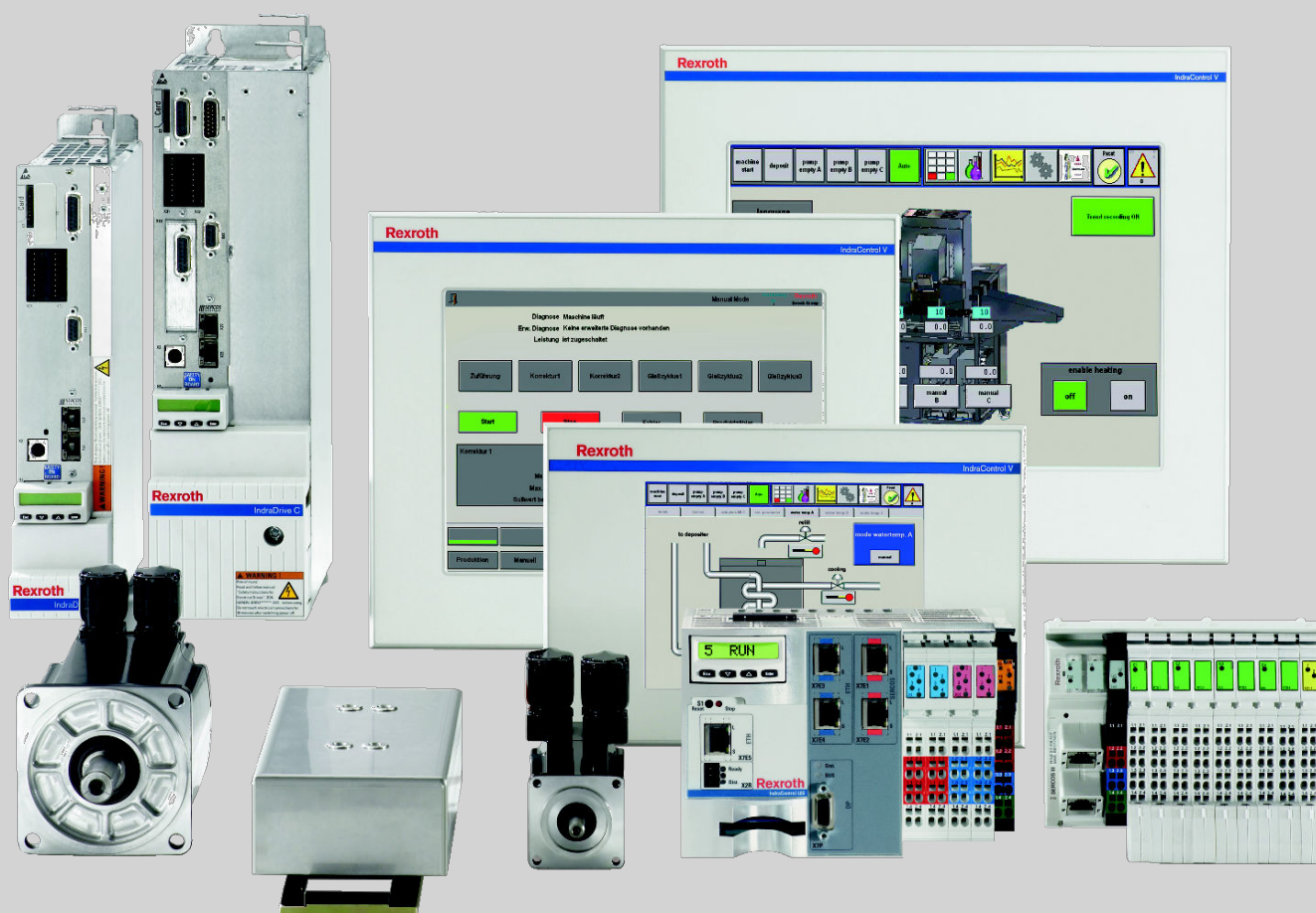


MLC 14VRS

Technology Libraries

Library
R911341511

Edition 08



Title	MLC 14VRS Technology Libraries
Type of Documentation	Library
Document Typecode	DOK-MLC***-TF*LIB**V14-LI08-EN-P
Internal File Reference	RS-d83f0c22df08c8b60a6846a5005708f8-8-en-US-3
Change Record	Edition 08, 2020-01 Refer to "Editions of this documentation" on page 1
Copyright	© Bosch Rexroth AG 2020 All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
Liability	The specified data is intended for product description purposes only and shall not be deemed to be a guaranteed characteristic unless expressly stipulated in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.
Editorial Department	System Development Automation Motion Logic Control, VK, (TaDo/MePe)

Table of Contents

	Page
1 About this documentation.....	1
1.1 Validity of the documentation.....	1
1.2 Libraries and target systems.....	2
1.3 Special features for the MLD.....	3
1.4 Using safety instructions.....	7
1.4.1 Structure of the safety instructions.....	7
1.4.2 Explaining signal words and safety alert symbol.....	7
1.4.3 Symbols used.....	9
1.4.4 Explaining the signal alert symbol on the device.....	9
1.5 Required and supplementing documentation XLC/MLC.....	9
1.6 Terms and abbreviations.....	15
1.7 Customer feedback.....	15
 2 ML_TechBase.library // MX_TechBase.library.....	 17
2.1 Introduction and overview.....	17
2.2 IMC interface.....	19
2.2.1 Introduction and Overview.....	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.....	22
2.2.10 MB_IMC_DIAGNOSIS.....	23
2.2.11 MB_IMC_ERROR_SEVERITY.....	23
2.3 Trigonometric functions.....	23
2.3.1 Introduction and overview.....	23
2.3.2 ML_DegToInc.....	24
2.3.3 ML_DegToRad.....	24
2.3.4 ML_IncToDeg.....	25
2.3.5 ML_IncToRad.....	25
2.3.6 ML_RadToDeg.....	26
2.3.7 ML_RadToInc.....	27
2.4 System information.....	27
2.4.1 MB_GetSystemInfo.....	27
2.4.2 MB_SYSTEM_TYPE.....	30
 3 ML_TechMotionBase.library // MX_TechMotionBase.library.....	 31
3.1 Introduction and overview.....	31
3.2 Backing up and restoring drive parameters.....	33
3.2.1 Introduction and overview.....	33

Table of Contents

	Page
3.2.2 ML_AxisBackup.....	33
3.2.3 ML_AxisRestore.....	36
3.2.4 ML_AxisBackup_Type02.....	39
3.2.5 ML_AxisRestore_Type02.....	42
3.2.6 ML_BACKUP_OPTIONS.....	45
3.2.7 ML_BACKUP_PARAMS.....	46
3.2.8 ML_BACKUP_ERRORS.....	46
3.3 Measurement encoder support.....	46
3.3.1 Introduction and overview.....	46
3.3.2 MB_DistanceAccumulator.....	47
3.3.3 MB_AxisDistanceAccumulator.....	48
3.4 Programmable limit switches, application.....	49
3.4.1 Introduction and overview.....	49
3.4.2 MC_DigitalCamSwitch.....	50
3.4.3 MC_CAMSWITCH_REF.....	65
3.4.4 ML_BASIC_CAMSWITCH_REF.....	65
3.4.5 MC_TRACK_REF.....	65
3.4.6 ML_BASIC_TRACK_REF.....	66
3.4.7 MC_OUTPUT_REF.....	66
3.4.8 MB_DigitalCamSwitchOvs.....	66
3.4.9 MB_CAMSWITCH_REF_OVS.....	92
3.4.10 ML_BASIC_CAMSWITCH_REF_OVS.....	93
3.4.11 MB_TRACK_REF_OVS.....	93
3.4.12 ML_BASIC_TRACK_REF_OVS.....	93
3.5 Cam download.....	94
3.5.1 Introduction and overview.....	94
3.5.2 MB_LoadCamData.....	94
3.5.3 MB_CAMPROF.....	97
3.5.4 MB_CAMTABLE.....	97
3.6 PositionFiltering (filtering incremental positions).....	97
3.6.1 Introduction and overview.....	97
3.6.2 MB_PositionFilterType01.....	98
3.7 Position monitoring.....	100
3.7.1 Introduction and overview.....	100
3.7.2 MB_PositionMonitoring_GantryCant.....	101
3.7.3 MB_POS_MON_GANTRY_CANT.....	102
3.7.4 MB_PositionMonitoring_PosCollision.....	103
3.7.5 MB_POS_MON_POS_COLLISION.....	105
3.8 S-curve, programming motion sequences.....	106
3.8.1 Introduction and overview.....	106
3.8.2 MB_ReachVelocityType02.....	106
3.9 Probe function blocks.....	110
3.9.1 Introduction and overview.....	110
3.9.2 Touch Probe Function Blocks - Components and Parameterization.....	111
Hardware.....	111
Firmware.....	112

	Page
Parameterizing the Touch Probe Function Blocks for MLC/XLC and MLD.....	112
3.9.3 MB_InitTouchProbe.....	120
3.9.4 MB_PROBE_EXECUTION_MODE.....	128
3.9.5 MB_TouchProbe.....	128
3.9.6 MB_TouchProbeContinuous.....	131
3.9.7 MB_WriteExpectWindow.....	134
3.9.8 MB_AbortTrigger.....	137
3.10 Cyclic data channels, administration	139
3.10.1 Introduction and overview.....	139
3.10.2 MB_AllocCyclicParameter.....	141
3.10.3 MB_FreeCyclicParameter.....	143
3.10.4 MB_GetCyclicChannelConfig.....	145
3.10.5 MB_GetCyclicParameterHandle.....	146
3.10.6 MB_ReadCyclicParameter.....	148
3.10.7 MB_ReadCyclicRealParameter.....	149
3.10.8 MB_WriteCyclicParameter.....	149
3.10.9 MB_WriteCyclicRealParameter.....	150
3.10.10 MB_GetLastCyclicParameterError.....	150
3.10.11 MB_CYCLIC_PARAM_REF.....	151
3.10.12 MB_CYCLIC_TYPES.....	152
3.10.13 MB_CYCLIC_CHANNELS.....	152
3.10.14 MB_CYCLIC_CHANNEL_CONFIG.....	153
4 ML_TechMotion.library // MX_TechMotion.....	155
4.1 Overview on Function Blocks.....	155
4.2 CamLock - Cycle-synchronous lock-on and lock-off	156
4.2.1 Introduction and Overview	156
4.2.2 CamLock - Application example.....	156
4.2.3 MB_PrepareCams.....	159
4.2.4 MB_CamLock.....	161
4.2.5 MB_CamLock Parameterization.....	168
4.3 FlyingShear.....	168
4.3.1 Introduction and Overview.....	168
4.3.2 ML_FlyingShear / MX_FlyingShear.....	169
4.3.3 MB_FS_RETAIN_DATA.....	176
4.3.4 ML_FlyingShear - Special Features of MLC.....	176
4.3.5 MX_FlyingShear - Special Features of MLD.....	177
4.4 MagicBelt.....	180
4.4.1 Introduction and overview.....	180
4.4.2 MB_MagicBeltType01.....	182
4.4.3 Data types, MagicBelt.....	189
MB_MBELT_MODES.....	189
MB_MBELT_GENERAL.....	189
MB_MBELT_MODEJOG.....	191
MB_MBELT_MODETUNE.....	192
MB_MBELT_MODEAUTO.....	193

Table of Contents

	Page
MB_MBELT_STARTTYPE.....	193
MB_MBELT_STOPTYPE.....	194
MB_MBELT_INFEED.....	194
MB_MBELT_FEEDTYPE.....	197
MB_MBELT_INFEEDSTEPS.....	197
MB_MBELT_OUTFEED.....	199
MB_MBELT_MOVE.....	200
MB_MBELT_STATUS.....	201
4.4.4 Operation modes, MagicBelt.....	203
Operation modes, MagicBelt, introduction and overview.....	203
Jog mode, MagicBelt.....	203
Tune mode, MagicBelt.....	204
Automatic mode, MagicBelt.....	204
Automatic mode, MagicBelt, general information.....	204
Start, automatic mode, MagicBelt.....	205
Loading, automatic mode, MagicBelt.....	206
Indexed loading, automatic mode, MagicBelt.....	206
Indexed loading with infeed in moving train, automatic mode, MagicBelt.....	209
Infeed with constant velocity, automatic mode, MagicBelt.....	210
Synchronous loading, automatic mode, MagicBelt.....	211
Motions between the loading and unloading station, automatic mode, MagicBelt.....	213
Unloading, automatic mode, MagicBelt.....	213
Stop, automatic mode, MagicBelt.....	215
Monitoring, automatic mode, MagicBelt.....	216
Restart, automatic mode, MagicBelt.....	218
4.4.5 Commissioning notes, MagicBelt.....	218
4.5 Sag Control.....	218
4.5.1 Introduction and Overview.....	218
4.5.2 ML_SagControlC.....	219
4.5.3 ML_SagControlA.....	223
4.6 SmartBelt.....	227
4.6.1 MB_SmartBeltType01.....	227
4.6.2 MB_SMARTBELT_STATISTICS.....	233
4.6.3 Introduction to the SmartBelt use.....	234
SmartBelt use.....	234
Accumulating conveyor.....	235
Degree of order.....	236
Friction.....	237
4.6.4 Project planning.....	238
General information.....	238
Mechanical requirements for belt and machine.....	238
Drive requirements.....	239
Drive parameter settings.....	240
PLC programming.....	242
Specification of the "SmartBeltPitch".....	243
Calculating the "ProbeSetpoint".....	245

	Page
4.6.5 SmartBelt limitations and behavior.....	246
Limitations.....	246
Two or more products on one belt.....	247
Multiple products per master cycle.....	247
4.6.6 Tips and tricks.....	247
Correction distance.....	247
Slippage.....	248
4.6.7 Common mistakes and misunderstandings.....	249
4.7 Cam tables for technology functions.....	249
4.7.1 Introduction and overview.....	249
4.7.2 MB_PilgrimStepCalcType01.....	250
4.7.3 MB_PrintLengthCorrCalcType01.....	260
4.8 Tension controller.....	265
4.8.1 Introduction and overview.....	265
4.8.2 MB_TensionControlLoadCellType01.....	266
4.8.3 MB_TensionControlLoadCellMatParType01.....	279
4.8.4 MB_TENSION_CONTROL_MATERIAL.....	292
4.8.5 MB_TENSION_CONTROL_MATERIAL_TYPE.....	292
4.8.6 MB_TENSION_CONTROL_MODE.....	293
5 ML_TechRegi.library // MX_TechRegi.lib.....	295
5.1 Introduction and overview.....	295
5.2 Measuring function blocks for register controllers.....	296
5.2.1 Introduction and overview.....	296
5.2.2 MB_RegiMeasuringType01.....	296
5.2.3 MB_RegiMeasuringType02.....	298
5.2.4 MB_RegiMeasuringType03.....	301
5.3 Register quality function block.....	305
5.3.1 Introduction and overview.....	305
5.3.2 IL_RegiQualityType01.....	305
5.3.3 IL_REGI_QUALITY.....	308
5.4 Regulating function blocks for register controllers.....	308
5.4.1 Introduction and overview.....	308
5.4.2 MB_RegiRegulateType01.....	308
5.4.3 MB_RegiRegulateType02.....	313
5.4.4 MB_RegiRegulateType03.....	321
5.4.5 MB_RegiRegulateType04.....	327
5.4.6 MB_CORRECTION_MODE.....	335
5.4.7 MB_RAMPING_MODE.....	335
5.5 Register setpoint lock function block.....	336
5.5.1 Introduction and overview.....	336
5.5.2 MB_SetpointLockType01.....	336
5.6 Register Controller Function Block.....	338
5.6.1 Introduction and overview.....	338
5.6.2 MB_RegisterControllerType02.....	338
5.6.3 MB_RegisterControllerType04.....	347

Table of Contents

	Page
5.6.4	Configuring the Register Controller..... 359
	Overview..... 359
	Using the Register Control..... 360
	Overview..... 360
	Configuring the Touch Probe Function Block..... 360
	Configuring the Touch Probe and RegisterControllerType04 Function Blocks..... 361
	Parameterizing the Register Controller..... 361
5.6.5	Register Controller - Use Cases..... 363
	Punching in Label Printing..... 363
	Closed-Loop Insetter Control..... 364
	Flow Wrapper..... 365
5.7	Side register controller function block..... 366
5.7.1	Introduction and overview..... 366
5.7.2	MB_RegisterControllerSideType01..... 366
5.7.3	Side Register Configuration..... 373
6	ML_TechCrank.library..... 381
6.1	Introduction and Overview..... 381
6.2	Bell crank kinematics..... 382
6.2.1	Introduction and overview..... 382
6.2.2	General Definitions..... 382
6.2.3	MB_BellCrankData..... 386
6.2.4	MB_YvirtToYmech..... 388
6.2.5	MB_CamTableBellCrank..... 390
6.2.6	MB_CamTableBellCrankSuperimposed..... 394
6.2.7	MB_PhiToYvirt..... 398
6.2.8	MB_YvirtToPhi..... 400
6.2.9	MB_YvirtToPhiPoly5..... 402
6.2.10	MB_YVirtToWorkRange..... 404
6.3	Crank kinematics..... 405
6.3.1	Introduction and Overview..... 405
6.3.2	General Definitions..... 407
	Definition of the Basic Variables in the Crank Kinematics..... 407
	Mechanical (Xmech) and Virtual (Xvirt) Translatory Position..... 409
6.3.3	MB_CamTableCrank..... 410
6.3.4	MB_CRANK..... 414
6.3.5	MB_CamTableCrankSuperimposed..... 415
6.3.6	MB_PhiToXvirt..... 420
6.3.7	MB_MasterToPhi..... 421
6.3.8	MB_XvirtToXmech..... 422
7	ML_TechTensionAdvanced.library..... 425
7.1	Introduction and overview..... 425
7.2	MB_TensionIdentModelParamType01..... 425
7.3	MB_TENSION_IDENT_EXTENDED_SETTINGS..... 432

	Page
7.4 MB_TensionControlLoadCellType02.....	433
7.5 MB_TensionControlLoadCellMatParType02.....	448
7.6 MB_TENSION_CONTROL_CONFIG.....	462
7.7 MB_TENSION_CONTROL_ADJUST.....	463
7.8 MB_TENSION_PRE_CTRL.....	463
7.9 Decoupling network.....	464
7.9.1 MB_TensionDecouplingNetworkType01.....	464
7.9.2 MB_TENSION_DECOUPLING_CONTROLLER_VALUE.....	471
7.9.3 MB_TENSION_DECOUPLING_AXIS_VALUE.....	472
8 RMB_TechWinder.library // MX_TechWinder.....	475
8.1 Introduction and overview.....	475
8.2 Closed-loop dancer position control.....	477
8.2.1 Introduction and overview.....	477
8.2.2 MB_DancerControlType03.....	479
8.2.3 MB_DancerControlType04.....	485
8.2.4 Commissioning of a closed-loop dancer position control.....	492
8.3 Diameter calculation with diameter calculator.....	494
8.3.1 Introduction and overview.....	494
8.3.2 MB_DiameterCalculatorType03	494
8.3.3 Commissioning the diameter calculator	502
8.4 Detecting an externally measured diameter.....	503
8.4.1 Introduction and overview.....	503
8.4.2 MB_DiameterMeasurementType01.....	503
8.5 Diameter calculator with dancer	508
8.5.1 Introduction and overview.....	508
8.5.2 MB_WinderDancerCtrlType01.....	509
8.5.3 MB_WinderDancerCtrlType02.....	516
8.5.4 Commissioning the diameter calculator with dancer	523
8.6 Diameter calculator with open-loop and closed-loop tension control	524
8.6.1 Introduction and overview.....	524
8.6.2 MB_WinderTensionCtrlType02.....	526
8.6.3 Commissioning the diameter calculator with open-loop and closed-loop tension control.....	540
8.7 Diameter calculator with closed-loop tension control and speed correction.....	544
8.7.1 Introduction and overview.....	544
8.7.2 MB_WinderTensionCtrlSpeedType01.....	544
8.7.3 Commissioning the winder with tension control and speed correction.....	555
8.8 Supplementary function blocks.....	559
8.8.1 Introduction and overview.....	559
8.8.2 MB_CalcInertiaLimitType02	559
8.8.3 MB_UnwindMaterialType02	562
8.8.4 MB_WindTaperProfileType01.....	567
8.8.5 MB_WindSpeedControlAdaptionType01.....	570
8.8.6 MB_TraverseControlType01.....	574
8.8.7 MB_DancerCalcControlParamType01.....	590
8.9 Data types.....	594

Table of Contents

	Page
8.9.1 Introduction and overview.....	594
8.9.2 MB_TRAVERSE_ADAPTION.....	594
8.9.3 MB_TRAVERSE_DIRECTION.....	595
8.9.4 MB_TRAVERSE_MODE_LAYER_COUNTER.....	595
8.9.5 MB_TRAVERSE_UNWIND.....	595
8.9.6 MB_TRAVERSE_UNWIND_CONTROL.....	596
8.9.7 MB_TRAVERSE_UNWIND_MODE.....	597
8.9.8 MB_WIND_CTRL_TYPE	597
8.9.9 MB_WIND_DANCER_LIMITS.....	597
8.9.10 MB_WIND_DIAM_CALC_MODE	598
8.9.11 MB_WIND_DIAMETER	598
8.9.12 MB_WIND_DIRECTION	600
8.9.13 MB_WIND_FRICTION	600
8.9.14 MB_WIND_TAPER_PROFILE.....	601
8.9.15 MB_WIND_TENS_DIAG	601
8.9.16 MB_WIND_TENSION	601
8.9.17 MB_WIND_TENSION_MODE	601
8.9.18 MB_WIND_TORQUE_LIMITS.....	602
8.9.19 MB_DANCER_CONTROL_MODE.....	602
8.10 Parameter overview.....	602
 9 RMB_TechCrossCutCrossSeal.library // MX_TechCrossCutCrossSeal.....	 605
9.1 Introduction and overview.....	605
9.2 MB_CrossCutSealType01.....	605
9.3 MB_CrossCutterCalcType04.....	619
9.4 MB_CrossCutterCalcType05.....	627
9.5 MB_CAM_TABLE_DATA.....	633
9.6 MB_CORR_PROFILE.....	635
9.7 MB_PUSH_OUT_CONFIG.....	636
9.8 MB_RESOLUTION.....	637
 10 RMB_TechCam.library.....	 639
10.1 Introduction and Overview.....	639
10.2 MB_CoordinatePairCamType01.....	639
10.3 MB_COORDINATE_PAIR_CAM_SELECT.....	646
10.4 MB_COORDINATE_PAIR_TYPE01.....	646
10.5 MB_FeedMovementCalcType01.....	647
10.6 MB_FeedMovementCalcType02.....	652
10.7 MB_CURVETYPE.....	658
10.8 Slotters, Introduction and Definitions.....	659
10.9 MB_SlotterCalcType01.....	664
10.10 MB_SlotterCalcType02.....	676
10.11 MB_CAMPROF_SLOTTER.....	686

	Page
11 RIL_ParameterChannel.library.....	687
11.1 Introduction and Overview.....	687
11.2 IL_ParameterChannel.....	687
11.3 IL_ParameterChannel - Protocol, Error Handling, and Commissioning.....	694
11.3.1 Description of the Protocol Used.....	694
Parameter Channel in the Cyclic Data Channel.....	694
Control Word/Status Word.....	694
Index and Subindex.....	696
Initializing the Communication.....	698
Error Telegrams.....	698
11.3.2 Commissioning of the Function Block.....	701
11.3.3 Differences to the Legacy Systems.....	705
General Information.....	705
On the Field Bus Slave.....	705
On the Field Bus Master.....	706
12 ML_JobComm.library.....	709
12.1 Introduction and overview.....	709
12.2 Architecture of the JobComm library.....	709
12.3 Function blocks.....	711
12.3.1 Introduction and overview.....	711
12.3.2 ML_SendRecvJobAss.....	711
12.3.3 ML_SendJobTlg.....	714
12.3.4 ML_RecvJobTlg.....	716
12.3.5 ML_CtrlJobConnect.....	719
12.3.6 ML_ShowJobConnect.....	722
12.3.7 ML_DebugJobConnect.....	724
12.3.8 ML_DebugJobConnect_Cyclic.....	727
12.4 Data types.....	730
12.4.1 ML_CFG_CONNECT_JOB.....	730
12.4.2 ML_CFG_JUNK_JOB.....	730
12.4.3 ML_CFG_SEGMENT_JOB.....	731
12.4.4 ML_CFG_SEND_RECV_JOB.....	731
12.4.5 ML_CONNECT_CMD_JOB.....	731
12.4.6 ML_CONNECT_STATUS_JOB.....	731
12.4.7 ML_JOB_ASS_BUFF.....	732
12.4.8 ML_JOB_ASS_BUFF_ADDR.....	732
12.4.9 ML_JOB_ASS_BUFF_DATA.....	732
12.4.10 ML_JOB_ASS_BUFF_SCL.....	733
12.4.11 ML_JOB_BUFF.....	733
12.4.12 ML_JOB_BUFF_DATA.....	733
12.4.13 ML_JOB_BUFF_MANAGEMENT.....	734
12.4.14 ML_JOB_DBG_JOB_ASS_BUFF.....	734
12.4.15 ML_JOB_DBG_JOB_BUFF.....	734
12.4.16 ML_JOB_DBG_TBL_CONNECT.....	734
12.4.17 ML_JOB_DBG_UDP_BUFF.....	735

Table of Contents

	Page
12.4.18 ML_LINK_SEND_RECV_JOB.....	735
12.4.19 ML_TBL_CONNECT_JOB.....	735
12.5 SCL Protocol.....	736
12.5.1 Introduction and overview.....	736
12.5.2 Sequence of communication.....	736
12.5.3 Structure of the UDP datagram.....	738
12.5.4 Error codes of the SCL protocols	739
 13 Service and support.....	 741
 Index.....	 743

1 About this documentation

Editions of this documentation

Edition	Release Date	Note
01	2014-05	First edition for MLC/XLC 14VRS
02	2014-08	MLD (IL 2G): MX_TechBase, MX_TechMotionBase MB_PilgrimStepCalcType01, MB_PositionFilterType01, MB_CrossCutterCalcType05
03	2015-07	Note: Using the point table with the FlexProfile Update: ML_TechMotion_MagicBelt, new: MB_DigitalCamSwitchOvs01, ML_JobComm.library
04	2015-09	MLD (IL 2G): RMB_TechWinder ML_TechMotionBase: ML_AxisBackup_Type02, ML_AxisRestore_Type02; MB_DigitalCamSwitch-Ovs updated
05	2016-05	ML_TechBase: ImcInterface ML_TechCrank: Description Bell-crank kinematics (svg) updated 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 for MLC and MLD
06	2017-08	Update: ML_TechMotionBase, RMB_TechCam, RMB_TechWinder, overview on "libraries and target systems"
07	2018-03	MB_TouchProbe_14, MB_TouchProbeContinuous_14, MB_RegiRegulateType02, MB_RegiMeasuringType03, MB_SetpointLockType01, MB_RegisterControllerType04, MB_DigitalCamSwitch-Ovs, MB_CAM_TABLE_DATA, MB_CORR_PROFILE, MB_PUSH_OUT_CONFIG supplemented or modified MB_DancerControlType03, MB_DancerControlType04, MB_DiameterCalculatorType03, MB_WinderDancerCtrlType01, MB_WinderDancerCtrlType02, MB_WinderTensionCtrlSpeedType01, MB_WinderTensionCtrlType02 supplement by side note "Commissioning" MB_WinderTensionCtrlSpeedType01 supplemented by "Affecting the engaging condition"
08	2020-01	New: chapter 8.8.7 "MB_DancerCalcControlParamType01" on page 590

Tab. 1-1: Change Record

1.1 Validity of the documentation

Target group This documentation describes how to use functions and function blocks from the technology libraries for your applications.

This documentation describes the libraries for the winder functionality, the register controller functionality and the CrossCutter functionality as well as their basics.

This documentation contains the function blocks, functions, interfaces and data types of the following libraries:

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

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

Overview on target groups and product phases

In the following illustration, the framed activities, product phases and target groups refer to the present documentation.

Example: In the product phase "Commissioning", the target group "Technologist" can execute the activities "Optimize" and "Test" using this documentation.

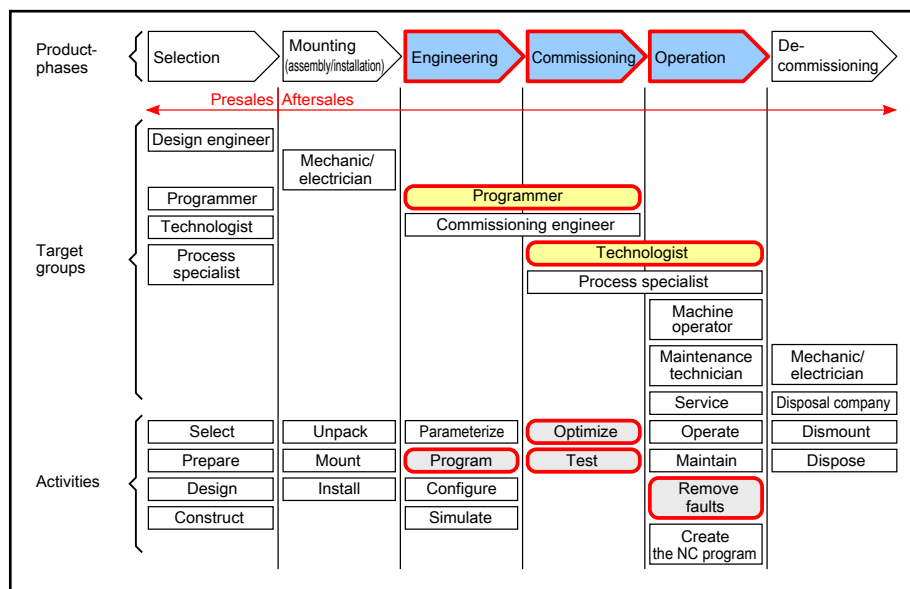


Fig. 1-1: Assigning the present documentation to the target groups, product phases and activities of the target group

1.2 Libraries and target systems

The following two tables list all supported target systems for the technology libraries as well as the required function packages.

Library	MLC with function package
ML_TechBase.library	PLC
ML_TechMotionBase.library	MOT
ML_TechMotion.library	MOT
ML_TechRegi.library	TEC
ML_TechCrank.library, page 381	TEC
ML_TechTensionAdvanced.library, page 425	License
ML_JobComm.library, page 709	PLC
RMB_TechWinder.library	TEC
RMB_TechCrosscutCrossSeal.library	TEC

Library	MLC with function package
RMB_TechCam.library, page 639	TEC
RIL_ParameterChannel.library, page 687	PLC

PLC Basic package
MOT Motion
ROCO Robotics
TEC Technology
HYD Hydraulics

Tab. 1-2: Overview on libraries and the target system MLC

Library	MLD with function package	Note
MX_TechBase.library, page 17	No function package required	
MX_TechMotionBase.library, page 31	No function package required	
MX_TechMotion, page 155	MA	For IndraLogic 1.x, from MPx20 with adapted functional scope
MX_TechRegi.library, page 295	MA	Only for IndraLogic 1.x and firm-ware < MPx18
MX_TechWinder.lib, page 475	MA	For IndraLogic 1.x, from MPx20 with adapted functional scope
MX_TechCrossCutCrossSeal, page 605	MA	For IndraLogic 1.x, from MPx20 with adapted functional scope

MA MLD Advanced

Tab. 1-3: Overview on libraries and the target system MLD



The technology function blocks (winder, register controller, tension controller...) are not supported for SercosDrive.

IndraDrive with interpolation in the control or drive is required.

1.3 Special features for the MLD

General information

MLD is an "onboard" control integrated into the hardware of the IndraDrive control section. The PLC firmware is thus part of the drive firmware. The PLC engineering environment is identical to the MLC engineering environment IndraLogic 2G for the drive firmware MPx18 and higher.

In contrast to the MLC, the PLC engineering environment IndraLogic 1.x. is used for a drive firmware < MPx18.



In MLC control systems up to firmware version MPx08, replace the I/O type "VAR_IN_OUT CONSTANT" by "VAR_IN_OUT".



Not all technology libraries are available with the same functional scope in the MLD. This can be checked using the target system tables for the technology libraries and the documentation of the respective function blocks.

Drive firmware, function package

The drive firmware is available in the following variants (with IndraLogic 1.x) from 07VRS (MPx 0x hardware):

- MPC firmware: Multi-axis MotionControl MLD-M for up to eight axes
- MPH firmware: Single-axis control MLD-S for one axis
- MPD firmware: No MLD functionality possible
- MPB firmware: Single-axis control MLD-S for one axis

The drive firmware is available in the following variants (with IndraLogic 2G) from 18VRS (MPx 1x hardware):

- MPC firmware: Multi-axis MotionControl MLD-M for up to eleven axes
- MPM firmware: No MLD functionality possible
- MPB firmware: Single-axis control MLD-S for one axis
- MPE firmware: No MLD functionality possible

Function packages

The drive firmware functionality is divided into several function packages. The function packages have to be ordered and licensed according to the application. The required function packages are then enabled.



The "MLD" function packages has to be enabled for the MLD.

The additional package "MLD Advanced" ("MA") has to be enabled for extended technology function blocks.



The additional package "MLD Advanced" is

- For the MPx 0x hardware: Only available for MPC firmware and MPH firmware (not for MPB firmware)
- available for the MPx 1x hardware: for MPC and MPB firmware

Technology libraries requiring the additional package "MLD Advanced" are for example:

- MX_TechWinder
- MX_TechRegi

Multi-axis MLD-M

The MPC drive firmware is required for the multiple axis control MLD-M. A special control section is required for this firmware (refer to the drive firmware description).

Additionally, set in the PLC configuration that the PLC has "permanent control" over the drive (P-0-1367 "PLC configuration" bit 4 set or dialog setting for the PLC configuration).



A special control section is required for the MPC firmware.

Software

IndraWorks 14VRS for MLD

Special parameterization for the MLD control

The "AxisData structure" of the drive has to be parameterized ("P-0-1367 PLC configuration", bit 6 is set or the dialog "PLC configuration")

For the multiple axis control MLD-M, set additionally in the PLC configuration that the PLC has "permanent control" over the drive ("P-0-1367 PLC configuration", bit 4 set or dialog for the PLC configuration).

For the MLD control, the virtual axis is realized using a master axis generator.

