_STEP" wenn die abzuwickelnde Spule regelmäßig aufgewickelt wurde.

Ist eine automatische Anpassung der Verlegerposition gewünscht, und eine regelmäßig aufgewickelte Spule nicht vorausgesetzt werden kann, müssen die Methoden "WIRE_TENSION_TRIM" oder "WINDING_STEP_TENSION_TRIM" verwendet werden. Dort wird die Zugspannung im Bereich des Verlegers mit der Bahnspannung außerhalb des Verlegers verglichen. Diese Bahnspannungsdifferenz wird über einen einstellbaren PID-Regler gewichtet und mit **Wert** und **Vorzeichen** auf die Ausgangsgeschwindigkeit des Verlegers "TraverseVelocity" gegeben.

Kalibrieren der Messdosen:

Die Kraftmessdosen müssen so kalibriert werden, dass beide bei einem senkrechten Verlauf des Drahts vom Verleger zur Spule den gleichen Wert anzeigen. Auch bei der Messdose außerhalb des Verlegers muss der Draht unbedingt wie im Betriebsfall um die Rolle geführt werden.

Kraftmessung

Der Draht bzw. das Material wird über eine Rolle geführt, in der die Zugkraft gemessen wird. Die so gemessene Kraft ist auch abhängig, wie weit die Rolle vom Draht bzw. Material umschlungen ist, d. h. wie groß der Umschlingungswinkel ist.

Die Beträge der Kräfte F_D und F_{D1} bzw. F_{D2} sind gleich groß und stellen die Zugkraft des Drahtes dar.

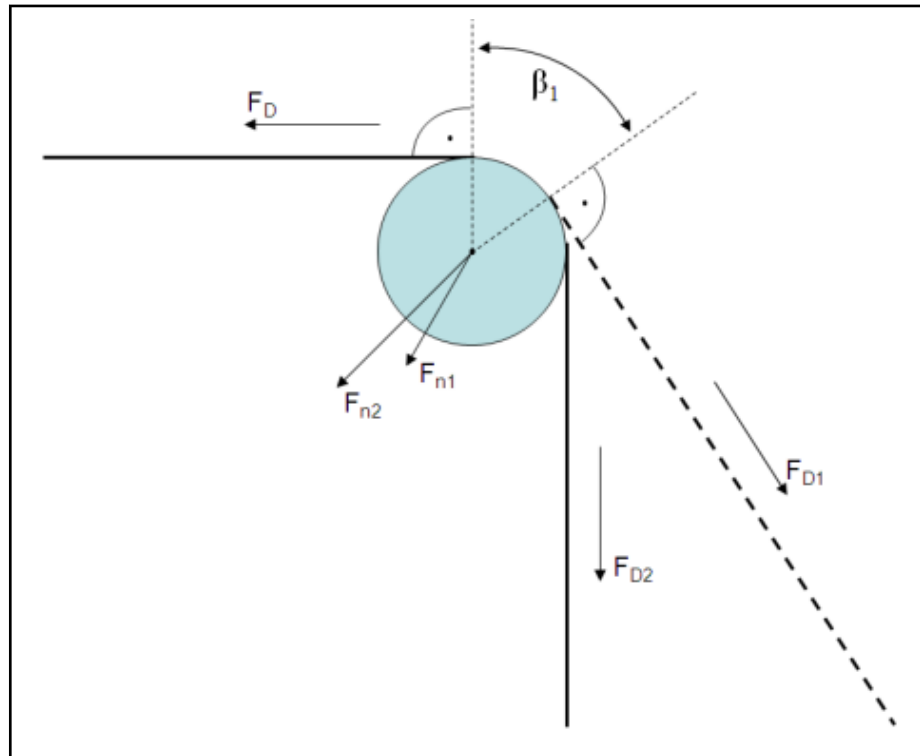


Abb. 8-54: Kräfte und Umschlingungswinkel

Die durch die Messdose gemessene Kraft F_{n1} (F_{n2} für β_2) ist abhängig vom Umschlingungswinkel β_1 . Der Umschlingungswinkel β kennzeichnet den Bereich der Rolle in dem der Draht bzw. das Material an der Rolle aufliegt. Anders ausgedrückt, den Winkel, den die beiden Senkrechten zum Draht bzw. Material durch den Rollenmittelpunkt miteinander einschließen.

Für die gemessene Zugskraft gilt:

$$F_n = 2 * \sin(\beta / 2) * F_{Mz}$$

Abb. 8-55: Zusammenhang Zugskraft und Umschlingungswinkel

Drahtrichtung ("ConfigUnwindTraverse.WireFeedDirection"):

Um eine richtige Wichtung der Geschwindigkeit sicherzustellen ist es nötig den Eingang "ConfigUnwindTraverse.WireFeedDirection" anzugeben. Dieser Eingang legt fest, aus welcher Richtung der Draht um den Verleger geführt wird, läuft z. B. der Draht aus Richtung des Spulenrands A zum Verleger und dann nach unten zur Spule, dann ist "A_TO_B" einzutragen.

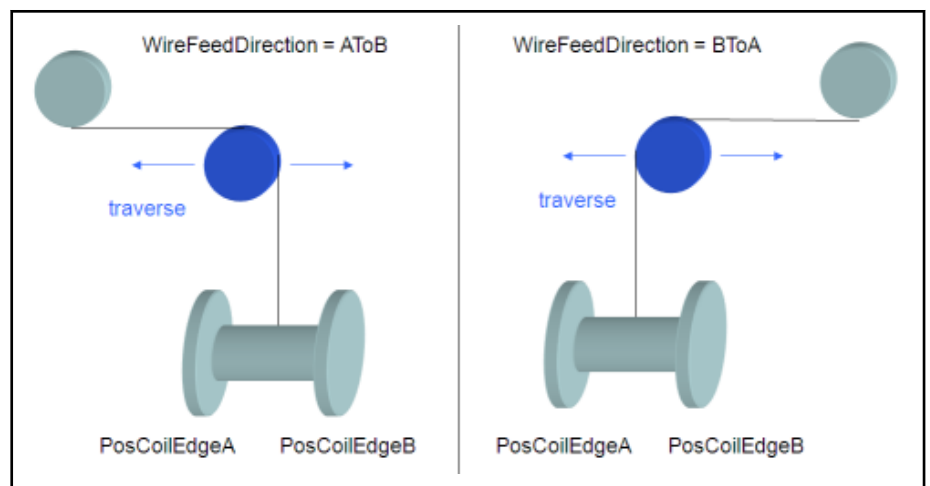


Abb. 8-56: WireFeedDirection

Lagenzähler ("LayerCounter"):

Der Lagenzähler wird fortgeführt, allerdings kann ein Lagenende nicht genau ermittelt werden. Stattdessen werden die theoretisch möglichen Wicklungen einer Lage berechnet und die Wickler-Umdrehungen gezählt. Sind die berechneten Wicklungen erreicht, wird eine Lage als beendet angenommen und der Lagenzähler dekrementiert.

Es muss der Anfangswert für den Lagenzähler ermittelt werden. Dies kann auf zwei Arten geschehen, die mit dem Modus Lagenzähler ("ModeLayerCounter") ausgewählt werden können.

Im ersten Fall gibt der Anwender die Anzahl der Lagen ("StartLayers") vor. Der Eingang "CoilStartDiameter" wird nicht ausgewertet, sondern aus der Lagenanzahl berechnet.

Im zweiten Fall wird die Anzahl der Lagen aus dem Startdurchmesser der Spule ("CoilStartDiameter") berechnet. Der Eingang "StartLayers" wird nicht ausgewertet.

Modus "MIDDLE_FIXED":

Bei dieser Abwickelmethode steht der Verleger fest in der Mitte der Spule. Die Position wird automatisch aus den Spulenrandpositionen berechnet und mit der mit „ConfigUnwindTraverse.Traverse VelMiddleFixed“ angegebenen Geschwindigkeit angefahren. Ist die Mittelstellung erreicht, bleibt der Verleger stehen.

Anwendung:

Dieser Modus ist nur bei relativ großem Abstand zwischen Verleger und Spule, relativ schmalen Spulen und unempfindlichen Drähten zu empfehlen, da der Draht durch seine Schrägstellung an den Spulenenden über die unteren Lagen scheuert und in Schwingung gerät.

Von diesem Modus kann nicht direkt in den Aufwickelmodus umgeschaltet werden, da die Verlegerposition verloren geht.

Modus "WINDING_STEP":

Bei dieser Abwickelmethode fährt der Wickler wie im Aufwickelmodus, mit allen seinen Eigenschaften. D. h. die Verlegerachse wird nach jeder Windung um Materialstärke und Lücke weiter bewegt.

Anwendung:

Diese Methode ist dann zu empfehlen, wenn der Draht vorher aufgewickelt wurde, also keine Sprünge oder Richtungswechsel zu erwarten sind, bzw. zu erwarten ist, dass die abzuwickelnde Spule regelmäßig aufgewickelt wurde.

Diese Methode ist mit geringstem Aufwand realisierbar.

Modus "WIRE_TENSION_TRIM":

Im Modus "WIRE_TENSION_TRIM" wird die Geschwindigkeit des Verlegers so geregelt, dass der Draht immer senkrecht vom Verleger zur Spule verläuft.

Kräftemessung:

Es sind zwei Kräfte nötig ("ConfigUnwindTraverse.ActWireTension" und "ConfigUnwindTraverse.ActWireTensionTraverse"). Diese werden jeweils über ein PT1-Glied gefiltert, deren Filterzeitkonstanten über "ConfigUnwindTraverse.FilterWireTension" bzw. "ConfigUnwindTraverse.FilterWireTensionTraverse" einstellbar sind (z. B. 20ms).

Geschwindigkeitsregelung:

Die beiden Kräfte auf den Regler geben, wobei die Kraft außerhalb des Verlegers der Soll- und die Kraft innerhalb des Verlegers der Istwert ist. Da die beiden Kräfte bei einer senkrechten Position des Verlegers über der Spule gleich sind (vergl. "Kalibrieren der Messdosen") wird die Geschwindigkeit des Verlegers so geregelt, dass er immer möglichst senkrecht steht.

Der Reglerausgang ist in beiden Fällen die Sollgeschwindigkeit des Verlegers (dies entspricht der Sollgeschwindigkeit des zugehörigen Kommandierungsbausteins MC_MoveAbsolute bzw. der Betriebsart "ModePosAbs" des TechInterface).

Die Regelparameter sind dabei über "ConfigUnwindTraverse.Control" einstellbar. Der P-Anteil verstärkt die anstehende Regeldifferenz. Er ist so zu wählen, dass der Verleger schnell auf Sprünge in der Drahtposition reagieren kann. Der I-Anteil sorgt für eine vollständige Ausregelung der Regeldifferenz. Der D-Anteil reagiert auf eine Änderung in der Steigung der Regeldifferenz.

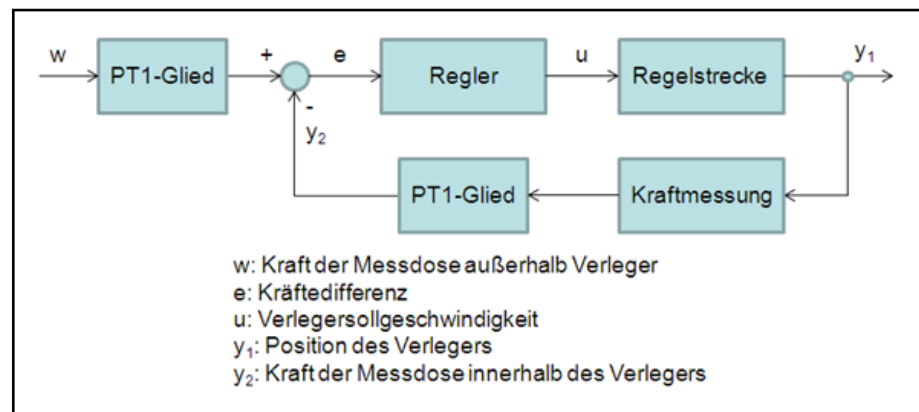


Abb. 8-57: Geschwindigkeitsregelung mit Kräftedifferenz

Im ersten Schritt kann erstmal nur ein P-Anteil verwendet werden.

Dieser sollte so hoch eingestellt werden, dass der Verleger im Stillstand nicht ins Schwingen kommt. Da die Regeldifferenz im Vergleich klein ist sollte ein möglichst großer Wert (z. B. 1000) gewählt werden, um eine möglichst schnelle Reaktion des Verlegers zu ermöglichen.

$$(K_p = 0.5 * k_{p\text{kritisch}})$$

Zielposition ("TraverseEndPos"):

Die Zielposition des Verlegers (dies entspricht der Zielposition des zugehörigen Kommandierungsbausteins MC_MoveAbsolute bzw. der Betriebsart "Mo-

dePosAbs" des TechInterfaces) ist der jeweilige Spulenrand (muss nicht unbedingt erreicht werden). Abhängig von der ermittelten Sollgeschwindigkeit werden die Richtung und damit die Zielposition (Spulenrand) gewechselt. D. h. bei einem Vorzeichenwechsel wird die "TraverseEndPos" gewechselt.

Modus "WINDING_STEP_TENSION_TRIM":

Dieser Abwickelmodus ist eine Kombination aus "WINDING_STEP" und "WI-RE_TENSION_TRIM". Der Verleger wird also je Wicklerumdrehung einen Wikkelschritt weitergefahren. Überlagert ist eine Geschwindigkeitsregelung, die Unregelmäßigkeiten in der vorgegebenen Wicklung ausgleichen soll.

Geschwindigkeitsregelung:

Die "Basisgeschwindigkeit" des Verlegers errechnet sich aus der Wicklerdrehzahl und dem Wikkelschritt (Materialbreite plus Spalt). Hier ist keine Beschleunigungskompensation durch Geschwindigkeitsspitzen vorgesehen.

Überlagert ist eine Geschwindigkeitsregelung wie bei der Einstellung "Wire-TensionTrim". Die Regelparameter sind dabei über "ConfigUnwindTraverse.Control" einstellbar. Der P-Wert sollte so gewählt werden, dass die überlagerte Geschwindigkeit bei kleinen Abweichungen von der Senkrechstellung die "Basisgeschwindigkeit" nicht überschreitet, sich aber bei größeren Abweichungen deutlich auswirkt. Außerdem müssen die Regelparameter, insbesondere der P-Wert, durch den Anwender während der Laufzeit an die Wicklergeschwindigkeit angepasst werden. Da sich bei steigender Wicklerdrehzahl auch die "Basisgeschwindigkeit" des Verlegers erhöht und sich somit das Verhältnis von "Basisgeschwindigkeit" und überlagelter geregelter Geschwindigkeit verschiebt.

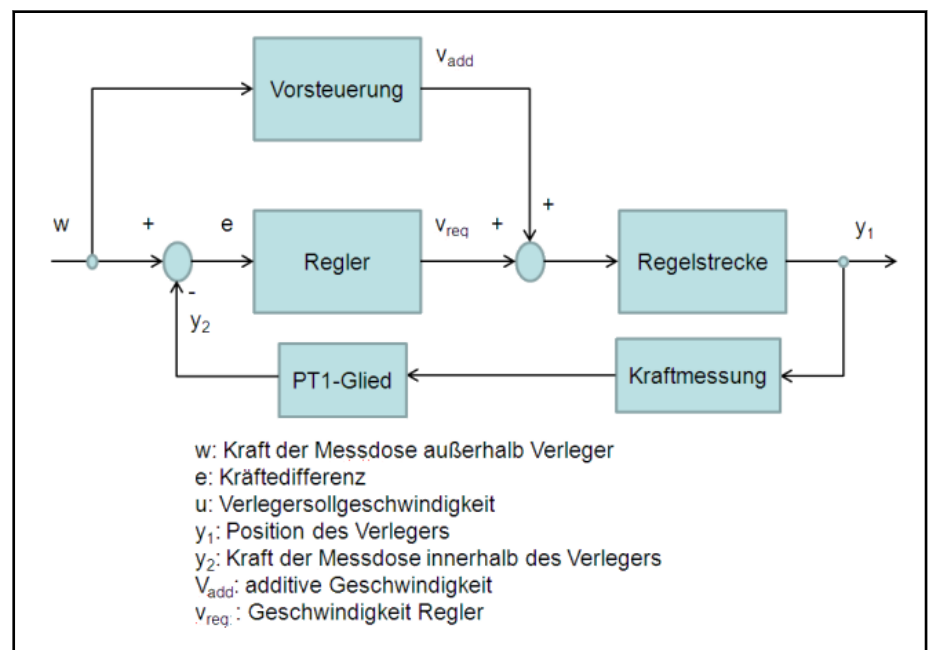


Abb. 8-58: Geschwindigkeitsregelung mit Kräftedifferenz und Vorsteuerung

Die Abwickelmethode "WindingStepTensionTrim" ist eine Kombination aus einer Steuerung und einer überlagerten Regelung der Verlegergeschwindigkeit. Die beiden ermittelten Geschwindigkeiten werden aufaddiert und die Summe am Ausgang des Verlegerbausteins ausgegeben. Der Regler muss deshalb so ausgelegt werden, dass er die Vorsteuerung nicht überschreitet, aber sich trotzdem auswirken kann. Es kann im ersten Schritt wieder ein P-Regler verwendet werden. Eine mögliche Vorgehensweise zur Ermittlung des P-Anteils:

Es wird die gewünschte Materialgeschwindigkeit " v_{Mat} " vorgegeben. Aus der Materialgeschwindigkeit kann die Verlegergeschwindigkeit " v_{traverse} " bei einer gleichmäßigen Verlegung des Materials mit folgenden Formeln ermittelt werden:

$$n_{\text{Wind}} = \frac{v_{\text{Mat}}}{d_{\text{coil}} * \pi} \quad \text{und} \quad v_{\text{traverse}} = \text{ws} * v_{\text{Wind}}$$

Beispielsweise ergibt sich für eine Materialgeschwindigkeit " v_{Mat} " = 1 m/s, einem Spulendurchmesser " d_{coil} " = $80 * 10^{-3}$ m und einem Wickschritt (Materialbreite + Spalt) ws = 0,13 mm eine Verlegergeschwindigkeit " v_{traverse} " = 31 mm/min.

Mit der Formel [Abb. 8-55 "Zusammenhang Zugistkraft und Umschlingungswinkel" auf Seite 602](#) kann eine Kraftdifferenz in Abhängigkeit des Umschlingungswinkels berechnet werden. Wird beispielsweise der Umschlingungswinkel um fünf Grad (gegen 90°) verändert, ergibt sich eine Kraft- und damit Regeldifferenz von ungefähr 0,6 (Kraft bei 90° minus Kraft bei 85°).

Soll der maximal auftretende Auslenkungswinkel gleich diesen fünf Grad sein, muss die proportionale Reglerverstärkung so eingestellt werden, dass in diesem Fall die Verlegergeschwindigkeit zu Null wird. Es muss also die Regelgröße und die durch die Vorsteuerung ermittelte Geschwindigkeit gleich gesetzt werden. Daraus folgt, dass die Verlegergeschwindigkeit v_{Verl} durch die Regeldifferenz geteilt werden muss, was einen P-Anteil von ca. 51,7 ergibt.

Diese Berechnung ist allerdings von der veränderlichen Wicklerdrehzahl abhängig. Auch die Wahl von fünf Grad als Grenzwert für die Veränderung des Umschlingungswinkels ist von der Anwendung abhängig und hier nur beispielhaft.

Durch die Vorsteuerung ist der Geschwindigkeitsverlauf glatter als bei „Wire-TensionTrim“, allerdings ist durch den deutlich kleineren P-Wert und die daraus resultierende kleinere Verlegergeschwindigkeit die Reaktion auf Sprünge etc. in der Drahtposition etwas langsamer.

Zielposition ("TraverseEndPos"):

Die Zielposition des Verlegers (Dies entspricht der Zielposition des zugehörigen Kommandierungsbausteins MC_MoveAbsolute bzw. der Betriebsart "ModePosAbs" des TechInterface) ist der jeweilige Spulenrand.

Die Richtung und damit die Zielposition des Verlegers werden gewechselt, wenn entweder ein Spulenrand erreicht und der Wartewinkel abgewartet wurde, oder wenn die überlagerte Geschwindigkeit die "Basisgeschwindigkeit" übersteigt und die Summe negativ wird.

Anwendung:

Diese Methode eignet sich durch die implizierte Vorsteuerung besonders für das Abwickeln von Spulen, die einen regelmäßig verlegten Draht/Material enthalten. Sie hat aber gegenüber dem gesteuerten Abwickeln mit Wickschritt den Vorteil, dass die beim Aufwickeln verwendeten Einstellungen (z. B. Lücke zwischen den Windungen) nicht genau bekannt sein müssen. Zudem kann auch auf Unregelmäßigkeiten in der Verlegung reagiert werden.

Wechsel zwischen Auf- und Abwickeln

Es ist möglich zwischen den unterschiedlichen Modi während des Betriebs des Bausteins zu wechseln.

Beim Wechsel vom Auf- in den Abwickelmodus, oder andersherum, wird die Wickelrichtung umgekehrt (z. B. "A_TO_B" wird zu "B_TO_A").

Vom Abwickelmodus "MiddleFixed" sollte nicht in den Aufwickelmodus gewechselt werden, da keine Startposition für den Verleger gegeben ist. Hierzu sollte der Baustein ausgeschaltet, der Verleger positioniert und anschließend der Baustein wieder eingeschaltet werden.

Prinzipielle Anwendungsfälle *Es sind grundsätzlich zwei Anwendungsfälle zu unterscheiden:*

- Im ersten Fall wird das Wickelgut (z. B. Draht) von einer Spule ab- bzw. aufgewickelt. Dabei wird die Spule komplett ab- und wieder aufgewickelt. Anschließend wird eine neue Spule eingelegt. In diesem Fall werden der Verleger- und auch der Wickelbaustein zum Spulenwechsel gestoppt und anschließend wieder neu initialisiert und gestartet. (Anwendungsbeispiel: Drahtspulenherstellung)
- Beim zweiten Anwendungsfall wird die Spule nicht unbedingt komplett ab- und wieder aufgewickelt, sondern ein Teil ab- und sofort wieder aufgewickelt (oder umgekehrt). In diesem Fall erfolgt also ein direkter Wechsel vom Ab- in den Aufwickelmodus (oder umgekehrt), ohne dass der Baustein gestoppt wird.

(Anwendungsbeispiel für den zweiten Anwendungsfall: Ingotsäge)

Fehlerbehandlung In der folgenden Tabelle sind die Fehlernummern des Funktionsbausteins MB_TraverseControlType01 aufgelistet und beschrieben.

Es werden die Fehlercodes aus der F_RELATED_TABLE (16#0170) verwendet.

Zusätzlich können die Fehlercodes der Funktionsbausteine MB_DancerControlType03, MB_DiameterCalculatorType03, IL_PIDType01, RIL_LoopControl sowie IL_PT1Type01, RIL_LoopControl, auftreten.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#0006	16#FFFF	Ungültiger Zustand der Zustandsmaschine
INPUT_RANGE_ERROR	16#1111	16#0001	WindDirection falsch angegeben
INPUT_RANGE_ERROR	16#1111	16#0002	SpikeAdaption falsch angegeben
INPUT_RANGE_ERROR	16#1111	16#0003	StartDirection falsch angegeben
INPUT_RANGE_ERROR	16#1111	16#0004	ModeLayerCounter falsch angegeben
INPUT_RANGE_ERROR	16#1111	16#0005	StartLayers < 0
INPUT_RANGE_ERROR	16#1111	16#0006	WireDiameter <= 0.0
INPUT_RANGE_ERROR	16#1111	16#0007	WindingGap < 0.0
INPUT_RANGE_ERROR	16#1111	16#0008	PosCoilEdgeA < 0.0
INPUT_RANGE_ERROR	16#1111	16#0009	PosCoilEdgeB < PosCoilEdgeA
INPUT_RANGE_ERROR	16#1111	16#000A	0° >= AngleCoilEdgeA >= 90°
INPUT_RANGE_ERROR	16#1111	16#000B	0° >= AngleCoilEdgeB >= 90°
INPUT_RANGE_ERROR	16#1111	16#000C	ReversingPauseAngle < 0.0
INPUT_RANGE_ERROR	16#1111	16#000D	TraverseAcc <= 0.0
INPUT_RANGE_ERROR	16#1111	16#000E	TraverseDec <= 0.0
INPUT_RANGE_ERROR	16#1111	16#000F	DistTraverseCoil <= 0.0

ErrorID	Additional1	Additional2	Text
INPUT_RANGE_ERROR	16#1111	16#0010	CoilCoreDiameter <= 0.0
INPUT_RANGE_ERROR	16#1111	16#0011	CoilStartDiameter < CoilCoreDiameter
INPUT_RANGE_ERROR	16#1111	16#0012	ConfigUnwindTraverse.Mode falsch angegeben
INPUT_RANGE_ERROR	16#1111	16#0013	ConfigUnwindTraverse.TraverseVelMiddleFixed < 0.0
INPUT_RANGE_ERROR	16#1111	16#0014	StrokeAdapton falsch angegeben
INPUT_RANGE_ERROR	16#1111	16#0015	PControl < 0.0
INPUT_RANGE_ERROR	16#1111	16#0016	ConfigUnwindTraverse.WireFeedDirection falsch angegeben
DEVICE_ERROR	16#1112	16#0001	Verlegerachse nicht in "BB"
DEVICE_ERROR	16#1112	16#0002	Verleger Achsnummer falsch
DEVICE_ERROR	16#1112	16#0003	Wickler Achsnummer falsch

Tab. 8-60: Fehlercodes FB MB_TraverseControlType01

Notwendige Hardware

- CMLx, XMx, VPx

- IndraDrive

Notwendige Firmware

- MLC Firmware FWA-CMLXX*-MLC-14VRS

XX - Gerätevariante, z. B. CML65

- IndraDrive Antriebsfirmware

Notwendige Software

- IndraWorks 14VRS für MLC

Besonderheiten für MLC Für die Verlegerachse muss translatorische Wichtung eingestellt werden.

8.8.7 MB_DancerCalcControlParamType01

Kurzbeschreibung

Der Funktionsbaustein berechnet aus physikalischen Daten des Tänzersystems (Durchmesser der Zugwalze bzw. Hülsendurchmesser beim Wickler sowie Verzögerungszeit des Systems) die Reglerparameter des Tänzerlagereglers. Die berechneten Reglerparameter für "PControl", "IControl" und "DControl" können für die Funktionsbausteine [MB_DancerControlType04](#), [Seite 501](#), und [MB_WinderDancerCtrlType02](#), [Seite 532](#), verwendet werden.

Diese Reglerparametrierung anhand von spezifischen Streckenparametern ist speziell dort zweckmäßig, wo der Inbetriebnehmer der Anlage keine regelungstechnischen Kenntnisse zur Reglereinstellung besitzt.

Zuweisung: Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS	RMB_TechWinder
Funktionspaket TEC	

Tab. 8-61: Referenztable des MB_DancerCalcControlParamType01

Schnittstellenbeschreibung

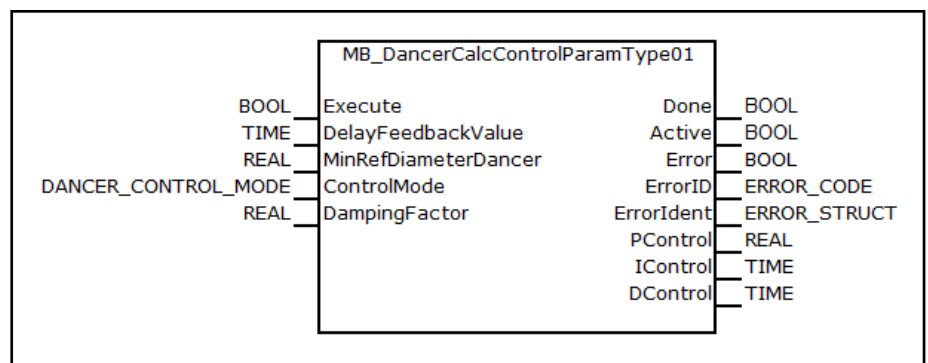


Abb. 8-59: Funktionsbaustein MB_DancerCalcControlParamType01

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	Startet den Funktionsbaustein (einmalig, flanken-gesteuert)
	DelayFeedbackValue	TIME	Regelkreisverzögerung (d. h. Summe aller vom Nut-zer hinzugefügten Filter- und Totzeiten, inklusive der Zykluszeit des Aufrufs des zugehörigen Wickler-Funk-tionsbausteins, z. B. MB_WinderDancerCtrlType02)
	MinRefDiameterDan- cer	REAL	Minimaler Bezugsdurchmesser (in mm) für Wickler mit Tänzer
	ControlMode	MB_DAN- CER_CONTROL_MO- DE	Auswahl des Reglertyps (P-, PI-, PID-Regler) MB_DANCER_CONTROL_MODE, Seite 620
	DampingFactor	REAL	Dieser Eingang beeinflusst das Regelverhalten (in %). Das Regelverhalten kann von langsam, aperiodisch/ kriechend bis zu schnell, überschwingend/oszillierend gewählt werden.
VAR_OUTPUT	Done	BOOL	Berechnung ohne Fehler beendet und Ausgangswerte sind gültig
	Active	BOOL	Berechnung läuft und Ausgangswerte sind ungültig
	Error	BOOL	Berechnung mit Fehler beendet (siehe ErrorID and Er-rorIdent)
	ErrorID	ERROR_CODE	Diagnosebeschreibung im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose Fehlerbehandlung, Seite 506
	PControl	REAL	Proportionalverstärkung
	IControl	TIME	Nachstellzeit
	DControl	TIME	Vorhaltzeit

Tab. 8-62: Schnittstellenvariablen des Funktionsbausteins MB_DancerCalc-ControlParamType01

Min./Max.-Werte und Default-Werte und die Übernahme der Eingänge

- Beschreibt die Min./Max.-Werte und Default-Werte und die Übernahme der Eingänge
- Boolean und Aufzählungseingänge verfügen weder über einen Minimal- noch einen Maximalwert (nur Default-Wert)

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Steigende Flanke
DelayFeedbackValue	TIME	T#10ms	T#1s	T#100ms	Steigende Flanke an "Enable"
ControlMode	MB_DANCER_CONTROL_MODE			PI_CONTROL	Steigende Flanke an "Enable"
DampingFactor	REAL	20.0	200.0	100.0	Steigende Flanke an "Enable"
MinRefDiameterDancer	REAL	> 0.0	N. def.	100.0	Steigende Flanke an "Enable"

Tab. 8-63: Min.-/Max.-Werte und Default-Werte der Eingänge des MB_DancerCalcControlParamType01

Funktionsbeschreibung

Dieser Funktionsbaustein berechnet die Reglerparameter für einen Tänzerlagereger durch die Verwendung der Bahngeometrie und einiger zusätzlicher Eingänge wie den Prozessverzögerungen.

Der Nutzer wählt zwischen drei verschiedenen Reglertypen:

- P-Regler
- PI-Regler
- PID-Regler

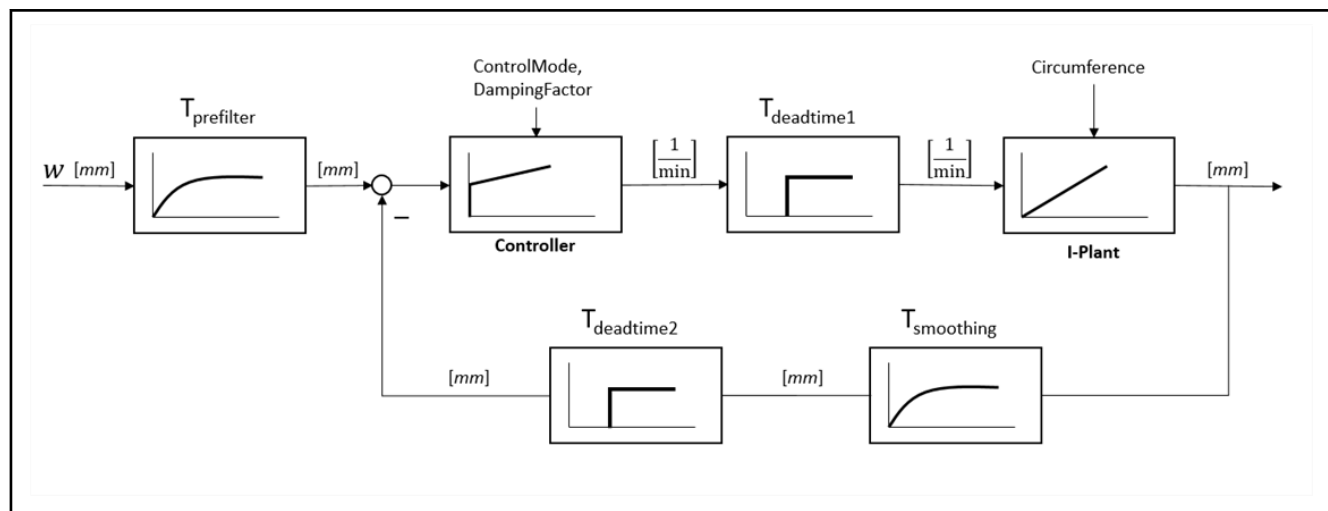


Abb. 8-60: Diagramm des MB_DancerCalcControlParamType01



Bei Verwendung dieses Funktionsbausteins müssen der Soll- und Istwert des Tänzerlageregers zwingend in der Einheit **mm** gewichtet sein!

DelayFeedbackValue:

Alle Übertragungsverzögerungen und Glättungszeiten werden vom Eingang DelayFeedbackValue berücksichtigt.

Entsprechend ergibt sich für das dargestellte Blockschaltbild eine gesamte Verzugszeit von $\text{DelayFeedbackValue} = T_{\text{deadtime1}} + T_{\text{smoothing}} + T_{\text{deadtime2}}$.

Der Wert DelayFeedbackValue kann auch mit einer Sprungantwort geschätzt werden. Führen Sie eine Sprungantwort durch und zeichnen Sie diese auf

(sprungförmige Aufschaltung eines additiven Drehzahlsollwertes auf der Wicklerachse und Messung der Tänzerlage $h(t)$ wie in der nachfolgenden Abbildung gezeigt). Messen Sie dann T_1 mittels der Cursorfunktion, um den Wert DelayFeedbackValue zu schätzen.

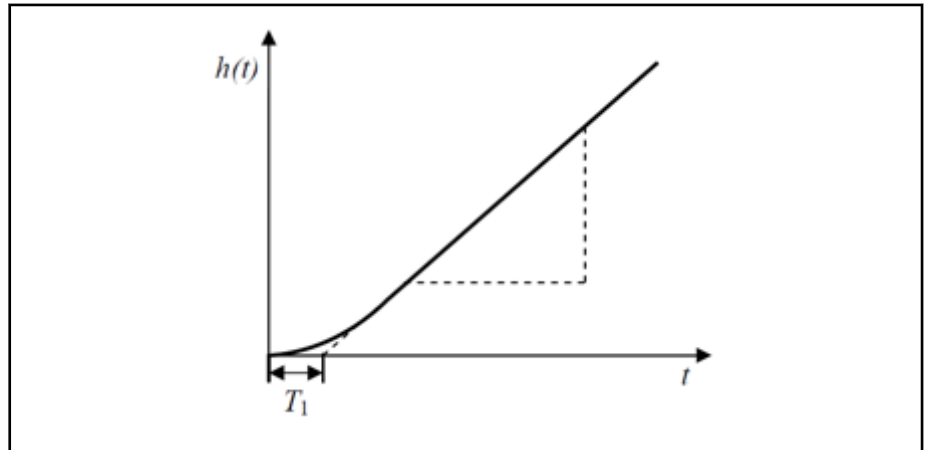


Abb. 8-61: Sprungantwort der Tänzerlage

MinRefDiameterDancer:

Der Eingang MinRefDiameterDancer bezieht sich auf den Eingang, der in der Datenstruktur [MB_WIND_DIAMETER](#), Seite 616 des Funktionsbausteins MB_WinderDancerCtrlType02 gleich benannt ist.

Es muss derselbe Wert wie bei den Funktionsbausteinen gesetzt werden (Durchmesser des Wickelkerns).

ControlMode:

Verschiedene Reglertypen (P-, PI- und PID-Regler) können mit Hilfe des Eingangs ControlMode, wie in [MB_DANCER_CONTROL_MODE](#), Seite 620 definiert, ausgewählt werden.

DampingFactor:

Dieser Eingang beeinflusst das Regelverhalten (in %). Das Regelverhalten kann von langsam, aperiodisch/kriechend bis zu schnell, überschwingend/oszillierend gewählt werden. Der richtige Wert hängt von der Dynamik des mechanischen Systems ab und kann mittels "trial and error" ermittelt werden, angefangen mit einem niedrigen Wert (z. B. 30 %).

Vorfilter:

Aufgrund ihrer Bestimmung halten Tänzerlagereger normalerweise eine feste Sollposition und regeln Störungen bestmöglich aus. Die berechnete Parametrierung führt im Allgemeinen zu einem Überschwingen der Tänzerlage, wenn sich der Tänzerlagesollwert des Tänzerlagereglers ändert, da die Reglerauslegung auf Störverhalten optimiert ist. Um dieses Verhalten zu vermeiden, kann vom Anwender im SPS-Programm ein zusätzliches Sollwertfilter (IL_PT1Type01) vor den Eingang "SetPointDancer" des Tänzerlagereglers geschaltet werden. Eine zweckmäßige Filterzeitkonstante für dieses Sollwertfilter liefert der Ausgangswert "IControl".

Fehlerbehandlung

Die nachfolgende Tabelle listet und beschreibt alle Fehlercodes des Funktionsbausteins MB_DancerCalcControlParamType01.

Die Fehlercodes aus F_RELATED_TABLE werden verwendet.

ErrorID	Additional1	Additional2	Text
INPUT_RANGE_ERROR	16#00000002	16#00000001	DampingFactor überschreitet den Grenzbereich
INPUT_RANGE_ERROR	16#00000002	16#00000002	DelayFeedbackValue überschreitet den Grenzbereich
INPUT_RANGE_ERROR	16#00000002	16#00000003	MinRefDiameterDancer ≤ 0.0

Tab. 8-64: Fehlercodes des Funktionsbausteins MB_DancerCalcControlParamType01

8.9 Datentypen

8.9.1 Einführung und Übersicht

In diesem Abschnitt sind die von den Funktionsbausteinen genutzten Datentypen zusammengefasst.

Bezeichnung	Typ	Beschreibung
MB_TRAVERSE_ADAPTION, Seite 612	ENUM	Adaption Ein/Aus
MB_TRAVERSE_MODE_LAYER_COUNTER, Seite 613	ENUM	Startwert für Lagenzähler
MB_TRAVERSE_UNWIND, Seite 613	STRUCT	Einstellungen am Verleger für Abwickelvorgang
MB_TRAVERSE_UNWIND_MODE, Seite 615	ENUM	Auswahl der Abwickelmethode
MB_TRAVERSE_DIRECTION, Seite 613	ENUM	Verfahrrichtung des Verlegers
MB_TRAVERSE_UNWIND_CONTROL, Seite 614	STRUCT	Beschaltung des PID-Reglers für den Abwickelvorgang
MB_WIND_CTRL_TYPE, Seite 615	ENUM	Zugkraftsteuerung/Zugkraftregelung
MB_WIND_DANCER_LIMITS, Seite 615	STRUCT	Begrenzer der Reglerstellgrößen
MB_WIND_DIAM_CALC_MODE, Seite 616	ENUM	Auswahl der Art der Durchmesserberechnung
MB_WIND_DIAMETER, Seite 616	STRUCT	Daten des Durchmesserrechners
MB_WIND_DIRECTION, Seite 618	ENUM	Typ der Wickeleinrichtung
MB_WIND_FRICTION, Seite 618	STRUCT	Daten des Reibmomentes
MB_WIND_TAPER_PROFILE, Seite 618	ENUM	Kurvenprofile
MB_WIND_TENS_DIAG, Seite 618	STRUCT	Reglerabweichung des Tänzerprofils
MB_WIND_TENSION, Seite 619	STRUCT	Daten des Zugspannungsreglers
MB_WIND_TENSION_MODE, Seite 619	ENUM	Auswahl des Modus zum Spannen der Materialbahn
MB_WIND_TORQUE_LIMITS, Seite 619	STRUCT	Bipolare Drehmomentenbegrenzung
MB_DANCER_CONTROL_MODE, Seite 620	ENUM	Mögliche Varianten einer Tänzerregelung

Tab. 8-65: Datentypen der verschiedene Wicklertypen aus der Bibliothek RMB_TechWinder

8.9.2 MB_TRAVERSE_ADAPTION

Mit dieser Aufzählung wird eine adaptive Größe bzw. ein Modus ein- bzw. ausgeschaltet. (z. B. Vorlaufadaption "StrokeAdaption", Geschwindigkeitsadaption "SpikeAdaption")

Element	Wert	Beschreibung
ADAPTION_OFF	0	Adaption ist ausgeschaltet
ADAPTION_ON	1	Adaption ist eingeschaltet

Tab. 8-66: Elemente des Aufzählungstyps MB_TRAVERSE_ADAPTION

8.9.3 MB_TRAVERSE_DIRECTION

Mit dieser Aufzählung wird die Verfahrerrichtung des Verlegers unterschieden. Es ist möglich vom Spulenrand A (PosCoilEdgeA) in Richtung des Spulenrands B (PosCoilEdgeB) zu fahren oder in entgegengesetzte Richtung:

Element	Wert	Beschreibung
A_TO_B	0	Von Spulenrand A nach Spulenrand B.
B_TO_A	1	Von Spulenrand B nach Spulenrand A.

Tab. 8-67: Elemente des Aufzählungstyps MB_TRAVERSE_DIRECTION

8.9.4 MB_TRAVERSE_MODE_LAYER_COUNTER

Der Aufzählungstyp legt fest, wie der Startwert für den Lagenzähler bei pos. Flanke an Enable ermittelt werden soll.

Element	Wert	Beschreibung
INPUT_START_LAYERS	0	Den Startwert für den Lagenzähler vom Eingang einlesen
CALCULATE_START_LAYERS	1	Den Startwert für den Lagenzähler aus dem Spulendurchmesser berechnen

Tab. 8-68: Elemente des Aufzählungstyps MB_TRAVERSE_MODE_LAYER_COUNTER

8.9.5 MB_TRAVERSE_UNWIND

Mit dieser Struktur werden die erforderlichen Einstellungen am Verleger für den Abwickelvorgang vorgenommen.

Es erfolgt die Auswahl der Abwickel-Methode mit dem Element "Mode".

Der Eingang "WireFeedDirection" beschreibt die Richtung aus der der Draht bzw. das Material den Verleger umschlingt.

Mit der Struktur "Control" wird ein PID-Regler konfiguriert, der (abhängig von der Abwickelmethode) die Verlegergeschwindigkeit regelt.

Der Eingang "TraverseVelMiddleFixed" legt die Geschwindigkeit fest, mit der der Verleger im Modus "MIDDLE_FIXED" in die Mittelstellung fährt.

Mit den beiden Eingängen "ActWireTension" und "ActWireTensionTraverse" werden die Werte der Kraftmessdosen (abhängig von der Abwickel-Methode) eingelesen. Die beiden Werte der Kraftmessdosen können jeweils mit einem PT1-Glied gefiltert werden, deren Zeitkonstanten mit den beiden Eingängen "FilterWireTensionTraverse" und "FilterWireTension" vorgegeben werden.

Name	Typ	Min.-Wert	Max.-Wert	Standardwert	Übernahme	Beschreibung
Mode	MB_TRAVERSE_UNWIND_MODE , Seite 615			MIDDLE_FIXED	Steigende Flanke an Enable	Auswahl der unterschiedlichen Abwickelmethode.
WireFeedDirection	MB_TRAVERSE_DIRECTION , Seite 613			A_TO_B	Kontinuierlich	Richtung aus der der Draht bzw. das Material um den Verleger geführt wird.
Control	MB_TRAVERSE_UNWIND_CONTROL , Seite 614				Kontinuierlich	Konfiguration des Reglers für die Geschwindigkeitsregelung beim Abwickeln
VelMiddleFixed	REAL	0.0	n.def.	50	Kontinuierlich	Geschwindigkeit mit der der Verleger im Modus "MIDDLE_FIXED" in die Mittelstellung fährt. [mm/min]
ActWireTensionTraverse	REAL	0.0	n.def.	0.0	Kontinuierlich	Aktueller Messwert der Draht- bzw. Materialspannung am Verleger, gemessen mit einer Kraftmessdose in der Verlegereinheit. [N]
ActWireTension	REAL	0.0	n.def.	0.0	Kontinuierlich	Aktueller Messwert der Draht- bzw. Materialspannung im "System", gemessen mit einer Kraftmessdose außerhalb der Verlegereinheit. [N]
FilterWireTensionTraverse	TIME	0s	n.def.	0s	Kontinuierlich	Zeitkonstante des PT1-Glieds zur Filterung von "ActWireTensionTraverse"
FilterWireTension	TIME	0s	n.def.	0s	Kontinuierlich	Zeitkonstante des PT1-Glieds zur Filterung von "ActWireTension"

Tab. 8-69: Min-/Max und Default-Werte der Struktur MB_TRAVERSE_UNWIND

8.9.6 MB_TRAVERSE_UNWIND_CONTROL

Mit dieser Struktur werden die Eingänge des PID-Reglers für den Abwicklungsvorgang beschalten. Der Ausgang des Reglers beeinflusst additiv die Geschwindigkeit der Verlegerachse.

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme	Beschreibung
PControl	REAL	0.0	n.def.	0.0	Kontinuierlich	Kp-Verstärkung des PID-Geschwindigkeitsreglers
IControl	TIME	0s	n.def.	0s	Kontinuierlich	Nachstellzeit des PID-Geschwindigkeitsreglers
DControl	TIME	0s	n.def.	0s	Kontinuierlich	Vorhaltezeit des PID-Geschwindigkeitsreglers
HighLimit	REAL	LowLimit	n.def.	0.0	Kontinuierlich	Obere Grenze des PID-Geschwindigkeitsreglers [mm/min]
LowLimit	REAL	n.def.	HighLimit	0.0	Kontinuierlich	Untere Grenze des PID-Geschwindigkeitsreglers [mm/min]

Tab. 8-70: Min-/Max und Default-Werte der Struktur MB_TRAVERSE_UNWIND_CONTROL

8.9.7 MB_TRAVERSE_UNWIND_MODE

Der Aufzählungstyp erlaubt die Auswahl der Abwickel-Methode.

Element	Wert	Beschreibung
MIDDLE_FIXED	0	Verleger steht fest in der Mitte der Spule
WINDING_STEP	1	Verleger verfährt gesteuert um Drahtbreite bzw. Materialbreite (plus Lücke)
WIRE_TENSION_TRIM	2	Automatische Anpassung der Verlegergeschwindigkeit in Abhängigkeit der Kräftedifferenz im Verleger und außerhalb des Verlegers. Endposition ist der Spulenrand.
WINDING_STEP_TENSION_TRIM	3	Kombination aus WINDING_STEP und WIRE_TENSION_TRIM, d. h. verfahren um Drahtbreite bzw. Materialbreite (plus Spalt) und additive Korrektur mittels Kräftedifferenz

Tab. 8-71: Elemente des Aufzählungstyps MB_TRAVERSE_UNWIND_MODE

8.9.8 MB_WIND_CTRL_TYPE

Kurzbeschreibung Der Aufzählungstyp MB_WIND_CTRL_TYPE erlaubt die Auswahl Zugkraftsteuerung (sensorlos) oder Zugkraftregelung (Messdose).

Name	Wert	Übernahme	Beschreibung
OPEN_LOOP	16#0000	bei steigender Flanke von Enable	Wickeltyp "open loop", sensorlos
CLOSED_LOOP	16#0001	bei steigender Flanke von Enable	Wickeltyp "closed loop", mit Messdose

Tab. 8-72: Schnittstellenbeschreibung MB_WIND_CTRL_TYPE

8.9.9 MB_WIND_DANCER_LIMITS

Kurzbeschreibung Die Datenstruktur MB_WIND_DANCER_LIMITS beschreibt die im Diagramm dargestellten Begrenzer der Reglerstellgröße.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
LowLimitControl	REAL	0.0	n. def.	0.0	kontinuierlich	Untere Begrenzung des Reglerausganges (bei Drehzahl Null) bipolar [U/min] (der Wert wird intern mit dem Durchmesser Verhältnis von "ReferenceDiameter" und "MinRefDiameter" normiert)
HighLimitControl	REAL	0.0	n. def.	0.0	kontinuierlich	Obere Begrenzung des Reglerausganges (bei Drehzahl VelocityHighLimit) bipolar [U/min] (der Wert wird intern mit dem Durchmesser Verhältnis von "ReferenceDiameter" und "MinRefDiameter" normiert)
VelocityHighLimit	REAL	0.0	n. def.	0.0	kontinuierlich	Drehzahl des Masters für HighLimitControl [U/min]

Tab. 8-73: Schnittstellenbeschreibung Datenstruktur MB_WIND_DANCER_LIMITS

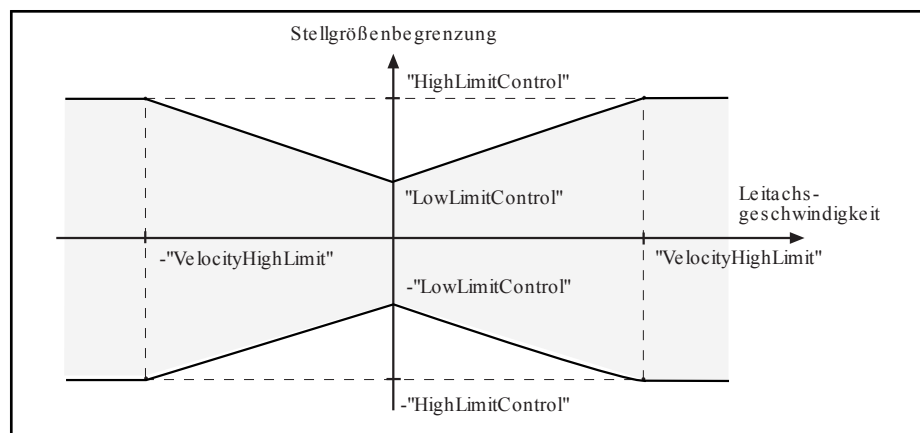


Abb. 8-62: Stellgrößenbegrenzung des Tänzerlagereglers

8.9.10 MB_WIND_DIAM_CALC_MODE

Kurzbeschreibung Der Aufzählungstyp MB_WIND_DIAM_CALC_MODE erlaubt die Auswahl der Art der Durchmesserberechnung.

Name	Wert	Übernahme	Beschreibung
CALC_DIAM_MANUAL	16#0000	bei steigender Flanke von Enable	Durchmesserberechnung abhängig von CalcRange
FIRST_REV_AUTO_CALC	16#0001	bei steigender Flanke von Enable	Durchmesserberechnung erfolgt für 1. Umdrehung alle 3 Grad, danach abhängig vom CalcRange

Tab. 8-74: Schnittstellenbeschreibung MB_WIND_DIAM_CALC_MODE

8.9.11 MB_WIND_DIAMETER

Kurzbeschreibung Die Datenstruktur MB_WIND_DIAMETER beschreibt die für den Durchmesserrechner erforderlichen Daten.

Name	Typ	Min-Wert	Max-Wert	Standardwert	Übernahme	Beschreibung
PresetVal	REAL	> 0.0	n. def.	100.0	kontinuierlich	Vorgabewert für den Presetdurchmesser [mm]
CoreVal	REAL	> 0.0	n. def.	100.0	steigende Flanke an "Enable"	Durchmesser Hülse [mm]
CalcRange	REAL	0.0	n. def.	360.0	kontinuierlich	Bereich für Durchmesserberechnung [Grad] (lastseitig)
RepeatLength	REAL	> 0.0	n. def.	100.0	steigende Flanke an "Enable"	Formatlänge [mm] (1 Umdrehung der Referenzachse abgewickelt)
Webwidth	REAL	0.0	n. def.	0.0	steigende Flanke an "Enable"	Bahnbreite [mm] (wird für das Trägheitsmoment benötigt)
Density	REAL	0.0	n. def.	1.0	steigende Flanke an "Enable"	Materialdichte [kg/dm³] (wird für das Trägheitsmoment benötigt)

Name	Typ	Min-Wert	Max-Wert	Standardwert	Übernahme	Beschreibung
AverageValue	UINT	0	20	1	kontinuierlich	Anzahl der Durchmesserwerte die gemittelt werden [-]
HighLimit	REAL	LowLimit	n. def.	2000.0	kontinuierlich	obere Grenze für den Durchmesser [mm] Der Durchmesser wird auf diesen Wert begrenzt
LowLimit	REAL	> 0.0	HighLimit	100.0	kontinuierlich	untere Grenze für den Durchmesser [mm] Der Durchmesser unterschreitet diesen Wert nicht
MinDiameter	REAL	> 0.0	n. def.	100.0	kontinuierlich	minimaler Durchmesser [mm]
MaxDiameter	REAL	> 0.0	n. def.	2000.0	kontinuierlich	maximaler Durchmesser [mm]
MechanicInertia	REAL	0.0	n. def.	0.0	kontinuierlich	Trägheitsmoment Mechanik [kg m ²] bezogen auf die Motorseite (wird für das resultierende Trägheitsmoment benötigt)
DancerConstant (wird nur bei einem Wickler mit Tänzer benötigt)	REAL	0.0	n. def.	0.0	kontinuierlich	Verstellung des Tänzers bezogen auf die Einheit der Tänzerverstellung [mm/-] (wird nur bei einem Wickler mit Tänzer benötigt)
MeasuredVal	REAL	0.0	n. def.	0.0	kontinuierlich	gemessener Durchmesser [mm]
PivotVelocity	REAL	0.0	n. def.	0.0	kontinuierlich	Schwenkgeschwindigkeit (für Relativbewegung der Wickelachse) [mm/s]
DiamCalcMode	MB_WIND_DIAM_CALC_MODE , Seite 616	CALC_DIAM_MANUAL	FIRST_REV_AUTO_CALC	CALC_DIAM_MANUAL	steigende Flanke an "Enable"	Modus für die Durchmesserberechnung CALC_DIAM_MANUAL Durchmesserberechnung abhängig von CalcRange FIRST_REV_AUTO_CALC Durchmesserberechnung erfolgt für 1. Umdrehung alle 5 Grad, danach abhängig vom CalcRange

Name	Typ	Min-Wert	Max-Wert	Standardwert	Übernahme	Beschreibung
DiamSource	BOOL				steigende Flanke an "Enable"	Durchmesserermittlung: FALSE: Durchmesser wird berechnet TRUE: Durchmesser wird gemessen (MeasuredVal)
MinRefDiameter-Dancer	REAL	0.0	n. def.	100.0	steigende Flanke an "Enable"	Minimaler Bezugsdurchmesser für Wickler mit Tänzer

Tab. 8-75: Schnittstellenbeschreibung Datenstruktur MB_WIND_DIAMETER

8.9.12 MB_WIND_DIRECTION

Kurzbeschreibung Der Aufzählungstyp MB_WIND_DIRECTION erlaubt die Auswahl des Typs der Wickelrichtung.

Name	Wert	Übernahme	Beschreibung
REWIND	16#0000	bei steigender Flanke von Enable	Aufwickler
UNWIND	16#0001	bei steigender Flanke von Enable	Abwickler

Tab. 8-76: Schnittstellenbeschreibung MB_WIND_DIRECTION

8.9.13 MB_WIND_FRICTION

Kurzbeschreibung Die Datenstruktur MB_WIND_FRICTION charakterisiert das Reibmoment.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
StandstillTorque	REAL	0.0	n. def.	0.0	kontinuierlich	Reibmoment im Stillstand
MaxTorque	REAL	0.0	n. def.	0.0	kontinuierlich	Reibmoment bei S-0-0091, Geschwindigkeitsgrenzwert bipolar
MinTorque	REAL	0.0	n. def.	0.0	kontinuierlich	minimales Reibmoment
MinTorqueVelocity	REAL	0.0	n. def.	0.0	kontinuierlich	Geschwindigkeit beim minimalen Reibmoment

Tab. 8-77: Schnittstellenbeschreibung MB_WIND_FRICTION

8.9.14 MB_WIND_TAPER_PROFILE

Element	Wert	Beschreibung
NO_PROFILE	0	Kein Profil
PROFILE_LINEAR	1	Lineares Kurvenprofil
PROFILE_HYPERBOLIC	2	Hyperbolisches Kurvenprofil

Tab. 8-78: Elemente des Aufzählungstyps, MB_WIND_TAPER_PROFILE

8.9.15 MB_WIND_TENS_DIAG

Kurzbeschreibung Die Datenstruktur MB_WIND_TENS_DIAG beschreibt einzelne Komponenten des Drehmomentengrenzwerts, sowie den aktuellen Zugspannungswert zur Diagnose.

Strukturelement	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
TensionCtrlOutput	REAL	n. def.	n. def.	0.0	kontinuierlich	Ausgang des Zugspannungsreglers [N]
TorqueLimit	REAL	n. def.	n. def.	0.0	kontinuierlich	Vorzeichenbehaftetes Torquelimit [%]
AccelerationTorque	REAL	n. def.	n. def.	0.0	kontinuierlich	Aktuelles Beschleunigungsmoment [%]
FrictionTorque	REAL	n. def.	n. def.	0.0	kontinuierlich	Aktuelles Reibmoment [%]
CurrentTension	REAL	0.0	n. def.	0.0	kontinuierlich	Aktueller Zugspannungswert [N]

Tab. 8-79: Schnittstellenbeschreibung Datenstruktur MB_WIND_TENS_DIAG

8.9.16 MB_WIND_TENSION

Kurzbeschreibung Die Datenstruktur MB_WIND_TENSION beschreibt die Daten des Zugspannungsreglers.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
PControl	REAL	0.0	n. def.	0.0	kontinuierlich	Kp-Verstärkung des Zugspannungsreglers
IControl	TIME	0.0	n. def.	0.0	kontinuierlich	Tn Nachstellzeit des Zugspannungsreglers
FeedbackVal	REAL	0.0	n. def.	0.0	kontinuierlich	Istwert der Zugspannung
PreControlWeighting	REAL	0.0	n. def.	1.0	kontinuierlich	Wichtung der Sollwertaufschaltung
HighLimit	REAL	0.0	n. def.	0.0	kontinuierlich	obere Grenze des Zugspannungsreglers
LowLimit	REAL	0.0	n. def.	0.0	kontinuierlich	untere Grenze des Zugspannungsreglers

Tab. 8-80: Schnittstellenbeschreibung MB_WIND_TENSION

8.9.17 MB_WIND_TENSION_MODE

Kurzbeschreibung Der Aufzählungstyp MB_WIND_TENSION_MODE erlaubt die Auswahl des Modus zum Spannen der Materialbahn.

Name	Wert	Übernahme	Beschreibung
CONST_SPEED	16#0000	bei steigender Flanke von Enable	Spannverfahren: konstante Spanndrehzahl
CONST_WEB_VEL	16#0001	bei steigender Flanke von Enable	Spannverfahren: konstante Bahngeschwindigkeit

Tab. 8-81: Schnittstellenbeschreibung MB_WIND_TENSION_MODE

8.9.18 MB_WIND_TORQUE_LIMITS

Struktur für die bipolare Drehmomentbegrenzung (S-0-0092).

Name	Typ	Min-Wert	Max-Wert	Standardwert	Übernahme	Beschreibung
DefaultTorqueLimit	REAL	0.0	400.0	100.0	kontinuierlich	Wert für bipolare Drehmomentengrenze nach Wegnahme von "Enable" [%]
TorqueLimitFactor	REAL	0.0	400.0	50.0	kontinuierlich	Faktor um den das berechnete TorqueLimit erhöht wird [%]
MinTorqueLimit	REAL	0.0	400.0	10.0	kontinuierlich	Minimales TorqueLimit um den das berechnete TorqueLimit erhöht wird [%]

Tab. 8-82: Schnittstellenbeschreibung Datenstruktur MB_WIND_TORQUE_LIMITS

8.9.19 MB_DANCER_CONTROL_MODE

Der Aufzählungstyp MB_DANCER_CONTROL_MODE beschreibt die möglichen Varianten einer Tänzerregelung.

Element	Wert	Beschreibung
P_CONTROL	1	Parameterberechnung des P-Reglers
PI_CONTROL	2	Parameterberechnung des PI-Reglers
PID_CONTROL	3	Parameterberechnung des PID-Reglers

Tab. 8-83: Elemente des Aufzählungstyps MB_DANCER_CONTROL_MODE

8.10 Parameterübersicht

In diesem Abschnitt werden die in der Dokumentation verwendeten Parameter aufgelistet.

Für die Steuerung MLC gibt es "Achsparameter" (A-Parameter), sowie für den jeweiligen Antrieb "Produktspezifische Parameter" (P-Parameter) und "Standardparameter" (S-Parameter).

Für die Steuerung MLD gibt es nur "Produktspezifische Parameter" (P-Parameter) und "Standardparameter" (S-Parameter).

Parameter	Bezeichnung
A-0-0029	Positionspolaritäten
A-0-0038	Drehmoment-/Kraftgrenzwert, bipolar
A-0-0050	Wichtungsart für Drehmoment-/Kraftdaten
A-0-0102	Geschwindigkeitswert
A-0-0500	Konfiguration des benutzerdefinierten Sollwerts A
A-0-0502	Konfiguration des benutzerdefinierten Sollwerts B
A-0-0504	Konfiguration des benutzerdefinierten Sollwerts C
A-0-0506	Konfiguration des benutzerdefinierten Sollwerts D
A-0-0520	Konfiguration des benutzerdefinierten Istwerts A
A-0-0522	Konfiguration des benutzerdefinierten Istwerts B
A-0-0524	Konfiguration des benutzerdefinierten Istwerts C

Parameter	Bezeichnung
A-0-0526	Konfiguration des benutzerdefinierten Istwerts D
A-0-2605	ELS, Feinabgleich der Getriebeübersetzung, Prozessregler
A-0-2615	ELS, Geschwindigkeits-Sollwert additiv, Prozessregler
A-0-2798	Polarität der Leitachsposition

Tab. 8-84: MLC-Achsparemeter (A-Parameter)

Parameter	Bezeichnung
S-0-0040	Geschwindigkeitsistwert Geber 1
S-0-0043	Geschwindigkeits-Polaritäten-Parameter
S-0-0055	Lage-Polaritäten
S-0-0080	Drehmoment-/Kraft-Sollwert
S-0-0081	Drehmoment-/Kraft-Sollwert additiv
S-0-0084	Drehmoment-/Kraft-Istwert
S-0-0085	Drehmoment-/Kraft-Polaritäten-Parameter
S-0-0091	Geschwindigkeits-Grenzwert bipolar
S-0-0092	Drehmoment/Kraft-Grenzwert bipolar
S-0-0100	Geschwindigkeitsregler-Proportionalverstärkung
S-0-0121	Lastgetriebe-Eingangsumdrehungen
S-0-0122	Lastgetriebe-Ausgangsumdrehungen
S-0-0124	Stillstandsfenster

Tab. 8-85: Standardparameter (S-Parameter)

Parameter	Bezeichnung
P-0-0108	Polarität Leitantrieb
P-0-0556	Achsregler-Konfiguration
P-0-0690	Geschwindigkeits-Sollwert additiv, Prozessregler
P-0-0694	Feinabgleich Getriebeübersetzung, Prozessregler
P-0-0912	Virtuelle Leitachse, Stillstandsfenster
P-0-1367	SPS Konfiguration

Tab. 8-86: Produktspezifische Parameter (P-Parameter)

Parameter	Bezeichnung
P-0-1623	CCD: Konfigurationsliste Master-Sollwerte
P-0-1624	CCD: Konfigurationsliste Master-Istwerte
P-0-1625	CCD: Konfigurationsliste Slave-Sollwerte
P-0-1626	CCD: Konfigurationsliste Slave-Istwerte
P-0-1730	CCD: Sollwert Datencontainer 1, Master 4Byte
P-0-1731-39	CCD: Sollwert Datencontainer 1, Slave 1 4Byte, Slave n (n = 1-9)

Parameter	Bezeichnung
P-0-1740	CCD: Sollwert Datencontainer 2, Master 4Byte
P-0-1741-49	CCD: Sollwert Datencontainer 2, Slave 1 4Byte , Slave n (n = 1-9)
P-0-1750	CCD: Sollwert Datencontainer 3, Master 4Byte
P-0-1751-59	CCD: Sollwert Datencontainer 3, Slave 1 4Byte , Slave n (n = 1-9)
P-0-1760	CCD: Sollwert Datencontainer 4, Master 4Byte
P-0-1761-69	CCD: Sollwert Datencontainer 4, Slave 1 4Byte , Slave n (n = 1-9)
P-0-1770	CCD: Istwert Datencontainer 1, Master 4Byte
P-0-1771-79	CCD: Istwert Datencontainer 1, Slave 1 4Byte , Slave n (n = 1-9)
P-0-1780	CCD: Istwert Datencontainer 2, Master 4Byte
P-0-1781-89	CCD: Istwert Datencontainer 2, Slave 1 4Byte , Slave n (n = 1-9)
P-0-1790	CCD: Istwert Datencontainer 3, Master 4Byte
P-0-1791-99	CCD: Istwert Datencontainer 3, Slave 1 4Byte , Slave n (n = 1-9)
P-0-1800	CCD: Istwert Datencontainer 4, Master 4Byte
P-0-1801-09	CCD: Istwert Datencontainer 4, Slave 1 4Byte , Slave n (n = 1-9)
P-0-1820.0.1	CCD: Konfigurationsliste Leitachsen
P-0-1820.0.2	CCD: Quelle des Messgeber-Lageistwertes
P-0-1820.0.3	CCD: Lageistwert Messgeber

Tab. 8-87: *Produktspezifische Parameter (P-Parameter) für den zyklischen Kanal der Steuerung MLD*

9 RMB_TechCrossCutCrossSeal.library // MX_TechCrossCutCrossSeal

9.1 Einführung und Übersicht

Die Bibliothek "RMB_TechCrossCutCrossSeal" beinhaltet Funktionsbausteine für Quersiegler- und Querschneider-Anwendungen.



Die Bibliothek benötigt das **TEC (Technologie) Funktionspaket** der Steuerung.

Funktionsbausteine von RMB_TechCrosscutCrossseal

Anwendungsgebiet (Ordner in der Bibliothek)	
Funktionsbaustein	Beschreibung
Quersiegler- und Querschneider-Anwendungen (CrossCutSeal)	
MB_CrossCutSealType01, Seite 623	berechnet, lädt und aktiviert ein Bewegungsprofil für Quersiegler- oder Querschneider-Anwendungen
Quersiegler- und Querschneider-Berechnungen (CrossCutter/Calculation)	
MB_CrossCutterCalcType04, Seite 638	stellt an seinen Ausgängen alle notwendigen Daten bereit, die in das Bewegungsprofil einer Achse geladen werden müssen, um einen Querschneider oder Quersiegler mit den gewünschten Parametern zu betreiben
MB_CrossCutterCalcType05, Seite 647	berechnet eine Kurvenscheibe für die Anwendung als Querschneider zur Verwendung in einer Achse mit Interpolation im Antrieb. Die Querschneider-Funktionalität ist dabei konfigurierbar.

Tab. 9-1: Überblick über Funktionsbausteinunterstützung durch die Bibliothek

9.2 MB_CrossCutSealType01

Kurzbeschreibung

Der Funktionsbaustein MB_CrossCutSealType01 berechnet, lädt und aktiviert ein Bewegungsprofil für Quersiegler- oder Querschneider-Anwendungen. Der Funktionsbaustein nutzt intern für die Berechnung den Funktionsbaustein [MB_CrossCutterCalcType04, Seite 638](#).

Mit dem errechneten Bewegungsprofil bewegt sich die Slaveachse innerhalb des Siegel- bzw. Schneidebereiches mit Produktgeschwindigkeit. Innerhalb dieses Bereiches können zusätzliche Funktionen, wie z.B. OverSpeed, PushOut oder Kosinus-Korrektur optional zugeschaltet werden.

Außerhalb des Siegel- bzw. Schneidebereiches wird eine nahezu beschleunigungs- und verlustenergieoptimale Ausgleichsbewegung ausgeführt, wenn die vorgegebene Formatlänge nicht dem Messer bzw. Siegelbackenabstand entspricht. Weiterhin kann ein Pendeln der Quersiegler- bzw. Querschneiderachse in frei wählbaren Grenzen vorgegeben werden, um bei großen Formatlängen das Beschleunigungsvermögen des Antriebes optimal zu nutzen.

Im Betrieb können Eingangsgrößen (wie z.B. Formatlänge) verändert werden, wodurch im Hintergrund automatisch ein neues Bewegungsprofil errechnet wird. Das neu errechnete Bewegungsprofil wird anschließend aktiviert und am Umschaltwinkel wirksam.

Bei negativer Flanke an Enable wird der Quersiegler- bzw. Querschneider deaktiviert. Hierbei wird die Achse sofort über die vorgegebene Verzögerungsrampe angehalten.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechCrosscutCrossseal
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_Technology07.lib

Schnittstellenbeschreibung

Tab. 9-2: Referenztable MB_CrossCutSealType01

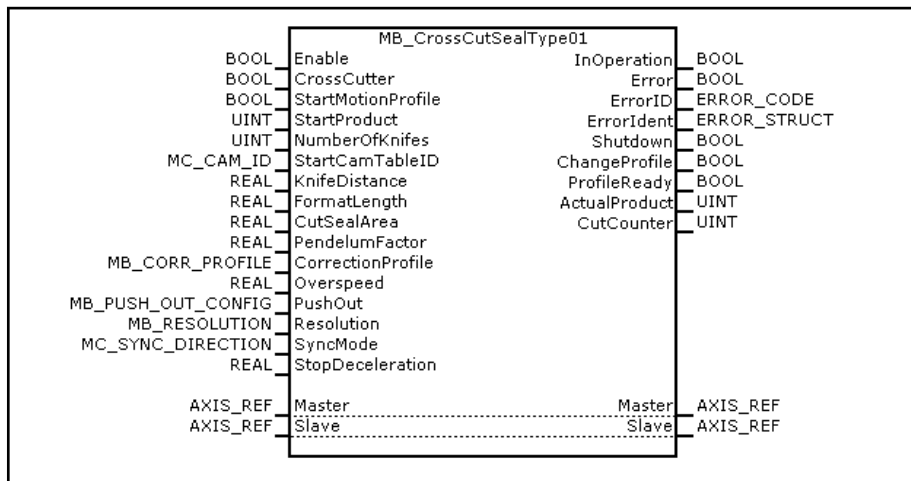


Abb. 9-1: Funktionsbaustein MB_CrossSealType01

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins
	CrossCutter	BOOL	Ist "CrossCutter" = TRUE, wird der Querschneider-Anwendungsfall (siehe Detailbeschreibung unten) ausgewählt. In diesem Anwendungsfall ist die Masterachse so zu normieren, dass in einer Umdrehung das zu schneidende Material die Strecke zurücklegt, die dem Messerabstand entspricht. In diesem Fall wird das verwendete elektronische Getriebe des Bewegungsprofils mit dem Übersetzungsverhältnis "FormatLength/ KnifeDistance" eingestellt. Weiterhin wirkt der Eingang "CutSealArea" als Schnittwinkel des Querschneiders und die Eingangsgrößen "ProductsPerCut" und "StartProduct" wirken nicht. Ist "CrossCutter" = FALSE, wird der Quersiegler-Anwendungsfall (siehe Detailbeschreibung unten) ausgewählt. In diesem Anwendungsfall ist die Masterachse so zu normieren, dass eine Umdrehung einem Folien-Vorschub von einer Produktlänge entspricht. In diesem Fall wird das verwendete elektronische Getriebe des Bewegungsprofils mit dem Übersetzungsverhältnis "1 / ProductsPerCut" eingestellt. Weiterhin entspricht der Eingang "CutSealArea" der Siegellänge und wird damit in translatorischen Einheiten (z.B. Millimeter [mm] oder Inch [in]) vorgegeben. Die Eingänge "ProductsPerCut" und "StartProduct" werden in diesem Fall ausgewertet
	StartMotionProfile	BOOL	Die Aktivierung eines neuen Bewegungsprofils wird blockiert, solange dieser Eingang = FALSE ist. "StartMotionProfile" = TRUE aktiviert das neue Bewegungsprofile, wenn der Ausgang "ProfileReady" = TRUE ist
	ProductsPerCut	UINT	Eingang ist nur bei "CrossCutter" = FALSE wirksam. Anzahl der Produkte pro Siegelvorgang. Entspricht der Anzahl der Masterumdrehungen pro Siegel- bzw. Schneidvorgang
	StartProduct	UINT	Eingang ist nur bei "CrossCutter" = FALSE wirksam. Bei mehr als einem Produkt pro Siegel- bzw. Schneidvorgang ("ProductsPerCut" > 1) kann über diesen Eingang das Start-Produkt bei Aktivierung des FB angegeben werden
	NumberOfKnives	UINT	Anzahl der Siegelbacken bzw. Messer die auf der Siegel- bzw. Schneidwalze am Kreisumfang vorhanden sind. Die Siegelbacken bzw. Messer sind in gleichen Abständen angeordnet
	StartCamTableID	MC_CAM_ID	Startnummer der ersten Kurvenscheibe Punktetabelle. Es wird ein Block von maximal 6 Punkttabellen benötigt

E/A-Typ	Name	Typ	Kommentar
	KnifeDistance	REAL	Messer- bzw. Siegelbackenabstand. Entspricht dem Kreisumfang dividiert mit der Anzahl der Messer oder Siegelbacken [mm oder in]
	FormatLength	REAL	Vorgegebene Schnittlänge bzw. Siegelabstand/Packungslänge [mm oder in]
	CutSealArea	REAL	Innerhalb dieses Bereiches bewegt sich das Messer bzw. die Siegelbacke mit der Produktgeschwindigkeit. Die Geschwindigkeit innerhalb dieses Fensters kann durch zusätzliche Eingänge, wie z.B. "OverSpeed", "PushOut", Kosinus Korrektur... beeinflusst werden. Ist der Eingang "CrossCutter" = FALSE, entspricht dieser Eingang der Siegellänge und wird damit in translatorischen Einheiten vorgegeben (z.B. in Millimeter [mm] oder Inch [in]). Ist der Eingang "CrossCutter" = TRUE, wirkt dieser Eingang rotatorisch als Winkel in [°]
	PendulumFactor	REAL	Maximaler Pendelweg in [%] 0%: kein Pendeln (Stillstand bei großer Formatlänge) 100%: maximales Pendeln (Pendeln bis an die Grenze des Siegel- bzw. Scheidebereiches möglich)
	CorrectionProfile	MB_CORR_PROFILE	Auswahl eines Spezialprofils. Zur Parametrierung des Spezialprofils müssen die Eingänge "OverSpeed" oder "PushOut" verwendet werden MB_CORR_PROFILE, Seite 655
	Overspeed	REAL	Konstante Geschwindigkeitsüberhöhung bzw. Verringerung im Schnittbereich in [%] der Leitachsgeschwindigkeit 0%: kein Overspeed 100%: doppelte Leitachsgeschwindigkeit -50%: halbe Leitachsgeschwindigkeit Dieser Eingang ist nur wirksam wenn die Option CORR_PROFILE_OVERSPEED an dem Eingang "CorrectionProfile" angewählt ist
	PushOut	MB_PUSH_OUT_CONFIG	Struktur zur Definition einer "PushOut"-Bewegung während des Schnitts bzw. Siegelung. MB_PUSH_OUT_CONFIG, Seite 655. In CorrectionProfile muss die Option CORR_PROFILE_PUSH_OUT oder CORR_PROFILE_COS_COMP_AND_PUSH_OUT angewählt sein. Ansonsten ist "PushOut" deaktiviert und somit unwirksam
	Resolution	MB_RESOLUTION	Angabe der Auflösung der Kurvenscheibenberechnung über die Festlegung der Anzahl an Stützstellen in 3 Stufen. MB_RESOLUTION, Seite 657

E/A-Typ	Name	Typ	Kommentar
	SyncMode	MC_SYNC_DIRECTION	Aufsynchronisieren der Richtung (Shortest_Way, Catch_Up, Slow_Down)
	StopDeceleration	REAL	Mit negativer Flanke von "Enable" wird die Slaveachse mit der hier definierten Verzögerung angehalten. Einheiten gemäß Achswichtung in rotatorischen Einheiten
VAR_OUTPUT	InOperation	BOOL	Querscheider läuft im fehlerfreien Betrieb
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	siehe Fehlerbehandlung, Seite 637
	Shutdown	BOOL	Quersiegler wird nach negativer Flanke von "Enable" definiert mit der Verzögerungsrampe StopDeceleration angehalten. In dieser Zeit ist der "ShutDown"-Ausgang gesetzt um das definierte Anhalten anzuzeigen
	ChangeProfile	BOOL	Werden Eingangsgrößen wie z.B. "FormatLength" im laufenden Betrieb verändert, wird im Hintergrund ein neues Bewegungsprofil gerechnet und geladen. Anschließend wird auf das neue Bewegungsprofil umgeschaltet. Da dieser Vorgang mehrere Zyklen dauert, wird in dieser Zeit dieser Ausgang gesetzt. Werden während diesem Vorgang erneut Eingangsgrößen (z.B. "FormatLength") verändert, wird der aktuelle Vorgang abgebrochen und erneut mit den neuen Daten durchgeführt. Der Ausgang wird zurückgesetzt, sobald das neu berechnete Bewegungsprofil wirksam wird und damit auch der Umschaltwinkel durchlaufen wurde
	ProfileReady	BOOL	Dieser Ausgang wird gesetzt, sobald das Bewegungsprofil berechnet geladen und aktiviert ist. Dieser Ausgang wird gelöscht sobald das Bewegungsprofil wirksam ist (nach Durchlaufen des Umschaltwinkels)
	ActualProduct	UINT	Bei mehr als einem Produkt pro Siegelvorgang ("ProductsPerCut" > 1) wird über diesen Ausgang das aktuell bearbeitete Produkt angezeigt. Bei steigender Flanke von "Enable" wird dieser Ausgang auf die Eingangsgröße "StartProduct" zurückgesetzt
	CutCounter	UINT	Bei jedem Moduloüberlauf (360° → 0°) der Slaveachse wird der "CutCounter" erhöht. Mit steigender Flanke von "Enable" wird der "CutCounter" auf 0 zurückgesetzt

E/A-Typ	Name	Typ	Kommentar
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Referenz zur Masterachse
	Slave	AXIS_REF	Referenz zur Slaveachse (= Querschneider- / Quersieglerachse) Der Modulowert beträgt 360° elektrisch und ist mechanisch definiert als Bereich zwischen zwei Messern/ Siegelbacken

Tab. 9-3: Schnittstellenvariablen Funktionsbaustein MB_CrossSealType01

Min-/Max-Werte und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
CrossCutter	BOOL			FALSE	Steigende Flanke an "Enable"
StartMotionProfile	BOOL			TRUE	kontinuierlich
ProductsPerCut	UINT	1	32	1	Steigende Flanke an "Enable"
StartProduct	UINT	1	32	1	Steigende Flanke an "Enable"
NumberOfKnives	UINT	1	8	1	Steigende Flanke an "Enable"
StartCamTableID	MC_CAM_ID	1	94 (bei MLC) 3 (bei MLD)	1	Steigende Flanke an "Enable"
KnifeDistance	REAL	> 0.0mm/in	n. def.	100.0mm/in	Steigende Flanke an "Enable"
FormatLength	REAL	> 0.2* KnifeDistance	30* KnifeDistance	100.0mm/in	kontinuierlich*
CutSealArea	REAL	> 0° > 0mm	< 360° < KnifeDistance	10° 10mm/in	kontinuierlich*
PendulumFactor	REAL	0.0%	100%	0.0%	kontinuierlich*
CorrectionProfile	MB_CORR_PROFILE	0	4	0	kontinuierlich*
Overspeed	REAL	- 50.0%	+100.0%	0.0%	kontinuierlich*
PushOut	MB_PUSH_OUT_CONFIG				kontinuierlich*
Resolution	MB_RESOLUTION	RESOLUTION_LOW	RESOLUTION_HIGH	RESOLUTION_MEDIUM	Steigende Flanke an "Enable"

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
SyncMode	MC_SYNC_DIRECTION	0	2	Catch_Up	Steigende Flanke an "Enable"
StopDeceleration	REAL	> 0.0	n. def.	1000.0	kontinuierlich

* Übernahme der Eingänge dauert einige Zyklen und wird über den Ausgang "ChangeProfile" angezeigt

Tab. 9-4: Min-/Max- und Default-Werte Funktionsbaustein MB_CrossSeal-Type01

Funktionsbeschreibung

Nach Bearbeitungsfreigabe mittels **Enable** berechnet der Funktionsbaustein ein Bewegungsprofil und Punktetabellen für die Applikation Quersiegler/Querschneider gemäß den angelegten Eingangsdaten. Der Funktionsbaustein unterstützt hierbei auch mehrnutzige Quersiegler/Querschneider mit gleichmäßig verteilten Messern bzw. Siegelbacken.



Für mehrnutzige Querschneider bzw. Quersiegler:

Bei mehrnutzigen Querschneider bzw. Quersiegleranwendungen muss die Parametrierung des mech. Getriebes angepasst werden (siehe auch "[Notwendige Randbedingungen und Voraussetzungen](#)" auf Seite 636).

Der errechnete Bewegungsablauf ist dazu in die folgenden zwei Abschnitte unterteilt:

- Der **Schneid- bzw. Siegelbereich** ist durch die Eingangsgröße **CutSealArea** festgelegt. In diesem Bereich bewegt sich das Messer bzw. die Siegelbacke mit Produktgeschwindigkeit. Zusätzlich können Funktionen wie zum Beispiel **OverSpeed** für Geschwindigkeitsüberhöhung oder Reduzierung, **PushOut** für eine Abtrennbewegung, oder die **Kosinus-Kompensation** optional hinzugeschaltet werden.

In diesem Bereich sind folgende Eingangsgrößen wirksam:

- **CorrectionProfile**

bietet Spezialprofile im Schneid- und Siegelbereich an (Overspeed, Pushout, Kosinus Korrektur, PushOut + Kosinus Korrektur).

- **OverSpeed**

gibt die Geschwindigkeitserhöhung bzw. Reduzierung im **CutSealArea** an [%]

- **PushOut**

legt eine spezielle PushOut-Bewegung während des Schneid- bzw. Siegelvorganges fest

- **Außerhalb** des Schneid- bzw. Siegelbereich wird eine ruck- und energieoptimale Ausgleichsbewegung durchgeführt, wenn die vorgegebene Formatlänge **FormatLength** nicht dem Messer bzw. Siegelbackenabstand **KnifeDistance** entspricht. Weiterhin kann ein **Pendeln** der Quersiegler- bzw. Querschneiderachse in einstellbaren Grenzen vorgegeben werden, um bei großen Formatlängen das Beschleunigungsvermögen des Antriebes optimal zu nutzen.

In diesem Bereich ist folgende Eingangsgröße wirksam:

- **PendelumFactor** gibt an, wieviel Pendelweg maximal zugelassen wird [%]

Das Bewegungsprofil und die Punktetabellen werden nach der Berechnung automatisch geladen und aktiviert. In dieser Zeit ist der Ausgang **ChangePro-**

file gesetzt. Ist das Bewegungsprofil wirksam, und die Slaveachse aufsynchronisiert, wird der Ausgang **InOperation** gesetzt. Die Slaveachse bewegt sich in diesem Zustand synchron zu dem Master unter Verwendung des zuvor berechneten Bewegungsprofiles. In diesem Zustand werden mit jeder Umdrehung der Querschneider- bzw. Quersieglerachse der Schnitt bzw. Siegelzähler **CutCounter** erhöht.

Werden im laufenden Betrieb Eingangsgrößen wie zum Beispiel **FormatLength** verändert, wird im Hintergrund ein neues Bewegungsprofil berechnet. Da dieser Vorgang mehrere Zyklen dauert, wird in dieser Zeit der Ausgang **ChangeProfile** gesetzt. Der Ausgang **ProfileReady** wird gesetzt, wenn das neue Bewegungsprofil aktiviert ist, jedoch der Umschaltwinkel noch nicht durchlaufen wurde. Wird anschließend der Umschaltwinkel erreicht, wird das berechnete Bewegungsprofil wirksam und der Ausgang **ChangeProfile** und **ProfileReady** zurückgesetzt. Werden während der laufenden Änderung des Bewegungsprofiles erneut Eingangsdaten verändert, wird die Änderung verworfen und die Berechnung mit den neuen Eingangsdaten neu aufgesetzt.

Die Stützstellen der verwendeten Punktetabellen innerhalb des Bewegungsprofiles kann über den Eingang **Resolution** beeinflusst werden. Die hohe Auflösung führt zur maximalen Genauigkeit, erfordert jedoch den höchsten Zeitbedarf bei der Berechnung. Der Zeitaufwand lässt sich mit den Einstellungen Medium sowie Low um den Faktor 3 bzw. 9 reduzieren. Die Reduzierung der Auflösung ist insbesondere dann sinnvoll wenn bei hohen Taktraten (> 500 Takte/Minute) ständig verschiedene Produktlängen bearbeitet werden.

Eine negative Flanke an **Enable** führt zum definierten Herunterfahren des Querschneiders/Quersieglers. Hierbei wird die Slaveachse mit der einstellbaren Rampe **StopDeceleration** sofort bis zum Stillstand verzögert. Während des definierten Herunterfahrens ist der Ausgang **Shutdown** gesetzt. Der Ausgang **Shutdown** wird zurückgesetzt, wenn die Slaveachse den Stillstand erreicht hat oder ein Fehler aufgetreten ist.

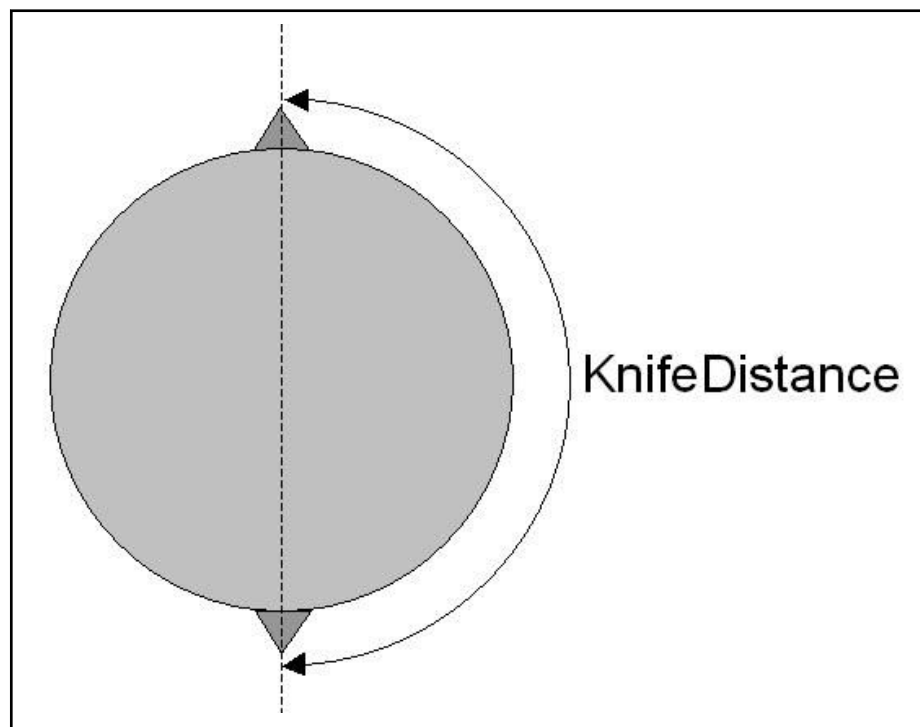


Abb. 9-2: Definition des KnifeDistance Einganges

**zur Wichtung:**

Die translatorischen FB-Eingangsgrößen (z.B. **FormatLength**) können wahlweise in Millimeter [mm] oder Inch [in] vorgegeben werden. Jedoch ist zu beachten, dass alle translatorischen Eingangsgrößen in der gleichen Einheit (mm oder in) vorgegeben werden.

Quersiegler-Anwendungsfall

Über den Eingang **CrossCutter** unterscheidet der Funktionsbaustein grundsätzlich zwischen dem Querschneider- und Quersiegler-Anwendungsfall:

Ist der Eingang **CrossCutter** = FALSE, wird von einer Quersiegelungs-Anwendung ausgegangen. Die Ansteuerung einer Quersiegelwalze von Schlauchbeutelmaschinen ist ein typisches Anwendungsbeispiel für diesen Anwendungsfall. Hierbei soll sich die Siegelbacke während des Siegelvorganges mit Produktgeschwindigkeit und optionalem Korrekturprofil bewegen. Außerhalb des Siegelbereiches soll eine Ausgleichsbewegung durchgeführt werden, wenn die Formatlänge **FormatLength** nicht dem Siegelbackenabstand **KnifeDistance** entspricht. Weiterhin kann der Quersiegler optional durch ein vorgeschaltetes taktisches Ein- und Auskoppeln (LockOn/LockOff) angesteuert werden, um die Funktion "NoGap/NoSeal" und "NoProduct/NoBag" umzusetzen.

Der Quersiegler-Anwendungsfall ist durch folgende Eigenschaften gekennzeichnet:

- Der Eingang **CutSealArea** entspricht der Länge der Siegelnaht und wird damit in translatorischen Einheiten (Millimeter [mm] oder Inch [in]) vorgegeben
- Eine Masterumdrehung (360°) entspricht dem Folienvorschub von einer Produktlänge
- Ein Siegelvorgang kann mehrere Produkte enthalten. Die Anzahl der Produkte pro Siegelvorgang wird über den Eingang **ProductsPerCut** vorgegeben. Dieser Eingang wirkt direkt auf das elektronische Getriebe des verwendeten Bewegungsprofiles
- Über den Eingang **StartProduct** wird das erste Produkt nach positiver Flanke von Enable eingestellt. Diese Größe wirkt intern auf einen Positions-Offset des verwendeten Bewegungsprofiles. Nach positiver Flanke von Enable wird der Ausgang **ActualProduct** auf den Eingang **StartProduct** gesetzt
- Optional kann das taktische Ein- und Auskoppeln (LockOn/LockOff) über einen weiteren Funktionsbaustein MB_CamLock vorgeschaltet werden

**zum Start des FB:**

Stehen Master- und Slaveachse bei Start des FB an einer beliebigen Position, wird die Slaveachse ein dynamisches Aufsynchronisieren ausführen, um sich auf die Masterachse auszurichten. Ist dies nicht gewünscht, müssen vor dem Start des FBs die Master- und Slaveachse auf die Position 0° oder 180° gestellt werden.

Die nachfolgende Abbildung zeigt die Leitachsstuktur und das Wirkschema des FB für den Quersiegler-Anwendungsfall mit optionalem taktischem Ein- und Auskoppeln (LockOn/LockOff):

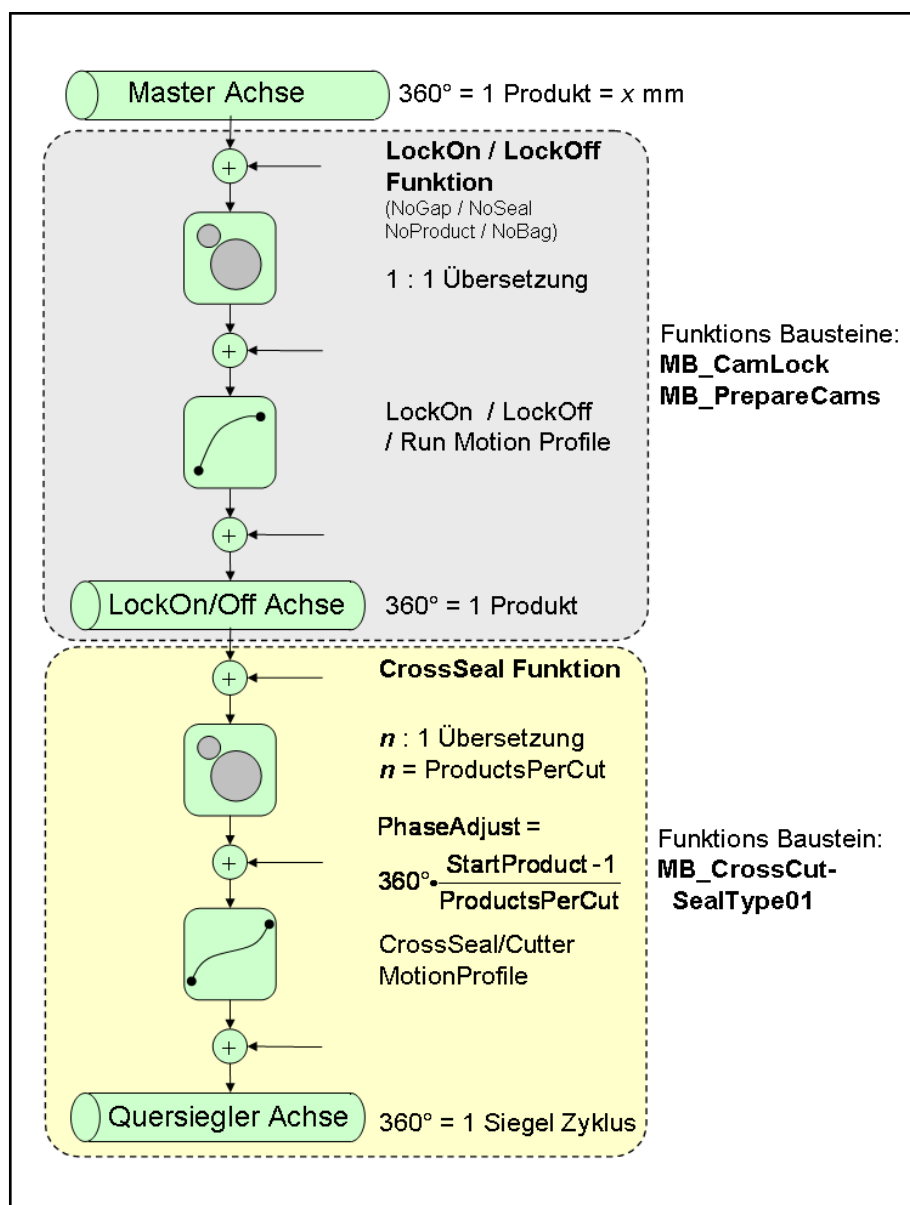


Abb. 9-3: Leitachsstruktur für Quersiegler-Anwendung mit LockOn/LockOff

Die nachfolgende Abbildung zeigt ein exemplarisches Signal-Zeit-Diagramm der Master- und Quersieglerachse für ein Produkt pro Packung:

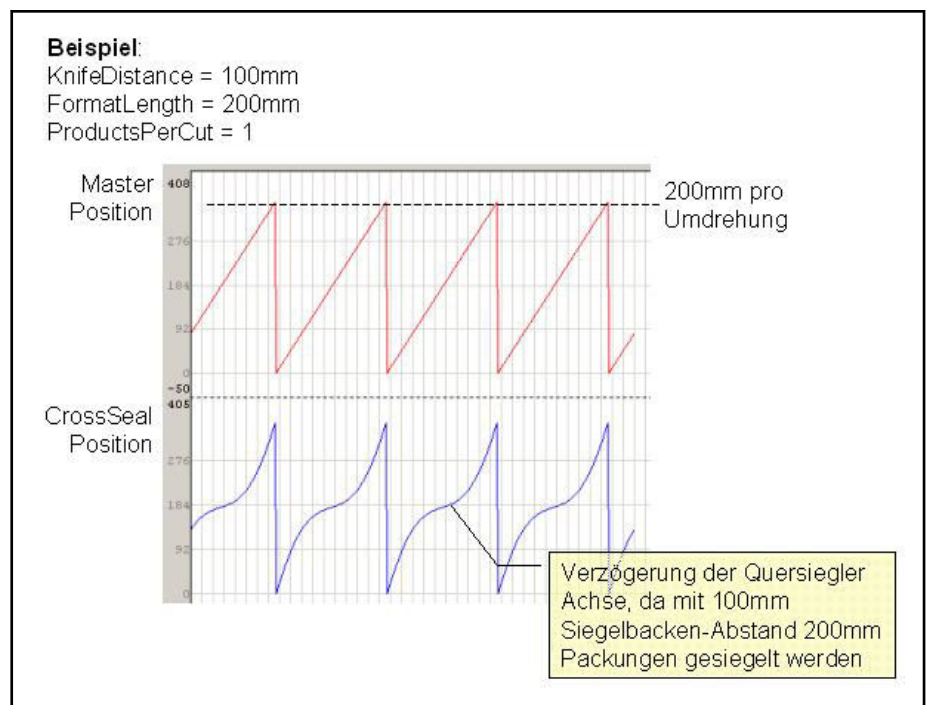


Abb. 9-4: Beispiel Signal-Zeit-Diagramm mit einem Produkt pro Siegelvorgang

Die nachfolgende Abbildung zeigt ein exemplarisches Signal-Zeit-Diagramm der Master- und Quersiegler-Achse für drei Produkte pro Packung:

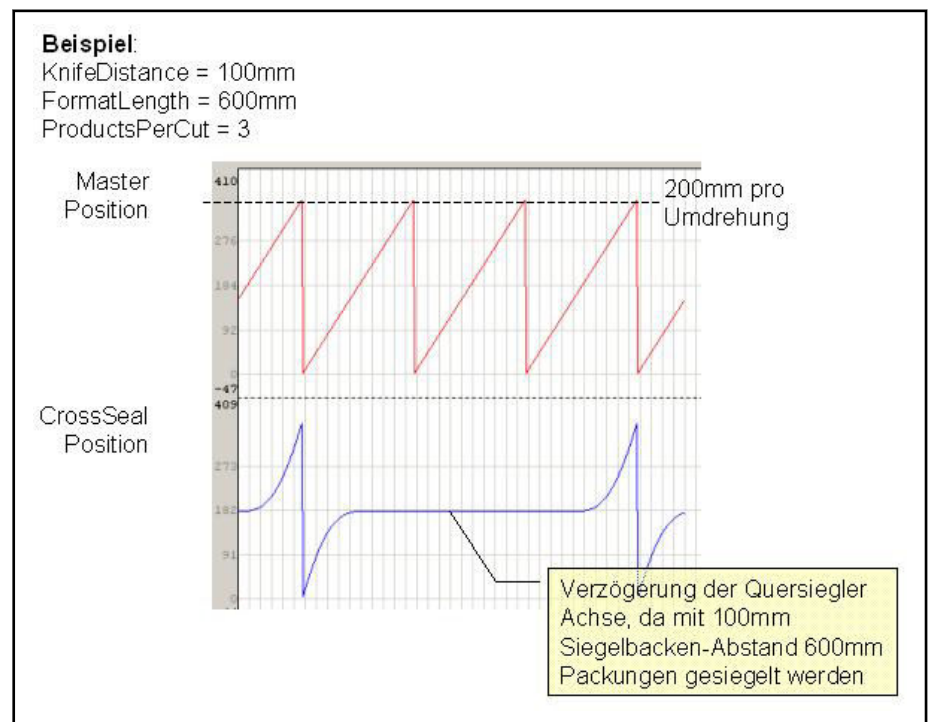


Abb. 9-5: Beispiel Signal-Zeit-Diagramm mit drei Produkten pro Siegelvorgang

Querschneider-Anwendungsfall

Über den Eingang **CrossCutter** unterscheidet der Funktionsbaustein grundsätzlich zwischen dem Querschneider- und Quersiegler-Anwendungsfall.

Ist der Eingang **CrossCutter** = TRUE, wird von einer Querschneider-Anwendung ausgegangen. Während des Schneidevorganges soll sich hierbei das Messer mit Produktgeschwindigkeit und optionalem Korrekturprofil bewegen.

Außerhalb des Schneiderbereiches soll eine Ausgleichsbewegung durchgeführt werden, wenn die Formatlänge **FormatLength** nicht dem Siegelbackenabstand **KnifeDistance** entspricht.

Der Querschneider-Anwendungsfall ist durch folgende Eigenschaften gekennzeichnet:

- Der Eingang **CutSealArea** entspricht dem Winkelbereich des Querschneiders indem er sich mit Produktgeschwindigkeit und dem optionalem Korrekturprofil bewegt
- Eine Master-Umdrehung entspricht der Material-Vorschublänge, die dem Messerabstand **KnifeDistance** entspricht
- Es wird immer von einem Schnitt pro Produkt ausgegangen. Der Eingänge **ProductsPerCut** und **StartProduct** sind nicht wirksam (diese Eingänge werden intern fest auf 1 gelegt)
- Der Ausgang **ActualProduct** ist immer auf "1" festgelegt
- Das verwendete elektronische Getriebe des Bewegungsprofiles wird mit dem Übersetzungsverhältnis **FormatLength / KnifeDistance** eingestellt.

Dieses Getriebe lässt nur ganzzahlige Werte von 1 bis 65535 für Zähler und Nenner der Übersetzung zu. Somit ergibt sich, bedingt durch die Quantisierung, eine maximale Schnittlängenabweichung.

Tritt durch die Quantisierung eine Schnittlängenabweichung auf, wird ein entsprechender Getriebe-Feinabgleich berechnet und vorgegeben. In diesem Falle entsteht jedoch systembedingt eine Drift zwischen Master- und Slave-Achse.

Die nachfolgende Abbildung zeigt die Leitachsstruktur und das Wirkschema des FB für die Querschneider-Anwendung:

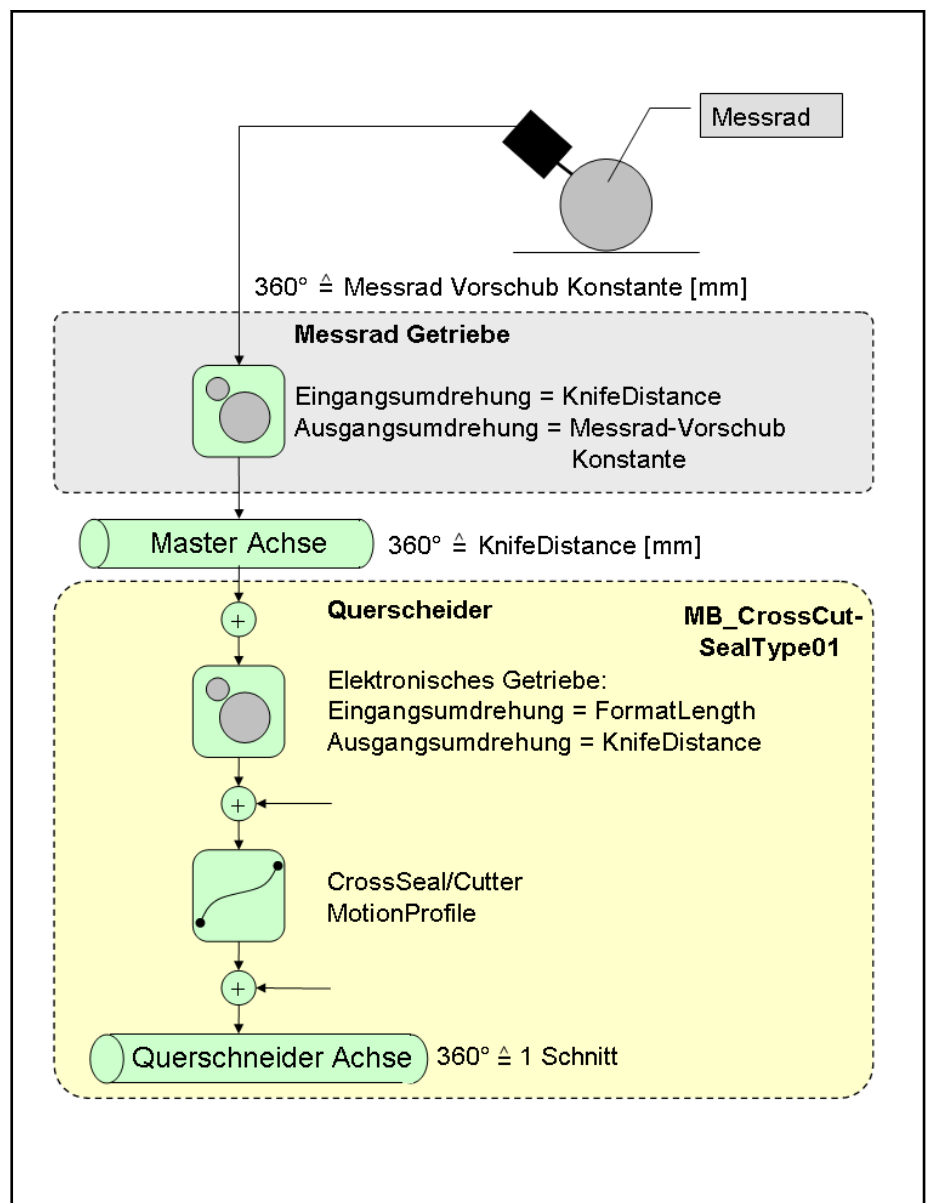


Abb. 9-6: Leitachsstruktur für die Querschneider-Anwendung

Die nachfolgende Abbildung zeigt ein exemplarisches Signal-Zeit-Diagramm der Messrad-, Master- und Querschneiderachse für ein kurzes Format (Formatlänge = 0.5 * Messerabstand):

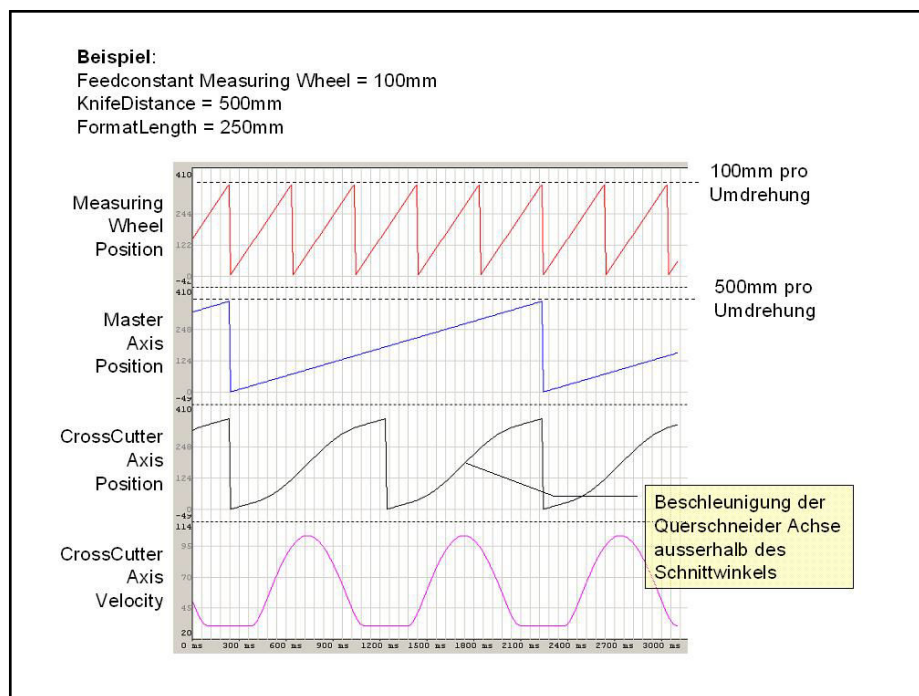


Abb. 9-7: Beispiel Signal-Zeit-Diagramm für Formatlänge = $0.5 \cdot \text{Messerabstand}$

Die nachfolgende Abbildung zeigt ein exemplarisches Signal-Zeit-Diagramm der Messrad-, Master- und Querschneiderachse für ein langes Format (Formatlänge = $2 \cdot \text{Messerabstand}$):

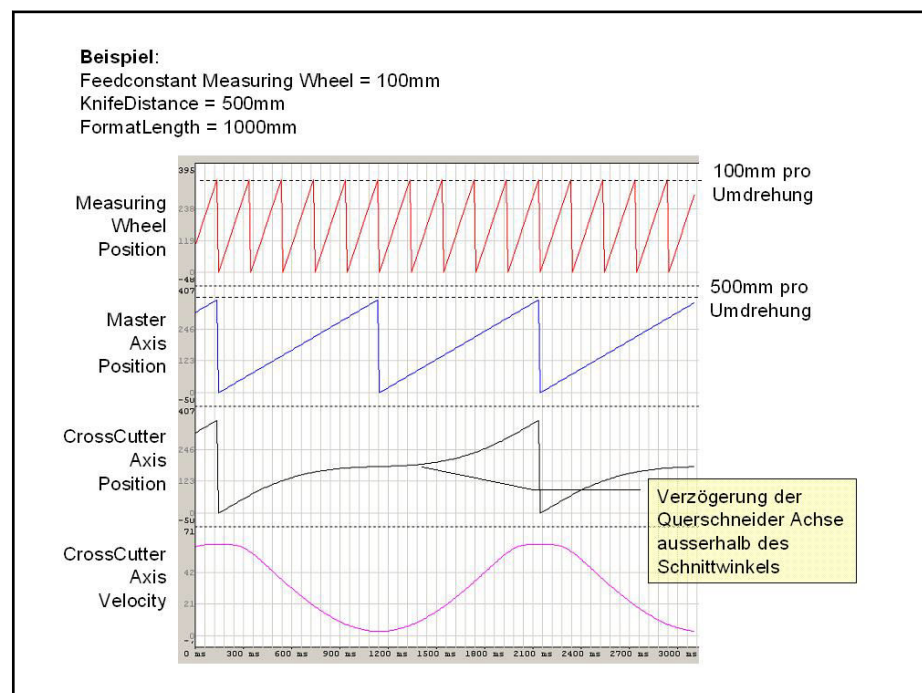


Abb. 9-8: Beispiel Signal-Zeit-Diagramm für Formatlänge = $2 \cdot \text{Messerabstand}$

Notwendige Randbedingungen und Voraussetzungen

- Siegelbacken/Messer sind symmetrisch über den Umfang verteilt
- Die 0° -Position der Slaveachse liegt immer genau in der Mitte des CutSealArea

**zur Antriebs-Parametrierung:**

Bei mehr als einer Siegelbacke bzw. Messer pro Kreisumfang müssen die Einstellungen für das mechanische Getriebe (Antriebsparameter: S-0-0121, S-0-0122) angepasst werden. Diese Parameter sind so einzustellen dass 360° elektrisch (= Modulowert der Slaveachse) dem Abstand zwischen zwei Messern bzw. Siegelbacken entspricht.

Beispiel Getriebeprojektierung

Konfiguration des mechanischen Getriebes bei Verwendung eines Querschneiders mit vier äquidistant auf der Querschneiderachse verteilten Messern

Eingangsgrößen:

Mechanisches Getriebe am Motor, Eingangsumdrehungen $n_{\text{Eingang}} = 10$

Mechanisches Getriebe am Motor, Ausgangsumdrehungen $n_{\text{Ausgang}} = 1$

Messeranzahl pro Kreisumfang $n_{\text{Messer}} = 4$

Berechnung der notwendigen Getriebeeinstellung:

$$\text{mechanisches Getriebe} = \frac{\text{Eingangsumdrehungen}}{\text{Ausgangsumdrehungen}} =$$

$$= \frac{n_{\text{Eingang}}}{n_{\text{Ausgang}} \cdot n_{\text{Messer}}} = \frac{10}{1 \cdot 4} = \frac{10}{4}$$

- S-0-0121 = 10
- S-0-0122 = 4

Bewegt sich bei dieser Parametrierung die Schneidwalze um 90° mechanisch, entspricht dies 360° elektrisch (= Modulowert der Slave-Achse).

Fehlerbehandlung

Der FB führt keine eigenständige motionseitige Fehlerreaktion aus. Bei Fehler-Erkennung wird kein weiteres Motion- Kommando an die Achse gegeben und der FB über den Ausgang **Error** verlassen. Eine negative Flanke von Enable löscht den Fehlerausgang und den Fehlerzustand im FB.

Der FB verwendet die F_RELATED-TABLE (16#0170) Fehlertabelle.

ErrorID	Additional1	Additional2	Beschreibung
ACCESS_ERROR (16#0004)	16#0003	16#0000	Funktionsbaustein wurde von einem anderen Funktionsbaustein unterbrochen
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0001	Der Eingang "ProductsPerCut" liegt außerhalb des zulässigen Bereiches (1..32)
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0002	Der Eingang "StartProduct" liegt außerhalb des zulässigen Bereiches (1..32)
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0003	Der Eingang "NumberOfKnives" liegt außerhalb des zulässigen Bereiches (1..8)
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0004	Der Eingang "StartCamTableID" liegt außerhalb des zulässigen Bereiches (1..94)
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0005	Der Eingang "KnifeDistance" liegt außerhalb des zulässigen Bereiches

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0006	Der Eingang "FormatLength" liegt außerhalb des zulässigen Bereiches
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0007	Der Eingang "CutSealArea" liegt außerhalb des zulässigen Bereiches
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0008	Der Eingang "PendelumFactor" liegt außerhalb des zulässigen Bereiches
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0009	Der Eingang "Overspeed" liegt außerhalb des zulässigen Bereiches
INPUT_RANGE_ERROR (16#0006)	16#0620	16#000A	Der Eingang "Resolution" liegt außerhalb des zulässigen Bereiches
INPUT_RANGE_ERROR (16#0006)	16#0621	16#0000	"CutSealArea" ist zu klein in Verbindung mit der Masterachsgeschwindigkeit und der sercos Zykluszeit

Tab. 9-5: Fehlercodes Funktionsbaustein MB_CrossCutSealType01

9.3

MB_CrossCutterCalcType04

Kurzbeschreibung

Der Funktionsbaustein MB_CrossCutterCalcType04 stellt an seinen Ausgängen alle notwendigen Daten bereit, die in das Bewegungsprofil einer Achse geladen werden müssen, um einen Querschneider oder Quersiegler mit den in den Parametern des Funktionsbausteins ausgewählten Funktionen zu betreiben. Alle Eingänge des Bausteins werden einmalig bei steigender Flanke von "Execute" übernommen. Der Baustein schreibt zudem während seiner Laufzeit direkt in die durch die Pointer spezifizierten Arrays. Die geschriebenen Daten sind erst dann gültig, wenn der Baustein fehlerfrei durchlaufen wurde und "Done" meldet.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechCrosscutCrossseal
MLD mit MPx07/08/17 mit MA Funktionspaket	MX_Technology07.lib

Tab. 9-6: Referenztable MB_CrossCutterCalcType04

Schnittstellenbeschreibung

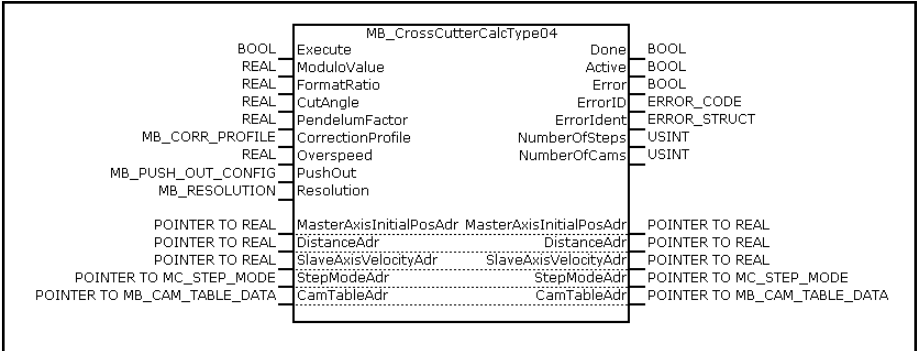


Abb. 9-9: Funktionsbaustein MB_CrossCutterCalcType04

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	Freigabe des Funktionsbausteins (einmalig, flanken-gesteuert)
	ModuloValue	REAL	Modulowert der Querschneiderachse in [°]
	FormatRatio	REAL	Formatverhältnis des zu schneidenden Produktes (einheitenlos)
	CutAngle	REAL	Länge des Schnittbereichs in [°]
	PendelumFactor	REAL	Maximaler Pendelweg in [%] (0%: kein Pendeln, 100%: maximales Pendeln (Pendeln bis an die Grenze des Schnittbereichs))
	CorrectionProfile	MB_CORR_PROFILE	Auswahl eines Spezialprofils. Zur Parametrierung des Spezialprofils müssen die Eingänge "Overspeed" oder "PushOut" benutzt werden. MB_CORR_PROFILE, Seite 655
	Overspeed	REAL	Konstante Geschwindigkeitsüberhöhung oder -redu-zierung im Schnittbereich in [%] der Leitachsge-schwindigkeit (0%: kein Overspeed 100%: doppelte Leitachsgeschwindigkeit -50%: halbe Leitachsgeschwindigkeit) In "CorrectionProfile" muss die Option CORR_PROFI-LE_OVERSPEED angewählt sein. Ansonsten ist "Overspeed" deaktiviert und somit unwirksam
	PushOut	MB_PUSH_OUT_CON-FIG	Struktur zur Definition einer PushOut Bewegung wäh-rend des Schnittes. In "CorrectionProfile" muss die Option CORR_PROFI-LE_PUSH_OUT oder CORR_PROFI-LE_COS_COMP_AND_PUSH_OUT angewählt sein. Ansonsten ist "PushOut" deaktiviert und somit unwirk-sam. MB_PUSH_OUT_CONFIG, Seite 655
	Resolution	MB_RESOLUTION	Angabe der Auflösung der Kurvenscheibenberechnung über die Festlegung der Anzahl an Stützstellen in 3 Stufen. MB_RESOLUTION, Seite 657
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangs-daten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Aus-gang eine grobe Klassifizierung des Fehlers

E/A-Typ	Name	Typ	Kommentar
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang detaillierte Informationen über den Fehler Fehlerbehandlung, Seite 645
	NumberOfSteps	USINT	Anzahl der erzeugten und herunterzuladenden Bewegungsprofilsegmente
	NumberOfCams	USINT	Anzahl der erzeugten und herunterzuladenden Kurvenscheiben
VAR_IN_OUT	MasterAxisInitialPos-Adr	POINTER TO REAL	Zeiger auf ein vom Benutzer anzulegendes Array mit drei Elementen, in das der Baustein die Leitachsenanfangspositionen der generierten Bewegungsprofilsegmente ablegt
	DistanceAdr	POINTER TO REAL	Zeiger auf ein vom Benutzer anzulegendes Array mit drei Elementen, in das der Baustein die Hübe der generierten Bewegungsprofilsegmente ablegt
	SlaveAxisVelocityAdr	POINTER TO REAL	Zeiger auf ein vom Benutzer anzulegendes Array mit drei Elementen, in das der Baustein die Geschwindigkeiten der Folgeachse an den Segmentgrenzen der Bewegungsprofilsegmente ablegt
	StepModeAdr	POINTER TO MC_STEP_MODE	Zeiger auf ein vom Benutzer anzulegendes Array mit drei Elementen, in das der Baustein die Bewegungsschrittmodi der generierten Bewegungsprofilsegmente ablegt
	CamTableAdr	POINTER TO MB_CAM_TABLE_DATA	Zeiger auf ein vom Benutzer anzulegendes Array mit drei Elementen, in das der Baustein Kurvenscheibentabellen, sowie Metadaten zu den Kurvenscheiben ablegt, die als Segment im Bewegungsprofil eingesetzt werden MB_CAM_TABLE_DATA, Seite 653

Tab. 9-7: Funktionsbaustein MB_CrossCutterCalcType04

Min-/Max- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
ModuloValue	REAL	>0.0°	n.def.	360.0°	bei steigender Flanke an "Execute"
FormatRatio	REAL	0.2	30.0	1.0	bei steigender Flanke an "Execute"
CutAngle	REAL	0.0°	< "ModuloValue"	12.0°	bei steigender Flanke an "Execute"
PendulumFactor	REAL	0.0%	100.0%	0.0%	bei steigender Flanke an "Execute"
CorrectionProfile	MB_CORR_PROFILE			CORR_PROFILE_NONE	bei steigender Flanke an "Execute"
Overspeed	REAL	-50.0%	100.0%	0.0%	bei steigender Flanke an "Execute"

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
PushOut	MB_PUSH_OUT_CONFIG				bei steigender Flanke an "Execute"
Resolution	MB_RESOLUTION			RESOLUTION_HIGH	bei steigender Flanke an "Execute"

Tab. 9-8: Min-/Max- und Default-Werte FB MB_CrossCutterCalcType04

Signal-Zeit-Diagramm

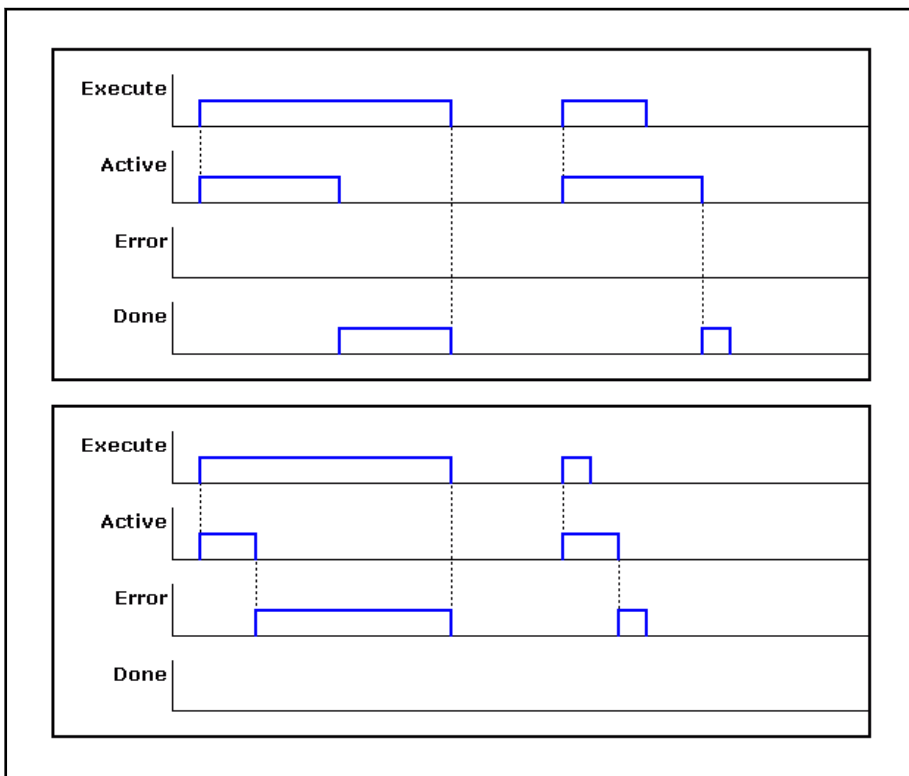


Abb. 9-10: Signal-Zeit-Diagramm Funktionsbaustein MB_CrossCutterCalcType04

Funktionsbeschreibung

Der Funktionsbaustein berechnet auf Basis der Betriebsart "Elektronisches Bewegungsprofil" Profile für die Applikation "Querschneider". Die einzelnen Segmente bestehen dabei entweder aus analytischen Segmenten oder aus Kurvenscheibentabellen. Nach der Bearbeitungsfreigabe mittels "Execute" wird auf Basis der Eingangsgrößen einmalig ein "Elektronisches Bewegungsprofil" aus 3 Segmenten berechnet. Die Kennwerte, die das Profil beschreiben, werden in die jeweiligen Arrays abgelegt, die durch Pointer an den Eingängen des Bausteins "MB_CrossCutterCalcType04" angelegt werden müssen. Die Arrays müssen groß genug sein, um die Anzahl an berechneten Bewegungsschritten (zur Zeit drei) aufnehmen zu können. Das Handling der Pointer ist ähnlich der FBs "MB_ChangeProfileStep"/"MB_ChangeProfileSet", da diese im Anschluss an die Berechnung vom Anwender aufgerufen werden müssen, um das berechnete Profil auf den jeweils freien Satz des "Elektronischen Bewegungsprofils" zu schreiben.

Die folgende Auflistung beschreibt die Bedeutung der einzelnen Eingänge des Funktionsbausteins "MB_CrossCutterCalcType04" genauer:

- Der Eingang "ModuloValue" gibt den Modulowert der Querschneiderachse an. Standardmäßig sind hier 360.0° vorgesehen
- Der Eingang "FormatRatio" kennzeichnet die zu schneidende Produktlänge. Er beschreibt das Verhältnis aus gewünschter Produktlänge und

Umfang des Querschneidezylinders. D.h., bei "FormatRatio==1" wird das geschnittene Produkt die selbe Länge haben, wie der Umfang des Querschneidezylinders (Synchronformat). Bei "FormatRatio==5" hingegen wird das geschnittene Produkt die fünffache Länge des Umfangs des Querschneidezylinders haben

- Der Eingang "CutAngle" kennzeichnet den Schnittbereich des Querschneiders. Im Schnittbereich wird entweder mit Materialvorschubgeschwindigkeit oder mit einer (auswählbaren) Korrekturbewegung gedreht
- Der Eingang "PendulumFactor" kennzeichnet das maximale Überspringen der Querschneiderachse in Prozent. D.h., bei "PendulumFactor==0%" wird die Querschneiderachse niemals mit negativer Geschwindigkeit (also rückwärts) drehen, bei "PendulumFactor==100%" wird die Querschneiderachse bis an den Rand des Schnittbereiches drehen, um die Energieverluste und die nötigen Beschleunigungen zu minimieren. Dies ist insbesondere bei größeren Werten für "FormatRatio" vorteilhaft. Bei der Einstellung "PendulumFactor==100%" wird jedoch nicht grundsätzlich bis an den Rand des Schnittbereiches gedreht, da dies nicht immer vorteilhaft ist. Die jeweils optimale Ausnutzung des verfügbaren Weges wird vom FB stets automatisch berechnet
- Der Eingang "CorrectionProfile" ermöglicht verschiedene Korrekturbewegungen im Schnittbereich, die abhängig vom jeweiligen Anwendungsfall benötigt werden. Auswählbar sind hierbei:
 - "CorrectionProfile==CORR_PROFILE_NONE"
(keine Korrektur: Querschneiderachse dreht mit Geschwindigkeit der Master-Achse)
 - "CorrectionProfile==CORR_PROFILE_OVERSPEED"
(konstante Überhöhung der Geschwindigkeit der Querschneiderachse im Schnittbereich)
 - "CorrectionProfile==CORR_PROFILE_COS_COMP"
(konstanter Geschwindigkeitsvektor in Materialrichtung, z.B. für Guillotineschneider)
 - "CorrectionProfile==CORR_PROFILE_PUSH_OUT"
(PushOut des Materials), sowie
 - "CorrectionProfile==CORR_PROFILE_COS_COMP_AND_PUSH_OUT"
(konstanter Geschwindigkeitsvektor in Materialrichtung sowie gleichzeitigem PushOut, z.B. für Guillotineschneider)
- Der Eingang "Overspeed" ermöglicht die exakte Einstellung der Geschwindigkeitsüberhöhung in [%] im Schnittbereich bei
"CorrectionProfile==CORR_PROFILE_OVERSPEED"
Die Eingabe von 10 [%] erhöht somit die Geschwindigkeit der Querschneiderachse im Schnittbereich um 10% gegenüber der Materialgeschwindigkeit
- Der Eingang "PushOut" parametrisiert die PushOut-Korrektur
Der Eingang wird nur verarbeitet bei
"CorrectionProfile==CORR_PROFILE_PUSH_OUT"
oder
"CorrectionProfile==CORR_PROFILE_COS_COMP_AND_PUSH_OUT"

- Der Eingang "Resolution" erlaubt die Auswahl der Auflösung der verwendeten Kurvenscheibentabellen in 3 Stufen. Durch Wahl einer geringeren Auflösung kann Rechenzeit eingespart werden, allerdings auf Kosten der Genauigkeit. Entsprechend kann mit einer hohen Auflösung auf Kosten der Rechenzeit eine höhere Genauigkeit erzielt werden

Die erfolgreiche Berechnung eines Profils wird durch den Ausgang "Done" signalisiert. Die Kennwerte werden in die durch die angelegten Pointer adressierten Datenfelder geschrieben. Diese müssen im Anschluss noch mittels "MB_ChangeProfileSet"/"MB_ChangeProfileStep" auf ein freies Bewegungsprofil geschrieben werden. Ebenso ist mit den Kurvenscheiben zu verfahren. Dabei ist darauf zu achten, dass in der Struktur "MB_CAM_TABLE_DATA" festgelegt ist, in welche Tabelle die Kurvenscheibe geladen werden muss. Anschließend muss das geladene Bewegungsprofil mittels "MB_MotionProfile" aktiviert werden. Die exakte Schnittlänge wird mittels des elektronischen Getriebes am "MB_MotionProfile" eingestellt. Die Getriebeübersetzung entspricht dabei dem Wert von "FormatRatio". Das heißt, für jede mögliche Produktlänge wird ein neues Profil berechnet, das exakt die Eingangsparameter (insbesondere "CutAngle" und "PendulumFactor" berücksichtigt. Die nachfolgende Abbildung zeigt ein Sequenzdiagramm, das die Verwendung des Bausteins MB_CrossCutterCalcType04 beschreibt.

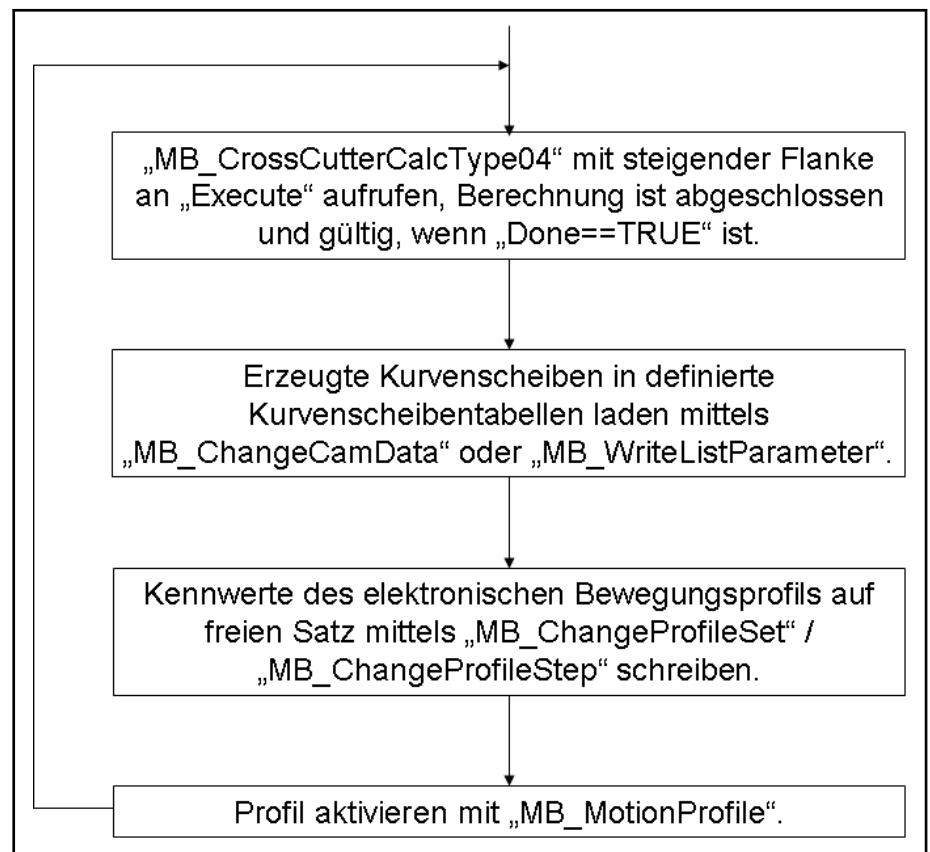


Abb. 9-11: Sequenzdiagramm zur Verwendung von MB_CrossCutterCalcType04

Zur Verwendung mehrerer gleichmäßig (äquidistant) verteilter Messer muss ein mechanisches Getriebe in IndraWorks mit Bezug zur Lastseite projiziert werden. Die Anzahl der Eingangsumdrehungen wird dabei auf 1, die Anzahl der Ausgangsumdrehungen auf die Anzahl der verwendeten Messer n eingestellt. Da jedoch kein reales mechanisches Getriebe vorhanden ist, wird der Antrieb nur jeweils eine n -tel Umdrehung zurücklegen. Somit ist die Ge-

geschwindigkeit der Folgeachse in einer Umdrehung n -mal identisch mit der Geschwindigkeit der Leitachse.

Beispiel:

Konfiguration des mechanischen Getriebes bei Verwendung eines Querschneiders mit vier äquidistant auf der Querschneiderachse verteilten Messern

Eingangsgrößen:

Mechanisches Getriebe am Motor, Eingangsumdrehungen $n_{\text{Eingang}} = 10$

Mechanisches Getriebe am Motor, Ausgangsumdrehungen $n_{\text{Ausgang}} = 1$

Messeranzahl pro Kreisumfang $n_{\text{Messer}} = 4$

Berechnung der notwendigen Getriebeeinstellung:

$$\text{mechanisches Getriebe} = \frac{\text{Eingangsumdrehungen}}{\text{Ausgangsumdrehungen}} =$$

$$= \frac{n_{\text{Eingang}}}{n_{\text{Ausgang}} \cdot n_{\text{Messer}}} = \frac{10}{1 \cdot 4} = \frac{10}{4}$$

- S-0-0121 = 10
- S-0-0122 = 4

Bewegt sich bei dieser Parametrierung die Schneidwalze um 90° mechanisch, so entspricht dies 360° elektrisch (= Modulowert der Slave-Achse).