Configuring the virtual master axis generator:

The configuration of the virtual master axis generator comprises the following three parameterizations:

- Operating modes of the virtual master axis

Setting the positioning mode

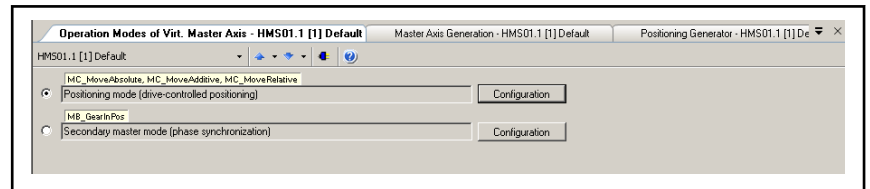


Fig. 1-2: Operation modes of the virtual master axis

- Master axis generation

Enter parameter "P-0-0758 Virtual master axis, actual position value"

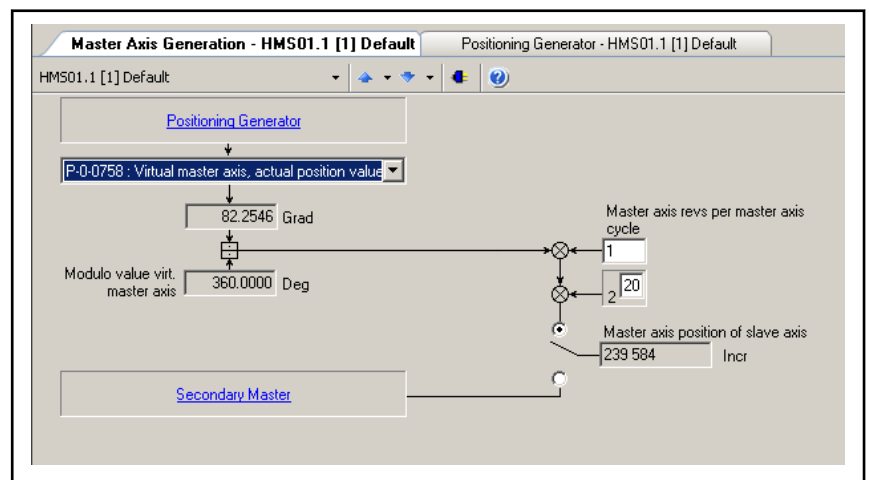


Fig. 1-3: Master axis generation

- Positioning generator

Enabling position command value

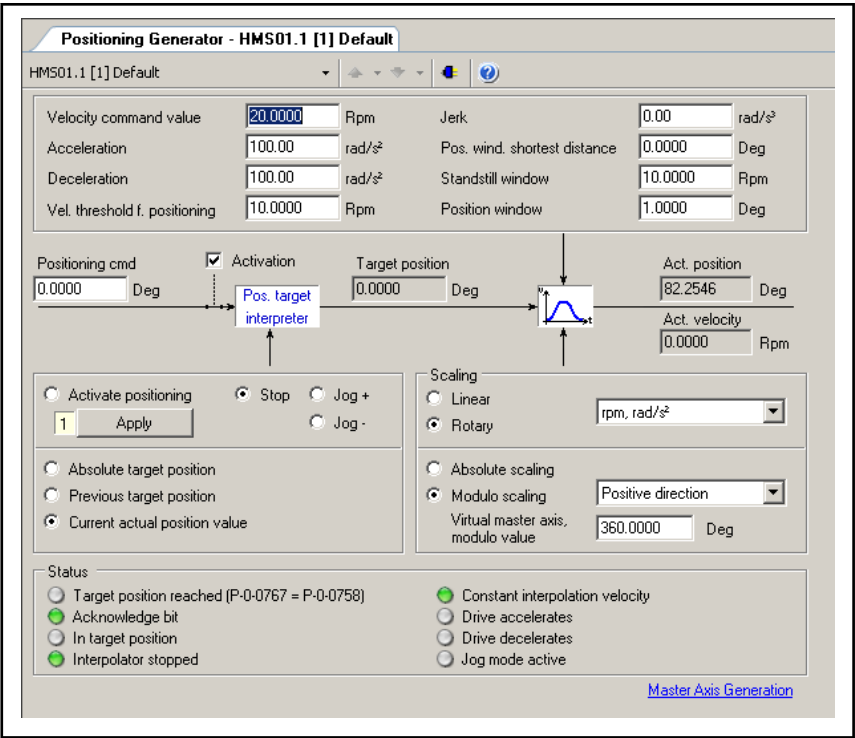


Fig. 1-4: Positioning generator

For a continuous velocity specification, set "Jog+" and "Jog-" and start the master axis generator using the "Apply" button.

Axis references for the MLD control:

Enter the axis references for the MLD control into the Global Variable List GVL_Base in the MX_Base library.

The following axis references apply from MPx18 (with IndraLogic 2G):

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

Fig. 1-5: Axis references for the MLD control (IL 2G)



The AxisData structure ("AxisData[AxisNumber]") is supported up to reference "VmAxis1" (number 20)!

For higher numbers or references (from "VmAxisInt", number 21), the AxisData structure cannot be used. In this case, the required variables have to be read in using individual parameters!

The AxisInterface cannot be used!



For technology function blocks, the following is recommended as axis reference for the master axis:

- Use axis reference "**VmAxis1**" for a virtual master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 20 (VMA_1).
- Use axis reference "**RmAxis1**" for a real master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 23 (RMA_1).
Additionally, the logic address of the measuring encoder has to be entered in parameter "P-0-1820.0.2 CCD: Source of measuring encoder position feedback value".

For detailed information, refer to Functional Description MLD.

1.4 Using safety instructions

1.4.1 Structure of the safety instructions

The safety instructions are structured as follows:

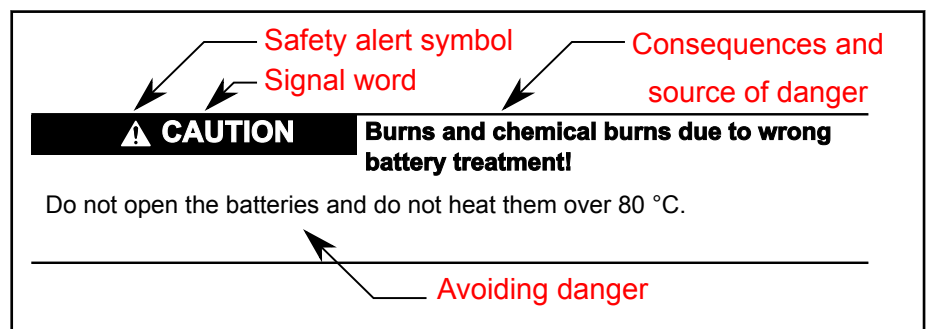


Fig. 1-6: Structure of the safety instructions

1.4.2 Explaining signal words and safety alert symbol

The safety instructions in this documentation contain specific signal words (danger, warning, caution, notice) and, if necessary, a safety alert symbol (according to ANSI Z535.6-2006).

The signal word draws attention to the safety instruction and indicates the risk potential.

The safety alert symbol (triangular safety reflector with exclamation marks), preceding the signal words Danger, Warning, Caution indicates hazards for persons.

DANGER

In case of non-compliance with this safety instruction, death or serious injury **will** occur.

WARNING

In case of non-compliance with this safety instruction, death or serious injury **can** occur.

⚠ CAUTION

In case of non-compliance with this safety instruction, minor or moderate injury can occur.

NOTICE

In case of non-compliance with this safety instruction, material damage can occur.

1.4.3 Symbols used

Pointers are displayed as follows:



This is a note.

Tips are displayed as follows:



This is a tip.

1.4.4 Explaining the signal alert symbol on the device



If this symbol is on your device, you have to observe the documentation on the device. The respective documentation informs on the type of hazard as well as the steps required to avoid this hazard.

1.5 Required and supplementing documentation XLC/MLC

Documentation titles with type codes and part numbers

System overview
IndraLogic XLC IndraMotion MLC 14VRS System Overview DOK-XLCMLC-SYSTEM**V14-PRRS-EN-P, R911341504 This documentation provides an overview on the possible hardware/software components of the automation systems IndraLogic XLC and IndraMotion MLC of the mentioned version. The documentation facilitates assembling a system.
First steps
IndraMotion MLC 14VRS First Steps DOK-MLC***-F*STEP**V14-CORS-EN-P, R911341517 This documentation describes the first steps of the IndraMotion MLC and the RobotControl. It includes the hardware and software prerequisites as well as the creation of a project.
IndraLogic XLC 14VRS First Steps DOK-XLC***-F*STEP**V14-CORS-EN-P, R911341489 This documentation describes the first steps of the IndraLogic XLC. It includes the hardware and software prerequisites as well as the creation of a project.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Project Conversion DOK-XLCMLC-PROCONV*V14-APRS-EN-P, R911341494 This documentation describes the project conversion of the IndraMotion MLC.

Tab. 1-4: XCL/MLC documentation overview - System overview

Tab. 1-5: XCL/MLC documentation overview - First steps

Engineering
IndraWorks 14VRS Software Installation DOK-IWORKS-SOFTINS*V14-CORS-EN-P, R911344286 This documentation describes the IndraWorks installation.
IndraWorks 14VRS Engineering DOK-IWORKS-ENGINEE*V14-APRS-EN-P, R911343566 This documentation describes the application of IndraWorks in which the Rexroth Engineering tools are integrated. It includes instructions on how to work with IndraWorks and how to operate the oscilloscope function.
IndraLogic XLC IndraMotion MLC 14VRS Function Description DOK-XLCMLC-FUNC****V14-APRS-EN-P, R911341700 This documentation describes wizards, context menus, dialogs, control commissioning, device configuration and functionalities of the IndraMotion MLC.

Tab. 1-6: XCL/MLC documentation overview - Engineering

Diagnostics and service
IndraLogic XLC IndraMotion MLC 14VRS Diagnostics DOK-XLCMLC-DIAG****V14-RERS-EN-P, R911341481 This documentation includes all control diagnostics implemented in the control systems IndraLogic XLC and IndraMotion MLC.
IndraLogic XLC IndraMotion MLC 14VRS Parameters DOK-XLCMLC-PARAM***V14-RERS-EN-P, R911341479 This documentation describes the parameters of the XLC/MLC systems as well as the interaction between parameterization and programming. It includes the axis parameters, control parameters, kinematic parameters, touch probe parameters and programmable limit switch parameters.
IndraLogic XLC IndraMotion MLC 14VRS Commissioning DOK-XLCMLC-STARTUP*V14-CORS-EN-P, R911341502 This documentation describes the steps to commission and service the IndraMotion MLC and IndraLogic XLC systems. It includes checklists for frequent tasks and a detailed description of the steps.
WebAssistant DOK-GENERL-WEB*ASSIST*-APRS-EN-P, R911381469 This documentation describes the WebAssistant. The WebAssistant is a webbased diagnostic tool used to access a control system via an Ethernet highspeed connection. The WebAssistant allows OEMs, end users and service engineers to access and to remotely diagnose a system. A PC with Internet Explorer version 11 or higher, Firefox version 45 or above or Google Chrome version 54 or above is required. The following control variants are supported: • IndraMotion MLC L45/L65/L75/XM21/XM22/XM42/VPx • IndraLogic XLC L45/L65/L75/VEP
IndraWorks IndraMotion Service Tool DOK-IWORKS-IMST*****-APRS-EN-P, R911341383 This documentation describes the IndraMotion Service Tool (IMST). IMST is a web-based diagnostic tool used to access a control system via an Ethernet high-speed connection. The IMST allows OEMs, end users and service engineers to access and remotely diagnose a system. The PC has to use at least Internet Explorer 8, Firefox 3.5 or a higher version. The following control variants are supported: • IndraMotion MLC L45/L65/L75/XM21/XM22/XM42/VPx • IndraLogic XLC L45/L65/L75

Tab. 1-7: XLC/MLC documentation overview - Diagnostics and service

Firmware
Rexroth IndraDrive MPx-20, Functions ¹⁾ DOK-INDRV*-MP*-20VRS**-AP01-EN-P, R911345608 This documentation describes all functional properties of the IndraDrive firmware in the variants MPB-20, MPM-20, MPC-20 and MPE-20.
Rexroth IndraDrive MPx-16 to MPx-20 und PSB, Parameters DOK-INDRV*-GEN1-PARA**-RERS-EN-P, R911328651 This documentation contains the descriptions of all parameters implemented in the firmware for drive controllers of the IndraDrive range. It supports the drive controller parameterization.
Rexroth IndraDrive MPx-16 to MPx-20 and PSB, Diagnostic Messages DOK-INDRV*-GEN1-DIAG**-RERS-EN-P, R911326738 This documentation contains the descriptions of all diagnostic messages implemented in the following firmware products: • Drive controller firmware variants MPx-16 to MPx-20 • PSB firmware variants for "HMu05" and "KMV03" type supply units • Firmware variants for "HMu01" type supply units It assists machine operators and installation programmers with troubleshooting.
Hardware
Rexroth IndraDrive Control Sections CSB02, CSE02, CSH02, CDB02 DOK-INDRV*-Cxx02*****-PRRS-EN-P, R911338962
Rexroth IndraDrive Cs Drive Systems with HCS01 DOK-INDRV*-HCS01*****-PRRS-EN-P, R911322210
Rexroth IndraDrive Mi Drive Systems with KCU02, KSM02, KMS02/03, KMV03 DOK-INDRV*-KCU02+KSM02-PRxx-EN-P, R911335703
Rexroth IndraDrive ML Drive Systems with HMu05 DOK-INDRV*-Hxx05*****-PRxx-EN-P, R911344279

1) In preparation
Tab. 1-8: Overview of documentations for drive controllers

PLC
IndraWorks 14VRS PLC Programming System IndraLogic 2G DOK-IWORKS-IL2GPRO*V14-APRS-EN-P, R911343571 This documentation describes the PLC programming tool IndraLogic 2G and its usage. It includes the basic use, first steps, visualization, menu items and editors.
IndraWorks 14VRS Basic Libraries IndraLogic 2G DOK-IL*2G*-BASLIB**V14-LIRS-EN-P, R911343920 This documentation describes the system-comprehensive PLC libraries.
IndraLogic XLC IndraMotion MLC 14VRS PLCopen Libraries DOK-XLCMLC-FUNLIB**V14-LIRS-EN-P, R911341491 This documentation describes the function blocks, functions and data types of the RIL_CommonTypes, ML_Base and ML_PLCOpen libraries for the IndraLogic XLC/IndraMotion MLC. It also includes the error reactions of function blocks.

Tab. 1-9: XCL/MLC documentation overview - PLC

Field buses
IndraWorks 14VRS Field Buses DOK-IWORKS-FB*****V14-APRS-EN-P, R911341485 This documentation describes the field bus and local periphery connections supported by the IndraLogic XLC, IndraMotion MLC and IndraMotion MTX systems. The focus of this documentation is on the configuration, parameterization, commissioning and the diagnostics of the different periphery connections. This documentation is the basis for the online help.
IndraWorks 14VRS Field Buses Libraries DOK-IWORKS-FB*LIB**V14-LIRS-EN-P, R911343575 This documentation describes the field bus libraries: RIL_ProfibusDP_02, RIL_ProfibusDP Slave, RIL_ProfinetIO, RIL_ProfinetIODevice, RIL_EtherNet/IPAdapter, RIL_MappingList, RIL_SERCOSIII, RIL_Inline including their diagnostics and error reactions of the function blocks.
Rexroth IndraWorks 12VRS FDT Container DOK-IWORKS-FDT*CON*V12-APRS-EN-P, R911334398 This documentation describes the IndraWorks FDT Container functionality. It includes the activation of the functionality in the project and working with DTMs.
Sercos System Manual for I/O Devices DOK-CTRL-ILS3*****-APxx-EN-P, R911333512 This documentation describes the configuration, parameterization, commissioning and diagnostics of I/O devices with a Sercos interface.

xx Corresponding edition
Tab. 1-10: XCL/MLC documentation overview - Field buses

HMI
IndraWorks 14VRS HMI DOK-IWORKS-HMI*****V14-APRS-EN-P, R911343569 This documentation describes the functions, configuration and operation of the user interfaces IndraWorks HMI Engineering and IndraWorks HMI Operation.
IndraWorks 14VRS WinStudio 7.4 DOK-IWORKS-WINSTUD*V14-APRS-EN-P, R911341585 This "User Manual and Technical Reference Book" facilitates working with the "Rexroth WinStudio" software for optimal results. This document provides technical information and step-by-step instructions to create web-enabled HMI/SCADA programs.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS HMI Connection DOK-XLCMLC-HMI*****V14-APRS-EN-P, R911341497 This documentation describes the visualization systems supported by the IndraLogic XLC and IndraMotion MLC and their connection.

Tab. 1-11: XCL/MLC documentation overview - HMI

Technology
IndraLogic XLC IndraMotion MLC 14VRS Generic Application Template DOK-XLCMLC-TF*GAT**V14-APRS-EN-P, R911341487 This documentation provides a structured template to the IndraLogic PLC programmer. This template can be used to add and edit the PLC programming code. It includes the template, the template wizard and example applications.
IndraMotion MLC 14VRS Technology Libraries DOK-MLC***-TF*LIB**V14-LIRS-EN-P, R911341511 This documentation describes the function blocks, functions and data types of the "ML_TechInterface.library", "ML_TechMotion.library", "RMB_TechCam.library" and "ML_TechBase.library". It also includes libraries for the winder functionality, register controller functionality and CrossCutter functionality.
IndraWorks 14VRS Energy Efficiency Management DOK-IWORKS-4EE*****V14-APRS-EN-P, R911339229 This documentation describes the HMI connection of the energy management system and the function blocks, functions and data types of the "RIL_4EE" library. The documentation also includes the error reactions of the function blocks.
Rexroth IndraMotion MLC 14VRS RegisterControl (Library) DOK-MLC***-REGI*CO*V14-LIRS-EN-P, R911341509 This documentation describes the inputs and outputs of the individual function blocks and provides notes on their use.
IndraMotion MLC 14VRS RegisterControl (Application Description) DOK-MLC***-REGI*CO*V14-APRS-EN-P, R911341507 This documentation describes the application of the integrated register control for a rotogravure printing machine. The components of the mark stream sensor, the HMI application and the error recovery options are described. This instruction provides information on how to operate the register control, react to errors and query diagnostics. This documentation is written for machine setters and machine operators.
Rexroth IndraMotion MLC 14VRS Winder Function Application DOK-MLC***-TF*WIND*V14-APRS-EN-P, R911341513 This application-related system documentation describes the application of the winder technology functions.
Rexroth IndraWorks CamBuilder 13VRS DOK-IWORKS-CAMBUIL*V13-APRS-EN-P, R911336291 This documentation describes the basic principles and operation of the CamBuilder, the cam editing tool.
IndraMotion MLC 14VRS Robot Control V2 DOK-MLC***-ROCO****V14-RERS-EN-P, R911341588 This documentation provides information about the Robot Control V2. The focus is on PLCopen programming. The program structure, variables, functions, motion statements and the required system parameters are described. This documentation also includes the description of the "ML_Robot" library. Robot Control V2 is supported by a kinematic interface. The components of this interface are described in the "ML_KinTech" library.

Tab. 1-12: XCL/MLC documentation overview - Technology

Open Core Engineering - OCE
IndraLogic XLC IndraMotion MLC 14VRS First Steps MLPI (online help) DOK-XLCMLC-MLPI****V14-CORS-EN-E, R911341033 This document describes the installation and the commissioning of the MLPI interface.
IndraLogic XLC IndraMotion MLC 14VRS Automation Interface (online help) DOK-XLCMLC-AUT*INT*V14-APRS-EN-E, R911341499 This documentation describes the script-based access to IndraWorks project data via the interface of the Automation Interface.

Tab. 1-13: XCL/MLC documentation overview - Open Core Engineering - OCE

SafeLogic
IndraWorks SafeLogic 14VRS First Steps DOK-IWORKS-SL*STEP*V14-CORS-EN-P, R911341520 This documentation describes the initial commissioning of the Safety function module and its external Safety periphery. Taking the example of a project, all required steps to commission a simple Safety application are executed.
IndraLogic XLC IndraMotion MLC 14VRS SafeLogic System Overview DOK-XLCMLC-SL**SYS*V14-PRRS-EN-P, R911341696 This documentation describes the project planning, configuration, creation and commissioning of safety-related devices.
IndraWorks 14VRS SafeLogic - Project Configuration, Version 1.2.0.0. DOK-IWORKS-SL**PRJ*V14-APRS-EN-P, R911341694 This documentation describes the configuration of Safety applications.
IndraControl CFL01.1-F1 Safety Function Module DOK-CONTRL-SL**FM*****-ITRS-EN-P, R911336576 This documentation is provided with the hardware and describes the commissioning, the installation, the decommissioning and disposal of the hardware.

Tab. 1-14: XCL/MLC documentation overview - SafeLogic

Hardware
Rexroth IndraControl L45/L65/L85 Control DOK-CONTRL-ICL45L65L85-PRxx-EN-P, R911332116 This documentation describes the IndraControl L45/L65/L85 controls.
Rexroth IndraControl Lxx 14VRS Function Modules DOK-CONTRL-FM*LXX**V14-APRS-EN-P, R911341583 This documentation describes all function modules of the Lxx controls including engineering and diagnostics.
IndraControl XM21, XM22 Controls DOK-CONTRL-IC*XM2*****-ITRS-EN-P, R911340667 This documentation describes the IndraControl XM21/XM22 controls.
IndraControl XM42 Controls DOK-CONTRL-XM4X*CTRL**-ITRS-EN-P, R911345566 This documentation describes the IndraControl XM42 controls.
IndraControl XFE 01.1 Extension Modules Profibus, RT-Ethernet, Sercos, CAN DOK-CONTRL-XFE**EXTMOD-ITRS-EN-P, R911345570 This documentation describes all extension modules of the XM controls.

xx Corresponding edition

Tab. 1-15: XCL/MLC documentation overview - Hardware

Hydraulics
Rexroth IndraMotion MLC 14VRS Hydraulic Functions DOK-MLC***-TF*HMOT*V14-LIRS-EN-P, R911341907 This documentation describes the function blocks, functions and data types of the MH_TechHydrBase and MH_TechHydrMotion libraries. The documentation also includes the error reactions of the function blocks.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Sequential Programming DOK-XLCMLC-SEQPROG*V14-LIRS-EN-P, R911341909 This documentation describes the sequential programming options, also when using GAT, and the MH_HydrControlProg library.

Tab. 1-16: XCL/MLC documentation overview - Hydraulics

1.6 Terms and abbreviations

Term	Explanation
FBD	Function block diagram, programming language according to IEC 61131-3
IndraDrive	Drive control section
XLC	Compact PLC system with integrated Motion functionality, basic function package PLC
MLC	Compact Motion Logic systems with Motion, Robot and Logic Control functionalities according to function packages
MLD	Drive-based Motion Logic system
IndraWorks	Configuration and commissioning tool by Bosch Rexroth
LD	Ladder diagram, programming language according to IEC 61131-3
MA	MLD Advanced
MLD-M	Motion Logic Drive (multi drive)
MLD-S	Motion Logic Drive (single drive)
PLCopen	It was founded in 1992 and is an international, manufacturer- and product-independent interest group of control manufacturers, software companies and institutes. Mandatory to the PLC standard IEC 61131-3, different standards are defined in technical committees facilitating a significant increase in efficiency in the engineering of the application software
PLS	Programmable Limit Switch
PWM	Pulse-width modulation
ST	Structured text, programming language according to IEC 61131-3
SYNAX	Automation systems for printing and paper-processing machinery
Tech-FB	Technology function block
VisualMotion	Programmable multi-axis motion control system
Target system	Control (e.g. MLC)

Tab. 1-17: Terms and abbreviations

1.7 Customer feedback

Customer requests, comments or suggestions for improvement are of great importance to us. Please email your feedback on the documentations to Feedback.Documentation@boschrexroth.de. Directly insert comments in the electronic PDF document and send the PDF file to Bosch Rexroth.

2 ML_TechBase.library // MX_TechBase.library

2.1 Introduction and overview

General information Technology function blocks (Tech FB) extend the basic functionality of the target systems MLC/XLC and MLD and provide application-specific functionalities. The library includes basic functionalities that can be used for several applications or as standalone functionality.

The function blocks described are provided via the "ML_TechBase" library for MLC/XLC.

This documentation describes the functionality as well as the inputs and outputs of the technology function blocks.



The library requires the PLC function package of the control.

Requirements Technology function blocks of the "ML_TechBase" library require the firmware support of the target system. Specific requirements of the technology function blocks are documented in the chapter of the respective function blocks.

Alias There is a "FP_Aliases" entry in the "Global variables" directory in the library. These are identifiers behind the parameters. The parameters depend on the system. Use these identifiers to be independent from the selected target system.

For example, the "FP_MODULO" identifier

on a MLC/XLC is set to axis parameter A-0-0045, Modulo value

on a MLD system is set to Sercos parameter S-0-0103, modulo value

Thus, the system can abstract better at application level and can write a generally valid code.

Areas of application and function blocks of ML_TechBase

Area of application (folder in library)	
Function blocks	Description
IMC interface (ImcInterface)	
IMC interface, page 19	Command and diagnostic interface
Trigonometric functions (trigonometric functions)	
ML_DegToInc, page 24	Converts angle from degree to corresponding incremental value
ML_DegToRad, page 24	Converts angle in degree to angle in radian measure incl. modulo
ML_IncToDeg, page 25	Converts angle specified in increments to degree
ML_IncToRad, page 25	Converts angle specified in increments to the corresponding radian measure
ML_RadToDeg, page 26	Converts an angle in radian measure to the corresponding angle in degree incl. modulo
ML_RadToInc, page 27	Converts an angle in radian measure to the corresponding incremental value

Auxiliary functions (utilities/system info)	
MB_GetSystemInfo, page 27	Checks function packages and system options

Tab. 2-1: Function Blocks in the ML_TechBase Library

_PhaseOut The subsequently specified function blocks (and their data types) have been moved to other libraries due to organizational reasons.
Sporadically, the name had to be adjusted.
Due to compatibility reasons, the function block is still available under the old name.

New library: RIL_LoopControl from 14.3.0.0

Folder (new)	Name (new)	POU type	Name (old)
TemperatureController	IL_TempControlType01	FB	
	IL_TempControlType02	FB	
TwoPosCtrl	IL_TwoPosCtrlType01	FB	ML_TwoPosCtrlType01

Tab. 2-2: POU's which have been moved from ML_TechBase to RIL_LoopControl

New library: RIL_Uilities from 14.3.0.0

Folder (new)	Name (new)	POU type	Name (old)
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 which have been moved from ML_TechBase to RIL_Uilities

New library: ML_TechMotionBase from 14.3.0.0

Folder (new)	Name (new)	POU type	Name (old)
AxisBackupRestore	ML_AxisBackup	FB	
	ML_AxisRestore	FB	

Folder (new)	Name (new)	POU type	Name (old)
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 which have been moved from ML_TechBase to ML_TechMotionBase

2.2 IMC interface

2.2.1 Introduction and Overview

The ML_TechBase libraries for MLC controls and the MX_ImcInterface for MLD-M drives provide an easy to use interface for the system.

The interface is called an IMC interface - (I)ndra (M)otion (C)ontrol interface.

Functions of the IMC interface

- Selects MLC and operating mode, P0 (Phase 0), P2 (Parameter Mode), BB (Operating Mode) for MLD-M.
- Executes the "Clear All Errors" (delete all errors) command.
- Executes "RetriggerOpMode" to allow users to re-execute the selected operation mode. Reset the control for the IMC interface operating mode if the Sercos ring is disconnected.
- In case of phase switching via an external instance (e.g. IndraWorks), the IMC interface sets the output "PassiveMode" and the selected mode is not actuated.

Use "RetriggerOpMode" or switch back into the selected mode to exit the "PassiveMode" again.



The following library names used depend on the system:

- MLC control systems use the library ML_TechBase, PLC basic package.
- MLD-M drive systems use the library MX_ImcInterface.lib

The following table lists the user's interface to the control interface data structure:

User interface	Type	Description
ImcCtrl[]	MB_IMC_CONTROL_TYPE01, page 20	Control structure that provides the inputs for operation mode, clearing errors and "retriggering the operation mode".
ImcStatus[]	MB_IMC_STATUS_TYPE01, page 21	Status structure that includes operation mode acknowledgements and diagnostic information.

Tab. 2-5: Control interface - user interface data structures

In order to use the IMC interface, only the program "MB_ImcInterface" must be called cyclically, e.g. in the PlcTask. When using the GAT or. GAT ^{compact} the program call is already integrated.

Libraries ML_TechBase for MLC, PLC basic package

MX_ImcInterface.lib for MLD-M

Dependence MX_AxisInterface.lib for MLD-M

2.2.2 MB_IMC_CONTROL_TYPE01

Brief description The "MB_IMC_CONTROL_TYPE01" data structure is accessed via the ImcCtrl[] instance.

Using the structure [MB_IMC_ADMINISTRATION, page 20](#) allows you to issue a task to the MotionControl component.

Name	Type	Description
Admin	MB_IMC_ADMINISTRATION	Task submission

Tab. 2-6: STRUCT MB_IMC_STATUS_TYPE01

2.2.3 MB_IMC_ADMINISTRATION

The "MB_IMC_ADMINISTRATION" structure supports giving commands to the control (LogicControl ⇒ MotionControl).

Name	Type	Description
_OpMode	MB_IMC_MODE_CONTROL, page 21	Operation mode selection (P0, P2, P4)
ClearError	BOOL	(Edge-controlled) deleting of all control errors (C-0-1030)
RetriggerOpMode	BOOL	New "triggering the operating mode"
arCheckSercos	ARRAY [ML_MIN_SERCOS_IFC..ML_MAX_SERCOS_IFC] OF BOOL	The Sercos phase of the onboard Sercos interface (ML_LOCAL_SERCOS_IFC) has to be taken into consideration in the calculation of "ImcStatus.Admin._OpModeAck ". Default = TRUE Array for future extensions with several interfaces. Only available in MLC

Tab. 2-7: Elements of the "MB_IMC_ADMINISTRATION" structure

2.2.4 MB_IMC_MODE_CONTROL

The enumeration type "MB_IMC_MODE_CONTROL" specifies the required operation mode.

Name	Value	Description
IMC_MODE_CTRL_P0	0	Selection of phase P0
IMC_MODE_CTRL_P2	2	Selection of phase P2
IMC_MODE_CTRL_P4	4	Selection of phase P4

Tab. 2-8: Elements of the "MB_IMC_ADMINISTRATION" structure

2.2.5 MB_IMC_STATUS_TYPE01

Brief description The "MB_IMC_STATUS_TYPE01" data structure is accessed via the **ImcStatus[]** instance.

On the one hand, it contains diagnostic information on the task issued to the MotionControl component ([MB_IMC_ADMIN_STATUS](#), page 21) and on the other, general diagnostic information on the control itself ([MB_IMC_DIAGNOSIS](#), page 23).

Name	Type	Description
InOperation	BOOL	TRUE: ImcInterface is operating
Admin	MB_IMC_ADMIN_STATUS	Diagnosis for issued task
Diag	MB_IMC_DIAGNOSIS	General diagnosis for control
PassiveMode	BOOL	TRUE: External source controls operation mode

Tab. 2-9: STRUCT MB_IMC_STATUS_TYPE01

2.2.6 MB_IMC_ADMIN_STATUS

The "MB_IMC_ADMIN_STATUS" structure contains status information on the currently running task (MotionControl → LogicControl).

Name	Type	Description
_OpModeAck	MB_IMC_MODE_STATUS	Confirmation of the selected operation mode P0..P2..BB. Only for MLC: Calculation of the control mode and Sercos phase (if not disabled in ImcCtrl.Admin.arCheckSercos[]). MB_IMC_MODE_STATUS , page 22
ModeStatus_P0	BOOL	Imc mode 0 confirmation
ModeStatus_P1	BOOL	Imc mode 1 confirmation
ModeStatus_P2	BOOL	Imc mode 2 confirmation
ModeStatus_P3	BOOL	Imc mode 3 confirmation
ModeStatus_BB	BOOL	Imc mode BB confirmation
PwrUpModeAck	BOOL	Control reaches PowerUp TargetMode (C-0-0450)
PhaseSwitchActive	BOOL	Sercos phase switching is active

Name	Type	Description
arSercosPhase	ARRAY [ML_MIN_SERCOS_IFC..ML_MAX_SERCOS_IFC] OF MB_IMC_SERCOS_PHASE	The Sercos phase of the onboard Sercos interface (ML_LOCAL_SERCOS_IFC). Array for future extensions with several Sercos interfaces. Only available in MLC MB_IMC_SERCOS_PHASE, page 22
ControlMode	MB_IMC_CONTROL_MODE_STATUS	Control mode (C-0-0023, Bits 27..31). Only available in MLC MB_IMC_CONTROL_MODE_STATUS, page 22

Tab. 2-10: Elements of the "MB_IMC_ADMIN_STATUS" structure

2.2.7 MB_IMC_MODE_STATUS

The enumeration type "MB_IMC_MODE_STATUS" contains the confirmation of the selected operating mode.

Name	Value	Description
IMC_MODE_STATUS_P0	0	Confirmation of the lmc mode 0
IMC_MODE_STATUS_P1	1	Confirmation of the lmc mode 1
IMC_MODE_STATUS_P2	2	Confirmation of the lmc mode 2
IMC_MODE_STATUS_P3	3	Confirmation of the lmc mode 3
IMC_MODE_STATUS_BB	4	Confirmation of the lmc mode BB
IMC_MODE_UNDEFINED	255	lmc mode undefined

Tab. 2-11: Elements of the "MB_IMC_ADMINISTRATION" structure

2.2.8 MB_IMC_SERCOS_PHASE

The enumeration type "MB_IMC_SERCOS_PHASE" contains the confirmation of the selected Sercos phase (only available in MLC).

Name	Value	Description
IMC_SERCOS_PHASE_P0	0	Sercos phase P0 confirmation
IMC_SERCOS_PHASE_P1	1	Sercos phase P1 confirmation
IMC_SERCOS_PHASE_P2	2	Sercos phase P2 confirmation
IMC_SERCOS_PHASE_P3	3	Sercos phase P3 confirmation
IMC_SERCOS_PHASE_BB	4	Sercos phase P4 confirmation
IMC_SERCOS_PHASE_UNDEFINED	255	Sercos phase undefined

Tab. 2-12: Elements of the "MB_IMC_SERCOS_PHASE" structure

2.2.9 MB_IMC_CONTROL_MODE_STATUS

The enumeration type "MB_IMC_CONTROL_MODE_STATUS" contains the confirmation of the selected control mode (only available in MLC).