Um die Geschwindigkeit von Leit- und Folgeachse im Schnittbereich auf den gleichen Wert zu bringen, muss noch die Vorschubkonstante zwischen Messrad und interner virtueller Leitachse angepasst werden. Dabei wird der Umfang der internen virtuellen Leitachse mit der Länge zwischen zwei Messern der realen Folgeachse angenommen. Mit dieser Parametrierung ist es möglich, bei entsprechender Vorlaufzeit, jedes Produkt mit anderer Länge zu schneiden.

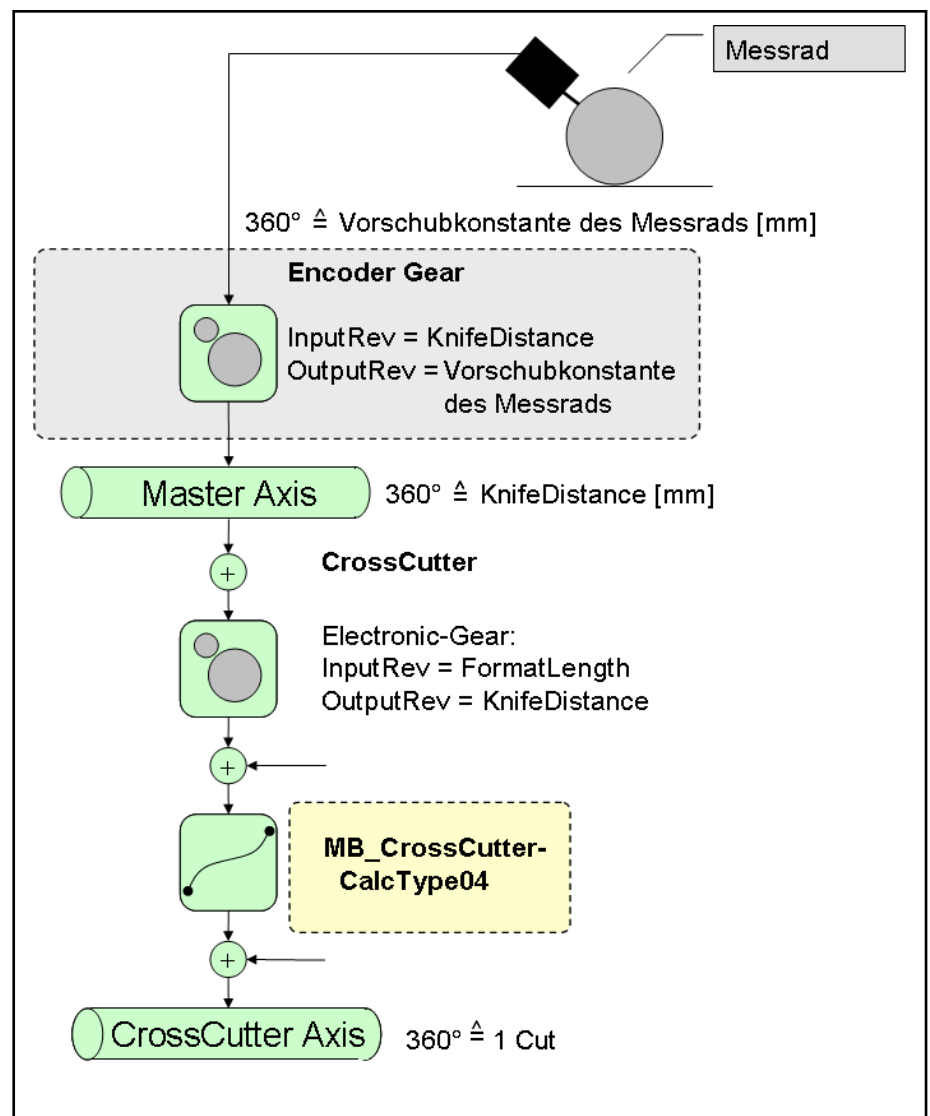


Abb. 9-12: Leitachsstruktur bei Betrieb eines Querschneiders

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#170. Er kann in Additional1 und Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0001	Ungültiger Wert des Eingangs "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0002	Ungültiger Wert des Eingangs "FormatRatio"
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0003	Ungültiger Wert des Eingangs "CutAngle"
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0004	Ungültiger Wert des Eingangs "CutAngle" in Kombination mit Cosinus Kompensation
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0005	Ungültiger Wert des Eingangs "PendulumFactor"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#0006	Ungültiger Wert des Eingangs "CorrectionProfile"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#0007	Ungültiger Wert des Eingangs "Overspeed"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#0008	Ungültiger Startpunkt für die PushOut Funktion

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR (16#0001)	16#0610	16#0009	Ungültiger Wert für "Overspeed_At_Cut" in der Struktur "MB_PUSH_OUT_CONFIG"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000A	Ungültiger Wert für "Overspeed_Max" in der Struktur "MB_PUSH_OUT_CONFIG"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000B	Ungültige Kombination von "Overspeed_At_Cut" und "Overspeed_Max"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000C	Zu kleiner Wert des Eingangs "CutAngle" in Kombination mit PushOut
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000D	Ungültiger Wert des Eingangs "Resolution"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000E	Ungültige Werte der "POINTER"-Eingänge
STATE_MACHINE_ERROR (16#0005)	16#0006	xyyy	Ungültiger Zustand der Zustandsmaschine xx gibt den ungültigen Schritt an yy gibt den ungültigen Zustand an

Tab. 9-9: Fehlercodes Funktionsbaustein MB_CrossCutterCalcType04

Besonderheiten der MLC

Zum Betrieb des Technologie-Funktionsbausteins MB_CrossCutterCalcType04, dem Herunterladen der erzeugten Bewegungsprofile und Kurvenscheiben, sowie zum Verknüpfen der Querschneiderachse mit der Leitachse sind für das System MLC folgende Inbetriebnahmehinweise zu beachten:

- Für die Querschneiderachse muss Modulowichtung mit dem Modulowert 360° aktiv sein
- Der Funktionsbaustein MB_MotionProfile muss stets mit der Einstellung "0: absolute Synchronisation, ohne erneutes Aufsynchronisieren" für den Eingang "StartMode" betrieben werden, um das exakte Auswechseln der Profile zu gewährleisten
- Der Funktionsbaustein MB_MotionProfile lässt für das elektronische Getriebe nur ganzzahlige Werte von 1 bis 65535 für Zähler und Nenner der Übersetzung zu. Somit ergibt sich, bedingt durch die Quantisierung, eine maximale Schnittlängenabweichung. Diese lässt sich durch Multiplikation mit einer großen Konstanten minimieren, um den Wertebereich möglichst voll ausschöpfen zu können
- Das Herunterladen des erzeugten Bewegungsprofils kann auf der MLC mit dem Funktionsbaustein MB_ChangeProfileSet, das Herunterladen der erzeugten Kurvenscheiben mit dem Funktionsbausteinen MB_ChangeCamData oder MB_WriteListParameter erfolgen. Im Unterschied zum MB_WriteListParameter ist MB_ChangeCamData schneller, die Kurvenscheibe wird jedoch nicht persistent gespeichert und muss nach einem Neustart der Steuerung erneut geladen werden
Beim Herunterladen der Kurvenscheiben ist darauf zu achten, dass die entsprechende Kurvenscheibe an die ihr zugeordnete Position (siehe Element "CamTableID" in Tabelle "Schnittstellenvariablen Struktur MB_CAM_TABLE_DATA, Seite 653") geladen wird
- Die Auswahl eines kleinen Formatverhältnisses (< 1) sollte mit Bedacht gewählt werden, da die Querschneiderachse im Ausgleichsbereich sehr hohe Geschwindigkeiten (bis zum 8-fachen der Leitachsgeschwindigkeit) erreicht
- Im Parameter A-0-2930 der Querschneiderachse muss Bit 1 gesetzt werden, damit das Getriebe analog zum Bewegungsprofil am Umschalt-

winkel von 0° ausgetauscht wird und nicht beim Aufruf des Funktionsbausteins MB_MotionProfile

Formatverhältnis	Verhältnis von Folgeachsgeschwindigkeit zur Leitachsgeschwindigkeit
1	1
0,8	1,5
0,6	2,2
0,4	3,7
0,2	8

Tab. 9-10: Auswirkung des Formatverhältnisses auf die maximale Geschwindigkeit der Querschneiderachse im Ausgleichsbereich

9.4 MB_CrossCutterCalcType05

Kurzbeschreibung Dieser Funktionsbaustein berechnet eine Kurvenscheibe für die Anwendung als Querschneider. Die Querschneider-Funktionalität ist dabei konfigurierbar.



Wenn der Baustein mehrere Messer verwenden soll, so ist nur eine eingeschränkte Funktion verfügbar.

Details siehe unter [Mehrere Messer an einem Querschneider auf Seite 651](#).

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS Funktionspaket TEC	RMB_TechCrosscutCrossseal
MLD mit MPx20VRS und höher mit MA Funktionspaket	MX_TechCrosscutCrossseal

Tab. 9-11: Referenztable MB_CrossCutterCalcType05

Schnittstellenbeschreibung

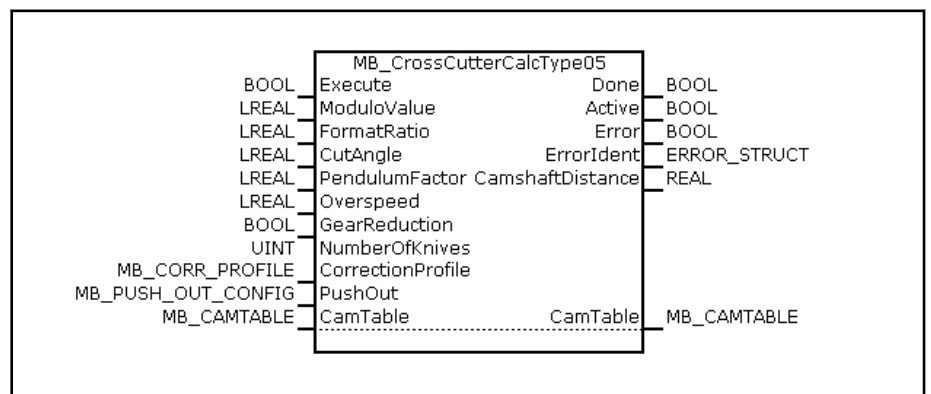


Abb. 9-13: FB MB_CrossCutterCalcType05

	Name	Type	Description
VAR_IN_OUT	CamTable	MB_CAMTABLE	Kurvenscheibe im Format MB_CAMTABLE MB_CAMTABLE auf Seite 102
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	ModuloValue	LREAL	Modulowert der Querschneiderachse in [°]

	Name	Type	Description
	FormatRatio	LREAL	Formatverhältnis des zu schneidenden Produktes (einheitenlos)
	CutAngle	LREAL	Länge des Schnittbereichs in [°].
	PendulumFactor	LREAL	Maximaler Pendelweg in [%] 0%: kein Pendeln, 100%: maximales Pendeln (Pendeln bis an die Grenze des Schnittbereichs)
	Overspeed	LREAL	Konstante Geschwindigkeitsüberhöhung oder -reduzierung im Schnittbereich in [%] der Leitachsgeschwindigkeit (0%: kein Overspeed, 100%: doppelte Leitachsgeschwindigkeit, -50%: halbe Leitachsgeschwindigkeit)
	GearReduction	BOOL	Betrieb mit Untersetzung: FALSE: keine Untersetzung, Kurvenscheibe beginnt bei 0% und endet bei 100% TRUE: mit Untersetzung, Kurvenscheibe beginnt und endet bei 0%
	NumberOfKnives	UINT	Dieser Eingang ist nur wirksam, wenn der Eingang "Gear-Reduction" TRUE ist. Er gibt die Anzahl der Messer an, die sich auf dem Querschneider befinden.
	CorrectionProfile	MB_CORR_PROFILE	Auswahl eines Profils. Die Profile dienen verschiedenen Anwendungsfällen, z.B. Überhöhung der Geschwindigkeit im Schnitt, beschleunigen während des Schnitts oder cosinus Kompensation. Details siehe MB_CORR_PROFILE auf Seite 655
	PushOut	MB_PUSH_OUT_CONFIG	Struktur zur Definition einer PushOut Bewegung während des Schnittes. In "CorrectionProfile" muss die Option CORR_PROFILE_PUSH_OUT oder CORR_PROFILE_COS_COMP_AND_PUSH_OUT angewählt sein. Ansonsten ist "PushOut" deaktiviert und somit unwirksam. Details siehe unter MB_PUSH_OUT_CONFIG auf Seite 655
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig.
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig.
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen.
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers.
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerinformation (siehe Fehlerbehandlung, Seite 652)
	CamshaftDistance	REAL	Wenn der Eingang "GearReduction" mit TRUE belegt wird, so wird an diesem Ausgang ein Wert angegeben, der bei mehreren Messern für den Kurvenscheiben Hub im Antrieb einzutragen ist.

Tab. 9-12: Schnittstellenbeschreibung FB: MB_CrossCutterCalcType05

Min-/Max- und Default-Werte der Eingänge

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Kontinuierlich
ModuloValue	LREAL	>0.0°	n.def.	360.0°	Bei steigender Flanke an "Execute"
FormatRatio	LREAL	0.02	30.0	1.0	Bei steigender Flanke an "Execute"
CutAngle	LREAL	0.0°	"ModuloValue"	12.0°	Bei steigender Flanke an "Execute"
PendulumFactor	LREAL	0.0%	100.0%	0.0%	Bei steigender Flanke an "Execute"
CorrectionProfile	MB_CORR_PROFILE			CORR_PROFILE_NONE	Bei steigender Flanke an "Execute"
GearReduction	BOOL	FALSE	TRUE	FALSE	Bei steigender Flanke an "Execute"
NumberOfKnives	UINT	1	20	1	Bei steigender Flanke an "Execute"
Overspeed	LREAL	-50.0%	100.0%	0.0%	Bei steigender Flanke an "Execute"
PushOut	MB_PUSH_OUT_CONFIG				Bei steigender Flanke an "Execute"

Tab. 9-13: Min-/Max- und Default-Werte FB: MB_CrossCutterCalcType05

Funktionsbeschreibung

Der Funktionsbaustein berechnet auf Basis der Betriebsart "Kurvenscheibe" Profile für die Applikation "Querschneider". Nach der Bearbeitungs freigabe mittels "Execute" wird auf Basis der Eingangsgrößen einmalig eine "Kurvenscheibe" berechnet. Die Kurvenscheibe liegt im Format MB_CAMTABLE vor und besitzt 1024 Stützpunkte. Die Kurvenscheiben werden im "neuen Format" berechnet, d. h. der letzte Punkt liegt bei 0% bzw. 100%.

([MB_CAMTABLE auf Seite 102](#)).

Die Kurvenscheibe kann nur bei einer Achse mit Interpolation im Antrieb verwendet werden.

Die folgende Auflistung beschreibt die Bedeutung der einzelnen Eingänge des Funktionsbausteins "MB_CrossCutterCalcType05" genauer:

- Der Eingang "ModuloValue" gibt den Modulowert der Querschneiderachse an. Standardmäßig sind hier 360.0° vorgesehen.
- Der Eingang "FormatRatio" kennzeichnet die zu schneidende Produktlänge. Er beschreibt das Verhältnis aus gewünschter Produktlänge und Umfang des Querschneidezylinders. D.h., bei "FormatRatio=1" wird das geschnittene Produkt die selbe Länge haben, wie der Umfang des Querschneidezylinders (Synchronformat). Bei `FormatRatio=5` hingegen wird das geschnittene Produkt die fünffache Länge des Umfangs des Querschneidezylinders haben.
- Der Eingang "CutAngle" kennzeichnet den Schnittbereich des Querschneiders. Im Schnittbereich wird entweder mit Materialvorschubgeschwindigkeit oder mit einer (auswählbaren) Korrekturbewegung verfahren. Der Wert bezieht sich auf die Querschneiderachse.

- Der Eingang "PendulumFactor" kennzeichnet das maximale Überspringen der Querschneiderachse in Prozent. D. h., bei "PendulumFactor=0%" wird die Querschneiderachse niemals mit negativer Geschwindigkeit (also rückwärts) drehen, bei "PendulumFactor=100%" wird die Querschneiderachse bis an den Rand des Schnittbereiches drehen, um die Energieverluste und die nötigen Beschleunigungen zu minimieren. Dies ist insbesondere bei größeren Werten für "FormatRatio" vorteilhaft. Bei der Einstellung "PendulumFactor=100%" wird jedoch nicht grundsätzlich bis an den Rand des Schnittbereiches gedreht, da dies nicht immer vorteilhaft ist. Die jeweils optimale Ausnutzung des verfügbaren Weges wird vom FB automatisch berechnet.



Wenn der Baustein mehrere Messer verwenden soll, so ist dieser Eingang NICHT wirksam.

Details siehe unter [Mehrere Messer an einem Querschneider auf Seite 651](#).

- Der Eingang "CorrectionProfile" ermöglicht verschiedene Korrekturbewegungen im Schnittbereich, die abhängig vom jeweiligen Anwendungsfall benötigt werden ([MB_CORR_PROFILE auf Seite 655](#)). Auswählbar sind hierbei:
 - `CorrectionProfile=CORR_PROFILE_NONE`
keine Korrektur: Querschneiderachse dreht mit Geschwindigkeit der Masterachse
 - `CorrectionProfile=CORR_PROFILE_OVERSPEED`
konstante Überhöhung der Geschwindigkeit der Querschneiderachse im Schnittbereich. Dieses Profil ist nur aus Gründen der Kompatibilität zum Baustein MB_CrossCutterCalcType04 vorhanden. Sobald ein Wert für den Eingang "Overspeed" angegeben wird (Wert ungleich 0), wird automatisch das Profil `CorrectionProfile=CORR_PROFILE_OVERSPEED` verwendet, auch wenn das Profil `CorrectionProfile=CORR_PROFILE_NONE` ausgewählt wurde.
 - `CorrectionProfile=CORR_PROFILE_COS_COMP`
konstanter Geschwindigkeitsvektor in Materialrichtung, z.B. für Guillotineschneider
 - `CorrectionProfile=CORR_PROFILE_PUSH_OUT`
PushOut des Materials
 - `CorrectionProfile=CORR_PROFILE_COS_COMP_AND_PUSH_OUT`
konstanter Geschwindigkeitsvektor in Materialrichtung sowie gleichzeitigem PushOut, z.B. für Guillotineschneider.
- Der Eingang "GearReduction" ermöglicht den Betrieb mit oder ohne Untersetzung. Wenn der Eingang TRUE ist, wird die Untersetzung im Antrieb verwendet. Die Kurvenscheibe wird dann so berechnet, dass Beginn und Ende jeweils bei 0% liegen.
- Der Eingang "NumberOfKnives" wird nur ausgewertet, wenn auch der Eingang "GearReduction" TRUE ist. In diesem Fall wird eine Kurvenscheibe erzeugt, bei deren Verwendung die Untersetzung im Antrieb verwendet werden muss. Damit ist es möglich einen Querschneider zu verwenden, der mehrere Messer besitzt. Die Messer müssen dabei

gleichmäßig über den Umfang der Querschneider Achse verteilt sein ([Mehrere Messer an einem Querschneider auf Seite 651](#)).

- Der Eingang "Overspeed" ermöglicht die exakte Einstellung der Geschwindigkeitsüberhöhung in [%] im Schnittbereich bei

`CorrectionProfile=CORR_PROFILE_OVERSPEED` oder `CorrectionProfile=CORR_PROFILE_NONE`.

Die Eingabe von 10 [%] erhöht somit die Geschwindigkeit der Querschneiderachse im Schnittbereich um 10% gegenüber der Materialgeschwindigkeit.

- Der Eingang "PushOut" parametrisiert die PushOut-Korrektur (`MB_PUSH_OUT_CONFIG`). Dabei wird zunächst mit Materialgeschwindigkeit verfahren und dann im Schnittbereich beschleunigt, um z. B. ein Produkt abzulösen.

[MB_PUSH_OUT_CONFIG auf Seite 655](#)

Der Eingang wird nur verarbeitet bei

`CorrectionProfile=CORR_PROFILE_PUSH_OUT`

oder

`CorrectionProfile=CORR_PROFILE_COS_COMP_AND_PUSH_OUT`.

Die erfolgreiche Berechnung eines Profils wird durch den Ausgang "Done" signalisiert. Die Kennwerte werden in die angelegte Datenstruktur "MB_CAM_TABLE" geschrieben. Die Kurvenscheibe muss im Anschluss mittels "MB_LoadCamData" in den Antrieb geschrieben und aktiviert werden.

[MB_LoadCamData auf Seite 99](#)

Mehrere Messer an einem Querschneider

GEFAHR

Werden mehrere Messer verwendet, so sind nicht alle Funktionen des Bausteins verfügbar.

Der Baustein kann NICHT genutzt werden wenn eine oder mehrere der folgenden Bedingungen erfüllt ist:

- Korrekturprofile (Eingang "CorrectionProfile"): `CORR_PROFILE_COS_COMP` und `CORR_PROFILE_COS_COMP_AND_PUSH_OUT`
- Formatverhältnis (Eingang "FormatRatio"): mit Werten 1 und kleiner mit `CorrectionProfile = CORR_PROFILE_PUSH_OUT`
- Modulowerte (Eingang "ModuloValue"): Werte die nicht 360° entsprechen

GEFAHR

Werden mehrere Messer verwendet, so ist die Überprüfung des "PendulumFactor" nicht mehr wirksam. Dies bedeutet, dass sich die Messer bei einer Pendelbewegung in den Schnittbereich bewegen können.

Ob bei den gegebenen Anlagenverhältnissen eine Pendelbewegung bis hin in den Schnittbereich erfolgt, muss durch Tests vorab ermittelt werden.

Zur Verwendung mehrerer gleichmäßig (äquidistant) verteilter Messer bietet der Baustein auch die notwendige Funktionalität. Die Anzahl der Messer wird am Eingang "NumberOfKnives" angelegt. Der Eingang "GearReduction" muss dabei mit TRUE vorbelegt werden. Der Baustein berechnet dann eine Kurvenscheibe, deren Start- und Endwert jeweils bei 0 liegt.

Im Antrieb muss für die Verwendung von mehreren Messern die Untersetzung eingeschaltet werden (Parameter P-0-0086, Bit 4 im Antrieb setzen). Zusätzlich muss auch dort die Anzahl der Messer im Parameter P-0-0755 eingegeben werden. Damit wird erreicht, dass die Messerpositionen zur Leitachse passen. Allerdings stimmt die Geschwindigkeit im Schnittpunkt noch nicht überein. Um dies zu erreichen berechnet der Funktionsbaustein auch noch einen Kurvenscheiben Hub ("CamshaftDistance"). Dadurch wird die Geschwindigkeit im Schnittbereich in der Art angepasst, dass sie der Leitachs-geschwindigkeit entspricht. Der Kurvenscheiben Hub muss im Antrieb in Parameter P-0-0093 eingetragen werden.

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#170. Er kann in Additional1 und Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0001	Ungültiger Wert des Eingangs "ModuloValue".
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0002	Ungültiger Wert des Eingangs "FormatRatio".
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0003	Ungültiger Wert des Eingangs "CutAngle".
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0004	Ungültiger Wert des Eingangs "CutAngle" in Kombination mit Cosinus Kompensation.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0005	Ungültiger Wert des Eingangs "PendulumFactor".
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0006	Ungültiger Wert des Eingangs "CorrectionProfile".
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0007	Ungültiger Wert des Eingangs "Overspeed".
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0008	Ungültiger Startpunkt für die PushOut Funktion.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0009	Ungültiger Wert für "Overspeed_At_Cut" in der Struktur "MB_PUSH_OUT_CONFIG"
INPUT_RANGE_ERROR, 16#0006	16#0610	16#000A	Ungültiger Wert für "Overspeed_Max" in der Struktur "MB_PUSH_OUT_CONFIG"
INPUT_RANGE_ERROR, 16#0006	16#0610	16#000B	Ungültige Kombination von "Overspeed_At_Cut" und "Overspeed_Max"
INPUT_RANGE_ERROR, 16#0006	16#0610	16#000C	Zu kleiner Wert des Eingangs "CutAngle" in Kombination mit PushOut.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#000D	Eingang "NumberOfKnives" besitzt ungültige Werte.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#000E	Bei einem Wert größer 1 für den Eingang "NumberOfKnives" ist das gewählte "CorrectionProfile", der "ModuloValue" oder die Kombination aus "CorrectionProfile" und "FormatRatio" ungültig.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#000F	Der Eingang "CamTable" besitzt keine gültige Adresse. Dies kann durch online change passieren.
STATE_MACHINE_ERROR, 16#0005	16#0006	xxyy	Ungültiger Zustand der Zustandsmaschine. xx gibt den ungültigen Schritt an. yy gibt den ungültigen Zustand an.

Tab. 9-14: Fehlernummern, verursacht durch MB_CrossCutterCalcType05

Besonderheiten für MLC

Um den Baustein mit Achsen an einer MLC zu verwenden, sind die Achsen mit **Interpolation im Antrieb** anzulegen. Andere Achstypen werden nicht unterstützt.

Es wird das Funktionspaket "TECH" in der Steuerung benötigt.

Der Baustein generiert lediglich eine Kurvenscheibe zur Verwendung im Antrieb. Für den weiteren Betrieb sind allerdings noch weitere Aktionen notwendig:

- Schreiben des P-0-0086 (Konfigurationswort Synchronbetriebsarten): Eines der Bits 8-11 muss in diesem Parameter gesetzt sein, damit die Kurvenscheibe im "neuen Format" interpretiert wird.
- Schreiben der Kurvenscheibe in den Antrieb mit Hilfe des Bausteins "MB_LoadCamData". Zu beachten ist, dass die generierte Kurvenscheibe immer 1024 Punkte enthält und damit nur die ersten vier Kurvenscheiben im Antrieb verwendet werden können (die anderen Kurvenscheiben unterstützen nur bis zu 128 Punkte).
- Schreiben des A-0-2745 (ELS, Untersetzung): Enthält die Untersetzung, mit der die Kurvenscheibe bei Betrieb mit mehreren Messern verwendet wird. Der geschriebene Wert muss der Anzahl der verwendeten Messern entsprechen. Nur notwendig wenn mehrere Messer benutzt werden.
- Schreiben des P-0-0086 (Konfigurationswort Synchronbetriebsarten): Bit 4 muss in diesem Parameter gesetzt sein, damit der Untersetzungspfad aktiviert wird. Nur notwendig wenn mehrere Messer benutzt werden.
- Die verwendete Achse muss anschließend in der Synchronbetriebsart "Kurvenscheibe" betrieben werden. Die dabei anzugebende "CamShaft-Distance" entspricht dem vom Baustein ausgegebenen Wert.

Besonderheiten für MLD Es wird das Funktionspaket "MA" in der MLD benötigt.

9.5 MB_CAM_TABLE_DATA

Kurzbeschreibung Die Struktur MB_CAM_TABLE_DATA wird von den Bausteinen [MB_Cross-CutterCalcType04, Seite 638](#), und [MB_CrossCutterCalcType05, Seite 647](#), verwendet.

Die Struktur enthält ein Array mit einer Kurvenscheibentabelle, sowie Meta-Daten, die die Anzahl der Stützpunkte des Arrays, sowie die Position, an die die Tabelle in die Steuerung/den Antrieb geladen werden muss, beschreiben.

Strukturelement	Typ	Beschreibung
CamTableID	DINT	Nummer der Kurvenscheibentabelle in die die beschriebene Kurvenscheibe geladen werden muss
NumberOfElements	UINT	Anzahl an Stützstellen der beschriebenen Kurvenscheibe
CamTable	ARRAY[0..1023] OF DINT	Array mit den Stützpunkten der Kurvenscheibe

Tab. 9-15: Schnittstellenvariablen Struktur MB_CAM_TABLE_DATA

Min-/Max- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
CamTableID	DINT	0		0
NumberOfElements	UINT	0	1024	0

Tab. 9-16: Min-/Max- und Default-Werte der Struktur MB_CAM_TABLE_DATA

Funktionsbeschreibung

Im Strukturelement "CamTableID" ist die Nummer der Kurvenscheibentabelle in dem Format abgelegt, wie es die ML_PLCOpen-Funktionsbausteine

Tab. 9-17: Zuordnung von CamTableID zu Parameternummern



9.6 MB_CORR_PROFILE

Kurzbeschreibung Die Enumeration MB_CORR_PROFILE wird von den Funktionsbausteinen [MB_CrossCutterCalcType04, Seite 638](#), und [MB_CrossCutterCalcType05, Seite 647](#), verwendet, um eine Korrekturbewegung im Schnittbereich zu definieren.

Elemente des Aufzählungstyps In der folgenden Tabelle sind die Elemente des Aufzählungstyps aufgelistet.

Element	Wert	Beschreibung
CORR_PROFILE_NONE	0	Keine Korrektur im Schnittbereich. Geschwindigkeit der Querschneiderachse entspricht der Geschwindigkeit der Masterachse
CORR_PROFILE_OVERSPEED	1	Konstante Überhöhung oder Reduzierung der Geschwindigkeit der Querschneiderachse im Schnittbereich. Die Geschwindigkeit der Querschneiderachse ist dabei um den am Eingang "Overspeed" des Funktionsbausteins MB_CrossCutterCalcType04 parametrisierten Wert schneller oder langsamer als die Geschwindigkeit der Masterachse
CORR_PROFILE_COS_COMP	2	Korrektur der Geschwindigkeit der Querschneiderachse. Die Änderung erfolgt so, dass die Querschneiderachse in Materialrichtung (also die x-Komponente des Geschwindigkeitsvektors) mit konstanter Geschwindigkeit dreht. Dabei muss die Drehzahl der Querschneiderachse in Abhängigkeit der aktuellen Lage im Schnitt angepasst werden
CORR_PROFILE_PUSH_OUT	3	Durchführen einer PushOut-Bewegung am Ende des Schnittes. Parametrierung über den Eingang "PushOut" des Funktionsbausteins MB_CrossCutterCalcType04. Dabei muss die Drehzahl der Querschneiderachse in Abhängigkeit der aktuellen Lage im Schnitt angepasst werden. MB_PUSH_OUT_CONFIG, Seite 655
CORR_PROFILE_COS_COMP_AND_PUSH_OUT	4	Kombination aus den Korrekturen "CORR_PROFILE_COS_COMP" und "CORR_PROFILE_PUSH_OUT". Parametrierung über den Eingang "PushOut" des Funktionsbausteins MB_CrossCutterCalcType04. MB_PUSH_OUT_CONFIG, Seite 655

Tab. 9-18: Elemente des Aufzählungstyps MB_CORR_PROFILE

9.7 MB_PUSH_OUT_CONFIG

Kurzbeschreibung Die Struktur MB_PUSH_OUT_CONFIG wird von den Bausteinen [MB_CrossCutterCalcType04, Seite 638](#), und [MB_CrossCutterCalcType05, Seite 647](#), verwendet.

Durch die Struktur kann ein definierter PushOut am Ende des Schnittes parametrisiert werden.

Schnittstellenbeschreibung In der folgenden Tabelle sind die Schnittstellenvariablen des Funktionsbausteins aufgelistet.

Strukturelement	Typ	Beschreibung
PushOutBegin	REAL	Beginn des PushOut-Bereichs (Parameter A). Wert muss ungleich 0 sein, Einheit in [°]
Overspeed_At_Cut	REAL	Überhöhung der Geschwindigkeit während des Schnittes (Parameter B). Wert muss ungleich 0 sein, Einheit in [%]
Overspeed_Max	REAL	Maximale Übergeschwindigkeit (Parameter C), Einheit in [%]

Tab. 9-19: Schnittstellenvariablen Struktur MB_PUSH_OUT_CONFIG

Min-/Max- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert
PushOutBegin	REAL	$-\frac{\text{"CutAngle"}}{2}$	$\frac{\text{"CutAngle"}}{2}$	-5.0°
Overspeed_At_Cut	REAL	-100.0%	100.0%	5.0%
Overspeed_Max	REAL	>0.0%	100.0%	10.0%

Tab. 9-20: Min-/Max- und Default-Werte der Struktur MB_PUSH_OUT_CONFIG

Funktionsbeschreibung

Die folgenden Grafiken zeigen den Geschwindigkeitsverlauf der Querschneiderwalze im Schnittbereich. Die einzelnen Parameter können durch die Elemente der Struktur MB_PUSH_OUT_CONFIG angepasst werden.

Für die Grafiken ist ein Kurvenprofil mit einem Schnittwinkel von 20° angenommen. Die Steilheit der Geschwindigkeitsänderung wird dabei über die beiden Variablen "Overspeed_At_Cut" und "Overspeed_Max" bestimmt. Der Beginn der Geschwindigkeitserhöhung wird mittels "PushOutBegin" festgelegt. Zur eindeutigen Parametrierung ist es notwendig, dass "PushOutBegin" und "Overspeed_At_Cut" ungleich Null sind. Die beiden Abbildungen zeigen exemplarisch die Parametrierung. Die grüne Linie kennzeichnet die Steilheit des Anstiegs und lässt sich durch Verbinden der Punkte A, B und C bestimmen.

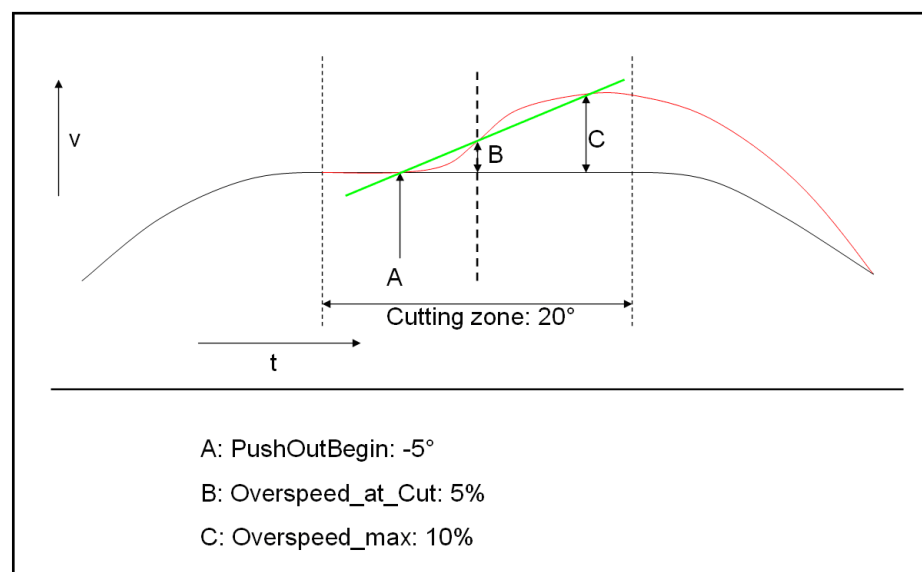


Abb. 9-15: Definition des PushOut. PushOutBegin < 0

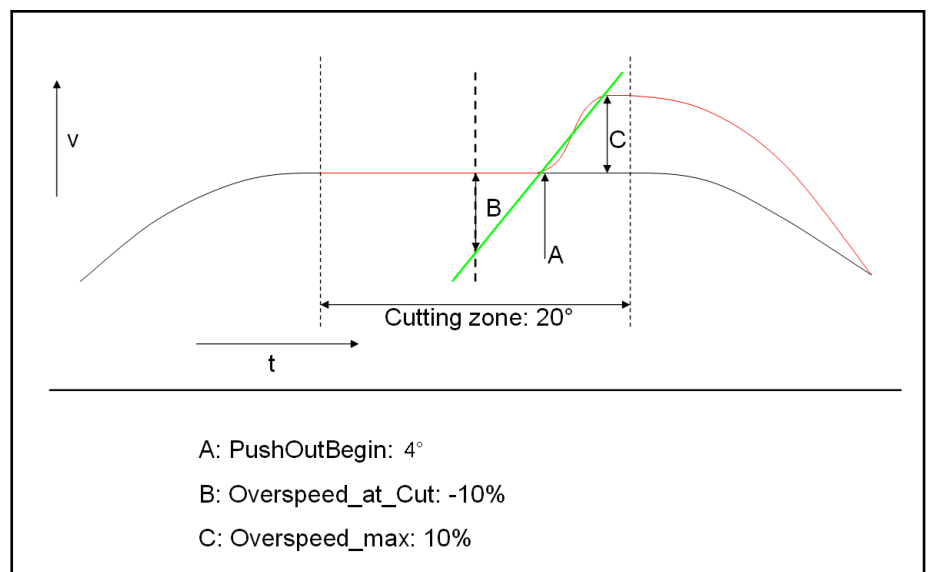


Abb. 9-16: Definition des PushOut.PushOutBegin > 0

9.8 MB_RESOLUTION

Kurzbeschreibung Die Enumeration MB_RESOLUTION wird vom Baustein [MB_CrossCutter-CalcType04](#), Seite 638, verwendet.

Mit der Enumeration kann die Anzahl der Stützpunkte der Kurvenscheibentabellen und somit die benötigte Rechenzeit variiert werden.

Elemente des Aufzählungstyps In der folgenden Tabelle sind die Elemente des Aufzählungstyps aufgelistet.

Element	Wert	Beschreibung	Anzahl Stützstellen (Schnittsegment + Ausgleichssegment + Schnittsegment)
RESOLUTION_LOW	1	Niedrige Auflösung, geringe Rechenzeit nötig (um ca. Faktor 3 schneller als bei mittlerer Auflösung)	10 + 100 + 10
RESOLUTION_MIDDLE	2	Mittlere Auflösung, mittlere Rechenzeit nötig (um ca. Faktor 3 schneller als bei hoher Auflösung)	25 + 300 + 25
RESOLUTION_HIGH	3	Hohe Auflösung, hohe Rechenzeit nötig	50 + 1000 + 50

Tab. 9-21: Elemente des Aufzählungstyps MB_RESOLUTION

Funktionsbeschreibung Mit der Enumeration MB_RESOLUTION kann die Anzahl der Stützpunkte der Kurvenscheibentabellen und somit die benötigte Rechenzeit variiert werden.

Für kurze Formate und wenig Vorlaufzeit für einen Formatwechsel kann die Rechenzeit des Bausteins durch Auswahl der Option "RESOLUTION_LOW" verkürzt werden. Dabei ist allerdings darauf zu achten, dass die Auflösung der erzeugten Kurvenscheiben sinkt und die Maschine unruhig laufen kann. Bei langen Formaten und großen Vorlaufzeiten für Formatwechsel kann die Option "RESOLUTION_HIGH" gewählt werden, womit die erzeugten Kurvenscheiben in größtmöglicher Auflösung generiert werden. Grundsätzlich ist diese Einstellung, also "RESOLUTION_HIGH", zu bevorzugen.

Welche Option hier gewählt werden muss, ist stark abhängig von der eingesetzten Applikation und sollte bei der Inbetriebnahme durch Versuche bestimmt werden.

10 RMB_TechCam.library

10.1 Einführung und Übersicht

Die hier beschriebene Bibliothek beinhaltet Funktionsbausteine zur Erstellung von Kurvenscheibentabellen.

Die Funktionsbausteine sind für unterschiedliche Anwendungen wie Vorschubbewegung, Querschneider, Slotter etc. verfügbar.

Die Funktionsbausteine MB_FeedMovementCalcType01, MB_FeedMovementCalcType02 und MB_CoordinatePairCamType01 arbeiten mit der Enumeration MB_CAMPROF, Seite 102 und der Struktur MB_CAMTABLE, Seite 102, welche sich in der Bibliothek "ML_TechMotionBase" befinden.

Die Funktionsbausteine MB_SlotterCalcType01 und MB_SlotterCalcType02 arbeiten mit der Enumeration MB_CAMPROF_SLOTTER, Seite 708, und der selben Struktur "MB_CAMTABLE".

Die Kurvenscheibentabellen der genannten Funktionsbausteine können mittels Funktionsbaustein MB_LoadCamData, Seite 99 der Bibliothek "ML_TechMotionBase" in die Steuerung oder in einen Antrieb geladen werden.



Die Bibliothek benötigt das **TEC (Technologie) Funktionspaket** der Steuerung

Die Funktionsbausteine von RMB_TechCam

Funktionsbaustein	Beschreibung
MB_CoordinatePairCamType01, Seite 659	Übernimmt die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen, die abhängig des ausgewählten Koordinatenpaartyps ermittelt wird
MB_FeedMovementCalcType01, Seite 667	Übernimmt die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen für eine Vorschubbewegung
MB_FeedMovementCalcType02, Seite 673	Übernimmt, unter Benutzung eines Polynom 7. Ordnung, die Berechnung einer Kurvenscheibentabelle mit bis zu 1024 Stützpunkten, bei der der Ruck-, Beschleunigungs- und Geschwindigkeits-Eingang begrenzt werden kann um eine optimale Ausführung zu gewährleisten
MB_SlotterCalcType01, Seite 685	Übernimmt die Berechnung einer Kurvenscheibentabelle mit 1024 Stützpunktelementen für einen Slotter mit einem Messer auf der Slotterwalze
MB_SlotterCalcType02, Seite 697	Übernimmt die Berechnung einer Kurvenscheibentabelle mit 1024 Stützpunktelementen für einen Slotter mit zwei Messern auf der Slotterwalze

Tab. 10-1: RMB_TechCam

10.2 MB_CoordinatePairCamType01

Kurzbeschreibung

Der Funktionsbaustein berechnet eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen.

Die Kurvenscheibentabelle wird abhängig vom ausgewählten Koordinatenpaartyp ermittelt.

Zuordnung Zielsystem/Bibliothek

MLC 14VRS	RMB_TechCam
Funktionspaket TEC	

Schnittstellenbeschreibung

Tab. 10-2: Referenztable MB_CoordinatePairCamType01

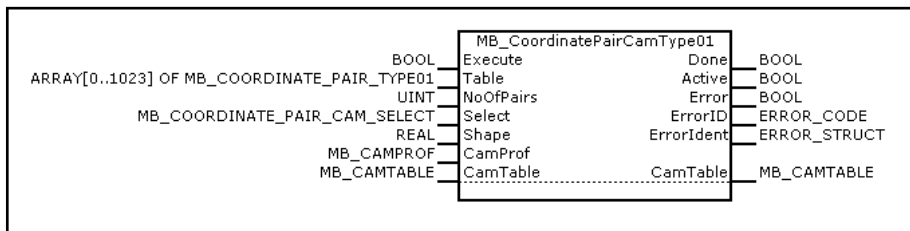


Abb. 10-1: Funktionsbaustein MB_CoordinatePairCamType01

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	Table	ARRAY[0..1023] OF MB_COORDINATE_PAIR_TYPE01	Liste der Koordinatenpaare MB_COORDINATE_PAIR_TYPE01, Seite 666
	NoOfPairs	UINT	Anzahl gültiger Koordinatenpaare
	Select	MB_COORDINATE_PAIR_CAM_SELECT	Auswahl der Koordinatenpaartypen MB_COORDINATE_PAIR_CAM_SELECT, Seite 666
	Shape	REAL	Faktor für Ruck-Begrenzung
	CamProf	MB_CAMPROF	Auswahl des Kurvenscheibenprofils
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe Fehlerbehandlung, Seite 666)
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellen-Elementen und Information über die Anzahl der Elemente

Tab. 10-3: Funktionsbaustein MB_CoordinatePairCamType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
NoOfPairs	UINT	1 bzw. 2	1023	0	Steigende Flanke an "Execute"
Select	MB_COORDINATE_PAIR_CAM_SELECT			COORDINATE_PAIR_POS_VEL_TYPE01	Steigende Flanke an "Execute"

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Shape	REAL	0.0	1.0	0.0	Steigende Flanke an "Execute"
CamProf	MB_CAMPROF			CAM_PROF_STANDARD	Steigende Flanke an "Execute"

Tab. 10-4: Min-/Max- und Default-Werte FB MB_CoordinatePairCamType01

Die Enumeration [MB_CAMPROF](#), Seite 102, wird verwendet, um zwischen verschiedenen Kurvenscheibenprofilen zu wählen.



Bei der Auswahl des Kurvenscheibenprofils "CAMPROF_ADDITIVE_SEGMENTABLE" sind folgende Hinweise zu beachten:

- Bei Kurventabellen für Bewegungen mit unbegrenztem (unendlichem) Verfahrbereich, deren erstes und letztes Element identisch sind, ist der Linearpfad über ein Häkchen zu aktivieren und die Reduktion auf den Wert "1" einzustellen, da es sich um ein Profil mit Untersetzung handelt. Bei allen weiteren Kurventabellen ($|\text{letztes\_Element} - \text{erstes\_Element}| = 100\%$) muss der Linearpfad deaktiviert sein, da es sich um Profile ohne Untersetzung handelt.
- Bei Kurventabellen für Bewegungen mit begrenztem (endlichem) Verfahrbereich muss der Linearpfad deaktiviert sein.

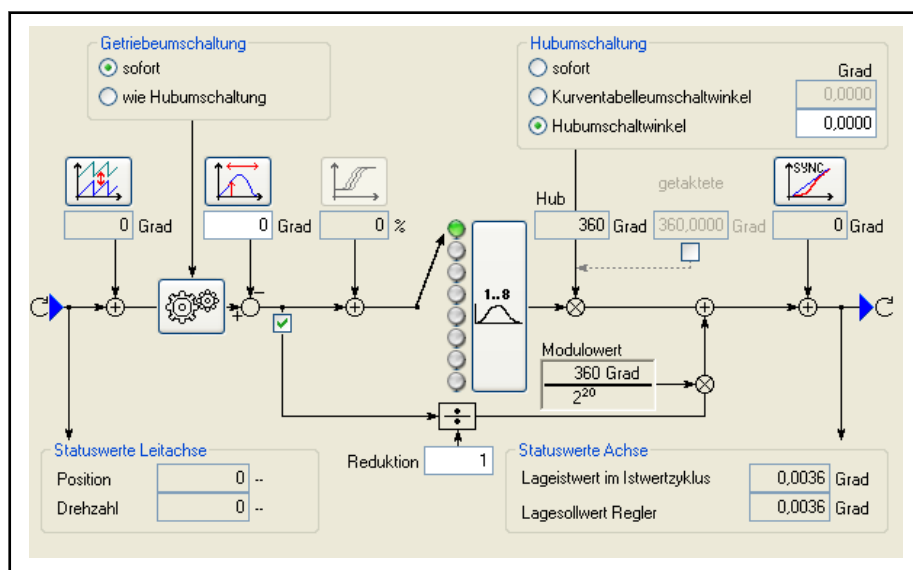


Abb. 10-2: Aktivierung des Linearpfades

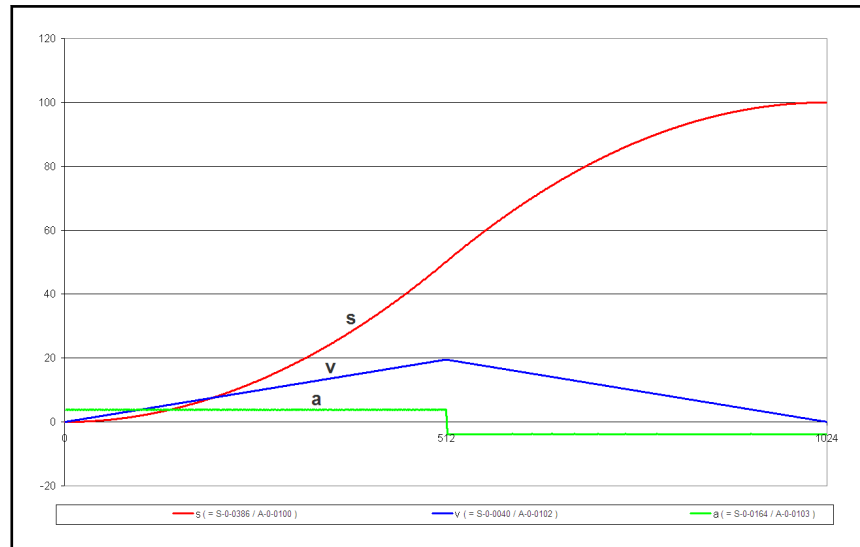
Die Struktur [MB_CAMTABLE](#), Seite 102, besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

Funktionsbeschreibung

Mit der Vorgabe der Koordinatenpaare berechnet der Funktionsbaustein MB_CoordinateCamPairType01 nach der Bearbeitungsfreigabe mittels "Execute" einmalig auf Basis der Eingangsgrößen eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen.

Hierbei muss die Anzahl der Stützpunktelemente einer Kurvenscheibentabelle durch das Element "NumberOfElements" der Struktur "CamTable" eingestellt werden.

- Select: COORDINATE_PAIR_POS_VEL_TYPE01
 - Ist der Stützpunkt (0; 0) kein vorgegebenes Koordinatenpaar, wird dieser Punkt automatisch bei der Berechnung der Kurvenscheibentabelle berücksichtigt.



Koordinatenpaare: (180;100)

Abb. 10-3: MB_CoordinatePairCamType01 Bewegungsprofil COORDINATE_PAIR_POS_VEL_TYPE01

- Select: COORDINATE_PAIR_POS_VEL_TYPE02
 - Zur Berechnung der Kurvenscheibentabelle müssen mindestens zwei Koordinatenpaare vorliegen.
 - Durch den Faktor Shape, der zur Begrenzung des Rucks vorgesehen ist, erfolgt ein Verschleifen der Geschwindigkeit.
 - Die berechnete Kurvenscheibe ist lediglich als Endloskurvenscheibe verwendbar, wenn der erste Stützpunkt (0; 0) und der letzte Stützpunkt (360; 0) beträgt. Ist das nicht der Fall, kommt es bei den Übergängen der Kurvenscheiben zu Unstetigkeiten in der Position.
 - Werden die oben genannten Stützpunkte (0; 0 und 360; 0) eingefügt und der Faktor Shape auf 0% gesetzt, ist die erstellte Kurvenscheibe mit der durch die Auswahl COORDINATE_PAIR_POS_VEL_TYPE01 erzeugten Kurvenscheibe identisch.

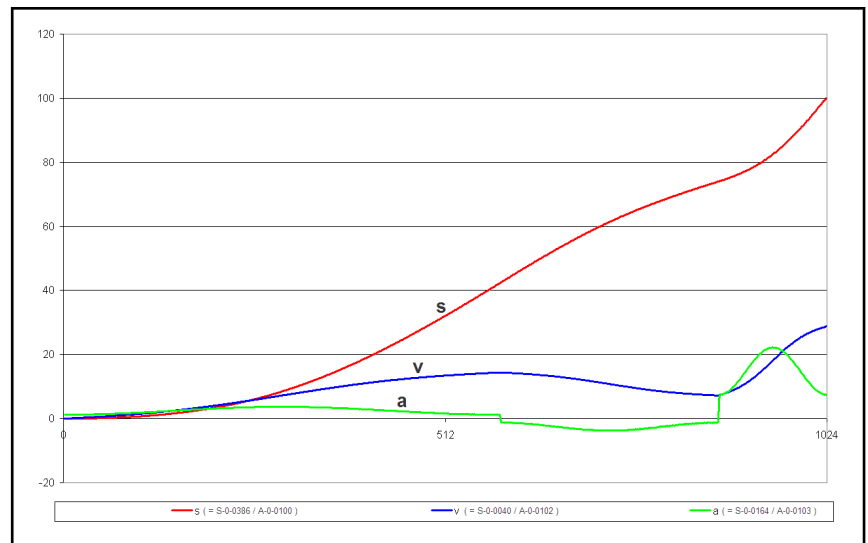


Abb. 10-4: MB_CoordinatePairCamType01 Bewegungsprofil COORDINATE_PAIR_POS_VEL_TYPE02

- Select: COORDINATE_PAIR_POS_ACC
 - Diese Auswahl ist einzusetzen, wenn das gewünschte Kurvenscheibenprofil durch einzelne Stützpunkte vorgegeben ist, die mittels einer Rast-in-Rast Bewegung abgefahren werden sollen. Dabei werden die Zielpunkte mit einem Polynom 5. Grades verbunden. Dies hat zur Folge, dass jeder Zielpunkt aus den Eingabedaten mit einer Beschleunigungs-/Bremsbewegung angefahren wird, d.h. an jedem Zielpunkt die Geschwindigkeit = 0 ist.
 - Zur Berechnung der Kurvenscheibentabelle müssen mindestens zwei Koordinatenpaare vorliegen.
 - Durch den Faktor Shape, der zur Begrenzung des Rucks vorgesehen ist, erfolgt ein Verschleifen der Geschwindigkeit.

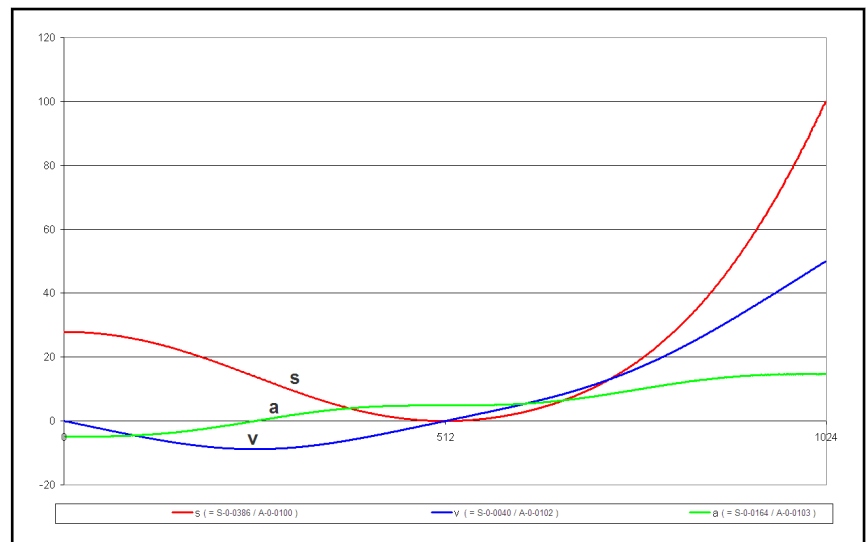
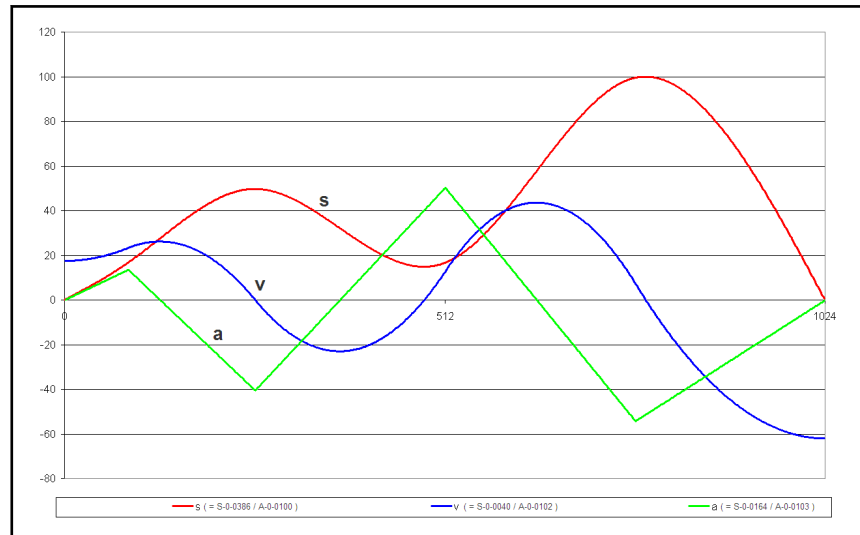


Abb. 10-5: MB_CoordinatePairCamType01 Bewegungsprofil COORDINATE_PAIR_POS_ACC

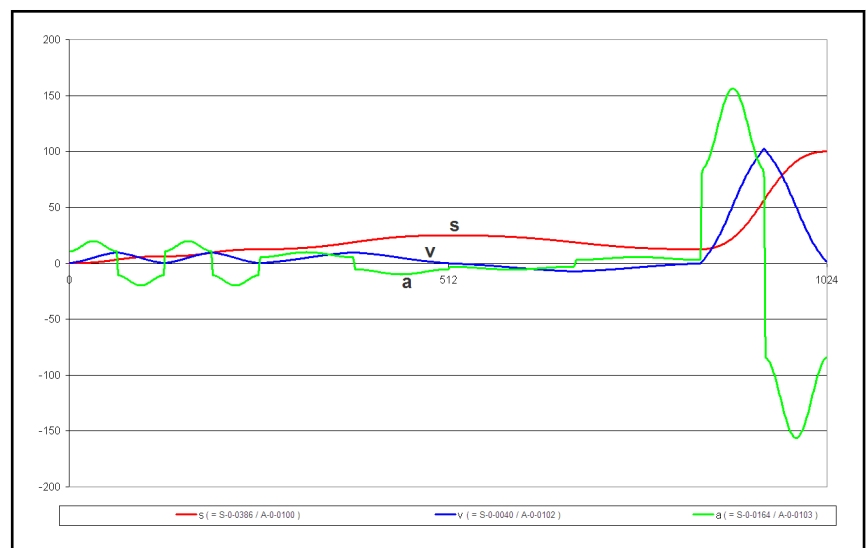
- Select: COORDINATE_PAIR_POS_POS_TYPE01
 - Zur Berechnung der Kurvenscheibentabelle müssen mindestens zwei Koordinatenpaare vorliegen.
 - Es werden die einzelnen Wertepaare ohne Rast durchfahren. Dabei werden die Zielpositionen mit einem Spline-Polynom 3. Grades verbunden.



Koordinatenpaare: (0;0),(30;10),(90;30),(180;10),(270;60),
(360;0)

Abb. 10-6: MB_CoordinatePairCamType01 Bewegungsprofil COORDINATE_PAIR_POS_POS_TYPE01

- Select: COORDINATE_PAIR_POS_POS_TYPE02
 - Diese Auswahl ist einzusetzen, wenn das gewünschte Kurvenscheibenprofil durch einzelne Stützpunkte vorgegeben ist, die mittels einer Rast-in-Rast Bewegung abgefahren werden sollen. Dabei werden die Zielpunkte mit einem Polynom 5. Grades verbunden. Dies hat zur Folge, daß jeder Zielpunkt aus den Eingabedaten mit einer Beschleunigungs-/Bremsbewegung angefahren wird, d.h. an jedem Zielpunkt die Geschwindigkeit = 0 ist.
 - Zur Berechnung der Kurvenscheibentabelle müssen mindestens zwei Koordinatenpaare vorliegen.
 - Durch den Faktor Shape, der zur Begrenzung des Rucks vorgesehen ist, erfolgt ein Verschleifen der Geschwindigkeit.



Koordinatenpaare: (0;0),(45;50),(90;100),(180;200),
(300;100),(360;800); Shape = 0,3

Abb. 10-7: MB_CoordinatePairCamType01 Bewegungsprofil COORDINATE_PAIR_POS_POS_TYPE02

Herunterladen der Kurvenscheibe in die Steuerung bzw. den Antrieb

Die zu berechnende Kurvenscheibentabelle wird in der Struktur "CamTable" mit Wichtung in Prozent abgelegt. Die Gültigkeit der Kurvenscheibenelemente wird durch den Ausgang "Done" signalisiert. Die Kurvenscheibentabelle kann mittels des FB [MB_LoadCamData](#), Seite 99, in eine Kurvenscheibe in der Steuerung oder in den Antrieb geladen werden.

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins MB_CoordinatePairCamType01 weist folgende Eigenschaften auf:

- Der Algorithmus zur Berechnung der Kurvenscheibenstützpunkte ist auf mehrere Zyklen, d. h. Aufrufe der Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden.
- Select: COORDINATE_PAIR_POS_VEL_TYPE01
 - Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist stetig bis zu Geschwindigkeit (s, v) und weist einen Sprung in der Beschleunigung (a) auf
- Select: COORDINATE_PAIR_POS_VEL_TYPE02
 - Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist in Abhängigkeit des Faktors Shape stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf
- Select: COORDINATE_PAIR_POS_ACC
 - Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist in Abhängigkeit des Faktors Shape stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf
- Select: COORDINATE_PAIR_POS_POS_TYPE01
 - Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf
- Select: COORDINATE_PAIR_POS_POS_TYPE02

- Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist in Abhängigkeit des Faktors Shape stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" werden im Fehlerfall nicht aktualisiert.

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
CALCULATION_ERROR	16#000C	16#0000	Division durch Null
INPUT_RANGE_ERROR	16#2200	16#0001	Eingang "NumberOfElements" ist < Minimum oder > Maximum
INPUT_RANGE_ERROR	16#2270	16#0001	Eingang "NoOfPairs" ist < Minimum oder > Maximum
INPUT_RANGE_ERROR	16#2270	16#0002	Eingang "Shape" ist < Minimum oder > Maximum
INPUT_RANGE_ERROR	16#2270	16#0003	Eingangselement von "Table" ist < Minimum oder > Maximum
INPUT_INVALID_ERROR	16#2201	16#0001	Adresse von Feld "CamTable" ist null
INPUT_INVALID_ERROR	16#2271	16#0001	Eingangselement von "Table" ist < vorheriges Eingangselement

Tab. 10-5: Fehlercodes FB MB_CoordinatePairCamType01

10.3 MB_COORDINATE_PAIR_CAM_SELECT

Die Enumeration "MB_COORDINATE_PAIR_CAM_SELECT" wird verwendet, um zwischen verschiedenen Typen (Position, Geschwindigkeit u. Beschleunigung) der Koordinatenpaare zu wählen.

Element	Wert	Beschreibung
COORDINATE_PAIR_POS_VEL_TYPE01	0	Koordinatenpaare aus Position (X) und Geschwindigkeit (Y)
COORDINATE_PAIR_POS_VEL_TYPE02	1	Koordinatenpaare aus Position (X) und Geschwindigkeit (Y) und Beachtung des Eingangs-Shape
COORDINATE_PAIR_POS_ACC	2	Koordinatenpaare aus Position (X) und Beschleunigung (Y) und Beachtung des Eingangs-Shape
COORDINATE_PAIR_POS_POS_TYPE01	3	Koordinatenpaare aus Position (X) und Position (Y)
COORDINATE_PAIR_POS_POS_TYPE02	4	Koordinatenpaare aus Position (X) und Position (Y) und Beachtung des Eingangs-Shape

Tab. 10-6: Aufzählungstyp MB_COORDINATE_PAIR_CAM_SELECT

10.4 MB_COORDINATE_PAIR_TYPE01

Die Struktur MB_COORDINATE_PAIR_TYPE01 enthält das Koordinatenpaar (X, Y) eines Punktes.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme	Beschreibung
X	LREAL	0.0	=<360.0	0.0	Steigende Flanke an Execute	X-Koordinate des Koordinatenpaares (Winkel)
Y	LREAL	n.def.	n.def.	0.0	Steigende Flanke an Execute	Y-Koordinate des Koordinatenpaares (Position, Geschwindigkeit oder Beschleunigung)

Tab. 10-7: Struktur MB_COORDINATE_PAIR_TYPE01 mit Min-, Default- und Max-Wert

10.5 MB_FeedMovementCalcType01

Kurzbeschreibung

Der Funktionsbaustein berechnet eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen für eine Vorschubbewegung.

Zuordnung Zielsystem/Bibliothek

MLC 14VRS Funktionspaket TEC	RMB_TechCam

Tab. 10-8: Referenztable MB_FeedMovementCalcType01

Schnittstellenbeschreibung

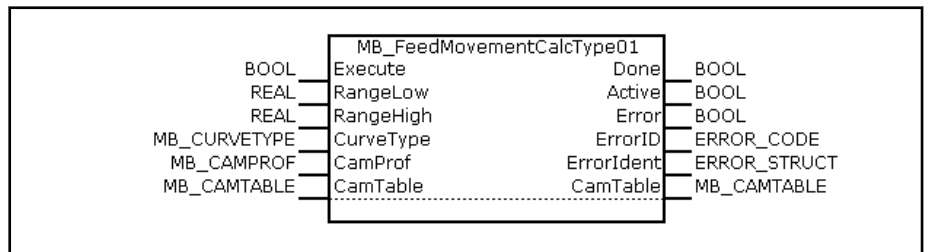


Abb. 10-8: Funktionsbaustein MB_FeedMovementCalcType01

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	RangeLow	REAL	Anfangspunkt der Vorschubbewegung; erster Punkt des Bewegungsprofils
	RangeHigh	REAL	Endpunkt der Vorschubbewegung; letzter Punkt des Bewegungsprofils
	CurveType	MB_CURVETYPE	Auswahl des Kurventyps MB_CURVETYPE , Seite 679
	CamProf	MB_CAMPROF	Auswahl des Kurvenscheibenprofils

E/A-Typ	Name	Typ	Kommentar
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe Fehlerbehandlung , Seite 673)
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellen-Elementen und Information über die Anzahl der Elemente

Tab. 10-9: Schnittstellenvariablen FB MB_FeedMovementCalcType01

Min-/Max- und Default-Werte

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
RangeLow	REAL	0.0	<360.0	0.0	Steigende Flanke an "Execute"
RangeHigh	REAL	>0.0	360.0	360.0	Steigende Flanke an "Execute"
CurveType	MB_CURVETYPE			CURVE-TYPE_POLY_5	Steigende Flanke an "Execute"
CamProf	MB_CAMPROF			CAM_PROF_STANDARD	Steigende Flanke an "Execute"

Tab. 10-10: Min-/Max- und Default-Werte FB MB_FeedMovementCalcType01

Die Enumeration [MB_CAMPROF](#), Seite 102, wird verwendet, um zwischen verschiedenen Kurvenscheibenprofilen zu wählen.



Bei der Auswahl des Kurvenscheibenprofils "CAMPROF_STANDARD" muss der Linearpfad deaktiviert sein, da es sich um ein Profil ohne Untersetzung handelt. Während bei den Profilen "CAMPROF_ADDITIVE" und "CAMPROF_ADDITIVE_SEGMENT" der Linearpfad über ein Häkchen aktiviert werden muss und die Reduktion auf den Wert "1" zu stellen ist, da es sich um Profile mit Untersetzung handelt.

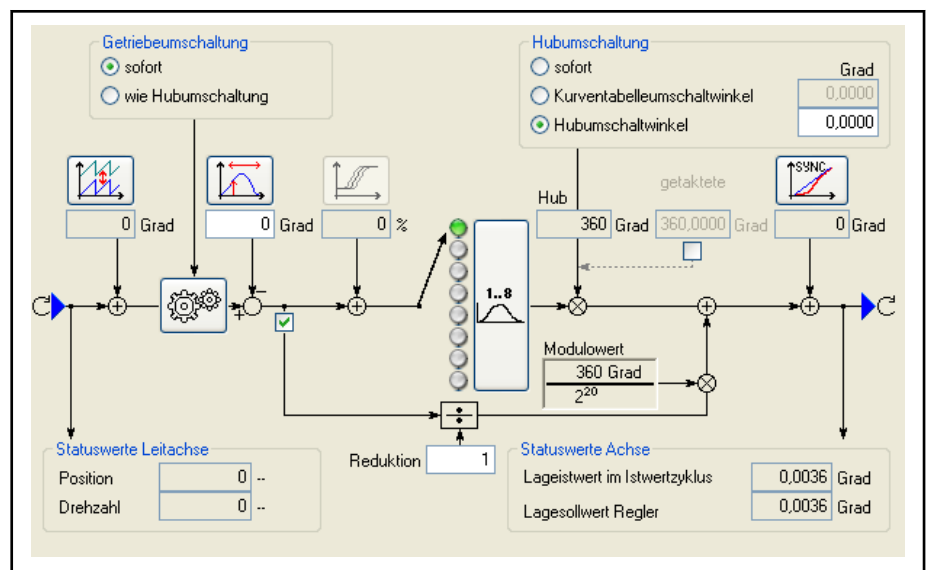


Abb. 10-9: *Aktivierung des Linearpfades*

Die Struktur [MB_CAMTABLE, Seite 102](#), besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

Funktionsbeschreibung

Mit der Vorgabe eines durch die Grenzen "RangeLow" und "RangeHigh" definierten Vorschubbereiches berechnet der Funktionsbaustein MB_FeedMovementCalcType01 nach der Bearbeitungsfreigabe mittels "Execute" einmalig auf Basis der Eingangsgrößen eine Kurvenscheibentabelle mit bis zu 1024 Stützpunktelementen. Hierbei wird die Anzahl der Stützpunktelemente einer Kurvenscheibentabelle durch das Element "NumberOfElements" der Struktur "CamTable" eingestellt.

Die Bewegung von Rast in Rast ist mit verschiedenen Ausgleichsbewegungen möglich. Die einstellbaren Bewegungen sind:

- Quadratische Parabel
- Einfache Sinuslinie
- Polynom 5. Grades
- Geneigte Sinuslinie
- Modifiziertes Beschleunigungstrapez
- Modifizierte Sinuslinie

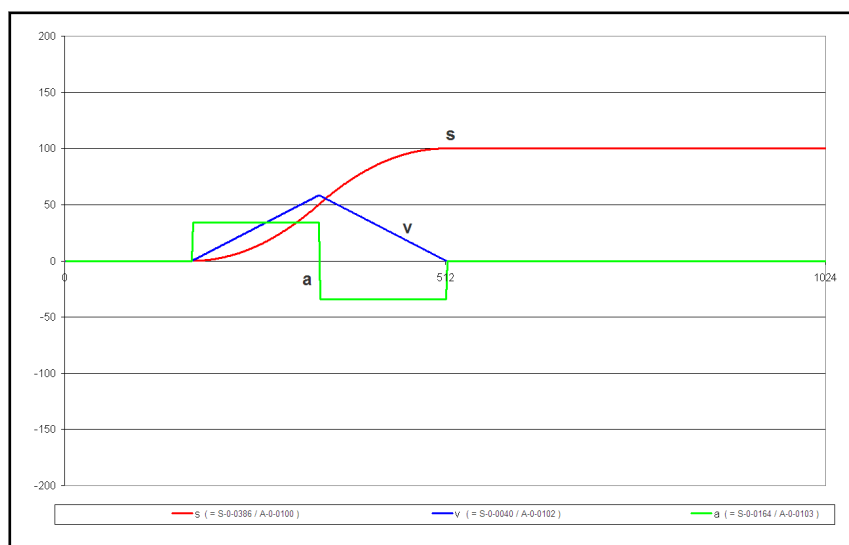
Die zu berechnende Kurvenscheibentabelle wird in der Struktur "CamTable" mit Wichtung in Prozent abgelegt. Die Gültigkeit der Kurvenscheibenelemente wird durch den Ausgang "Done" signalisiert. Die Kurvenscheibentabelle kann mittels des FB [MB_LoadCamData](#), [Seite 99](#), in eine Kurvenscheibe in der Steuerung oder in den Antrieb geladen werden.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" werden im Fehlerfall nicht aktualisiert.

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins MB_FeedMovementCalc-Type01 weist folgende Eigenschaften auf:

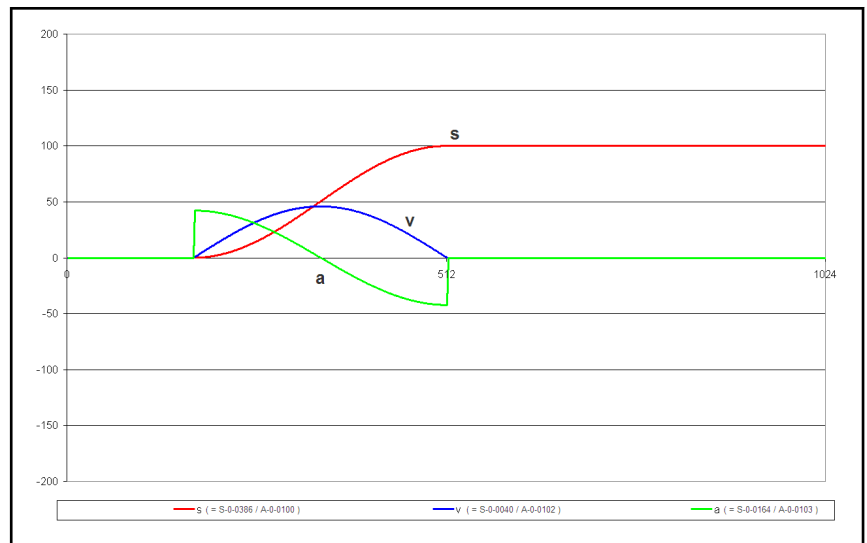
- Die vom Funktionsbaustein erzeugte Kurvenscheibentabelle beschreibt eine Rast-in-Rast-Bewegung
- Der Algorithmus zur Berechnung der Kurvenscheibenstützpunkte ist abhängig von der Differenz (RangeHigh-RangeLow) auf max 1026 Zyklen, d. h. Aufrufe der Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden
- Das Kurvenscheibenprofil ist für die zyklische Abarbeitung im Antrieb als Parameter P-0-0072, P-0-0092 etc. vorgesehen
- CurveType: CURVETYPE_QUADRATIC_PARABLE
 - Das durch die Kurvenscheibe charakterisierte Bewegungsprofil ist stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf



RangeLow = 60; RangeHigh = 180

Abb. 10-10: MB_FeedMovementCalcType01 Bewegungsprofil QUADRATIC_PARABLE

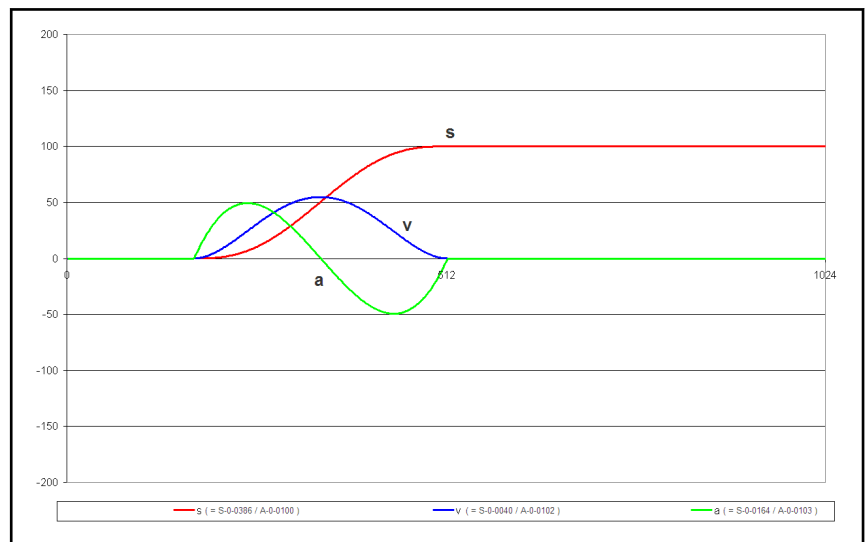
- CurveType: CURVETYPE_SIMPLE_SINUSLINE
 - Das durch die Kurvenscheibe charakterisierte Bewegungsprofil ist stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf



RangeLow = 60; RangeHigh = 180

Abb. 10-11: MB_FeedMovementCalcType01 Bewegungsprofil SIMPLE_SINUSLINE

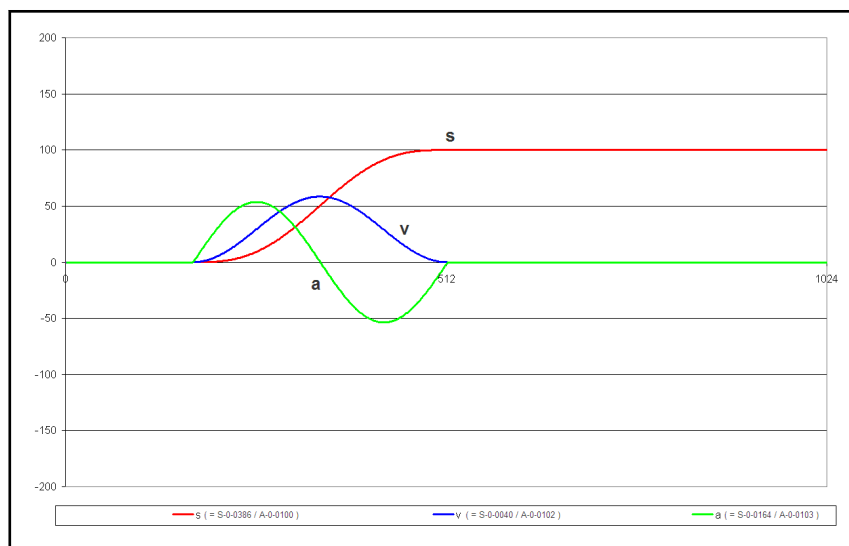
- CurveType: CURVETYPE_POLY_5
 - Das durch die Kurvenscheibe charakterisierte Bewegungsprofil ist stetig bis zum Ruck (s, v, a, j)



RangeLow = 60; RangeHigh = 180

Abb. 10-12: MB_FeedMovementCalcType01 Bewegungsprofil POLY_5

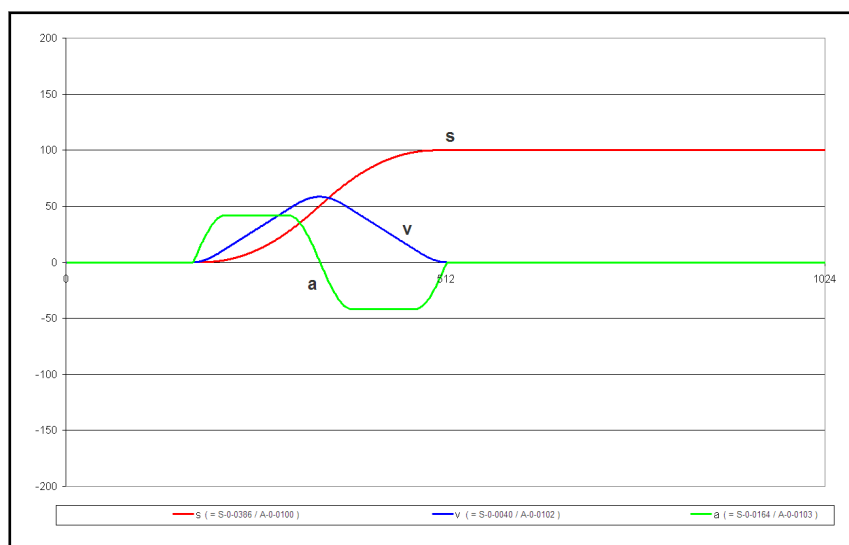
- CurveType: CURVETYPE_SLOPING_SINUSLINE
 - Das durch die Kurvenscheibe charakterisierte Bewegungsprofil ist stetig bis zum Ruck (s, v, a, j)



RangeLow = 60; RangeHigh = 180

Abb. 10-13: MB_FeedMovementCalcType01 Bewegungsprofil SLOPING_SINUSLINE

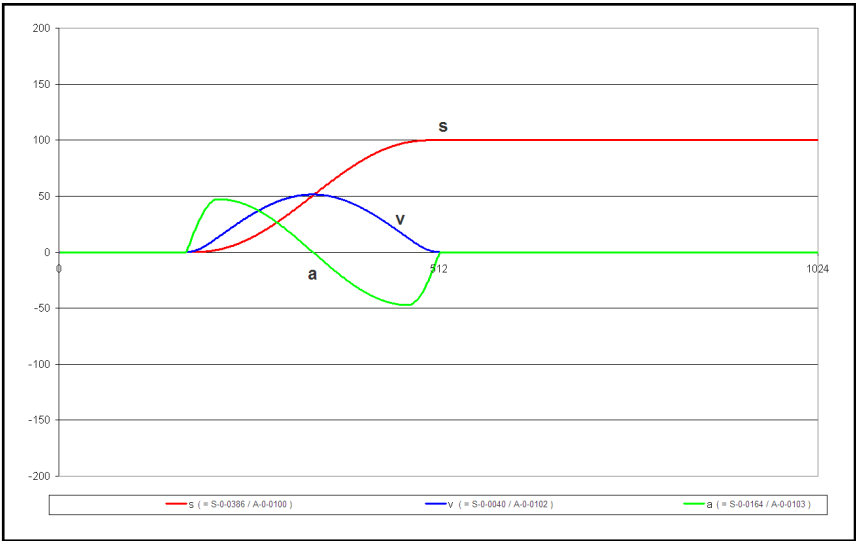
- CurveType: CURVETYPE_MODIFIED_ACCELERATIONTRAPEZE
 - Das durch die Kurvenscheibe charakterisierte Bewegungsprofil ist stetig bis zum Ruck (s, v, a, j)



RangeLow = 60; RangeHigh = 180

Abb. 10-14: MB_FeedMovementCalcType01 Bewegungsprofil MODIFIED_ACCELERATIONTRAPEZE

- CurveType: CURVETYPE_MODIFIED_SINUSLINE
 - Das durch die Kurvenscheibe charakterisierte Bewegungsprofil ist stetig bis zum Ruck (s, v, a, j)



RangeLow = 60; RangeHigh = 180

Abb. 10-15: MB_FeedMovementCalcType01 Bewegungsprofil MODIFIED_SINUSLINE

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
CALCULATION_ERROR	16#000C	16#0000	Division durch Null
INPUT_RANGE_ERROR	16#2200	16#0001	Eingang "NumberOfElements" ist < Minimum oder > Maximum
INPUT_RANGE_ERROR	16#2220	16#0001	Eingang "RangeLow" ist < Minimum oder > Maximum
INPUT_RANGE_ERROR	16#2220	16#0002	Eingang "RangeHigh" ist < Minimum oder > Maximum
INPUT_INVALID_ERROR	16#2201	16#0001	Adresse von Feld "CamTable" ist null
INPUT_INVALID_ERROR	16#2221	16#0001	Eingang "RangeLow" ist ≥ RangeHigh

Tab. 10-11: Fehlercodes FB MB_FeedMovementCalcType01

10.6 MB_FeedMovementCalcType02

Kurzbeschreibung Der Funktionsbaustein berechnet unter Benutzung eines Polynom 7. Ordnung eine Kurvenscheibentabelle mit bis zu 1024 Stützpunkten, bei der der Ruck-, Beschleunigungs- und Geschwindigkeits-Eingang begrenzt werden kann, um eine optimale Ausführung zu gewährleisten.

Zuordnung Zielsystem/Bibliothek

MLC 14VRS Funktionspaket TEC	RMB_TechCam

Tab. 10-12: Referenztable MB_FeedMovementCalcType02

Schnittstellenbeschreibung

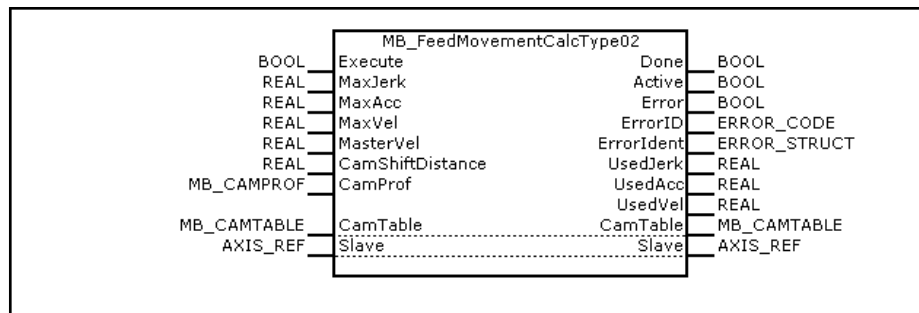


Abb. 10-16: Funktionsbaustein MB_FeedMovementCalcType02

E/A-Typ	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	MaxJerk	REAL	Maximaler Ruck
	MaxAcc	REAL	Maximale Beschleunigung
	MaxVel	REAL	Maximale Geschwindigkeit
	MasterVel	REAL	Geschwindigkeit der Leitachse
	CamShiftDistance	REAL	Länge der Kurvenscheibe
	CamProf	MB_CAMPROF	Auswahl des Kurvenscheibenprofils
VAR_OUTPUT	Done	BOOL	Bearbeitung ohne Fehler abgeschlossen, Ausgangsdaten gültig
	Active	BOOL	Bearbeitung läuft, Ausgangsdaten sind ungültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen, Ausgangsdaten sind ungültig, Fehlerausgänge sind gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine detaillierte Beschreibung des Fehlers (siehe Fehlerbehandlung , Seite 679)
	UsedJerk	REAL	Verwendeter Ruck in Relation zum Eingang "MaxJerk" in [%]
	UsedAcc	REAL	Verwendete Beschleunigung in Relation zum Eingang "MaxAcc" in [%]
	UsedVel	REAL	Verwendete Geschwindigkeit in Relation zum Eingang "MaxVel" in [%]
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellenelementen und Information über die Anzahl der Elemente
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Referenz zur Folgeachse

Tab. 10-13: Schnittstellenvariablen FB MB_FeedMovementCalcType02

Min-/Max- und Default-Werte der Eingänge

In der folgenden Tabelle sind die Min-/Max- und Defaultwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
MaxJerk	REAL	>0.0	n.def.	20.0	Steigende Flanke an "Execute"
MaxAcc	REAL	>0.0	n.def.	20.0	Steigende Flanke an "Execute"
MaxVel	REAL	>0.0	n.def.	20.0	Steigende Flanke an "Execute"
MasterVel	REAL	>0.0	n.def.	10.0	Steigende Flanke an "Execute"
CamShaftDis- tance	REAL	>0.0	n.def.	360.0	Steigende Flanke an "Execute"
CamProf	MB_CAMPROF			CAM_PROF_STA NDARD	Steigende Flanke an "Execute"

Tab. 10-14: Min-/Max-/Default-Werte FB MB_FeedMovementCalcType02

Die Enumeration **MB_CAMP**ROF, Seite 102, wird verwendet, um zwischen verschiedenen Kurvenscheibenprofilen zu wählen.



Bei der Auswahl des Kurvenscheibenprofils "CAMPROF_STANDARD" muss der Linearpfad deaktiviert sein, da es sich um ein Profil ohne Untersetzung handelt. Während bei den Profilen "CAMPROF_ADDITIVE" und "CAMPROF_ADDITIVE_SEGMENTABLE" der Linearpfad über ein Häkchen aktiviert werden muss und die Reduktion auf den Wert "1" zu stellen ist, da es sich um Profile mit Untersetzung handelt.

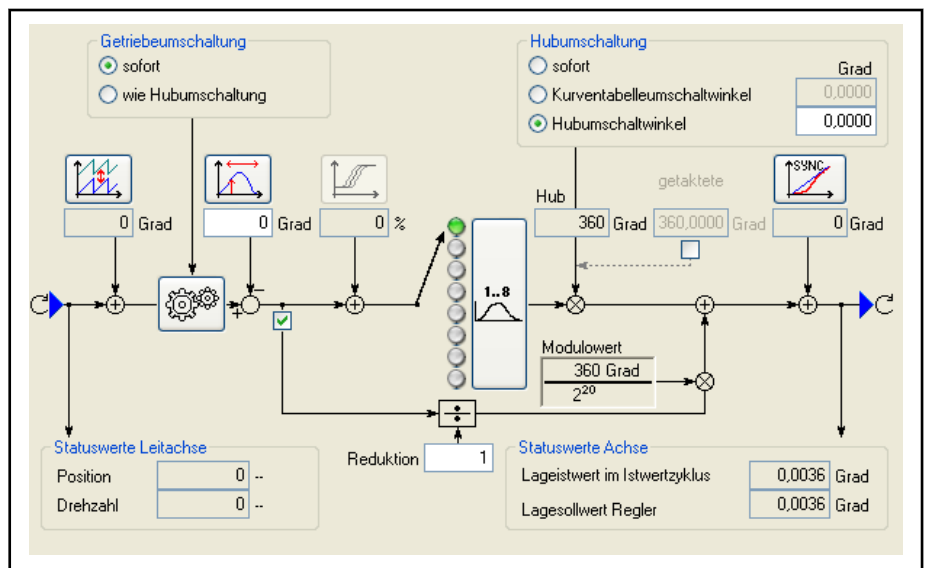


Abb. 10-17: *Aktivierung des Linearpfades*

Die Struktur **MB_CAMTABLE**, Seite 102, besitzt das Array "CamTable", welches die berechneten Kurvenscheibenpunkte enthält und die Variable "NumberOfElements", über welche die Anzahl der gültigen Kurvenscheibenpunkte des Arrays "CamTable" ersichtlich ist.

Funktionsbeschreibung

Der Funktionsbaustein MB_FeedMovementCalcType02 berechnet eine Kurvenscheibentabelle mit bis zu 1024 Stützpunkten, bei der Ruck, Beschleunigung

gung und Geschwindigkeit begrenzt werden können. Hierbei wird die Anzahl der Stützpunktelemente einer Kurvenscheibentabelle durch das Element "NumberOfElements" der Struktur "CamTable" eingestellt.

Der Anwender stellt Leitachsgeschwindigkeit, Kurvenscheibenhub, maximale Geschwindigkeit, Beschleunigung und Ruck ein. Aus diesen Vorgabewerten berechnet der Funktionsbaustein dann ein Bewegungsprofil in Form einer Kurvenscheibe, welche eine Rast in Rast Bewegung darstellt. Das Ruckprofil wird entweder aus einzelnen Parabelstücken zusammengesetzt oder als Polynom 7. Ordnung berechnet. Aus diesem Ruckprofil errechnet sich das Bewegungsprofil. Das Bewegungsprofil ermöglicht eine Endlosbewegung der Slave-Achse.

Anmerkung:

Die Variante des Funktionsbausteins für Antriebskurvenscheiben holt weitere für die Berechnung des Bausteins benötigte Informationen, wie maximaler Ruck des Antriebs oder Modulowert, aus den Parametern des betrachteten Antriebs. D.h. während der Baustein aktiv ist, werden Parameter aus dem Antrieb gelesen. Falls die Vorgabewerte möglich sind, wird eine Kurvenscheibe berechnet oder ein Fehler ausgegeben, dass die Bewegung mit diesen Vorgabewerten nicht möglich ist.

Der Modulowert des Antriebs muss dem Wert der Eingangsvariablen CamShaftDistance entsprechen, ansonsten wird der FB mit Fehler beendet. Der Parameter CamShaftDistance des Befehls MC_CamIn muss deshalb genauso groß eingestellt werden, wie der Modulowert der Slaveachse, ansonsten ist keine Endlosbewegung möglich.

Die zu berechnende Kurvenscheibentabelle wird in der Struktur "CamTable" mit Wichtung in Prozent abgelegt. Die Gültigkeit der Kurvenscheibenelemente wird durch den Ausgang "Done" signalisiert. Die Kurvenscheibentabelle kann mittels des FB [MB_LoadCamData, Seite 99](#), in eine Kurvenscheibe in der Steuerung oder in den Antrieb geladen werden.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" sowie die Ausgänge "UsedJerk", "UsedAcc" und "UsedVel" werden im Fehlerfall nicht aktualisiert.

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins MB_FeedMovementCalc-Type02 weist folgende Eigenschaften auf:

- Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist stetig bis zum Ruck (s, v, a, j)
- Der Algorithmus zur Berechnung der Kurvenscheibenstützpunkte ist auf mehrere Zyklen, d. h. Aufrufe der Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden

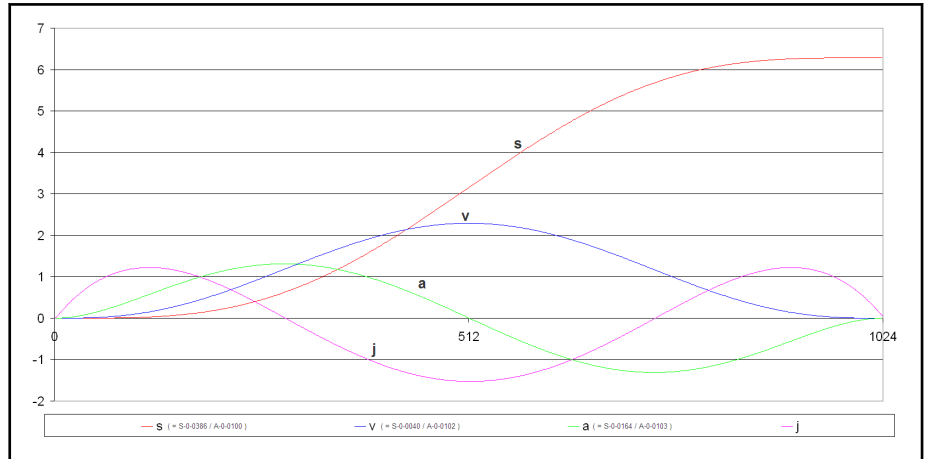
Übersicht möglicher Berechnungsverfahren

Die Umschaltung zwischen den einzelnen Varianten übernimmt der Funktionsbaustein, wobei bevorzugt eine Kurvenscheibe mit Polynom 7. Ordnung berechnet wird. Falls dies aufgrund der eingestellten Grenzwerte nicht möglich ist, wird auf die Berechnung aus Parabelstücken ausgewichen. Falls auch diese Möglichkeit die Einhaltung der Grenzwerte nicht erlaubt, wird der Ausgang "Error" des Bausteins gesetzt.

Anmerkung:

Bei der Berechnung aus Parabelstücken kann es bei ungünstigem Verhältnis aus Ruck zu Beschleunigung (hoher Ruck bei geringer Beschleunigung) zu Ungenauigkeiten kommen, da die Parabel für den Ruck sehr hoch, aber gleichzeitig sehr schmal, wird. Aufgrund der begrenzten Genauigkeit der Berechnung kann dies zu Ungenauigkeiten führen.

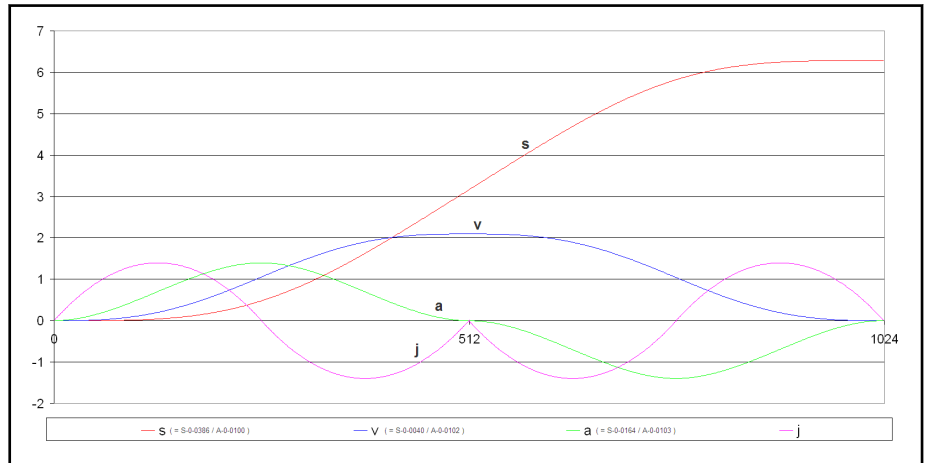
Variante 1: Polynom 7. Ordnung



MaxJerk = 10 rad/s³; MaxAcc = 10 rad/s²; MaxVel = 10 rad/s; MasterVel = 10 U/min; Modulowert = 2*Pi rad
Berechnung als Polynom 7. Ordnung

Abb. 10-18: MB_FeedMovementCalcType02 Kurvenscheibenprofil (Slave = f(Master))

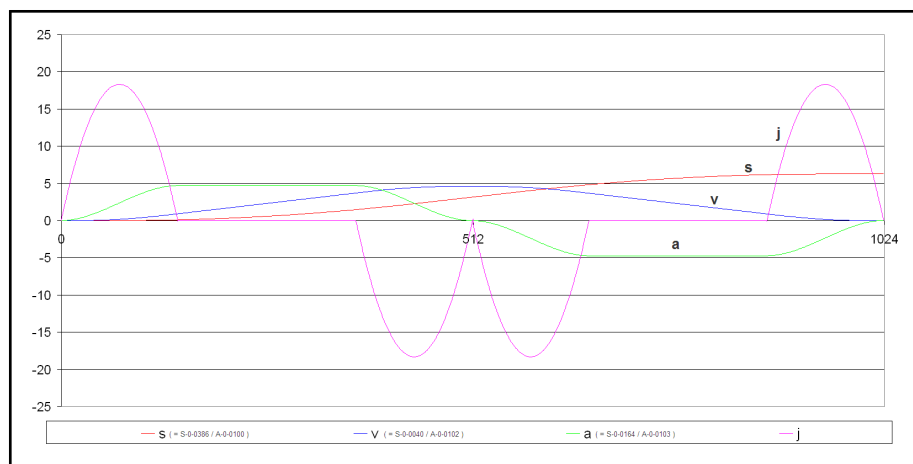
Variante 2: Geschwindigkeitsgrenzwert und Beschleunigungsgrenzwert werden nicht erreicht (Variante kommt in der Praxis nur selten vor)



MaxJerk = 10 rad/s³; MaxAcc = 10 rad/s²; MaxVel = 10 rad/s; MasterVel = 10 U/min; Modulowert = 2*Pi rad
Berechnung aus Parabelstücken

Abb. 10-19: MB_FeedMovementCalcType02 Kurvenscheibenprofil (Slave = f(Master))

Variante 3: Geschwindigkeitsgrenzwert wird nicht erreicht, Beschleunigungsgrenzwert wird erreicht (Variante kommt in der Praxis nur selten vor)

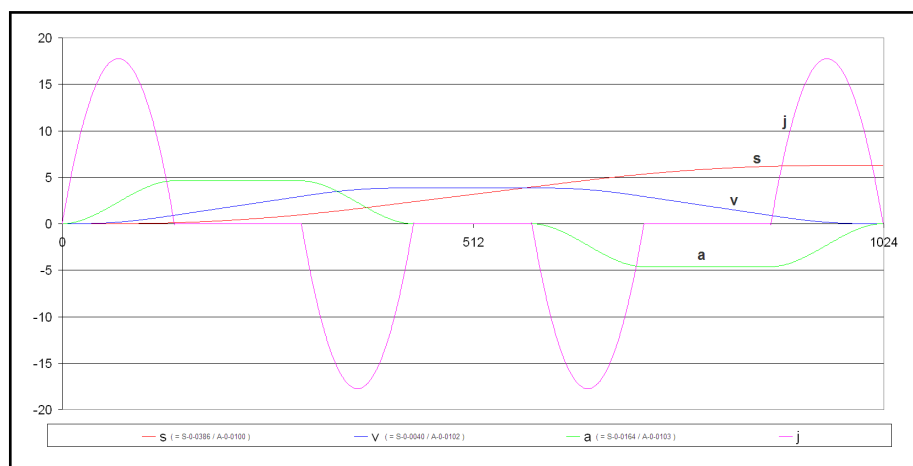


MaxJerk = 20 rad/s³; MaxAcc = 5 rad/s²; MaxVel = 5 rad/s;
MasterVel = 22 U/min; Modulowert = 2*Pi rad

Berechnung aus Parabelstücken

Abb. 10-20: MB_FeedMovementCalcType02 Kurvenscheibenprofil (Slave = f(Master))

Variante 4: Geschwindigkeitsgrenzwert und Beschleunigungsgrenzwert werden erreicht (Variante kommt in der Praxis nur selten vor)

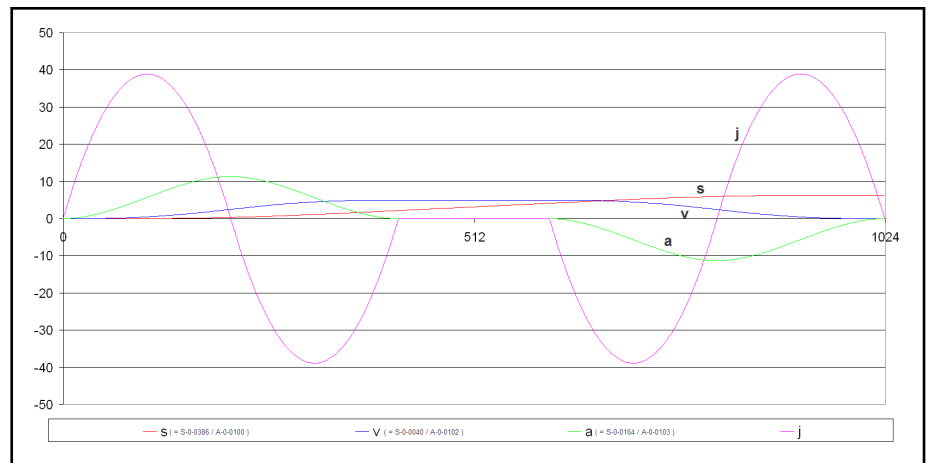


MaxJerk = 20 rad/s³; MaxAcc = 5 rad/s²; MaxVel = 4 rad/s;
MasterVel = 21 U/min; Modulowert = 2*Pi rad

Berechnung aus Parabelstücken

Abb. 10-21: MB_FeedMovementCalcType02 Kurvenscheibenprofil (Slave = f(Master))

Variante 5: Geschwindigkeitsgrenzwert wird erreicht, Beschleunigungsgrenzwert wird nicht erreicht (Variante kommt in der Praxis nur selten vor)



MaxJerk = 40 rad/s³; MaxAcc = 20 rad/s²; MaxVel = 5 rad/s;
 MasterVel = 28 U/min; Modulwert = 2*Pi rad
 Berechnung aus Parabelstücken

Abb. 10-22: MB_FeedMovementCalcType02 Kurvenscheibenprofil (Slave = f(Master))

Fehlerbehandlung Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
INPUT_RANGE_ERROR	16#2200	16#0001	Eingang "NumberOfElements" ist < Minimum oder > Maximum
INPUT_RANGE_ERROR	16#2230	16#0001	Eingang "MaxJerk" ist < Minimum
INPUT_RANGE_ERROR	16#2230	16#0002	Eingang "MaxAcc" ist < Minimum
INPUT_RANGE_ERROR	16#2230	16#0003	Eingang "MaxVel" ist < Minimum
INPUT_RANGE_ERROR	16#2230	16#0004	Eingang "MasterVel" ist < Minimum
INPUT_RANGE_ERROR	16#2230	16#0005	Eingang "CamShaftDistance" ist < Minimum
INPUT_INVALID_ERROR	16#2201	16#0001	Adresse von Feld "CamTable" ist null
INPUT_INVALID_ERROR	16#2231	16#0001	"CamShaftDistance" ungleich dem Modulwert der Achsreferenz
INPUT_INVALID_ERROR	16#2231	16#0002	Eingangswerte für Ruck, Beschleunigung oder Geschwindigkeit sind größer als in den Achsparametern voreingestellt
CALCULATION_ERROR	16#2232	16#0001	Geschwindigkeit der Leitachse zu groß oder maximaler Ruck, maximale Beschleunigung oder maximale Geschwindigkeit zu klein

Tab. 10-15: Fehlercodes FB MB_FeedMovementCalcType02

10.7 MB_CURVETYPE

Die Enumeration "MB_CURVETYPE" wird verwendet, um zwischen verschiedenen Ausgleichsbewegungen von Rast in Rast zu wählen.

Element	Wert	Beschreibung
CURVETYPE_QUADRATIC_PARABE	0	Quadratische Parabel
CURVETYPE_SIMPLE_SINUSLINE	1	Einfache Sinuslinie
CURVETYPE_POLY_5	2	Polynom 5. Grades
CURVETYPE_SLOPING_SINUSLINE	3	Geneigte Sinuslinie
CURVETYPE_MODIFIED_ACCELERATIONTRAPEZE	4	Modifiziertes Beschleunigungstrapez
CURVETYPE_MODIFIED_SINUSLINE	5	Modifizierte Sinuslinie

Tab. 10-16: Elemente des Aufzählungstyps MB_CURVETYPE

10.8 Slotter, Einführung und Definitionen

Ein Slotter ist eine rotierende Walze mit einem oder mehreren darauf angebrachten Messern, die in einer Wellpappendruckmaschine Schlitze im Wellpappebogen erstellen und dadurch die späteren Boden- und Deckellaschen einer Wellpappekiste freischneiden kann.

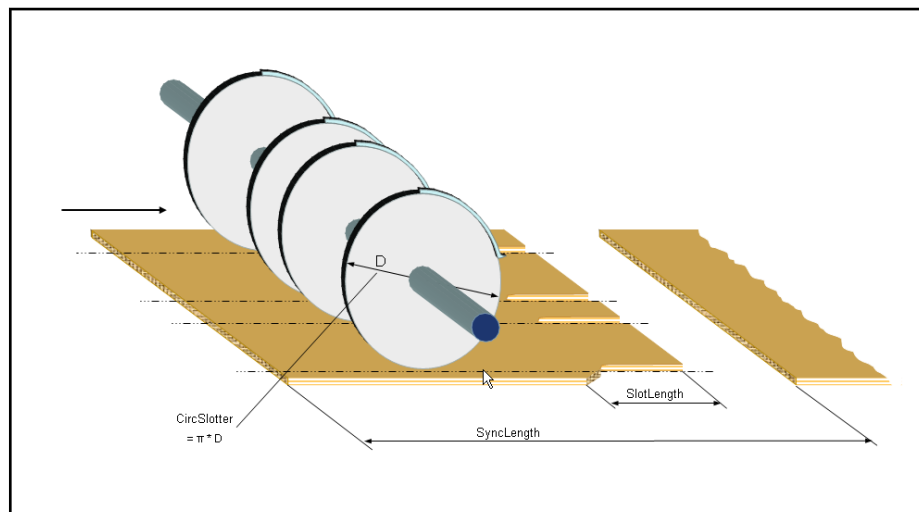


Abb. 10-23: Grafische Darstellung eines Slotters

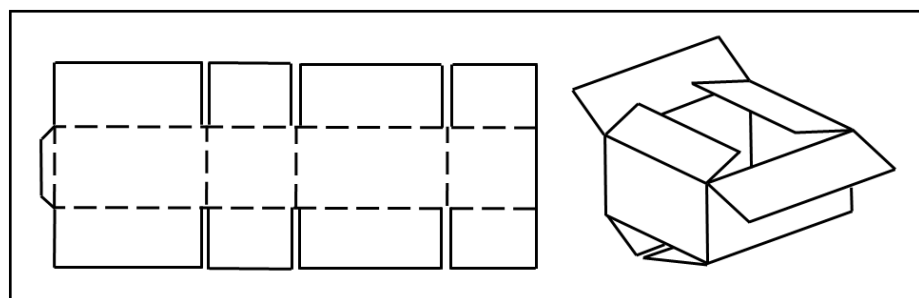


Abb. 10-24: Beispielhafte Darstellung eines geslotterten Kartons

Mehrere Slotter können in der Verarbeitungslinie hintereinander aufgestellt werden, um die Anzahl der Schlitze pro Bogen zu erhöhen oder Schlitze zu erstellen, die länger sind als das Messer einer der Slotterwalzen. Typisch sind Einzelslotter (Slotterwalze mit einem Messer) und Doppelslotter (Slotterwalze mit zwei Messern).

Nachfolgend werden Definitionen und Festlegungen beschrieben, deren Kenntnisse bei Verwendung der Funktionsbausteine [MB_SlotterCalcType01](#), Seite 685, und [MB_SlotterCalcType02](#), Seite 697, zum Berechnen von Kurvenscheiben für Slotter notwendig sind.

Drehrichtung, Bewegungsrichtung und Nullmarken

Alle hier beschriebenen Definitionen und Bezüge gehen davon aus, dass die messerbestückte Welle oberhalb der Kartontransportebene angeordnet ist.

Die korrekte Unterscheidung von Drehrichtung und Bewegungsrichtung ist für die Definition und Parametrierung der Slotter essentiell.

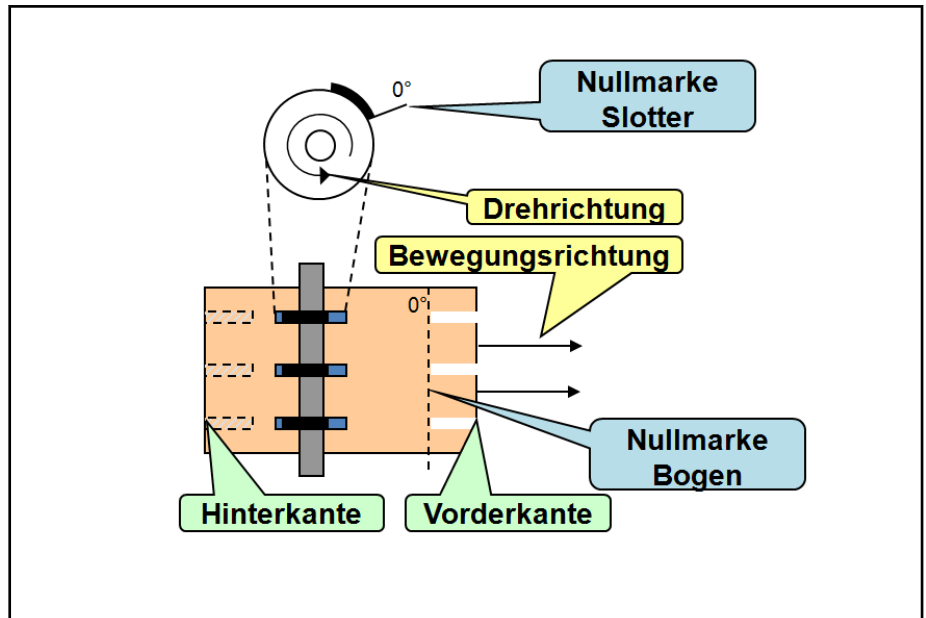


Abb. 10-25: Unterscheidung von Drehrichtung und Bewegungsrichtung

Drehrichtung

Die Drehrichtung bezieht sich auf die Drehbewegung der Slotterwalze. Die Bewegungsrichtung bezieht sich stets auf die Bewegungsrichtung des zu schlitzenden Bogens. In diesem Fall wird immer von der vorderen Kante des Bogens - also der Kante, die als erstes die Slotterwalze passiert - in Richtung der hinteren Kante des Bogens gezählt. Die nachfolgende Abbildung zeigt ein Beispiel für die richtige Nummerierung von drei Punkten auf der Slotterwalze. Punkt 1 ist demnach der erste, Punkt 3 der letzte Punkt in Drehrichtung.

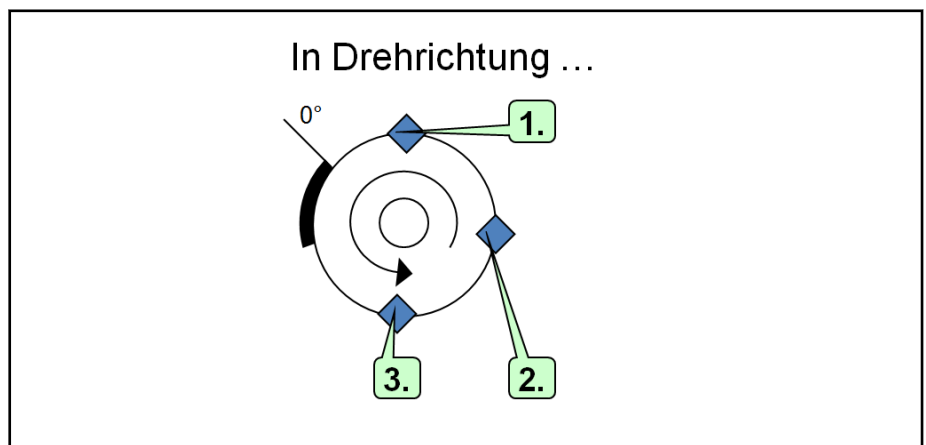


Abb. 10-26: Zählweise bei Angaben in Drehrichtung

Bewegungsrichtung

Des Weiteren ist in [Abb. 10-25 "Unterscheidung von Drehrichtung und Bewegungsrichtung"](#) auf Seite 681 dargestellt, dass sich die Bewegungsrichtung stets auf die Bewegung des zu schlitzenden Bogens bezieht. In diesem Fall wird immer von der vorderen Kante des Bogens - also der Kante, die als erstes die Slotterwalze passiert - in Richtung der hinteren Kante des Bogens gezählt. Nachfolgende Abbildung zeigt ein Beispiel für die Zählweise in Bewe-

gungsrichtung. Punkt 1 ist demnach der erste, Punkt 3 der letzte Punkt in Bewegungsrichtung.

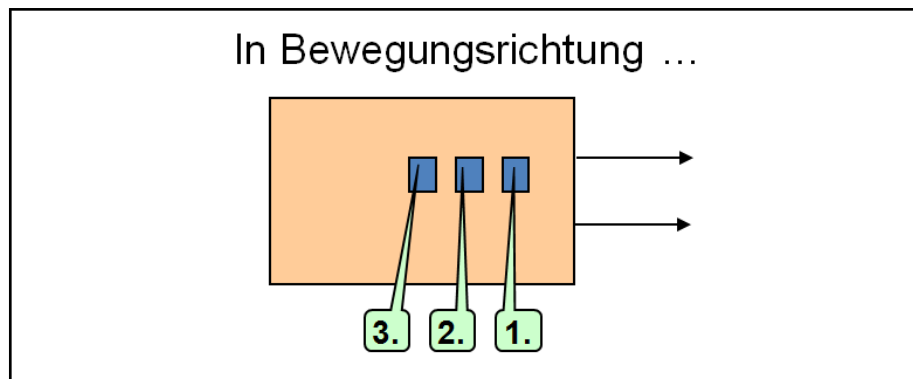


Abb. 10-27: Zählweise bei Angaben in Bewegungsrichtung

Slotnummern

Mit den hier dokumentierten Slotter-Funktionsbausteinen können ein oder zwei Schlitze pro Bogen erstellt werden.

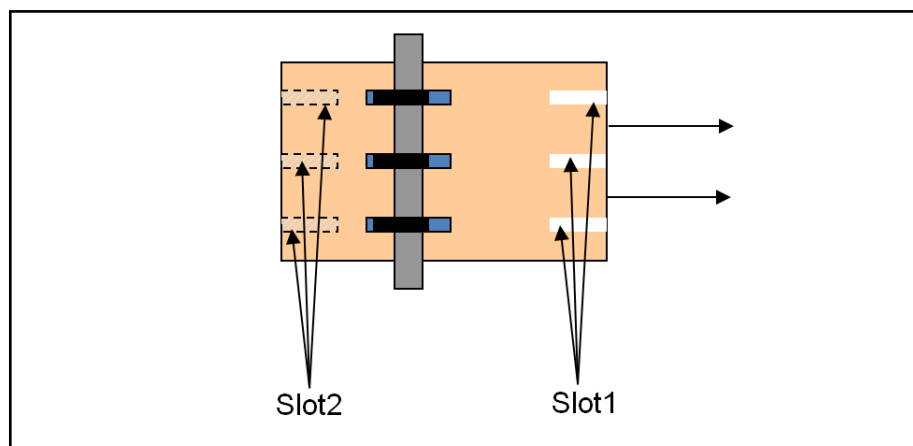
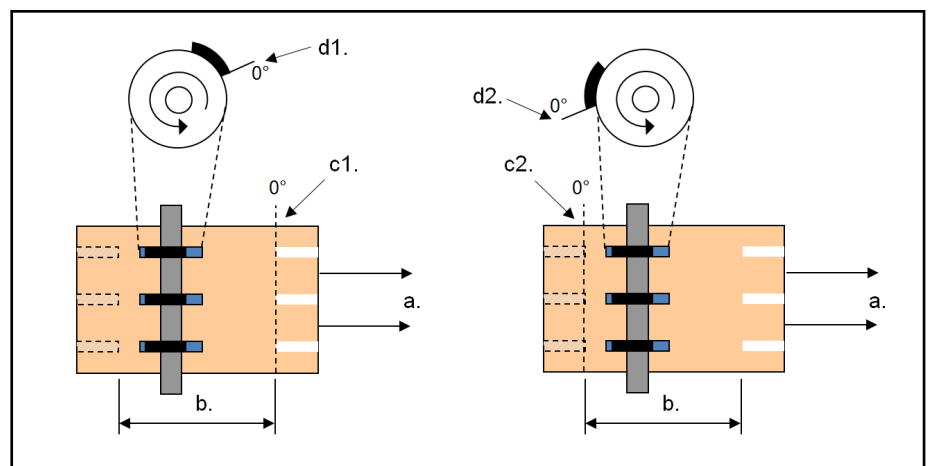


Abb. 10-28: Nummerierung der Slots bei mehreren Schlitzen in einem Bogen

Der in Bewegungsrichtung erste Schlitz wird immer mit "Slot 1" und der in Bewegungsrichtung zweite Schlitz mit "Slot 2" bezeichnet.

Kartonhöhe und Position der Nullmarken

Die Position der Nullmarke der Slotterwalze definiert die Relation zwischen Slotterwalze und zu schlitzendem Bogen. Sie entspricht der 0° Absolutposition der Slotterachse.



- a. Bewegungsrichtung
b. Kartonhöhe ("BoxHeight")
c1., c2. Mögliche Position der Nullmarke des Bogen
d1., d2. Mögliche Position der Nullmarke des Slotters

Abb. 10-29: Festlegung der Nullmarke auf der Slotterwalze und im zu schlitzenen Bogen

Die Position der Nullmarke auf dem Bogen definiert eine Kante der Kartonhöhe ("BoxHeight"). Die Kartonhöhe ist der Bereich zwischen "Slot1" und "Slot2".

Auf der Slotterwalze muss die Nullmarke immer an einem Messer anliegen. Bei Slottern mit einem Messer gibt es demnach zwei, bei Slottern mit zwei Messern auf der Messerwalze vier, mögliche Positionen der Nullmarke.

Auf dem zu schlitzenen Bogen liegt die Nullmarke per Definition immer an der späteren Kartonhöhe an Slot 1 oder Slot 2 an. In [Abb. 10-29 "Festlegung der Nullmarke auf der Slotterwalze und im zu schlitzenen Bogen" auf Seite 683](#) sind die beiden verschiedenen Möglichkeiten für die Position der Nullmarke bei einem Slotter mit einem Messer dargestellt.

Bei den Slotter-Funktionsbausteinen muss bei der Berechnung jeweils vorgegeben werden, welche Nullmarkenposition auf dem Bogen gewünscht wird (Eingang "CamProf"). Damit ist definiert, ob die Slotterwalze "Slot1" oder "Slot2" schlitzt.

Messernummern

Mit dem Slotter-Funktionsbaustein MB_SlotterCalcType02 ist es möglich, Kurvenscheiben für Slotter mit zwei Messern auf der Slotterwalze zu berechnen. Die Nullmarke der Slotterwalze, die sich stets am Anfang oder am Ende eines Messers befinden muss, definiert die Messernummer.

Das Messer, das in Drehrichtung von der Position der Nullmarke aufgehend als erstes auf der Slotterwalze angebracht ist, ist Messer 2, das andere ist Messer 1. Durch diese Festlegung schneidet Messer 1 immer Slot 1 und Messer 2 immer Slot 2.

Die nachfolgende Abbildung zeigt die Nummerierung der Messer bei allen vier möglichen Positionen der Nullmarke der Messerwalze.

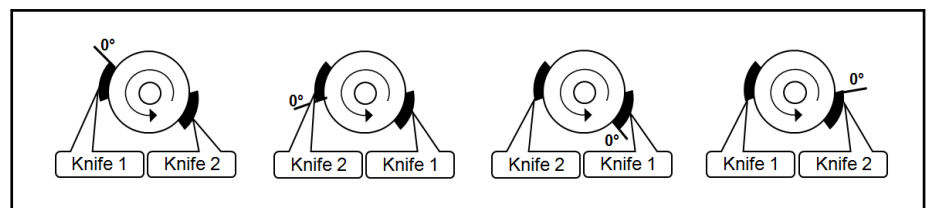


Abb. 10-30: Festlegung der Messernummern bei Slottern mit zwei Messern auf der Slotterwalze

Messerabstand Mit dem Slotter-Funktionsbaustein MB_SlotterCalcType02 ist es möglich, Kurvenscheiben für Slotter mit zwei Messern auf der Slotterwalze zu berechnen. Die Messer müssen in diesem Fall nicht symmetrisch auf der Slotterwalze angebracht sein, sondern können einen beliebigen Abstand zueinander haben.

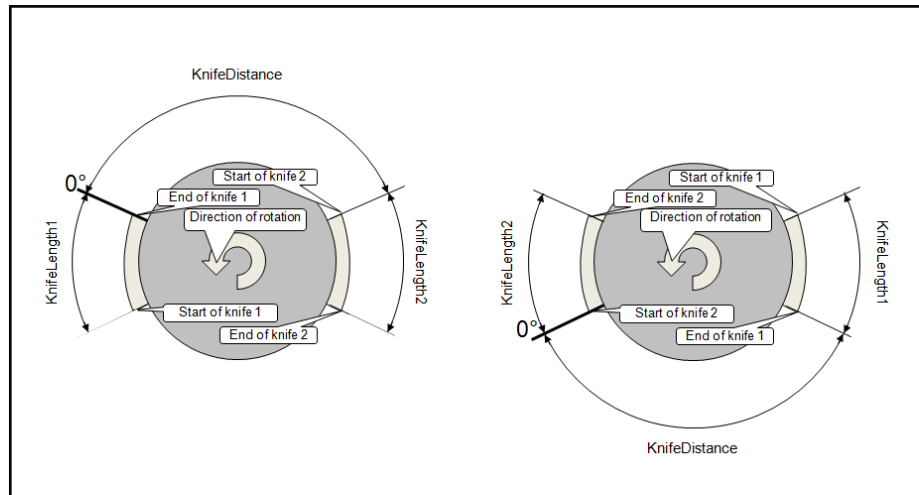


Abb. 10-31: Definition des Messerabstands bei Slottern mit zwei Messern auf der Slotterwalze

Wie in Abb. 10-31 "Definition des Messerabstands bei Slottern mit zwei Messern auf der Slotterwalze" auf Seite 684 gezeigt wird der Messerabstand immer vom Ende des Messers 1 bis zum Anfang des Messers 2 gemessen. Da die Nummerierung der Messer mit der Position der Nullmarke zusammenhängt, ist auch der Messerabstand von der Position der Nullmarke abhängig.

Multislotter Erstellen mehrere hintereinanderstehende Slotter gemeinsam einen überlangen Slot, so spricht man von Multislottern. Die im folgenden beschriebenen Definitionen und Festlegungen, die für Multislotter gelten sind in Abb. 10-32 "Definitionen und Festlegungen für Multislotter" auf Seite 685 grafisch dargestellt.

Im Gegensatz zu der Definitionsmöglichkeit der Nullmarke bei Verwendung von Einfachslottern, ist bei Multislottern die Nullmarke der einzelnen Slotter im Verbund festgelegt. Wie dargestellt, hat Slotter 1 seine Nullmarke stets in Drehrichtung am Ende des Messers, Slotter 2 bis n am Anfang.



Der Funktionsbaustein kann Falscheingaben an dieser Stelle nicht abfangen. Die Verantwortung die Nullmarken der einzelnen Slotter korrekt zu parametrieren obliegt dem Benutzer.

Zwischen den Nullmarken von Slotter 1 und Slotter n im Bogen liegt die Kartonhöhe (BoxHeight) an. Die Nullmarke von Slotter n-1 ist um $1/n$ -tel der "SlotLength2" in Richtung der hinteren Bogenkante versetzt.

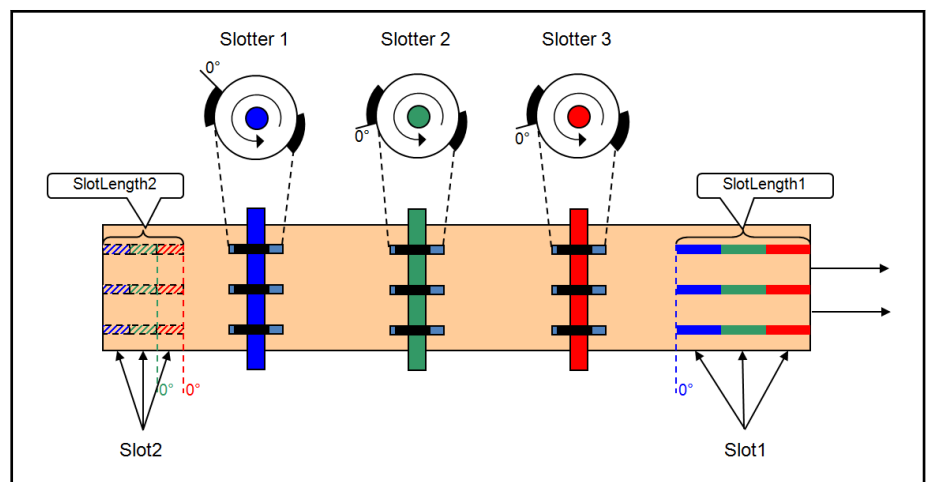


Abb. 10-32: Definitionen und Festlegungen für Multislotter

- Der Slotter, der als erstes mit dem Bogen in Berührung kommt ist Slotter 1. Der zweite Slotter 2 ... der letzte Slotter in der Reihe ist Slotter n.
- Für eine korrekte Berechnung der Kurvenscheiben, muss der Funktionsbaustein mit den unterschiedlichen Eingangsgrößen n mal aufgerufen, oder n eigene Instanzen des Funktionsbausteins angelegt, werden.
- Die von den Multislottern freigeschnittenen Schlitzte werden immer durch ihre vollständige Länge ("SlotLength1" bzw. "SlotLength2") angegeben. In keinem Fall muss die Länge eines einzelnen Teilslots (SlotLength/n) an einem Baustein angelegt werden.
- Die einzelnen Slotter teilen sich "SlotLength1" und "SlotLength2" zu gleichen Teilen auf. Jeder der n Slotter schneidet 1/n der SlotLength von Slot1 und Slot2 frei.
- Die Messer der einzelnen Slotter müssen gleich lang sein. Bei Slottern mit mehreren Messern müssen die Messer, die gemeinsam an einem Slot schneiden identische Längen besitzen.

10.9 MB_SlotterCalcType01

Kurzbeschreibung

Der Funktionsbaustein MB_SlotterCalcType01 berechnet eine Kurvenscheibe mit 1024 Stützpunktelementen für einen einzelnen Slotter (Einfachslotter) mit einem Messer auf der Messerwalze, oder für einen Slotter im Verbund von Multi-Slottern mit je einem Messer auf der Messerwalze.



In [Slotter, Einführung und Definitionen, Seite 680](#), finden Sie die Grundlage zum Verständnis für den nachfolgend beschriebenen Funktionsbaustein.

Der Funktionsbaustein MB_SlotterCalcType01 deckt folgende Funktionalitäten ab:

- Ein Schlitz pro Bogen (Einfachslotter). Ein Schlitz am Anfang oder am Ende des Bogens (entspricht MSV_CamTable14 von Rexroth SY-NAX200).
- Ein Schlitz pro Bogen im "Skip feed"-Modus mit einem Bogen über mehrere Maschinenabwicklungen.
- Ein langer Schlitz am Anfang oder am Ende des Bogens (Multislotter). Dabei arbeiten zwei oder mehrere Slotterwalzen an einem Schlitz.

- Ein langer Schlitz pro Bogen im "Skip feed"-Modus mit einem Bogen über mehrere Maschinenabwicklungen.
- Zwei Schlitz pro Bogen (Einfachslotter). Ein Schlitz am Anfang des Bogens und ein Schlitz am Ende des Bogens.
- Zwei Schlitz pro Bogen im "Skip feed"-Modus mit einem Bogen über mehrere Maschinenabwicklungen.
- Zwei lange Schlitz pro Bogen (Multislotter). Ein langer Schlitz am Anfang des Bogens und ein langer Schlitz am Ende des Bogens.
- Zwei lange Schlitz pro Bogen im "Skip feed"-Modus mit einem Bogen über mehrere Maschinenabwicklungen.
- Schlitz mit Slotterscheiben deren Umfangslänge (Messerflugkreis / "CircSlotter") kleiner ist als die Maschinenabwicklung ("SyncLength").

Der Baustein berechnet immer eine Kurvenscheibe für einen Slotter. Bei n-Slottern im Multislotterverbund muss der Baustein n-mal aufgerufen, oder n eigene Instanzen des Funktionsbaustein angelegt, werden. Für welchen Slotter der Baustein die Kurvenscheibe berechnen soll muss am Baustein über den Eingang "SlotterToCalculate" definiert werden.

Zuordnung Zielsystem/Bibliothek

MLC 14VRS	RMB_TechCam
Funktionspaket TEC	

Tab. 10-17: Referenztable MB_SlotterCalcType01

Schnittstellenbeschreibung

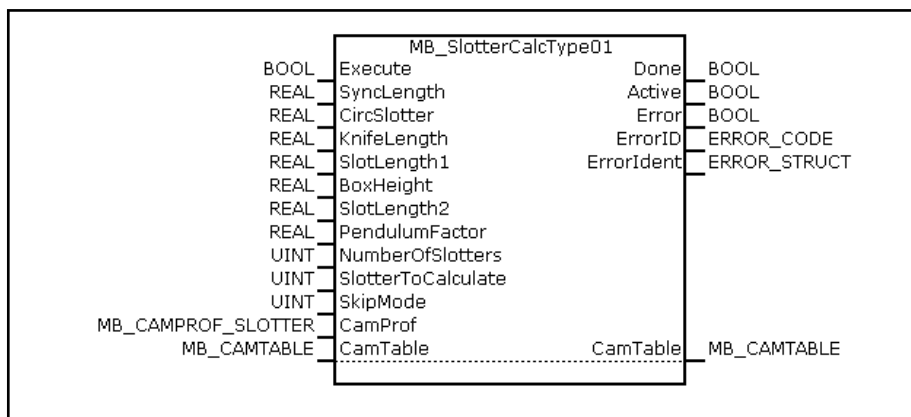


Abb. 10-33: Schnittstelle Funktionsbaustein: MB_SlotterCalcType01

	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	SyncLength	REAL	Synchrone Formatlänge, d. h. mechanische Abwicklungslänge der Maschine, z. B. Umfang des Druckzylinders [u]*
	CircSlotter	REAL	Umfang der Slotterscheibe bzw. Messerflugkreis [u]*
	KnifeLength	REAL	Länge des Messers [u]*
	SlotLength1	REAL	Gesamte Länge des in Bewegungsrichtung ersten Schlitzes im Bogen [u]*
	BoxHeight	REAL	Abstand zwischen den beiden Schlitz im Bogen ("SlotLength1" + "BoxHeight" + "SlotLength2" = Bogenlänge) [u]*

	Name	Typ	Kommentar
	SlotLength2	REAL	Gesamte Länge des in Bewegungsrichtung zweiten Schlitzes im Bogen [u]*
	PendulumFactor	REAL	Maximaler Pendelweg der Ausgleichsbewegung [%] (ohne Synchronbereich) 0%: Kein Pendeln (Stillstand bei großer Formatlänge) 100%: Maximales Pendeln (Maximale Rückwärtsdrehung bis zur Grenze an der das Messer rückwärts in den Karton schneidet)
	NumberOfSlotters	UINT	Anzahl an Slottern im Multi-Slotter-Verbund, die gemeinsam die beiden Schlitzte am Anfang und am Ende des Bogens erstellen [-]**
	SlotterToCalculate	UINT	Nummer des Slotters, für den beim Aufruf des Bausteins die Kurvenscheibe berechnet werden soll [-]**
	SkipMode	UINT	Anzahl der Maschinenzyklen pro Bogen - 1 [-]**
	CamProf	MB_CAMPROF_SLOTTER	Auswahl des Kurvenscheibenprofils und Lage der Nullmarke der Slotterwalze MB_CAMPROF_SLOTTER, Seite 708
VAR_OUTPUT	Done	BOOL	Berechnung der Kurvenscheibenelemente ohne Fehler abgeschlossen, Liste der Kurvenscheibenelemente gültig
	Active	BOOL	Berechnung der Kurvenscheibenelemente aktiv, Liste der Kurvenscheibenelemente ungültig
	Error	BOOL	Berechnung der Kurvenscheibenelemente mit Fehler abgeschlossen, Ausgangsvariablen "ErrorID" und "ErrorIdent" gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang detaillierte Informationen über den Fehler Fehlerbehandlung, Seite 696
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellen-Elementen und Information über die Anzahl der Elemente

* Die Einheit [u] bedeutet „unit“. Dies kann eine beliebige Einheit wie z. B. [mm] oder [inch] sein. Es ist darauf zu achten, dass die gewählte Einheit einheitlich verwendet wird.