Name	Value	Description
IMC_CONTROL_MODE_STATUS_P0	0	Control mode 0 confirmation
IMC_CONTROL_MODE_STATUS_P1	1	Control mode 1 confirmation

Name	Value	Description
IMC_CONTROL_MODE_STATUS_P2	2	Control mode 2 confirmation
IMC_CONTROL_MODE_STATUS_P3	3	Control mode 3 confirmation
IMC_CONTROL_MODE_STATUS_BB	4	Control mode 4 confirmation
IMC_CONTROL_MODE_UNDEFINED	255	Control mode undefined

Tab. 2-13: Elements of the "MB_IMC_CONTROL_MODE_STATUS" structure

2.2.10 MB_IMC_DIAGNOSIS

This structure contains diagnostic information on the control (MotionControl ⇒ LogicControl).

Name	Type	Description
Error	BOOL	Control error (C-0-0023, bit 26: System status, error)
Warning	BOOL	Control warning (C-0-0023, bit 25: System status, warning)
ErrorID	ERROR_CODE	Error code, only valid if Error= TRUE
ErrorIdent	ERROR_STRUCT	Detailed error information, only valid if Error= TRUE
ClearErrorAck	BOOL	Confirmation of execution of clear error command
ErrSeverity	MB_IMC_ERROR_SEVERITY, page 23	Error category
Number	DWORD	Display of error number (C-0-0627, diagnosis number)
Message	STRING	Display of error text (C-0-0626, diagnosis)

Tab. 2-14: Elements of the "MB_IMC_DIAGNOSIS" structure

2.2.11 MB_IMC_ERROR_SEVERITY

The enumeration type "MB_IMC_ERROR_SEVERITY" from type WORD indicates the severity of the error / warning. It supports the structure of language-dependent texts.

Name	Value	Description
IMC_ERROR_UNDEFINED	16#00	Error undefined
IMC_ERROR_F0	16#F0	F0 error (non-fatal error)
...	...	...
IMC_ERROR_F9	16#F9	F9 error (fatal system error)
IMC_WARNING_E0	16#E0	E0 warning
...	...	...
IMC_WARNING_E9	16#E9	E9 warning

Tab. 2-15: Elements of the "MB_IMC_ERROR_SEVERITY" enumeration

2.3 Trigonometric functions

2.3.1 Introduction and overview

The following functions allow users to convert between various units.

Since these are functions, no errors can be output. If invalid values are used, 0 is returned.

2.3.2 ML_DegToInc

Brief description The function ML_DegToInc converts the angle indicated in degrees ($0^\circ \leq x < 360^\circ$) to the corresponding increment value ($0 \leq x < 2^20$).

Assignment: Target system/library

Target system	Library
MLC/XLC 14VRS PLC basic package	ML_TechBase
MLD (IL 2G)	MX_TechBase

Tab. 2-16: Reference table

Interface description

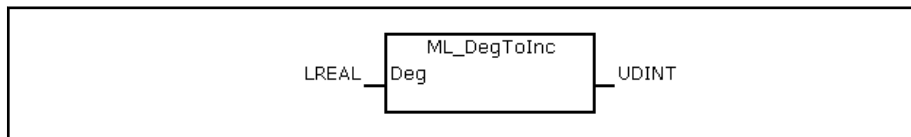


Fig. 2-1: ML_DegToInc function

I/O type	Name	Data type	Description
VAR_INPUT	Deg	LREAL	Angle in degrees [DEG]
Return value		UDINT	Angle in increments

Tab. 2-17: Interface variables of the ML_DegToInc function

Min./max. and default value

The following table lists the min./max. and default values of the function block input:

Name	Type	Min. value	Max. value	Default value
Deg	LREAL	$-2^{32} * 360.0$	$2^{32} * 360.0$	0.0

Tab. 2-18: Min./max. and default value for the ML_DegToInc function

Functional description

The function ML_DegToInc converts the angle indicated in degrees ($0^\circ \leq x < 360^\circ$) to the corresponding increment value ($0 \leq x < 2^20$). If an angle is specified as larger than or equal to 360° , it is converted to the respective angle within the valid value area (modulo 360).

If a value outside the valid value range is specified, 0 is returned.

Error codes

The function ML_DegToInc does not generate any errors.

2.3.3 ML_DegToRad

Brief description The function ML_DegToRad converts a phase value in degrees to the corresponding radian incl. modulo.

Assignment: Target system/library

Target system	Library
MLC/XLC 14VRS PLC basic package	ML_TechBase
MLD (IL 2G)	MX_TechBase

Tab. 2-19: Reference table

Interface description

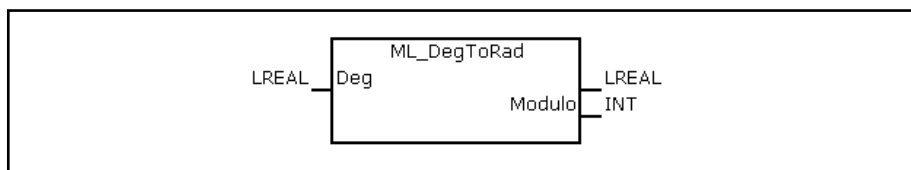


Fig. 2-2: ML_DegToRad function

I/O type	Name	Data type	Description
VAR_INPUT	Deg	LREAL	Angle [DEG]
Return value	Rad	LREAL	Angle [RAD]
VAR_OUTPUT	Modulo	INT	Modulo

Tab. 2-20: Interface variables of the ML_DegToRad function

Min./max. and default value The following table lists the min./max. and default values of the function block input:

Name	Type	Min. value	Max. value	Default value
Deg	LREAL	$-2^{15} * 360.0$	$2^{15} * 360.0$	0.0

Tab. 2-21: Min./max. and default value for the ML_DegToRad function

Functional description At an instance call, the ML_DegToRad function cyclically converts the indicated angle in degrees to the respective radian incl. modulo.

If a value outside the valid value range is specified, 0 is returned.

Error codes The function ML_DegToRad does not generate any errors.

2.3.4 ML_IncToDeg

Brief description The function ML_IncToDeg converts the angle specified in increments ($0 \leq x < 2^{20}$) to the corresponding angle in degrees ($0^\circ \leq x < 360^\circ$).

Assignment: Target system/library

Target system	Library
MLC/XLC 14VRS PLC basic package	ML_TechBase
MLD (IL 2G)	MX_TechBase

Tab. 2-22: Reference table

Interface description

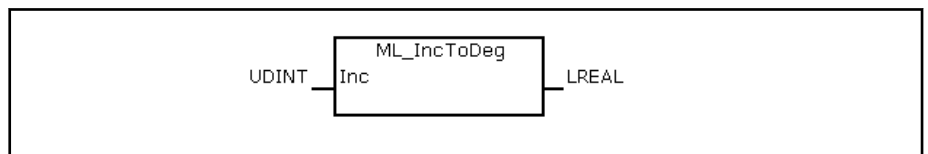


Fig. 2-3: ML_IncToDeg function

I/O type	Name	Data type	Description
VAR_INPUT	Inc	UDINT	Angle in increments
Return value		LREAL	Angle in degrees [DEG]

Tab. 2-23: Interface variable of the ML_IncToDeg function

Functional description The function ML_IncToDeg converts the angle specified in increments ($0 \leq x < 2^{20}$) to the corresponding angle in degrees ($0^\circ \leq x < 360^\circ$). If an angle is specified larger than or equal to 2^{20} increments, it is converted to the respective angle in the valid value range (modulo 2^{20}).

Error codes The function ML_IncToDeg does not generate any errors.

2.3.5 ML_IncToRad

Brief description The function ML_IncToRad converts the angle specified in increments ($0 \leq x < 2^{20}$) to the corresponding angle in radian ($0 \leq x < 2\pi$).

ML_TechBase.library // MX_TechBase.library

Assignment: Target system/library

Target system	Library
MLC/XLC 14VRS PLC basic package	ML_TechBase
MLD (IL 2G)	MX_TechBase

Tab. 2-24: Reference table

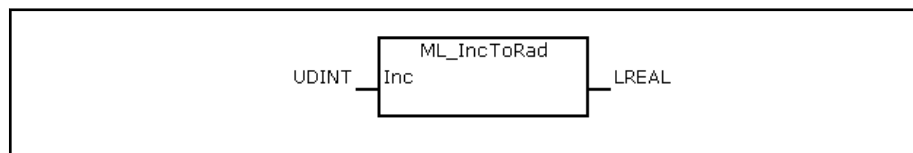
Interface description

Fig. 2-4: ML_IncToRad function

I/O type	Name	Data type	Description
VAR_INPUT	Inc	UDINT	Angle in increments
Return value		LREAL	Angle in radian [RAD]

Tab. 2-25: Interface variables of the ML_IncToRad function

Functional description

The function ML_IncToRad converts the angle specified in increments ($0 \leq x < 2^{20}$) to the corresponding angle in radian ($0 \leq x < 2 \cdot \pi$). If an angle is specified larger than or equal to 2^{20} increments, it is converted to the respective angle in the valid value range (modulo 2^{20}).

Error codes

The function ML_IncToRad does not generate any errors.

2.3.6 ML_RadToDeg

Brief description

The function ML_RadToDeg converts an phase value in radian to the corresponding degrees incl. Modulo.

Assignment: Target system/library

Target system	Library
MLC/XLC 14VRS PLC basic package	ML_TechBase
MLD (IL 2G)	MX_TechBase

Tab. 2-26: Reference table

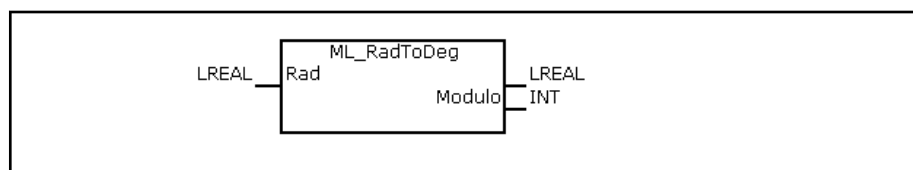
Interface description

Fig. 2-5: ML_RadToDeg function

I/O type	Name	Data type	Description
VAR_INPUT	Rad	LREAL	Angle [RAD]
Return value		LREAL	Angle [DEG]
VAR_OUTPUT	Modulo	INT	Modulo

Tab. 2-27: Interface variable of the ML_RadToDeg function

Min./max. and default value

The following table lists the min./max. and default values of the function block input:

Name	Type	Min. value	Max. value	Default value
Rad	LREAL	$-2 * \pi * 2^{15}$	$2 * \pi * 2^{15}$	0.0

Tab. 2-28: Min./max. and default value for the ML_RadToDeg function

Functional description At an instance call, the function ML_RadToDeg cyclically converts the indicated angle in radian to the respective degrees incl. Modulo.

If a value outside of the valid value range is specified, 0 is output for "Deg" and "Modulo".

Error codes The function ML_RadToDeg does not generate any errors.

2.3.7 ML_RadToInc

Brief description The function ML_RadToInc converts the angle indicated in radian ($0 \leq x < 2\pi$) to the corresponding increment value ($0 \leq x < 2^{20}$).

Assignment: Target system/library

Target system	Library
MLC/XLC 14VRS PLC basic package	ML_TechBase
MLD (IL 2G)	MX_TechBase

Tab. 2-29: Reference table

Interface description

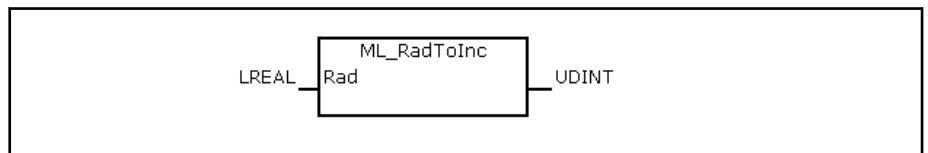


Fig. 2-6: ML_RadToInc function

I/O type	Name	Data type	Description
VAR_INPUT	Rad	LREAL	Angle in radian [RAD]
Return value		UDINT	Angle in increments

Tab. 2-30: Interface variable of the ML_RadToInc function

Min./max. and default value The following table lists the min./max. and default values of the function block input:

Name	Type	Min. value	Max. value	Default value
Rad	LREAL	$-2 * \pi * 2^{32}$	$2 * \pi * 2^{32}$	0.0

Tab. 2-31: Min./max. and default value for the ML_RadToInc function

Functional description The function ML_RadToInc converts the angle indicated in radian ($0 \leq x < 2\pi$) to the corresponding increment value ($0 \leq x < 2^{20}$). If an angle is specified larger than or equal to 2π , it is converted to the respective angle within the valid value area (modulo 2π).

If a value outside the valid value range is specified, 0 is returned.

Error codes The function ML_RadToInc does not generate any errors.

2.4 System information

2.4.1 MB_GetSystemInfo

Brief description This function block checks and system options and displays all activated system options together with the system type.

ML_TechBase.library // MX_TechBase.library

Assignment: Target system/library

Target system	Library
MLC/XLC 14VRS PLC basic package	ML_TechBase
MLD (IL2G)	MX_TechBase

Tab. 2-32: Reference table of the MB_GetSystemInfo

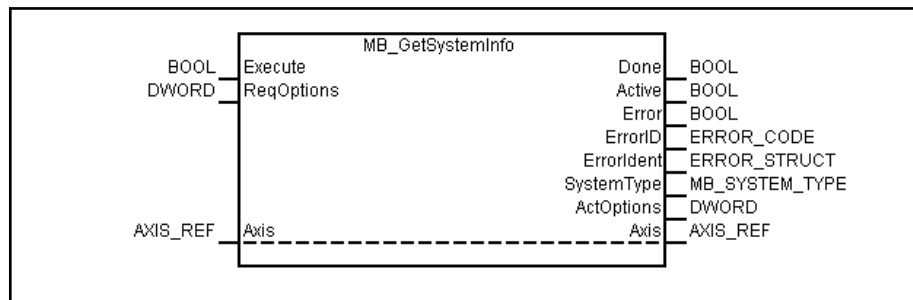
Interface description

Fig. 2-7: MB_GetSystemInfo function block

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Starts the check
	ReqOptions	DWORD	Required system options
VAR_OUTPUT	Done	BOOL	Check completed without errors
	Active	BOOL	The function block is active
	Error	BOOL	Indicates that an error occurred
	ErrorID	ERROR_CODE	Basic error cause
	ErrorIdent	ERROR_STRUCT	Error structure with further error division Error codes, page 30
	SystemType	MB_SYSTEM_TYPE	System type MB_SYSTEM_TYPE, page 30
	ActOptions	DWORD	Actually enabled system options
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Axis to be checked

Tab. 2-33: Interface variables of the MB_GetSystemInfo function block

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
ReqOptions	DWORD	0	N. def.	0	Rising edge at "Execute"

Tab. 2-34: Input behavior of the ML_GetSystemInfo

Functional description

System MLC/XLC: From 14VRS, this function blocks operates independently from the Sercos phase.

MLD system: This function block operates in the Sercos phases 2 or 4.

At a rising edge at "Execute", the "ReqOptions" input is applied. Subsequently, the system is checked.

If options or function packages are set in "ReqOptions" which are not available, a corresponding error is generated (see [Error codes, page 30](#)).

After a successful completion, the system type is displayed in "SystemType" and all enabled options are displayed in "ActOptions" and "Done" is set to TRUE.

Option	MLC/XLC	MLD	Value	Description
MB_OPEN_LOOP	Axis	Axis	16#00000001	Basic package "Open Loop"
MB_CLOSED_LOOP	Axis	Axis	16#00000002	Basic package "Close-dLoop"
MB_SERVO		Axis	16#00000010	Function package "Servo"
MB_SYNCHRONIZATION	Axis	Axis	16#00000020	Function package synchronization during interpolation in the drive or virtual axis
MB_MAIN_SPINDLE		Axis	16#00000040	Function package "Main spindle"
MB_TF	Always set	First axis	16#00000200	Option "Advanced technology function"
MB_ML	Always set	First axis	16#00000200	Option "Technology function"
MB_MA	Always set	First axis	16#00000300	
MB_SAFETY			16#00010000	Safety option in axis enabled
MB_AXISDATA	Local axes	First axis	16#00100000	AxisData enabled
MB_REALAXIS	Real axis	Always set	16#01000000	
MB_VIRTUALAXIS	Virtual axis		16#02000000	
MB_LINKAXIS	Axes via cross link		16#04000000	
MB_LOCAL_INTERPOL	Interpolation in the control		16#10000000	
MB_INDRADRIIVE	IndraDrive	Always set	16#20000000	
MB_HYDRAULICDRIVE	HydraulicDrive		16#40000000	

Tab. 2-35: Possible basic packages, function packages and options in "ReqOptions" and "ActOptions"



For the currently possible options, refer to the GVL "SystemOptions".

Required options are to be specified in "ReqOptions".

If at least one option is not available on the specified axis, a corresponding error message is generated.

If all options requested in "ReqOptions" are available, "Done" becomes active.

It is also possible to specify options that are not required and to perform the evaluation independently.

Checking the options requires several function block calls. During this period, the "Active" output is set.

Error behavior

If an option specified in "ReqOptions" is not available or if another error occurs, "Error" is set to TRUE.

Error codes The function block generates the following error messages of the "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
ACCESS_ERROR	16#1503	16#0000	AxisData structure not enabled
RESOURCE_ERROR	16#000F	16#0001	Required TF option not enabled
RESOURCE_ERROR	16#000F	16#0002	Required ML option not enabled
RESOURCE_ERROR	16#000F	16#0003	Required MA option not enabled
RESOURCE_ERROR	16#000F	16#0010	Required basic package "OpenLoop" not active
RESOURCE_ERROR	16#000F	16#0011	Required basic package "ClosedLoop" not active
RESOURCE_ERROR	16#000F	16#0020	Required function package "Synchronization" is not active
RESOURCE_ERROR	16#000F	16#0021	Required function package "Servo" is not active
RESOURCE_ERROR	16#000F	16#0022	Required functional package "Main Spindle" is not active
RESOURCE_ERROR	16#000F	16#0023	Required axis type (REAL,VIRTUAL, LINK) is incorrect

Tab. 2-36: Error codes of the MB_GetSystemInfo function block

2.4.2 MB_SYSTEM_TYPE

The enumeration types includes all currently available control systems.

System type		Description
MB_SYSTEM_UNKNOWN	0	Unknown 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: Possible system types

3 ML_TechMotionBase.library // MX_TechMotionBase.library

3.1 Introduction and overview

General information Technology function blocks (Tech FB) extend the basic functionality of the target systems MLC/XLC and MLD and provide application-specific functionalities. The library includes basic functionalities that can be used for several applications or as standalone functionality.

The function blocks described are provided via the "ML_TechMotionBase" library for MLC/XLC.

This documentation describes the functionality as well as the inputs and outputs of the technology function blocks.

Prerequisites Technology function blocks of the "ML_TechMotionBase" library require the firmware support of the target system. Specific prerequisites of the technology function blocks are documented with the corresponding function blocks.



The library requires the **MOT (Motion) function package** of the control system.

Areas of application and function blocks of ML_TechMotionBase

Area of application (folder in library)	
Function blocks	Description
Backing up and restoring axes (AxisBackupRestore)	
ML_AxisBackup, page 33	Parameter backup of an axis on the Compact Flash card
ML_AxisRestore, page 36	Parameter restoration of an axis from the Compact Flash card
Measurement encoder, determination of traveled distances (DistanceAccumulator)	
MB_DistanceAccumulator, page 47	Sums up the relatively traveled distance using an encoder and stops when the summed up distance is higher than or equal to the TotalDistance input value. The current encoder position is entered using the CurrentPosition input
MB_AxisDistanceAccumulator, page 48	Sums up the relatively traveled distance using an encoder and stops when the summed up distance is higher than or equal to the TotalDistance input value. This function block uses the axis reference input (AXIS_REF) to query the current encoder position automatically
Cam tables, download (LoadCamData)	
MB_LoadCamData, page 94	Downloads cams to the drive and control to encapsulate the existing detail differences
Programmable limit switches (PhaseLimitSwitch)	
MC_DigitalCamSwitch, page 50	Implements a freely programmable limit switch with up to 32 switches on 8 outputs

MB_DigitalCamSwitchOvs, page 66	Implements a freely programmable limit switch with up to 64 switches on 32 switch tracks Supports oversampling. Thus, a corresponding higher temporal resolution of the PLS signals can be achieved Requires I/O modules supporting oversampling
Filtering a position (PositionFiltering)	
MB_PositionFilterType01, page 98	Filters an incremental position and calculates position and velocity
Position monitoring (PositionMonitoring)	
MB_PositionMonitoring_GantryCant, page 101	Monitors the relative position of two axes with regard to the tilting to each other. The retract motion is possible using the tolerance band functionality. The direction of the motion out of the tilting is continuously monitored
MB_PositionMonitoring_PosCollision, page 103	Monitors the absolute and relative position of two axes with regard to indentation and collision
S-curve (SCurve)	
MB_ReachVelocityType02, page 106	Calculates the respective jerk, acceleration and target velocity to travel along the specified jerk-limited velocity characteristics (e.g. S-curve) and allows up to three changes in the target velocities
Touch probe function (TouchProbe)	
MB_InitTouchProbe, page 120	Initializing the touch probe functionality
MB_TouchProbe, page 128	Recording an axis position, master axis position, time or single-shot measurement
MB_TouchProbeContinuous, page 131	Recording an axis position, master axis position or time at continuous measurement
MB_WriteExpectWindow, page 134	Configures the expectation window for the touch probe functionality
MB_AbortTrigger, page 137	Cancels the MB_TouchProbe function block
Cyclic data channel, management (Utilities/CyclicData)	
MB_AllocCyclicParameter, page 141	Checking and revising the configuration of a parameter to be read or written cyclically in the cyclic data channel
MB_FreeCyclicParameter, page 143	Enabling an already configured parameter to be read or written cyclically in the cyclic data channel
MB_GetCyclicChannelConfig, page 145	Reads the configuration of cyclic read and write data channels and saves them in a data structure
MB_GetCyclicParameterHandle, page 146	Searches for a cyclic parameter to be read or written in the cyclic data channel and stores all required information in one handle
MB_GetLastCyclicParameterError, page 150	Returns the last error while reading or writing a cyclic value for a handle and deletes the error
MB_ReadCyclicParameter, page 148	Reads a cyclic parameter via the handle and returns the current content as DINT
MB_ReadCyclicRealParameter, page 149	Reads a cyclic parameter via the handle and returns the current content as REAL

MB_WriteCyclicParameter, page 149	Writes a cyclic parameter via the handle. Number of decimals are converted according to the Sercos declaration
MB_WriteCyclicRealParameter, page 150	Writes a cyclic parameter via the handle. Number of decimals are rounded

Tab. 3-1: ML_TechMotionBase library

3.2 Backing up and restoring drive parameters

3.2.1 Introduction and overview

Drives are normally parameterized via IndraWorks. At running systems, it is often required to exchange drives without using IndraWorks and to parameterize them again.

This use case is covered by the following function blocks.

To restore parameters, the drive has to be in the parameterization mode (P2). This is also recommended for the parameter backup.

- [ML_AxisBackup, page 33](#), creates a parameter backup and stores it as text file on the Compact Flash card of the control.
- [ML_AxisRestore, page 36](#), loads the parameter backup again to the axis
- [ML_AxisBackup_Type02, page 39](#), backs up the parameters of an axis in a text file on the Compact Flash card of the control
- [ML_AxisRestore_Type02, page 42](#), loads the parameters from a text file on the Compact Flash card of the control to an axis

3.2.2 ML_AxisBackup

Brief description The ML_AxisBackup function block backs up the parameters of an axis in a text file on the Compact Flash card of the control.

 In version 14VRS, the "ML_AxisRestore (ML_TechBase library)" function block was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC/XLC 14VRS MOT function package	ML_TechMotionBase

Tab. 3-2: Reference table of the ML_AxisBackup

Interface description

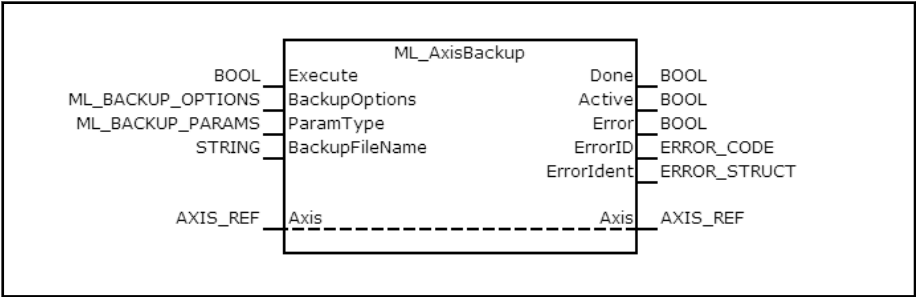


Fig. 3-1: ML_AxisBackup function block

ML_TechMotionBase.library // MX_TechMotionBase.library

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Processing of the function block enabled (edge-controlled)
	BackupOptions	ML_BACKUP_OPTIONS	Scope of the parameter backup ML_BACKUP_OPTIONS, page 45
	ParamType	ML_BACKUP_PARAMS	Selection of the parameters to be backed up: all, A, S and P ML_BACKUP_PARAMS, page 46
	BackupFileName	STRING	File name on the Compact Flash card of the control
VAR_OUTPUT	Done	BOOL	Processing completed, output data valid
	Active	BOOL	Processing running, output data valid
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Error diagnostics (refer to Error codes, page 35)
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Reference to the axis

Tab. 3-3: Interface variables of the ML_AxisBackup function block

Min./max. and default values of the inputs

The following value ranges and default values can be applied at the function block:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
BackupOptions	ML_BACKUP_OPTIONS			TO_SAVE_PARAMS	Rising edge at "Execute"
ParamType	ML_BACKUP_PARAMS			ASP_PARAMS	Rising edge at "Execute"
BackupFileName	STRING			"	Rising edge at "Execute"

Tab. 3-4: Min./max. and default values of the ML_AxisBackup function block

Functional description

The ML_AxisBackup function block backs up the parameters of an axis in a text file on the Compact Flash card partition of the control. If the file already exists, it is overwritten. The file can be read again by the ML_AxisRestore function blocks and by IndraWorks.

BackupOptions

The parameters to be backed up are selected via the "BackupOptions" input. The following selections are possible:

- ALL_PARAM: All parameters are backed up. This option is primarily required to check the configuration
- MODIFIED_PARAM: The parameters deviating from the default values are backed up
- TO_SAVE_PARAM: The parameters required for restoration are backed up. This is the presetting



The "BackupOptions" is currently only evaluated for S- and P-parameters. When exporting A-parameters, all parameters (ALL_PARAM) or the parameters required for restoring data (MODIFIED_PARAM and TO_SAVE_PARAM) are always exported.

ParamType

The parameter types to be backed up are selected via the "ParamType" input. The following options can be selected:

- ASP_PARAMS: Parameters in the drive and control-internal axis parameters
- A_PARAMS: Control-internal axis parameters
- SP_PARAMS: Parameters in the drive

BackupFileName

Via "BackupFileName", the file name on the Compact Flash card of the control can be specified. It is possible to use a path specification. The path has to be available on the Compact Flash card.

Example:

- "Drive1.par" (stores the file "Drive1.par" on the user partition of the Compact Flash card.)
- "Params/drive1.par" (stores the file "Drive1.par" in the "Params" directory on the user partition of the Compact Flash card)
- "/ata0a/drive2.par" (stores the file "drive2.par" on the OEM partition of the control)

Time response

The function block ML_AxisBackup works asynchronously. Due to a rising edge at the "Execute" input after the start, it reads the inputs "BackupOptions", "ParamType" and "BackupFileName". "Active" is set as long as the function block is operating. After the successful operation, the "Done" output is set. If an error occurs, the "Error" output is set.

Error codes

The function block uses the error table **F_RELATED_TABLE**. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR	16#0000000D	16#00000000	AxisRef is not in the valid range
INPUT_RANGE_ERROR	16#00001E00	16#00000000	Invalid "BackupOptions" input
INPUT_RANGE_ERROR	16#00001E01	16#00000000	Invalid "ParamType" input
RESOURCE_ERROR	16#00001E02	16#00000000	The parameter type selected in "ParamType" is not supported by this axis
RESOURCE_ERROR	16#00001E03	16#xxxxxxx	Error when reading a parameter
ACCESS_ERROR	16#00001E10	16#00000000	Error when creating the backup file
ACCESS_ERROR	16#00001E12	16#00000000	Error when writing the backup file
ACCESS_ERROR	16#00001E14	16#00000000	Error when closing the backup file
STATE_MACHINE_ERROR	16#00000006	16#0000FFx	Invalid state of the function block

Tab. 3-5: Error codes of the ML_AxisBackup function block

3.2.3 ML_AxisRestore

Brief description The ML_AxisRestore function block loads the parameters from a text file on the Compact Flash card of the control to an axis.



In version 14VRS, the "ML_AxisRestore (ML_TechBase library)" function block was moved to the "TechMotionBase library".

Assignment: Target system/library

Target system	Library
MLC/XLC 14VRS	ML_TechMotionBase
MOT function package	

Tab. 3-6: Reference table of the ML_AxisRestore

Interface description

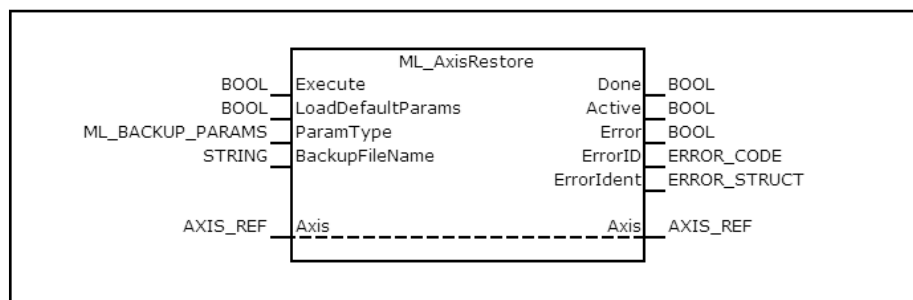


Fig. 3-2: ML_AxisRestore function block

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Processing of the function block enabled (edge-controlled)
	LoadDefaultParams	BOOL	Load default parameters before loading the parameters from the file
	ParamType	ML_BACKUP_PARAMS	Selection of the parameters to be restored: All A, S and P ML_BACKUP_PARAMS, page 46
	BackupFileName	STRING	File name on the Compact Flash card of the control
VAR_OUTPUT	Done	BOOL	Processing completed, output data valid
	Active	BOOL	Processing running, output data valid
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Error diagnostics (refer to Error codes, page 38)
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Reference to the axis

Tab. 3-7: Interface variables of the ML_AxisRestore function block

Min./max. and default values of the inputs

The following value ranges and default values can be applied at the function block:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
LoadDefaultParams	BOOL			FALSE	Rising edge at "Execute"

Name	Type	Min. value	Max. value	Default value	Effective
ParamType	ML_BACK-UP_PARAMS			ASP_PARAMS	Rising edge at "Execute"
BackupFileName	STRING			"	Rising edge at "Execute"

Tab. 3-8: Min./max. and default values of the ML_AxisRestore function block

Functional Description

The ML_AxisRestore function block loads the parameters from a text file on the Compact Flash card of the control to an axis. After loading the parameter file successfully, the configuration is saved on the drive or the axis. It is thus also available after deactivation. The parameter file can be created by the ML_AxisBackup function block or by IndraWorks.

The function block does **not** back up the following parameters, even if they are in the parameter file:

- All parameters that are read-only
- All parameters that are commands
- All currently write-protected parameters (e.g. protected by password, write-protected as PLC in RUN, etc.)
- Parameters 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

If the "LoadDefaultParams" input is set, the default parameters of the drive or the axis are loaded before loading the parameters. This is especially reasonable if the parameter file was created using the options "MODIFIED_PARAM" or "TO_SAVE_PARAM".



Loading the default parameters takes several seconds per axis.



When loading the default parameters for an IndraDrive, the parameter P-0-4090 is written with the value 16#51 before triggering the command. Thus, default values are not assigned to the parameters of the master communication and the Engineering interface. If further and/or other parameter groups are not initialized, load the default parameters personally.

ParamType

The parameter types to be restored, are selected via the "ParamType" input. The following options can be selected:

- ASP_PARAMS: Parameters in the drive and control-internal axis parameters
- A_PARAMS: Control-internal axis parameters
- SP_PARAMS: Parameters in the drive

BackupFileName

Via "BackupFileName", the file name on the Compact Flash card of the control can be specified. It is possible to use a path specification. The path has to be available on the Compact Flash card.

ML_TechMotionBase.library // MX_TechMotionBase.library

Example:

- "Drive1.par" (reads the file "Drive1.par" from the user partition of the Compact Flash card.)
- "Params/drive1.par" (reads the file "Drive1.par" from the "Params" directory on the user partition of the Compact Flash card)
- "/ata0a/drive2.par" (reads the file "drive2.par" from the OEM partition of the control)

Time response

The function block ML_AxisRestore operates asynchronously. Due to a rising edge at the "Execute" input after the start, it reads the inputs "LoadDefaultParams", "ParamType" and "BackupFileName". "Active" is set as long as the function block is operating. After the successful operation, the "Done" output is set. If an error occurs, the "Error" output is set.

If an error occurs when saving a parameter, the function blocks continues with the next parameter until all are processed. The first error is displayed at the output.

Properties "plgnoreSourceAxis-Number" and "pUsedSourceAxis-Number"

The function block only imports files of the same axis instance. If a parameter file exported by another axis is used, no parameters are imported during an import as the axis instances are not identical. This is used to import the correct instance in case of a parameter file with many axis instances.

It is also recommended to ignore this axis instance to have only one parameter file with only one instance during series commissioning but to import this instance to many axes. In this case, the function block has one more property "plgnoreSourceAxisNumber". If this property is set to TRUE before using the function block, the function block ignores the axis instance and imports the parameters. The default setting of the property is FALSE, i.e. only the relevant axis instance is imported.



When using a parameter file with many instances and if the property is set to TRUE, the function block imports all instances into the same axis. Subsequently, this axis is then assigned all parameters of the last data set in the parameter file.