** Die Einheit [-] steht für eine einheitenlose Größe.

Tab. 10-18: Schnittstellenvariablen Funktionsbaustein MB_SlotterCalcType01

Min-/Max- und Standardwerte

In der folgenden Tabelle sind die Min-/Max- und Standardwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standardwert	Übernahme
Execute	BOOL	FALSE	TRUE	FALSE	Kontinuierlich
SyncLength	REAL	>0.0	n.def.	0.0	Steigende Flanke an "Execute"
CircSlotter	REAL	>0.0	n.def.	0.0	Steigende Flanke an "Execute"
KnifeLength	REAL	>0.0	"CircSlotter"	0.0	Steigende Flanke an "Execute"
SlotLength1	REAL	>0.0	"KnifeLength" * "NumberOfSlot- ters"	0.0	Steigende Flanke an "Execute"
BoxHeight	REAL	0.0	n.def.	0.0	Steigende Flanke an "Execute" und Slot- Length2 > 0
SlotLength2	REAL	0.0	"KnifeLength" * "NumberOfSlot- ters"	0.0	Steigende Flanke an "Execute"
PendulumFactor	REAL	0.0	100.0	95.0	Steigende Flanke an "Execute"
NumberOfSlotters	UINT	1	10	1	Steigende Flanke an "Execute"
SlotterToCalculate	UINT	1	10	1	Steigende Flanke an "Execute"
SkipMode	UINT	0	10	0	Steigende Flanke an "Execute"
CamProf	MB_CAM- PROF_SLOTTER	0	3	0	Steigende Flanke an "Execute"

Tab. 10-19: Min-/Max- und Standardwerte Funktionsbaustein MB_SlotterCalc-Type01

Signal-Zeit-Diagramm

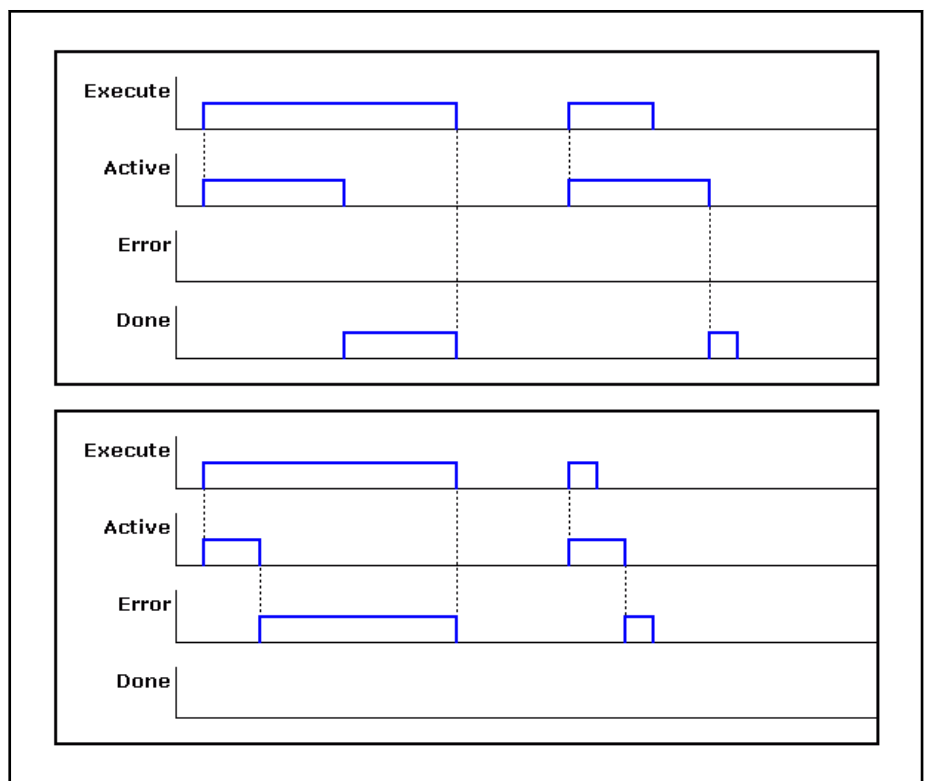


Abb. 10-34: Signal-Zeit-Diagramm Funktionsbaustein MB_SlotterCalcType01

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins "MB_SlotterCalcType01" weist folgende Eigenschaften auf:

- Das Kurvenscheibenprofil kann für die zyklische Abarbeitung als Kurvenscheibenparameter in den Antrieb (P-0-0072, P-0-0092, P-0-0780 und P-0-0781) oder in die Steuerung (C-0-2001 bis C-0-2099) geladen werden.
- Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf.
- Der Funktionsbaustein kann Profile sowohl für endlos drehende Kurvenscheiben ohne Untersetzung (P-0-0086 Bit4 = 0) als auch für Überlagerungsprofile mit Untersetzung (P-0-0086 Bit4 = 1) erzeugen.
- Die Nullmarke der Slotterwalze kann, in Drehrichtung, sowohl am Anfang wie auch am Ende des Messers liegen. Siehe [Kartenhöhe und Position der Nullmarken](#), Seite 682,
- Die Anzahl der Berechnungszyklen einer Instanz vom MB_SlotterCalcType01 richtet sich automatisch nach der aktuellen Systemauslastung, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden.
- Bei überlangen Synchronbereichen kann die Ausgleichsbewegung zu einer Rückwärtsdrehung führen. Die maximale Rückwärtsdrehung des Schlitzmessers kann vorgegeben werden.
- Bei Kurvenscheiben die zwei Schlitze pro Bogen schneiden, muss durch Parametrierung eines elektronischen Getriebes sichergestellt werden, dass beide Umdrehungen in der einen, vom Funktionsbaustein erstellten Kurvenscheibe, enthalten sind. Der Funktionsbaustein berechnet die Synchronbereiche so, dass die Slotterwalze nach dem elektronischen Getriebe während der Synchronbereiche mit Materialgeschwindigkeit dreht.

- Im SkipFeed-Modus wird davon ausgegangen, dass sich ein Produkt über mehrere Maschinenzyklen ("synchrone Formatlängen") erstreckt. SkipMode n streckt eine Arbeitsumdrehung auf n+1 Maschinenzyklen. Durch Einsatz eines elektronischen Getriebes muss sichergestellt sein, dass die vom Funktionsbaustein erstellte Kurvenscheibe auf die entsprechende Anzahl an Maschinenzyklen gestreckt wird. Der Funktionsbaustein berechnet im SkipFeed-Modus die Synchronbereiche so, dass die Slotterwalze nach dem elektronischen Getriebe während der Synchronbereiche mit Materialgeschwindigkeit dreht.

Eigenschaften im Multislotter-Verbund

Beim Betrieb von Multislottern sind folgende weitere Punkte zu beachten:

- Multislotter die einen langen Schlitz am Anfang oder am Ende des Bogens schneiden, können durch Verwendung einer gemeinsamen Kurvenscheibe ("NumberOfSlotters"=1) sowie verschobener Nullmarken auf dem Bogen realisiert werden.
- Der Slotter, der als erstes mit dem Bogen in Berührung kommt ist Slotter 1 der letzte ist Slotter n.
- Um die Slotter dynamisch optimal auszunutzen schneidet Slotter 1 zur Erstellung von Slot 1 immer zunächst ein Fenster in den Bogen. Slotter 2 bis n schneiden dann den Slot frei. Auf der anderen Seite des Bogens (Slot 2) schneidet Slotter 1 das Stück direkt an der Bogenkante und Slotter 2 bis n vergrößern den Schlitz bis zur "SlotLength2".
- Die Messer der einzelnen Slotter müssen gleich lang sein.
- Die einzelnen Slotter teilen sich "SlotLength1" und "SlotLength2" zu gleichen Teilen auf. Jeder der n Slotter schneidet 1/n-tel der SlotLength von Slot1 und Slot2 frei.
- Zwischen den Nullmarken von Slotter 1 und Slotter n im Bogen liegt die Kartonhöhe (BoxHeight) an. Die Nullmarke von Slotter n-1 ist um 1/n der "SlotLength2" in Richtung der hinteren Bogenkante versetzt.
- An den Eingängen "SlotLength1" und "SlotLength2" muss die gesamte Länge der Slots 1 und 2 angelegt werden und nicht die Länge des Teilslots, die der zu berechnende Slotter erstellt.
- Am Eingang "NumberOfSlotters" muss definiert werden, wie viele Slotter sich im Multi-Slotter-Verbund befinden und gemeinsam an den beiden Slots arbeiten.
- Am Eingang "SlotterToCalculate" muss definiert werden, für welchen der "NumberOfSlotters"-Slotter die Kurvenscheibe berechnet werden soll. Die Nummerierung der Slotter erfolgt in der Reihenfolge in der sie den Bogen berühren (Slotter 1 berührt Bogen als erstes, Slotter2 als zweites, ...).
- Da Slotter1 bei Slot1 ein Fenster in das Material schneidet, muss er für seine komplette Messerlänge synchron mit der Materialgeschwindigkeit laufen. Alle weiteren Slotter laufen nur für ihre jeweilige Teilslot-Länge synchron. Aus diesem Grund können sich Slotter 2-n dynamisch günstiger verhalten als Slotter 1.

Funktionsbeschreibung

Vorgabe ist, in einem vom Schneidprozess bestimmten Synchronbereich (vorgegeben durch "SlotLength1" und "SlotLength2") die Umfangsgeschwindigkeit der Messerwalze synchron zur Geschwindigkeit der Materialbahn zu halten. Dazu berechnet der Funktionsbaustein nach der Bearbeitungsfreigabe mittels "Execute" auf Basis der Eingangsgrößen einmalig eine Kurvenscheibentabelle mit 1024 Stützpunktelementen.

Die konkrete Bedeutung der Eingangsgrößen zeigt nachfolgende Darstellung.

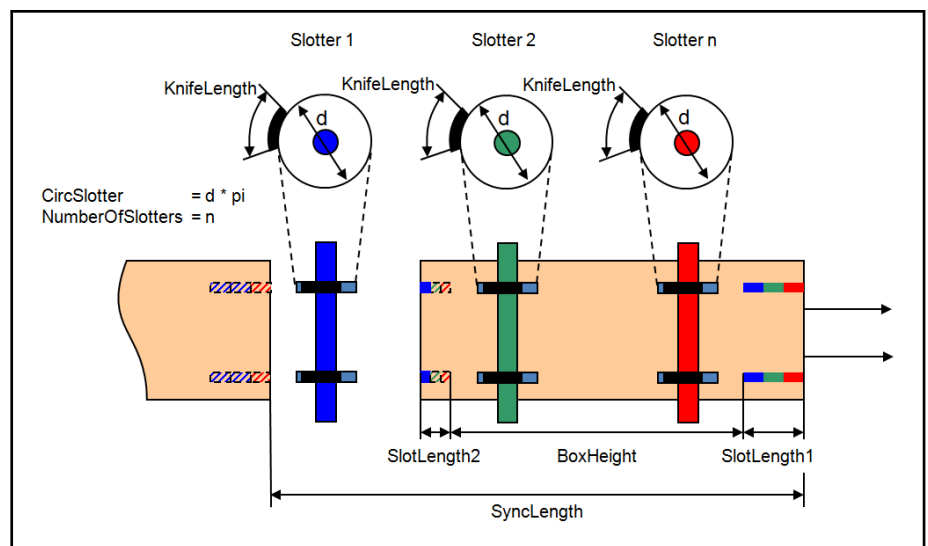


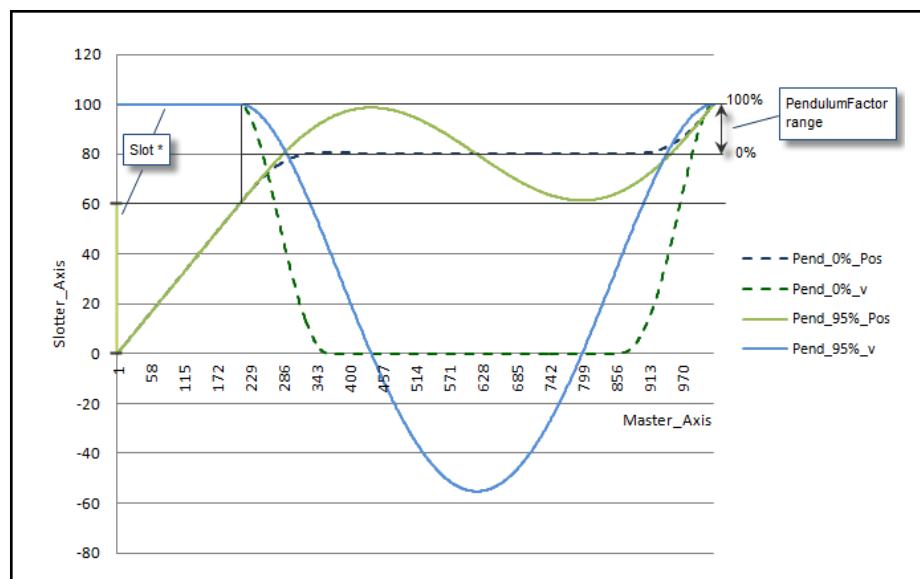
Abb. 10-35: Bedeutung der Eingangsgrößen des Funktionsbausteins MB_Slotter-CalcType01

Um Formate, die in ihrer Länge größer oder kleiner als der Umfang der Messerwalze sind, mit nur einer Schneidwalze synchron schneiden zu können, muss eine durch den Funktionsbaustein berechnete Ausgleichsbewegung stattfinden. Diese Ausgleichsbewegung wird durch eine Stetigkeit bis zur Beschleunigung charakterisiert.

PendulumFactor

In ungünstigen Verhältnissen zwischen Kartonlängen und Slotlängen sowie Umfängen der Slotterscheiben, kann es unter Umständen zu Rückwärtsdrehungen in den Ausgleichsbewegungen kommen. Diese sind durchaus gewünscht, da damit die dynamische Belastung des Motors durch Spitzenbeschleunigungen günstiger ausfällt.

In extremen Fällen kann es bei der Rückwärtsdrehung zum Eintauchen des Messers in den Karton gegen die Transportrichtung kommen. Um dieses zu verhindern steht dem Anwender der Eingang "PendulumFactor" zur Verfügung mittels dem die maximale Rückwärtsdrehung in der Ausgleichsbewegung einschränkt ($0\% < \text{"PendulumFactor"} < 100\%$) oder auch komplett unterdrückt ($\text{"PendulumFactor"} = 0\%$) werden kann. Die Ausgleichsbewegung zwischen den Synchronbereichen entspricht $\pm 100\%$ ($\text{"PendulumFactor"} = 100\%$).



Pend_0%_Pos Kurventabelle bei "PendulumFactor" = 0%

Pend_0%_v Geschwindigkeit bei "PendulumFactor" = 0%

Pend_95%_Pos Kurventabelle bei "PendulumFactor" = 95%

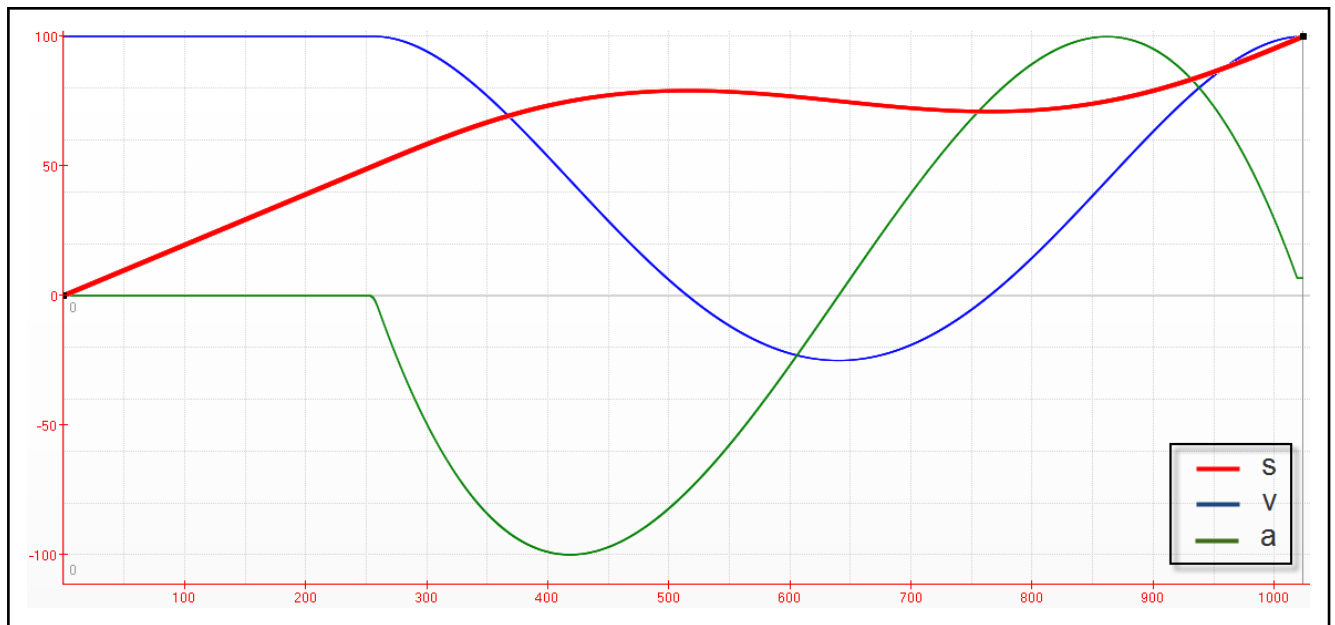
Pend_0%_v Geschwindigkeit bei "PendulumFactor" = 95%

Slot* Synchrongeschwindigkeit während des Materialeingriffs

Abb. 10-36: Erklärung des Eingangs "PendulumFactor" anhand von Kurventabellen

Funktionsvarianten Mittels verschiedener Vorbelegungen der Eingänge sind verschiedene Funktionsvarianten mit dem Funktionsbaustein MB_SlotterCalcType01 möglich:

Beispiel 1 Die erste Variante [Beispielhaftes Bewegungsprofil 1, Seite 693](#), eignet sich, um Formate mit einem Slotter zu schneiden, dessen Messerflugkreis "Circ-Slotter" kleiner ist als die mechanische Abwicklungslänge der Maschine "SyncLength". Die wirksame Messerlänge wird hierbei auf die Materialtransportgeschwindigkeit im Werkzeugeingriff angepasst. Der verbleibende Scheibenumfang des Slotters ohne Schneidmesser wird als Bereich für eine Ausgleichsbewegung definiert.

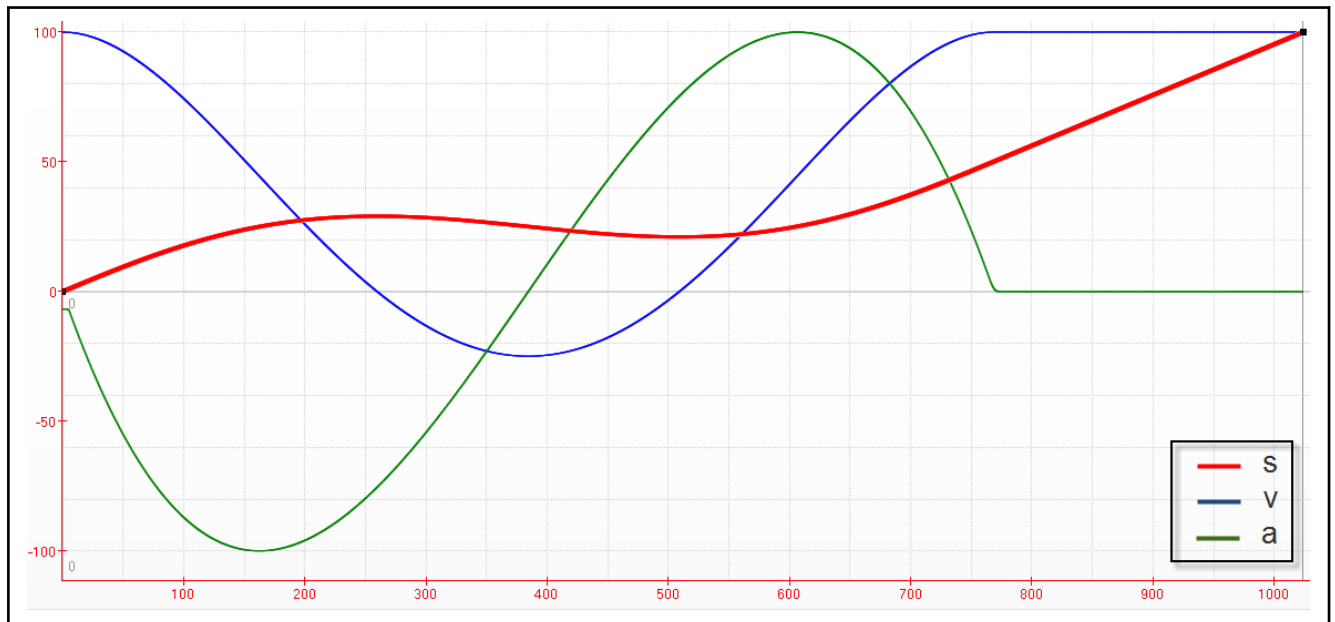


Beispiel 1: Nullmarke vor dem Messer, Ausgleichsbewegung am Ende (SyncLength = 1024; CircSlotter = 512; SlotLength1 = 256; SkipMode = 0; CamProf = 0)

Abb. 10-37: Beispielhaftes Bewegungsprofil 1

Beispiel 2

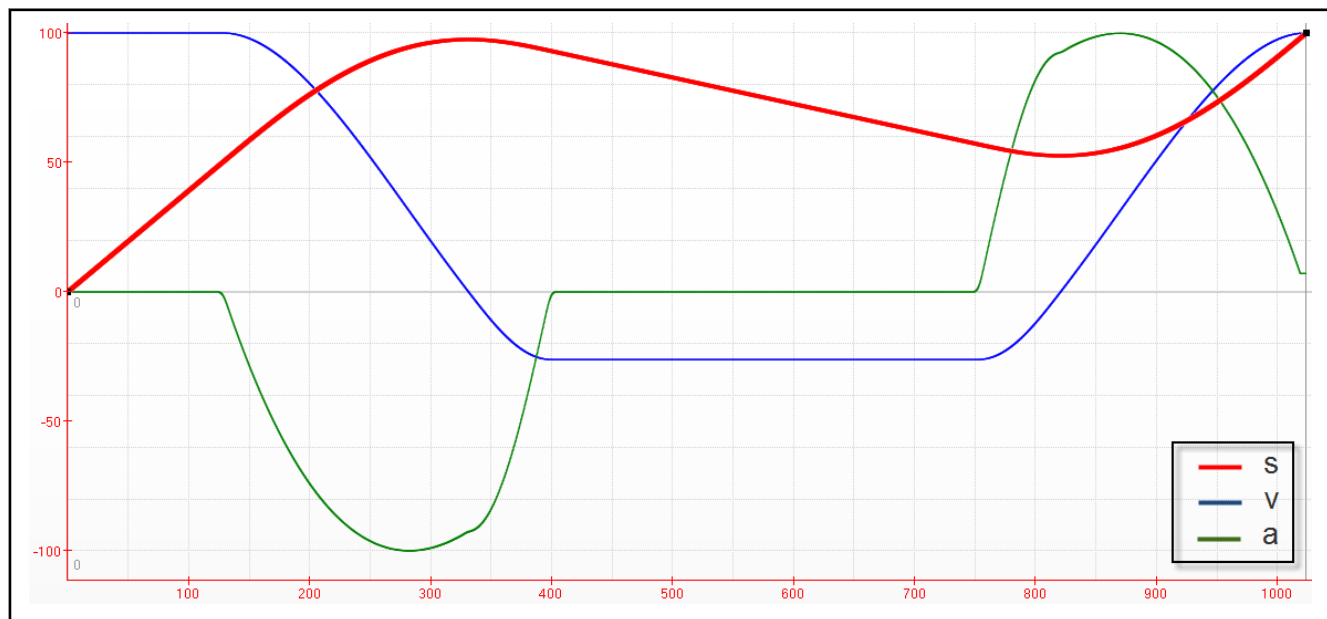
In sämtlichen Einfachslotter-Varianten "NumberOfSlotters" = 1 kann die Nullmarke der Slotterwalze durch Parametrierung des "CamProf"-Eingangs vor oder nach dem Messer definiert werden. In der nachfolgenden Variante [Abb. 10-38 "Beispielhaftes Bewegungsprofil 2" auf Seite 693](#) wurde stellvertretend die erste Variante [Abb. 10-37 "Beispielhaftes Bewegungsprofil 1" auf Seite 693](#) erneut aufgegriffen, und mit dem Eingang "CamProf" eine Ausgleichsbewegung am Anfang und die Synchronbewegung am Ende gewählt.



Beispiel 2: Nullmarke nach dem Messer, Ausgleichsbewegung am Anfang (SyncLength = 1024; CircSlotter = 512; SlotLength1 = 256; SkipMode = 0; CamProf = 2)

Abb. 10-38: Beispielhaftes Bewegungsprofil 2

Beispiel 3 Die dritte Variante [Abb. 10-39 "Beispielhaftes Bewegungsprofil 3"](#) auf [Seite 694](#) eignet sich, um überlange Formate, d. h. welche größer als die maximale synchrone Formatlänge sind, verarbeiten zu können (im "SkipMode" üblich). Dabei wird der Schlitzprozess auf mehrere Maschinenzyklen gestreckt, wobei der Schneidbereich des Messers synchron zur Oberflächegeschwindigkeit bleibt ("SkipMode" = n). Der Slotter dreht sich mit einer mechanischen Umdrehung während n+1 Maschinenzyklen. Während der Bereich mit der wirksamen Messerlänge ("SlotLength1") synchron zum Material fährt, wird in der restlichen Bewegung bis zum Vollenden des n+1 Zyklusses eine Ausgleichsbewegung ausgeführt. Nachfolgend ist des Weiteren die Funktionsweise der Pendelbegrenzung erkennbar. Durch die Vorgabe "PendulumFactor" = 90 ist vorgegeben, dass die Rückwärtsdrehung in dem Ausgleichsbereich auf $75\% + 25\% \cdot 0,9 = 97,5\%$ begrenzt wird.



Beispiel 3: Nullmarke vor dem Messer, Ausgleichsbewegung am Ende (SyncLength = 1024; CircSlotter = 512; SlotLength1 = 256; PendulumFactor = 90; SkipMode = 1; CamProf = 0)

Abb. 10-39: Beispielhaftes Bewegungsprofil 3

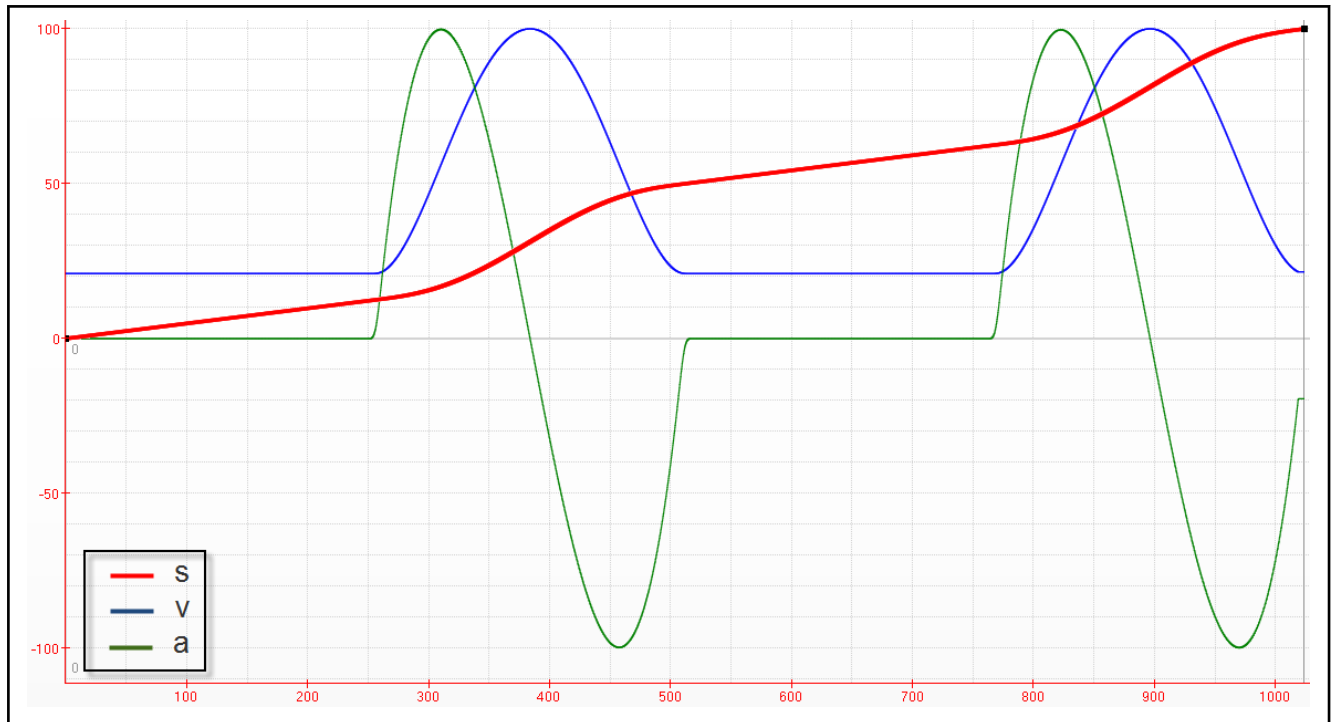
Beispiel 4 In der vierten Variante [Abb. 10-40 "Beispielhaftes Bewegungsprofil 4"](#) auf [Seite 695](#) wird eine Kurvenscheibe dargestellt, die einen Schlitz am Anfang und einen Schlitz am Ende des Bogens erzeugt. Bei Kurvenscheiben die zwei Schlitz pro Bogen schneiden, muss sich die Slotterwalze zweimal vollständig drehen, während der Bogen sie passiert. Um zwei Umdrehungen pro Leitachsumdrehung erreichen zu können müssen beide Umdrehungen in der einen, vom Funktionsbaustein erstellten Kurvenscheibe enthalten sein.



Bei Formatlängen, die nahezu dem Synchronformat entsprechen, liegen Slot2 des ersten und Slot1 des nachfolgenden Bogens direkt hintereinander. Die Slotterwalze muss in sehr kurzer Zeit eine fast komplette Umdrehung vollführen, damit das Messer zum richtigen Zeitpunkt in den zweiten Bogen eindringen kann. Dies stellt hohe Ansprüche an das Beschleunigungsvermögen der verwendeten Antriebe und Motoren und ist dynamisch ungünstig.

Unter dieser Voraussetzung wird empfohlen den Baustein nicht mit "SkipFeed" = 0 zu betreiben.

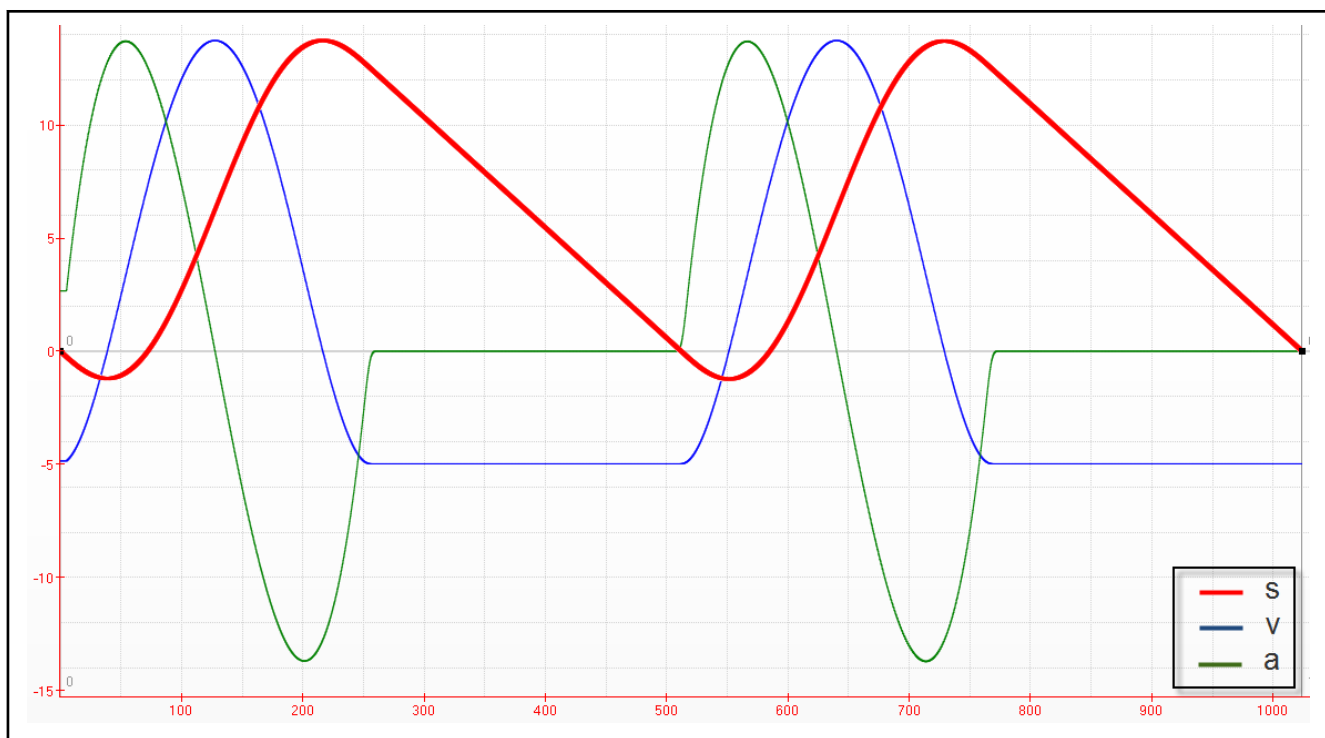
Die Kurvenscheibe zeichnet sich durch zwei Synchronbereiche (vorgegeben durch "SlotLength1" und "SlotLength2"), sowie zwei Ausgleichsbewegungen zwischen den Synchronbereichen aus. Während der Synchronbereiche befindet sich das auf der Slotterwalze angebrachte Messer im Materialeingriff und schneidet die beiden Schlitz.



Beispiel 4: Nullmarke vor dem Messer (SyncLength = 1024; CircSlotter = 1024; KnifeLength = 256; BoxHeight = 256; SlotLength1 = SlotLength2 = 256; SkipMode = 0; CamProf = 0)

Abb. 10-40: Beispielhaftes Bewegungsprofil 4

Beispiel 5 Basierend auf dem vorangehenden Beispiel 4 [Abb. 10-40 "Beispielhaftes Bewegungsprofil 4" auf Seite 695](#) folgt stellvertretend für eine additive Kurve das Beispiel 5 [Abb. 10-41 "Beispielhaftes Bewegungsprofil 5" auf Seite 696](#). Durch den Eingang "CamProf" wurde jedoch diesmal sowohl ein Überlagerungsprofil mit Untersetzung als auch eine Ausgleichsbewegung am Anfang und Synchronbewegung am Ende gewählt.



Beispiel 5: Nullmarke nach dem Messer (SyncLength = 1024; CircSlotter = 1024; KnifeLength = 256; BoxHeight = 256; SlotLength1 = SlotLength2 = 256; SkipMode = 0; CamProf = 3)

Abb. 10-41: Beispielhaftes Bewegungsprofil 5

Die zu berechnende Kurvenscheibentabelle wird im Falle der fehlerfreien Bearbeitung des Funktionsbausteins in der Struktur "CamTable" abgelegt, wobei die Gültigkeit der Kurvenscheibenelemente durch den Ausgang "Done" signalisiert wird.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" werden im Fehlerfall nicht aktualisiert.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR, 16#0005	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
CALCULATION_ERROR, 16#0007	16#000C	16#0000	Division durch Null
INPUT_INVALID_ERROR, 16#0001	16#2201	16#0001	Adresse von Feld "CamTable" ist NULL
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0001	"SyncLength" ist kleiner als Minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0002	"CircSlotter" ist kleiner als Minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0003	"KnifeLength" ist kleiner als Minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0004	"BoxHeight" ist kleiner als Minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0005	"SlotLength1" ist kleiner als Minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0006	"SlotLength2" ist kleiner als Minimum

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0007	"PendulumFactor" ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0008	"SkipMode" ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0009	"CamProf" ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0010	"NumberOfSlotters" ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0011	"SlotterToCalculate" ist kleiner als Minimum oder größer als "NumberOfSlotters"
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0012	"CamTable.NumberOfElements" ist ungleich 1024
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0001	"KnifeLength" ist größer als "CircSlotter"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0002	"SlotLength1" ist größer als "KnifeLength" * "NumberOfSlotters"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0003	"SlotLength2" ist größer als "KnifeLength" * "NumberOfSlotters"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0004	"SlotLength1" + "BoxHeight" + "SlotLength2" ist größer als "SyncLength" * ("SkipMode" + 1)
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0005	Wert von "CamProf" ist ungültig

Tab. 10-20: Fehlercodes FB MB_SlotterCalcType01

10.10 MB_SlotterCalcType02

Kurzbeschreibung

Der Funktionsbaustein MB_SlotterCalcType02 berechnet eine Kurvenscheibe mit 1024 Stützpunktelementen für einen einzelnen Slotter (Einfachslotter) mit zwei Messern auf der Messerwalze oder für einen Slotter im Verbund von Multi-Slottern mit je zwei Messern auf der Messerwalze.

Der Funktionsbaustein MB_SlotterCalcType02 deckt folgende Funktionalitäten ab:

- Zwei Schlitz pro Bogen (Einfachslotter). Ein Schlitz am Anfang des Bogens und ein Schlitz am Ende des Bogens.
- Zwei Schlitz pro Bogen im "Skip feed"-Modus mit einem Bogen über mehrere Maschinenabwicklungen.
- Zwei lange Schlitz pro Bogen (Multislotter). Ein langer Schlitz am Anfang des Bogens und ein langer Schlitz am Ende des Bogens.
- Zwei lange Schlitz pro Bogen im "Skip feed"-Modus mit einem Bogen über mehrere Maschinenabwicklungen.

Der Baustein berechnet immer nur eine Kurvenscheibe für einen Slotter. Bei n-Slottern im Multislotterverbund muss der Baustein n-mal aufgerufen werden. Für welchen Slotter der Baustein die Kurvenscheibe berechnen soll muss am Baustein über den Eingang "SlotterToCalculate" definiert werden.

Zuordnung Zielsystem/Bibliothek

MLC 14VRS Funktionspaket TEC	RMB_TechCam

Tab. 10-21: Referenztable MB_SlotterCalcType02

Schnittstellenbeschreibung Type02

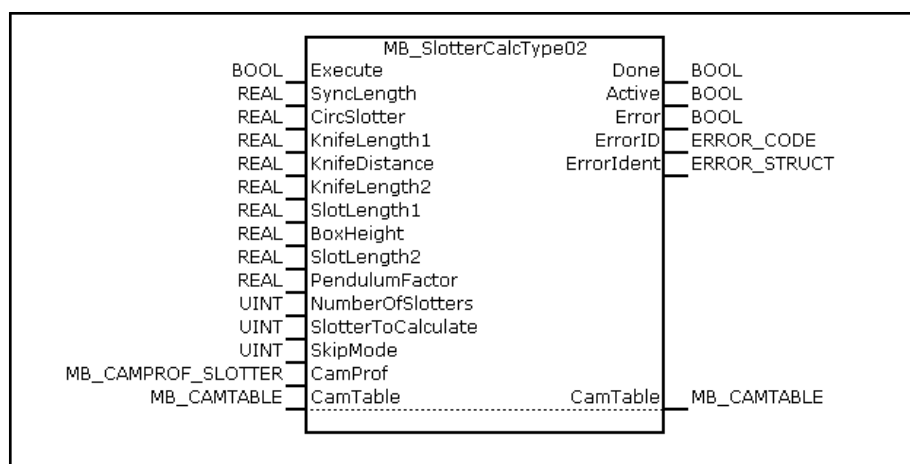


Abb. 10-42: Funktionsbaustein: MB_SlotterCalcType02

	Name	Typ	Kommentar
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flanken-gesteuert)
	SyncLength	REAL	Synchrone Formatlänge, d. h. mechanische Abwicklungslänge der Maschine, z. B. Umfang des Druckzylinders [u]*
	CircSlotter	REAL	Umfang der Slotterscheibe bzw. Messerflugkreis [u]*
	KnifeLength1	REAL	Länge des ersten Messers [u]*
	KnifeDis- tance	REAL	Abstand zwischen dem Ende von Messer 1 und Anfang von Messer 2 [u]*
	KnifeLength2	REAL	Länge des zweiten Messers [u]*
	SlotLength1	REAL	Gesamte Länge des in Bewegungsrichtung ersten Schlitzes im Bo- gen [u]*
	BoxHeight	REAL	Abstand zwischen den beiden Schlitzten im Bogen ("SlotLength1" + "BoxHeight" + "SlotLength2" = Bogenlänge) [u]*
	SlotLength2	REAL	Gesamte Länge des in Bewegungsrichtung zweiten Schlitzes im Bogen [u]*
	Pendulum- Factor	REAL	Maximaler Pendelweg [%] 0%: Kein Pendeln (Stillstand bei großer Formatlänge) 100%: Maximales Pendeln (Maximale Rückwärtsdrehung bei großer Formatlänge)
	NumberOfS- lotters	UINT	Anzahl an Slottern im Multi-Slotter-Verbund, die gemeinsam die bei- den Schlitzte am Anfang und am Ende des Bogens erstellen [-]
	SlotterToCal- culate	UINT	Nummer des Slotters, für den beim Aufruf des Bausteins die Kurvenscheibe berechnet werden soll [-]
	SkipMode	UINT	Anzahl der Maschinenzyklen pro Bogen - 1 [-]
	CamProf	MB_CAM- PROF_SLOTTER	Auswahl des Kurvenscheibenprofils und Lage der Nullmarke der Slotterwalze MB_CAMPROF_SLOTTER, Seite 708
VAR_OUTPUT	Done	BOOL	Berechnung der Kurvenscheibenelemente ohne Fehler abgeschlos- sen, Liste der Kurvenscheibenelemente gültig

	Name	Typ	Kommentar
	Active	BOOL	Berechnung der Kurvenscheibenelemente aktiv, Liste der Kurvenscheibenelemente ungültig
	Error	BOOL	Berechnung der Kurvenscheibenelemente mit Fehler abgeschlossen, Ausgangsvariablen "ErrorID" und "ErrorIdent" gültig
	ErrorID	ERROR_CODE	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang eine grobe Klassifizierung des Fehlers
	ErrorIdent	ERROR_STRUCT	Bei gesetztem "Error"-Ausgang enthält dieser Ausgang detaillierte Informationen über den Fehler Fehlerbehandlung, Seite 707
VAR_IN_OUT	CamTable	MB_CAMTABLE	Struktur mit den errechneten Kurvenscheibentabellen-Elementen und Information über die Anzahl der Elemente

*

Die Einheit [u] bedeutet „unit“. Dies kann eine beliebige Einheit wie z. B. [mm] oder [inch] sein. Es ist darauf zu achten, dass die gewählte Einheit einheitlich verwendet wird.

Tab. 10-22: Schnittstellenvariablen Funktionsbaustein MB_SlotterCalcType02

Min-/Max- und Standardwerte

In der folgenden Tabelle sind die Min-/Max- und Standardwerte der Bausteineingänge aufgelistet.

Name	Typ	Min-Wert	Max-Wert	Standardwert	Übernahme
Execute	BOOL	FALSE	TRUE	FALSE	Kontinuierlich
SyncLength	REAL	>0.0	n.def.	0.0	Steigende Flanke an "Execute"
CircSlotter	REAL	>0.0	n.def.	0.0	Steigende Flanke an "Execute"
KnifeLength1	REAL	>0.0	n.def.	0.0	Steigende Flanke an "Execute"
KnifeDistance	REAL	>0.0	n.def.	0.0	Steigende Flanke an "Execute"
KnifeLength2	REAL	>0.0	n.def.	0.0	Steigende Flanke an "Execute"
SlotLength1	REAL	>0.0	n.def.	0.0	Steigende Flanke an "Execute"
BoxHeight	REAL	>0.0	n.def.	0.0	Steigende Flanke an "Execute" und SlotLength2 > 0
SlotLength2	REAL	>0.0	n.def.	0.0	Steigende Flanke an "Execute"
PendulumFactor	REAL	0.0	100.0	95.0	Steigende Flanke an "Execute"
NumberOfSlotters	UINT	1	10	1	Steigende Flanke an "Execute"
SlotterToCalculate	UINT	1	10	1	Steigende Flanke an "Execute"

Name	Typ	Min-Wert	Max-Wert	Standardwert	Übernahme
SkipMode	UINT	0	10	0	Steigende Flanke an "Execute"
CamProf	MB_CAM- PROF_SLOTTER	0	3	0	Steigende Flanke an "Execute"

Tab. 10-23: Min-/Max- und Default-Werte Funktionsbaustein MB_SlotterCalcType02

Signal-Zeit-Diagramm

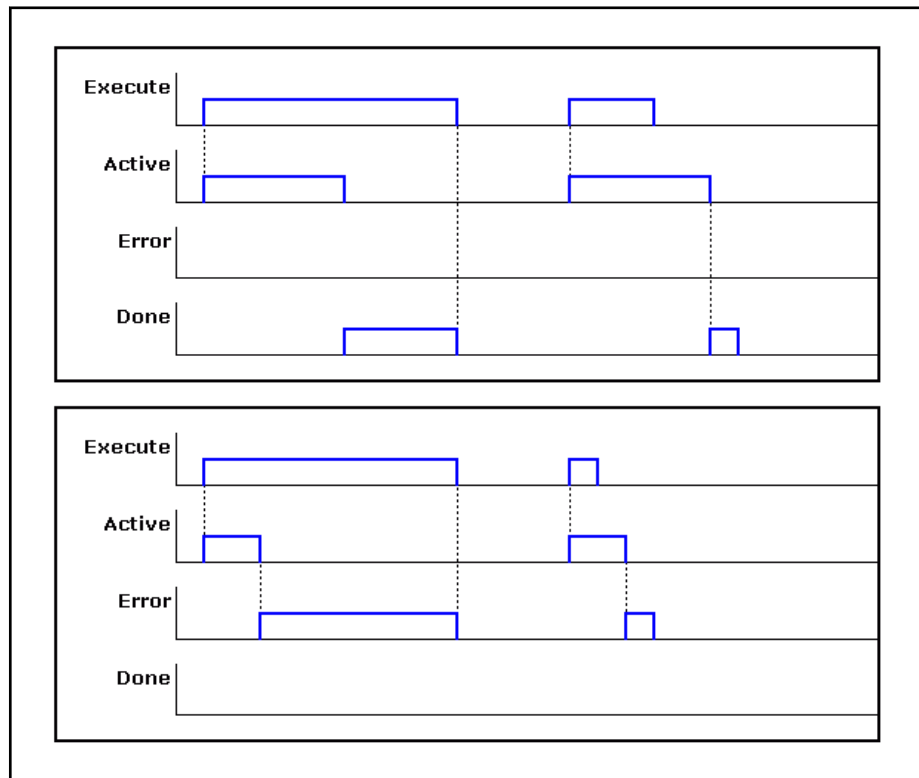


Abb. 10-43: Signal-Zeit-Diagramm Funktionsbaustein MB_SlotterCalcType02

Eigenschaften der Kurvenscheibe

Die Kurvenscheibentabelle des Funktionsbausteins "MB_SlotterCalcType02" weist folgende Eigenschaften auf:

- Das Kurvenscheibenprofil kann für die zyklische Abarbeitung als Kurvenscheibenparameter in den Antrieb (P-0-0072, P-0-0092, P-0-0780 und P-0-0781) oder in die Steuerung (C-0-2001 bis C-0-2099) geladen werden.
- Das durch die Kurvenscheibentabelle charakterisierte Bewegungsprofil ist stetig bis zur Beschleunigung (s, v, a) und weist einen Sprung im Ruck (j) auf.
- Der Funktionsbaustein kann Profile sowohl für endlos drehende Kurvenscheiben ohne Untersetzung (P-0-0086 Bit4 = 0) als auch für Überlagerungsprofile mit Untersetzung (P-0-0086 Bit4 = 1) erzeugen.
- Die Nullmarke der Slotterwalze kann, in Drehrichtung, sowohl am Anfang wie auch am Ende des Messers liegen.
- Der Algorithmus zur Berechnung der Kurvenscheibenstützpunkte ist auf mehrere Zyklen, d. h. Aufrufe der Instanz, verteilt, um einen merklichen Einfluss auf die Zykluszeit der SPS zu vermeiden.

- Bei überlangen Synchronbereichen kann die Ausgleichsbewegungen zur Rückwärtsdrehung führen. Die maximale Rückwärtsdrehung des Schlitzmessers kann vorgegeben werden.
- Im SkipFeed-Modus wird davon ausgegangen, dass sich ein Produkt über mehrere Maschinenzyklen ("synchrone Formatlängen") erstreckt. SkipMode n streckt eine Arbeitsumdrehung auf n+1 Maschinenzyklen. Durch Einsatz eines elektronischen Getriebes muss sichergestellt sein, dass die vom Funktionsbaustein erstellte Kurvenscheibe auf die entsprechende Anzahl an Maschinenzyklen gestreckt wird.

Eigenschaften im Multislotter-Verbund

Beim Betrieb von Multislottern sind folgende weitere Punkte zu beachten:

- Der Slotter, der als erstes mit dem Bogen in Berührung kommt ist Slotter 1 der letzte ist Slotter n.
- Slotter 1 schneidet zur Erstellung von Slot 1 immer zunächst ein Fenster in den Bogen. Slotter 2 bis n schneiden dann den Slot frei. Auf der anderen Seite des Bogens (Slot 2) schneidet Slotter 1 das Stück direkt an der Bogenkante und Slotter 2 bis n vergrößern den Schlitz bis zur "SlotLength2".
- Die Messer der einzelnen Slotter müssen gleich lang sein.
- Die einzelnen Slotter teilen sich "SlotLength1" und "SlotLength2" zu gleichen Teilen auf. Jeder der n Slotter schneidet 1/n-tel der SlotLength von Slot1 und Slot2 frei.
- Zwischen den Nullmarken von Slotter 1 und Slotter n im Bogen liegt die Kartonhöhe (BoxHeight) an. Die Nullmarke von Slotter n-1 ist um 1/n der "SlotLength2" in Richtung der hinteren Bogenkante versetzt.
- An den Eingängen "SlotLength1" und "SlotLength2" muss die gesamte Länge der Slots 1 und 2 angelegt werden und nicht die Länge des Teilslots, die der zu berechnende Slotter erstellt.
- Am Eingang "NumberOfSlotters" muss definiert werden, wie viele Slotter sich im Multi-Slotter-Verbund befinden und gemeinsam an den beiden Slots arbeiten.
- Am Eingang "SlotterToCalculate" muss definiert werden, für welchen der "NumberOfSlotters"-Slotter die Kurvenscheibe berechnet werden soll. Die Nummerierung der Slotter erfolgt in der Reihenfolge in der sie den Bogen berühren (Slotter 1 berührt Bogen als erstes, Slotter2 als zweites, ...).
- Da Slotter1 bei Slot1 ein Fenster in das Material schneidet, muss er für seine komplette Messerlänge synchron mit der Materialgeschwindigkeit laufen. Alle weiteren Slotter laufen nur für ihre jeweilige Teilslot-Länge synchron. Aus diesem Grund können sich Slotter 2-n dynamisch günstiger verhalten als Slotter 1.

Funktionsbeschreibung

Mit der Vorgabe, in einem vom Schneidprozess bestimmten Synchronbereich (vorgegeben durch "SlotLength1" und "SlotLength2") die Umfangsgeschwindigkeit der Messerwalze synchron zur Geschwindigkeit der Materialbahn zu halten, berechnet der Funktionsbaustein nach der Bearbeitungsfreigabe mittels "Execute" auf Basis der Eingangsgrößen einmalig eine Kurvenscheibentabelle mit 1024 Stützpunktelementen.

Die konkrete Bedeutung der Eingangsgrößen zeigt nachfolgende Darstellung.

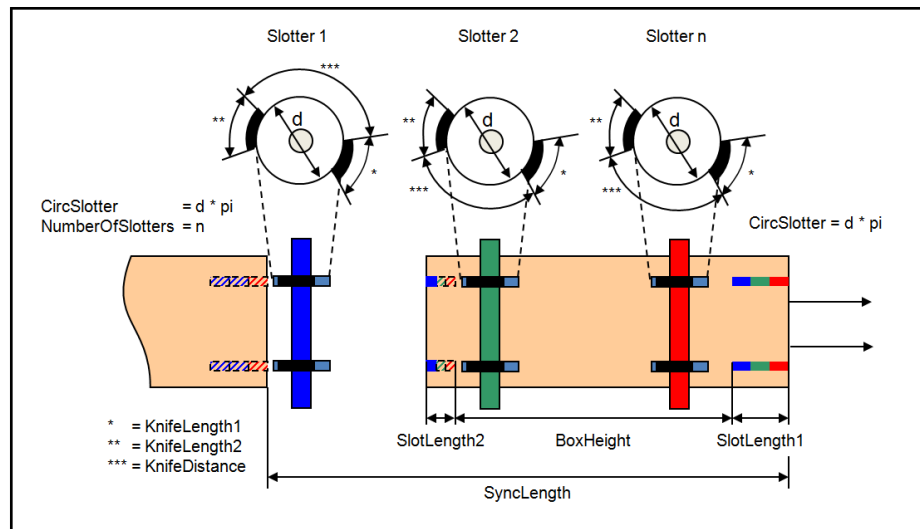
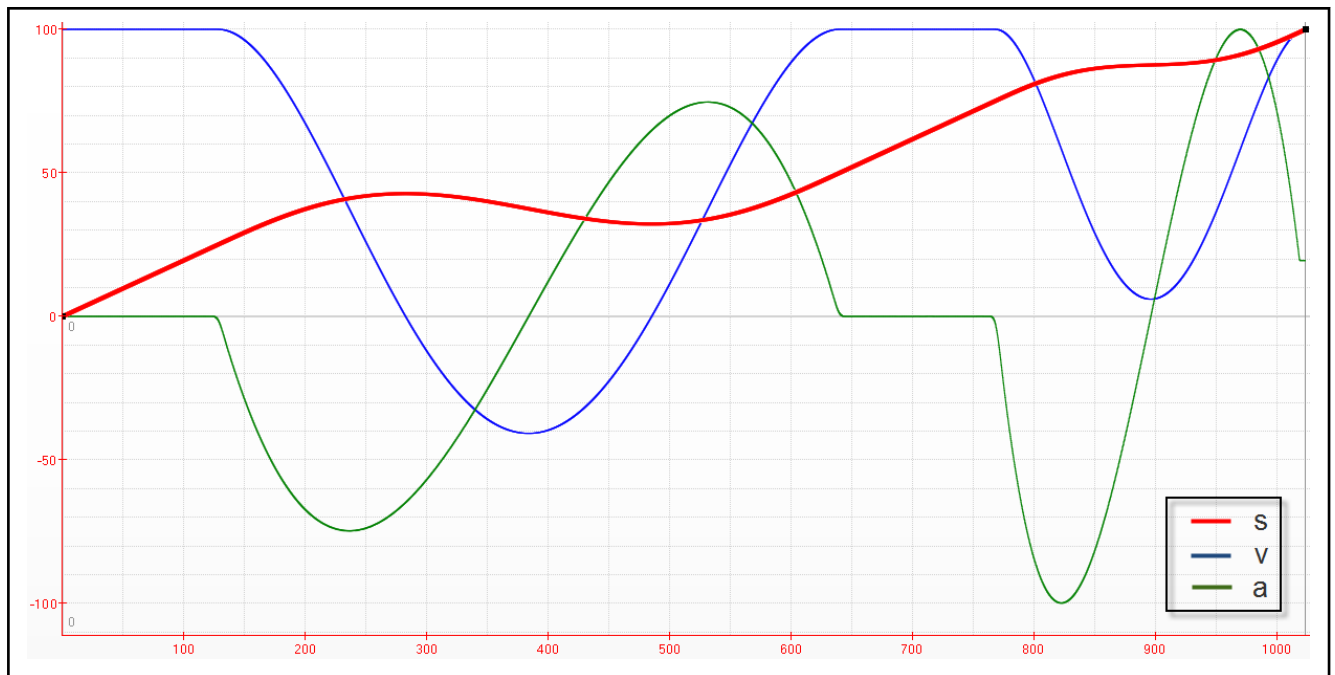


Abb. 10-44: Bedeutung der Eingangsgrößen des Funktionsbausteins MB_SlotterCalcType02

Entsprechend den zwei vorgegebenen Synchronbereichen, müssen zwei durch den Funktionsbaustein berechnete Ausgleichsbewegung stattfinden. Diese Ausgleichsbewegungen werden durch eine Stetigkeit bis zur Beschleunigung charakterisiert.

Mittels verschiedener Vorbelegungen der Eingänge sind verschiedene Funktionsvarianten mit dem Funktionsbaustein MB_SlotterCalcType02 möglich:

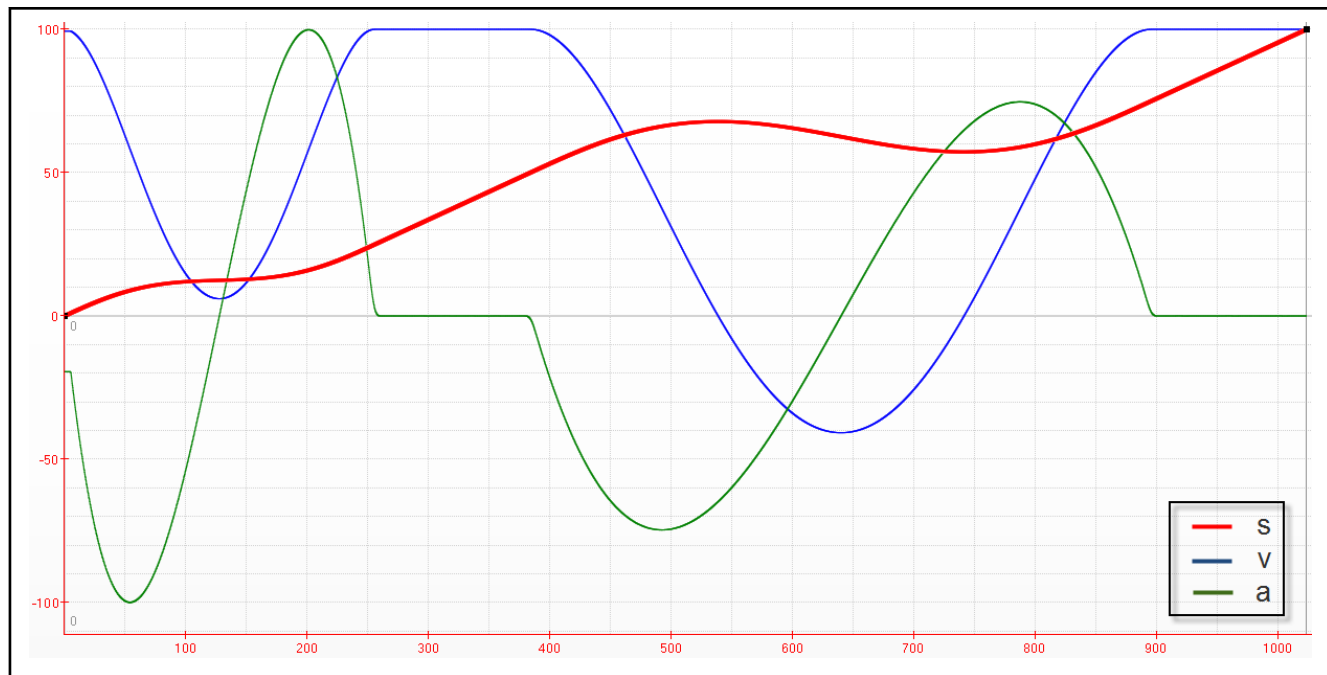
Die erste Variante [Abb. 10-45 "Beispielhaftes Bewegungsprofil 1" auf Seite 703](#) eignet sich, um Formate mit einem Slotter zu schneiden, dessen Messerflugkreis "CircSlotter" kleiner ist als die mechanische Abwicklungslänge der Maschine "SyncLength". Die wirksamen Messerlängen werden hierbei auf die Materialtransportgeschwindigkeit im Werkzeugeingriff angepasst. Der verbleibende Scheibenumfang des Slotters ohne Schneidmesser wird als Bereich für die Ausgleichsbewegungen definiert.



Beispiel 1: Nullmarke vor dem Messer, Ausgleichsbewegung am Ende (SyncLength = 1024; CircSlotter = 512; SlotLength1 = SlotLength2 = 128; BoxHeight = 256; KnifeLength1 = KnifeLength2 = KnifeDistance = 128; SkipMode = 0; CamProf = 0)

Abb. 10-45: Beispielhaftes Bewegungsprofil 1

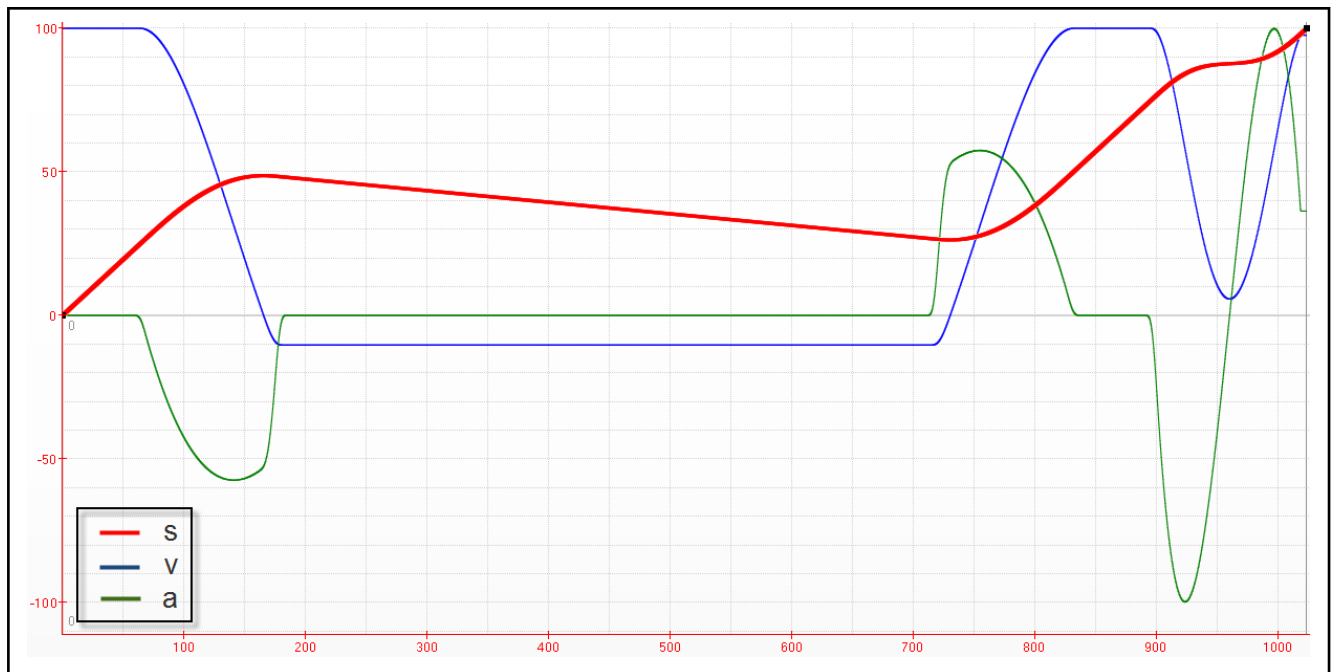
In sämtlichen Einfachslotter-Varianten "NumberOfSlotters" = 1 kann die Nullmarke der Slotterwalze durch Parametrierung des "CamProf"-Eingangs am Ende des ersten Messers oder am Anfang des zweiten Messers definiert werden. In der nachfolgenden Variante [Abb. 10-46 "Beispielhaftes Bewegungsprofil 2" auf Seite 704](#) wurde stellvertretend die erste Variante [Abb. 10-45 "Beispielhaftes Bewegungsprofil 1" auf Seite 703](#) erneut aufgegriffen, und mit dem Eingang "CamProf" eine Ausgleichsbewegung am Anfang und die Synchronbewegung am Ende gewählt.



Beispiel 2: Nullmarke nach dem Messer, Ausgleichsbewegung am Anfang (SyncLength = 1024; CircSlotter = 512; SlotLength1 = SlotLength2 = 128; BoxHeight = 256; KnifeLength1 = KnifeLength2 = KnifeDistance = 128; SkipMode = 0; CamProf = 2)

Abb. 10-46: Beispielhaftes Bewegungsprofil 2

Die dritte Variante [Abb. 10-47 "Beispielhaftes Bewegungsprofil 3"](#) auf Seite 705 eignet sich, um überlange Formate, d. h. welche größer als die maximale synchrone Formatlänge sind, verarbeiten zu können. Dabei wird der Schlitzprozess auf mehrere Maschinenzyklen gestreckt, wobei die Schneidbereiche der Messer synchron zur Oberflächengeschwindigkeit bleiben ("SkipMode" = n). Der Slotter dreht sich mit einer mechanischen Umdrehung während n+1 Maschinenzyklen. Während die Bereiche mit den wirksamen Messerlängen synchron zum Material bleiben, wird in der restlichen Bewegung bis zum Vollenden des n+1 Zyklusses eine Ausgleichsbewegung ausgeführt. Nachfolgend ist des Weiteren die Funktionsweise der Pendelbegrenzung erkennbar. Durch die Vorgabe "PendulumFactor" = 90 ist vorgegeben, dass die Rückwärtsdrehung in dem Ausgleichsbereich auf $37,5\% + 12,5\% \cdot 0,9 = 48,75\%$ begrenzt wird.



Beispiel 3: Nullmarke vor dem Messer, Ausgleichsbewegung am Ende (SyncLength = 1024; CircSlotter = 512; SlotLength1 = SlotLength2 = 128; BoxHeight = 256; KnifeLength1 = KnifeLength2 = KnifeDistance = 128; PendulumFactor = 90; Skip-Mode = 1; CamProf = 0)

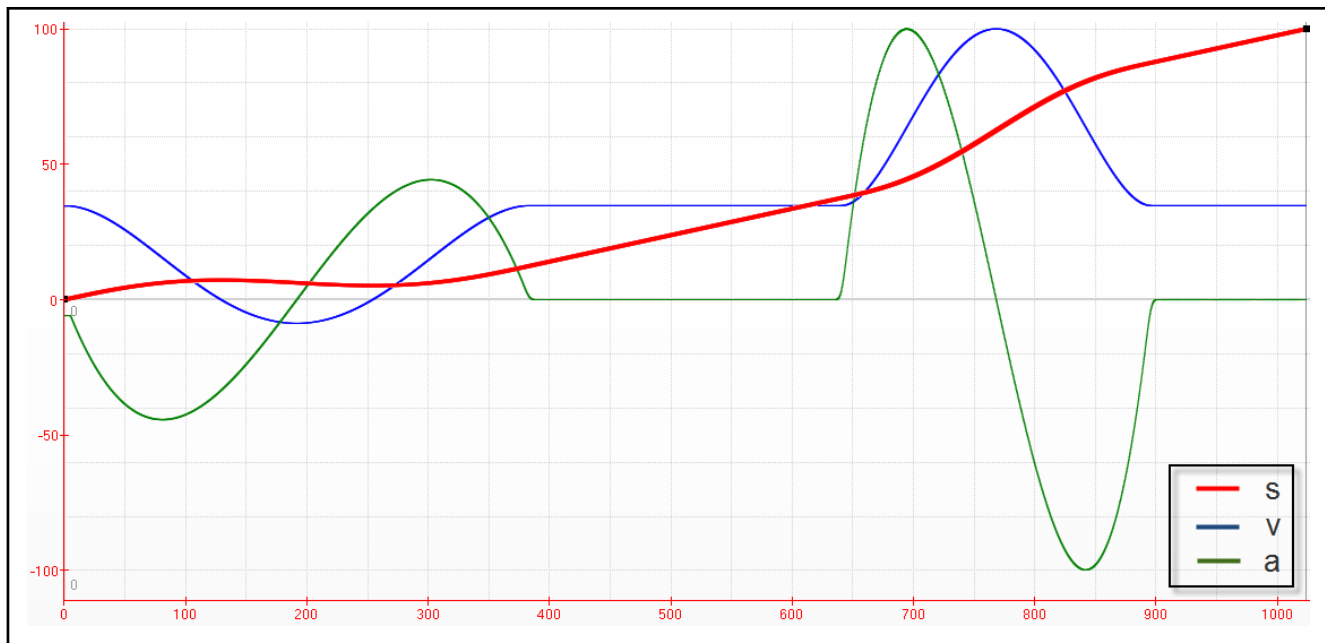
Abb. 10-47: Beispielhaftes Bewegungsprofil 3

In der vierten Variante werden Kurvenscheiben dargestellt, die einen langen Schlitz am Anfang und einen langen Schlitz am Ende des Bogens erzeugen (Multislotter).



Im Gegensatz zu der Definitionsmöglichkeit der Nullmarke bei Verwendung von Einfachslottern, sind bei Multislottern die Nullmarken der einzelnen Slotter entsprechend [Abb. 10-32 "Definitionen und Festlegungen für Multislotter"](#) auf Seite 685 festgelegt. Slotter 1 (NumberOfSlotter = 1) hat seine Nullmarke stets in Drehrichtung am Ende des Messers (CamProf = 2, 3), und Slotter 2 bis n (NumberOfSlotter = 2,...,n) am Anfang (CamProf = 0, 1).

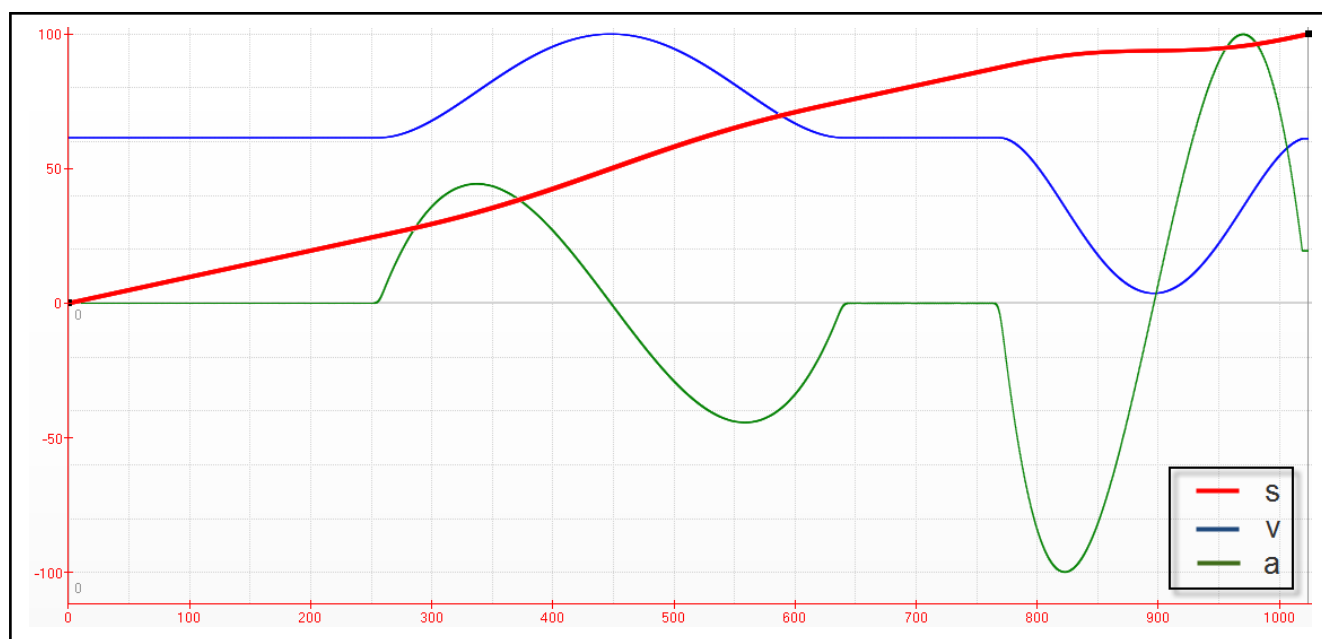
Die Kurvenscheibe des ersten Slotters zeichnet sich entsprechend der Lage der Nullmarke durch eine Kurvenscheibe aus, die mit einer Ausgleichsbewegung beginnt und mit einem Synchronbereich endet.



Beispiel 4: Erster Slotter im Multislotterverbund (SyncLength = 1024; CircSlotter = 1024; KnifeLength1 = 128; KnifeLength2 = 256; KnifeDistance = 128; BoxHeight = 128; SlotLength1 = 256; SlotLength2 = 512; NumberOfSlotters = 2, SlotterToCalculate = 1; SkipMode = 0; CamProf = 2)

Abb. 10-48: Beispielhaftes Bewegungsprofil 4

Die Kurvenscheibe des zweiten Slotters zeichnet sich entsprechend der Lage der Nullmarke durch eine Kurvenscheibe aus, die mit einem Synchronbereich beginnt und mit einer Ausgleichsbewegung endet.



Beispiel 5: Zweiter Slotter im Multislotterverbund (SyncLength = 1024; CircSlotter = 1024; KnifeLength1 = 128; KnifeLength2 = 256; KnifeDistance = 128; BoxHeight = 128; SlotLength1 = 256; SlotLength2 = 512; NumberOfSlotters = 2, SlotterToCalculate = 2; SkipMode = 0; CamProf = 0)

Abb. 10-49: Beispielhaftes Bewegungsprofil 5

Die zu berechnende Kurvenscheibentabelle wird im Falle der fehlerfreien Bearbeitung des Funktionsbausteins in der Struktur "CamTable" abgelegt, wobei die Gültigkeit der Kurvenscheibenelemente durch den Ausgang "Done" signalisiert wird.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert. Die Elemente der Struktur "CamTable" werden im Fehlerfall nicht aktualisiert.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#170. Er kann in Additional1 und Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
STATE_MACHINE_ERROR, 16#0005	16#0006	16#0000	Ungültiger Zustand des Funktionsbausteins
CALCULATION_ERROR, 16#0007	16#000C	16#0000	Division durch Null
INPUT_INVALID_ERROR, 16#0001	16#2201	16#0001	Adresse von Feld "CamTable" ist NULL
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0001	"SyncLength" ist kleiner als Minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0002	"CircSlotter" ist kleiner als Minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0004	"BoxHeight" ist kleiner als Minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0005	"SlotLength1" ist kleiner als Minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0006	"SlotLength2" ist kleiner als Minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0007	"PendulumFactor" ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0008	"SkipMode" ist kleiner als Minimum oder größer als Maximum

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0009	"CamProf" ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0010	"NumberOfSlotters" ist kleiner als Minimum oder größer als Maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0011	"SlotterToCalculate" ist kleiner als Minimum oder größer als "NumberOfSlotters"
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0012	"CamTable.NumberOfElements" ist ungleich 1024
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0013	"KnifeLength1" ist kleiner als Minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0014	"KnifeDistance" ist kleiner als Minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0015	"KnifeLength2" ist kleiner als Minimum
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0004	"SlotLength1" + "BoxHeight" + "SlotLength2" ist größer als "SyncLength" * ("SkipMode" + 1)
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0005	Wert von "CamProf" ist ungültig
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0006	"KnifeLength1" ist größer als "CircSlotter"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0007	"KnifeLength2" ist größer als "CircSlotter"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0008	"SlotLength1" ist größer als "KnifeLength1" * "NumberOfSlotters"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0009	"SlotLength2" ist größer als "KnifeLength2" * "NumberOfSlotters"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0010	"KnifeLength1" + "KnifeDistance" + "KnifeLength2" ist größer als "CircSlotter"

Tab. 10-24: Fehlercodes FB MB_SlotterCalcType02

10.11 MB_CAMPFROF_SLOTTER

Die Enumeration "MB_CAMPFROF_SLOTTER" wird verwendet um zwischen verschiedenen Kurvenscheibenprofilen und Nullmarken zu wählen.

Element	Wert	Beschreibung
CAMPFROF_STANDARD_SEGMENTTABLE_ZERO_KNIFE	0	Kurvenscheibenprofil ohne Untersetzung. Synchronbereich am Anfang, Ausgleichsbewegung am Ende
CAMPFROF_ADDITIVE_SEGMENTTABLE_ZERO_KNIFE	1	Überlagerungsprofil mit Untersetzung. Synchronbereich am Anfang, Ausgleichsbewegung am Ende
CAMPFROF_STANDARD_SEGMENTTABLE_KNIFE_ZERO	2	Kurvenscheibenprofil ohne Untersetzung. Synchronbereich am Ende, Ausgleichsbewegung am Anfang
CAMPFROF_ADDITIVE_SEGMENTTABLE_KNIFE_ZERO	3	Überlagerungsprofil mit Untersetzung. Synchronbereich am Ende, Ausgleichsbewegung am Anfang

Tab. 10-25: Elemente des Aufzählungstyps MB_CAMPFROF_SLOTTER

11 RIL_ParameterChannel.library

11.1 Einführung und Übersicht

Die Bibliothek RIL_ParameterChannel mit ihrem Baustein [IL_ParameterChannel](#), Seite 709, ermöglicht eine azyklische Kommunikation über den zyklischen Kanal einer Feldbusverbindung.



Die Bibliothek benötigt das Grundpaket PLC der Steuerung.

11.2 IL_ParameterChannel

Kurzbeschreibung

Der Baustein ermöglicht eine azyklische Kommunikation über den zyklischen Datenkanal. Der Datenaustausch erfolgt über diesen Funktionsbaustein auf der Steuerung, sowie einen korrespondierenden Funktionsbaustein im Feldbus-Master.

Zuordnung Zielsystem/Bibliothek

Zielsystem	Bibliothek
MLC 14VRS	RIL_ParameterChannel
Grundpaket PLC	

Tab. 11-1: Referenztable FB IL_ParameterChannel

Schnittstellenbeschreibung

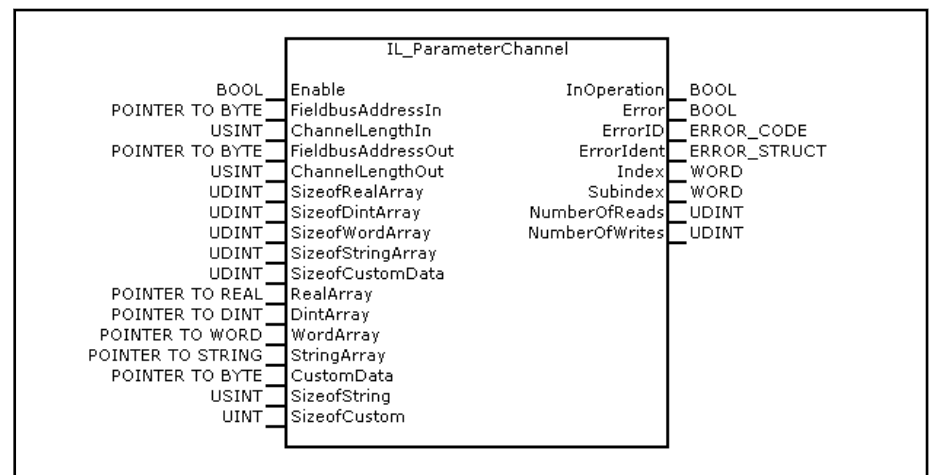


Abb. 11-1: FB IL_ParameterChannel

	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (einmalig, flankengesteuert)
	FieldbusAddressIn	POINTER TO BYTE	Adresse des Eingangsbereiches für den zyklischen Datenverkehr
	ChannelLengthIn	USINT	Länge der zyklischen Eingänge für den Parameterkanal in Bytes.
	FieldbusAddressOut	POINTER TO BYTE	Adresse des Ausgangsbereiches für den zyklischen Datenverkehr
	ChannelLengthOut	USINT	Länge der zyklischen Ausgänge für den Parameterkanal in Bytes.
	SizeofRealArray	UDINT	Anzahl der REAL-Daten die über die Adresse RealArray zugänglich sind in Bytes.

	Name	Datentyp	Beschreibung
	SizeofDintArray	UDINT	Anzahl der DINT-Daten die über die Adresse DintArray zugänglich sind in Bytes.
	SizeofWordArray	UDINT	Anzahl der WORD-Daten die über die Adresse WordArray zugänglich sind in Bytes.
	SizeofStringArray	UDINT	Anzahl der STRING-Daten die über die Adresse StringArray zugänglich sind in Bytes.
	SizeofCustomData	UDINT	Anzahl der gesamten Custom-Daten die über die Adresse CustomData zugänglich sind in Bytes.
	RealArray	POINTER TO REAL	Ein Zeiger auf ein Feld von REAL-Werten. Das Feld kann eine Größe von 4845 Elementen besitzen
	DintArray	POINTER TO DINT	Ein Zeiger auf ein Feld von DINT-Werten. Das Feld kann eine Größe von 4845 Elementen besitzen
	WordArray	POINTER TO WORD	Ein Zeiger auf ein Feld von WORD-Werten. Das Feld kann eine Größe von 2295 Elementen besitzen
	StringArray	POINTER TO STRING	Ein Zeiger auf ein Feld von STRING-Werten Das Feld kann eine Größe von 2040 Elementen besitzen
	CustomData	POINTER TO BYTE	Ein Zeiger auf ein Feld von BYTE-Werten. Das Feld kann eine Größe von bis zu 255 * 8 KByte Daten besitzen
	SizeofString	USINT	Größe eines STRINGs im StringArray. Vorsicht: dies ist die Größe, die bei der Definition des Strings angegeben wird (ohne Ende-Kennung).
	SizeofCustom	UINT	Größe eines Datenfeldes (1 – 8192 Bytes)
VAR_OUTPUT	InOperation	BOOL	Der Baustein arbeitet korrekt, Ausgangsdaten gültig
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose Fehlerbehandlung, Seite 714
	Index	WORD	Letzter erfolgreich verwendeter Index
	Subindex	WORD	Letzter erfolgreich verwendeter Subindex
	NumberOfReads	UDINT	Anzahl der erfolgreichen Lesezugriffe durch den Master. Der Wert kann überlaufen (von 65535 auf 0).
	NumberOfWrites	UDINT	Anzahl der erfolgreichen Schreibzugriffe durch den Master. Der Wert kann überlaufen (von 65535 auf 0).

Tab. 11-2: Schnittstellenbeschreibung Funktionsbaustein IL_ParameterChannel

Min-/Max-Default-Werte und Übernahme der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
FieldbusAddressIn	POINTER TO BYTE	16#00	16#FFFFFFFF	16#00	Steigende Flanke an Enable

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
ChannelLengthIn	USINT	6	18	12	Steigende Flanke an Enable
FieldbusAddressOut	POINTER TO BYTE	16#00	16#FFFFFFFF	16#00	Steigende Flanke an Enable
ChannelLengthOut	USINT	6	18	12	Steigende Flanke an Enable
SizeofRealArray	UDINT	0	19380 (4845 * 4)	0	Steigende Flanke an Enable
SizeofDintArray	UDINT	0	19380 (4845 * 4)	0	Steigende Flanke an Enable
SizeofWordArray	UDINT	0	4590 (2295 * 2)	0	Steigende Flanke an Enable
SizeofStringArray	UDINT	0	522240 (2040 * 256)	0	Steigende Flanke an Enable
SizeofCustomData	UDINT	0	2088960 (255 * 8192)	0	Steigende Flanke an Enable
RealArray	POINTER TO REAL	16#00	16#FFFFFFFF	16#00	Steigende Flanke an Enable
DintArray	POINTER TO DINT	16#00	16#FFFFFFFF	16#00	Steigende Flanke an Enable
WordArray	POINTER TO WORD	16#00	16#FFFFFFFF	16#00	Steigende Flanke an Enable
StringArray	POINTER TO STRING	16#00	16#FFFFFFFF	16#00	Steigende Flanke an Enable
CustomData	POINTER TO BYTE	16#00	16#FFFFFFFF	16#00	Steigende Flanke an Enable
SizeofString	USINT	0	255	80	Steigende Flanke an Enable
SizeofCustom	UINT	0	8192	128	Steigende Flanke an Enable

Tab. 11-3: Min-/Max-/Default-Werte Funktionsbaustein IL_ParameterChannel

Funktionsbeschreibung

Der Baustein ermöglicht eine azyklische Kommunikation über einen zyklischen Kanal. Dazu wird ein Bereich aus dem zyklischen Kanal für die Kommunikation genutzt. Für Daten vom Master zum Slave werden die Eingänge "FieldbusAddressIn" und "ChannelLengthIn" verwendet. Für die Richtung vom Slave zum Master werden die Eingänge "FieldbusAddressOut" und "ChannelLengthOut" verwendet. Die Begriffe "out" und "in" beziehen sich hier auf die Richtung aus Sicht des Slaves. Die Datenbereiche können an beliebiger Stelle innerhalb der zyklischen Daten liegen und auch unterschiedlich lang sein. Die minimale Länge des Parameterkanals beträgt 6, die maximale Länge in beide Richtungen 18 Byte.

SPS-Daten können nur über Datenfelder ausgetauscht werden. Dabei müssen die unterschiedlichen Datentypen berücksichtigt werden.

Übertragbare Datentypen

- REAL (bis zu 4845 Datenfelder)
- DINT (bis zu 4845 Datenfelder)
- WORD (bis zu 2295 Datenfelder)
- STRING (bis zu 2040 Datenfelder)

- Custom (bis zu 255 Datenfelder)

Alle Datenbereiche werden als Zeiger dem Baustein zugewiesen. Die zur Verfügung stehende Größe muss dann jeweils auch angegeben werden. Ist die Datengröße 0 oder ist der zugewiesene Zeiger 0, so interpretiert der Baustein dies als nicht vorhandenen Datenbereich und wird Anfragen auf diese Daten mit Fehler beantworten. Die Datengröße muss in Bytes erfolgen, d.h. der SIZEOF()-Operator kann verwendet werden.

Für Strings muss zusätzlich zur Größe des Feldes auch noch die Größe des Strings selbst am Eingang "SizeofString" angegeben werden. Dies ist deshalb erforderlich, da ein String mit unterschiedlicher Größe definiert werden kann. Die Größe der Strings muss innerhalb des Feldes einheitlich sein und darf maximal aus 255 Zeichen bestehen.



die Größe des Strings ist nur die Größe, die bei der Definition des Strings angegeben wird. Indralogic wird aber noch ein weiteres Byte für die Endekennung anhängen. Für die Definition

```
strExample: STRING(20);
```

werden in der SPS 21 Bytes belegt, denn nach den 20 Zeichen muss noch die Endekennung angefügt werden.

Der Funktionsbaustein IL_ParameterChannel berücksichtigt dies automatisch bei der Überprüfung der Größen, so dass im vorherigen Beispiel "SIZEOF(strExample)" als Wert für "SizeofString" angegeben werden muss.

Die Custom-Daten können dazu verwendet werden eine größere Anzahl an Daten konsistent zu übertragen. Dabei kann man sich die Custom-Daten z.B. als Datenstruktur vorstellen, die komplett übertragen werden soll. Es kann bis zu 255 solcher Strukturen geben. Um diese Funktionalität zu nutzen, muss wiederum ein Zeiger zugewiesen werden ("CustomData"). Ebenso muss die Größe einer einzelnen Struktur bekannt sein. Dies erfolgt im Eingang "SizeofCustom". Schließlich wird noch die Gesamtgröße benötigt ("SizeofCustomData").

Da der Funktionsbaustein keine Kenntnis über den Inhalt der Daten besitzt, werden die Daten byteweise kopiert. Es ist Aufgabe der jeweiligen Applikation sicherzustellen, dass das Datenformat auf Slave und Master konsistent ist.



Da bei den Custom-Daten ein Datensatz immer vollständig kopiert wird, nimmt die Kopierzeit einen signifikanten Stellenwert in der Laufzeit des Bausteins ein.

Laufzeit für das Kopieren und Bearbeiten des Protokolls

- 256 Bytes ca. 12µs
- 4 KBytes ca. 20µs
- 8 KBytes ca. 40µs auf einer L65

Bitte berücksichtigen Sie dies bei der Taskkonfiguration.



Werden Funktionen zur Prüfung der Gültigkeit von Zeigern und Feldergrenzen verwendet (z.B. RIL_CheckRtv.library), so nimmt die Laufzeit des Bausteins drastisch zu.

Bei 4 KByte Custom-Daten werden dann statt 20µs bis zu 30ms benötigt!



Die angelegten Pointer und Längen dürfen sich nicht ändern, solange der "Enable"-Eingang TRUE ist.

Dies ist besonders bei einem Online Change zu berücksichtigen, wenn dabei die Größen der angelegten Felder verändert wird.

Ändern sich Pointer oder Längen während der Übertragung, so wird ein Fehler generiert.

Die Ausgänge "NumberOfReads" und "NumberOfWrites" dienen dazu im Slave eine Aktion durch den Master zu erkennen und darauf reagieren zu können. So kann in der SPS auf dem Slave der jeweilige Wert von "NumberOfReads" und "NumberOfWrites" gespeichert werden. Bei einer Änderung kann dann eine Aktion erfolgen. So ist es z.B. möglich nach einer Änderung von "NumberOfWrites" über den ebenso angegebenen "Index" und "Subindex" den vom Master gerade geschriebenen Wert zu verarbeiten.

Die Ausgänge werden jeweils nur dann aktualisiert, wenn das Lesen, bzw. Schreiben erfolgreich war. Trat allerdings ein Fehler auf, so werden die Zähler nicht erhöht. Die Zähler sind jeweils vom Typ UDINT. Damit haben sie nur einen Wertebereich von 0 bis 65535. Steht der Zähler bei dem Maximalwert von 65535 und es muss erneut inkrementiert werden, so läuft der Zähler über und wird wieder bei 0 beginnen.

Randbedingungen und Einschränkungen:

- Der Baustein darf nur einmal instanziiert werden
- Für Sercos Parameter kann nur der Parametersatz 0 gelesen werden
- Es ist nur möglich Sercos Parameter als 16-Bit "Ident Number" (IDN) zu lesen. Das Lesen von "Extended Ident Number" (EIDN) ist nicht möglich.
- Es kann nur das Sercos Element 7 (Daten) bei Sercos Parametern gelesen werden

Die Daten können vom Master mittels Index und Subindex angefragt werden. Folgende Liste gibt die entsprechenden Werte für Index/Subindex an.

Startindex (16Bit)	Endindex (16Bit)	Subindex (8Bit)	Rechenvorschrift
16#0000	16#1FFF		Reserviert
16#2000	16#2FFF	Drive # (1-64)	Index = 16#2000 + S-Parameter (nur Parametersatz 0 möglich, keine EIDN)
16#3000	16#3FFF	Drive # (1-64)	Index = 16#3000 + P-Parameter (nur Parametersatz 0 möglich, keine EIDN)
16#4000	16#4FFF	Axis # (1-64)	Index = 16#4000 + A-Parameter (nur Parametersatz 0 möglich, keine EIDN)
16#5000	16#5AEC		Frei
16#5AED	16#5AED	16#01 – 16#FF	Index = 16#5AED; Subindex = Custom Data Array Element; (Element 1 - 255)
16#5AEE	16#5AF7	16#01 – 16#FF	Index = (16#5AEE + [(WORD Array Element - 1) / 255]; Subindex = (WORD Array Element) – [(Index - 16#5AEE) * 255]; (Element 1 - 2295)
16#5AF8	16#5AFF	16#01 – 16#FF	Index = (16#5AF8 + [(STRING Array Element - 1) / 255]; Subindex = (STRING Array Element) – [(Index - 16#5AF8) * 255]; (Element 1 - 2040)

16#5B00	16#5CEB	16#00 – 16#FF	Frei
16#5CEC	16#5CFF	16#01 – 16#FF	Index = (16#5CEC + [(REAL Array Element - 1) / 255]; Subindex = (REAL Array Element) – [(Index - 16#5CEC) * 255]; (Element 1 - 4845)
16#5D00	16#5D13	16#01 – 16#FF	Index = (16#5D00 + [(DINT Array Element - 1) / 255]; Subindex = (DINT Array Element) – [(Index - 16#5D00) * 255]; (Element 1 - 4845)
16#5D14	16#5FFF	16#00 – 16#FF	Frei
16#6000	16#6FFF	16#00	Index = 16#6000 + C-Parameter (nur Parametersatz 0 möglich, keine EIDN)
16#7000	16#7FFF	Kinematic # (1-16)	Index = 16#7000 + K-Parameter (nur Parametersatz 0 möglich, keine EIDN)
16#8000	16#8FFF	Probe # (1-99)	Index = 16#8000 + M-Parameter (nur Parametersatz 0 möglich, keine EIDN)
16#9000	16#9FFF	PLS # (1-4)	Index = 16#9000 + N-Parameter (nur Parametersatz 0 möglich, keine EIDN)
16#A000	16#AFFF	Osci # (1-16)	Index = 16#A000 + O-Parameter (nur Parametersatz 0 möglich, keine EIDN)
16#B000	16#FFFF		Frei

Tab. 11-4: Zugewiesene Adressbereiche für Index / Subindex

Fehlerbehandlung

Bei der Fehlerbehandlung muss man zwischen zwei verschiedenen Fehlertypen unterscheiden.

- Fehler beim Datenzugriff und der Datenübertragung
- Fehler in der Konfiguration des Bausteins

Fehler in der Konfiguration (z.B. Parameterkanal zu kurz/lang) werden am Baustein selbst als Fehler angezeigt und müssen durch das SPS-Programm ausgewertet werden.

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0001	"FieldbusAddressIn" ist nicht belegt oder hat den Wert 0.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0002	"ChannelLengthIn" ist nicht belegt oder hat einen ungültigen Wert (gültige Werte liegen im Bereich $6 \leq \text{"ChannelLengthIn"} \leq 18$).
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0003	"FieldbusAddressOut" ist nicht belegt oder hat den Wert 0.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0004	"ChannelLengthOut" ist nicht belegt oder hat einen ungültigen Wert (gültige Werte liegen im Bereich $6 \leq \text{"ChannelLengthOut"} \leq 18$).
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0005	"SizeofRealArray" hat einen ungültigen Wert (gültige Werte liegen im Bereich $0 \leq \text{"SizeofRealArray"} \leq 19380$).
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0006	"SizeofDintArray" hat einen ungültigen Wert (gültige Werte liegen im Bereich $0 \leq \text{"SizeofDintArray"} \leq 19380$).
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0007	"SizeofWordArray" hat einen ungültigen Wert (gültige Werte liegen im Bereich $0 \leq \text{"SizeofWordArray"} \leq 4590$).

ErrorID	Additional1	Additional2	Beschreibung
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0008	"SizeOfStringArray" hat einen ungültigen Wert (gültige Werte liegen im Bereich $0 \leq \text{"SizeOfStringArray"} \leq 522240$).
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0009	"SizeofCustomData" ist größer als der zulässige Wert (2088960).
INPUT_RANGE_ERROR (16#0001)	16#0002	16#000A	"SizeofRealArray" hat einen ungültigen Wert. "SizeofRealArray" muss durch 4 ohne Rest teilbar sein.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#000B	"SizeofDintArray" hat einen ungültigen Wert. "SizeofDintArray" muss durch 4 ohne Rest teilbar sein.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#000C	"SizeofWordArray" hat einen ungültigen Wert. "SizeofWordArray" muss durch 2 ohne Rest teilbar sein.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#000D	"SizeOfStringArray" oder "SizeofString" hat einen ungültigen Wert. "SizeOfStringArray" muss durch "SizeofString" + 1 ohne Rest teilbar sein.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#000E	"SizeofCustomData" oder "SizeofCustom" hat einen ungültigen Wert. "SizeofCustomData" muss durch "SizeofCustom" ohne Rest teilbar sein.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#000F	"SizeofCustom" hat einen ungültigen Wert (gültige Werte liegen im Bereich $0 \leq \text{"SizeofCustom"} \leq 8192$).
STATE_MACHINE_ERROR (16#0005)	16#0006	16#000x	Es wurde ein ungültiger Zustand innerhalb des Bausteins erkannt. Baustein muss neu gestartet werden.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0010	Der Pointer in "FieldbusAddressIn" oder die Länge "ChannelLengthIn" hat sich geändert.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0011	Der Pointer in "FieldbusAddressOut" oder die Länge "ChannelLengthOut" hat sich geändert.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0012	Der Pointer in "RealArray" oder die Länge "SizeofRealArray" hat sich geändert.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0013	Der Pointer in "DintArray" oder die Länge "SizeofDintArray" hat sich geändert.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0014	Der Pointer in "WordArray" oder die Länge "SizeofWordArray" hat sich geändert.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0015	Der Pointer in "StringArray", die Länge "SizeOfStringArray" oder die Länge "SizeofString" hat sich geändert.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0016	Der Pointer in "CustomArray", die Länge "SizeofCustomArray" oder die Länge "SizeofCustom" hat sich geändert.

Tab. 11-5: Fehlercodes FB IL_ParameterChannel

Fehler beim Datenzugriff (z.B. nicht vorhandenen Index Lesen/Schreiben) und der Datenübertragung (z.B. falsche Längenangabe) werden nicht am Baustein Ausgang sichtbar. Diese werden innerhalb des Parameterkanals als Fehler gekennzeichnet und Fehlerwerte angegeben. Der Master muss diese Fehlerwerte auswerten.

ErrorCode	Text
16#8500	Die übertragene Datenmenge ist zu groß für den internen Zwischenpuffer
16#8B00	Die Formatangabe im Steuerwort ist falsch (muss 16#0C sein)

ErrorCode	Text
16#8C00	Die angegebene Länge ist größer als der Parameterkanal selbst
16#9500	Es wurden Daten mitgesendet, obwohl eine Leseanforderung gemacht wurde
16#9600	Bit "C1" ist im Steuerwort nicht gesetzt
16#9700	Das "Last" Bit wurde nicht gesendet, obwohl es sich um eine Leseanforderung handelt
16#1F00	Steuerungsspezifischer Fehler. Weitere Fehlercodes (siehe Fehlertabelle der Steuerung) folgen in den nächsten vier Worten. Im Wort 2 und 3 ist der Fehlercode "Additional1" und in Wort 4 und 5 der Fehlercode "Additional2" enthalten.
16#F200	Der angegebene Index ist unbekannt
16#F300	Der angegebene Subindex ist unbekannt

Tab. 11-6: Fehlercodes innerhalb des Parameterkanals, die vom FB IL_ParameterChannel erzeugt werden können.

11.3 IL_ParameterChannel – Protokoll, Fehlerbehandlung und Inbetriebnahme

11.3.1 Beschreibung des unterlagerten Protokolls

Parameterkanal im zyklischen Datenkanal

Das verwendete Protokoll im Parameterkanal verwendet lediglich ein Steuerwort, um vom Master Daten an den Slave zu senden.

Damit der Slave Daten an den Master senden kann, verwendet er das Statuswort.

Der Master kann entweder Daten schreiben oder Daten lesen. Dabei kann nur der Master eine Datenübertragung anstoßen. Steuer- und Statuswort sind immer fest im Parameterkanal enthalten, wie dies im folgendem Bild zu sehen ist:

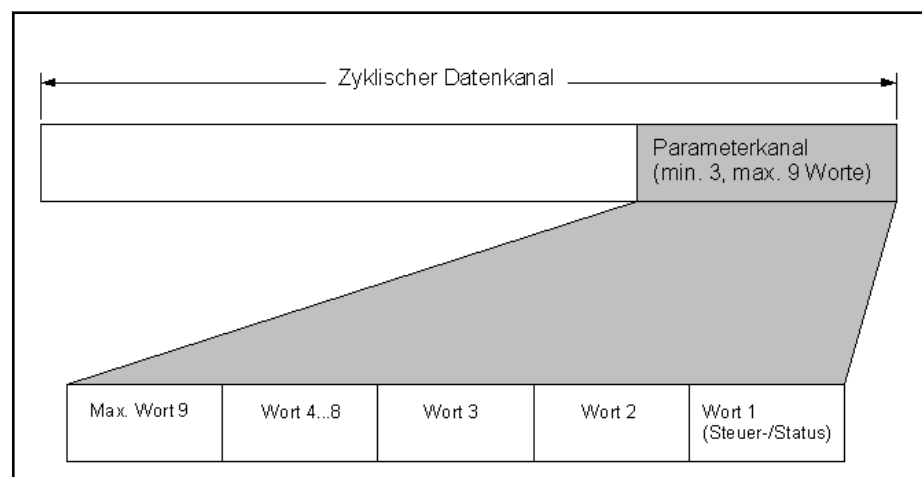


Abb. 11-2: Dateninhalt des Feldbus mit Parameterkanal

Steuer-/Statuswort

Das Steuerwort hat folgende Belegung:

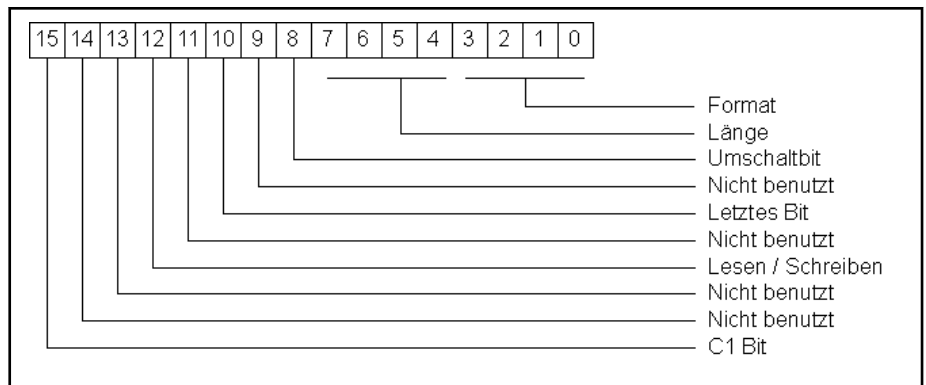


Abb. 11-3: Belegung des Steuerwortes im Parameterkanal

Das Statuswort hat folgende Belegung:

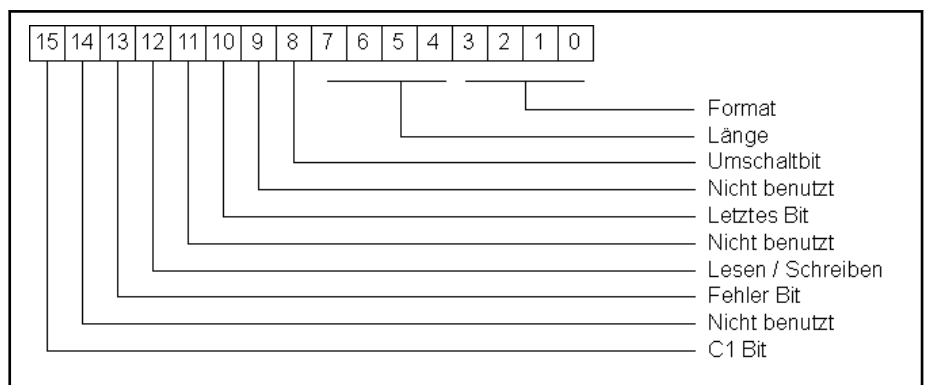


Abb. 11-4: Belegung des Statuswortes im Parameterkanal

Die einzelnen Bits haben folgende Bedeutung:

- Format:** Diese Bits geben Verwendungszweck und Bedeutung der folgenden Datenworte im Parameterkanal an. Ihr Wert ist, außer bei Abbruch und Initialisierung der Kommunikation, auf 1100b fixiert.
- Länge:** Diese vier Bits spezifizieren die Länge der gültigen Daten in Bytes, das Steuerwort nicht mitgerechnet. Die Daten im Rest des Parameterkanals sind nicht definiert.
- Umschaltbit (T):** Dieses Bit wird immer dann umgeschaltet, wenn ein neuer Satz Daten gesendet wird. Es stellt einen Handshake zwischen Master und Slave her. Der Master darf dieses Bit nur dann umschalten, wenn das Umschalt-Bit im Statuswort denselben Zustand hat wie das im Steuerwort gesendete Umschalt-Bit.
- Letztes Bit (L):** Das letzte Bit wird gesetzt, wenn der letzte Teil des Datenblocks gesendet wird. Es wird verwendet, um ein Fragmentieren zu ermöglichen.
- Lese-Schreib-Bit (R/W):** "Lesen" entspricht dem Bitwert 1, der besagt, dass der Master Daten lesen will. Soll geschrieben werden, muss dieses Bit 0 sein.
- C1:** Dieses Bit muss aus Kompatibilitätsgründen zu Altsystemen fest auf 1 gesetzt werden.
- Fehlerbit (E):** Dieses Bit signalisiert das Auftreten eines Fehlers im Slave. Die Fehlerursache (Fehlercode) wird mit den folgenden Daten angegeben.



Nicht benutzte Bits haben den Wert 0.

Index und Subindex

Bei der ersten Anforderung durch den Master ist nach dem Steuerwort ein Wort für den Index und ein weiteres Wort für den Subindex vorgesehen. Darüber können alle verfügbaren Daten adressiert werden. Die Zuordnung von Index/Subindex zu den Steuerungsparametern ist wie folgt:

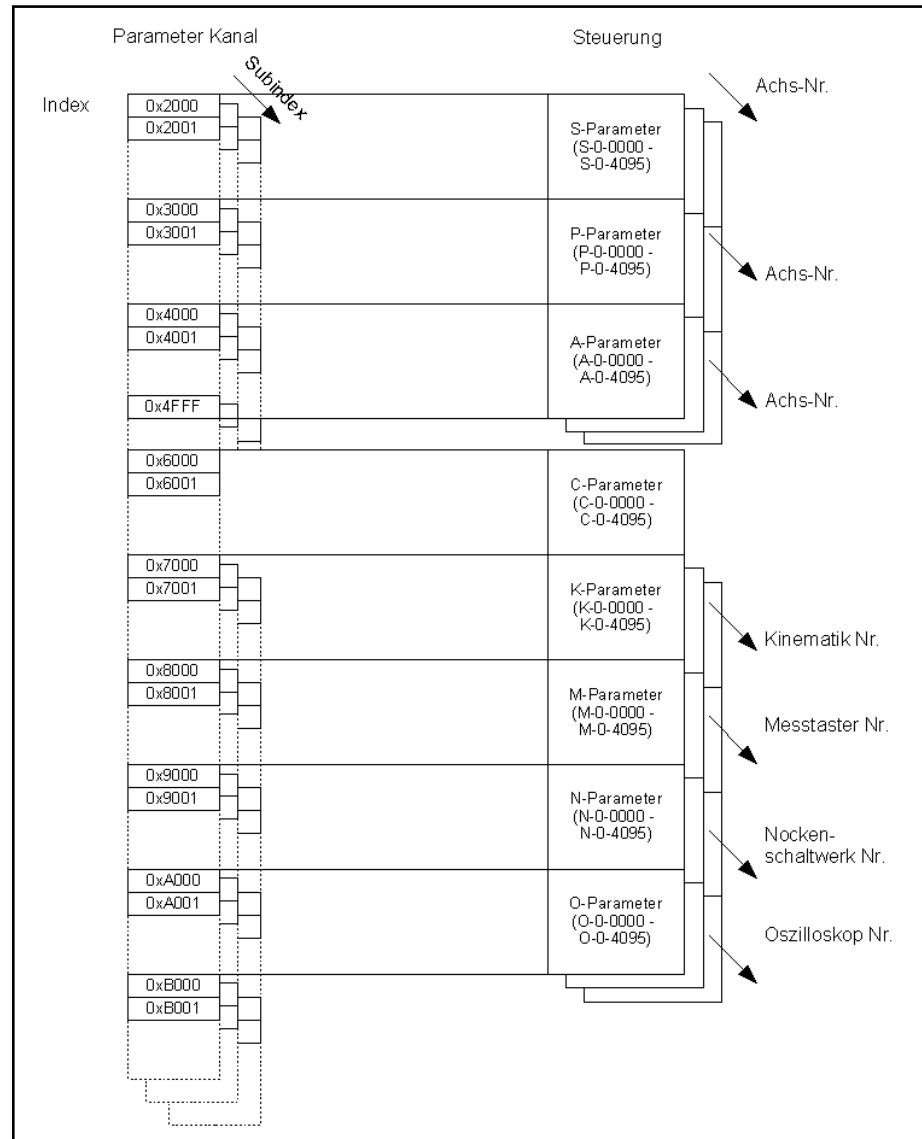


Abb. 11-5: Zuordnung von Index/Subindex zu Parametern

Die Zuordnung ist sehr einfach. Um von einem Parameter auf den gewünschten Index/Subindex zu kommen, ist lediglich die Parameternummer zum Startoffset des Indexes zu addieren. Der Subindex ergibt sich aus der jeweiligen Nummer der Achse, Kinematik, etc.

Beispiel:

Es soll der Achsparameter A-0-0100 der Achse 22 gelesen werden.

In diesem Fall muss für den Index $16\#4000 + 100 = 16\#4064$ und für den Subindex die Achsnummer = 22 gewählt werden.

Die Zuordnung von Index/Subindex zu den SPS-Variablen ist nachfolgend dargestellt. Wichtig hierbei ist, dass der Subindex0 nicht benutzt wird und in der Graphik auch nicht dargestellt ist.

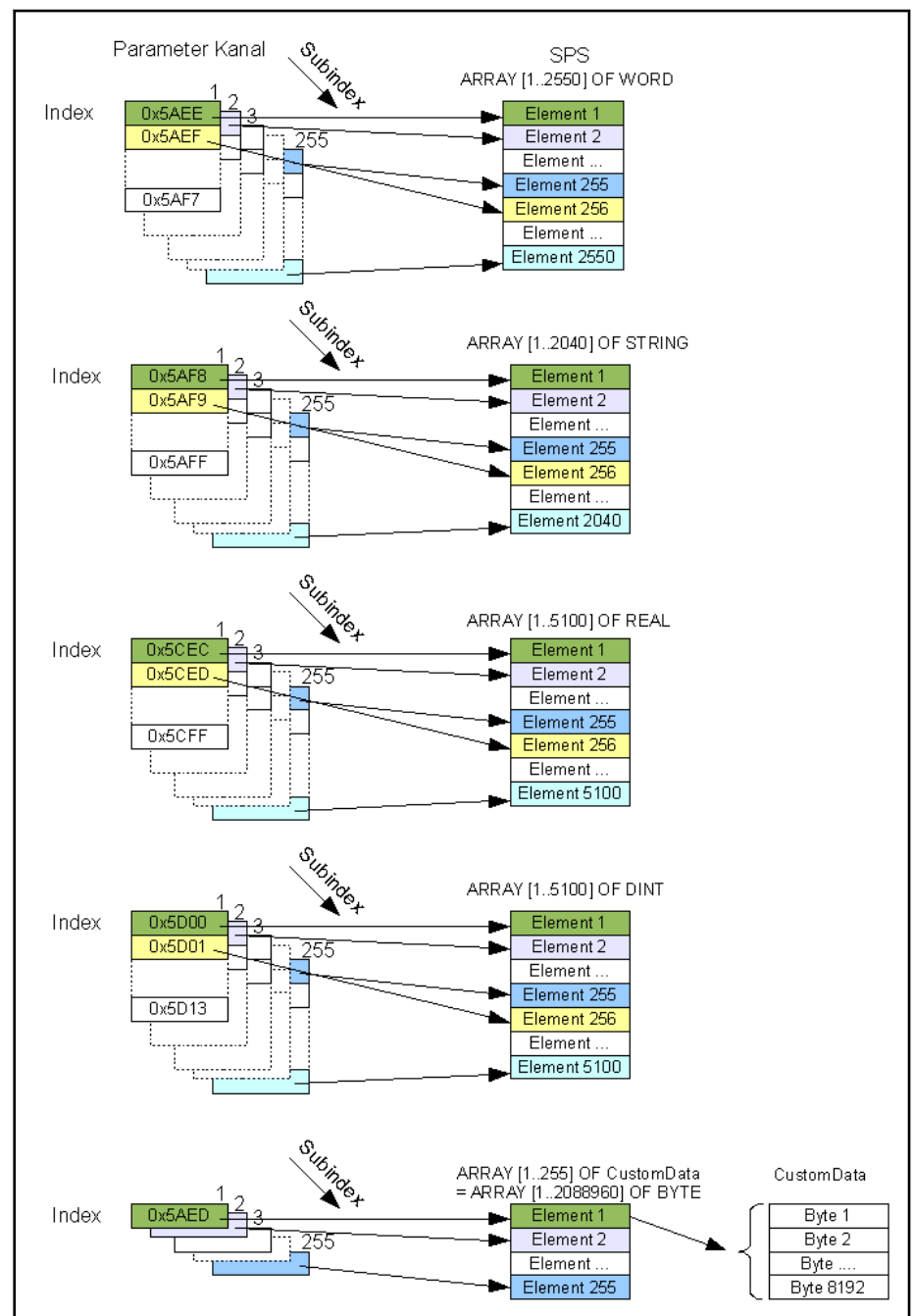


Abb. 11-6: Zuordnung von Index/Subindex zu SPS-Variablen

Hier ist die Abbildungsvorschrift etwas umfangreicher. Um von einer gewünschten Elementennummer auf den Index/Subindex zu kommen, können folgende Umrechnungsformeln für STRING, REAL, DINT und WORD Datentypen verwendet werden:

Index = Startindex + [(Elementnummer - 1) / 255] (die Division ist eine ganzzahlige Division, d.h. Ergebnis wird ohne Nachkommastellen berechnet)

Subindex = Elementnummer - [(Index - Startindex) * 255]

Beispiel:

Es soll das 257. Element des REAL-Feldes gelesen werden.

In diesem Fall ist der Startindex 16#5CEC und die Elementnummer ist 257.

Damit ergibt sich für den Index:

$$\text{Index} = 16\#5\text{CEC} + (257 - 1) / 255 = 16\#5\text{CEC} + (256 / 255) = 16\#5\text{CEC} + 1 = 16\#5\text{CED}$$

Für den Subindex ergibt sich:

$$\text{Subindex} = 257 - [(16\#5\text{CED} - 16\#5\text{CEC}) * 255] = 257 - [1 * 255] = 2$$

Um vom Index/Subindex auf die Elementnummer umzurechnen gilt folgende Formel:

$$\text{Elementnummer} = [(\text{Index} - \text{Startindex}) * 255] + \text{Subindex}$$

Beispiel:

Welches Element wird durch Index 16#5D02 und Subindex 3 angefordert?

Der Index liegt im Bereich 16#5D00 bis 16#5D13, deshalb handelt es sich um ein Element im DINT Feld. Die Elementnummer lautet:

$$\text{Elementnummer} = [(16\#5\text{D02} - 16\#5\text{D00}) * 255] + 3 = 2 * 255 + 3 = 513$$

Für die CUSTOM-Daten gilt die gleiche Abbildungsvorschrift wie oben, da jedoch nur ein Index verwendet wird, kann sie entsprechend vereinfacht werden:

$$\text{Index} = \text{Startindex} = 16\#5\text{AED}$$

$$\text{Subindex} = \text{Elementnummer}$$

Um im SPS-Programm die Berechnung in umgekehrter Reihenfolge zu ermöglichen (Ermitteln der Elementnummer aus Index und Subindex), kann folgende Formeln verwendet werden:

$$\text{Elementnummer} = (\text{Index} - \text{Startindex}) * 255 + \text{Subindex}$$

Beispiel:

Der Funktionsbaustein zeigt an seinem Ausgang, dass auf Index = 16#5AF9 und Subindex 5 geschrieben wurde. Auf welches Element wurde dadurch geschrieben?

16#5AF9 liegt im Bereich 16#5AF8 bis 16#5AFF. Es handelt sich daher um einen STRING.

Die Elementnummer ergibt sich dann aus:

$$\text{Elementnummer} = (16\#5\text{AF9} - 16\#5\text{AF8}) * 255 + 5 = 16\#1 * 255 + 5 = 260$$

Es wurde auf das 260. Element innerhalb des "STRING-Arrays" geschrieben.

Initialisierung der Kommunikation

Um die Kommunikation zu initialisieren, oder auch eine bestehende Anfrage abubrechen, kann der Master eine spezielle Anforderung senden. Diese hat im Steuerwort den Inhalt 2#1000 000x 0000 1111. Der Wert des Umschaltbit (T) ist dabei unerheblich.

Sendet der Master dieses Steuerwort, so bricht der Slave jegliche Abarbeitung ab und versetzt sich in den Zustand, dass er eine neue Anforderung bearbeiten kann. Er antwortet im Statuswort mit Inhalt 2#1000 000x 0000 1111.

Fehlertelegramme

Wenn der Slave einen Fehler ermittelt, so sendet er ein Antworttelegramm an den Master, in dem das Fehlerbit (E) im Statuswort gesetzt ist. Im zweiten Wort ist dann ein Fehlercode enthalten, der die Ursache spezifiziert (siehe dazu auch [Fehlerbehandlung, Seite 714](#)).

Eine Besonderheit hierbei ist der Fehlercode 16#1F00. Dieser spezifiziert einen Fehler der innerhalb der Steuerung auftrat. In diesem Fall enthält Wort 3 und 4 einen Fehlercode der Steuerung, der in der Dokumentation der Steuerung enthalten ist. In Wort 5 und 6 sind zusätzliche Informationen zum Fehler enthalten. Sollte der Parameterkanal kürzer als 6 Worte sein, so wird das Telegramm fragmentiert.

Beispiel:

Ablauf einer Leseanforderung

Im nachfolgenden Beispiel wird von folgenden Voraussetzungen ausgegangen:

- Parameterkanal ist in beiden Richtungen 8 Byte = 4 Worte lang
- Das Umschaltbit zu Beginn den Wert 0 hat
- Es soll eine SPS-STRING-Variable (Element 1) angefordert werden. Damit muss als Index 16#5AF8 und Subindex 1 gewählt werden
- In diesem Fall antwortet der Slave mit nur einem Telegramm. Ist die Antwort länger, so wäre "Letztes Bit" im Statuswort nicht gesetzt und der Master müsste mit weiteren Telegrammen solange anfordern bis "Letztes Bit" im Statuswort gesetzt ist

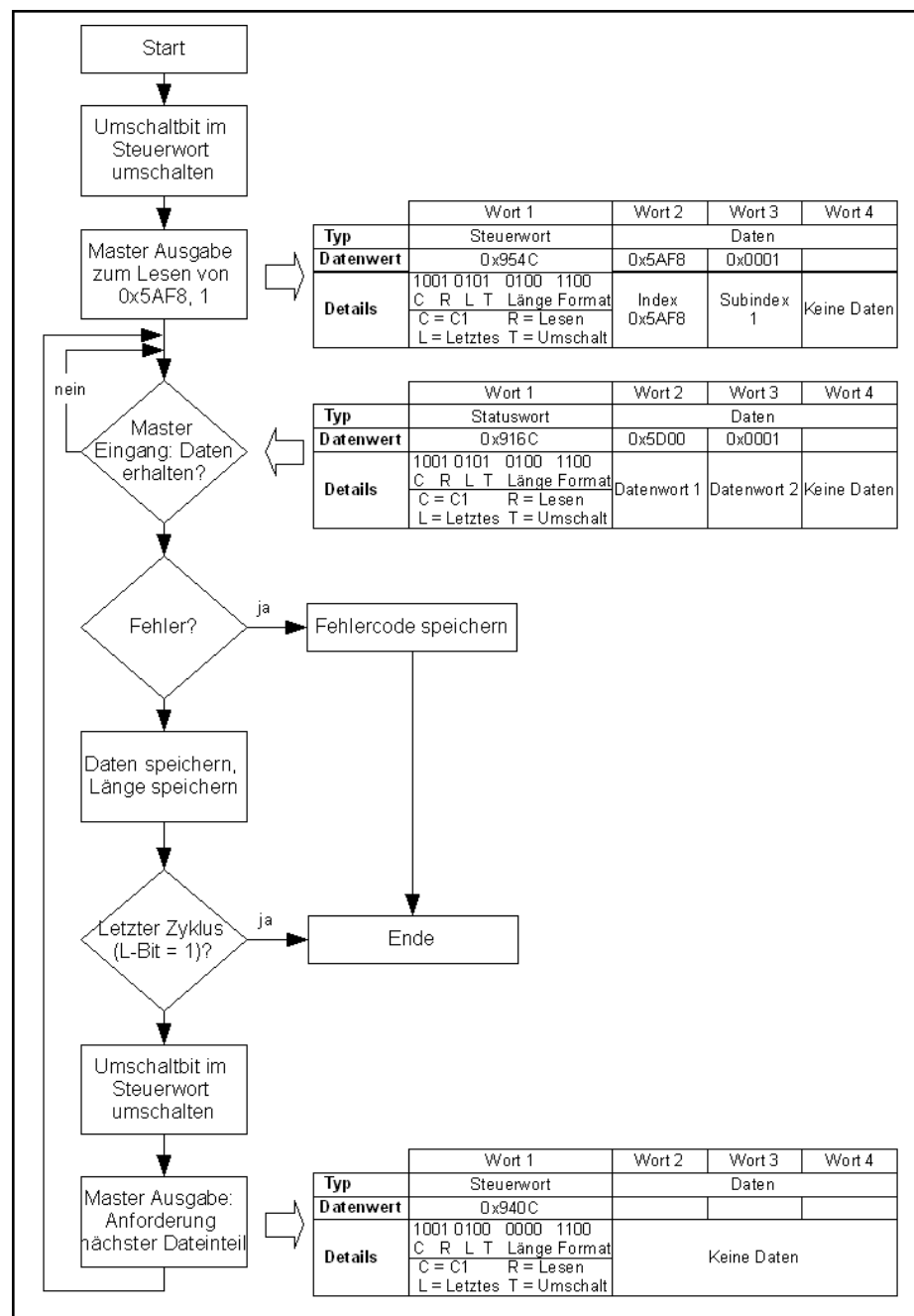


Abb. 11-7: Beispiel zum Lesen einer SPS-STRING-Variablen

Beispiel:**Ablauf einer Schreibanforderung**

Im nachfolgenden Beispiel wird von folgenden Voraussetzungen ausgegangen:

- Parameterkanal ist in beiden Richtungen 8 Byte = 4 Worte lang
- Das Umschaltbit zu Beginn den Wert 0 hat
- Es soll der Achsparameter "A-0-0002, Achsname" der Achse 22 geschrieben werden. Damit muss als Index 16#4002 und Subindex 22 gewählt werden

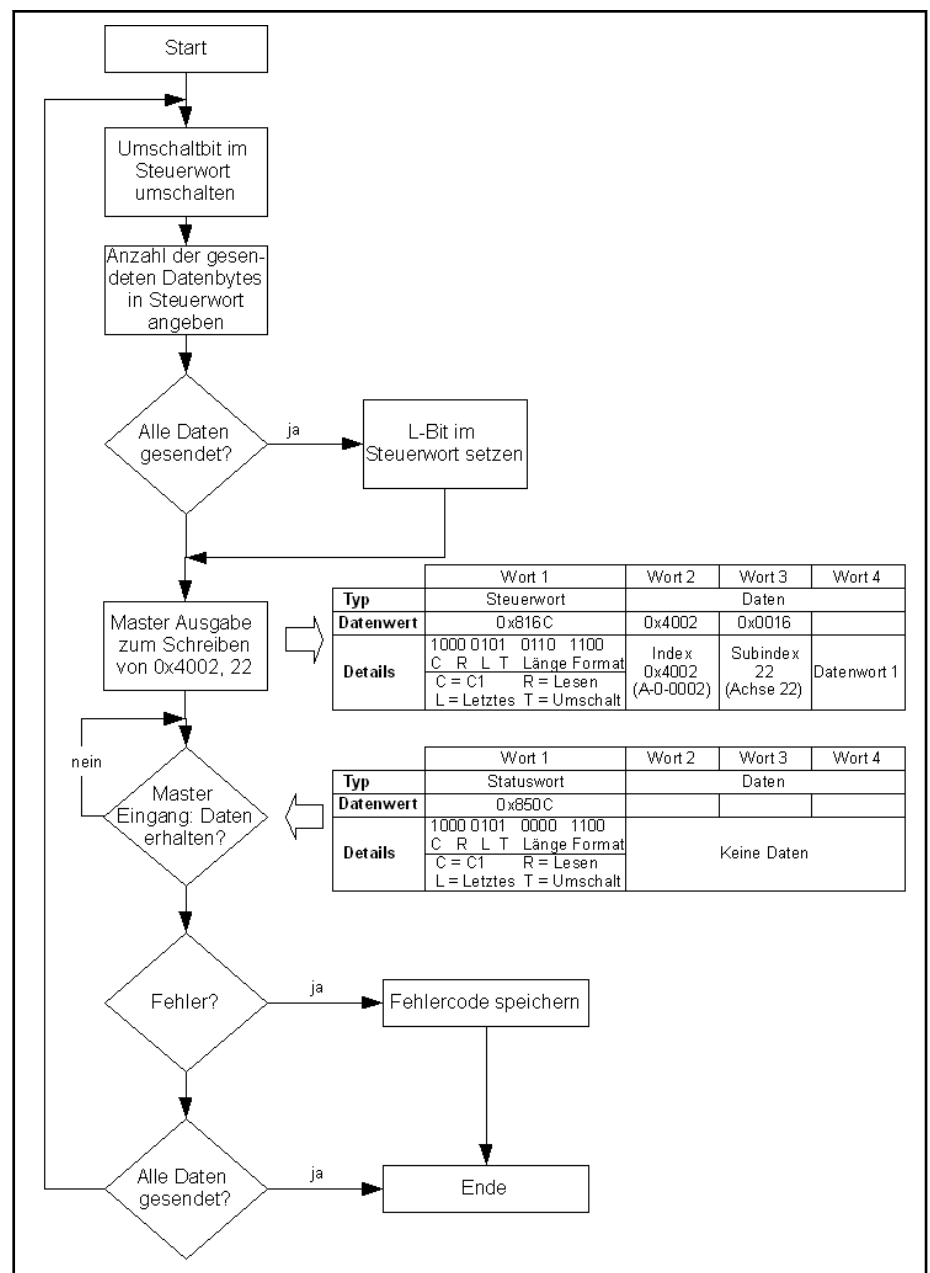


Abb. 11-8: Beispiel zum Schreiben eines Achsparameters

11.3.2 Inbetriebnahme des Bausteins

Bevor der Baustein benutzt werden kann, muss eine Feldbusverbindung bestehen. Am Beispiel des PROFIBUS wird hier gezeigt, wie dieser zum Betrieb konfiguriert wird.

Bei PROFIBUS werden einzelne Module konfiguriert. Für den Parameterkanal wird empfohlen, eigene Module für den Parameterkanal zu verwenden um auch hier eine Strukturierung zu erreichen. Es bieten sich daher Module der Größe 8 und 16 Byte an (Parameterkanal kann 6 bis 18 Bytes lang sein). Alternativ können auch nur Teile eines Moduls verwendet werden. Dies ist dann aber bei der Beschaltung des Parameterkanal-Bausteins zu berücksichtigen. Wichtig ist, dass es sich bei den eingefügten Modulen um Module mit Byte Ausrichtung handelt. Im nachfolgenden Bild sind für den Parameterka-

nal je ein Modul mit 16 Byte als Ein- und Ausgänge definiert ("ParameterChannel_In" und "ParameterChannel_Out").

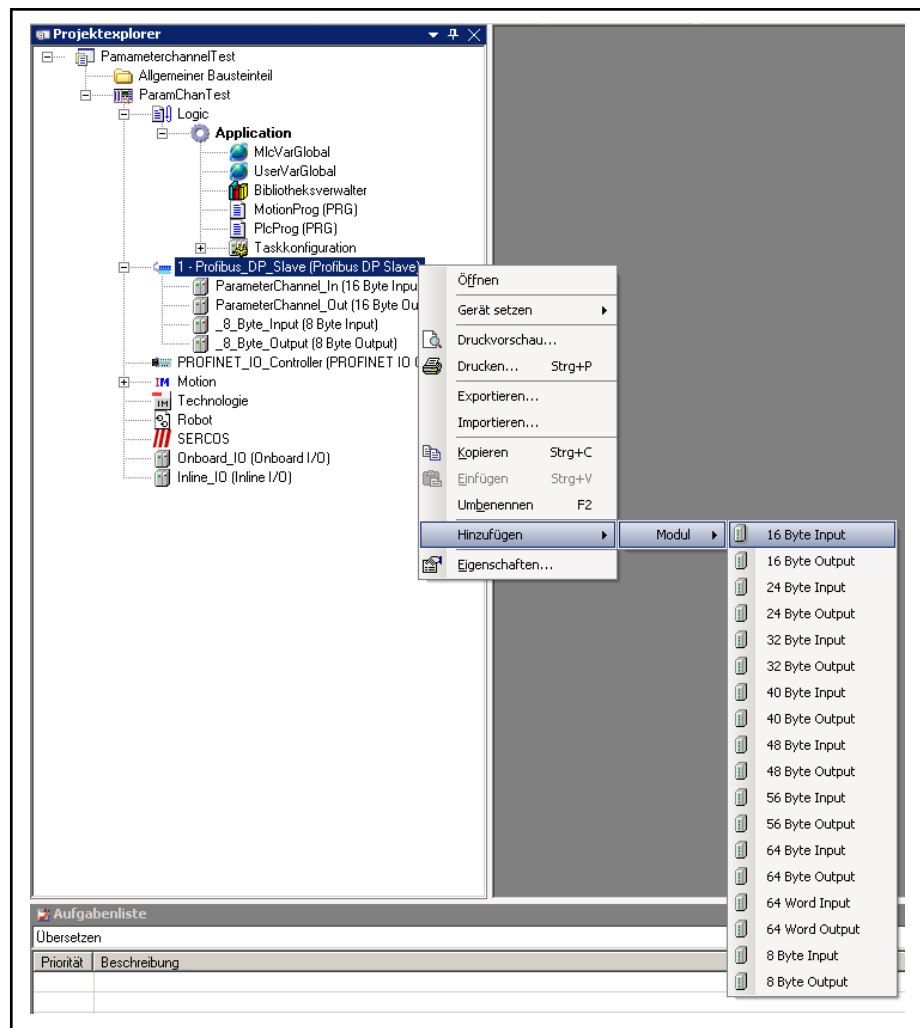


Abb. 11-9: PROFIBUS-Module zur Verwendung als Parameterkanal

Um dem Baustein betreiben zu können, müssen die Adressen für den Feldbus und die jeweilige Länge angegeben werden. Dazu gibt es mehrere Möglichkeiten. Es kann ein symbolischer Name vergeben werden, oder die Adresse des IO-Bereiches benutzt werden.

Im folgenden Bild wurde für die Ausgangsdaten des Feldbusmoduls "ParameterChannel_Out" der Name "myFieldbusOut" vergeben. Dies geschieht durch Doppelklick auf das gewünschte Modul und im sich öffnenden Fenster im Reiter "DP-Module I/O Abbild". Bei anderen Feldbussen hat der Reiter ggf. andere Namen. Alternativ kann auch die Adresse verwendet werden (hier "%QB1").

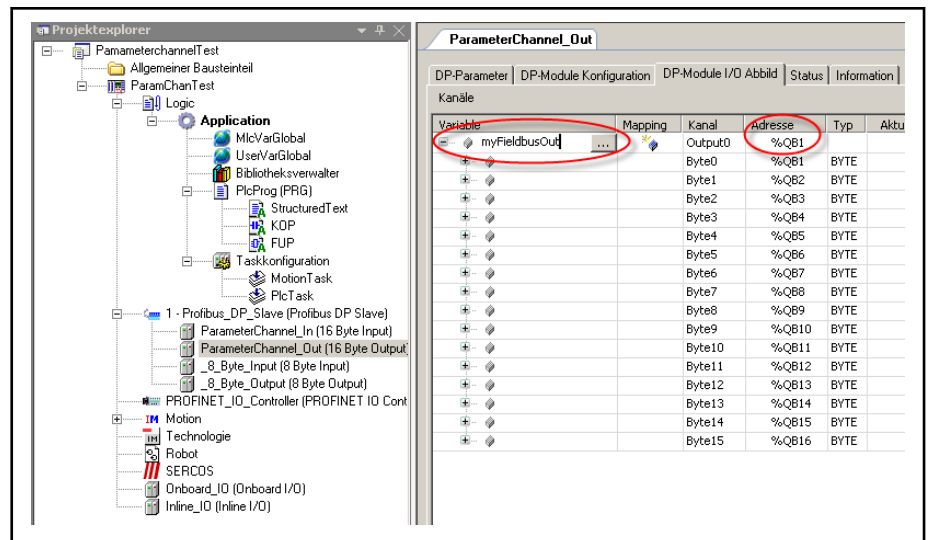


Abb. 11-10: Name und Adresse eines PROFIBUS-Moduls

Für die Beschaltung des IL_ParameterChannel Bausteins sind neben den Standard Ein- und Ausgängen noch die Eingänge für SPS-Objekte zu zuweisen. Dabei wird jeweils ein Zeiger sowie die Länge der zugewiesenen Daten benötigt. Eine Besonderheit sind hier die STRING- und CUSTOM-Daten, da diese wiederum selbst anwenderspezifische Längen haben können. Deshalb muss hier auch noch die Länge eines Strings angegeben werden. Wichtig ist in jedem Fall, dass die Felder jeweils bei 1 beginnen. Im nachfolgenden Quelltext sind Beispiele für die Deklaration der SPS-Variablen dargestellt. Für die Felder sind jeweils die maximalen Längen angegeben. Auch sind die Standard-Ein- und Ausgänge, sowie der Funktionsbaustein selbst deklariert.