In the property "pUsedSourceAxisNumber", the address found and used in the parameter file by the function block can be retrieved after executing the function block.

Error codes

The function block uses the error table **F_RELATED_TABLE**. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR	16#0000000D	16#00000000	AxisRef is not in the valid range
INPUT_RANGE_ERROR	16#00001E01	16#00000000	Invalid "ParamType" input
RESOURCE_ERROR	16#00001E02	16#00000000	The parameter type selected in "ParamType" is not supported by this axis
RESOURCE_ERROR	16#00001E04	16#xxxxxxx	Error when writing a parameter. Additional2 provides incorrect parameter numbers.
ACCESS_ERROR	16#00001E11	16#00000000	Error when opening the backup file
ACCESS_ERROR	16#00001E13	16#00000000	Error when reading the backup file
ACCESS_ERROR	16#00001E14	16#00000000	Error when closing the backup file

ErrorID	Additional1	Additional2	Description
OTHER_ERROR	16#00001E15	16#xxxxxxx	Error when parsing the backup file
STATE_MACHINE_ERROR	16#00000006	16#0000FFx	Invalid state of the function block

Tab. 3-9: Error codes of the ML_AxisRestore function block

3.2.4 ML_AxisBackup_Type02

Brief description The ML_AxisBackup_Type02 function block backs up the parameters of an axis in a text file on the Compact Flash card of the control.

The function block is an improved version of the ML_AxisBackup. Depending on the parameter type, data backup is significantly faster ([ML_AxisBackup, page 33](#)).

The function block remained compatible to a great extent.

Assignment target system/library

Target system	Library
MLC/XLC 14VRS MOT function package	ML_TechMotionBase

Tab. 3-10: Reference table of the ML_AxisBackup_Type02

Interface description

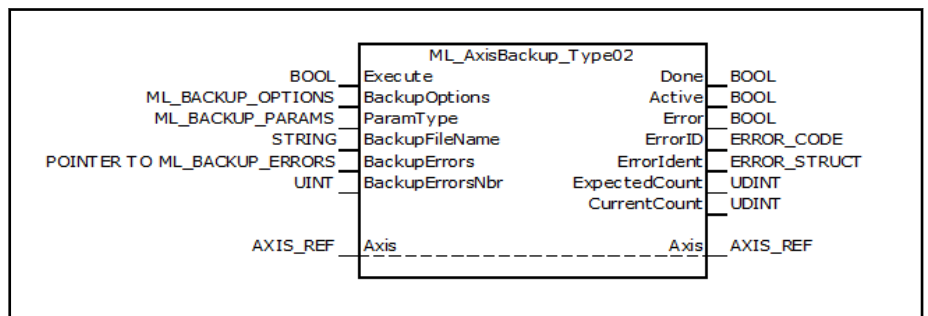


Fig. 3-3: ML_AxisBackup_Type02 function block

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Processing of the function block enabled (edge-controlled)
	BackupOptions	ML_BACKUP_OPTIONS	Scope of the parameter backup ML_BACKUP_OPTIONS, page 45
	ParamType	ML_BACKUP_PARAMS	Selection of the parameters to be backed up: A, S, P ML_BACKUP_PARAMS, page 46
	BackupFileName	STRING	File name on the Compact Flash card of the control
	BackupErrors	POINTER TO ML_BACKUP_ERRORS	Pointer to an array of ML_BACKUP_ERRORS structures ML_BACKUP_ERRORS, page 46
	BackupErrorsNbr	UINT	Number of arrays in the BackupErrors array

I/O type	Name	Data type	Description
VAR_OUTPUT	Done	BOOL	Processing completed, output data valid
	Active	BOOL	Processing running, output data valid
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Error diagnostics (refer to Error codes, page 41)
	ExpectedCount	UDINT	Number of parameters expected for export
	CurrentCount	UDINT	Number of already exported parameters
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Reference to the axis

Tab. 3-11: Interface variables of the ML_AxisBackup_Type02 function block

Min./max. and default values of the inputs

The following value ranges and default values can be applied at the function block:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL	FALSE	TRUE	FALSE	Continuous
BackupOptions	ML_BACKUP_OPTIONS			TO_SAVE_PARAMS	Rising edge at "Execute"
ParamType	ML_BACKUP_PARAMS			ASP_PARAMS	Rising edge at "Execute"
BackupFile-Name	STRING			"	Rising edge at "Execute"
BackupErrors	POINTER TO ML_BACKUP_ERRORS	> 0			Continuous
BackupErrorsNbr	UINT	0	65535	0	Continuous

Tab. 3-12: Min./max. and default values of the ML_AxisBackup_Type02 function block

Functional description

The ML_AxisBackup_Type02 function block backs up the parameters of an axis in a text file on the Compact Flash card of the control. If the file already exists, it is overwritten. The file can be read again by the IL_AxisRestore_Type02 function blocks and by IndraWorks.

BackupOptions

The parameters to be backed up are selected via the "BackupOptions" input. The following selections are possible:

- ALL_PARAM: All parameters are backed up. This option is primarily required to check the configuration
- MODIFIED_PARAM: The parameters deviating from the default values are backed up
- TO_SAVE_PARAM: The parameters required for restoration are backed up. This is the presetting

ParamType

The parameter types to be backed up are selected via the "ParamType" input. The following options can be selected:

- ASP_PARAMS: Parameters in the drive and control-internal axis parameters
- A_PARAMS: Control-internal axis parameters
- SP_PARAMS: Parameters in the drive

BackupFileName

Specify the file name on the Compact Flash card of the control via "BackupFileName". Always specify the complete path. To be hardware-independent, use the function "ML_GetSystemSpecialPath" from the "ML_Base" library to determine the desired path. Subordinate paths can be specified, but ensure that they exist.

Program:

```
// 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'
```

Time response

The function block ML_AxisBackup_Type02 works asynchronously. Due to a rising edge at the "Execute" input after the start, it reads the inputs "BackupOptions", "ParamType" and "BackupFileName" once and the inputs "BackupErrors" and "BackupErrorsNbr" continuously. "Active" is set as long as the function block is operating. After the successful operation, the "Done" output is set. If an error occurs, the "Error" output is set.

Do not use the function block in the Motion task, as it sometimes requires longer execution times.

Error codes

The function block uses the error table **F_RELATED_TABLE**.

It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR	16#0000000D	16#00000000	AxisRef is not in the valid range
INPUT_RANGE_ERROR	16#00001E00	16#00000000	Invalid "BackupOptions" input
INPUT_RANGE_ERROR	16#00001E01	16#00000000	Invalid "ParamType" input
RESOURCE_ERROR	16#00001E02	16#00000000	The parameter type selected in "ParamType" is not supported by this axis
RESOURCE_ERROR	16#00001E03	16#xxxxxxx	Error when reading a parameter
ACCESS_ERROR	16#00001E05	16#00000000	"BackupErrors" is 0 or "BackupErrorsNbr" is 0
ACCESS_ERROR	16#00001E10	16#00000000	Error while creating the parameter file
STATE_MACHINE_ERROR	16#00000006	16#0000FFx	Invalid state of the function block

Tab. 3-13: Error codes of the ML_AxisBackup_Type02 function block

3.2.5 ML_AxisRestore_Type02

Brief description The ML_AxisRestore_Type02 function block loads the parameters from a text file on the Compact Flash card of the control to an axis.

Assignment: Target system/library

Target system	Library
MLC/XLC 14VRS MOT function package	ML_TechMotionBase

Tab. 3-14: Reference table of the ML_AxisRestore_Type02

Interface description

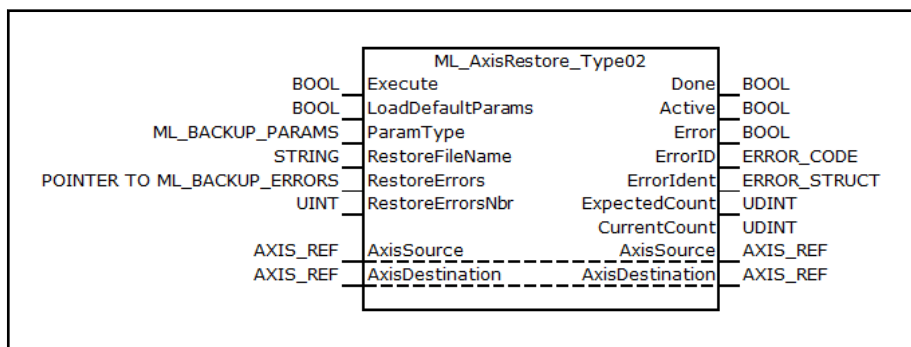


Fig. 3-4: ML_AxisRestore_Type02 function block

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Processing of the function block enabled (edge-controlled)
	LoadDefaultParams	BOOL	Load default parameters before loading the parameters from the file
	ParamType	ML_BACKUP_PARAMS	Selection of the parameters to be restored: All A, S and P ML_BACKUP_PARAMS, page 46
	RestoreFileName	STRING	File name on the Compact Flash card of the control
	BackupErrors	POINTER TO ML_BACKUP_ERRORS	Pointer to an array of ML_BACKUP_ERRORS structures ML_BACKUP_ERRORS, page 46
	BackupErrorsNbr	UINT	Number of arrays in the BackupErrors array
VAR_OUTPUT	Done	BOOL	Processing completed, output data valid
	Active	BOOL	Processing running, output data valid
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Error diagnostics (refer to Error codes, page 45)
VAR_IN_OUT CONSTANT	AxisSource	AXIS_REF	Reference to the axis from which the parameters are to be imported
	AxisDestination	AXIS_REF	Reference to the axis to which the parameters are to be imported

Tab. 3-15: Interface variables of the ML_AxisRestore_Type02 function block

Min./max. and default values of the inputs

The following value ranges and default values can be applied at the function block:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL	FALSE	TRUE	FALSE	Continuous
LoadDefaultParams	BOOL	FALSE	TRUE	FALSE	Rising edge at "Execute"
ParamType	ML_BACKUP_PARAMS			ASP_PARAMS	Rising edge at "Execute"
RestoreFilename	STRING			"	Rising edge at "Execute"
BackupErrors	POINTER TO ML_BACKUP_ERRORS	> 0			Continuous
BackupErrorsNbr	UINT	0	65535	0	Continuous

Tab. 3-16: Min./max. and default values of the ML_AxisRestore_Type02 function block

Functional Description

The ML_AxisRestore_Type02 function block loads the parameters from a text file on the Compact Flash card of the control to an axis. After loading the parameter file successfully, the configuration is saved on the drive or the axis. It is thus also available after deactivation. The parameter file can be created by the function block [ML_AxisBackup_Type02, page 39](#), or by IndraWorks.

The function block does **not** back up the following parameters, even if they are in the parameter file:

- All parameters that are read-only
- All parameters that are commands
- All currently write-protected parameters (e.g. protected by password, write-protected as PLC in RUN, etc.)
- More parameters such as A-0-0005 and A-0-0016



As it cannot be written on all parameters in the Sercos phase 4 and loading the default values is also not possible, only execute the function block in Sercos phase 2 or in the parameterization mode (PM).

LoadDefaultParams

If the "LoadDefaultParams" input is set, the default parameters of the drive or the axis are loaded before loading the parameters. This is especially reasonable if the parameter file was created using the options "MODIFIED_PARAM" or "TO_SAVE_PARAM".



Loading the default parameters takes several seconds per axis.



When loading the default parameters for an IndraDrive, the parameter P-0-4090 is written with the value 16#51 before triggering the command. Thus, default values are not assigned to the parameters of the master communication and the Engineering interface. If further and/or other parameter groups are not initialized, load the default parameters personally.

ML_TechMotionBase.library // MX_TechMotionBase.library

ParamType

The parameter types to be restored, are selected via the "ParamType" input. The following options can be selected:

- ASP_PARAMS: Parameters in the drive and control-internal axis parameters
- A_PARAMS: Control-internal axis parameters
- SP_PARAMS: Parameters in the drive

RestoreFilename

Specify the file name on the Compact Flash card of the control via "RestoreFilename". Always specify the complete path. To be hardware-independent, use the function ML_GetSystemSpecialPath from the "ML_Base" library to determine the desired path.

Subordinate paths can be specified, but ensure that they exist.