Deklaration eines FB "IL_ParameterChannel" mit Datensatz

```

TYPE MY_DATA_STRUCTURE :
STRUCT
    // Custom structure, here 2048 * 4 = 8kB Data
    StructureData: ARRAY [1..2048] OF UDINT;
END_STRUCT
END_TYPE

VAR
    // Function block declaration
    Fb: IL_ParameterChannel;
    // PLC Data
    myRealArray:    ARRAY [1..4845] OF REAL;
    myDintArray:    ARRAY [1..4845] OF DINT;
    myWordArray:    ARRAY [1..2295] OF WORD;
    myStringArray:  ARRAY [1..2040] OF STRING(255);
    myCustomData:   ARRAY [1..255] OF MY_DATA_STRUCTURE;
    // In-/outputs of function block
    En:             BOOL;
    InOp:           BOOL;
    Error:          BOOL;
    ErrorID:        ERROR_CODE;
    ErrorIdent:     ERROR_STRUCT;
    Index:          WORD;
    Subindex:       WORD;
    Reads:          UDINT;
    Writes:         UDINT;
END_VAR

```

Der Baustein-Aufruf erfolgt dann wie folgt:

```

1
2  Fb( Enable:= En,
3      FieldbusAddressIn:= %IB1,
4      ChannelLengthIn:= 16,
5      FieldbusAddressOut:= ADR(myFieldbusOut),
6      ChannelLengthOut:= 16,
7      SizeofRealArray:= SIZEOF(myRealArray),
8      SizeofDintArray:= SIZEOF(myDintArray),
9      SizeofWordArray:= SIZEOF(myWordArray),
10     SizeofStringArray:= SIZEOF(myStringArray),
11     SizeofCustomData:= SIZEOF(myCustomData),
12     RealArray:= ADR(myRealArray[1]),
13     DintArray:= ADR(myDintArray[1]),
14     WordArray:= ADR(myWordArray[1]),
15     StringArray:= ADR(myStringArray[1]),
16     CustomData:= ADR(myCustomData[1]),
17     SizeofString:= 255,
18     SizeofCustom:= SIZEOF(MY_DATA_STRUCTURE),
19     InOperation=> InOp,
20     Error=> Error,
21     ErrorID=> ErrorID,
22     ErrorIdent=> ErrorIdent
23     Index=> Index,
24     Subindex=> Subindex,
25     NumberOfReads=> Reads,
26     NumberOfWrites=> Writes);
27

```

Abb. 11-11: Verwendung des Funktionsbausteins IL_ParameterChannel im strukturierten Text

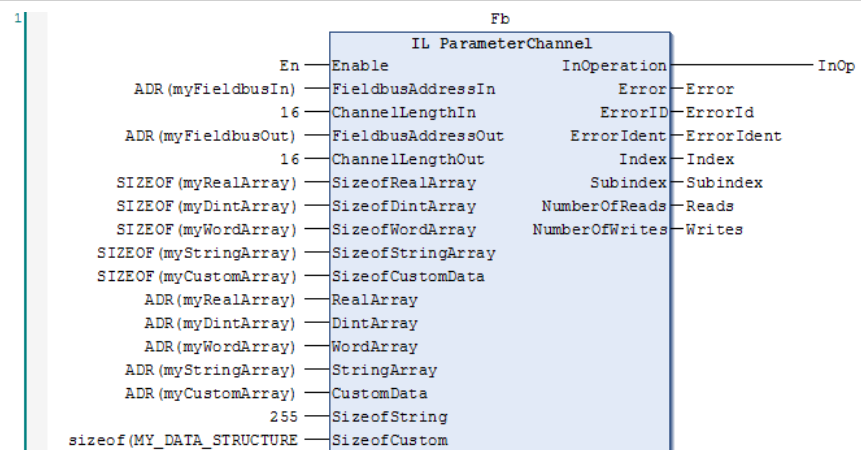


Abb. 11-12: Verwendung des Funktionsbausteins IL_ParameterChannel im FUP

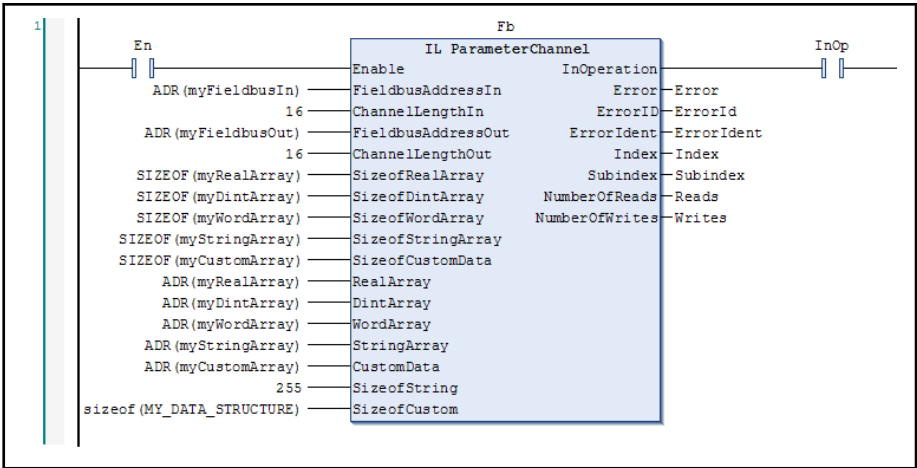


Abb. 11-13: Verwendung des Funktionsbausteins IL_ParameterChannel im KOP
In den obigen Beispielen wurden für die Adressen der Feldbusdaten jeweils im strukturierten Text die Methode mit symbolischer Adresse ("myFieldbusOut") bzw. mit direkter Adressierung ("%IB1") verwendet.

11.3.3 Unterschiede zu Altsystemen

Allgemeines

Der Baustein dient als Ersatz für das bereits bestehende System VisualMotion. Bestehende Unterschiede werden hier beschrieben.

Auf dem Feldbus-Slave

Um den Parameterkanal betreiben zu können, muss auf dem Slave zunächst der Feldbus so eingerichtet werden, dass zusätzliche zyklische Daten für den Parameterkanal konfiguriert werden. Dies war auch bei dem Altsystem der Fall.

Zusätzlich müssen bei der MLC/XLC der Funktionsbaustein IL_ParameterChannel instanziiert, sowie unterschiedliche Felder angelegt werden.

Der Baustein darf nur einmal instanziiert werden. Er wird mit Setzen des "Enable"-Eingangs gestartet und verhält sich im Bezug auf die Standardausgänge ("InOperation", "Error", etc.) genauso wie andere MLC/XLC-Funktionsbausteine.

Die anzulegenden Felder repräsentieren die Variablen und Register in VisualMotion. Wichtig dabei ist, dass die Felder beim Index 1 beginnen! Die Felder von Strings und CustomData sind zusätzlich vorhanden und hatten keine vergleichbaren Variablen in VisualMotion.

MLC/XLC		VisualMotion	
Deklaration	Datentyp	Name	Datentyp
myReals: ARRAY [1..4845] of REAL	REAL (32-bit)	Program Floats (4845 Stück)	FLOAT (32-bit)
myDints: ARRAY [1..4845] of DINT	DINT (32-bit)	Program Ints (4845 Stück)	INT (32-bit)
myWord: ARRAY [1..2295] of WORD	WORD (16-bit)	Register (2295 Stück)	INT (16-bit)
myString: ARRAY [1..255] of STRING	STRING (myStringSize)	--	--

Tab. 11-7: Gegenüberstellung der SPS-Daten MLC/XLC / VisualMotion

Auf dem Feldbus-Master

Auf der Seite des Masters können bestehende Bausteine weiter verwendet werden. Allerdings müssen Anpassungen in der Beschaltung des Bausteins erfolgen.

Die Adressierung über "Index" und "Subindex" wurde geändert, bzw. erweitert.

Die S- und P-Parameter sind identisch mit dem System VisualMotion.

Für Register, Program Floats und Program Ints ändern sich lediglich die Dateninhalte:

- "Index" 16#5AEE – 16#5AF7 adressierte in VisualMotion die Register, jetzt wird auf das WORD-Feld in der SPS zugegriffen
- "Index" 16#5CEC – 16#5CFF adressierte in VisualMotion die Program Floats, jetzt wird auf das REAL-Feld in der SPS zugegriffen
- "Index" 16#5D00 – 16#5D13 adressierte in VisualMotion die Program Integers, jetzt wird auf das DINT-Feld in der SPS zugegriffen

Datentypen und die Berechnung von "Index" und "Subindex" bleiben aber identisch.

Die A-Parameter wurden kompatibel erweitert. Hier waren bisher nur A-Parameter bis zur Nummer 2047 möglich. Mit diesem Funktionsbaustein kann auch auf A-Parameter bis Nummer 4096 zugegriffen werden.

Alle anderen Daten sind freie oder zusätzliche Daten, die neu hinzu gekommen sind.

Startindex (16Bit)	Endindex (16Bit)	Subindex (8Bit)	Rechenvorschrift
16#0000	16#1FFF		reserviert
16#2000	16#2FFF	Drive # (1-64)	Index = 16#2000 + S-Parameter
16#3000	16#3FFF	Drive # (1-64)	Index = 16#3000 + P-Parameter
16#4000	16#4FFF	Axis # (1-64)	Index = 16#4000 + A-Parameter
16#5000	16#5AEC		Frei
16#5AED	16#5AED	16#01 – 16#FF	Index = 16#5AED Subindex = Custom Data Array Element (Element 1 - 255)
16#5AEE	16#5AF7	16#01 – 16#FF	Index = (16#5AEE + [(WORD Array Index - 1) / 255]) Subindex = (WORD Array Index) – [(Index - 16#5AEE) * 255] (Index 1 - 2295)
16#5AF8	16#5BEC	16#01 – 16#FF	Index = (16#5AF8 + [(STRING Array Index - 1) / 255]) Subindex = (STRING Array Index) – [(Index - 16#5AF8) * 255] (Index 1 - 244)
16#5BED	16#5CEB	16#00 – 16#FF	Frei
16#5CEC	16#5CFF	16#01 – 16#FF	Index = (16#5CEC + [(REAL Array Index - 1) / 255]) Subindex = (REAL Array Index) – [(Index - 16#5CEC) * 255] (Index 1 - 4845)
16#5D00	16#5D13	16#01 – 16#FF	Index = (16#5CEC + [(DINT Array Index - 1) / 255]) Subindex = (DINT Array Index) – [(Index - 16#5CEC) * 255] (Index 1 - 4845)
16#5D14	16#5FFF	16#00 – 16#FF	Frei

16#6000	16#6FFF	16#00	Index = 16#6000 + C-Parameter
16#7000	16#7FFF	Kinematic # (1-16)	Index = 16#7000 + K-Parameter
16#8000	16#8FFF	Probe # (1-99)	Index = 16#8000 + M-Parameter
16#9000	16#9FFF	PLS # (1-4)	Index = 16#9000 + N-Parameter
16#A000	16#AFFF	Osci # (1-16)	Index = 16#A000 + O-Parameter
16#B000	16#FFFF		Frei

Tab. 11-8: Zugewiesene Adressbereiche für Index/Subindex

Die Fehlercodes innerhalb des Parameterkanals haben sich (teilweise) geändert. Die Unterschiede werden in der folgenden Tabelle dargestellt:

ErrorCode MLC/XLC	ErrorCode VisualMotion	Beschreibung
16#8500	16#8500	Daten zu lang (VisualMotion: >128 Byte; MLC > 64kByte)
--	16#8800	Fehler auf MLC/XLC nicht vorhanden
16#8B00	16#8B00	Identisch (die Formatangabe im Steuerwort ist falsch (muss 16#0C sein))
--	16#8D00	Fehlerfall wird auf der MLC/XLC bereits beim Start des Bausteins geprüft
16#8C00	16#8C00	Identisch (die angegebene Länge ist größer als der Parameterkanal selbst)
--	16#9000	Format-Bits 0 bis 3 des Steuerwortes haben sich während der Übertragung geändert. Dieser Fehler wird auf der MLC/XLC durch die Fehlernummer 16#8B00 angezeigt
16#9500	16#9500	Identisch (es wurden Daten mit geschickt, obwohl eine Leseanforderung gemacht wurde)
16#9600	--	C1 bit ist nicht gesetzt. Fehler ist auf VisualMotion nicht vorhanden, da hier die alten Formate des Parameterkanals noch unterstützt werden
16#9700	--	Das "Last" Bit wurde nicht geschickt, obwohl es sich um eine Leseanforderung handelt. Dieser Fehler kann nur auftreten, wenn eine Leseanforderung innerhalb eines Fragmentes übertragen werden kann. Bei VisualMotion war dies nicht garantiert, da der Parameterkanal hier auch nur 2 Worte groß sein durfte. Auf der MLC/XLC ist allerdings festgelegt, dass der Parameterkanal mindestens 3 Worte groß sein muss. Damit kann eine Leseanforderung immer mit einem Telegramm übertragen werden
16#1F00	16#1Fxx	Steuerungsspezifischer Fehler. Bei VisualMotion wird der steuerungsspezifische Fehler im unteren Byte angegeben. Bei der MLC/XLC folgen nach dem Fehlercode noch 8 weitere Bytes. Diese beinhalten die Fehlercodes der unterlagerten Funktionsbausteine zum Lesen/Schreiben von Parametern. In Wort 3 und 4 des Parameterkanals ist der "Additional1" Fehlercode enthalten, im Wort 5 und 6 der "Additional2"

ErrorCode MLC/XLC	ErrorCode VisualMotion	Beschreibung
16#F200	16#F200	Identisch (der angegebene Index ist unbekannt)
16#F300	16#F300	Identisch (der angegebene Subindex ist unbekannt)

Tab. 11-9: Unterschiede von MLC/XLC- und VisualMotion-Fehlercodes innerhalb des Parameterkanals

12 ML_JobComm.library

12.1 Einführung und Übersicht

Die Bibliothek ML_JobComm ermöglicht den Versand von bis zu 255 Byte großen Telegrammen, hier Jobs genannt, an andere Steuerungen.

Die Datenübertragung erfolgt über das SCL-Protokoll, welches auf UDP aufsetzt. Die Übertragungsschicht stellt sicher, dass die Daten korrekt und in der richtigen Reihenfolge beim Empfänger ankommen.

Aufgrund der einstellbaren maximalen Übertragungszeit für Jobs ist die Job-Kommunikation echtzeitfähig. Es wird sichergestellt, dass ein Datenpaket innerhalb der geforderten Zeit beim Empfänger ankommt. Wird dies nicht erreicht, wird der Sender informiert.

- Vorteile** Durch die Nutzung der Job-Kommunikation ergeben sich folgende Vorteile:
- Die Datenübertragung ist echtzeitfähig, d.h. ein Job muss in einer vorgegebenen Zeit beim Kommunikationspartner ankommen
 - Die Datenübertragung ist durch eine Prüfsumme abgesichert
 - Die Reihenfolge der Jobs wird sichergestellt
 - Alle Jobs erreichen den Kommunikationspartner
 - Der Sender erkennt Übertragungsfehler
 - Es kann mit mehreren Kommunikationspartnern kommuniziert werden, ohne dass sich diese gegenseitig beeinflussen
 - Jobs können dezentral an mehreren Stellen im SPS-Programm gesendet oder empfangen werden
 - Jobs können anhand einer Signatur aus dem Empfangspuffer gefiltert empfangen werden
 - Die Implementierung erlaubt die gleichzeitige Verwendung in mehreren SPS-Tasks

Kommunikation Die Kommunikation über die Job-Kommunikationsschicht zwischen mehreren Steuerungen erfolgt über Ethernet.

Hierbei wird auf das UDP-Protokoll aufgesetzt. Es wird ein freier UDP-Port auf der Steuerung benötigt. Als Standardport verwendet die JobComm den Port 7101. Es ist möglich, andere Ports zu verwenden.

Die folgenden Ports sollten nicht verwendet werden:

- Ports mit Portnummer kleiner oder gleich 1024, da diese vom Betriebssystem für zahlreiche Dienste verwendet werden.
- Port 1740: SPS, IndraLogic
- Port 6040: ComServer

Sowohl das Senden als auch der Empfang von Datenpaketen erfolgt über einen UDP-Port. Es wird kein zweiter UDP-Port geöffnet.

12.2 Architektur der JobComm-Bibliothek

Die Bibliothek JobComm stellt mehrere Funktionsbausteine zur Verfügung. Alle Funktionsbausteine können mehrfach instanziiert werden. Die Auswertung der Eingänge sowie die Ansteuerung der Ausgänge der Funktionsbausteine erfolgt entsprechend den allgemeinen Regeln für Funktionsbausteine von Bosch Rexroth.

Zentrale Kommunikation	Die Hauptarbeit übernimmt der zentrale Funktionsbaustein ML_SendRecvJobAss (ML_SendRecvJobAss, Seite 734). Der Baustein sollte mit hoher Priorität und kurzer Zykluszeit in einer eigenen SPS-Task laufen. Ohne den zyklischen Aufruf dieses Bausteins ist keine Datenübertragung möglich. Er stellt einen Handle zur Verfügung, über den die anderen Funktionsbausteine auf die Instanz zugreifen können. Alle Operationen der anderen Funktionsbausteine werden über diesen Handle ausgelöst.
Verbindungen	Bevor Job-Telegramme zwischen zwei Kommunikationspartnern übertragen werden können, muss eine Verbindung aufgebaut werden. Eine Instanz des Funktionsbausteins ML_SendRecvJobAss kann mehrere Verbindungen verwalten. Die Verbindungen werden durch den Funktionsbaustein ML_CtrlJobConnect gesteuert (ML_CtrlJobConnect, Seite 742). Ein Verbindungsversuch kann aktiv oder passiv erfolgen. Bei einem aktivem Verbindungsaufbau (Command = ML_CONNECT_CMD_JOB) wird versucht den Kommunikationspartner direkt zu kontaktieren. Bei einem passivem Verbindungsaufbau (Command = ML_SYNC_CMD_JOB) wird darauf gewartet, dass die Gegenstelle einen aktiven Verbindungsaufbau versucht. Der Abbau einer Verbindung erfolgt ebenfalls durch den Funktionsbaustein ML_CtrlJobConnect (Command = ML_DISCONNECT_JOB). Da der eigentliche Verbindungsauf- bzw. -abbau durch die Instanz des Funktionsbausteins ML_SendRecvJobAss erfolgt, kommt der Funktionsbaustein ML_CtrlJobConnect nicht sofort mit einem Ergebnis zurück. Vielmehr sind in der Regel mehrere Aufrufe notwendig, bevor ein Ende der Aktion signalisiert wird. Der Funktionsbaustein ML_ShowJobConnect ruft die Liste der bekannten Verbindungen mit dem jeweiligen Verbindungsstatus ab. Hierdurch kann der Verbindungsstatus der Steuerung leicht überprüft werden.
Senden	Der Funktionsbaustein ML_SendJobTlg dient dem Senden von Jobs an einen Kommunikationspartner (ML_SendJobTlg, Seite 737). Bei der Abarbeitung des Bausteins wird der Job lediglich in den Sendepuffer übertragen. Der eigentliche Versand erfolgt in der Instanz des Funktionsbausteins ML_SendRecvJobAss. Kann ein Job innerhalb einer vorgegebenen Zeitspanne nicht gesendet werden, so wird es auf den Sende-Junkbuffer verschoben. Die Ursache für einen nicht erfolgten Versand kann eine fehlende Verbindung zum Kommunikationspartner sein.
Empfangen	Der Funktionsbaustein ML_RecvJobTlg durchsucht den Empfangspuffer nach einem Datenpaket mit der angegebenen Signatur. Der eigentliche Empfang der Datenpakete erfolgt in der Instanz des Funktionsbausteins ML_SendRecvJobAss. Es wird immer nur ein Paket gelesen. Wird ein Job innerhalb einer vorgegebenen Zeitspanne nicht aus dem Empfangspuffer abgeholt, so wird es auf den Empfangs-Junkbuffer verschoben. Dies passiert in der Regel, wenn kein Aufruf des Funktionsbausteins ML_RecvJobTlg auf die Signatur des Jobs passt.
Fehleranalyse	Um die Fehleranalyse zu erleichtern, ermöglichen die Funktionsbausteine ML_DebugJobConnect bzw. ML_DebugJobConnect_Cyclic eine Ausgabe der internen Puffer der ML_SendRecvJobAss-Instanz (ML_DebugJobConnect, Seite 747 bzw. ML_DebugJobConnect_Cyclic, Seite 750).

Die folgenden Puffer können eingesehen werden:

- SendJobBuff: enthält alle sendebereiten Jobs
- RecvJobBuff: enthält alle empfangenen und noch nicht abgeholten Jobs
- SendJobAssBuff: enthält das aktuell zu sendende SCL-Paket
- SendUdpBuff: enthält das aktuell zu sendende SCL-Paket
- RecvUdpBuff: enthält das zuletzt empfangene SCL-Paket

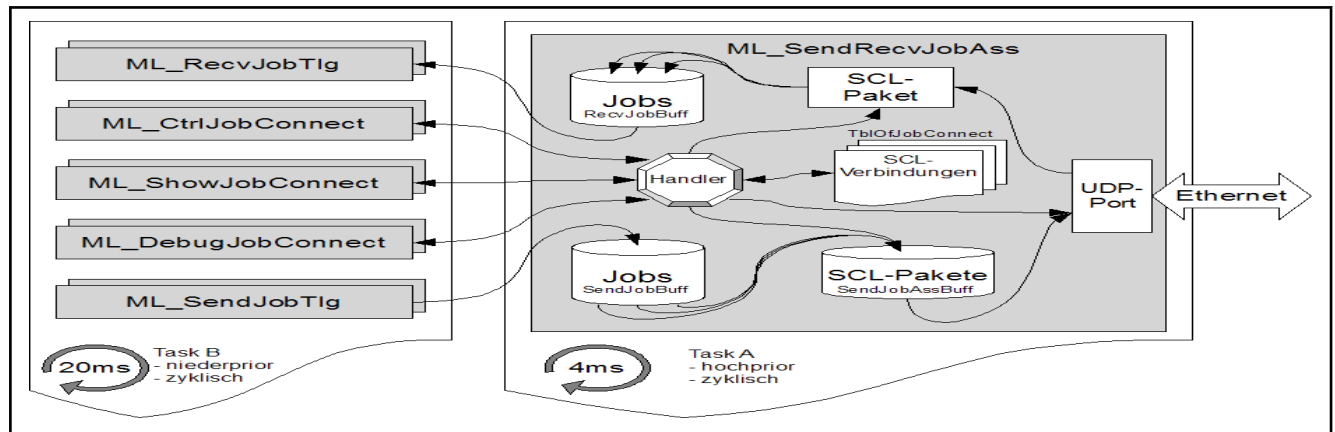


Abb. 12-1: Kommunikationsmodell der ML_JobComm

12.3 Funktionsbausteine

12.3.1 Einführung und Übersicht

Die Bibliothek "ML_JobComm" implementiert das SCL-Protokoll. Die Implementation wurde so gestaltet, dass ein möglichst leichter Zugriff auf die zu übertragenden Daten möglich ist. Jobs können jederzeit gesendet werden. Empfangene Jobs können anhand einer Signatur aus dem Empfangspuffer geholt werden. Dies ermöglicht eine dezentrale Bearbeitung der Jobs.

Die Funktionsbausteine von ML_JobComm

Funktionsbausteine	Beschreibung
ML_SendRecvJobAss, Seite 734	dient zum Vorbereiten, Senden, Empfangen und Nachbereiten von zusammengefassten Aufträgen (Jobs)
ML_SendJobTlg, Seite 737	dient der Kommandierung eines Jobs in den Ausgangspuffer für Job-Telegramme
ML_RcvJobTlg, Seite 739	dient der definierten Suche und dem Auslesen eines Jobs aus dem Eingangspuffer für Job-Telegramme
ML_CtrlJobConnect, Seite 742	dient der lokalen Verwaltung einer bidirektionalen Port-Verbindung ermöglicht die Kommandierung des Verbindungsauf- und -abbaus sowie das Beobachten des Status einer Verbindung
ML_ShowJobConnect, Seite 745	dient dem einmaligen Lesen des Status aller lokalen bidirektionalen Port-Verbindungen
ML_DebugJobConnect, Seite 747	dient der einmaligen Übertragung der Beobachtungsdaten aus den internen Puffern
ML_DebugJobConnect_Cyclic, Seite 750	dient der zyklischen Übertragung der Beobachtungsdaten aus den internen Puffern

Tab. 12-1: Bibliothek ML_JobComm

12.3.2 ML_SendRecvJobAss

Kurzbeschreibung Der Funktionsbaustein ML_SendRecvJobAss dient zum Vorbereiten, Senden, Empfangen und Nachbereiten von zusammengefassten Aufträgen (Jobs). Der Funktionsbaustein ist für den Einsatz in einer hochprioren, zyklisch zeitgesteuerten SPS Anwender-Task vorgesehen.

Zuordnung Zielsystem/ Bibliothek

Zielsystem	Bibliothek
MLC ab 10VRS	ML_JobComm
MLD/MPx18	MX_JobComm.lib
MLD/MPC07	MX_JobComm.lib

Tab. 12-2: Referenztablette Funktionsbaustein ML_SendRecvJobAss

Schnittstellenbeschreibung

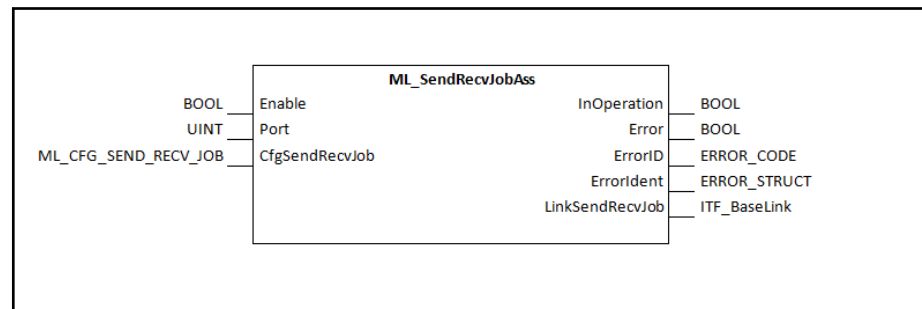


Abb. 12-2: Funktionsbaustein ML_SendRecvJobAss

	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe
	Port	UINT	UDP-Port für Kommunikation
	CfgSend-RecvJob	ML_CFG_SEND_RECV_JOB	Konfiguration der Job-Telegramme und Verbindungen ML_CFG_SEND_RECV_JOB, Seite 754
VAR_OUTPUT	InOperation	BOOL	TRUE, wenn die zyklische Kommunikation aktiv ist. Der Ausgang LinkSendRecvJob ist gültig.
	Error	BOOL	Zeigt an, dass ein Fehler innerhalb des Funktionsbausteins aufgetreten ist.
	ErrorID	ERROR_CODE	Fehlerkennung
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe Fehlercodes, Seite 736 .
	LinkSend-RecvJob	ITF_BaseLink	Handle auf die Instanz des Funktionsbausteins

Tab. 12-3: Schnittstelle ML_SendRecvJobAss

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
Port	UINT	0	65535	7101	steigende Flanke von "Enable"

Tab. 12-4: Eingangsverhalten ML_SendRecvJobAss

Funktionsbeschreibung Der Funktionsbaustein ML_SendRecvJobAss dient zum Vorbereiten, Senden, Empfangen und Nachbereiten von zusammengefassten Aufträgen (Jobs). Der Funktionsbaustein ist für den Einsatz in einer hochprioren, zyklisch zeitgesteuerten SPS-Anwendertask vorgesehen.

Initialisierung

Bei steigender Flanke an "Enable" werden alle lokalen und globalen Puffer gelöscht, der vorgegebene Port und die Konfigurationsdaten CfgSendRecvJob auf Gültigkeit geprüft. Anschließend wird der UDP-Port geöffnet. Nach erfolgreichem Abschluss und Bereitschaft zur zyklischen Job-Kommunikation ist der Ausgang "InOperation" TRUE.

Normalbetrieb

Bei aktivem "Enable"-Eingang werden im zyklischen Aufruf die folgenden Schritte ausgeführt:

1. Prüfung auf Zeitüberschreitung der Verbindungen und UDP-Telegramme

Alle Verbindungen der Tabelle TbOfJobConnect werden auf Zeitüberschreitung geprüft. Fehlerhafte Verbindungen werden den aufsynchronisierten Instanzen des Bausteins ML_CtrlJobConnect signalisiert und aus der Liste der offenen Verbindungen TbOfJobConnect ohne koordinierten Verbindungsabbau gelöscht. UDP-Telegramme, deren Verbindung gelöscht wurde, werden im Sendepuffer SendJobAssBuff als ungültig gekennzeichnet. Eine fehlerhafte Verbindung hat keine weitere Auswirkung auf andere Verbindungen der Instanz des freigegebenen Funktionsbausteins.

2. Prüfung auf Zeitüberschreitung Job-Telegramme

Alle gültigen Job-Telegramme werden auf Zeitüberschreitung geprüft. Job-Telegramme mit überschrittener Verweildauer werden aus dem jeweiligen Puffer gelöscht, d. h. als ungültig gekennzeichnet. Der Header eines gelöschten Job-Telegramms sowie die entsprechende IP wird auf den vom Anwender vorgesehenen Speicher (ML_CFG_JUNK_JOB) kopiert. Das Telegrammfragment enthält die IP Adresse (4 Byte) sowie eine konfigurierbare Anzahl Bytes des Job-Telegramms. Der über den Eingang "CfgSendRecvJob" adressierte Speicher muss eine Mindestgröße von (4+Anzahl der JunkBuffer-Größe) Byte besitzen sowie ein ganzzahliges Vielfaches der Mindestgröße sein. Die Größe des Telegrammfragments wird in der Konfigurationsstruktur ML_CFG_JUNK_JOB bestimmt. Der Default-Wert ist 12.

3. Prüfung und Bearbeitung der Kommandopuffer

Aus den Kommandopuffern wird zyklisch jeweils ein Kommando entnommen und bearbeitet.

4. Zusammenstellung und Übertragung von Job-Telegrammen

Abhängig von einer zugehörigen offenen Verbindung werden die Job-Telegramme aus dem Sendepuffer ausgewertet, zu UDP-Telegrammen zusammengesetzt und in den UDP-Telegrammpuffer übertragen.

5. Lesen des Ethernet-Empfangspuffers

Der Ethernet-Empfangspuffer wird ausgelesen und die Telegramme bzgl. Aufbau und Inhalt ausgewertet. Der Lesezugriff wird solange wiederholt, bis der Ethernet Empfangspuffer geleert bzw. die maximale Anzahl der Lesezugriffe erreicht wurde.

Die empfangenen Telegramme bedingen abhängig vom Inhalt und Zustand der Verbindung (Daten, ACK, SYN, SYN ACK, FIN, FIN ACK, RST, ...) unterschiedliche Reaktionen wie Einlagerung von Jobs, direkte, einmalige Reaktionstelegramme oder wiederholbare Reaktionstelegramme. Enthält das UDP-Telegramm Aufträge (Jobs), so werden diese in den Empfangspuffer übertragen. Zusammengesetzte UDP-Telegramme, d. h. Reaktionstelegramme mit Aufträgen, werden als ungültig verworfen.

6. Übertragung der UDP-Telegramme

Nach der Prüfung auf Zeitüberschreitung werden, abhängig von den Elementen der Verbindungstabelle, alle gültigen UDP-Telegramme versendet.

7. Defragmentierung der Sende- und Empfangspuffer für Job-Telegramme

Nach der Deaktivierung des Bausteins über "Enable" bzw. im Fehlerfall werden keine weiteren Telegramme entgegen genommen und geöffnete Verbindungen ohne koordinierten Verbindungsabbau geschlossen.

Port

Am Eingang "Port" wird die zu verwendende UDP-Portnummer angegeben. Dieser Port darf nicht anderweitig verwendet werden. Ist der Port bereits belegt oder nicht nutzbar, generiert der Funktionsbaustein bei steigender Flanke an "Enable" einen entsprechenden Fehlercode. Verwendbare UDP-Ports siehe [Kommunikation, Seite 731](#).

CfgSendRecvJob

An CfgSendRecvJob wird die Konfiguration für die Kommunikation angelegt ([ML_CFG_SEND_RECV_JOB, Seite 754](#)).

Fehlercodes

Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR, 16#0001	16#0000F790	16#00900001	Der Eingang "Port" ist ungültig.
INPUT_INVALID_ERROR, 16#0001	16#0000F790	16#00900002	Fehlerhafte Konfiguration der Junk-Eigenschaften des Sendepuffers
INPUT_INVALID_ERROR, 16#0001	16#0000F790	16#00900003	Fehlerhafte Konfiguration der Junk-Eigenschaften des Empfangspuffers
DEVICE_ERROR, 16#0008	16#0000F790	16#00910001	Einlagerung eines Job-Telegramms fehlgeschlagen - Pufferüberlauf im Empfangspuffer für Job-Telegramme
DEVICE_ERROR, 16#0008	16#0000F790	16#0095nnnn	Fehler beim Anlegen eines Kommunikations-Sockets (create) - Fehlercode Additional2, LowWord nnnn siehe Betriebssystem
DEVICE_ERROR, 16#0008	16#0000F790	16#0096nnnn	Fehler beim Schließen eines Kommunikations-Sockets (close) - Fehlercode Additional2, LowWord nnnn siehe Betriebssystem
DEVICE_ERROR, 16#0008	16#0000F790	16#0097nnnn	Fehler bei der Konfiguration eines Kommunikations-Sockets (ioctl) - Fehlercode Additional2, LowWord nnnn siehe Betriebssystem
DEVICE_ERROR, 16#0008	16#0000F790	16#0098nnnn	Fehler während des BIND Prozesses eines Kommunikations-Sockets (bind) - Fehlercode Additional2, LowWord nnnn siehe Betriebssystem
DEVICE_ERROR, 16#0008	16#0000F790	16#0099nnnn	Fehler beim Empfang von Daten aus dem Ethernet Empfangspuffer (recvfrom) - Fehlercode Additional2, LowWord nnnn siehe Betriebssystem

Tab. 12-5: Fehlercodes des Funktionsbausteins ML_SendRecvJobAss

12.3.3 ML_SendJobTlg

Kurzbeschreibung Der Funktionsbaustein ML_SendJobTlg dient der Kommandierung eines Jobs in den Ausgangspuffer für Job-Telegramme. Der Funktionsbaustein ist für den Einsatz in einer zyklisch zeitgesteuerten SPS Anwendertask mittlerer Priorität vorgesehen.

Zuordnung Zielsystem/ Bibliothek

Zielsystem	Bibliothek
MLC ab 10VRS	ML_JobComm
MLD/MPx18	MX_JobComm.lib
MLD/MPC07	MX_JobComm.lib

Tab. 12-6: Referenztablette Funktionsbaustein ML_SendJobTlg

Schnittstellenbeschreibung

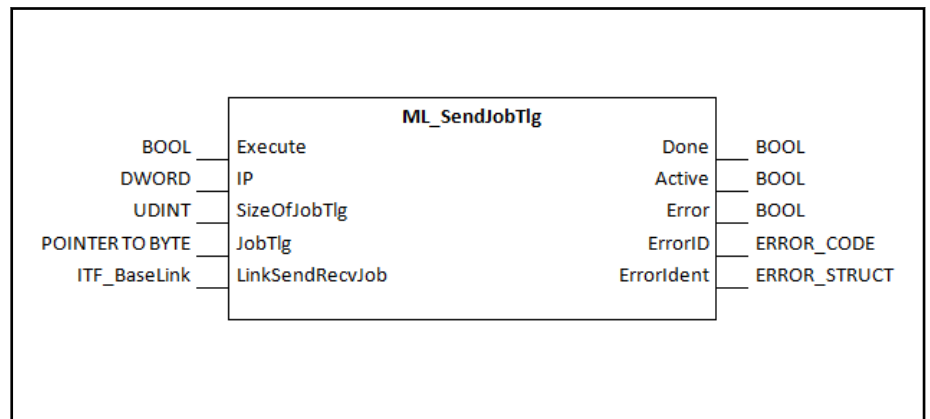


Abb. 12-3: Funktionsbaustein ML_SendJobTlg

	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe
	IP	DWORD	IP-Adresse des Remote-Gerätes
	SizeOfJobTlg	UDINT	Anzahl der gültigen Bytes im Datenbereich "JobTlg", welcher den zu übertragenden Auftrag enthält
	JobTlg	POINTER TO BYTE	Zeiger auf den Datenbereich, welcher den zu übertragenden Auftrag enthält
	LinkSend-RecvJob	ITF_BaseLink	Handle einer "ML_SendRecvJobAss"-Instanz ML_SendRecvJobAss, Seite 734
VAR_OUTPUT	Done	BOOL	Übertragung des Auftrags ohne Fehler abgeschlossen
	Active	BOOL	Übertragung des Auftrags aktiv
	Error	BOOL	Zeigt an, dass ein Fehler innerhalb des Funktionsbausteins aufgetreten ist.
	ErrorID	ERROR_CODE	Fehlerkennung
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe Fehlercodes, Seite 738 .

Tab. 12-7: Schnittstelle ML_SendJobTlg

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
IP	DWORD	0	16#FFFFFFFF	0	steigende Flanke von "Execute"

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
SizeOfJobTlg	UINT	1	255	0	steigende Flanke von "Execute"
JobTlg	POINTER TO BYTE			0	steigende Flanke von "Execute"
LinkSendRecvJob	ITF_BaseLink			0	steigende Flanke von "Execute"

Tab. 12-8: Eingangsverhalten ML_SendJobTlg

Funktionsbeschreibung

Der Funktionsbaustein ML_SendJobTlg dient der Kommandierung eines Jobs in den Ausgangspuffer für Job-Telegramme SendJobBuff. Der Funktionsbaustein ist für den Einsatz in einer zyklisch zeitgesteuerten SPS Anwender-Task mittlerer Priorität vorgesehen.

Bei steigender Flanke an "Execute" überträgt der Funktionsbaustein einmalig die Daten des durch die Eingänge "JobTlg" und "SizeOfJobTlg" definierten Auftrags in den Ausgangspuffer für Job-Telegramme. Eine aktivierte Übertragung kann nicht unterbrochen und durch eine weitere ersetzt werden (keine Retriggerunterstützung).

Die fehlerfreie Kommandierung des Auftrags in den Ausgangspuffer für Job-Telegramme wird durch den Ausgang "Done" signalisiert. Der Ausgang "ErrorID" sowie die Elemente der Struktur "ErrorIdent" werden in diesem Fall nicht aktualisiert.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert sowie durch die in diesem Fall aktualisierten Ausgänge "ErrorID" und "ErrorIdent" spezifiziert.

IP

Der Eingang "IP" gibt die IP-Adresse des Kommunikationspartners als Zahl an. Eine Wandlung einer IP-Adresse in String-Form in das für den Eingang "IP" benötigte Format kann über die Funktion "SysSocketNetAddr" der Bibliothek SysSocket erfolgen.

JobTlg, SizeOfJobTlg

Am Eingang "SizeOfJobTlg" wird durch den Anwender die Anzahl der zu übertragenden Bytes im mittels des Eingangs "JobTlg" adressierten Datenbereich vorgegeben.

LinkSendRecvJob

Der Eingang "LinkSendRecvJob" spezifiziert den Link auf die zugehörige Instanz des Funktionsbausteins ML_SendRecvJobAss und ermöglicht u. a. die Auswahl des Kommunikationspfades und der Priorität ([ML_SendRecvJobAss, Seite 734](#)).

Fehlercodes

Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR, 16#0001	16#0000F770	16#00700001	Ungültige Telegrammadresse (Nullpointer)
INPUT_INVALID_ERROR, 16#0001	16#0000F770	16#00700002	Ungültige Telegrammgröße (0)
INPUT_INVALID_ERROR, 16#0001	16#0000F770	16#00700003	Ungültige Telegrammgröße (>255)
INPUT_INVALID_ERROR, 16#0001	16#0000F770	16#00700004	Ungültige IP Adresse

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR, 16#0001	16#0000F770	16#00700005	Ungültiger Link
DEVICE_ERROR, 16#0008	16#0000F770	16#00710001	Pufferüberlauf im Sendpuffer für Job-Telegramme

Tab. 12-9: Fehlercodes des Funktionsbausteins ML_SendJobTlg

12.3.4 ML_RecvJobTlg

Kurzbeschreibung Der Funktionsbaustein ML_RecvJobTlg dient der definierten Suche und dem Auslesen eines Jobs aus dem Eingangspuffer für Job-Telegramme. Der Funktionsbaustein ist für den Einsatz in einer zyklisch zeitgesteuerten SPS-Anwendertask mittlerer Priorität vorgesehen.

Zuordnung Zielsystem/ Bibliothek

Zielsystem	Bibliothek
MLC ab 10VRS	ML_JobComm
MLD/MPx18	MX_JobComm.lib
MLD/MPC07	MX_JobComm.lib

Tab. 12-10: Referenztablette Funktionsbaustein ML_RecvJobTlg

Schnittstellenbeschreibung

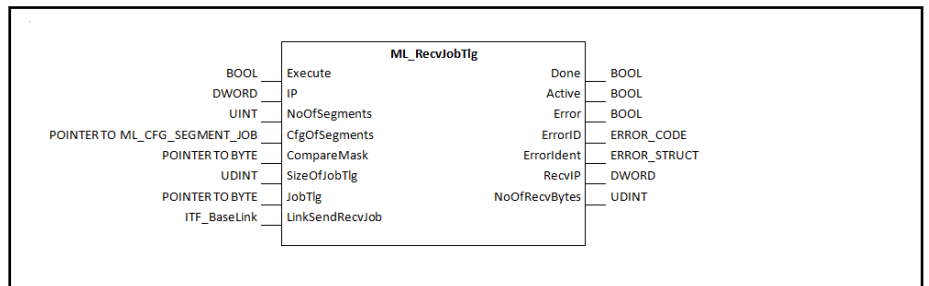


Abb. 12-4: Funktionsbaustein ML_RecvJobTlg

	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe
	IP	DWORD	IP-Adresse des Remote-Gerätes
	NoOfSegments	UINT	Anzahl der Segmente der Suchmaske (Anzahl der Arrayelemente)
	CfgOfSegments	POINTER TO ML_CFG_SEGMENT_JOB	Definition der Segmente der Suchmaske ML_CFG_SEGMENT_JOB, Seite 754
	CompareMask	POINTER TO BYTE	Suchmaske
	SizeOfJobTlg	UDINT	Anzahl der gültigen Bytes im Datenbereich "JobTlg", welcher den zu übertragenden Auftrag enthält
	JobTlg	POINTER TO BYTE	Zeiger auf den Datenbereich, welcher den zu übertragenden Auftrag enthält
	LinkSendRecvJob	ITF_BaseLink	Handle einer "ML_SendRecvJobAss"-Instanz ML_SendRecvJobAss, Seite 734
VAR_OUTPUT	Done	BOOL	Übertragung des Auftrags ohne Fehler abgeschlossen
	Active	BOOL	Übertragung des Auftrags aktiv
	Error	BOOL	Zeigt an, dass ein Fehler innerhalb des Funktionsbausteins aufgetreten ist.

	Name	Datentyp	Beschreibung
	ErrorID	ERROR_CODE	Fehlerkennung
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe Fehlercodes , Seite 742.
	RecvIP	DWORD	IP Adresse des Senders
	NoOfRecv-Bytes	UDINT	Anzahl der gültigen Bytes in dem mittels JobAdr adressierten Datenbereich

Tab. 12-11: Schnittstelle ML_RecvJobTlg

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
IP	DWORD	0	16#FFFFFFFF	0	steigende Flanke von "Execute"
NoOfSegments	UINT	0	20	0	steigende Flanke von "Execute"
CfgOfSegments	POINTER TO ML_CFG_SEGMENT_JOB			0	steigende Flanke von "Execute"
CompareMask	POINTER TO BYTE			0	steigende Flanke von "Execute"
SizeOfJobTlg	UDINT	1	255	0	steigende Flanke von "Execute"
JobTlg	POINTER TO BYTE			0	steigende Flanke von "Execute"
LinkSendRecvJob	ITF_BaseLink			0	steigende Flanke von "Execute"

Tab. 12-12: Eingangsverhalten ML_RecvJobTlg

Funktionsbeschreibung

Der Funktionsbaustein ML_RecvJobTlg dient der definierten Suche und dem Auslesen eines Jobs aus dem Eingangspuffer für Job-Telegramme RecvJob-Buff. Der Funktionsbaustein ist für den Einsatz in einer zyklisch zeitgesteuerten SPS Anwender-Task mittlerer Priorität vorgesehen.

Der Funktionsbaustein ML_RecvJobTlg sucht nach der Bearbeitungsfreigabe des Funktionsbausteins mittels "Execute" innerhalb eines vorgegebenen Bereiches ein definiertes Telegramm. Das gesuchte Telegramm wird über die Eingänge "NoOfSegments", "CfgOfSegments" und "CompareMask" spezifiziert. Am Eingang "NoOfBytes" wird durch den Anwender die Anzahl der zur Verfügung stehenden Bytes im mittels des Eingangs "JobTlg" adressierten Datenbereich vorgegeben. Eine aktivierte Übertragung kann nicht unterbrochen und durch eine weitere ersetzt werden (keine Retriggerunterstützung).

Das fehlerfreie Auffinden und Auslesen des Auftrags aus dem Eingangspuffer für Job-Telegramme wird durch den Ausgang "Done" signalisiert. Der Ausgang "ErrorID" sowie die Elemente der Struktur "ErrorIdent" werden in diesem Fall nicht aktualisiert. Der Ausgang "NoOfRecBytes" gibt Auskunft, wie viele Bytes gelesen wurden und im Falle der fehlerfreien Bearbeitung des Funktionsbausteins im mittels des Eingangs "JobTlg" adressierten Datenbereich gültig sind. Wurde kein Telegramm mit den entsprechenden Vorgaben gefunden, ist der Ausgang "NoOfRecBytes" 0.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert sowie durch die in diesem Fall aktualisierten Ausgänge "ErrorID" und "ErrorIdent" spezifiziert.

Die Lesezugriffe der Instanzen des Bausteins ML_RecvJobTlg auf den Eingangspuffer für Job-Telegramme sind geschützt, d. h. zu einem Zeitpunkt kann nur eine SPS-Task Job-Telegramme aus dem Lesebuffer auslesen.

IP

Der Eingang "IP" gibt die IP-Adresse des Kommunikationspartners als Zahl an. Eine Wandlung einer IP-Adresse in String-Form in das für den Eingang "IP" benötigte Format kann über die Funktion "SysSocketInetAddr" der Bibliothek SysSocket erfolgen.

NoOfSegments, CfgOfSegments, CompareMask

Die Suchmaske "CompareMask" für einen Auftrag wird über seine Segmentdefinition ("NoOfSegments", "CfgOfSegments") spezifiziert. Ein Segment definiert sich aus einem Startpunkt und der Länge des Segments in Bytes. Das erste Element der Suchmaske ist Byte 0.

Die Segmentdefinition definiert, welche Bytes eines Jobs geprüft werden. Jedes Segment ist über die Startposition und die Länge definiert ([ML_CFG_SEGMENT_JOB, Seite 754](#)).

Die Suchmaske "CompareMask" enthält das Suchmuster. Werden mehrere Segmente definiert, so werden die Suchmuster der einzelnen Segmente fortlaufen, d.h. ohne Trennung, in "CompareMask" abgelegt. Die Anzahl der Bytes in "CompareMask" muss die Summe aller in "CfgOfSegments" definierten Segmentgrößen sein.

Beispiel:

Es werden 2 Segmente definiert. Das erste Segment enthält 4 Bytes und startet bei Byte 0, das zweite Segment enthält 2 Bytes und startet bei Byte 12.

Daraus ergibt sich die folgende Definition:

```
NoOfSegment := 2
```

```
CfgOfSegments := [(0,4), (12,2)]
```

Die Suchmaske besteht demzufolge aus 6 Bytes und ist in diesem Beispiel wie folgt definiert:

```
CompareMask := [47, 12, 8, 15, 3, 98]
```

Ein empfangenes Job-Datenpaket könnte wie folgt aussehen:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
47	12	8	15	54	12	54	15	47	92	36	63	3	98	48	43	...
Segment 1												Segment 2				

Tab. 12-13: Job-Telegramm mit Vergleichsmaske

Das Telegramm würde also in diesem Beispiel auf die Suchmaske passen und somit vom Funktionsbaustein aus dem Empfangspuffer in den Datenbereich "JobTlg" übertragen.

JobTlg, SizeOfJobTlg

Am Eingang "SizeOfJobTlg" wird durch den Anwender die Anzahl der zu übertragenden Bytes im mittels des Eingangs "JobTlg" adressierten Datenbereich vorgegeben.

LinkSendRecvJob

Der Eingang "LinkSendRecvJob" spezifiziert den Link auf die zugehörige Instanz des Funktionsbausteins ML_SendRecvJobAss und ermöglicht u. a. die Auswahl des Kommunikationspfades und der Priorität.

Fehlercodes Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR, 16#0001	16#0000F780	16#00800001	Ungültige Telegrammadresse (Nullpointer)
INPUT_INVALID_ERROR, 16#0001	16#0000F780	16#00800002	Ungültige Telegrammgröße (0)
INPUT_INVALID_ERROR, 16#0001	16#0000F780	16#00800003	Ungültige IP Adresse
INPUT_INVALID_ERROR, 16#0001	16#0000F780	16#00800004	Ungültiger Link
INPUT_INVALID_ERROR, 16#0001	16#0000F780	16#00800005	Ungültige Suchmaskendefinition (Anzahl der Segmente gleich Null oder größer 20, Nullpointer der Maske und / oder der Konfiguration)
INPUT_INVALID_ERROR, 16#0001	16#0000F780	16#00800006	Ungültige Suchmaskendefinition (Längenangaben der Maske bzgl. Telegrammlänge)
DEVICE_ERROR, 16#0008	16#0000F780	16#00810001	Nicht genug Speicher zur Ablage des Telegramms vorhanden

Tab. 12-14: Fehlercodes des Funktionsbausteins ML_RecvJobTlg

12.3.5 ML_CtrlJobConnect

Kurzbeschreibung Der Funktionsbaustein ML_CtrlJobConnect dient der lokalen Verwaltung einer bidirektionalen Port-Verbindung. Der Baustein ermöglicht die Kommandierung des Verbindungsauf- und -abbaus sowie das Beobachten des Status einer Verbindung. Der Funktionsbaustein ist für den Einsatz in einer zyklisch zeitgesteuerten SPS-Anwendertask mittlerer Priorität vorgesehen.

Zuordnung Zielsystem/ Bibliothek

Zielsystem	Bibliothek
MLC ab 10VRS	ML_JobComm
MLD/MPx18	MX_JobComm.lib
MLD/MPC07	MX_JobComm.lib

Tab. 12-15: Referenztable Funktionbaustein ML_CtrlJobConnect

Schnittstellenbeschreibung

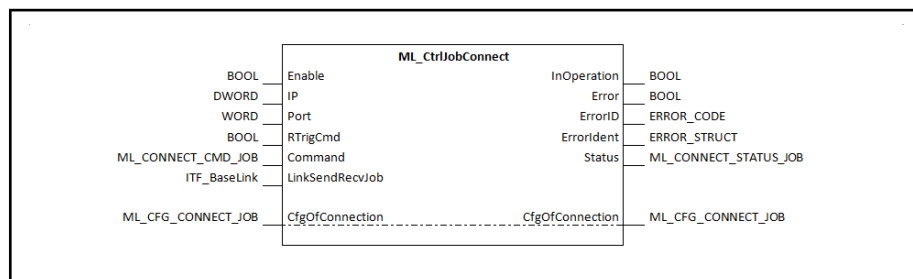


Abb. 12-5: Funktionsbaustein ML_CtrlJobConnect

	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe (zyklisch, zustandsgesteuert)
	IP	DWORD	IP-Adresse des Remote Geräts
	Port	WORD	UDP Port des Remote Geräts
	RTrigCmd	BOOL	Übernahme eines neuen Kommandos bei steigender Flanke
	Command	ML_CONNECT_CMD_JOB	Kommando ML_CONNECT_CMD_JOB , Seite 754.

	Name	Datentyp	Beschreibung
	LinkSend-RecvJob	ITF_BaseLink	Handle einer "ML_SendRecvJobAss"-Instanz ML_SendRecvJobAss, Seite 734
VAR_OUTPUT	InOperation	BOOL	Der Baustein arbeitet an seiner eigentlichen Aufgabe
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist
	ErrorID	ERROR_CODE	Fehlerkennung
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe Fehlercodes, Seite 745 .
	Status	ML_CONNECT_STATUS_JOB	Status der Verbindung ML_CONNECT_STATUS_JOB, Seite 754
VAR_IN_OUT	CfgOfConnection	ML_CFG_CONNECT_JOB	Konfiguration der Verbindung ML_CFG_CONNECT_JOB, Seite 753

Tab. 12-16: Schnittstelle ML_CtrlJobConnect

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
IP	DWORD	0	16#FFFFFFFF	0	steigende Flanke von "Enable"
Port	WORD	0	65535	0	steigende Flanke von "Enable"
RTrigCmd	BOOL			FALSE	kontinuierlich, wenn "Enable" TRUE
Command	ML_CONNECT_CMD_JOB			ML_SYNC_CMD_JOB	steigende Flanke von "Enable" oder "RTrigCmd"
LinkSendRecvJob	ITF_BaseLink			0	steigende Flanke von "Enable"

Tab. 12-17: Eingangsverhalten ML_CtrlJobConnect

Funktionsbeschreibung

Der Funktionsbaustein ML_CtrlJobConnect dient der lokalen Verwaltung einer bidirektionalen Port Verbindung. Der Baustein ermöglicht die Kommandierung des Verbindungsauf- und -abbaus sowie das Beobachten des Status einer Verbindung. Der Funktionsbaustein ist für den Einsatz in einer zyklisch zeitgesteuerten SPS Anwender-Task mittlerer Priorität vorgesehen.

Der Baustein kann zur Strukturierung des SPS Programms mehrfach instanziiert werden, eine Verbindung kann jedoch stets nur von einer Bausteininstanz gleichzeitig beobachtet werden. Der Baustein kann sich auf eine Verbindung erstmalig bzw. wieder aufsynchronisieren, sofern diese Verbindung bereits besteht und sich keine andere Instanz aufsynchronisiert hat.

Der Funktionsbaustein ML_CtrlJobConnect übermittelt nach der Bearbeitungsfreigabe des Funktionsbausteins mittels "Enable" das am Eingang "Command" vorgegebene Kommando ([ML_SendRecvJobAss, Seite 734](#)). Der Eingang "LinkSendRecvJob" spezifiziert den Link auf die zugehörige Instanz des Funktionsbausteins ML_SendRecvJobAss und ermöglicht u. a. die Auswahl des Kommunikationspfades und der Priorität.

Die aktive Kontrolle einer Verbindung durch den Funktionsbaustein wird durch "InOperation" signalisiert. Der Ausgang "ErrorID" sowie Elemente der Struktur "ErrorIdent" werden in diesem Fall nicht aktualisiert.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert sowie durch die in diesem Fall aktualisierten Ausgänge "ErrorID" und "ErrorIdent" spezifiziert.

IP

Der Eingang "IP" gibt die IP-Adresse des Kommunikationspartners als Zahl an. Eine Wandlung einer IP-Adresse in String-Form in das für den Eingang "IP" benötigte Format kann über die Funktion "SysSocketInetAddr" der Bibliothek SysSocket erfolgen.

RTrigCmd, Command

Der Eingang "Command" bestimmt die Auszuführende Aktion. Ein neues Kommando wird bei steigender Flanke an Enable oder bei steigende Flanke an RTrigCmd übernommen.

Die folgenden Kommandos stehen zur Verfügung:

- **ML_SYNC_CMD_JOB:** Ermöglicht das Aufsynchronisieren auf eine bereits bestehende Verbindung (default). Existiert keine Verbindung für die angegebene IP, wird ein Fehler zurückgegeben.
- **ML_CONNECT_CMD_JOB:** Ermöglicht das Aufsynchronisieren bzw. Öffnen einer Verbindung zu der angegebenen IP (3-Wege-Handshake). Existiert bereits eine Verbindung zur Zieladresse und ist keine andere Instanz bereits aufsynchronisiert, so erfolgt lediglich ein Aufsynchronisieren auf die bereits bestehende Verbindung. Wird die Verbindung zur Zieladresse bereits durch eine andere Instanz beobachtet, wird ein Fehler zurückgegeben.
- **ML_DISCONNECT_CMD_JOB:** Ermöglicht das Schließen und Freigeben einer Verbindung zu der angegebenen IP (3-Wege-Handshake). Ist die Verbindung nicht vorhanden oder eine andere Instanz aufsynchronisiert, wird ein Fehler zurückgegeben.
- **ML_FREE_CMD_JOB:** Ermöglicht das Freigeben der Verwaltungsrechte der eigenen oder einer durch eine andere Bausteininstanz verwaltete Verbindung. Werden die Verwaltungsrechte einer anderen Bausteininstanz freigegeben, so wird diese benachrichtigt und signalisiert dies ihrerseits durch das Setzen der Fehlerausgänge.

Status

Der Ausgang "Status" gibt den Status der über den Eingang "IP" spezifizierten Verbindung aus.

Die folgenden Ausgabewerte sind möglich:

- **ML_INVALID_STATUS_JOB:** Signalisiert einen ungültigen, nicht auswertbaren Zustand.
- **ML_DISCONNECTED_STATUS_JOB:** Signalisiert, dass zur angegebenen Zieladresse keine Verbindung besteht (default).
- **ML_CONNECTING_STATUS_JOB:** Signalisiert einen Verbindungsaufbau zur angegebenen Zieladresse.
- **ML_CONNECTED _STATUS_JOB:** Signalisiert eine etablierte Verbindung zur angegebenen Zieladresse.
- **ML_DISCONNECTING_STATUS_JOB:** Signalisiert einen Verbindungsabbau zur angegebenen Zieladresse.

CfgOfConnection

Am Eingang CfgOfConnection werden Werte für den Verbindungs-Timeout und das Timeout-Verhalten von Datenpaketen hinterlegt ([ML_CFG_CONNECT_JOB, Seite 753](#)).

Fehlercodes Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR, 16#0001	16#0000F750	16#00500001	Ungültiges Kommando
INPUT_INVALID_ERROR, 16#0001	16#0000F750	16#00500002	Ungültige Konfiguration
INPUT_INVALID_ERROR, 16#0001	16#0000F750	16#00500003	Ungültige IP Adresse
INPUT_INVALID_ERROR, 16#0001	16#0000F750	16#00500004	Ungültiger Link
INPUT_INVALID_ERROR, 16#0001	16#0000F750	16#00500005	Die Verbindung wird durch eine andere Bauelementinstanz verwaltet.

Tab. 12-18: Fehlercodes des Funktionsbausteins ML_CtrlJobConnect

12.3.6 ML_ShowJobConnect

Kurzbeschreibung Der Funktionsbaustein ML_ShowJobConnect dient dem einmaligen Lesen des Status aller lokalen bidirektionalen Port-Verbindungen. Der Funktionsbaustein ist für den Einsatz in einer zyklisch zeitgesteuerten SPS-Task mittlerer Priorität vorgesehen.

Zuordnung Zielsystem/ Bibliothek

Zielsystem	Bibliothek
MLC ab 10VRS	ML_JobComm
MLD/MPx18	MX_JobComm.lib
MLD/MPC07	MX_JobComm.lib

Tab. 12-19: Referenztablette Funktionsbaustein ML_ShowJobConnect

Schnittstellenbeschreibung

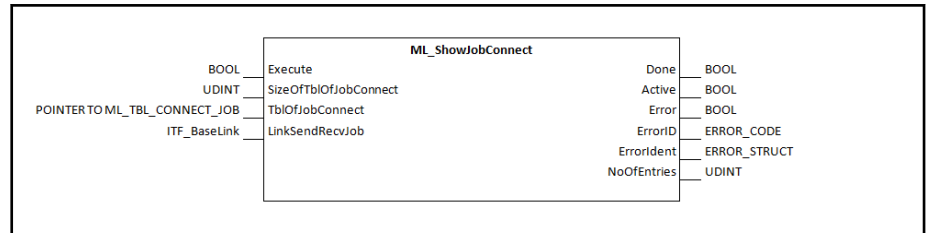


Abb. 12-6: Funktionsbaustein ML_ShowJobConnect

	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe
	SizeOfTblOfJobConnect	UDINT	Anzahl der zur Verfügung gestellten Bytes in dem mittels "TblOfJobConnect" adressierten Datenbereich
	TblOfJobConnect	POINTER TO ML_TBL_CONNECT_JOB	Zeiger auf den zur Verfügung gestellten Datenbereich für die zu lesenden Verbindungen ML_TBL_CONNECT_JOB, Seite 758.
	LinkSendRecvJob	ITF_BaseLink	Handle einer "ML_SendRecvJobAss"-Instanz ML_SendRecvJobAss, Seite 734
VAR_OUTPUT	Done	BOOL	Übertragung der Verbindungen ohne Fehler abgeschlossen
	Active	BOOL	Übertragung der Verbindungen aktiv
	Error	BOOL	Zeigt an, dass ein Fehler innerhalb des Funktionsbausteins aufgetreten ist.
	ErrorID	ERROR_CODE	Fehlerkennung

	Name	Datentyp	Beschreibung
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe Fehlercodes, Seite 747 .
	NoOfEntries	UDINT	Anzahl der gültigen Verbindungen

Tab. 12-20: Schnittstelle ML_ShowJobConnect

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
SizeOfTblOfJobConnect	UDINT	0	16#FFFFFFFF	0	steigende Flanke von "Execute"
TblOfJobConnect	POINTER TO ML_TBL_CONNECT_JOB			0	steigende Flanke von "Execute"
LinkSendRecvJob	ITF_BaseLink			0	steigende Flanke von "Execute"

Tab. 12-21: Eingangsverhalten ML_ShowJobConnect

Funktionsbeschreibung

Der Funktionsbaustein ML_ShowJobConnect dient dem einmaligen Lesen des Status aller lokalen bidirektionalen Port Verbindungen. Der Funktionsbaustein ist für den Einsatz in einer zyklisch zeitgesteuerten SPS-Task mittlerer Priorität vorgesehen.

Wurde das Lesen des aktuellen Status der Verbindungen mittels ML_ShowJobConnect angestoßen, wird mit jedem Aufruf der Instanz ML_SendRecvJobAss genau eine Verbindung auf den Zielbereich übertragen, d. h. die Einträge im Zielbereich sind nicht konsistent zueinander ([ML_SendRecvJobAss, Seite 734](#)).

Der Baustein kann mehrfach instanziiert und gleichzeitig aktiviert werden, da jedoch stets nur eine Verbindung auf einen Zielbereich übertragen werden kann, verlängert sich die Übertragungsdauer aller Instanzen dem entsprechend.

Der Funktionsbaustein ML_ShowJobConnect überträgt nach der Bearbeitungsfreigabe des Funktionsbausteins mittels "Execute" einen Auftrag zum Auslesen der Verbindungstabelle. Eine aktivierte Übertragung kann unterbrochen und durch eine neue ersetzt werden (Retriggerunterstützung vorhanden).

Am Eingang "NoOfBytes" wird durch den Anwender die Anzahl der zur Verfügung stehenden Bytes im mittels des Eingangs "TblOfJobConnect" adressierten Datenbereich vorgegeben. Der für den Auftrag bereitgestellte Datenbereich wird mittels des Eingangs "TblOfJobConnect" spezifiziert und muss auf ein Array vom Datentyp ML_TBL_CONNECT_JOB verweisen ([ML_TBL_CONNECT_JOB, Seite 758](#)). Der Eingang "LinkSendRecvJob" spezifiziert den Link auf die zugehörige Instanz des Funktionsbausteins ML_SendRecvJobAss und ermöglicht die Auswahl des Kommunikationspfades und der Priorität.

Die fehlerfreie Übertragung aller Verbindungen wird durch den Ausgang "Done" signalisiert. Der AusgangErrorID sowie Elemente der Struktur "ErrorIdent" werden in diesem Fall nicht aktualisiert. Der Ausgang "NoOfEntries" gibt Auskunft, wie viele Einträge gelesen wurden und im Falle der fehlerfreien Bearbeitung des Funktionsbausteins im mittels des Eingangs "TblOfJobConnect" adressierten Datenbereich gültig sind.

Der Ausgang "Active" kennzeichnet die Dauer der Übertragung der Daten, während dieses Zeitraumes sind die weiteren Ausgänge ungültig und können nicht ausgewertet werden.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert sowie durch den in diesem Fall aktualisierten Ausgang "ErrorID" sowie die Strukturelemente von "ErrorIdent" spezifiziert.

SizeOfTblOfJobConnect, TblOfJobConnect

Der über den Eingang "TblOfJobConnect" adressierte Speicher muss eine Mindestgröße von sizeof(ML_TBL_CONNECT_JOB) Byte aufweisen sowie ein ganzzahliges Vielfaches von sizeof (ML_TBL_CONNECT_JOB) betragen.

TblOfJobConnect ist ein Zeiger auf ein Array des Datentyps "ML_TBL_CONNECT_JOB" (ML_TBL_CONNECT_JOB, Seite 758).

Fehlercodes

Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR, 16#0001	16#0000F760	16#00600001	Ungültiger Datenbereich (Nullpointer)
INPUT_INVALID_ERROR, 16#0001	16#0000F760	16#00600002	Ungültige Größe des Datenbereichs (0)
INPUT_INVALID_ERROR, 16#0001	16#0000F760	16#00500003	Ungültiger Link

Tab. 12-22: Fehlercodes des Funktionsbausteins ML_ShowJobConnect

12.3.7 ML_DebugJobConnect

Kurzbeschreibung

Der Funktionsbaustein ML_DebugJobConnect dient der einmaligen Übertragung der Beobachtungsdaten aus internen Puffern.

Zuordnung Zielsystem/ Bibliothek

Zielsystem	Bibliothek
MLC ab 10VRS	ML_JobComm
MLD/MPx18	MX_JobComm.lib
MLD/MPC07	MX_JobComm.lib

Tab. 12-23: Referenztablette Funktionsbaustein ML_DebugJobConnect

Schnittstellenbeschreibung

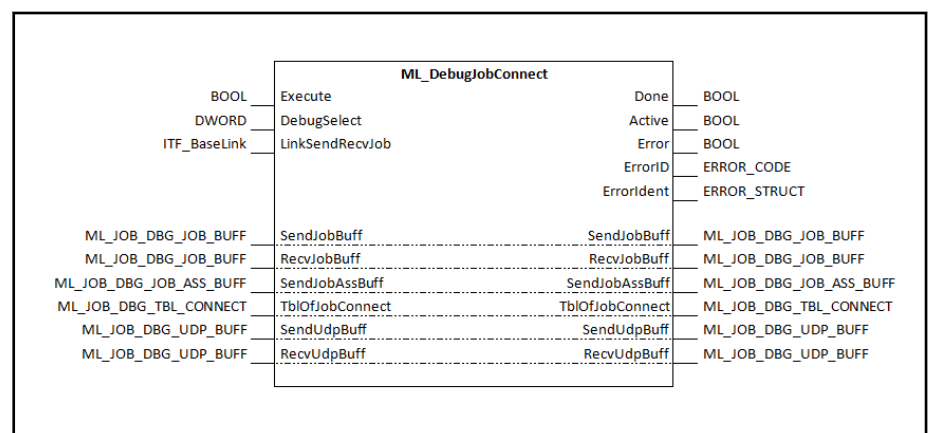


Abb. 12-7: Funktionsbaustein ML_DebugJobConnect

	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe
	DebugSelect	DWORD	Auswahl der Debug-Views
	LinkSend-RecvJob	ITF_BaseLink	Handle einer "ML_SendRecvJobAss"-Instanz ML_SendRecvJobAss, Seite 734
VAR_OUTPUT	Done	BOOL	Übertragung der Beobachtungsdaten ohne Fehler abgeschlossen
	Active	BOOL	Übertragung der Beobachtungsdaten aktiv
	Error	BOOL	Zeigt an, dass ein Fehler innerhalb des Funktionsbausteins aufgetreten ist.
	ErrorID	ERROR_CODE	Fehlerkennung
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe Fehlercodes, Seite 750 .
VAR_IN_OUT	SendJobBuff	ML_JOB_DBG_JOB_BUFF	Beobachtungsdaten des SendJobBuff ML_JOB_DBG_JOB_BUFF, Seite 757 .
	RecvJobBuff	ML_JOB_DBG_JOB_BUFF	Beobachtungsdaten des RecvJobBuff ML_JOB_DBG_JOB_BUFF, Seite 757 .
	SendJobAss-Buff	ML_JOB_DBG_JOB_ASS_BUFF	Beobachtungsdaten des SendJobAssBuff ML_JOB_DBG_JOB_ASS_BUFF, Seite 757 .
	TblOfJob-Connect	ML_JOB_DBG_TBL_CONNECT	Beobachtungsdaten des TblOfJobConnect ML_JOB_DBG_TBL_CONNECT, Seite 758 .
	SendUdpBuff	ML_JOB_DBG_UDP_BUFF	Beobachtungsdaten des SendUdpBuff ML_JOB_DBG_UDP_BUFF, Seite 758 .
	RecvUdpBuff	ML_JOB_DBG_UDP_BUFF	Beobachtungsdaten des RecvUdpBuff ML_JOB_DBG_UDP_BUFF, Seite 758 .