Program:

```
// 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"

The ML_AxisRestore function block used as basis for this function block, is only provided with one "Axis" input. Only the parameters of this axis are always implicitly imported. This is sufficient for parameter files containing only data of one axis. This dependency is avoided for the function block ML_AxisRestore_Type02. Thus, it is possible to import the parameters of one axis into another axis or to find the correct parameters in a parameter file with multiple data records. This is reasonable for example if multiple devices are in one parameter file and only one specific axis is to be imported. Another use case is to create only one parameter file with one axis, but to import it on multiple axes. "AxisSource" and "AxisDestination" define data source and data target. "AxisSource" specifies the logical axis address from where the parameters of the parameter file are to be used. "AxisDestination" is the axis in the system to the parameters to. For a behavior analog to the behavior of the function block ML_AxisRestore, set the same value for "AxisSource" and "AxisDestination".

Properties "plgnoreSourceAxisNumber" and "pUsedSourceAxisNumber"

In some cases, "AxisSource" cannot be specified. This is for example the case if a parameter file was created by one axis and if the logical address of this axis is not known anymore. This information can also be found in the parameter file. In this case, the property "plgnoreSourceAxisNumber" can be written with "True" before executing the function block. The function block does not use the "AxisSource" input internally anymore, but searches for the

logical address in the parameter file instead. This means that "AxisSource" does not have to be known. In the property "pUsedSourceAxisNumber", the address found and used in the parameter file by the function block can be retrieved after executing the function block.

This is only successful if only data of one axis is in the parameter file (in case of multiple data records, the latest data record is used). The default state for the property "plgnoreSourceAxisNumber" is "FALSE". Thus, specifying "AxisSource" is required.

Time response

The function block ML_AxisRestore_Type02 works asynchronously. Due to a rising edge at the "Execute" input after the start, it reads the inputs "LoadDefaultParams", "ParamType" and "RestoreFilename". "Active" is set as long as the function block is operating. After the successful operation, the "Done" output is set. If an error occurs, the "Error" output is set.

If an error occurs when saving a parameter, the function blocks continues with the next parameter until all are processed. All errors that occurred are shown at the output below the "BackupErrors" pointer.

Error codes

The function block uses the error table **F_RELATED_TABLE**. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR	16#0000000D	16#00000000	AxisRef is not in the valid range
INPUT_RANGE_ERROR	16#00001E01	16#00000000	Invalid "ParamType" input
RESOURCE_ERROR	16#00001E02	16#00000000	The parameter type selected in "ParamType" is not supported by this axis
ACCESS_ERROR	16#00001E05	16#00000000	"BackupErrors" is 0 or "BackupErrorsNbr" is 0
OTHER_ERROR	16#00001E16	16#xxxxxxx	The subordinate firmware functionality returns an error. The error code of the firmware function is provided in Additional2. If Additional2 = 16#F00A0008, an import error occurred. For the individual errors and parameters affected, refer to the created error structure ""BackupErrors"".
STATE_MACHINE_ERROR	16#00000006	16#0000FFx	Invalid state of the function block

Tab. 3-17: Error codes of the ML_AxisRestore_Type02 function block

3.2.6 ML_BACKUP_OPTIONS

Type definitions

The following definition describes the enumeration type ML_BACKUP_OPTIONS used in the following function blocks.

- [ML_AxisBackup, page 33](#)
- [ML_AxisBackup_Type02, page 39](#)

The type definitions are included in the library.

The data type "ML_BACKUP_OPTIONS" is defined as follows:

ML_TechMotionBase.library // MX_TechMotionBase.library

Enumeration element	Value	Description
ALL_PARAM	1	All parameters
MODIFIED_PARAM	2	Parameters deviating from the default value
TO_SAVE_PARAM	3	Parameters required for restoration

Tab. 3-18: Defining the enumeration elements of the ML_BACKUP_OPTIONS data type

3.2.7 ML_BACKUP_PARAMS

Type definitions The following definition describes the enumeration type ML_BACKUP_PARAMS used in the following function blocks.

- [ML_AxisBackup](#), page 33
- [ML_AxisRestore](#), page 36
- [ML_AxisBackup_Type02](#), page 39
- [ML_AxisRestore_Type02](#), page 42

The type definitions are included in the library.

The data type "ML_BACKUP_PARAMS" is defined as follows:

Enumeration element	Value	Description
ASP_PARAMS	3	Parameters in the drive and control-internal axis parameters
A_PARAMS	2	Control-internal axis parameters
SP_PARAMS	1	Parameters in the drive

Tab. 3-19: Defining the enumeration elements of the ML_BACKUP_PARAMS data type

3.2.8 ML_BACKUP_ERRORS

Type definitions The following definition describes the data type ML_BACKUP_ERRORS used in the following function blocks.

- [ML_AxisBackup_Type02](#), page 39
- [ML_AxisRestore_Type02](#), page 42

The type definitions are included in the library.

The data type "ML_BACKUP_ERRORS" is defined as follows:

Structural element	Data type	Description
ParameterNumber	MB_IDN	Parameter number in IDN format
ErrorCode	DWORD	Error code

Tab. 3-20: Defining the ML_BACKUP_ERRORS data type

3.3 Measurement encoder support

3.3.1 Introduction and overview

The function block [MB_DistanceAccumulator](#), page 47, sums up **any** relatively traveled distance via an encoder.

The function block [MB_AxisDistanceAccumulator](#), page 48, sums up the relatively traveled distance **of one axis** via an encoder.

3.3.2 MB_DistanceAccumulator

Brief description The MB_DistanceAccumulator function block sums up **any** relatively traveled distance via an encoder.

With the rising edge of the **Enable** input, the distance traveled is summed up by the function block until the summed up distance is higher than or equal to the **TotalDistance** input value. The **DistanceReached** output is set to TRUE and the function block processing is completed.

The current encoder position is specified using the **CurrentPosition** input.



In version 14VRS, the "MB_DistanceAccumulator (ML_TechMotion library)" function block was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC 14VRS	ML_TechMotionBase
MOT function package	
MLD (IL2G)	MX_TechMotionBase

Tab. 3-21: Reference table of the MB_DistanceAccumulator

Interface description

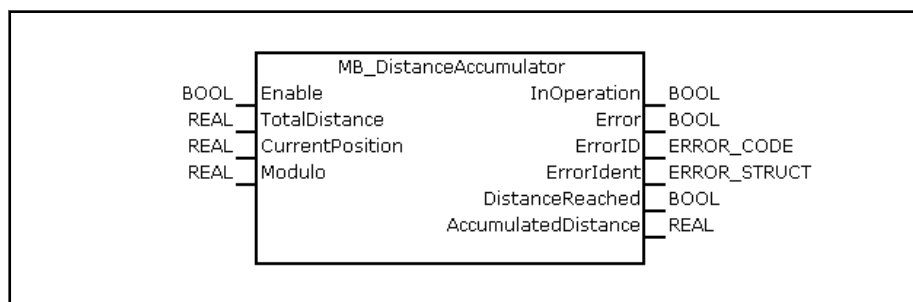


Fig. 3-5: MB_DistanceAccumulator function block

I/O type	Name	Data type	Comment
VAR_INPUT	Enable	BOOL	Enables the function block with a rising edge
	TotalDistance	REAL	Total distance to be traveled before the process is completed
	CurrentPosition	REAL	Current encoder position
	Modulo	REAL	Modulo value: (360: rotatory 0: linear) In case of a Servo axis, this value can be read from A-0-0045 and/or S-0-0103

ML_TechMotionBase.library // MX_TechMotionBase.library

I/O type	Name	Data type	Comment
VAR_OUTPUT	InOperation	BOOL	The function block is currently active
	Error	BOOL	Processing completed with errors
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 48
	DistanceReached	BOOL	TotalDistance was reached and the processing of the function block is completed.
	AccumulatedDistance	REAL	Indicates the currently traveled distance

Tab. 3-22: Interface variables of the MB_DistanceAccumulator function block

Troubleshooting The function block generates the following error messages in Additional1/Additional2 for the **F_RELATED_TABLE**.

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR	16#0003	16#0000	The function block aborted by another function block
STATE_MACHINE_ERROR	16#0006	16#0000	Invalid status of the state machine

Tab. 3-23: Error codes of the MB_DistanceAccumulator function block

3.3.3 MB_AxisDistanceAccumulator

The MB_AxisDistanceAccumulator function block sums up the relatively traveled distance of one axis via an encoder.

With the rising edge of the **Enable** input, the distance traveled is summed up by the encoder until the summed up distance is higher than or equal to the **TotalDistance** input value. As a result, the **DistanceReached** output is set to TRUE and the function block stops. This function block uses the axis reference input (AXIS_REF) to query the current encoder position automatically.



In version 14VRS, the "MB_AxisDistanceAccumulator (ML_TechMotion library)" function block was moved to the "TechMotion-Base library".

Assignment target system/library

Target system	Library
MLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-24: Reference table of the MB_AxisDistanceAccumulator

Interface description

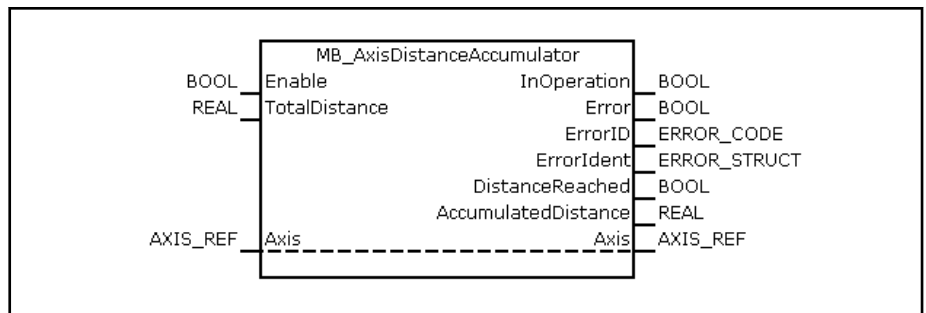


Fig. 3-6: MB_AxisDistanceAccumulator function block

I/O type	Name	Data type	Comment
VAR_INPUT	Enable	BOOL	Enables the function block with a rising edge
	TotalDistance	REAL	Total distance to be traveled before the process is completed
VAR_OUTPUT	InOperation	BOOL	The function block is currently active
	Error	BOOL	Processing completed with errors
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 49
	DistanceReached	BOOL	TotalDistance was reached and the processing of the function block is completed
	AccumulatedDistance	REAL	Indicates the currently traveled distance
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Axis reference

Tab. 3-25: Interface variables of the MB_AxisDistanceAccumulator function block

Troubleshooting

The function block generates the following error messages in Additional1/ Additional2 for the "F_RELATED_TABLE".

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR	16#0003	16#0000	The function block aborted by another function block
STATE_MACHINE_ERROR	16#0006	16#0000	Invalid status of the state machine

Tab. 3-26: Error codes of the MB_AxisDistanceAccumulator function block

3.4 Programmable limit switches, application

3.4.1 Introduction and overview

The function block for the application "Programmable limit switch" is a PLCopen-compliant function block ([MC_DigitalCamSwitch, page 50](#)).

Using this function block, up to 8 outputs with up to 32 switches can be freely programmed.

The function block [MB_DigitalCamSwitchOvs, page 66](#) implements a freely programmable limit switch with up to 64 switches on 32 switch tracks. In contrast to the MC_DigitalCamSwitch function block, it supports oversampling.

Thus, a corresponding higher temporal resolution of the PLS signals can be achieved. Use the function blocks with an I/O module that supports oversampling, e.g. the Sercos block I/O module R-ILB S3 24 DI16 DIO16/F.

Data structures

- [MC_OUTPUT_REF](#), page 66
- [MC_CAMSWITCH_REF](#), page 65
- [ML_BASIC_CAMSWITCH_REF](#), page 65
- [MB_CAMSWITCH_REF_OVS](#), page 92
- [ML_BASIC_CAMSWITCH_REF_OVS](#), page 93
- [MC_TRACK_REF](#), page 65
- [ML_BASIC_TRACK_REF](#), page 66
- [MB_TRACK_REF_OVS](#), page 93
- [ML_BASIC_TRACK_REF_OVS](#), page 93

3.4.2 MC_DigitalCamSwitch

Brief description The MC_DigitalCamSwitch function block implements freely programmable limit switches with up to 32 PLS on 8 outputs.



In version 14VRS, the "MC_DigitalCamSwitch (ML_TechBase library)" function block was moved to the "TechMotionBase library".

Assignment: Target system/library

Target system	Library
MLC/XLC 14VRS	ML_TechMotionBase
MOT function package	
MLD (IL2G)	MX_TechMotionBase

Tab. 3-27: Reference table of the MC_DigitalCamSwitch function block

Interface description

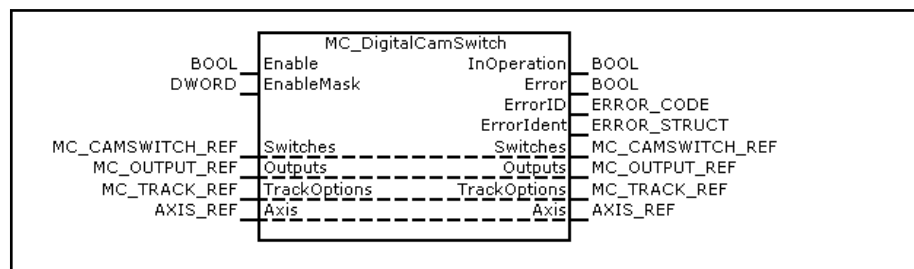


Fig. 3-7: MC_DigitalCamSwitch function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enables the switching outputs
	EnableMask	DWORD	32 bits of type BOOL. Enables the different switch tracks. The lowest-order bit is assigned to the lowest "TrackNumber". If the value is set to 1 or TRUE, the assigned "TrackNumber" is enabled. Since only the eight lowest-order bits are available, the number is restricted to eight switch tracks per function block. If the bit is not set, the corresponding output is set to 0

I/O type	Name	Data type	Description
VAR_OUTPUT	InOperation	BOOL	The enabled switch tracks are calculated and the outputs are updated
	Error	BOOL	Indicates that an error occurred in the function block
	ErrorID	ERROR_CODE	Error ID
	ErrorIdent	ERROR_STRUCT	Error structure with further error subdivision. Error codes, page 64
VAR_IN_OUT	Switches	MC_CAMSWITCH_REF	Reference to the switch structure containing all definitions for all switches. MC_CAMSWITCH_REF, page 65
	Outputs	MC_OUTPUT_REF	Reference to the signal outputs directly related to the corresponding switch tracks (max. eight per function block, first output = first "Track Number"). MC_OUTPUT_REF, page 66
	TrackOptions	MC_TRACK_REF	Reference to the structure containing the corresponding switch track characteristics, e.g. the on/off lead time per output/switch track. MC_TRACK_REF, page 65
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Reference to the axis to which the programmable limit switch is assigned

Tab. 3-28: Interface variables of the MC_DigitalCamSwitch function block

Functionality

The following function block MC_DigitalCamSwitch provides a PLCopen-specific (part 2) interface for a programmable limit switch. The function block calculates eight binary switch track outputs according to the specification below.

Every binary output is assigned to a switch track. According to the following specification, a switch track can contain several on/off positions. Up to 32 switches can be freely distributed to the eight switch tracks.

Terminology

Name	Description
MLC/XLC	BRC rack-based MotionLogic control
Digital cam switch	Programmable limit switch (PLS)

Tab. 3-29: Terminology of the programmable limit switch

Settings

The units for position and velocity have to be set for the axis assigned to the programmable limit switch. If the axis positions are defined in degrees for example, the programmable limit switch positions are also interpreted in degrees.

Data types

The function block "MC_DigitalCamSwitch" uses the following data types:

A switch track is an element describing the characteristics of the programmable limit switch related to an output.

- [MC_TRACK_REF, page 65](#), contains an array with eight basic structures (that means that eight switch tracks are assigned to eight outputs). Several switches can be assigned to each switch track
- [ML_BASIC_TRACK_REF, page 66](#)
- [MC_CAMSWITCH_REF, page 65](#), contains an array of 32 basic structures each describing a single switch

ML_TechMotionBase.library // MX_TechMotionBase.library

- [ML_BASIC_CAMSWITCH_REF](#), page 65
- [MC_OUTPUT_REF](#), page 66, contains an array with eight Boolean variables assigned to the outputs of the programmable limit switch

Example program

A short example explains how the function block is programmed. The different structure elements are explained later in detail.

It is assumed that a single switch is to be created:

- With number 4 on switch track 3
- with a "FirstOnPosition" of 35 and
- A "LastOnPosition" of 55
- The switch should trigger 10 ms earlier and the switch track has a hysteresis of 3
- The switch should switch on/off with regard to the position and not to the time and it should only be applied in positive direction

The variable declaration and the function block call are as follows:

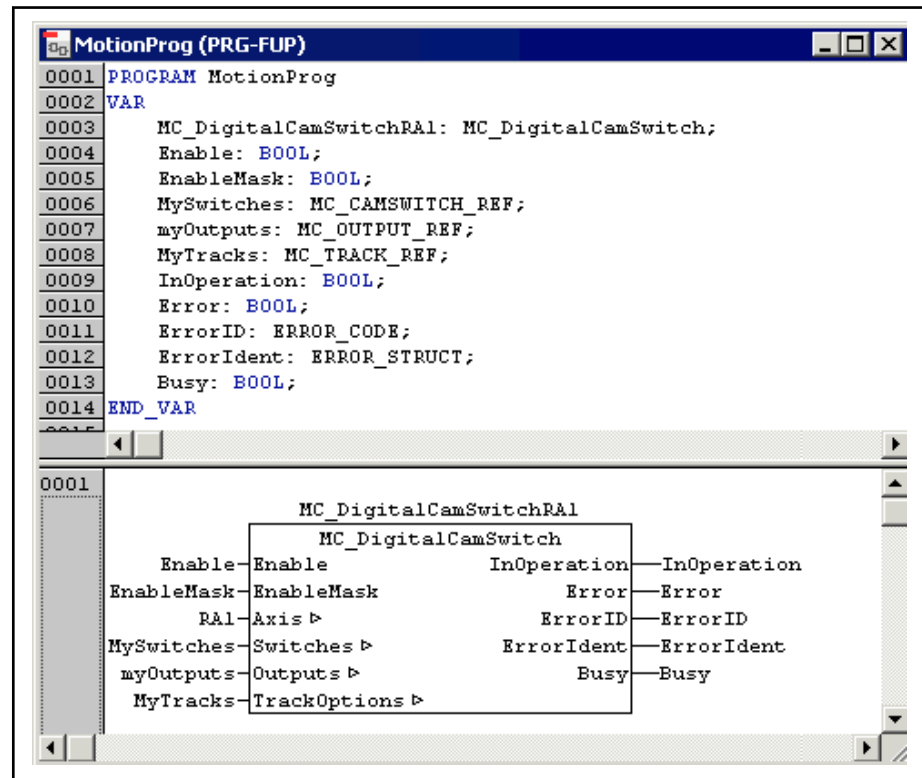


Fig. 3-8: Variable declaration and function block call

Refer to the three input structures "MySwitches", "MyOutputs" and "MyTracks" which define all switch characteristics, switch tracks and outputs. To understand these structures better, have a closer look at the data content of the structures provided for the results of the example above:

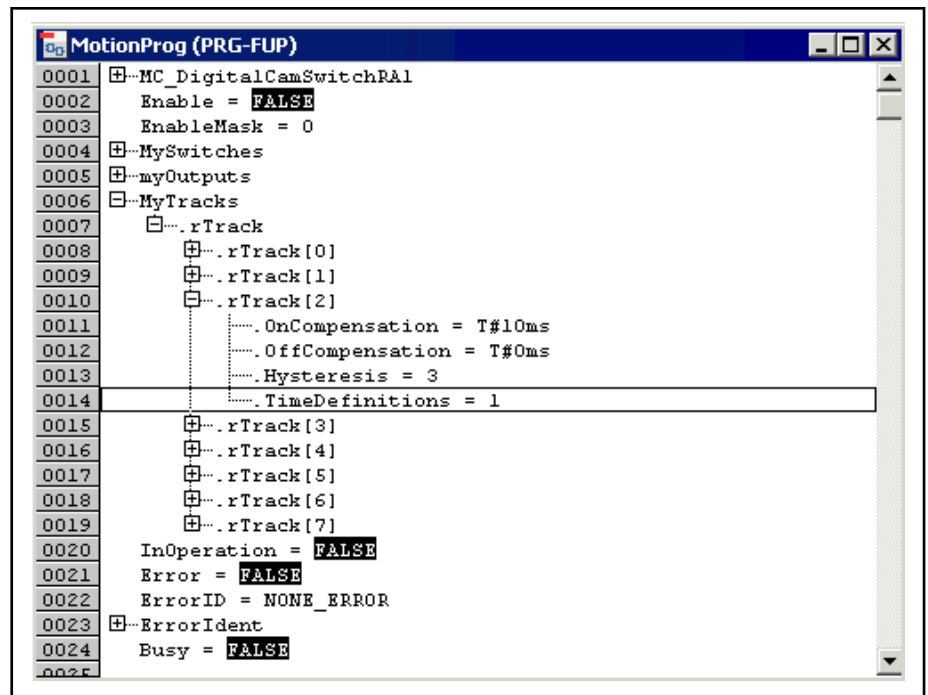


Fig. 3-9: Content of the "MyTracks" structure

First, the switch track is defined (figure above).

1. To use switch track 3, initialize `rTrack[2]`, as the array starts with 0.
2. The OnCompensation time is set to 10 ms. To ensure that the switch switches earlier and not later, set bit 0 in "TimeDefinitions" defining the "OnCompensation" time as negative.
3. Finally, set the hysteresis to 3.

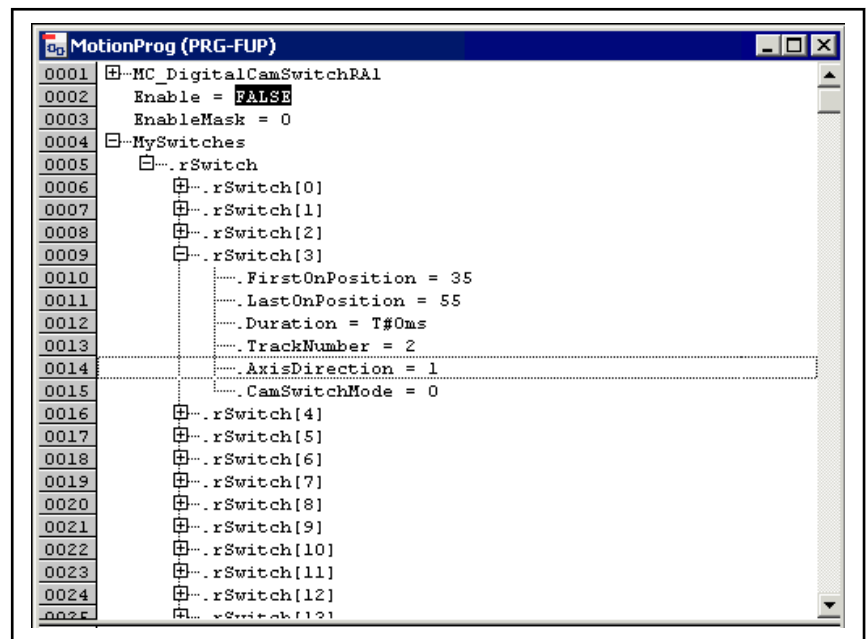


Fig. 3-10: Content of the "MySwitches" structure

The switch is defined (figure above).

ML_TechMotionBase.library // MX_TechMotionBase.library

1. Initialize the switch structure with index 3 to initialize switch 4, since the array starts with the index 0.
2. "FirstOnPosition" is set to 35, "LastOnPosition" is set to 55 and the "AxisDirection" is set to 1. This means that the switch is only applied in positive direction.
3. To assign the switch to a switch track, set the "TrackNumber" to 2. This specifies that this switch is assigned to the switch track with index 2 in the switch track structure.
4. Finally, the "CamSwitchMode" has to be set to 0 to switch on the position-related mode.



The individual elements of the REF structures can be accessed according to the following pattern:

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

Preparing the function block start:

To start the function block, add the remaining variables to the function block and set their values accordingly. The "Enable" input has to be set and the "EnableMask" has to set the corresponding bit:

2#00000100, bit 2 for third switch track rTrack[2].

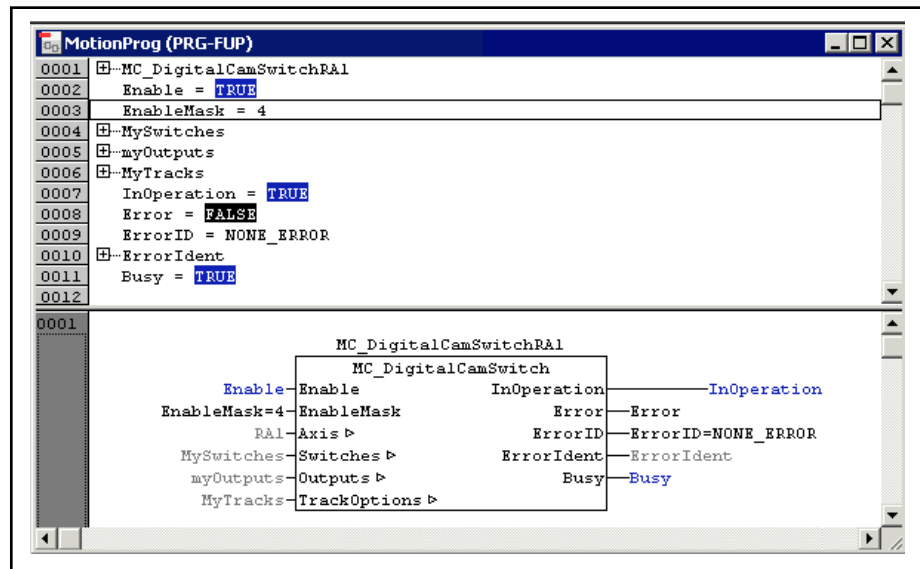


Fig. 3-11: Running function block

Functional description

The **Axis** input defines the position and velocity for the calculations. If the "Axis" input changes while the "Enable" input is not yet set, an error is generated.

When the **Enable** input is set for the first time, the "Busy" output is also set, but the "InOperation" output remains inactive. Reason: Some internal values have to be updated before the calculation continues. Several PLC cycles are required. After this initialization sequence, the "InOperation" output becomes active again.

With the **EnableMask** input, outputs can be quickly enabled or blocked. The lowest-order bit is assigned to the lowest "TrackNumber". If the value is set to 1 or TRUE, the corresponding "TrackNumber" is enabled. Due to the restric-

tion to the eight lowest-order bits, only eight switch tracks are available per function block. If the bit is not set, the respective output is set to 0 irrespective of whether the current position is located in the area of a switch. If "Enable-Mask" is 0, "InOperation" becomes inactive, since none of the outputs is enabled.

Each switch has a starting and an end position. This allows to specify a **FirstOnPosition** and a **LastOnPosition** (or time) for each switch. In addition to a mechanical switch, the switch can be enabled and a lead time and a hysteresis can be assigned to the switch for a certain period. If ("FirstOnPosition" > "LastOnPosition"), the switch acts inversely and is only switched off within these positions.

Area/modulo calculations

Most likely, the programmable limit switch is only used for rotary processes. However, the programmable limit switches can also be used for a linear axis which runs forward and backward. In both cases, the programmable limit switch units are based on the units defined for the axis.

AxisDirection indicates in which direction of motion the switch becomes active. If the "AxisDirection" is 0 (default), the switch becomes active in both directions. If the "AxisDirection" is 1, the output is only set if it is traversed in positive direction. If the "AxisDirection" is set to 2, the output is only set if it is traversed in negative direction.

CamSwitchMode switches between the position or time display. "CamSwitchMode" = 0 refers to the position mode. The output is enabled at "FirstOnPosition" and disabled at "LastOnPosition". If "CamSwitchMode" = 1, the output is enabled at "FirstOnPosition" and disabled after the specified "Duration".

Duration: Switched-on period of the time-controlled switch. If the position is still in the switch area after this time, the output is not set again until the position has switched off the switch.

The lead time (**OnCompensation** and **OffCompensation**) can be positive or negative. Negative means that the output switches before the switching position is reached. The lead time is indicated in ms (also refer to "TimeDef").

Hysteresis: This parameter prevents the output from switching too often if the axis is close to the switching point and the current position varies around the switching point. The hysteresis is part of "MC_TRACK_REF". That means that an individual hysteresis can be set for every switch track.

Both, in forward and in backward direction, the hysteresis has to be run through first before the switch switches.

The following figure shows two PLS, each defined from 150° to 180°.

The upper curve shows the switch output without hysteresis, the center curve shows the switch output with a hysteresis of 10°. The bottom curve represents the master position.

The master first moves in positive direction over the switches, afterwards in negative and finally in positive direction again. It can be seen that the switch with hysteresis is in each case switched on with a delay corresponding to the amount of the hysteresis, irrespective of the direction from which the switch is run through.

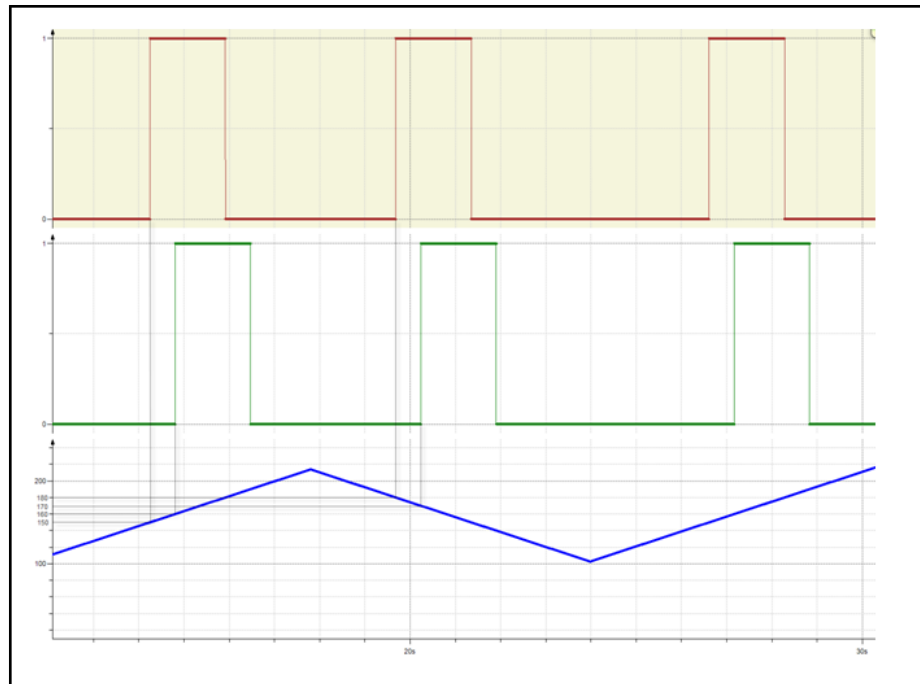


Fig. 3-12: Effect of the hysteresis on the switching points of a switch

TimeDef: This parameter was added to differ between positive and negative times for the "On/OffCompensation". If bit 0 is set, the "OnCompensation" time is negative. If bit 1 is set, the "OffCompensation" time is negative.

Example:

"OffCompensation" is set to 300 ms, "TimeDef" is set to 2.

This means that the lead time is "-300ms".

The following figure shows the different variables and structures and their respective use:

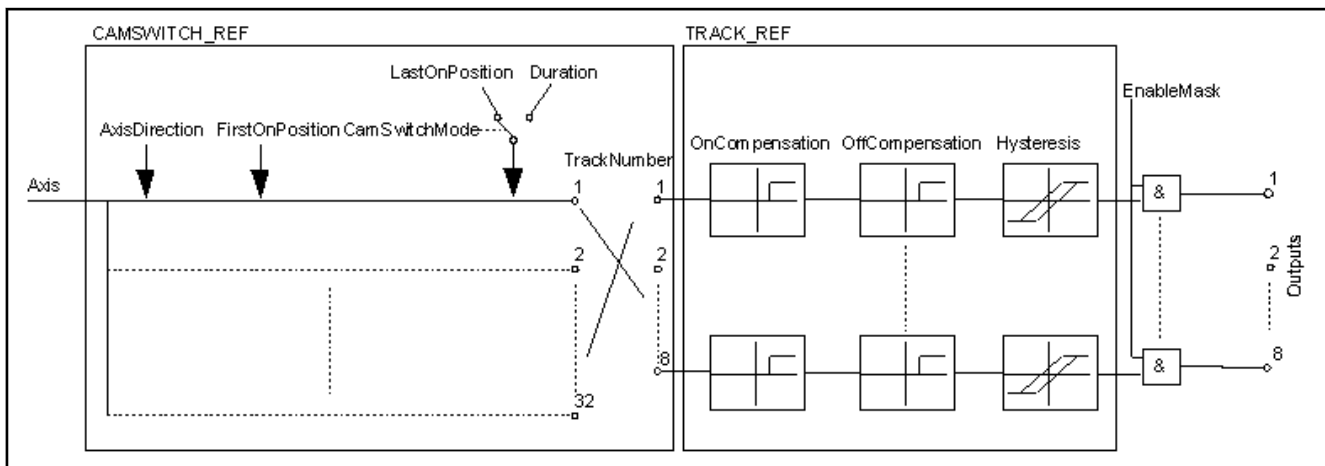


Fig. 3-13: Relation of the structures and the corresponding functionality of the MC_DigitalCamSwitch function block

The following figure shows the mechanical equivalent of the digital programmable limit switch.

Note that not all functions were shown, since it is either not possible or very difficult to show them.

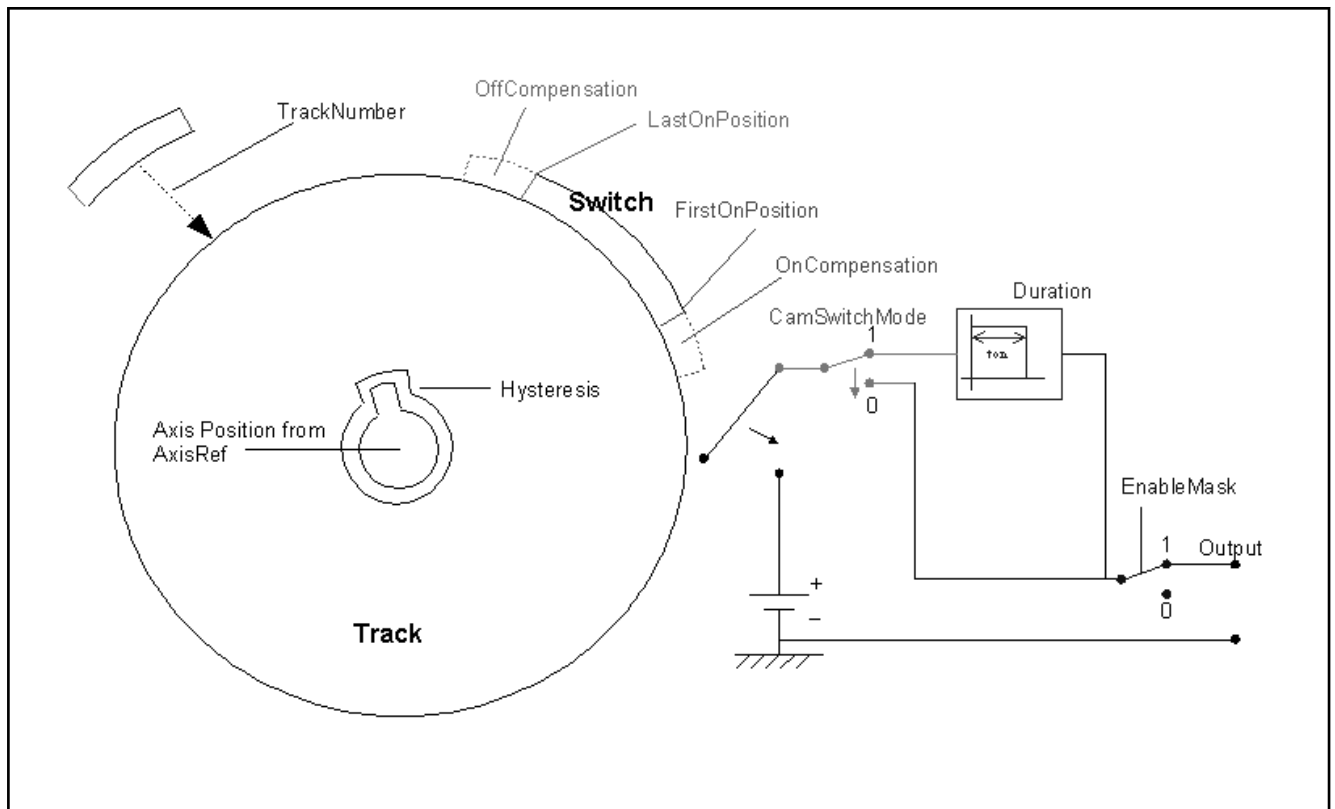


Fig. 3-14: Mechanical equivalent of the digital programmable limit switch

Time diagram

The following example shows the use of the "MC_CAMSWITCH_REF" structure and of the time-based switches. Find a specified value configuration in the following list:

Parameters	Type	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=Both	0=Both		
CamSwitchMode	INT	0=Position	0=Position	0=Position	1=Time		

Tab. 3-30: Example of the MC_CAMSWITCH_REF structure



Values are given in units [u]. These relate to the corresponding axis.

If e.g. a switch-on position is at 20u and the corresponding axis is scaled in degrees, the switch-on position is 20°.

This example uses the values from the example above "MC_CAMSWITCH_REF". It neither uses "On/OffCompensation" nor "Hysteresis".

This is the behavior of the outputs when the axis moves continuously in positive direction. The axis is a modulo axis with a modulo length of 5,000 u (units).

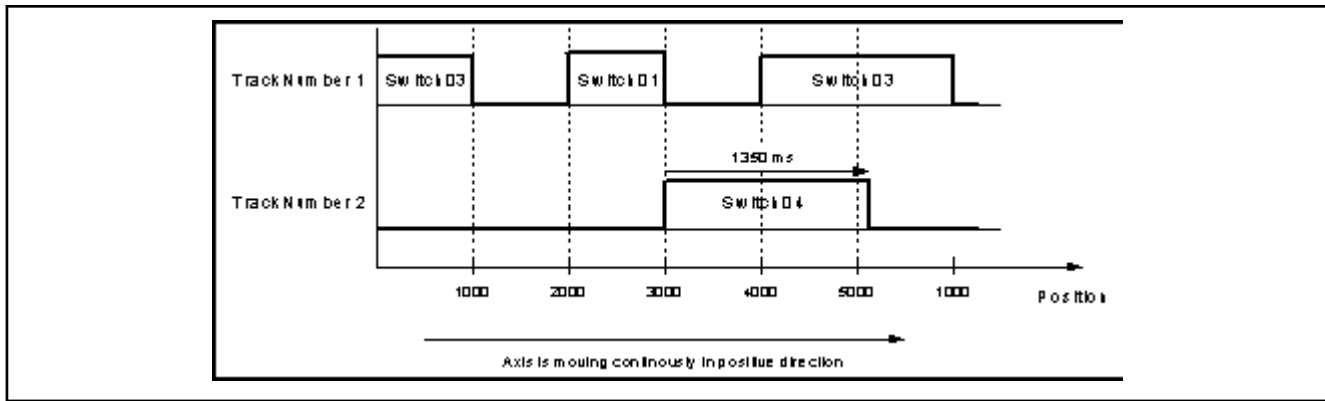


Fig. 3-15: Behavior of the outputs in the example

Detailed description of the switch "Switch01"

This example uses additionally "OnCompensation" = -125 ms and "OffCompensation" = +250 ms.

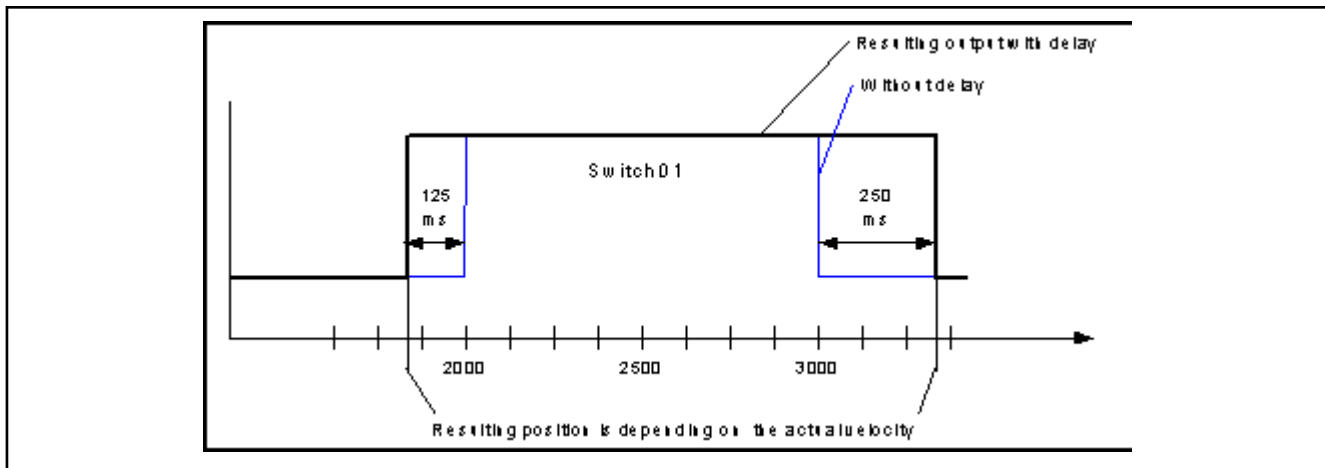


Fig. 3-16: Detailed behavior of Switch01 when moving in positive direction

This is the behavior of the outputs when the axis moves continuously in negative direction without "On" or "OffCompensation" and without "Hysteresis":

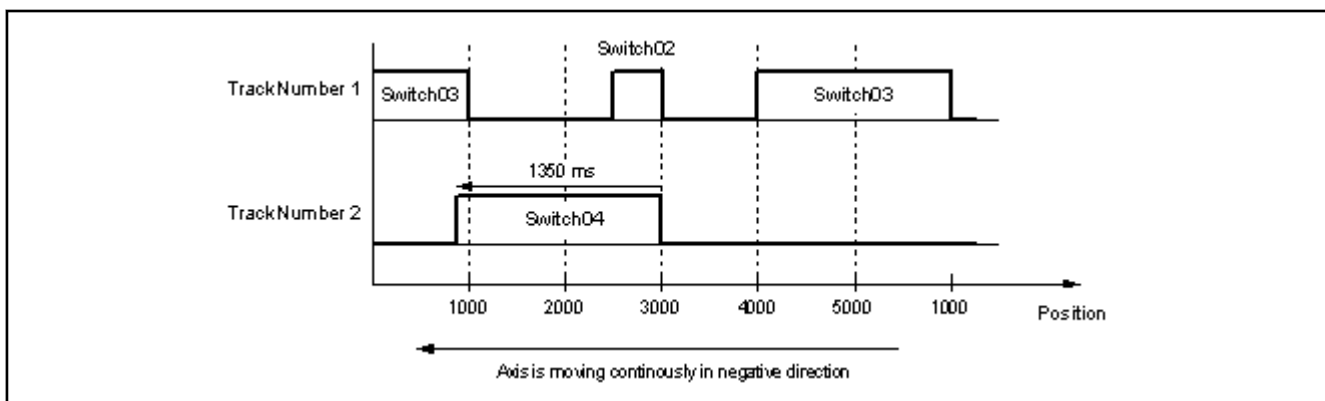


Fig. 3-17: Detailed behavior of the switches when moving in negative direction

Implementation details

The PLCopen extension specifies the interface of the PLS functionality. However, some application details are not specified. Therefore, these were defined for an implementation with a better runtime behavior.

OnCompensation/OffCompensation

PLCopen defines the "On/OffCompensation" only for motions in positive direction. In this case, the "OnCompensation" relates to the "FirstOnPosition" while the "OffCompensation" relates to the "LastOnPosition". How can this be defined for motions in negative direction?

Based on applications for which the lead times are used to compensate for the on and off delays with actuators, this means that the "OnCompensation" is responsible for the rising edge of the signal. In other words: If it is traveled in positive direction, the "OnCompensation" relates to the "FirstOnPosition" and if it is traveled in negative direction, the "OnCompensation" relates to the "LastOnPosition". The "OffCompensation" is defined accordingly

The following figure shows the behavior for "OnCompensation"= -0.1s and "OffCompensation"= +0.2s:

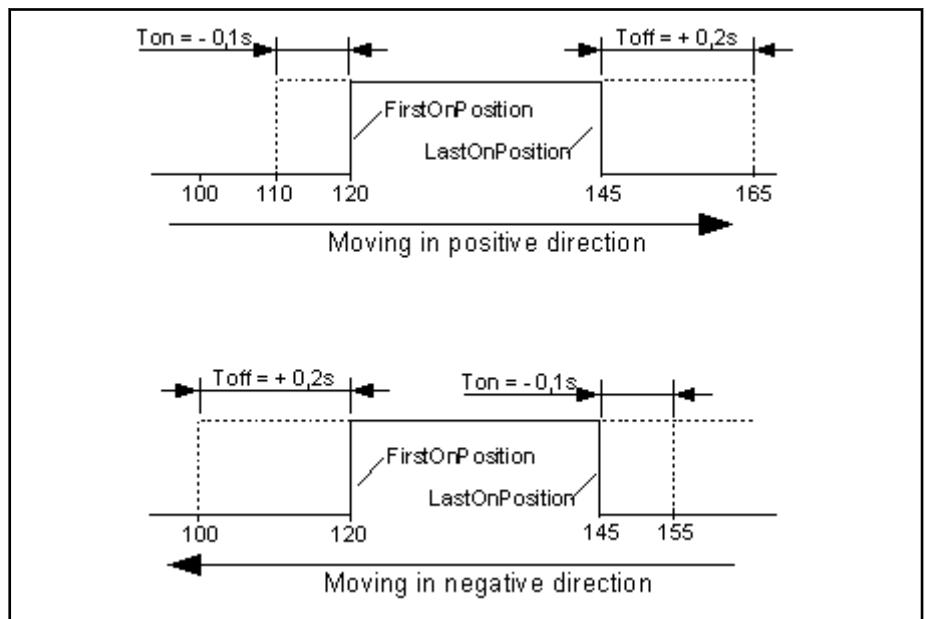


Fig. 3-18: Defining the OnCompensation and OffCompensation time

Duration

The "Duration" is defined as the time the switch remains switched on after it has been switched on. The question is what happens if the period has expired but the current position is still in the "FirstOnPosition" and "LastOnPosition". Does the output has to be switched on immediately or do you have to wait until the current position left the "FirstOnPosition" and "LastOnPosition"?

It has been decided to wait for a rising edge of the switch output. In other words: The current position has to be beyond the "FirstOnPosition" and "LastOnPosition" before the "Duration" can be started again.

The same applies to the "OnCompensation"/"OffCompensation". The "Duration" begins at the rising edge of a switch that is at the "LastOnPosition" when moving in negative direction.

The situation is explained in the following figure:

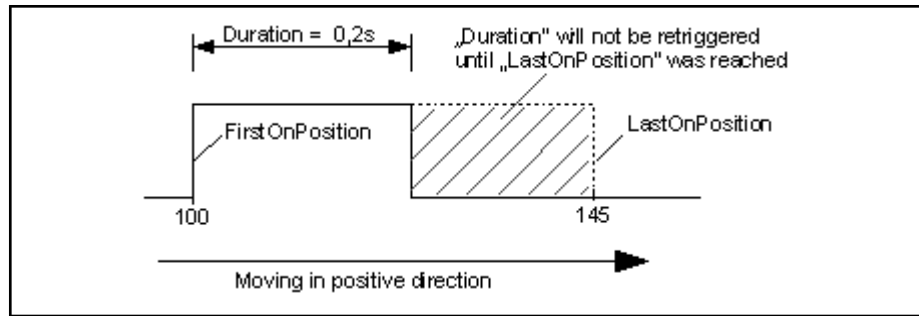


Fig. 3-19: Defining the duration

Combination of OnCompensation/OffCompensation and duration

Should these be considered when a time-based switch with "On/OffCompensation" times is used? The "OnCompensation" is considered. This leads to an earlier/later activation of the output. The "OffCompensation" time is more difficult, since the output is disabled after the lead time ("Duration"). The "OffCompensation" time is only relevant if it has to be determined that the output has to be retriggered (hatched area in the figure above).

In this application, the "OnCompensation" and the "OffCompensation" were considered.

How a switch is not used

The PLS has up to 32 switches. The element "TrackNumber" in the MC_CAMSWITCH_REF structure determines whether a switch is used.

This variable contains the switch track number. Valid values are between 0 and 7, since there is a maximum of 8 switch tracks. The default setting of this variable is 8, it is marked as not used. Every invalid number (8 to 65535) is interpreted as an unused switch and is therefore not evaluated by the firmware.

Limitations of the programmable limit switches (PLS)

The limitations of the programmable limit switch (PLS) are discussed in the following. The restrictions are only definitions which do not cause any error.

The system behavior is determined with these definitions. Therefore, the user has to expect different behaviors.

Lost switching actions

The sampling theorem indicates that the sampling rate is twice as high as the highest signal frequency required to detect the signal. This also applies to the PLS. The PLS samples the position data every time it is called by the PLC. Usually, this is the Motion-synchronous task or every other cyclic task. Every switch switched on for a time shorter than the sampling rate is not identified correctly. The following equations show how the velocity or the size of the switch, for which switching actions are lost, are calculated.

Rotary systems:

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

n is the revolution per second [1/s].

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

ρ is the angle in rad and t is the time in seconds [s].

Conversion factor from degree to rad: $\pi/180$

Use the equations 1 and 2 and the following results:

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

for the angle or

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

If the factors are adapted for common units, the following results:

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

or for the velocity:

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

The equations A1 and B1 provide the two equations required to determine whether switching actions can be lost. For example, if the maximum speed of $1,000 \text{ min}^{-1}$ is known and the PLS is called in a cyclic task with a cycle time of 2 ms. What is the minimum size of a switch to ensure that it can be identified by the PLS?

Result when using equation A1:

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

I.e., the minimum switch size has to be 12° to ensure that it can be identified at a speed of max. $1,000 \text{ min}^{-1}$.

Linear systems:

$$v = s / t$$

s is the distance in millimeters [mm] and t is the time in seconds [s].

If the PLC cycle time is used as the sampling time for the PLS and if you add the corresponding factors to adjust the units, the following equations result:

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

or for the velocity:

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

ML_TechMotionBase.library // MX_TechMotionBase.library

These equations also apply to systems with inch [in] as distance unit:

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

or for the velocity:

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

Example: A switch in a linear system has a length of 0.02. The PLS is located in a PLC task with a cycle time of 2 ms. Which maximum velocity is possible to ensure that the switch is identified?

Result when using equation B2:

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

Switches seem to have the wrong size

The boundary conditions for "lost switching actions" were described above.

Even if the conditions for lost switches are not fulfilled, the switching times might not be exact. The PLS can only change the state of its outputs if it is called by the PLC. If a switch should be closed at a specified velocity for 3.7 ms for example, but the PLS is only called every 2 ms, the output is set to "On" either for 2 ms (case B) or for 4 ms (case A) depending on the sampling time of the PLS. A more detailed explanation can be found in the following figure

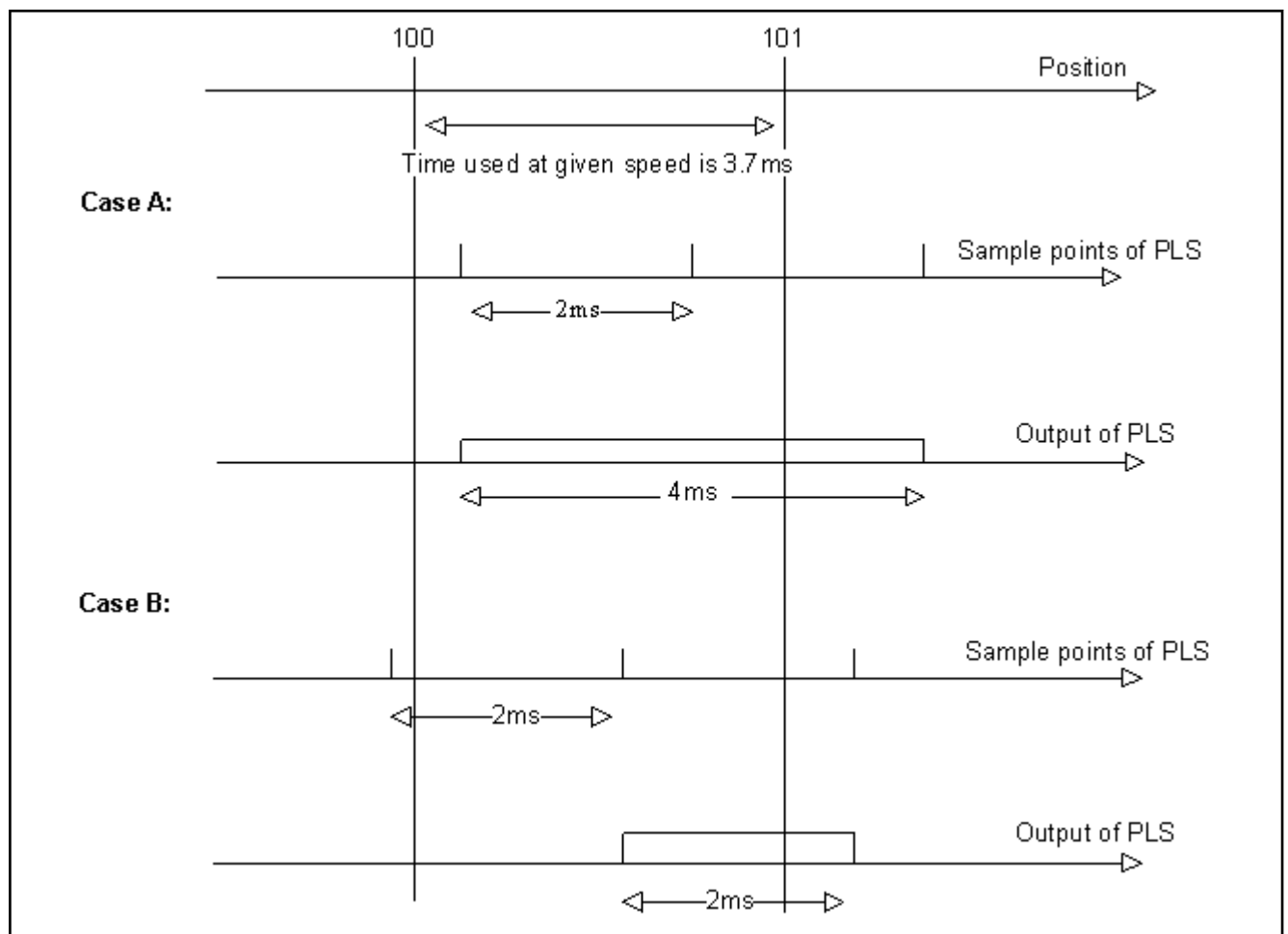


Fig. 3-20: Inexact switch sizes

Switching takes place twice during acceleration/deceleration

If "OnCompensation" and "OffCompensation" are used for a switch, it can be switched on and off several times when the PLS master position accelerates or decelerates. This is caused, since the PLS requires a constant velocity which cannot be ensured under certain conditions. The bigger problem occurs during deceleration. For some situations when a switch can be enabled at several points in time, refer to the following.

The following equations can be used:

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

s_v is the distance traveled at the specified velocity v_0 in the "OnCompensation" time t_{lead} .

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

s_b is the distance necessary to decelerate at a specified starting velocity v_0 and deceleration a in the time t .

The general equations are:

$$v = a * t \quad \text{or} \quad t = v / a$$

ML_TechMotionBase.library // MX_TechMotionBase.library

The "OnCompensation" time of a programmable limit switch "moves" the switch in the direction of lower positions according to equation 3.

The distance required to come to a standstill at a specified velocity can be determined with equation 4.

The following general rule applies:

If the position is reached with $v = v_0$ so that the output of a programmable limit switch can be switched on with "OnCompensation" time - and the input position is decelerated in a way which ensures that the stop position is the switch-on position of the programmable limit switch without "OnCompensation" time ($v=0$), the output in between is not switched off.

When equating the equations 3 and 4, an equation results which ensures that the switch is only switched on once. Decelerate a little bit faster and the switch is switched on several times.

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

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

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

In equation 5, the starting velocity v_0 used to switch on a limit switch several times increases when the "OnCompensation" time t_{lead} or the deceleration a is increased.

There is no solution for this problem. Only the "OnCompensation" time and the deceleration for this application can be kept as low as possible.

The "Hysteresis" can also be used to consider the variations due to the acceleration/deceleration.

The same considerations can be used for the acceleration and/or negative velocities.

Error codes The function block uses the error table **F_RELATED_TABLE**, 16#0170.
It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR	16#0006	16#00xx	Invalid function block state xx specifies the invalid state
RESSOURCE_ERROR	16#0009	16#0000	Axis reference number has changed during the block execution
RESSOURCE_ERROR	16#0012	16#00xx	Control number is invalid xx refers to the control number
INPUT_INVALID_ERROR	16#2B00	16#0000	Invalid combination of units for velocity, time and position

Tab. 3-31: Error codes of the MC_DigitalCamSwitch function block

3.4.3 MC_CAMSWITCH_REF

Type definitions The following definition describes data types used in the function block [MC_DigitalCamSwitch](#), page 50.

The data type "MC_CAMSWITCH_REF" contains an array of 32 basic structures each of which describes a single switch ([ML_BASIC_CAMSWITCH_REF](#), page 65).

Type definition

```

TYPE MC_CAMSWITCH_REF :
STRUCT
  // This is an array of basic structures, one for each individual switch.
  // Numbering starts at 0!
  // I.e. the first switch is addressed by rSwitch[0].xxxx
  rSwitch : ARRAY [0..31] OF ML_BASIC_CAMSWITCH_REF;
END_STRUCT
END_TYPE

```

3.4.4 ML_BASIC_CAMSWITCH_REF

Type definitions The following definition describes data types used in the function block [MC_DigitalCamSwitch](#), page 50.

The data type [MC_CAMSWITCH_REF](#), page 65, contains an array of 32 basic structures each of which describes a single switch.

Values are given in units [u]. These relate to the corresponding axis.

If e.g. a switch-on position is at 20u and the corresponding axis is scaled in degrees, the switch-on position correspondingly is 20°.

The basic structure "ML_BASIC_CAMSWITCH_REF" is as follows:

Structure element	Type	Description
FirstOnPosition [u]	REAL	Lower limit at which the output is ON (switch-on position)
LastOnPosition [u]	REAL	Upper limit at which the output is ON (turn-off position)
Duration	TIME	Coupled with time-related "CamSwitchMode". I.e. the output remains on for the specified time
TrackNumber	INT	"TrackNumber" is the reference for the switch track
AxisDirection	INT	Direction in which the programmable limit switch works: both (=0; Default); positive (1); negative (2)
CamSwitchMode	INT	position-related (=0; Default); time-related (=1)

Tab. 3-32: Definition of the basic elements of the [ML_BASIC_CAMSWITCH_REF](#) structure

3.4.5 MC_TRACK_REF

Type definitions The following definition describes data types used in the function block [MC_DigitalCamSwitch](#), page 50.

A switch track is an element describing the characteristics of the programmable limit switch related to an output.

The data type "MC_TRACK_REF" contains an array with 8 basic structures (i.e. 8 switch tracks which are allocated to 8 outputs ([ML_BASIC_TRACK_REF](#), page 66).

Type definition

```

TYPE MC_TRACK_REF :
STRUCT
  // This is an array of basic structures, one for each individual track.

```

ML_TechMotionBase.library // MX_TechMotionBase.library

```
// Numbering starts at 0!
// I.e. the first track is addressed by rTrack[0].xxxx
rTrack: ARRAY [0..7] OF ML_BASIC_TRACK_REF;
END_STRUCT
END_TYPE
```

Several switches can be assigned to each switch track.

3.4.6 ML_BASIC_TRACK_REF

Type definitions The following definition describes data types used in the function block [MC_DigitalCamSwitch](#), page 50.

A switch track is an element describing the characteristics of the programmable limit switch related to an output.

The data type [MC_TRACK_REF](#), page 65, contains an array with 8 basic structures (i.e. 8 switch tracks which are allocated to 8 outputs).

The basic structure ("ML_BASIC_TRACK_REF") of a switch track is as follows:

Structure element	Type	Description
OnCompensation	TIME	Lead time of the switch track related to the time of turn-on.
OffCompensation	TIME	Lead time of the switch track related to the time of turn-off.
Hysteresis [u]	REAL	Hysteresis can be set in order to filter noise-afflicted input signals.
TimeDef	WORD	Time definition: ¹⁾ Bit 0 set: "OnCompensation" time is negative; Bit 1 set: "OffCompensation" time is negative;

Tab. 3-33: Definition of the basic elements of the MC_TRACK_REF structure

Several switches can be assigned to each switch track.

3.4.7 MC_OUTPUT_REF

Type definitions The following definition describes data types used in the function block [MC_DigitalCamSwitch](#), page 50. The data type definitions are contained in the library.

The data type describes the 8 outputs of the programmable limit switch.

"MC_OUTPUT" contains an ARRAY with 8 boolean variables which are assigned to the outputs and it looks as follows:

Structure element	Type	Description
OUTPUT	ARRAY [0..7] OF BOOL	Outputs of the programmable limit switch.

Tab. 3-34: Definition of the basic elements of the MC_OUTPUT_REF structure

3.4.8 MB_DigitalCamSwitchOvs

Brief description The MB_DigitalCamSwitchOvs function block implements a freely programmable limit switch with up to 64 switches on 64 switch tracks.

In contrast to the MC_DigitalCamSwitch function block, the function block MB_DigitalCamSwitchOvs supports oversampling ([MC_DigitalCamSwitch](#), page 50). Thus, a corresponding higher temporal resolution of the PLS signals can be achieved.

¹⁾ A negative time means that the output is more likely to switch, a positive time means that the output switches at a later time, measured at the switching point.



Use the function blocks with an I/O module that supports oversampling, e.g. with the Sercos block I/O module R-ILB S3 24 DI16 DIO16/F.

Assignment: Target system/library

Target system	Library
MLC/XLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-35: Reference table of the MB_DigitalCamSwitchOvs function block

Interface description

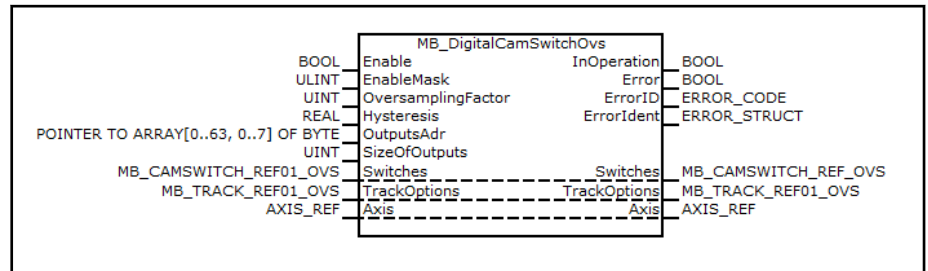


Fig. 3-21: MB_DigitalCamSwitchOvs function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enables the switching outputs
	EnableMask	ULINT	64 bits of type BOOL. Enables the different switch tracks. The lowest-order bit is assigned to the lowest "Track-Number". If the value is set to 1 or TRUE, the assigned TrackNumber is enabled. If the bit is not set, the corresponding bit stream of the switch tracks is permanently set to 0.
	OversamplingFactor	UINT	Factor for oversampling: Indicates the number of samplings in one Sercos cycle and thus determines the best-possible position resolution for position-related switches depending on axis velocity and Sercos cycle time. Thus, the oversampling factor also determines up to what number of times an output (a switch track) can change its state in a Sercos cycle up to the next function block call.
	Hysteresis	REAL	Hysteresis can be set to filter noise-afflicted input signals.
	OutputsAdr	POINTER TO ARRAY [0..63,0..7] OF BYTE	Pointer to output data array: Reference to the signal outputs directly related to the corresponding switch tracks. First array index: Indicates the switch track. Second array index: There are up to 64 states in one Sercos cycle for each switch track: 8 bytes with 8 bits each. The least significant bit in byte 0 corresponds to the first state of the switch track during the PLC cycle.

ML_TechMotionBase.library // MX_TechMotionBase.library

I/O type	Name	Data type	Description
	SizeOfOutputs	UINT	Size of the output data array: The oversampling factor specifies the minimum size of the output data array. The output data array requires up to 512 bytes: If the oversampling factor is 64, 8 bytes = 64 bits are required for each switch track. This results in a maximum output data length of 64x8 bytes = 512 bytes.
VAR_OUTPUT	InOperation	BOOL	The enabled switch tracks are calculated and the outputs are updated
	Error	BOOL	Indicates that an error occurred in the function block
	ErrorID	ERROR_CODE	Error ID
	ErrorIdent	ERROR_STRUCT	Error structure with further error subdivision. Error codes, page 89
VAR_IN_OUT	Switches	MB_CAMS- WITCH_REF_OVS	Reference to the switch structure containing all definitions for all switches. MB_CAMSWITCH_REF_OVS, page 92
	TrackOptions	MB_TRACK_REF_OVS	Reference to the structure containing the corresponding switch track characteristics, e.g. the on/off lead time per output/switch track. MB_TRACK_REF_OVS, page 93
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Reference to the axis to which the programmable limit switch is assigned

Tab. 3-36: Interface variables of the MB_DigitalCamSwitchOvs function block

Note on the "OutputsAdr" input

Depending on the oversampling factor, the output data array, whose address is transferred to the "OutputsAdr" input, can be of variable size. The switch track is referenced via the array index. As the number of switch tracks is specified (64), the dimension of the output data array always has to be declared from 0 to 63. The size of the individual index of the output data array can vary depending on the oversampling factor. In case of a 16-times oversampling, there are for example 16 states in one Sercos cycle for each a switch track. The 16 states can be mapped with one word (= 16 Bits).

A possible declaration of the "MyOutputs" output data array is shown in the following with an oversampling factor of 16.

Declaration of the output data array

```
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;
```

To increase the oversampling factor, change the data type of the output data array respectively.

Data types depending on the oversampling factor

Oversampling: Factor	Basis: Data type	Declaration for "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 factors, data types and declaration***Recommendation**

If no oversampling factor should be specified, always use an LWORD (64 bits) for the data type of the output data array. For this declaration, the oversampling factor can be freely selected. If an oversampling factor smaller than 64 is used, it is only written to the least significant bits for the respective track (depending on the oversampling factor).

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL	FALSE	TRUE	FALSE	Continuous
EnableMask	ULINT	0	2 ⁶⁴ -1	0	Continuous
OversamplingFactor	UINT	1	64	1	Positive edge at "Enable"
Hysteresis	REAL			0.0	Positive edge at "Enable"
OutputsAdr	POINTER TO ARRAY[0..63,0..7] OF BYTE				Continuous
SizeOfOutputs	UINT	64	512	0	Positive edge at "Enable"
Switches	MB_CAMSWITCH_REF_OVS	Refer to the data type description	Refer to the data type description	Refer to the data type description	Continuous
TrackOptions	MB_TRACK_REF_OVS	Refer to the data type description	Refer to the data type description	Refer to the data type description	Positive edge at "Enable"
Axis	AXIS_REF	CntrlNo: 0 AxisNo: 1	CntrlNo: 0 AxisNo: 512	CntrlNo: 0 AxisNo: 0	Positive edge at "Enable"

*Tab. 3-38: Min./max. and default values of the function block MB_DigitalCamSwitchOvs***Functionality**

The MB_DigitalCamSwitchOvs function block implements a freely programmable limit switch with up to 64 switches on 64 switch tracks. The switches can be freely distributed to the 64 switch tracks.

The switches are not output as individual bits, but as "bit stream". Imagine as if the bit stream is output via a temporally-clocked shift register at a binary output via oversampling.

The oversampling is specified using the "oversampling factor" (input "OversamplingFactor"). It refers to the Sercos cycle time.

Oversampling factors of 2^n with $n = [0..6]$ are permitted. This results in the factors 1, 4, 8, 16, 32 and 64. In case of an oversampling factor of 1, there is no oversampling.



As the Sercos block I/O module R-ILB S3 24 DI16 DIO16/F does not support the oversampling factor 2, factor 2 is also not permitted by the function block.

Settings

The units for position and velocity have to be set for the axis assigned to the programmable limit switch. If the axis positions are defined in degrees for example, the programmable limit switch positions are also interpreted in degrees.

To correctly support the oversampling functionality with an oversampling-capable I/O module, the function blocks have to be called mandatorily in a Sercos-synchronous task.

Therefore, select "EVENT_OP_MODE_SERCOS_CYCLIC" as external event and "External" in the task configuration for an **MLC/XLC**.

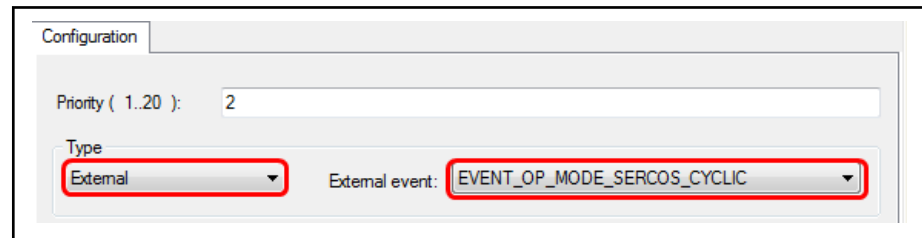


Fig. 3-22: Presetting of the Sercos-synchronous task MLC/XLC

In case of an **MLD**, the settings in the task configuration are as follows: Select the type "External" and "CCD_SYNCHRONIZED_TASK" as external event.

If the function block is not called in a Sercos-synchronous task, the error 16#2B03 is generated at the error outputs of the function block.

The Motion cycle time (C-0-0400) the Sercos cycle time (C-0-0503) have to have the same value in an MLC/XLC. For the dialog to set the Sercos cycle time and the Motion cycle time, go to the context menu of the "Motion" node in the Project Explorer.

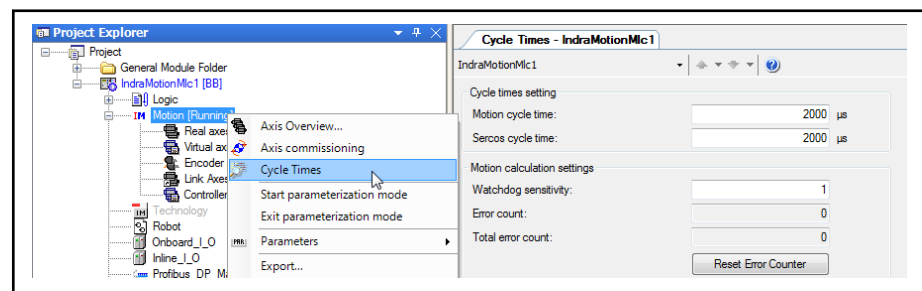


Fig. 3-23: Dialog to set the Sercos cycle time or the Motion cycle time in an MLC/XLC

If the Motion cycle time and the Sercos cycle time are provided with different values, the error 16#2B09 is generated at the error outputs of the function block.

Data types

The function block MB_DigitalCamSwitchOvs uses the following data types:

- [MB_TRACK_REF_OVS](#), page 93

The data type MB_TRACK_REF_OVS contains an array with 64 basic structures, i.e. 64 switch tracks that can be assigned to 64 oversampling-capable outputs.

Several switches can be assigned to each switch track.

- [ML_BASIC_TRACK_REF_OVS, page 93](#)

A switch track is an element describing the characteristics of the programmable limit switch related to an output capable of oversampling.

The basic structure ML_BASIC_TRACK_REF_OVS contains elements to set the behavior for a switch track.

- [MB_CAMSWITCH_REF_OVS, page 92](#)

The data type MB_CAMSWITCH_REF_OVS contains an array of 64 basic structures each describing a single switch.

- [ML_BASIC_CAMSWITCH_REF_OVS, page 93](#)

The basic structure ML_BASIC_TRACK_REF_OVS contains elements to set the behavior for each single switch.

Example:

Functional description

When using the "Axis" input, the current axis position and velocity required for the computations can be accessed.

After a rising edge at the "Enable" input, all values are applied at the function block inputs and the function block-internal initialization phase is started. During the initialization phase, the "InOperation" output is set to FALSE. The initialization phase requires some PLC cycles. After the initialization phase, the "InOperation" output is set to indicate that the function block is performing its cyclical task.



During the initialization, Motion and the Sercos cycle times are retrieved once and not checked anymore during further function block processing. The Sercos time is significant for the switching accuracy of the switch in case of the given oversampling factor.

When changing the Sercos cycle time in the parameter C-0-0503 and/or the Motion cycle time in the parameter (MLC/XLC), or S-0-0001 (MLD), start a new initialization at a rising edge at the "Enable" input of the function block.

With the "EnableMask" input, outputs can be enabled or blocked.

The lowest-order bit is assigned to the lowest "TrackNumber". If the value is set to 1 or TRUE, the corresponding "TrackNumber" is enabled. If the bit is not set, the respective output is set to 0 irrespective of whether the positions for the current Sercos cycle interpolated with oversampling are within the range of a switch on the switch track.

If "EnableMask" is 0, "InOperation" becomes inactive, since none of the outputs is enabled.



To reach the lowest computation times possible, always disable all switch tracks that are not required using the "EnableMask" input!

Each switch has a starting and an end position. This allows to specify a "FirstOnPosition" and a "LastOnPosition" (or time) for each switch. In addition

ML_TechMotionBase.library // MX_TechMotionBase.library

to a mechanical switch, the switch can be enabled for a certain period and a lead time can be defined for each switch track.

If (`FirstOnPosition` > `LastOnPosition`), the switch acts inversely and is only switched off within these positions.

Area/modulo calculations

The PLS can be used for rotary as well as for translatory axis scalings. Position data can be computed in modulo format as well as in absolute format (axis running forward and backward).

In both cases, the programmable limit switch units are based on the units defined for the axis.

AxisDirection indicates in which direction of motion the switch becomes active. If the "AxisDirection" is 0 (default), the switch becomes active in both directions.

If the "AxisDirection" is 1, the output is only set if it is traversed in positive direction.

If the "AxisDirection" is set to 2, the output is only set if it is traversed in negative direction.

CamSwitchMode switches between the position or time display.

"CamSwitchMode" = 0 refers to the position mode. The output is enabled at "FirstOnPosition" and disabled at "LastOnPosition".

If "CamSwitchMode" = 1, the output is enabled at "FirstOnPosition" and disabled after the specified "Duration".

Duration: Switched-on period of the time-controlled switch. The output is disabled when the time elapses, even if the position is still in the switch range. Due to this time-controlled switch, the output is switched on again only after the axis position left the switch area and was positioned again in the switch area.



The switch-on period of the switch (duration) can be specified in nanoseconds.

The resolution of the switching accuracy (switching off the switch at a defined point in time) results from the settings of the Sercos cycle time (MLC/XLC: C-0-0503, MLD: S-0-0001) and the set oversampling factor.

If the set "Duration" is lower than the duration of the sample, the output is at least switched on for one sample.

Example:

If the Sercos cycle time is 2 ms and the oversampling factor is 16, a time resolution of $2\text{ ms}/16 = 125\text{ }\mu\text{s}$ results.

If the duration is 50 ns, the output is still switched on for 125 μs in case of these settings.

The lead time (**OnCompensation** and **OffCompensation**) can be positive or negative. Negative means that the output switches before the switching position is reached. The lead time is indicated in nanoseconds (also refer to "TimeDef"). The lead times can be set separately for each switch track.

Hysteresis: This parameter prevents the output from switching too often if the axis is close to the switching point and the current position varies around the switching point. The value set at the "Hysteresis" input is applied to all switches on all switch tracks. If a negative value is entered, it is automatically applied as positive value.



The hysteresis affects this function block differently than the MC_DigitalCamSwitch function block.

The hysteresis becomes only effective when the direction is changed. In forward as well as in backward direction, the hysteresis has to be run through first before the master position is applied. Thus, a switch can switch at switching points different from the parameterization of the switch.

The following figure shows two PLS, each defined between 100° and 150°.

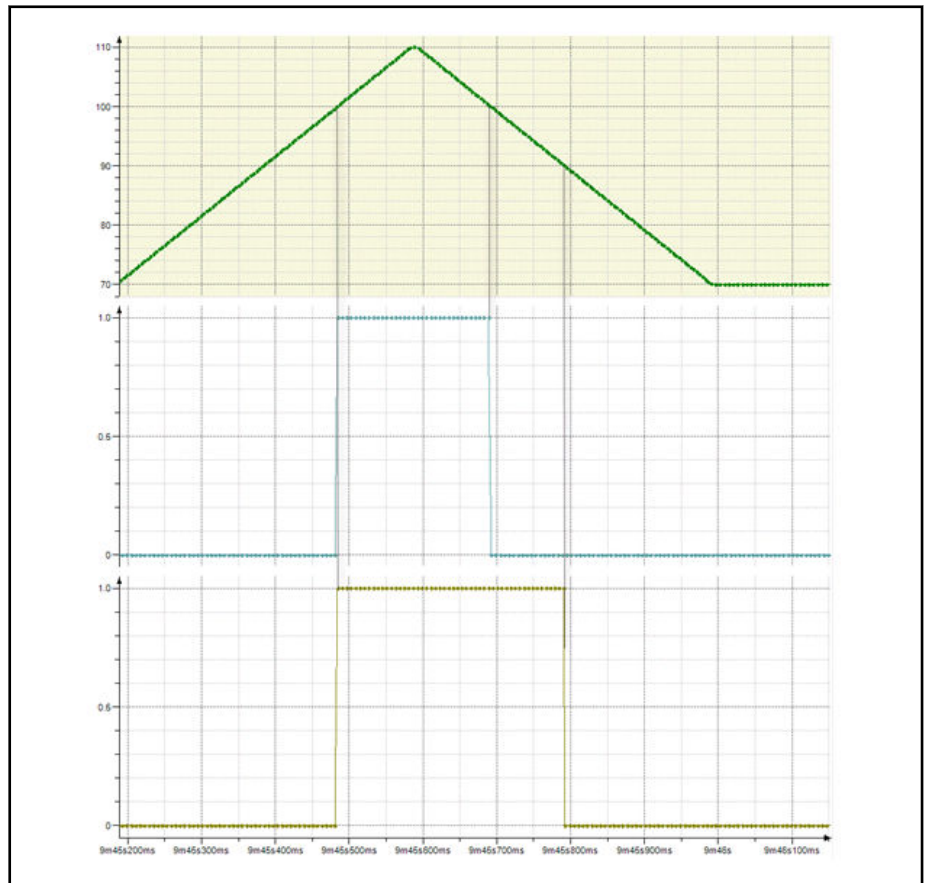


Fig. 3-24: Effects of the hysteresis on the switching points of a switch

The center curve shows the switch output without hysteresis, the lower curve shows the switch output with a hysteresis of 20°. The upper curve represents the master position.

The master first moves in positive direction via the beginning of the switches and then in negative direction. It can be seen that the switch with and without hysteresis switch on at 100°. At 110°, the master changes its direction and moves now in negative direction. The switch without hysteresis switches off again at 100°. In contrast, the switch with hysteresis switches off at 90°. This is caused, as for the switch with hysteresis, the hysteresis has to be crossed after a change in direction before the master position is applied again. In the example, the direction is changed at 110° and the hysteresis is 20°. Thus, the master position is only applied again at $110^\circ - 20^\circ = 90^\circ$. At 90°, the switch is switched off. Thus, the output is deleted.

TimeDef: Use this parameter to differ between positive and negative times for the "On/OffCompensation".

If bit 0 is set, the "OnCompensation" time is negative

If bit 1 is set, the "OffCompensation" time is negative.

Example:

"OffCompensation" is set to 300 ms, "TimeDef" is set to 2.

This means that the lead time is "-300ms".

The following figure shows the mechanical equivalent of the digital programmable limit switch.

Note that not all functions were shown, since it is either not possible or very difficult to show them.

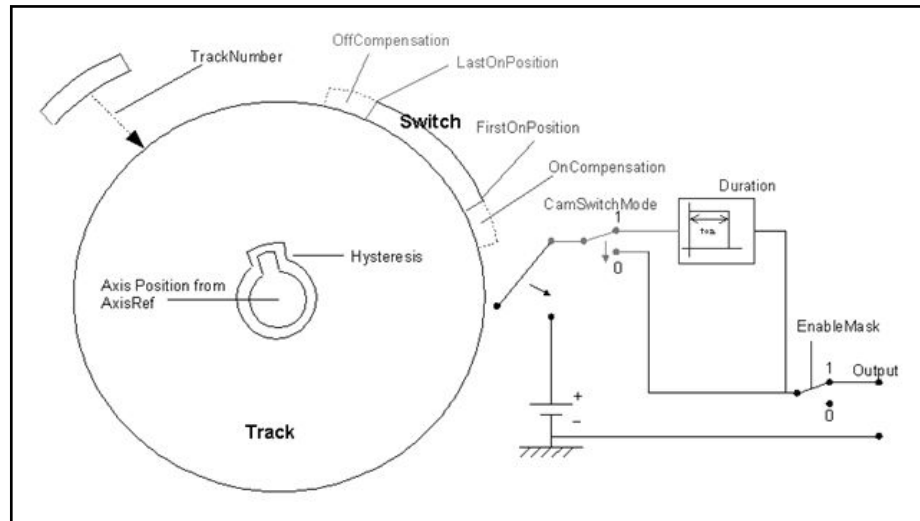


Fig. 3-25: Mechanical equivalent of the digital programmable limit switch

Example program

A short example explains how the function block is programmed. The different structure elements are explained later in detail.

It is assumed that a single switch is to be created:

- With number 4 on switch track 3
- With a "FirstOnPosition" of 35 and
- A "LastOnPosition" of 36
- The switch should trigger 125 μ s earlier and the switch track has a hysteresis of 3
- the switch should switch on/off with regard to the position and not to the time and it should only be applied in positive direction
- The axis should move with a velocity of 1000 rpm. Ensure, that the over-sampling factor can safely detect the switch.

The variable declaration and the function block call are as follows:

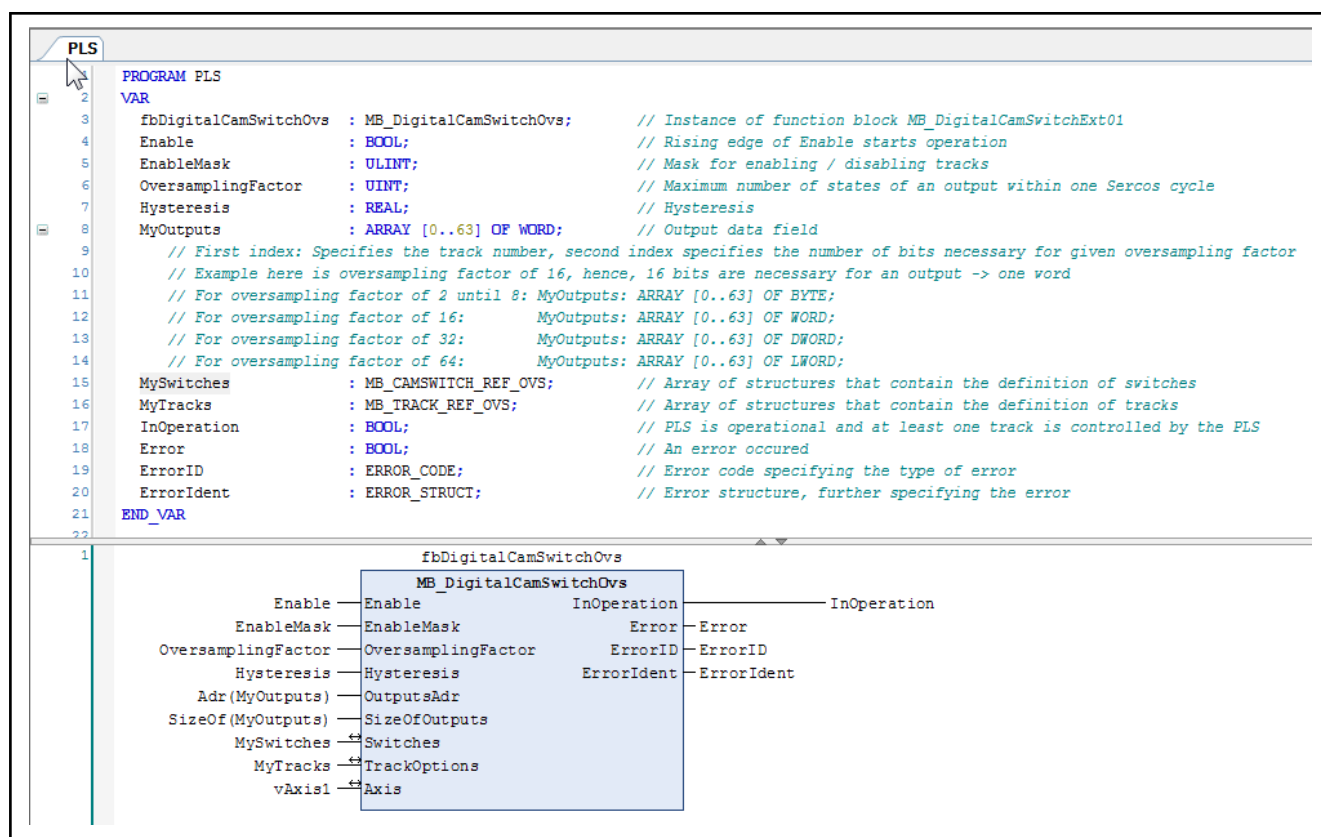


Fig. 3-26: Example program: Variable declaration and function block call

Refer to the two input structures "MySwitches" and "MyTracks" which define all switch characteristics, switch tracks and outputs. To understand these structures better, have a closer look at the data content of the structures provided for the results of the example above:

XH2.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 compensation time. Positive values are defined as delay. See also TimeDefinit
OffCompensation	LTIME	LTIME#0ns			Off compensation 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				

Fig. 3-27: Definition of the switch track

First, the switch track is defined (figure above).

- To use switch track 3, initialize **rTrack[2]**, as the array starts with the index 0.
- The OnCompensation time is set to 125 μ s. To ensure that the switch switches earlier and not later, set bit 0 in "TimeDefinitions" defining the "OnCompensation" time as negative.

ML_TechMotionBase.library // MX_TechMotionBase.library

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 assig
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				

Fig. 3-28: Defining a switch

The switch is defined (figure above).

1. Initialize the switch structure with index 3 to initialize switch 4, since the array starts with the index 0.
2. "FirstOnPosition" is set to the 35, "LastOnPosition" is set to 36 and "AxisDirection" is set to 1. This means that the switch is only applied in positive direction.
3. To assign the switch to a switch track, set the "TrackNumber" to 2. This specifies that this switch is assigned to the switch track with index 2 in the switch track structure.
4. Finally, the "CamSwitchMode" has to be set to 0 to switch on the position-related mode.



The individual elements of the REF structures can be accessed according to the following pattern:

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

Oversampling

- The function block is called in a Sercos-synchronous task: The Sercos cycle time should be 2 ms.
- The axis should move with a velocity of 1000 rpm.

The sampling theorem indicates that the sampling rate is twice as high as the highest signal frequency required to detect the signal. This also applies to the PLS. The PLS samples the position data every time it is called by the PLC and interpolates a number of position values depending on the oversampling factor for the current Sercos cycle. This increases the sampling rate by the oversampling factor.

Every switch switched on for a time shorter than the sampling rate (resulting from the Sercos cycle time and the oversampling factor) is not identified correctly. The following equations show how the velocity or the size of the switch, for which switching actions can be lost, are calculated.

Rotary systems:

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

n is the revolution per second [1/s].

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

ρ is the angle in rad and t is the time in seconds [s].

Conversion factor from degree to rad: $\pi / 180$

Result when using the equations A1 and A2:

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

for the angle or

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

If the factors are adapted for common units, the following results:

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

or for the velocity:

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

The equations 3 and 4 provide the two equations required to determine whether switching actions can be lost.

For example, if the maximum speed of 1,000 min⁻¹ is known and the PLS is called in a cyclic task with a cycle time of 2 ms. What is the minimum size of a switch to ensure that it can be identified by the PLS?

Result when using equation 3:

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

I.e., the minimum switch size has to be 12° to ensure that it can be identified at a speed of max. 1,000 min⁻¹.

Using oversampling increases the sampling rate by the oversampling factor.

As the size of the switch in the example is 1° and the axis moves with a velocity of 1000 rpm, the oversampling factor has to be at least 12 to ensure that the switch is safely detected.

As only squares are allowed for the oversampling factor, the oversampling factor has to be set at least to 16 to ensure that the switch is safely detected.

Declaration of the output data array

```
MyOutputs: ARRAY [0..63] OF WORD;           // Output data field
// First index: Specifies the track number,
```

ML_TechMotionBase.library // MX_TechMotionBase.library