Tab. 12-24: Schnittstelle ML_DebugJobConnect

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	kontinuierlich
DebugSelect	DWORD	16#00	16#3F	0	steigende Flanke von "Execute"
LinkSendRecvJob	ML_LINK_SEND_RECV_JOB			0	steigende Flanke von "Execute"

Tab. 12-25: Eingangsverhalten ML_DebugJobConnect

Funktionsbeschreibung

Der Funktionsbaustein ML_DebugJobConnect überträgt nach der Bearbeitungsfreigabe des Funktionsbausteins mittels "Execute" einen Auftrag zur Übertragung der Beobachtungsdaten (View). Eine aktivierte Übertragung kann unterbrochen und durch eine neue ersetzt werden (Retriggerunterstützung vorhanden).

Die fehlerfreie Übertragung der selektierten Beobachtungsdaten (Views) wird durch den Ausgang "Done" signalisiert. Der Ausgang "ErrorID" sowie Elemente der Struktur "ErrorIdent" werden in diesem Fall nicht aktualisiert.

Der Ausgang "Active" kennzeichnet die Dauer der Übertragung der Daten. Während dieses Zeitraumes sind die weiteren Ausgänge ungültig und können nicht ausgewertet werden.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert sowie durch den in diesem Fall aktualisierten Ausgang "ErrorID" sowie die Strukturelemente von "ErrorIdent" spezifiziert.

DebugSelect

Am Eingang "DebugSelect" werden durch den Anwender die zu erzeugenden Views vorgegeben, wobei für die Auswahl globale Konstanten bereit stehen, welche mittels OR verknüpft werden können.

Die folgenden Konstanten werden unterstützt:

- K_ML_JOB_DBG_SEND_JOB_BUFF
- K_ML_JOB_DBG_RECV_JOB_BUFF
- K_ML_JOB_DBG_JOB_ASS_BUFF
- K_ML_JOB_DBG_TBL_CONNECT
- K_ML_JOB_DBG_SEND_UDP_BUFF
- K_ML_JOB_DBG_RECV_UDP_BUFF
- K_ML_JOB_DBG_NON
- K_ML_JOB_DBG_ALL

LinkSendRecvJob

Der Eingang "LinkSendRecvJob" spezifiziert den Link auf die zugehörige Instanz des Funktionsbausteins "ML_SendRecvJobAss" und ermöglicht die Auswahl des Kommunikationspfades und der Priorität ([ML_SendRecvJobAss](#), Seite 734).

SendJobBuff

Die Referenz "SendJobBuff" kann Informationen über den Job-Sendepuffer aufnehmen. Es ist somit möglich, den Iststand der Job-Abarbeitung zu betrachten. "SendJobBuff" wird durch den Datentyp "ML_JOB_DBG_JOB_BUFF" definiert ([ML_JOB_DBG_JOB_BUFF](#), Seite 757).

RecvJobBuff

Die Referenz "RecvJobBuff" kann Informationen über den Job-Empfangspuffer aufnehmen. Es ist somit möglich, den Iststand der Job-Abarbeitung zu betrachten. "RecvJobBuff" wird durch den Datentyp "ML_JOB_DBG_JOB_BUFF" definiert ([ML_JOB_DBG_JOB_BUFF](#), Seite 757).

SendJobAssBuff

Die Referenz "SendJobAssBuff" kann Informationen über den UDP-Sendepuffer aufnehmen. Es ist somit möglich, das aktuelle UDP-Paket zu betrachten. SendJobAssBuff wird durch den Datentyp "ML_JOB_DBG_JOB_ASS_BUFF" definiert ([ML_JOB_DBG_JOB_ASS_BUFF](#), Seite 757).

TblOfJobConnect

Die Referenz "TblOfJobConnect" kann Informationen über die Verbindungen zu anderen Kommunikationspartnern aufnehmen. Die Funktionalität ist vergleichbar mit dem Funktionsbaustein "ML_ShowJobConnect" ([ML_ShowJobConnect](#), Seite 745). TblOfJobConnect wird durch den Datentyp "ML_JOB_DBG_TBL_CONNECT" definiert ([ML_JOB_DBG_TBL_CONNECT](#), Seite 758).

SendUdpBuff

Die Referenz "SendUdpBuff" kann Informationen über den UDP-Sendepuffer aufnehmen. Es ist somit möglich, das aktuell gesendete UDP-Paket zu be-

trachten. SendUdpBuff wird durch den Datentyp "ML_JOB_DBG_UDP_BUFF" definiert ([ML_JOB_DBG_UDP_BUFF](#), Seite 758).

RecvUdpBuff

Die Referenz "RecvUdpBuff" kann Informationen über den UDP-Empfangspuffer aufnehmen. Es ist somit möglich, empfangene UDP-Pakete zu betrachten. RecvUdpBuff wird durch den Datentyp "ML_JOB_DBG_UDP_BUFF" definiert ([ML_JOB_DBG_UDP_BUFF](#), Seite 758).

Fehlercodes Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR, 16#0001	16#0000F7A0	16#00A00001	Ungültige Auswahl der Debug Views
INPUT_INVALID_ERROR, 16#0001	16#0000F7A0	16#00A00003	Ungültiger Link

Tab. 12-26: Fehlercodes des Funktionsbausteins ML_DebugJobConnect

12.3.8 ML_DebugJobConnect_Cyclic

Kurzbeschreibung

Der Funktionsbaustein ML_DebugJobConnect_Cyclic dient der zyklischen Übertragung der Beobachtungsdaten aus den internen Puffern.

Zuordnung Zielsystem/ Bibliothek

Zielsystem	Bibliothek
MLC ab 14VRS	ML_JobComm

Tab. 12-27: Referenztablette Funktionsbaustein ML_DebugJobConnect_Cyclic

Schnittstellenbeschreibung

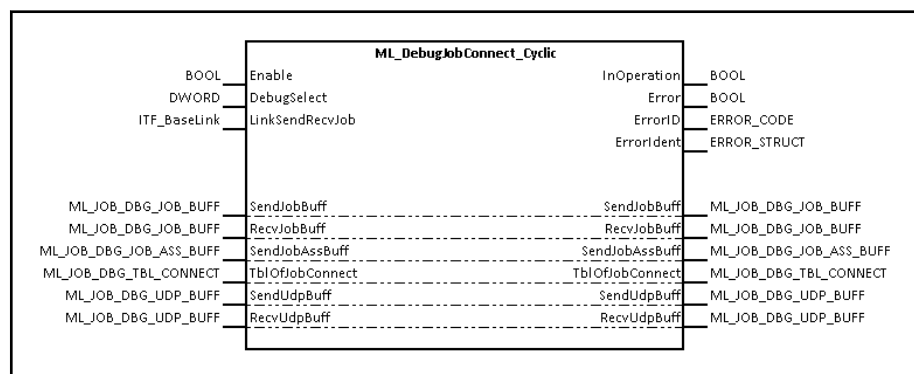


Abb. 12-8: Funktionsbaustein ML_DebugJobConnect_Cyclic

	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe
	DebugSelect	DWORD	Auswahl der Debug-Views
	LinkSend-RecvJob	ITF_BaseLink	Handle einer "ML_SendRecvJobAss"-Instanz
VAR_OUTPUT	InOperation	BOOL	Übertragung der Beobachtungsdaten ohne Fehler abgeschlossen
	Error	BOOL	Zeigt an, dass ein Fehler innerhalb des Funktionsbausteins aufgetreten ist.
	ErrorID	ERROR_CODE	Fehlererkennung
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung der Fehler, siehe Fehlercodes , Seite 752.

	Name	Datentyp	Beschreibung
VAR_IN_OUT	SendJobBuff	ML_JOB_DBG_JOB_BUFF	Beobachtungsdaten des SendJobBuff ML_JOB_DBG_JOB_BUFF, Seite 757.
	RecvJobBuff	ML_JOB_DBG_JOB_BUFF	Beobachtungsdaten des RecvJobBuff ML_JOB_DBG_JOB_BUFF, Seite 757.
	SendJobAss-Buff	ML_JOB_DBG_JOB_ASS_BUFF	Beobachtungsdaten des SendJobAssBuff ML_JOB_DBG_JOB_ASS_BUFF, Seite 757.
	TblOfJob-Connect	ML_JOB_DBG_TBL_CONNECT	Beobachtungsdaten des TblOfJobConnect ML_JOB_DBG_TBL_CONNECT, Seite 758.
	SendUdpBuff	ML_JOB_DBG_UDP_BUFF	Beobachtungsdaten des SendUdpBuff ML_JOB_DBG_UDP_BUFF, Seite 758.
	RecvUdpBuff	ML_JOB_DBG_UDP_BUFF	Beobachtungsdaten des RecvUdpBuff ML_JOB_DBG_UDP_BUFF, Seite 758.

Tab. 12-28: Schnittstelle ML_DebugJobConnect_Cyclic

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	kontinuierlich
DebugSelect	DWORD	16#00	16#3F	0	Kontinuierlich
LinkSendRecvJob	ML_LINK_SEND_RECV_JOB			0	steigende Flanke von "Enable"

Tab. 12-29: Eingangsverhalten ML_DebugJobConnect_Cyclic

Funktionsbeschreibung

Der Funktionsbaustein ML_DebugJobConnect_Cyclic überträgt nach der Bearbeitungsfreigabe des Funktionsbausteins mittels "Enable" die in Auftrag gegebenen Beobachtungsdaten (View). Die Beobachtungsdaten werden zyklisch ausgegeben.

Die fehlerfreie Übertragung der selektierten Beobachtungsdaten (Views) wird durch den Ausgang "Inoperation" signalisiert. Der Ausgang "ErrorID" sowie Elemente der Struktur "ErrorIdent" werden in diesem Fall nicht aktualisiert.

Tritt während der Bearbeitung des Funktionsbausteins ein Fehler auf, so wird dieser durch den Ausgang "Error" signalisiert sowie durch den in diesem Fall aktualisierten Ausgang "ErrorID" sowie die Strukturelemente von "ErrorIdent" spezifiziert.

DebugSelect

Am Eingang "DebugSelect" werden durch den Anwender die zu erzeugenden Views vorgegeben, wobei für die Auswahl globale Konstanten bereit stehen, welche mittels OR verknüpft werden können.

Die folgenden Konstanten werden unterstützt:

- K_ML_JOB_DBG_SEND_JOB_BUFF
- K_ML_JOB_DBG_RECV_JOB_BUFF
- K_ML_JOB_DBG_JOB_ASS_BUFF
- K_ML_JOB_DBG_TBL_CONNECT
- K_ML_JOB_DBG_SEND_UDP_BUFF
- K_ML_JOB_DBG_RECV_UDP_BUFF
- K_ML_JOB_DBG_NON
- K_ML_JOB_DBG_ALL

LinkSendRecvJob

Der Eingang "LinkSendRecvJob" spezifiziert den Link auf die zugehörige Instanz des Funktionsbausteins ML_SendRecvJobAss und ermöglicht die Auswahl des Kommunikationspfades und der Priorität ([ML_SendRecvJobAss, Seite 734](#))

SendJobBuff

Die Referenz "SendJobBuff" kann Informationen über den Job-Sendepuffer aufnehmen. Es ist somit möglich, den Iststand der Job-Abarbeitung zu betrachten. "SendJobBuff" wird durch den Datentyp "ML_JOB_DBG_JOB_BUFF" definiert ([ML_JOB_DBG_JOB_BUFF, Seite 757](#)).

RecvJobBuff

Die Referenz "RecvJobBuff" kann Informationen über den Job-Empfangspuffer aufnehmen. Es ist somit möglich, den Iststand der Job-Abarbeitung zu betrachten. "RecvJobBuff" wird durch den Datentyp "ML_JOB_DBG_JOB_BUFF" definiert ([ML_JOB_DBG_JOB_BUFF, Seite 757](#)).

SendJobAssBuff

Die Referenz "SendJobAssBuff" kann Informationen über den UDP-Sendepuffer aufnehmen. Es ist somit möglich, das aktuelle UDP-Paket zu betrachten. SendJobAssBuff wird durch den Datentyp "ML_JOB_DBG_JOB_ASS_BUFF" definiert ([ML_JOB_DBG_JOB_ASS_BUFF, Seite 757](#)).

TblOfJobConnect

Die Referenz "TblOfJobConnect" kann Informationen über die Verbindungen zu anderen Kommunikationspartnern aufnehmen. Die Funktionalität ist vergleichbar mit dem Funktionsbaustein "ML_ShowJobConnect" ([ML_ShowJobConnect, Seite 745](#)). TblOfJobConnect wird durch den Datentyp "ML_JOB_DBG_TBL_CONNECT" definiert ([ML_JOB_DBG_TBL_CONNECT, Seite 758](#)).

SendUdpBuff

Die Referenz "SendUdpBuff" kann Informationen über den UDP-Sendepuffer aufnehmen. Es ist somit möglich, das aktuell gesendete UDP-Paket zu betrachten. SendUdpBuff wird durch den Datentyp "ML_JOB_DBG_UDP_BUFF" definiert ([ML_JOB_DBG_UDP_BUFF, Seite 758](#)).

RecvUdpBuff

Die Referenz "RecvUdpBuff" kann Informationen über den UDP-Empfangspuffer aufnehmen. Es ist somit möglich, empfangene UDP-Pakete zu betrachten. RecvUdpBuff wird durch den Datentyp "ML_JOB_DBG_UDP_BUFF" definiert ([ML_JOB_DBG_UDP_BUFF, Seite 758](#)).

Fehlercodes

Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional1/Additional2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional1	Additional2	Beschreibung
INPUT_INVALID_ERROR, 16#0001	16#0000F7A0	16#00A00001	Ungültige Auswahl der Debug Views
INPUT_INVALID_ERROR, 16#0001	16#0000F7A0	16#00A00003	Ungültiger Link

Tab. 12-30: Fehlercodes des FB ML_DebugJobConnect_Cyclic

12.4 Datentypen

12.4.1 ML_CFG_CONNECT_JOB

Typdefinitionen Die Struktur "ML_CFG_CONNECT_JOB" wird innerhalb der Struktur "ML_CFG_SEND_RECV_JOB" für die Konfiguration der Funktionsbausteine "ML_SendRecvJobAss" und "ML_CtrlJobConnect" verwendet.

Die Struktur "ML_CFG_CONNECT_JOB" enthält die Konfiguration der Verbindungen.

Strukturelement	Typ	Beschreibung
TimeOutOfExchange	TIME	Wartezeit ohne Datentransfer bevor die Verbindung geschlossen wird
TimeOutUntilRepeat	TIME	Wartezeit ohne Bestätigung bevor das UDP-Paket erneut gesendet wird
MaxNofRepeat	UDINT	Maximale Wiederholungsanzahl für ein UDP-Paket

Tab. 12-31: Elemente der Struktur ML_CFG_CONNECT_JOB

Siehe auch

- [ML_CFG_SEND_RECV_JOB, Seite 754](#)
- [ML_SendRecvJobAss, Seite 734](#)
- [ML_CtrlJobConnect, Seite 742](#)

12.4.2 ML_CFG_JUNK_JOB

Typdefinitionen Die Struktur "ML_CFG_JUNK_JOB" wird innerhalb der Struktur ML_CFG_SEND_RECV_JOB für die Konfiguration des Funktionsbausteins "ML_SendRecvJobAss" verwendet.

Die Struktur "ML_CFG_JUNK_JOB" enthält die Konfiguration für die Junk-Puffer. Die Felder "SizeOfStorage", "AdrOfStorage" und "TimeOutOfJunk" müssen beschrieben werden, um den Junk-Puffer zu aktivieren.

Strukturelement	Typ	Standard	Beschreibung
SizeOfStorage	UDINT	0	Gesamtgröße des Junk Job Puffers. SizeOfStorage=Anzahl der Einträge*(4+SizeOfJunk-Job)
AdrOfStorage	POINTER TO BYTE	0	Zeiger auf den Junk-Job-Puffer
ActIndex	UDINT	0	aktiver Index des Junk-Job-Puffers
TimeOutOfJunk	TIME	t#0s	Maximale Zeit bevor ein Job aus dem Puffer entfernt wird
SizeOfJunkJob	UDINT	12	Anzahl der Bytes des Jobs, die in den Puffer kopiert werden sollen.

Tab. 12-32: Elemente der Struktur ML_CFG_JUNK_JOB

Siehe auch

- [ML_CFG_SEND_RECV_JOB](#), Seite 754
- [ML_SendRecvJobAss](#), Seite 734

12.4.3 ML_CFG_SEGMENT_JOB

Typdefinitionen Die Struktur "ML_CFG_SEGMENT_JOB" wird vom Funktionsbaustein ML_RecvJobTlg verwendet ([ML_RecvJobTlg](#), Seite 739).

Die Struktur "ML_CFG_SEGMENT_JOB" enthält die Konfiguration der Vergleichsmaske.

Strukturelement	Typ	Beschreibung
OffsetOfSegment	UDINT	Segment-Offset innerhalb des Maskenstroms
SizeOfSegment	UDINT	Größe des Segments

Tab. 12-33: Elemente der Struktur ML_CFG_SEGMENT_JOB

12.4.4 ML_CFG_SEND_RECV_JOB

Typdefinitionen Die Struktur "ML_CFG_SEND_RECV_JOB" wird für die Konfiguration des Funktionsbausteins "ML_SendRecvJobAss" verwendet ([ML_SendRecvJobAss](#), Seite 734).

Die Struktur "ML_CFG_SEND_RECV_JOB" enthält die Konfiguration für Verbindungen sowie die Junk-Puffer.

Strukturelement	Typ	Beschreibung
SendJunk	ML_CFG_JUNK_JOB	Konfiguration des Sende-Junk-Puffers ML_CFG_JUNK_JOB , Seite 753
RecvJunk	ML_CFG_JUNK_JOB	Konfiguration des Empfangs-Junk-Puffers ML_CFG_JUNK_JOB , Seite 753
CfgOfConnection	ML_CFG_CONNECT_JOB	Konfiguration der Verbindung ML_CFG_CONNECT_JOB , Seite 753

Tab. 12-34: Elemente der Struktur ML_CFG_SEND_RECV_JOB

12.4.5 ML_CONNECT_CMD_JOB

Der Aufzählungstyp "ML_CONNECT_CMD_JOB" wird im Funktionsbaustein "ML_CtrlJobConnect" verwendet ([ML_CtrlJobConnect](#), Seite 742).

Element	Wert	Beschreibung
ML_SYNC_CMD_JOB	0	Passiver Verbindungsaufbau
ML_CONNECT_CMD_JOB	1	Aktiver Verbindungsaufbau
ML_DISCONNECT_CMD_JOB	2	Verbindungsabbau
ML_FREE_CMD_JOB	3	Lösche Verbindung ohne Verbindungsabbau

Tab. 12-35: Definition des Aufzählungstyps ML_CONNECT_CMD_JOB

12.4.6 ML_CONNECT_STATUS_JOB

Der Aufzählungstyp "ML_CONNECT_STATUS_JOB" wird im Funktionsbaustein "ML_CtrlJobConnect" verwendet ([ML_CtrlJobConnect](#), Seite 742).

Element	Wert	Beschreibung
ML_ERROR_MSG_STATUS_JOB	-2	Es ist ein Fehler aufgetreten
ML_INVALID_STATUS_JOB	-1	Es ist kein gültiger Status verfügbar
ML_DISCONNECTED_STATUS_JOB	0	Keine Verbindung
ML_CONNECTING_STATUS_JOB	1	Verbindungsaufbau erfolgt gerade
ML_CONNECTED_STATUS_JOB	2	Verbindung ist aufgebaut
ML_DISCONNECTING_STATUS_JOB	3	Verbindungsabbau erfolgt gerade

Tab. 12-36: Definition des Aufzählungstyps ML_CONNECT_STATUS_JOB

12.4.7 ML_JOB_ASS_BUFF

Typdefinitionen Die Struktur "ML_JOB_ASS_BUFF" wird von der Datenstruktur "ML_JOB_DBG_JOB_ASS_BUFF" verwendet ([ML_JOB_DBG_JOB_ASS_BUFF](#), Seite 757).

Die Struktur "ML_JOB_ASS_BUFF" enthält Informationen über den Job-Assembly-Puffer.

Strukturelement	Typ	Beschreibung
DataValid	UDINT	Daten gültig
Length	UDINT	Länge des SCL-Telegramms
Destination	ML_JOB_ASS_BUFF_ADDR	Zieladresse ML_JOB_ASS_BUFF_ADDR , Seite 755
Scl	ML_JOB_ASS_BUFF_SCL	Kontrolldaten ML_JOB_ASS_BUFF_SCL , Seite 756
Data	ML_JOB_ASS_BUFF_DATA	Job-Daten ML_JOB_ASS_BUFF_DATA , Seite 755

Tab. 12-37: Elemente der Struktur ML_JOB_ASS_BUFF

12.4.8 ML_JOB_ASS_BUFF_ADDR

Typdefinitionen Die Struktur "ML_JOB_ASS_BUFF_ADDR" wird von der Datenstruktur "ML_JOB_ASS_BUFF" verwendet ([ML_JOB_ASS_BUFF](#), Seite 755).

Die Struktur "ML_JOB_ASS_BUFF_ADDR" enthält Informationen über den Job-Assembly Puffer.

Strukturelement	Typ	Beschreibung
Family	INT	Family
Port	UINT	Port
Addr	UDINT	IP-Adresse
Dummy	ARRAY[0..7] OF BYTE	Dummy

Tab. 12-38: Elemente der Struktur ML_JOB_ASS_BUFF_ADDR

12.4.9 ML_JOB_ASS_BUFF_DATA

Typdefinitionen Die Struktur "ML_JOB_ASS_BUFF_DATA" wird von der Datenstruktur "ML_JOB_ASS_BUFF" verwendet ([ML_JOB_ASS_BUFF](#), Seite 755).

Die Struktur "ML_JOB_ASS_BUFF_DATA" enthält Informationen die Daten des Job-Assembly Puffers.

Strukturelement	Typ	Beschreibung
Data	ARRAY [1..K_ML_DBG_JOB_SIZE_OF_UDP_DATAGRAM] OF BYTE	UDP-Paket

Tab. 12-39: Elemente der Struktur M_JOB_ASS_BUFF_DATA

12.4.10 ML_JOB_ASS_BUFF_SCL

Typdefinitionen Die Struktur "ML_JOB_ASS_BUFF_SCL" wird von der Datenstruktur "ML_JOB_ASS_BUFF" verwendet ([ML_JOB_ASS_BUFF](#), Seite 755).

Die Struktur "ML_JOB_ASS_BUFF_SCL" enthält Informationen über die Kontrollstruktur.

Strukturelement	Typ	Beschreibung
SeqNoLocal	UDINT	lokale Sequenznummer
SeqNoRemote	UDINT	Sequenznummer der Gegenstelle
Ctrl	WORD	Kontrollwort
ErrorCode	WORD	Fehlercode
Res01	WORD	reserviert
Res02	WORD	reserviert
TimeStampLow	DWORD	Zeitstempel (low 64 bits)
TimeStampHigh	DWORD	Zeitstempel (high 64 bits)
Res03	DWORD	reserviert
Version	UINT	Protokollversion
Res04	WORD	reserviert
NofDataSet	UINT	Anzahl der Job-Telegramme
Res05	WORD	reserviert

Tab. 12-40: Elemente der Struktur ML_JOB_ASS_BUFF_SCL

12.4.11 ML_JOB_BUFF

Typdefinitionen Die Struktur "ML_JOB_BUFF" wird von der Datenstruktur "ML_JOB_DBG_JOB_BUFF" verwendet ([ML_JOB_DBG_JOB_BUFF](#), Seite 757).

Die Struktur "ML_JOB_BUFF" enthält Informationen über den Job-Puffer.

Strukturelement	Typ	Beschreibung
Management	ML_JOB_BUFF_MANAGEMENT	Verwaltungsinformationen des Job-Telegramms ML_JOB_BUFF_MANAGEMENT , Seite 757
Data	ML_JOB_BUFF_DATA	Daten des Job-Telegramms ML_JOB_BUFF_DATA , Seite 756

Tab. 12-41: Elemente der Struktur ML_JOB_BUFF

12.4.12 ML_JOB_BUFF_DATA

Typdefinitionen Die Struktur "ML_JOB_BUFF_DATA" wird von den Datenstrukturen "ML_JOB_BUFF" und "ML_JOB_BUFF_MANAGEMENT" verwendet.

Die Struktur "ML_JOB_BUFF_DATA" enthält die Job-Daten.

Strukturelement	Typ	Beschreibung
Ip	DWORD	IP-Adresse des Ziels
Length	UDINT	Länge des Job-Telegramms
TimeStamp	UDINT	Zeitstempel des Job-Telegramms
Job	ARRAY[1..256] OF BYTE	Daten des Job-Telegramms

Tab. 12-42: Elemente der Struktur ML_JOB_BUFF_DATA

Siehe auch

- [ML_JOB_BUFF, Seite 756](#)
- [ML_JOB_BUFF_MANAGEMENT, Seite 757](#)

12.4.13 ML_JOB_BUFF_MANAGEMENT

Typdefinitionen Die Struktur "ML_JOB_BUFF_MANAGEMENT" wird von der Datenstruktur "ML_JOB_BUFF" verwendet ([ML_JOB_BUFF, Seite 756](#)).

Die Struktur "ML_JOB_BUFF_MANAGEMENT" enthält Informationen über den Job-Puffer.

Strukturelement	Typ	Beschreibung
DataValid	UDINT	1, Daten sind gültig 0, Daten sind nicht gültig
Data	POINTER TO ML_JOB_BUFF_DATA	Zeiger auf dem Datenbereich, wo die Daten des Job-Telegramms gespeichert. ML_JOB_BUFF_DATA, Seite 756

Tab. 12-43: Elemente der Struktur ML_JOB_BUFF_MANAGEMENT

12.4.14 ML_JOB_DBG_JOB_ASS_BUFF

Typdefinitionen Die Struktur "ML_JOB_DBG_JOB_ASS_BUFF" wird vom Funktionsbaustein "ML_DebugJobConnect" verwendet ([ML_DebugJobConnect, Seite 747](#)).

Die Struktur "ML_JOB_DBG_JOB_ASS_BUFF" enthält Informationen über den UDP-Sendepuffer.

Strukturelement	Typ	Beschreibung
Buffer	POINTER TO ML_JOB_ASS_BUFF	Zeiger auf den UDP-Puffer ML_JOB_ASS_BUFF, Seite 755
NofElements	UDINT	Anzahl der Jobs im UDP-Puffer

Tab. 12-44: Elemente der Struktur ML_JOB_DBG_JOB_ASS_BUFF

12.4.15 ML_JOB_DBG_JOB_BUFF

Typdefinitionen Die Struktur "ML_JOB_DBG_JOB_BUFF" wird vom Funktionsbaustein "ML_DebugJobConnect" verwendet ([ML_DebugJobConnect, Seite 747](#)).

Die Struktur "ML_JOB_DBG_JOB_BUFF" enthält Informationen über den Job-Puffer.

Strukturelement	Typ	Beschreibung
Buffer	POINTER TO ML_JOB_BUFF	Zeiger auf den Job-Puffer ML_JOB_BUFF, Seite 756
NofElements	UDINT	Anzahl der Zeilen im Job-Puffer

Strukturelement	Typ	Beschreibung
RdIdx	POINTER TO UDINT	Leseindex des Job-Puffers
WrIdx	POINTER TO UDINT	Schreibindex des Job-Puffers

Tab. 12-45: Elemente der Struktur ML_JOB_DBG_JOB_BUFF

12.4.16 ML_JOB_DBG_TBL_CONNECT

Typdefinitionen Die Struktur "ML_JOB_DBG_TBL_CONNECT" wird vom Funktionsbaustein "ML_DebugJobConnect" verwendet ([ML_DebugJobConnect, Seite 747](#)).

Die Struktur "ML_JOB_DBG_TBL_CONNECT" enthält Informationen über der Verbindungstabelle.

Strukturelement	Typ	Beschreibung
Buffer	POINTER TO ML_TBL_CONNECT_JOB	Zeiger auf die Verbindungstabelle ML_TBL_CONNECT_JOB, Seite 758
NofElements	UDINT	Anzahl der Verbindung

Tab. 12-46: Elemente der Struktur ML_JOB_DBG_TBL_CONNECT

12.4.17 ML_JOB_DBG_UDP_BUFF

Typdefinitionen Die Struktur "ML_JOB_DBG_UDP_BUFF" wird vom Funktionsbaustein "ML_DebugJobConnect" verwendet ([ML_DebugJobConnect, Seite 747](#)).

Die Struktur "ML_JOB_DBG_UDP_BUFF" enthält Informationen über den UDP Ein- bzw. Ausgangspuffer.

Strukturelement	Typ	Beschreibung
Buffer	POINTER TO ML_JOB_ASS_BUFF	Zeiger auf den UDP-Puffer ML_JOB_ASS_BUFF, Seite 755
NofBytes	UDINT	Größe des aktuellen UDP-Pakets in Byte

Tab. 12-47: Elemente der Struktur ML_JOB_DBG_UDP_BUFF

12.4.18 ML_LINK_SEND_RECV_JOB

Der Datentyp "ML_LINK_SEND_RECV_JOB" wird vom Funktionsbaustein "ML_SendRecvJobAss" am Ausgang "CfgSendRecvJob" ausgegeben und enthält einen Handle auf die Instanz des "ML_SendRecvJobAss" Funktionsbausteins ([ML_SendRecvJobAss, Seite 734](#)).

Andere Funktionsbausteine der ML_JobComm benötigen diesen Handle zur Steuerung des "ML_SendRecvJobAss" Funktionsbausteins.

Die interne Definition des Datentyps "ML_LINK_SEND_RECV_JOB" ist nicht anwendungsrelevant.

12.4.19 ML_TBL_CONNECT_JOB

Typdefinitionen Die Struktur "ML_TBL_CONNECT_JOB" wird vom Funktionsbaustein ML_ShowJobConnect verwendet ([ML_ShowJobConnect, Seite 745](#)).

Die Struktur "ML_TBL_CONNECT_JOB" enthält einen Verbindungseintrag.

Strukturelement	Typ	Beschreibung
IP	DWORD	IP-Adresse des Kommunikationspartners
Port	UDINT	Port auf dem Zielrechner

Strukturelement	Typ	Beschreibung
Status	ML_CONNECT_STATUS_JOB	Verbindungsstatus ML_CONNECT_STATUS_JOB, Seite 754
StatusOld	ML_CONNECT_STATUS_JOB	letzter Verbindungsstatus ML_CONNECT_STATUS_JOB, Seite 754
ActCntLostTlg	UDINT	Verlorene UDP-Pakete
SeqNoRemote	UDINT	Sequenznummer der Gegenstelle
SeqNoLocal	UDINT	lokale Sequenznummer
TimeOutOfExchange	TIME	Verbindungstimeout ohne Datenverkehr
TimeOfLastExchange	TIME	Letzter Datenverkehr
TimeOutUntilRepeat	TIME	Timeout bevor ein UDP-Paket erneut gesendet wird
TimeOfLastRepeat	TIME	Zeit der letzten Wiederholung
MaxNofRepeat	UDINT	Maximale Wiederholungen
ActCntRepeat	UDINT	Wiederholungszähler
CtrlJobConnect	ITF_CTRLJOBCONNECT	Handle auf die Kontrollierende Instanz eines ML_CtrlJobConnect Funktionsbausteins ML_SendRecvJobAss, Seite 734

Tab. 12-48: Elemente der Struktur ML_TBL_CONNECT_JOB

12.5 SCL-Protokoll

12.5.1 Einführung und Übersicht

Das SCL-Protokoll (Secure Communication Layer) bildet einen gesicherten Transportmechanismus für so genannte Job-Telegramme ab. Ein Job-Telegramm ist ein bis zu 255 Byte großes Datenpaket. Für den Datenaustausch wird UDP über Ethernet verwendet. Das SCL-Protokoll arbeitet verbindungsorientiert, d.h. vor Beginn der Datenübertragung muss ein Transportweg explizit geöffnet werden. Die Datenübertragung ist abgesichert, d.h. eventuell verloren gegangene Pakete werden wiederholt. Die Anzahl der Wiederholungen sowie die Zeit bis zur Wiederholung des Sendens ist einstellbar.

Vorgaben Das SCL-Protokoll wurde mit den folgenden Vorgaben entwickelt:

- Alle Jobs müssen den Kommunikationspartner erreichen (gesicherte Datenübertragung).
- Die Reihenfolge der Jobs muss eingehalten werden.
- Ein Übertragungsfehler muss vom Sender erkannt werden, darf jedoch nicht zum blockieren des Gesamtsystems führen.
- Störungen oder fehlende Verbindungen zu einem Kommunikationspartner darf die Kommunikation zu anderen Kommunikationspartnern nicht beeinflussen.
- Sicherung der übertragenen Daten durch eine Prüfsumme.

12.5.2 Ablauf der Kommunikation

Die folgende Beschreibung des Kommunikationsablaufs ist für die Verwendung des SCL-Protokolls nicht notwendig. Sie dient nur zu Verständniss des Kommunikationsablaufs wenn Fehler analysiert werden sollen.

Verbindungsaufbau	Vor der Datenübertragung muss ein Verbindungsaufbau erfolgen. Hierfür gibt es zwei Möglichkeiten: 1. aktiver Verbindungsaufbau: Es wird versucht eine Verbindung mit dem angegebenen Kommunikationspartner herzustellen. Um eine Verbindung zu erhalten, muss der Kommunikationspartner ebenfalls einen aktiven oder passiven Verbindungsaufbau durchführen. 2. passiver Verbindungsaufbau: Es wird auf einen aktiven Verbindungsaufbau des Kommunikationspartners gewartet. Um eine Verbindung zu erhalten muss der Kommunikationspartner einen aktiven Verbindungsaufbau durchführen. Der Verbindungsaufbau erfolgt durch einen 3-Wege-Handshake. Hierbei sendet der aktive Kommunikationspartner ein "SYN"-Paket mit einer frei gewählten Sequenznummer und der Bestätigungsnummer 0. Der angesprochene Kommunikationspartner antwortet mit "SYN" "ACK". Als Sequenznummer wird eine frei gewählte Zahl eingetragen. Die Bestätigungsnummer ist die Sequenznummer des Kommunikationspartners. Dieses Paket wird dann vom Initiator der Verbindung mit mit "ACK" bestätigt. Die Sequenznummer ist bei diesem Paket 0, und die Bestätigungsnummer ist die Sequenznummer des Kommunikationspartners. Danach ist die Verbindung aufgebaut und steht für die Datenübertragung zur Verfügung. Der verbindungsstatus wird auf verbunden gesetzt. Antwortet der Kommunikationspartner nicht, so wird nach Ablauf der Wartezeit erneut versucht eine Verbindung aufzubauen. Schlägt dieser nach der Anzahl der konfigurierten Verbindungsversuchen immer noch fehl, wird abgebrochen. Tritt ein Fehler auf, so wird "ERR" im Feld "Ctrl" gesetzt und der entsprechende Fehlercode im Feld "Fehlercode" eingetragen. Meldet der Kommunikationspartner eine bereits bestehende Verbindung (Fehlercode 12), so kann der Initiator des neuen Verbindungsaufbaus die alte Verbindung durch setzen des "RST"-Flags im Feld "Ctrl" ohne geregelten Verbindungsabbau zurücksetzen.
Verbindungsabbau	Der Verbindungsabbau erfolgt über ein 3-Wege-Handshake. Er wird durch das "FIN"-Flag im Feld "Ctrl" eingeleitet. Die Sequenznummer ist die um eins erhöhte Sequenznummer des letzten gesendeten Datenpakets. Die Bestätigungsnummer ist 0. Die Gegenstelle antwortet mit gestztem "FIN"- und "ACK"-Flag im "Ctrl"-Feld. Die Sequenznummer ist die um eins erhöhte Sequenznummer des letzten gesendeten Datenpakets. Die Bestätigungsnummer ist die Sequenznummer des Kommunikationspartners. Dieses Paket wird dann vom Initiator des Verbindungsabbau mit mit "ACK" bestätigt. Die Sequenznummer ist bei diesem Paket 0, und die Bestätigungsnummer ist die Sequenznummer des Kommunikationspartners. Danach ist die Verbindung abgebaut. Es können keine Datenpkater mehr zwischen diesen beiden Kommunikationspartnern ausgetauscht werden. Der Verbindungsstatus wird auf nicht verbunden gesetzt. Antwortet der Kommunikationspartner nicht, so wird nach Ablauf der Wartezeit erneut versucht die Verbindung abzubauen. Schlägt dieser nach der Anzahl der konfigurierten Verbindungsversuchen immer noch fehl, wird abgebrochen und die Verbindung ohne kontrollierten Verbindungsabbau auf nicht verbunden gesetzt. Tritt ein Fehler auf, so wird "ERR" im Feld "Ctrl" gesetzt und der entsprechende Fehlercode im Feld "Fehlercode" eingetragen.

Datenübertragung

Ein Datenpaket ist dadurch gekennzeichnet, dass das Feld "Ctrl" den Wert 0 enthält. Die Sequenznummer wird bei jedem neuen Datenpaket um eins erhöht. Die Bestätigungsnummer ist immer 0. Die Anzahl der zu gesendeten Jobs steht im Feld "Anzahl der Datensätze". Die Jobs werden durch die Offseteinträge der Offsettabelle referenziert. Ein Offset zählt dabei immer ab Start des SCL-Headers. Die Länge der Jobs wird durch die Differenz zweier aufeinanderfolgender Offsets bestimmt. Das letzte Offset zeigt auf das Ende des SCL-Pakets.

Ein korrekt empfangenes Paket wird vom Empfänger durch ein gesetztes "ACK"-Flag im Feld "Ctrl" bestätigt. Die Bestätigungsnummer ist die Sequenznummer des empfangenen Datenpakets. Die Sequenznummer des Bestätigungspakets ist immer 0.

Empfängt der Sender innerhalb der konfigurierten Zeitspanne keine Bestätigung, so wird das Paket noch einmal gesendet. Erfolgt auch nach der konfigurierten Anzahl an Wiederholungsversuchen keine Bestätigung, so wird die Verbindung als unterbrochen angenommen. Der Verbindungsstatus wird in diesem Fall ohne kontrollierten Verbindungsabbau auf nicht verbunden gesetzt.

Ist das empfangene Datenpaket fehlerhaft oder tritt ein anderer Fehler auf, so wird der Sender durch das Setzen des "ERR"-Flags im Feld "Ctrl" und eintragen des entsprechenden Fehlercodes im Feld "Fehlercode" benachrichtigt.

12.5.3 Aufbau des UDP-Datagramms

Die nachfolgende Tabelle stellt den Aufbau eines UDP Telegramms inkl. zusätzlicher Protokollebene ohne ausführliche Datasets dar. Die Größe des gesamten Rahmens (IP-Header: 20 Byte + UDP-Header: 8 Byte + SCL-Header: 36 Byte + Daten) darf 1500 Bytes nicht überschreiten.

Protokoll	Offset	WORD	WORD
IP	0		Length
	4	Identification	Flags
	8	TTL	Header Checksum
	12	Source Address	
	16	Destination Address	
UDP	0	Quell-Port	Ziel-Port
	4	Länge	Prüfsumme

Proto- koll	Offset	WORD	WORD
SCL	0	Sequenznummer	
	4	Bestätigungsnummer	
	8	Ctrl Bit 0: SYN Bit 1: ACK Bit 2: FIN Bit 3: ERR Bit 4: RST alle anderen Bits werden nicht verwendet und sind somit 0	Fehlercodes des SCL-Protokolls, Seite 762
	12	Reserve	Reserve
	16	Zeitstempel (low)	
	20	Zeitstempel (high)	
	24	Reserve	Reserve
	28	Versionsnummer	Reserve
	32	Anzahl der Datensätze	Reserve
	36	Offset Datensatz 1	Offset Datensatz 2
	40	Offset Datensatz 3	Offset Datensatz 4
	...	...	...
	...	Offset Datensatz n	Offset nach letztem Datensatz
	Offset Datensatz 1	Datensatz 1	
	Offset Datensatz 2	Datensatz 2	
	...	...	
	Offset Datensatz n	Datensatz n	

Tab. 12-49: Aufbau eines SCL-Datenpakets über Ethernet

12.5.4 Fehlercodes des SCL-Protokolls

Die nachfolgende Tabelle stellt eine Übersicht des SCL-Feldes "Fehlercode".

Code	Beschreibung
0	Prüfung des Telegrams wurde erfolgreich abgeschlossen
1	Das empfangene Telegramm hat eine ungültige Telegrammlänge
2	Das empfangene Telegramm enthält eine inkompatible Telegrammversion
3	Die Reservefelder sind nicht 0
4	Das SCL-Element "Ctrl" enthält ungültige Steuerbits
5	Das SCL-Element "Error Code" steht im Widerspruch zum Steuerbit "ERR" des SCL-Elements "Ctrl"

Code	Beschreibung
6	Das SCL-Element "Ctrl" steht im Widerspruch zum SCL-Element "Anzahl der Datensätze"
7	Das SCL-Element "Anzahl der Datensätze" mit dem Wert 0 steht im Widerspruch zur resultierenden minimalen Telegrammlänge
8	Das SCL-Element "Anzahl der Datensätze" steht im Widerspruch zur resultierenden minimalen Telegrammlänge
9	Das SCL-Element "Offset Dataset x" ist größer als das SCL Element "Offset Dataset x+1"
10	Das SCL-Element "Offset Dataset n+1" (letztes Element) stimmt nicht mit der Telegrammlänge überein
11	Das Element "Sequenznummer" stimmt nicht mit dem erwarteten Wert überein
12	Es besteht bereits eine Verbindung zum Gerät mit der der anfragenden IP-Adresse. Es kann keine weitere Verbindung zur selben IP-Adresse aufgebaut werden
13	Es kann keine weitere Verbindung aufgebaut werden, da keine weiteren Ressourcen zur Verfügung stehen
14	nicht verwendet
15	Das Element "Sequenznummer" und / oder "Bestätigungsnummer" stimmt nicht mit dem erwarteten Wert überein
16	nicht verwendet
17	nicht verwendet

Tab. 12-50: Fehlercodes des SCL-Protokolls

13 Service und Support

Für Ihre schnelle und optimale Unterstützung verfügen wir über ein dichtes weltweites Servicenetz. Unsere Experten stehen Ihnen mit Rat und Tat zur Seite. Sie erreichen uns täglich **rund um die Uhr – auch an Wochenenden und Feiertagen**.

Service Deutschland Unser technologieorientiertes Competence Center in Lohr deckt alle Belange rund um den Service für elektrische Antriebe und Steuerungen ab.

Sie erreichen unsere **Service-Hotline** und unseren **Service-Helpdesk** unter:

Telefon: **+49 9352 40 5060**
Fax: **+49 9352 18 4941**
E-Mail: service.svc@boschrexroth.de
Internet: <http://www.boschrexroth.com>

Auf unseren Internetseiten finden Sie ergänzende Hinweise zu Service, Reparatur (z. B. Anlieferadressen) und Training.

Service weltweit Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit Ihrem Ansprechpartner auf. Die Hotline-Rufnummern entnehmen Sie bitte den Vertriebsadressen im Internet.

Vorbereitung der Informationen Wir können Ihnen schnell und effizient helfen, wenn Sie folgende Informationen bereithalten:

- Eine detaillierte Beschreibung der Störung und der Umstände
- Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern
- Ihre Kontaktdaten (Telefon-, Faxnummer und E-Mail-Adresse)

Index

A

Abkürzungen.....	15
Anregungen.....	16
ANSI Z535.6-2006.....	7
Antriebsparameter	
Backup.....	35, 41
Restore.....	38, 44

B

Beschwerde.....	16
Besonderheiten von MLD.....	3
Betriebsarten, MagicBelt.....	213
Bezeichnungen und Abkürzungen.....	15
Bibliothek	
ML_JobComm.....	731
ML_TechBase.....	17
ML_TechCrank.....	395
ML_TechMotion.....	163
ML_TechMotionBase.....	33
ML_TechRegi.....	307
ML_TechTensionAdvanced.....	439
MX_TechBase.....	17
MX_TechMotion.....	163
MX_TechMotionBase.....	33
MX_TechRegi.lib.....	307
MX_TechWinder.lib.....	491
MX_TechWinder.library.....	491
RIL_ParameterChannel.....	709
RMB_TechCam.....	659
RMB_TechCrossCutCrossSeal.....	623
RMB_TechWinder.....	491

C

CamLock.....	164
Applikationsbeispiel.....	165
CamLock Funktionsbausteine	
MB_CamLock.....	170
MB_PrepareCams.....	168

D

Durchmesser extern messen.....	519
Durchmesserberechnung Wickler.....	509

E

Entkopplungsnetzwerk.....	479
Erfassung eines extern gemessenen Durchmessers.....	519

F

Feedback.....	16
Fehlercodes	
IL_ParameterChannel.....	715
IL_ParameterChannel, Parameterkanal.....	716
IL_RegiQualityType01.....	321

MB_AbortTrigger.....	147
MB_AllocCyclicParameter.....	150
MB_AxisDistanceAccumulator.....	52
MB_BellCrankData.....	402
MB_CalcInertiaLimitType02.....	579
MB_CamTableBellCrank.....	408
MB_CamTableBellCrankSuperimposed.....	413
MB_CamTableCrank.....	429
MB_CamTableCrankSuperimposed.....	435
MB_CoordinatePairCamType01.....	666
MB_CrossCutSealType01.....	638
MB_CrossCutterCalcType04.....	646
MB_CrossCutterCalcType05.....	652
MB_DancerCalcControlParamType01.....	611
MB_DancerControlType03.....	500
MB_DancerControlType04.....	506
MB_DiameterCalculatorType03.....	515
MB_DiameterMeasurementType01.....	522
MB_DigitalCamSwitchOvs.....	95
MB_DistanceAccumulator.....	51
MB_FeedMovementCalcType01.....	673
MB_FeedMovementCalcType02.....	679
MB_FreeCyclicParameter.....	152
MB_GetCyclicChannelConfig.....	153
MB_GetCyclicParameterHandle.....	155
MB_GetLastCyclicParameterError.....	159
MB_GetSystemInfo.....	30
MB_InitTouchProbe.....	134
MB_LoadCamData.....	102
MB_MagicBeltType01.....	199
MB_PhiToXvirt.....	436
MB_PhiToYvirt.....	415
MB_PilgrimStepCalcType01.....	272
MB_PositionFilterType01.....	106
MB_PositionMonitoring_GantryCant.....	108
MB_PositionMonitoring_PosCollision.....	111
MB_PrepareCams.....	170
MB_PrintLengthCorrCalcType01.....	277
MB_ReachVelocityType02.....	116
MB_RegiMeasuringType01.....	310
MB_RegiMeasuringType02.....	314
MB_RegiMeasuringType03.....	318
MB_RegiRegulateType01.....	326
MB_RegiRegulateType02.....	334
MB_RegiRegulateType03.....	341
MB_RegiRegulateType04.....	349
MB_RegiSetpointLockType01.....	352
MB_RegisterControllerSideType01.....	388
MB_RegisterControllerType02.....	361
MB_RegisterControllerType04.....	374
MB_SlotterCalcType01.....	697
MB_SlotterCalcType02.....	708
MB_SmartBeltType01.....	244
MB_TensionControlLoadCellMatPar-Type01.....	305

Index

MB_TensionControlLoadCellMatPar- Type02.....	478	IL_REGI_QUALITY.....	321
MB_TensionControlLoadCellType01.....	291	IL_RegiQualityType01.....	318
MB_TensionControlLoadCellType02.....	463	IMC-Interface.....	19
MB_TensionDecouplingNetworkType01.....	487	Inbetriebnahme	
MB_TensionIdentModelParamType01.....	447	Durchmesserrechner.....	518
MB_TouchProbe.....	138	Tänzerlageregler.....	508
MB_TouchProbeContinuous.....	142	Wickelrechner mit Kraftmessdose.....	556
MB_TraverseControlType01.....	607	Wickelrechner mit Tänzer.....	539
MB_UnwindMaterialType02.....	584	Wickelrechner mit Zugkraftsteuerung/- gelung.....	556
MB_WinderDancerCtrlType01.....	530	Wickelrechner sensorlos.....	556
MB_WinderDancerCtrlType02.....	537		
MB_WinderTensionCtrlSpeedType01.....	570	K	
MB_WinderTensionCtrlType02.....	553	Kniehebelkinematik.....	396
MB_WindSpeedControlAdaptionType01.....	591	Kritik.....	16
MB_WindTaperProfileType01.....	588	Kundenfeedback.....	16
MB_WriteExpectWindow.....	145	Kurbelkinematik.....	420
MB_XvirtToXmech.....	438	Einzelachsbetrieb.....	421
MB_YvirtToPhi.....	417	mechanische translatorische Position.....	424
MB_YvirtToPhiPoly5.....	419	Synchronbetrieb mit Kurvenscheibe.....	421
MB_YvirtToYmech.....	404	Variablen und Zählrichtungen.....	422
MC_DigitalCamSwitch.....	69	virtuelle translatorische Position.....	424
ML_AxisBackup.....	38	Kurvenscheibentabellen	
ML_AxisBackup_Type02.....	44	Parameter in MLC und MLD.....	653
ML_AxisRestore.....	41		
ML_AxisRestore_Type02.....	48	M	
ML_CtrlJobConnect.....	745	MagicBelt.....	191
ML_DebugJobConnect.....	750	Automatik-Mode.....	215
ML_DebugJobConnect_Cyclic.....	753	Betriebsarten.....	213
ML_FlyingShear.....	186	Jog-Mode.....	214
ML_RecvJobTlg.....	742	Tune-Mode.....	214
ML_SagControlC.....	233	MB_AbortTrigger.....	145
ML_SendJobTlg.....	739	MB_AllocCyclicParameter.....	148
ML_SendRecvJobAss.....	736	MB_AxisDistanceAccumulator.....	51
ML_ShowJobConnect.....	747	MB_BellCrankData.....	400
MX_FlyingShear.....	186	MB_CalcInertiaLimitType02.....	577
Filtern inkrementeller Positionen.....	103	Notwendige Parametrierung.....	579
Fliegende Säge.....	177	MB_CAM_TABLE_DATA.....	653
		MB_CamLock.....	170
G		MB_CamLock Parametrierung.....	177
Gefahrenhinweise.....	7	MB_CAMPROF.....	102
Getriebeeinstellung - Berechnung.....	644	MB_CAMPROF_SLOTTER.....	708
Gültigkeit der Dokumentation.....	1	MB_CAMSWITCH_REF_OVS.....	97
		MB_CAMTABLE.....	102
H		MB_CamTableBellCrank.....	404
Helpdesk.....	765	MB_CamTableBellCrankSuperimposed.....	408
Hotline.....	765	MB_CamTableCrank.....	425
		MB_CamTableCrankSuperimposed.....	430
I		MB_COORDINATE_PAIR_CAM_SELECT.....	666
IL_ParameterChannel.....	709	MB_COORDINATE_PAIR_TYPE01.....	666
Fehlertelegamme.....	721	MB_CoordinatePairCamType01.....	659
Inbetriebnahme des Bausteins.....	723	MB_CORR_PROFILE.....	655
Index und Subindex.....	718	MB_CORRECTION_MODE.....	349
Initialisierung der Kommunikation.....	720	MB_CRANK.....	429
Parameterkanal im zyklischen Datenka- nal.....	716	MB_CrossCutSealType01.....	623
Steuer-/Statuswort.....	716	Querschneider-Anwendungsfall.....	633
		Quersiegler-Anwendungsfall.....	631

MB_CrossCutterCalcType04.....	638	MB_MBELT_MOVE.....	210
Besonderheiten der MLC.....	646	MB_MBELT_OUTFEED.....	209
MB_CrossCutterCalcType05.....	647	MB_MBELT_STARTTYPE.....	204
Besonderheiten für MLC.....	652	MB_MBELT_STATUS.....	211
Besonderheiten für MLD.....	653	MB_MBELT_STOPTYPE.....	204
MB_CURVETYPE.....	679	MB_PhiToXvirt.....	435
MB_CYCLIC_CHANNEL_CONFIG.....	161	MB_PhiToYvirt.....	413
MB_CYCLIC_CHANNELS.....	160	MB_PilgrimStepCalcType01.....	262
MB_CYCLIC_PARAM_REF.....	159	MB_POS_MON_GANTRY_CANT.....	108
MB_CYCLIC_TYPES.....	159	MB_POS_MON_POS_COLLISION.....	111
MB_DANCER_CONTROL_MODE.....	620	MB_PositionFilterType01.....	103
MB_DancerCalcControlParamType01.....	608	MB_PositionMonitoring_GantryCant.....	106
MB_DancerControlType03.....	495	MB_PositionMonitoring_PosCollision.....	109
Besonderheiten für MLC.....	500	MB_PrepareCams.....	168
Besonderheiten für MLD.....	501	MB_PrintLengthCorrCalcType01.....	272
MB_DancerControlType04.....	501	MB_PROBE_EXECUTION_MODE.....	135
Besonderheiten für MLC.....	507	MB_PUSH_OUT_CONFIG.....	655
Besonderheiten für MLD.....	507	MB_RAMPING_MODE.....	350
MB_DiameterCalculatorType03.....	509	MB_ReachVelocityType02.....	112
Besonderheiten für MLC.....	516	MB_ReadCyclicParameter.....	155
Besonderheiten für MLD.....	517	MB_ReadCyclicRealParameter.....	156
MB_DiameterMeasurementType01.....	519	MB_RegiMeasuringType01.....	308
Besonderheiten für MLC.....	523	MB_RegiMeasuringType02.....	310
Besonderheiten für MLD.....	523	MB_RegiMeasuringType03.....	314
MB_DigitalCamSwitchOvs.....	71	MB_RegiRegulateType01.....	321
MB_DistanceAccumulator.....	50	MB_RegiRegulateType02.....	326
MB_FeedMovementCalcType01.....	667	MB_RegiRegulateType03.....	334
MB_FeedMovementCalcType02.....	673	MB_RegiRegulateType04.....	341
MB_FreeCyclicParameter.....	150	MB_RegiSetpointLockType01.....	350
MB_FS_RETAIN_DATA.....	186	MB_RegisterControllerSideType01.....	381
MB_GetCyclicChannelConfig.....	152	MB_RegisterControllerType02.....	352
MB_GetCyclicParameterHandle.....	154	MB_RegisterControllerType04.....	362
MB_GetLastCyclicParameterError.....	158	MB_RESOLUTION.....	657
MB_GetSystemInfo.....	28	MB_SlotterCalcType01.....	685
MB_IMC_ADMIN_STATUS.....	21	MB_SlotterCalcType02.....	697
MB_IMC_ADMINISTRATION.....	20	MB_SMARTBELT_STATISTICS.....	245
MB_IMC_CONTROL_MODE_STATUS.....	23	MB_SmartBeltType01.....	238
MB_IMC_CONTROL_TYPE01.....	20	MB_SYSTEM_TYPE.....	31
MB_IMC_DIAGNOSIS.....	23	MB_TENSION_CONTROL_ADJUST.....	478
MB_IMC_ERROR_SEVERITY.....	23	MB_TENSION_CONTROL_CONFIG.....	478
MB_IMC_MODE_CONTROL.....	21	MB_TENSION_CONTROL_MATERIAL.....	305
MB_IMC_MODE_STATUS.....	22	MB_TENSION_CONTROL_MATERIAL_TY- PE.....	305
MB_IMC_SERCOS_PHASE.....	22	MB_TENSION_CONTROL_MODE.....	306
MB_IMC_STATUS_TYPE01.....	21	MB_TENSION_DECOUPLING_AXIS_VA- LUE.....	488
MB_IMCInterface.....	19	MB_TENSION_DECOUPLING_CONTROL- LER_VALUE.....	487
MB_InitTouchProbe.....	127	MB_TENSION_IDENT_EXTENDED_SET- TINGS.....	447
MB_LoadCamData.....	99	MB_TENSION_PRE_CTRL.....	479
MB_MagicBeltType01.....	192	MB_TensionControlLoadCellMatParType01.....	292
Inbetriebnahmehinweise.....	229	MB_TensionControlLoadCellMatParType02.....	463
MB_MasterToPhi.....	436	MB_TensionControlLoadCellType01.....	278
MB_MBELT_FEEDTYPE.....	207	Besonderheiten für MLD.....	291
MB_MBELT_GENERAL.....	200	MB_TensionControlLoadCellType02.....	447
MB_MBELT_INFEED.....	204	MB_TensionDecouplingNetworkType01.....	479
MB_MBELT_INFEEDSTEPS.....	207	MB_TensionIdentModelParamType01.....	439
MB_MBELT_MODEAUTO.....	203		
MB_MBELT_MODEJOG.....	202		
MB_MBELT_MODES.....	200		
MB_MBELT_MODETUNE.....	202		

Index

MB_TouchProbe.....	135	ML_AxisBackup.....	35
MB_TouchProbeContinuous.....	138	ML_AxisBackup_Type02.....	41
MB_TRACK_REF_OVS.....	98	ML_AxisRestore.....	38
MB_TRAVERSE_ADAPTION.....	612	ML_AxisRestore_Type02.....	44
MB_TRAVERSE_DIRECTION.....	613	ML_BACKUP_ERRORS.....	49
MB_TRAVERSE_MODE_LAYER_COUNTER.....	613	ML_BACKUP_OPTIONS.....	49
MB_TRAVERSE_UNWIND.....	613	ML_BACKUP_PARAMS.....	49
MB_TRAVERSE_UNWIND_CONTROL.....	614	ML_BASIC_CAMSWITCH_REF.....	69
MB_TRAVERSE_UNWIND_MODE.....	615	ML_BASIC_CAMSWITCH_REF_OVS.....	98
MB_TraverseControlType01.....	592	ML_BASIC_TRACK_REF.....	70
Besonderheiten für MLC.....	608	ML_BASIC_TRACK_REF_OVS.....	98
MB_UnwindMaterialType02.....	579	ML_CFG_CONNECT_JOB.....	753
Notwendige Parametrierung.....	584	ML_CFG_JUNK_JOB.....	753
MB_WIND_CTRL_TYPE.....	615	ML_CFG_SEGMENT_JOB.....	754
MB_WIND_DANCER_LIMITS.....	615	ML_CFG_SEND_RECV_JOB.....	754
MB_WIND_DIAM_CALC_MODE.....	616	ML_CONNECT_CMD_JOB.....	754
MB_WIND_DIAMETER.....	616	ML_CONNECT_STATUS_JOB.....	754
MB_WIND_DIRECTION.....	618	ML_CtrlJobConnect.....	742
MB_WIND_FRICTION.....	618	ML_DebugJobConnect.....	747
MB_WIND_TAPER_PROFILE.....	618	ML_DebugJobConnect_Cyclic.....	750
MB_WIND_TENS_DIAG.....	618	ML_DegToInc.....	24
MB_WIND_TENSION.....	619	ML_DegToRad.....	25
MB_WIND_TENSION_MODE.....	619	ML_FlyingShear.....	178
MB_WIND_TORQUE_LIMITS.....	619	Besonderheiten der MLC.....	186
MB_WinderDancerCtrlType01.....	525	ML_IncToDeg.....	25
Besonderheiten für MLC.....	531	ML_IncToRad.....	26
Besonderheiten für MLD.....	531	ML_JOB_ASS_BUFF.....	755
MB_WinderDancerCtrlType02.....	532	ML_JOB_ASS_BUFF_ADDR.....	755
Besonderheiten für MLC.....	538	ML_JOB_ASS_BUFF_DATA.....	755
Besonderheiten für MLD.....	538	ML_JOB_ASS_BUFF_SCL.....	756
MB_WinderTensionCtrlSpeedType01.....	560	ML_JOB_BUFF.....	756
Besonderheiten für MLC.....	571	ML_JOB_BUFF_DATA.....	756
Besonderheiten für MLD.....	571	ML_JOB_BUFF_MANAGEMENT.....	757
MB_WinderTensionCtrlType02.....	542	ML_JOB_DBG_JOB_ASS_BUFF.....	757
Besonderheiten für MLC.....	554	ML_JOB_DBG_JOB_BUFF.....	757
Besonderheiten für MLD.....	555	ML_JOB_DBG_TBL_CONNECT.....	758
MB_WindSpeedControlAdaptionType01.....	588	ML_JOB_DBG_UDP_BUFF.....	758
Besonderheiten für MLC.....	592	ML_JobComm.....	
MB_WindTaperProfileType01.....	585	Fehlercodes des SCL-Protokolls.....	762
MB_WriteCyclicParameter.....	157	ML_JobComm.library.....	731
MB_WriteCyclicRealParameter.....	157	ML_LINK_SEND_RECV_JOB.....	758
MB_WriteExpectWindow.....	142	ML_RadToDeg.....	27
MB_XvirtToXmech.....	437	ML_RadToInc.....	27
MB_YvirtToPhi.....	415	ML_RecvJobTlg.....	739
MB_YvirtToPhiPoly5.....	417	ML_SagControlA.....	234
MB_YVirtToWorkRange.....	419	ML_SagControlC.....	231
MB_YvirtToYmech.....	402	ML_SendJobTlg.....	737
MC_CAMSWITCH_REF.....	69	ML_SendRecvJobAss.....	734
MC_DigitalCamSwitch.....	53	ML_ShowJobConnect.....	745
MC_OUTPUT_REF.....	70	ML_TBL_CONNECT_JOB.....	758
MC_TRACK_REF.....	69	ML_TechBase.library.....	17
Messfunktionsbausteine für Registerregler.....	308	ML_TechCrank.library.....	395
Messtaster.....		ML_TechMotion.library.....	163
Komponenten.....	118	ML_TechMotionBase.library.....	33
Parametrierung.....	118	ML_TechRegi.library.....	307
Überblick.....	116	ML_TechTensionAdvanced.library.....	439
Messwertgeber-Unterstützung.....	50	MX_FlyingShear.....	178
		Komponenten und Parametrierung.....	187

MX_TechCrossCutCrossSeal.....	623	Konfiguration.....	389
MX_TechMotion.....	163	Seitenregisterregler-FB.....	381
MX_TechRegi.lib.....	307	Service-Hotline.....	765
MX_TechWinder		Sicherheitshinweise.....	7
Datentypen.....	612	Sichern und Wiederherstellen von Antrieb-	
Durchmesserberechnung.....	509	sparametern.....	35
Ergänzende Funktionsbausteine.....	576	Signalgrafik.....	7
Parameterübersicht.....	620	Signalwörter.....	7
Tänzerlageregelung.....	493	Slotter.....	680
Wickelrechner mit Tänzer.....	524	SmartBelt	
Wickelrechner mit Zugkraftsteuerung/-re-		Abfolgegrad.....	247
gelung.....	541	Anforderungen an Band und Maschine.....	249
Wickler mit Zugkraftregelung und Dreh-		Antriebsanforderungen.....	250
zahlkorrektur.....	560	Antriebsparametereinstellungen.....	251
MX_TechWinder.lib.....	491	Berechnung des ProbeSetpoint.....	257
MX_TechWinder.library.....	491	Berechnung des SmartBeltPitch.....	255
		Beschränkungen und Verhalten.....	258
		Projektplanung.....	249
		Reibung.....	248
		SPS-Programmierung.....	253
		Stauförderer.....	246
		Tipps und Tricks.....	259
		Verbreitete Fehler und Missverständnis-	
		se.....	261
		Verwendung.....	245
		Stellfunktionsbausteine für Registerregler.....	321
		Support.....	765
		Symbole.....	9
		Systeminformationen.....	28
N		T	
Nockenschaltwerk.....	52	Taktsynchrones Ein- und Auskopplung.....	164
		Tänzerlageregler	
		Inbetriebnahme.....	508
		Tänzerregelung.....	493
		Trigonometrische Funktionen.....	24
P		U	
Positionsüberwachung.....	106	Übersicht Bibliotheken und Zielsysteme.....	2
Q		W	
Querschneider.....	633	Warnhinweise.....	7
Quersiegler.....	631	Wickelrechner mit Tänzer.....	524
		Inbetriebnahme.....	539
		Wicklerachse kommandieren.....	539
		Wickelrechner mit Zugkraftsteuerung/-rege-	
		lung.....	541
		Inbetriebnahme.....	556
		Wickler, Durchmesserberechnung	
		Inbetriebnahme.....	518
		Wicklerrechner mit Zugkraftsteuerung/-regelung	
		Wicklerachse kommandieren.....	573
R		Z	
Registerqualitäts-FB.....	318	Zugkraftregler.....	277
Registerregler		Zyklische Datenkanäle.....	147
Anwendungsbeispiel Insetter-Regelung.....	379		
Anwendungsbeispiel Schlauchbeutelma-			
schine.....	380		
Beispiel Stanzen beim Etikettendruck.....	378		
Registerregler-FB.....	352		
Registersollwertübernahme-FB.....	350		
RIL_ParameterChannel.library.....	709		
RMB_TechCam.library.....	659		
RMB_TechCrosscutCrossseal.library.....	623		
RMB_TechWinder			
Datentypen.....	612		
Durchmesserberechnung.....	509		
Ergänzende Funktionsbausteine.....	576		
Parameterübersicht.....	620		
Tänzerlageregelung.....	493		
Wickelrechner mit Tänzer.....	524		
Wickelrechner mit Zugkraftsteuerung/-re-			
gelung.....	541		
Wickler mit Zugkraftregelung und Dreh-			
zahlkorrektur.....	560		
RMB_TechWinder.library.....	491		
S			
S-Kurve, Programmierung von Bewegungs-			
abläufen.....	112		
Schlaufenregelung.....	230		
Seitenregister			

Notizen

Notizen

Bosch Rexroth AG

Electric Drives and Controls

Postfach 13 57

97803 Lohr, Deutschland

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr, Deutschland

Tel. +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911341510