```
// 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;
```

In the example, the "MyOutputs" array is used for the output data array.

The switch track is referenced via the array index. In case of a 16-times oversampling, there are 16 states in one Sercos cycle for each a switch track. The 16 states can be mapped with one WORD (16 bits per WORD). The least significant bit in Word corresponds to the first state of the switch track during the PLC cycle.

Depending on the oversampling rate, the size of the output data array can be variable. Due to the variable size, the address (starting address) of the output data array is applied at the function block input "OutputsAdr". For that purpose, use the address operator "ADR".

To ensure that the function block knows the size of the output data array, it has to be applied at the "SizeOfOutputs" input. Use the "SIZEOF" operator determining the number of bytes of the output data array.



The output data array in the interface of the function block is declared as POINTER TO ARRAY [0..63,0..7] OF BYTE. The second dimension is also an array of BYTE. The notes on the input "OutputsAdr" (see ["Note on the OutputsAdr input" on page 68](#)) only inform on the use of a one-dimensional array with different data types. However, as only the address is transferred, there is no contradiction. However, it is required to use the variant with different data types for a later assignment of the I/O area as shown in the following example.

Preparing the function block start:

To start the function block, add the remaining variables to the function block and set their values accordingly.

1. The "Enable" input has to be set
2. In the "EnableMask" input, the corresponding bit has to be set:
2#00000100, bit 2 for third switch track **rTrack[2]**.
3. The "Hysteresis" input has to be set to the requested value 3.

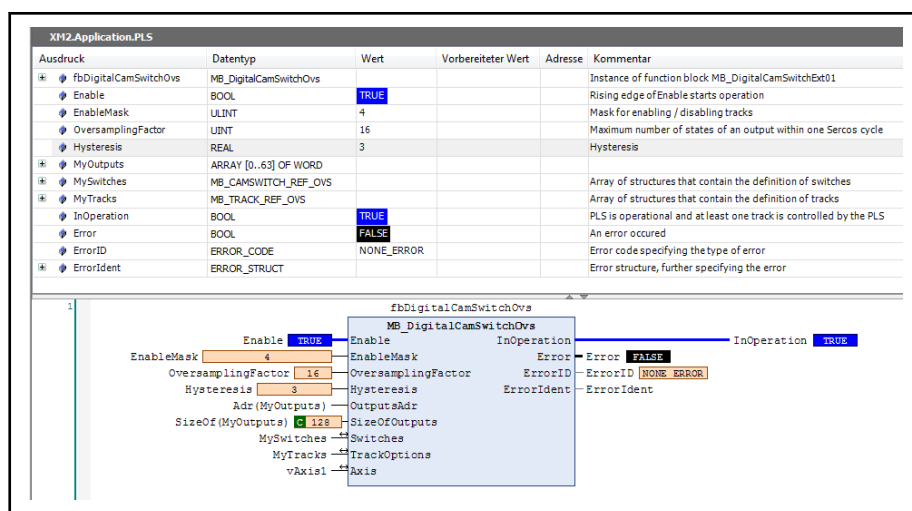


Fig. 3-29: Running function block

Configuring the I/O module

To output the I/O signals, use an I/O module that supports oversampling. In this example, the oversampling configuration with the Sercos block I/O module R-ILB S3 24 DI16 DIO16/F is shown. To use the oversampling, set the connection type "cycle-synchronous" and the configuration with "IDNs" in the "Extended settings" section (refer to the following figure).

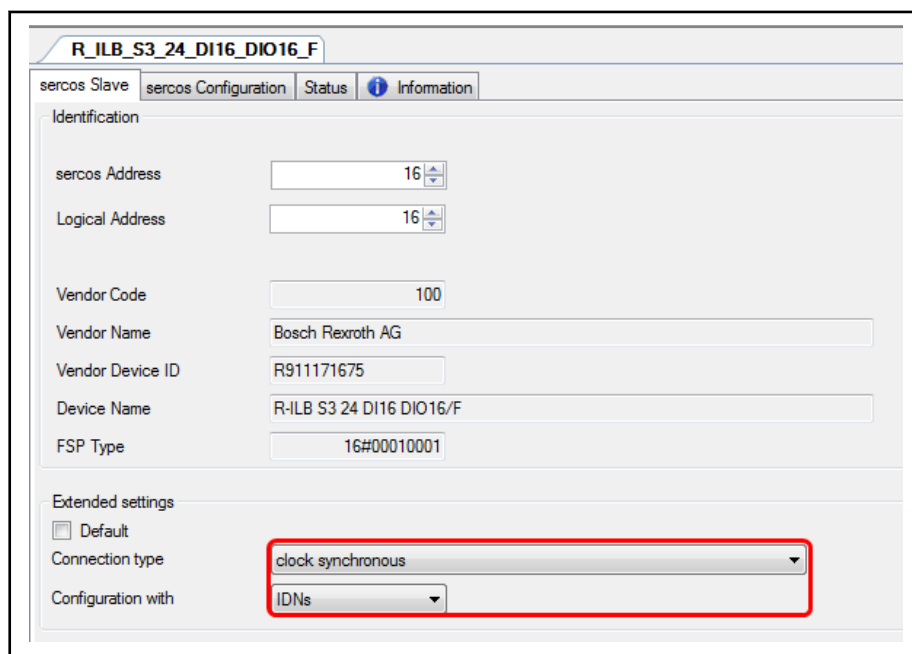


Fig. 3-30: Sercos slave dialog "Extended settings"

The number of samplings can be set in the "Oversampling" tab.

To set the number of samplings (OversamplingFactor) on the output channels, double-click on the "Number of samplings" column in the "OVS OUT" line.

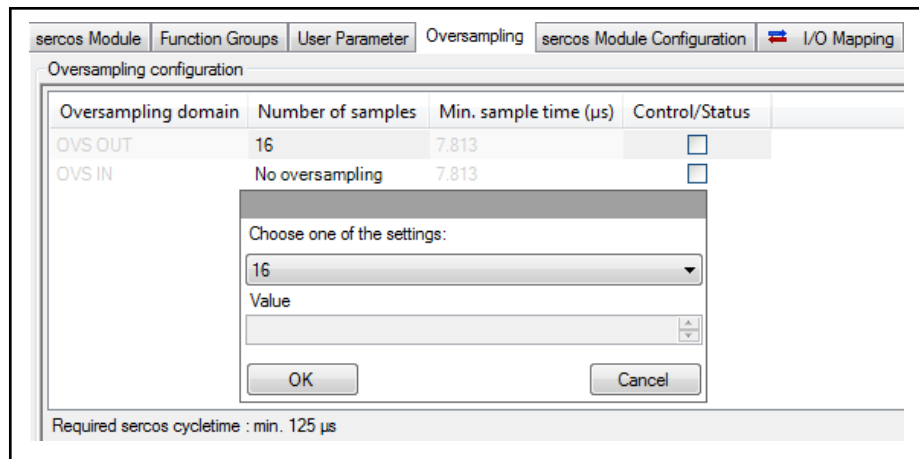


Fig. 3-31: Tabulator oversampling



The function block does not check whether the value applied at the function block input "OversamplingFactor" matches the factor set at the I/O module.

Select the outputs of the I/O module to be enabled for oversampling in the lower section. As only one track is enabled in the example, the oversampling functionality has also to be enabled for only one output.

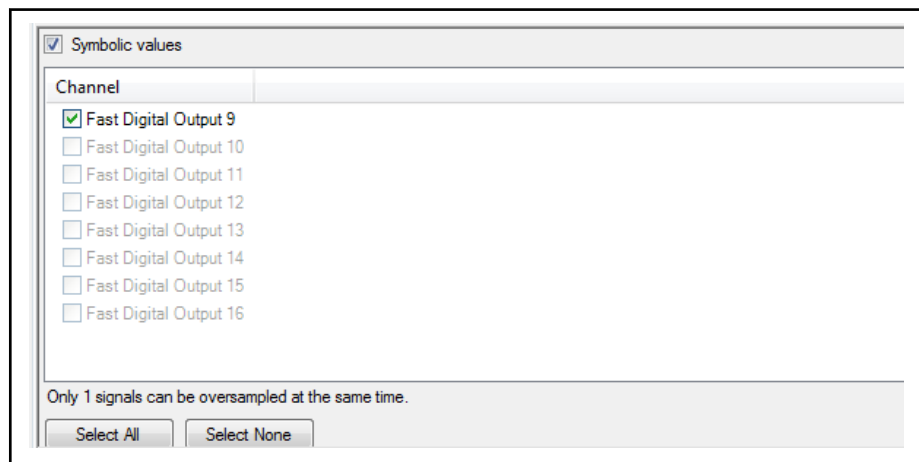


Fig. 3-32: Enabling oversampling



The Sercos block-I/O module R-ILB S3 24 DI16 DIO16/F supports 8 oversampling-capable outputs. To output more switch tracks of the function block with oversampling, more I/O modules are required accordingly. For all I/O modules, whose outputs are assigned to the tracks of a function block, the oversampling factor set has to be the same as at the function block.

At the end, the output data array of the third switch has to be mapped in the I/O mapping: The following figure shows the mapping of the 16 bits for the third switch on the output 9 (Fast Digital Output 9) of the I/O module.



After enabling the oversampling for at least one output, the folder "OVS OUT" folder is created in the I/O mapping.

As switch track 3 is used in the example, configure the starting address of the bit stream for switch track 3 for the oversampling-capable output:

The starting address of the bit stream for switch track 3 is "My-Outputs[2].

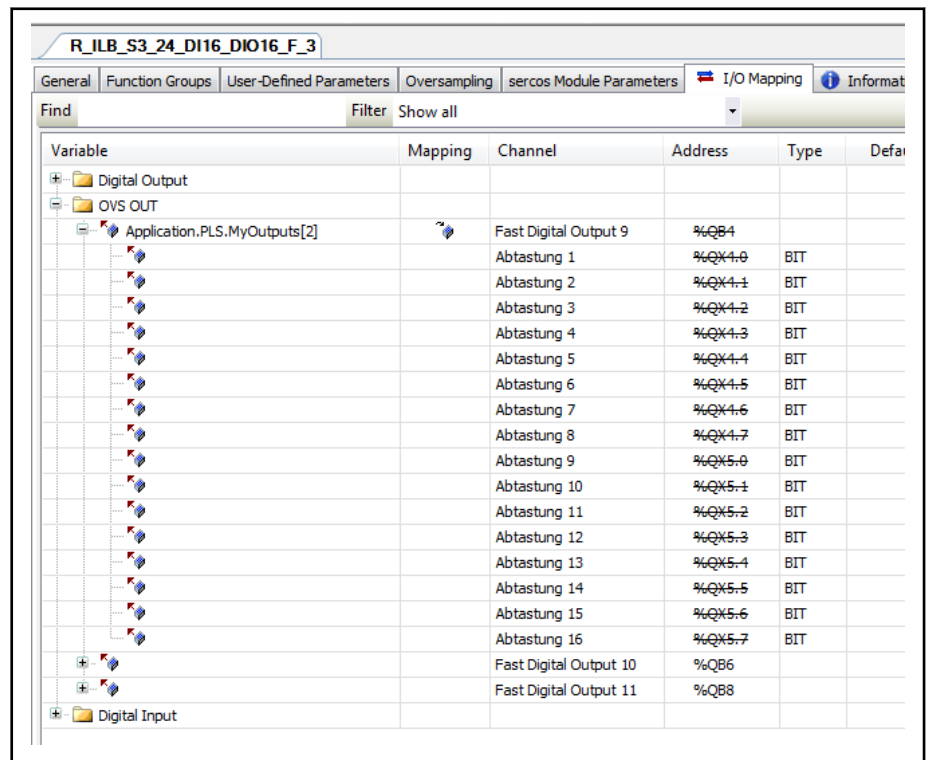


Fig. 3-33: Mapping in I/O mapping

Time diagrams

The following example shows the use of the MB_CAMSWITCH_REF_OVS structure and of the time-based switches. Find a specified value configuration in the following list:

Parameters	Type	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 = Both	0 = Both		
CamSwitchMode	BYTE	0 = Position	0 = Position	0 = Position	1 = Time		

Tab. 3-39: Example: MB_CAMSWITCH_REF_OVS structure



Values are given in units [u]. These relate to the corresponding axis. If e.g. a switch-on position is at 20u and the corresponding axis is scaled in degrees, the switch-on position is 20°.

ML_TechMotionBase.library // MX_TechMotionBase.library

This example uses the values from the example "MB_CAMSWITCH_REF_OVS" above. It neither uses "On/OffCompensation" nor "Hysteresis".

This is the behavior of the outputs when the axis moves continuously in positive direction. The axis is a modulo axis with a modulo length of 50 u (units).

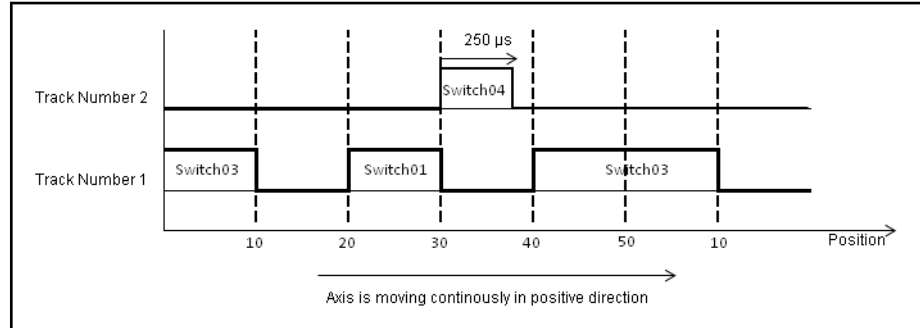


Fig. 3-34: Switch "Switch01" when moving in positive direction

This example uses additionally

OnCompensation= -125μs and OffCompensation= +250μs.

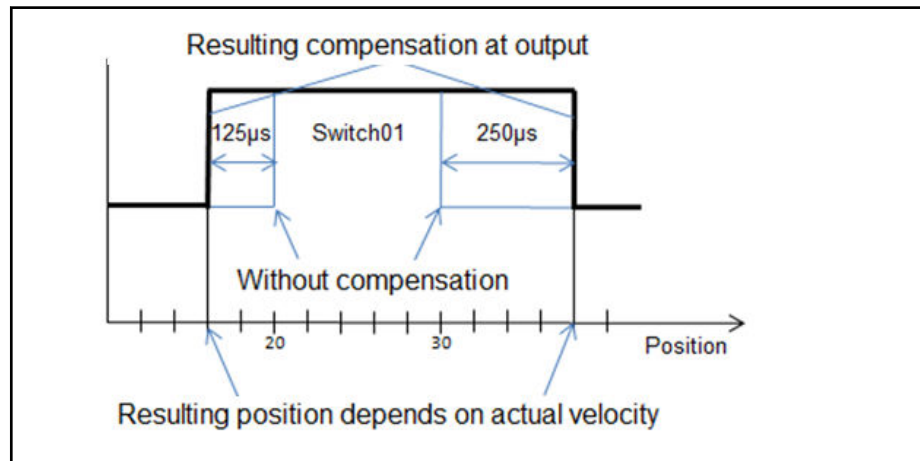


Fig. 3-35: Switch "Switch01" with On/OffCompensation

This is the behavior of the outputs when the axis moves continuously in negative direction without "On" or "OffCompensation" and without "Hysteresis":

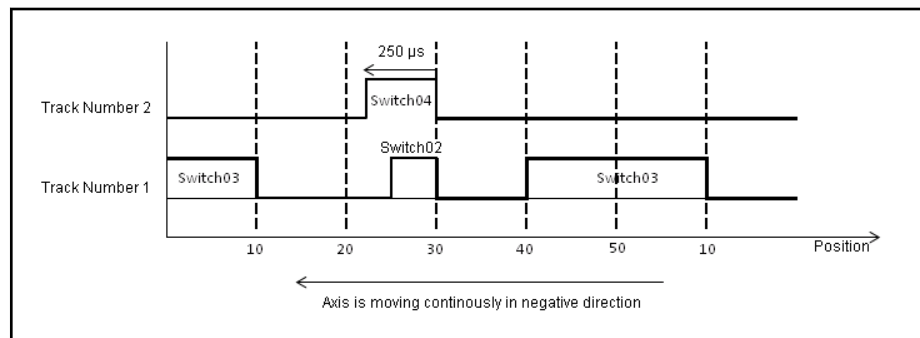


Fig. 3-36: Switch when moving in negative direction

Implementation details

The PLCopen extension specifies the interface of the PLS functionality. However, some application details are not specified. Therefore, these were defined for an implementation with a better runtime behavior.

OnCompensation/OffCompensation

PLCopen defines the "On/OffCompensation" only for motions in positive direction. In this case, the "OnCompensation" relates to the "FirstOnPosition" while the "OffCompensation" relates to the "LastOnPosition". How can this be defined for motions in negative direction?

Based on applications for which the lead times are used to compensate for the on and off delays with actuators, this means that the "OnCompensation" is responsible for the rising edge of the signal.

In other words: If it is traveled in positive direction, the "OnCompensation" relates to the "FirstOnPosition" and if it is traveled in negative direction, the "OnCompensation" relates to the "LastOnPosition". The "OffCompensation" is defined accordingly

The following figure shows the behavior for

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

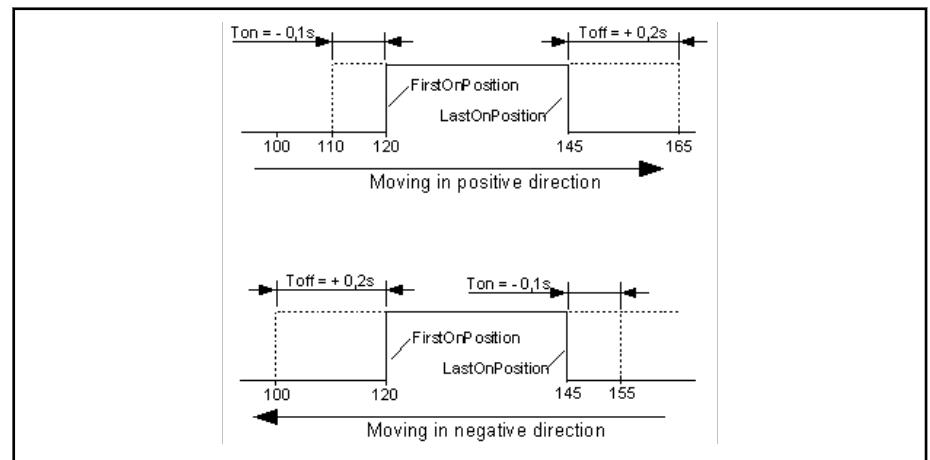


Fig. 3-37: On/OffCompensation when moving in positive and negative direction

Duration

The "Duration" is defined as the time the switch remains switched on after it has been switched on.

The question is what happens if the period has expired but the current position is still in the "FirstOnPosition" and "LastOnPosition". Does the output has to be switched on or remain switched off until the current position left the "FirstOnPosition" and "LastOnPosition" and entered again?

It has been decided to wait for a rising edge of the switch output. In other words: When the time elapses, the output is disabled. Only after the current position moved out of the switch limits and enters again, the output is switched on again and the "Duration" is restarted.

The same applies to the "OnCompensation"/"OffCompensation". The "Duration" begins with the rising edge of a switch, that is at the "LastOnPosition" when moving in negative direction.

The situation is explained in the following figure:

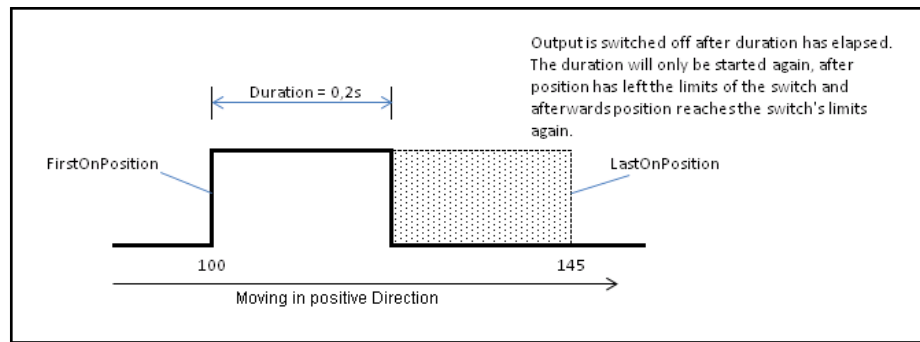


Fig. 3-38: Duration

Combination of OnCompensation/OffCompensation and duration

When using a time-based switch with "On/OffCompensation" times, should these be considered?

The "OnCompensation" is considered. This leads to an earlier/later activation of the output.

The "OffCompensation" period is not considered for time-based switches. The output is switched off after "Duration" elapsed.

How a switch is not used

The PLS has up to 64 switches. The element "TrackNumber" in the MB_CAMSWITCH_REF_OVS structure determines whether a switch is used. This variable contains the switch track number. Valid values are between 0 and 63, since there is a maximum of 64 switch tracks.

The default setting of this variable is 64, it is marked as not used. Every invalid number (64 to 65535) is interpreted as an unused switch and is therefore not evaluated by the firmware.

Limitations of the programmable limit switches (PLS)

The limitations of the programmable limit switch (PLS) are discussed in the following. The restrictions are only definitions which do not cause any error. The system behavior is determined via these definitions. Therefore, the user has to expect different behaviors.

Lost switching actions

The sampling theorem indicates that the sampling rate is twice as high as the highest signal frequency required to detect the signal. This also applies to the PLS. The PLS samples the position data every time it is called by the PLC and interpolates a number of position values depending on the oversampling factor for the current Sercos cycle. This increases the sampling rate by the oversampling factor.

Every switch switched on for a time shorter than the sampling rate (resulting from the Sercos cycle time and the oversampling factor) is not identified correctly. The following equations show how the velocity or the size of the switch, for which switching actions can be lost, are calculated.

Rotary systems:

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

n is the revolution per second [1/s].

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

ρ is the angle in rad and t is the time in seconds [s].

Conversion factor from degree to rad: $\pi / 180$

Result when using the equations A1 and A2:

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

for the angle or

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

If the factors are adapted for common units, the following results:

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

or for the velocity:

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

The equations A3 and A4 provide the two equations required to determine whether switching actions can be lost.

For example, if the maximum speed of 1,000 min⁻¹ is known and the PLS is called in a Sercos-cyclic task with a cycle time of 2 ms and an oversampling factor of 16. What is the minimum size of a switch to ensure that it can still be identified by the PLS? Result when using equation A3:

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

A switch has to be at least 0,75° to ensure that it is identified at a speed of max. 1,000 min⁻¹, a Sercos cycle time of 2 ms and an oversampling factor of 16.

Linear systems:

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

s is the distance in millimeters [mm] and t is the time in seconds [s].

If the Sercos cycle time divided by the oversampling factor is used as sampling time for the PLS and if the corresponding factors to adjust the units are considered, the following equations result:

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

or for the velocity:

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

ML_TechMotionBase.library // MX_TechMotionBase.library

These equations also apply to systems with inch [in] as distance unit:

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

or for the velocity:

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

Example: A switch in a linear system has a length of 0.02. The Sercos cycle time is 2 ms if the oversampling factor is 16. Which maximum velocity is possible to ensure that the switch is identified?

Result when using the equation B5:

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

Switches seem to have the wrong size

The boundary conditions for "lost switching actions" were described above.

Even if the conditions for lost switches are not fulfilled, the switching times might not be exact. The PLS can only change the state of its outputs at the sampling points in time. The distance of two sampling points results from the set Sercos cycle time and the oversampling factor. It is assumed that the PLS is called in a Sercos-synchronous task with a cycle time of 2 ms and an oversampling factor of 16.

If a switch should for example be active at a specified velocity of 200 μ s, the output is set to "On" for either 250 μ s (case A) or 125 μ s (case B).

The switch-on period of the switch and the size of the resulting error (error case A: 50 μ s, error case B: 75 μ s) depend on the sampling point in time of the PLS. When increasing the oversampling factor to 32 (case C) a switch-on period of the switch of 187.5 μ s and an error of 12.5 μ s result.

For a more detailed explanation, refer to the following figure:

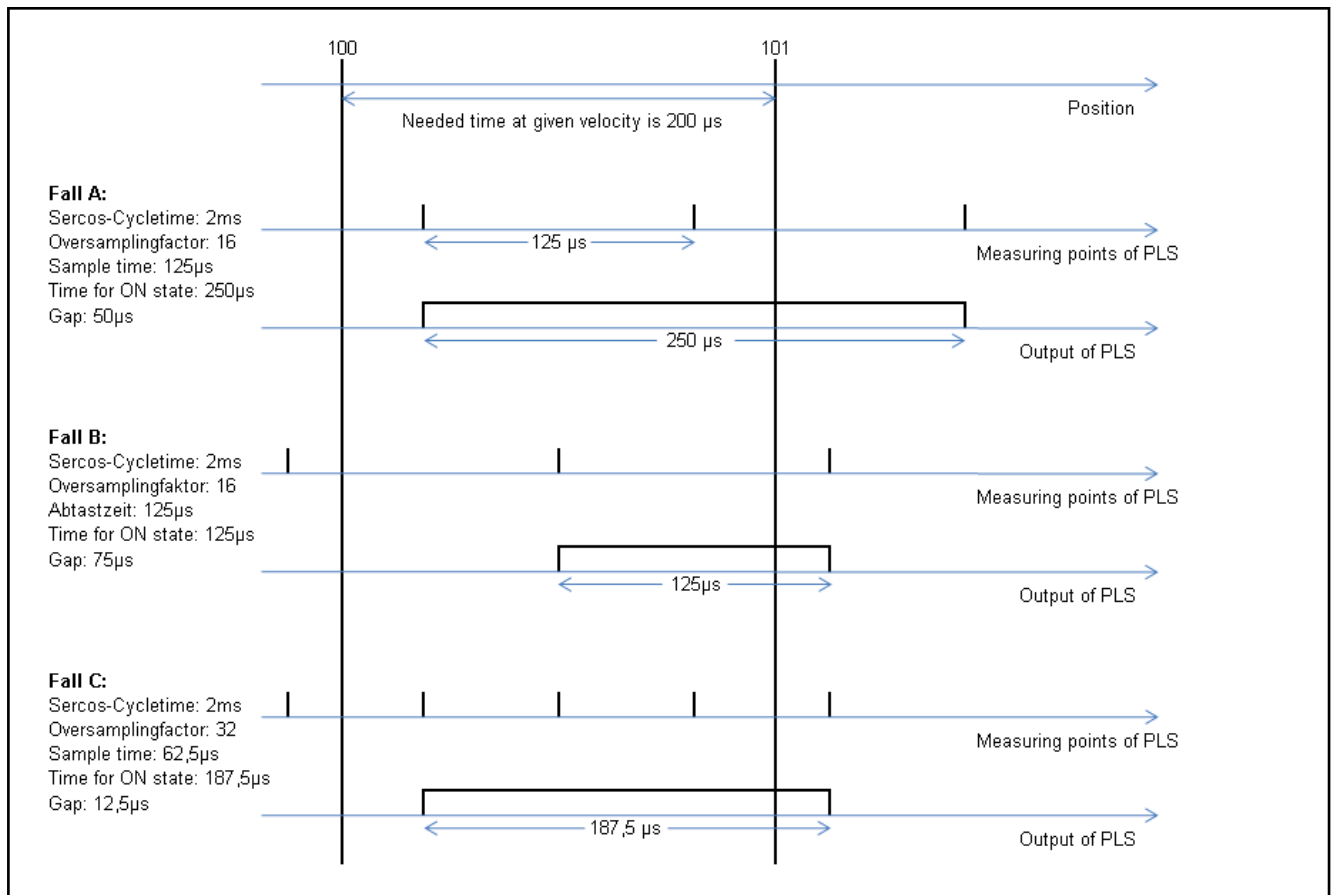


Fig. 3-39: Inexact switch sizes

Switching takes place twice during acceleration/deceleration

If "OnCompensation" and "OffCompensation" are used for a switch, it can be switched on and off several times when the PLS master position accelerates or decelerates. This is caused, since the PLS requires a constant velocity which cannot be ensured under certain conditions. The bigger problem occurs during deceleration.

Example:

- Rotary system with the modulo value 5000°
- The master position moves continuously in positive direction
- FirstOnPosition = 1000
- OnCompensation = 0.2s
- The master velocity is decelerated from 1,000 rpm to 100 rpm, refer to the velocity characteristics in the following figure:

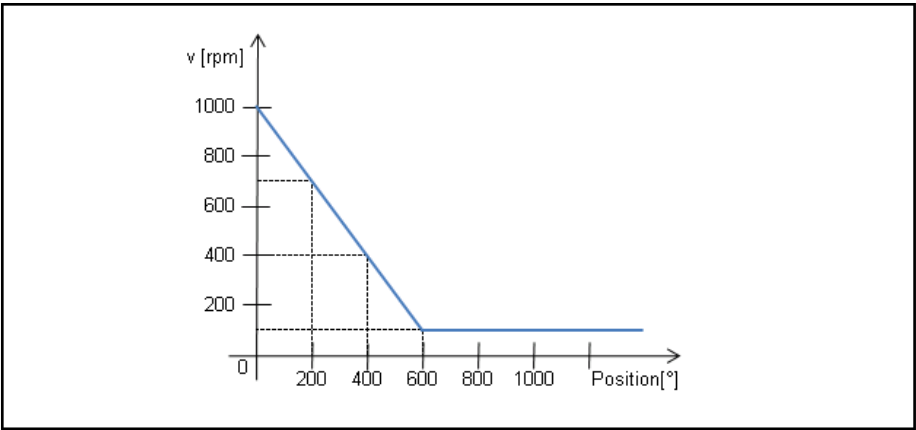


Fig. 3-40: Master velocity

The switch-on position of the switch is calculated based on the "OnCompensation" and the current velocity using the following equation:

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

If 200° is sampled for example, the switch-on position is calculated as follows:

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

This means that the master has already exceeded the switch-on position of the switch and the output is enabled.

The following table shows the computed switch-on positions as well as the corresponding output state at different sampling points in time:

Sampling position	Calculated switch-on position	Output state
0°	4800°	ON
200°	160°	ON
400°	520°	OFF
600°	880°	OFF
800°	880°	OFF
900°	880°	ON

The preceding table shows that it is switched twice due to the delay. Some situations, when a switch can be enabled at several points in time, are shown in the following:

The following equations can be used:

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

s_v is the distance traveled at the specified velocity v_0 in the "OnCompensation" time t_{lead} .

$$s_d = v_0 * t - \frac{1}{2} a t^2 \text{ (Equation 2)}$$

s_b is the distance necessary to decelerate at a specified starting velocity v_0 and deceleration a in the time t .

The general equations are:

$$v = a * t \quad \text{or} \quad t = v / a$$

The "OnCompensation" time of a programmable limit switch "moves" the switch in the direction of lower positions according to equation 1.

The distance required to come to a standstill at a specified velocity can be determined with equation 2.

The following general rule applies:

If the position is reached with $v = v_0$ so that the output of a programmable limit switch can be switched on with "OnCompensation" time and the input position is decelerated in a way which ensures that the stop position is the switch-on position of the programmable limit switch without "OnCompensation" time ($v=0$), the output in between is not switched off.

When equating the equations 1 and 2, an equation results which ensures that the switch is only switched on once.

Decelerate a little bit faster and the switch is switched on several times.

$$\begin{aligned} s_b = s_v &\Rightarrow v_0 t - \frac{1}{2} a t^2 = v_0 t_{lead} \Rightarrow v_0 \frac{v_0}{a} - \frac{1}{2} a \frac{v_0^2}{a^2} = v_0 t_{lead} \\ &\Rightarrow \frac{v_0^2}{2a} = v_0 t_{lead} \\ &\Rightarrow v_0 = 2 t_{lead} a \quad (\text{Equation 3}) \end{aligned}$$

In equation 3, the starting velocity v_0 used to switch on a limit switch several times increases when the "OnCompensation" time t_{lead} or the deceleration a is increased.

There is no solution for this problem. Only the "OnCompensation" time and the deceleration for this application can be kept as low as possible.

The "Hysteresis" can also be used to consider the variations due to the acceleration/deceleration.

The same considerations can be used for the acceleration and/or negative velocities.

Runtime behavior

The MB_DigitalCamSwitchOvs function block implements a freely programmable limit switch with up to 64 switches on 64 switch tracks. Additionally, oversampling is supported by a maximum oversampling factor of 64. In the most CPU-intensive configuration, 64 switches distributed on 64 switch tracks could be operated with 64-times oversampling.



To reach the lowest computation times possible, always disable all switch tracks that are not required using the "EnableMask" input!

Error codes

The function block MB_DigitalCamSwitchOvs uses the error table F_RELATED_TABLE, 16#0170.

It can generate the following error messages in Additional1/Additional2:

ML_TechMotionBase.library // MX_TechMotionBase.library

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR	16#0006	16#xx	Invalid function block state. 16#xx in Additional2 contains the internal number of the state machine.
INPUT_INVALID_ERROR	16#000D	16#xx	Cause: AxisRef is not in the valid range. Solution: Bring AxisRef to the valid range. 16#xx in Additional2 contains the invalid axis address.
RESSOURCE_ERROR	16#0012	16#xx	Control number is invalid. 16#xx in Additional2 contains the invalid control number.
INPUT_INVALID_ERROR	16#2B00	16#0000	Cause: Invalid combination of units and scaling of the axis Solution: Parameterize axis correctly. Valid combinations are: Linear scaling with length units in [m] or [inch] and velocity units in [m]/[min] / [inch]/[min] or [m]/[s] / [inch]/[s] or translatory scaling with unit [degrees] and velocity units in [rpm] or [rps]
RESOURCE_ERROR	16#2B01	Runtime error codes (CmpErrors.library)	Cause: When determining the handle of the task, in which the function block is called, an error occurred. Solution: Restart function block initialization with a rising edge at the "Enable" input.
RESOURCE_ERROR	16#2B02	Runtime error codes (CmpErrors.library)	Cause: When determining the properties of the task, in which the function block is called, an error occurred. Solution: Restart function block initialization with a rising edge at the "Enable" input.

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR	16#2B03	16#0000	Cause: The task, in which the function block is called, is not synchronous to the Sercos cycle. Solution: The function block has to be called in a Sercos-synchronous task. Type of "External" task, external event "EVENT_OP_MODE_SERCOS_CYCLIC" (MLC/XLC), or "CCD_SYNCHRONIZED_TASK" (MLD)
RESOURCE_ERROR	16#2B04	Additional1 of MB_ReadParameter	Cause: When reading the Sercos cycle time (Parameter C-0-0503 (MLC/XLC) or S-0-0001 (MLD)), an error occurred. Additional2 contains the error number of the error-generating function block MB_ReadParameter (MLC/XLC); MB_ReadParameter (MLD). Solution: Restart function block initialization with a rising edge at the "Enable" input.
INPUT_RANGE_ERROR	16#2B05	Maximum oversampling factor allowed	Cause: The value specified at the OversamplingFactor function block input for the oversampling rate is too high. Solution: The maximum value of the "OversamplingFactor" input is 64.
INPUT_INVALID_ERROR	16#2B06	16#0000	Cause: The value specified at the OversamplingFactor function block input for the oversampling rate is invalid. Solution: Only squares (1, 4, 8, 16...) are supported for the OversamplingFactor. Note: As the Sercos block I/O module R-ILB S3 24 DI16 DIO16/F does not support the oversampling factor 2, factor 2 is also not permitted by the function block.

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR	16#2B07	Minimum size of the output data array	Cause: The size of the output data array (input SizeOfOutputs) is too small. Solution: The oversampling factor specifies the minimum size of the output data array. The minimum size of the output data array is 8 bytes (up to the oversampling factor 8). For 8 bytes and higher, the minimum size of the output data array results from the following formula: Minimum size in bytes = Number of switch tracks (32) x Oversampling factor / 8 (number of bits in one byte). Example: In case of a 32-times oversampling, the minimum size of the output data array is $32 \times 32 / 8 = 128$ bytes.
RESOURCE_ERROR	16#00002B09	16#0000	Cause: The Motion cycle time (C-0-0400.0.0) and the Sercos cycle time (C-0-0503.0.0) have different values. Solution: Set the same values for the Motion cycle time (C-0-0400.0.0) and the Sercos cycle time (C-0-0503.0.0).
RESOURCE_ERROR	16#00002B0A	Additional1 of MB_ReadParameter	Cause: When reading the Motion cycle time (parameter C-0-0400), an error occurred. Additional2 contains the error number of the error-generating function block MB_ReadParameter. Solution: Restart function block initialization with a rising edge at the "Enable" input.

Tab. 3-40: Error codes of the MB_DigitalCamSwitchOvs function block

3.4.9 MB_CAMSWITCH_REF_OVS

Type definitions The following definition describes a data type used in the function block [MB_DigitalCamSwitchOvs](#), page 66,.

The data type "MB_CAMSWITCH_REF_OVS" contains an array of 64 basic structures each describing a single switch ([ML_BASIC_CAMSWITCH_REF_OVS](#), page 93).

Type definition

```

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

Type definitions The following definition describes a data type used in the function block [MB_DigitalCamSwitchOvs](#), page 66.

The data type [ML_BASIC_CAMSWITCH_REF_OVS](#), page 92, contains an array of 64 basic structures each describing a single switch.

Values are given in units [u]. These relate to the corresponding axis.

If e.g. a switch-on position is at 20u and the corresponding axis is scaled in degrees, the switch-on position is 20°.

The basic structure "ML_BASIC_CAMSWITCH_REF_OVS" is as follows:

Structural element	Type	Description
FirstOnPosition [u]	REAL	Lower limit at which the output is ON (switch-on position)
LastOnPosition [u]	REAL	Upper limit at which the output is ON (switch-off position)
Duration	LTIME	Coupled with time-related "CamSwitchMode". I.e. the output remains on for the specified time
TrackNumber	INT	"TrackNumber" is the reference for the switch track
AxisDirection	INT	Direction in which the programmable limit switch is effective: both (=0; default) Positive (1); Negative (2)
CamSwitchMode	INT	Position-related (=0; default); Time-related (=1)

Tab. 3-41: ML_BASIC_CAMSWITCH_REF_OVS structure

3.4.11 MB_TRACK_REF_OVS

Type definitions The following definition describes a data type used in the function block [MB_DigitalCamSwitchOvs](#), page 66.

The data type MB_TRACK_REF_OVS contains an array with 64 basic structures ([ML_BASIC_TRACK_REF_OVS](#), page 93), i.e. 64 switch tracks that can be assigned to 64 oversampling-capable outputs.

Type definition

```

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

```

Several switches can be assigned to each switch track.

3.4.12 ML_BASIC_TRACK_REF_OVS

Type definitions The following definition describes a data type used in the function block [MB_DigitalCamSwitchOvs](#), page 66.

A switch track is an element describing the characteristics of the programmable limit switch related to an output capable of oversampling.

The basic structure ML_BASIC_TRACK_REF_OVS contains elements to set the behavior for a switch track.

Structural element	Type	Description
OnCompensation	LTIME	Lead time of the switch track related to the switch-on time of a switch
OffCompensation	LTIME	Lead time of the switch track related to the switch-off time of a switch
TimeDef	WORD	Time definition: Bit 0 set: "OnCompensation" time is negative; Bit 1 set: "OffCompensation" time is negative A negative time means that the output switches earlier. A positive time (measured at the switching point in time) means that the output switches later.

Tab. 3-42: ML_BASIC_TRACK_REF_OVS structure

3.5 Cam download

3.5.1 Introduction and overview

The following function block downloads cams to the drive and the control to encapsulate the existing differences in detail. The structure [MB_CAMTABLE](#), [page 97](#), supports the function block [MB_LoadCamData](#), [page 94](#), in loading the cam tables.

The enumeration [MB_CAMPFOL](#), [page 97](#), allows also for the use of the old cam table format in order to process cam tables in the drive.

3.5.2 MB_LoadCamData

Brief description MB_LoadCamData downloads cams to the drive and control to encapsulate the existing detail differences.

 In version 14VRS, the "MC_LoadCamData (ML_TechBase library)" function block was moved to the "TechMotionBase library".

Assignment target system/library	MLC/XLC 14VRS	ML_TechMotionBase
	MOT function package	
	MLD (IL2G)	MX_TechMotionBase

Tab. 3-43: Reference table of the MB_LoadCamData

Interface description

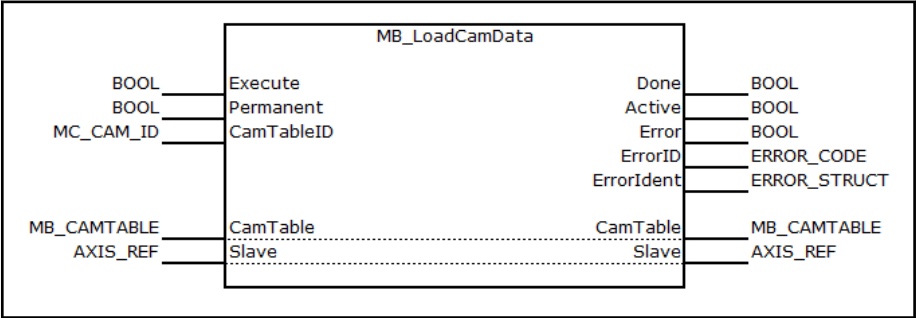


Fig. 3-41: MB_LoadCamData function block

I/O type	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Function block processing enabled (once, edge-controlled)
	Permanent	BOOL	Setting option whether the cam should be written permanently to the parameter or only temporarily to the volatile memory
	CamTableID	MC_CAM_ID	Selection of the desired cam table of the drive or control
VAR_OUTPUT	Done	BOOL	Processing completed without errors, output data valid
	Active	BOOL	Processing not yet completed, output data invalid
	Error	BOOL	Processing completed with error, output data invalid, error outputs valid
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, it contains a detailed error description (see Troubleshooting, page 96)
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements MB_CAMTABLE, page 97 ,
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Reference selection (drive or control)

Tab. 3-44: Interface variables of the MB_LoadCamData function block

Min./max. and default values

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
Permanent	BOOL			FALSE	Rising edge at "Execute"
CamTableID	MC_CAM_ID			CAM_TABLE_1	Rising edge at "Execute"

Tab. 3-45: Min./max. and default values of the MB_LoadCamData function block

The enumeration MC_CAM_ID, RIL_CommonTypes, supports the selection of the desired cam table.

Element	Value	Description
CAM_TABLE_1	1	Cam table 1
CAM_TABLE_2	2	Cam table 2
.....		
CAM_TABLE_255	255	Cam table 255

Tab. 3-46: Elements of the enumeration type MC_CAM_ID

The structure [MB_CAMTABLE, page 97](#), is provided with the "CamTable" array containing the calculated cam points and the "NumberOfElements" variable which shows the number of valid cam points of the "CamTable" array.

Functional description

When using the "MLD (IL2G)" target system, the explanations regarding cam downloads to an XLC/MLC are not relevant.

After processing enable via "Execute", the function block checks the type of format it can be written on the selected cam via "CamTableID".

The cam table in the "MB_CAMTABLE" structure is then converted to this format and loaded to the control or drive using a "WriteParameter" function block.

The selection of the memory location (control system or drive) of the cam table is determined by the "Axis" input.

With the "Permanent" input, the decision can either be made for the permanent cam storage or the temporary storage (RAM) of the device. The different types of storages are implemented on the control by two different "WriteParameter" function blocks.

The function block sets the S-0-0269 "Storage mode" parameter of the drive to the value "1"(not storing) if it is called by means of the "Permanent" input = "false".

The number of valid cam points included in the "CamTable" array can be seen via the "NumberOfElements" element of the "MB_CAMTABLE" structure. If a drive cam with less than 1,024 data points is selected via "CamTableID", but "NumberOfElements" signalizes more elements, an error is generated.

The converted cam table is transferred to the "WriteParameter" function block in case of a correct function block processing and loaded to the control or drive. The validity of the conversion and the loading is signaled by the "Done" output.

If an error occurs while processing the function block, this is signaled by the "Error" output. The elements of the "CamTable" structure and "Axis" are not updated in case of error.

Cam characteristics

The cam table of the ML_LoadCamData function block provides the following characteristic:

- The algorithm to convert the cam data points is distributed across several cycles, i.e. calls of the instance to avoid a considerable effect on the PLC cycle time

	Cam in the control		Cam in the drive	
Axis	CntrlNo = LOCAL_CNTRL AxisNo = NO_OBJECT		CntrlNo = LOCAL_CNTRL AxisNo = AXIS_1..AXIS_99	
Permanent	FALSE (temporary cam)	TRUE (permanent cam)	FALSE (temporary cam)	TRUE (permanent cam)
CamTableID	1..99 (3-1,024 data points)		1..4 (3-1,024 data points); 5..8 (3-128 data points)	

Tab. 3-47: Input assignments of the MB_LoadCamData function block to load the cam to the drive or control

Troubleshooting

The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR	16#00000006	16#00000000	Invalid function block state
INPUT_RANGE_ERROR	16#00002200	16#00000001	The "NumberOfElements" input is lower than the minimum or higher than the maximum
INPUT_INVALID_ERROR	16#00002201	16#00000001	Address of the CamTable array is zero
INPUT_INVALID_ERROR	16#00002210	16#00000001	"CamTableID" input too high to select "Axis" input
INPUT_INVALID_ERROR	16#00002210	16#00000002	"NumberOfElements" input too high for selected "CamTableID"

Tab. 3-48: Error codes of the MB_LoadCamData function block

3.5.3 MB_CAMPROF

Type definitions The enumeration selects between different cam profiles. The selections of CAMPROF_STANDARD and CAMPROF_ADDITIVE contain the old format at which the last element of the cam table is the data point $\varphi = 360^\circ - d\varphi$. The selections of CAMPROF_STANDARD and CAMPROF_ADDITIVE contain the old format at which the last element of the cam table is the data point $\varphi = 360^\circ - d\varphi$.

The format can be selected for cam tables in the drive via the parameter P-0-0086. Only the new format is available for cam tables in the control.

Element	Value	Description
CAMPROF_STANDARD	0	Cam profile without reduction, cannot be segmented (old format)
CAMPROF_ADDITIVE	1	Superimposition profile with reduction, cannot be segmented (old format)
CAMPROF_ADDITIVE_SEGMENTABLE	2	Superimposition profile with reduction, can be segmented (new format)

Tab. 3-49: Elements of the enumeration type MB_CAMPROF

3.5.4 MB_CAMTABLE

The structure is provided with the array "CamTable" containing the calculated cam points and the variable "NumberOfElements" which shows the number of valid cam points of the "CamTable" array.

Name	Type	Min. value	Max. value	Default value	Effective	Description
lrcamTable	ARRAY [0...1023] OF LREAL				Rising edge at "Execute"	Array with the calculated cam points
uiNumberOfElements	UINT	3	1024		Rising edge at "Execute"	Number of valid cam points of the "CamTable" array

Tab. 3-50: Min./max. and default values of the MB_CAMTABLE structure

3.6 PositionFiltering (filtering incremental positions)

3.6.1 Introduction and overview

The PositionFiltering block filters an incremental position and calculates position and velocity in LREAL. The position may e.g. be read in by a Sercos encoder or any other encoder.

ML_TechMotionBase.library // MX_TechMotionBase.library

1. [MB_PositionFilterType01](#), page 98, filters an incremental position and calculates position and velocity in LREAL.

3.6.2 MB_PositionFilterType01

Brief description The MB_PositionFilterType01 function block filters an incremental position and calculates position and velocity in LREAL. The position may e.g. be read in by a Sercos encoder or any other input.

In addition to processing in the PLC program, the function block outputs may also be used for an MLC controller axis (using the ML_ControlWriteActualData function block).

Assignment target system/library

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

Tab. 3-51: MB_PositionFilterType01 reference table

Interface description

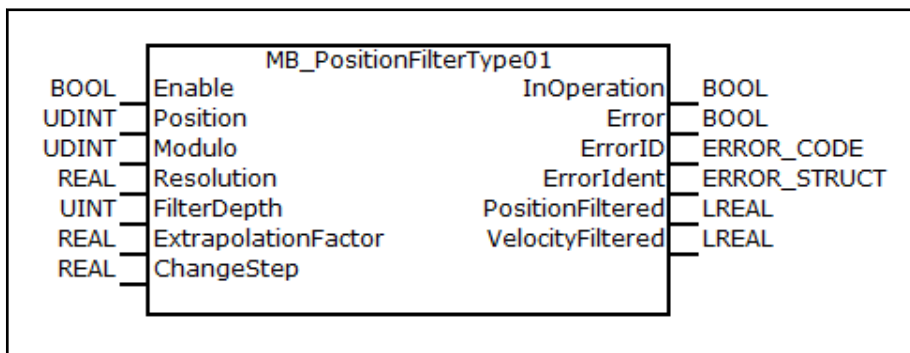


Fig. 3-42: MB_PositionFilterType01 function block

I/O type	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Processing of function block enabled (cyclic, state-controlled)
	Position	UDINT	Non-scaled position in increments. Is processed cyclically
	Modulo	UDINT	Modulo value in increments per revolution. 0 = absolute. Is latched in case of rising edge at Enable.
	Resolution	REAL	Resolution in increments per unit, e.g. Inc/degree or Inc/mm. Is latched in case of rising edge at Enable.
	FilterDepth	UINT	Filter depth. Mean value determination using N values. Is processed cyclically
	ExtrapolationFactor	REAL	Dead time compensation. Specification in number of cycles of the function block call-up interval. Is processed cyclically
	ChangeStep	REAL	Changes at the "FilterDepth" and "ExtrapolationFactor" inputs are adjusted with this step width in order to keep transition processes small. Is latched in case of rising edge at Enable.

I/O type	Name	Type	Comment
VAR_OUTPUT	InOperation	BOOL	The block processes the position cyclically.
	Error	BOOL	Processing completed with error, output data invalid, error outputs valid
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, it contains a detailed error description (see Troubleshooting, page 100)
	PositionFiltered	LREAL	Filtered position. Unit depending on the "Resolution" input e.g. in degree or mm
	VelocityFiltered	LREAL	Filtered velocity. Unit depending on the "Resolution" and "Modulo" inputs, e.g. in rpm or mm/min

Tab. 3-52: Interface variables of the MB_PositionFilterType01 function block

Min./max. and default values

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Position	UDINT			0	Continuous
Modulo	UDINT			0	Rising edge at Enable
Resolution	REAL			0 (0 is invalid)	Rising edge at Enable
FilterDepth	UINT	1	256	10	Continuous
ExtrapolationFactor	REAL	-16.0	+16.0	2.0 (e.g. for Sercos encoder)	Continuous
ChangeStep	REAL	0.01	10.0	0.1	Rising edge at Enable

Tab. 3-53: Min./max. and default values of the MB_PositionFilterType01 function block

Functional description

After processing enable by means of "Enable", the block will check the inputs for valid values. As soon as the "InOperation" output is set, the "Position" input will be cyclically converted into the outputs for Position and Velocity by means of the "Resolution" and "Modulo" values.



The function block must be called cyclically with a constant cycle time in order to supply correct values. Call-up in a motion-synchronous task is recommended. If the Sercos cycle time is less than the motion cycle time, call-up in a Sercos-synchronous task provides for advantages in the filter function.

The "FilterDepth" input determines the number of cycles over which the sliding mean value filter is calculated. The filter depth may be changed during operation, e.g. in order to parameterize the filter differently, depending on the velocity. The filter runtime is automatically compensated (extrapolated).

The dead time compensation is parameterized by the "ExtrapolationFactor" input. The value is specified in number of cycles of the function block call-up interval. The default value of two cycles is e.g. suitable for a Sercos encoder.

The "ChangeStep" input defines how the values will be accepted step by step if the "FilterDepth" and/or "ExtrapolationFactor" inputs are changed while enable is provided. A change in the "ExtrapolationFactor" input may e.g. lead to a position jump at the output. A small "ChangeStep" keeps the transition processes small, e.g. in order to protect sensitive mechanics.

With "ChangeStep" values of less than one, a change in the "FilterDepth" will be distributed to several cycles. With a "ChangeStep" of 0.1, e.g., the "FilterDepth" will be changed by 1 every tenth cycle.

The filter time adjustment will only be completed after several cycles. If the "FilterDepth" is, e.g., changed from 10 to 16 with "ChangeStep" = 0.1, the adjustment will only be completed after 60 cycles.



As example for the application of the MB_PositionFilterType01 function block, the GAT Plug-In "SercosEncoderRealMaster" "Connection of a Sercos encoder with a control axis" is available on the Bosch Rexroth-internal motion logic Wiki for download.

Mean value filter

The MB_PositionFilterType01 function block implements a sliding mean value filter with dead time compensation. The filter depth may be changed during operation.

Recommendation: set a large filter depth for standstill and low velocities. For high velocities, set the filter depth to small values.

As approximation for the filter cutoff frequency, you can use the following formula:

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

f Filter cutoff frequency in Hertz

N FilterDepth input

T_{cyc} Cycle time. Function block call-up in seconds

Fig. 3-43: Calculation of the filter limit frequency

Example with FilterDepth = 16 and cycle time = 2ms:

$f \sim \sqrt{2} / (10 * 0.002s) \sim 70.71 \text{ Hertz}$

Troubleshooting

The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR	16#0002	16#0001	"Resolution" input = 0
INPUT_INVALID_ERROR	16#0002	16#0002	"FilterDepth" input is not within the valid range
INPUT_INVALID_ERROR	16#0002	16#0003	"ExtrapolationFactor" input is not within the valid range
INPUT_INVALID_ERROR	16#0002	16#0004	"ChangeStep" input is not within the valid range
INPUT_INVALID_ERROR	16#0002	16#0005	"Resolution" (increments per degree) is less than "Modulo" (increments per revolution).

Tab. 3-54: Error codes of the MB_PositionFilterType01 FB

3.7 Position monitoring

3.7.1 Introduction and overview

The position monitoring function blocks monitor the positions of two axes.

Several function block versions are available:

1. [MB_PositionMonitoring_GantryCant](#), [page 101](#), monitoring of the relative position of two axes with regard to tilting.
2. [MB_PositionMonitoring_PosCollision](#), [page 103](#), monitoring of the absolute and relative positions of two axes with regard to indentation and collision.

3.7.2 MB_PositionMonitoring_GantryCant

Brief description The function block "MB_PositionMonitoring_GantryCant" monitors the relative position of two the axes with regard to tilting.

The retract motion is possible using the tolerance band functionality. The direction of motion out of the tilting is continuously monitored.



In version 14VRS, the "MB_PositionMonitoring_GantryCant (ML_TechMotionBase library)" function block was moved to the "TechMotionBase library".

The function block can be used to monitor:

- Infeed axes of Flexo printing units
- Gantry axes with mechanically parallel travel distances, etc.

Assignment target system/library

Target system	Library
MLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-55: Reference table of the MB_PositionMonitoring_GantryCant

Interface description

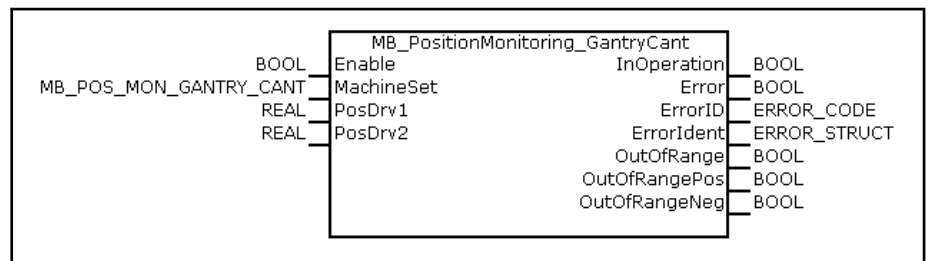


Fig. 3-44: MB_PositionMonitoring_GantryCant function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Processing of function block enabled (cyclic, state-controlled)
	MachineSet	MB_POS_MON_GANTRY_CANT	Machine-specific characteristics MB_POS_MON_GANTRY_CANT , page 102
	PosDrv1	REAL	Current position of axis 1
	PosDrv2	REAL	Current position of axis 2
VAR_OUTPUT	InOperation	BOOL	Monitoring the axes completed without errors; output signals valid
	Error	BOOL	Monitoring the axes completed with errors, output variable "ErrorIdent" valid
	ErrorID	ERROR_CODE	Diagnostic description in the event of an error

I/O type	Name	Data type	Description
	ErrorIdent	ERROR_STRUCT	Diagnostic description in the event of an error See Troubleshooting, page 102
	OutOfRange	BOOL	Permissible range is exceeded
	OutOfRangePos	BOOL	Permissible range is exceeded. Axis position of axis 1 is higher than or equal to the axis position of axis 2
	OutOfRangeNeg	BOOL	Permissible range is exceeded. Axis position of axis 1 is lower than the axis position of axis 2

Tab. 3-56: Interface variables of the MB_PositionMonitoring_GantryCant function block

Functional description The function block "MB_PositionMonitoring_GantryCant" cyclically monitors the positions of two axes with regard to tilting after enabling the process via "Enable". If the distance of the axes exceeds the maximum "range" allowed, this is signaled via the outputs "OutOfRange" and "OutOfRangePos" or "OutOfRangeNeg".

In the following cycle, the output "OutOfRange" is reset. The outputs "OutOfRangePos" or "OutOfRangeNeg" continue to signal the requirement of a position adjustment until reaching the admissible distance minus the tolerance.

If the maximum distance reached plus the tolerance is exceeded again or even more during this position adjustment in this period, this is signaled by setting the output "OutOfRange" again.

If an error occurs during the function block processing, this is signaled via the output "Error" and, in this case, specified by the updated elements of the output structure "ErrorIdent". The other outputs are not updated in the event of an error.

Error codes In the event of an error, the "MB_PositionMonitoring_GantryCant" function block generates a comprehensive diagnostics which can be locally interpreted.

Troubleshooting The function block uses the error table **F_RELATED_TABLE**, 16#0170. It can generate the following error messages in Additional1 and Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Inputs are not in the valid range
STATE_MACHINE_ERROR (16#0005)	16#0006	16#????	Invalid function block state

Tab. 3-57: Error codes of the MB_PositionMonitoring_GantryCant function block

3.7.3 MB_POS_MON_GANTRY_CANT

The data type MB_POS_MON_GANTRY_CANT represents the machine set of the function block MB_PositionMonitoring_GantryCant.

The structure contains the following data:

Structure element	Type	Min. value	Max. value	Default value	Description
Range	REAL	$\geq 2 \cdot \text{Tolerance}$	n.def.	10.0	Maximum position difference allowed between the two axes
Tolerance	REAL	0.0	n.def.	5.0	Tolerance of the limits

Tab. 3-58: Data structure of the MB_POS_MON_GANTRY_CANT

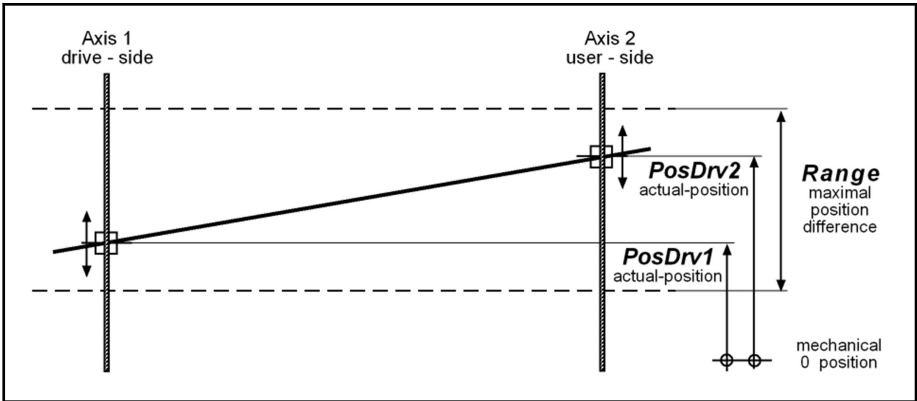


Fig. 3-45: Printing unit top view – Maximum position difference allowed (range)

The specified tolerance has to fulfill the following condition:

$$Range > 2 * Tolerance$$

Fig. 3-46: Condition machine set type 01

3.7.4 MB_PositionMonitoring_PosCollision

Brief description The function block "MB_PositionMonitoring_PosCollision" monitors the absolute position and the relative position of two axes with regard to indentation and collision.

 In version 14VRS, the "MB_PositionMonitoring_PosCollision" (ML_TechMotion library) function block was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-59: Reference table of the MB_PositionMonitoring_PosCollision

Interface description

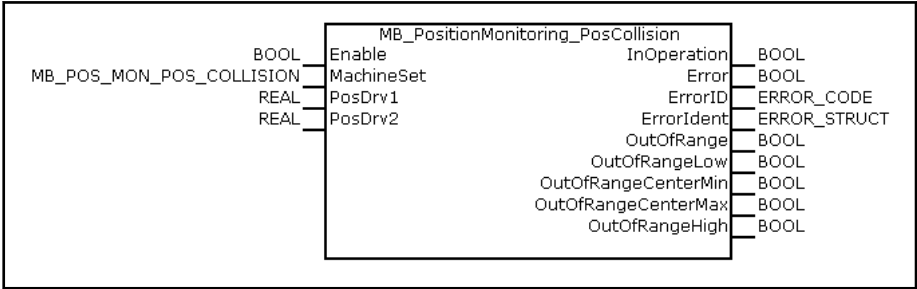


Fig. 3-47: MB_PositionMonitoring_PosCollision function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Processing of function block enabled (cyclic, state-controlled)
	MachineSet	MB_POS_MON_POS_COLLISION	Machine-specific characteristics MB_POS_MON_POS_COLLISION , page 105
	PosDrv1	REAL	Current position of axis 1
	PosDrv2	REAL	Current position of axis 2
VAR_OUTPUT	InOperation	BOOL	Monitoring the axes completed without errors; output signals valid
	Error	BOOL	Monitoring the axes completed with errors, output variable "ErrorIdent" valid
	ErrorID	ERROR_CODE	Diagnostic description in the event of an error
	ErrorIdent	ERROR_STRUCT	Diagnostic description in the event of an error See Error codes , page 105
	OutOfRange	BOOL	Permissible range is exceeded
	OutOfRangeLow	BOOL	Permissible range is exceeded ("DeltaLow")
	OutOfRangeCenterMin	BOOL	Permissible range is exceeded ("DeltaCenterMin")
	OutOfRangeCenterMax	BOOL	Permissible range is exceeded ("DeltaCenterMax")
	OutOfRangeHigh	BOOL	Permissible range is exceeded ("DeltaHigh")

Tab. 3-60: Interface variables of the MB_PositionMonitoring_PosCollision function block



The following information applies to the "MB_PositionMonitoring_PosCollision" function block:

- The command position of axis 2 results from the parameters of machine set 2
- The current position of axis 1 always has to be higher than the sum of the current position of axis 2 and the radii of both axes
- The function block can be cascaded for three or more axes

Functional description

The function block "MB_PositionMonitoring_PosCollision" cyclically monitors the positions of two axes with regard to indentation and collision after having enabled processing with the help of "Enable". If the distance of the axes exceeds a maximum admissible range, this is signaled via the outputs "OutOfRange" and the corresponding ranges "OutOfRangeLow", "OutOfRangeCenterMin", "OutOfRangeCenterMax" and/or "OutOfRangeHigh".

In the following cycle, the output "OutOfRange" is reset. The other outputs continue to signal the requirement of a position adjustment until reaching the respective admissible distance minus the tolerance.

If the maximum distance reached plus the tolerance is exceeded again or even more during this position adjustment in this period, this is signaled by setting the output "OutOfRange" again.

If an error occurs during the function block processing, this is signaled via the output "Error" and, in this case, specified by the updated elements of the output structure "ErrorIdent". The other outputs are not updated in the event of an error.

Error codes In case of error, the function block MB_PositionMonitoring_PosCollision generates a comprehensive analysis which can be locally interpreted.

The function block uses the error table "F-RELATED-TABLE", 16#0170.

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Inputs are not in the valid range
STATE_MACHINE_ERROR	16#0006	16#????	Inputs are not in the valid range
CALCULATION_ERROR	16#000C	16#0000	Calculation error, wrong axis position

Tab. 3-61: Error codes of the MB_PositionMonitoring_PosCollision function block

3.7.5 MB_POS_MON_POS_COLLISION

Data types The data type MB_POS_MON_POS_COLLISION represents the machine set of the function block MB_PositionMonitoring_PosCollision.

The structure contains the following data:

Name	Type	Min. value	Max. value	Default value	Description
Position1	REAL	n.def.	n.def.	50.0	Command position of axis 1
Diameter1	REAL	0.0	n.def.	20.0	Diameter of axis 1
Diameter2	REAL	0.0	n.def.	20.0	Diameter of axis 2
DeltaLow	REAL	0.0	n.def.	5.0	Limitation of the position offset of axis 2
DeltaCenter-Min	REAL	0.0	$\leq \Delta\text{Center-Max} - 2 \cdot \text{Tolerance}$	2.0	Minimum distance between the axes 1 and 2
DeltaCenter-Max	REAL	$\geq 2 \cdot \text{Tolerance}$	n.def.	5.0	Maximum distance between the axes 1 and 2
DeltaHigh	REAL	$\geq 2 \cdot \text{Tolerance}$	n.def.	5.0	Limitation of axis position 2
Tolerance	REAL	0.0	n.def.	1.0	Tolerance of the limits

Tab. 3-62: Data structure of the MB_POS_MON_POS_COLLISION

The specified tolerance has to fulfill the following condition:

$$\text{Parameters} > 2 \cdot \text{Tolerance}$$

Fig. 3-48: Conditions for the machine set PosCollision

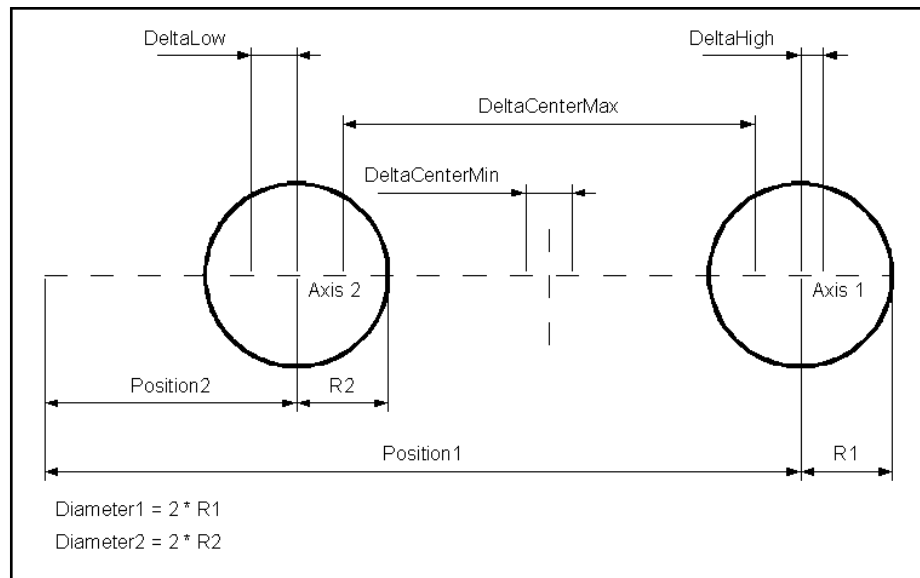


Fig. 3-49: Machine set of the MB_PositionMonitoring_PosCollision function block

3.8 S-curve, programming motion sequences

3.8.1 Introduction and overview

The function block [MB_ReachVelocityType02](#), page 106, calculates the respective jerk, acceleration and target velocity to travel along the specified jerk-limited velocity characteristics (e.g. S curve) and allows up to three changes in the target velocities.

3.8.2 MB_ReachVelocityType02

Brief description Calculates the respective jerk, acceleration and target velocity to travel along specified jerk-limited velocity characteristics (e.g. S-curve) and allows up to three changes in the target velocities

Therefore, a jerk limitation from 0% to 100 % can be selected. If the input value is 0, the velocity is linear. If it is 100%, it is shaped as S-curve. The command values for velocity, acceleration and jerk are calculated from the acceleration period, the current velocity, the target velocity and the JerkFactor. The acceleration phase can be stopped via "StopAcc".

The target velocity can be changed during the acceleration. New command value for velocity, acceleration and jerk are calculated to reach the new target velocity. A new jerk is always effective in the remaining time. Thus, the created JerkFactor becomes invalid after a change in the target velocity. The complete acceleration period changes only if the new target velocity is very similar to the current velocity.



In version 14VRS, the "ReachVelocityType02 (ML_TechMotion library)" function block was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-63: Reference table of the MB_ReachVelocityType02

Interface description

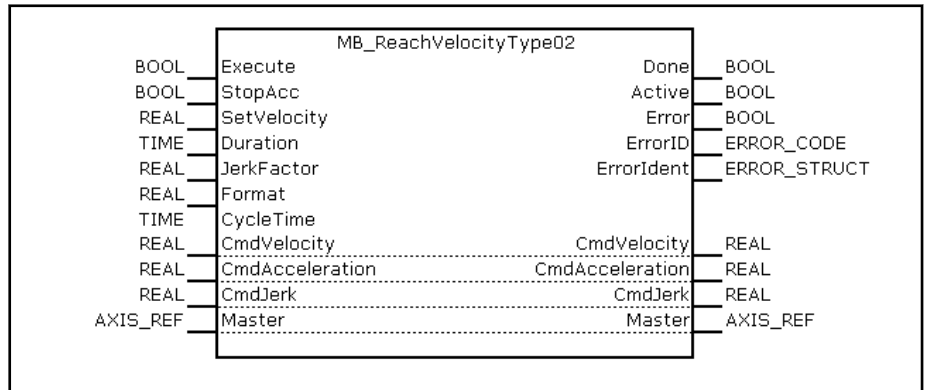


Fig. 3-50: MB_ReachVelocityType02 function block

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Processing of function block enabled
	StopAcc	BOOL	Stops further acceleration
	SetVelocity	REAL	Target velocity [m/min]
	Duration	TIME	Acceleration time
	CycleTime	TIME	Cycle time of the task in which the function block is called
	JerkFactor	REAL	Ramping the acceleration profile [%]
	Format	REAL	Format [mm]
	Master	AXIS_REF	Reference to the master axis
VAR_OUTPUT	Done	BOOL	Processing completed. Data outputs are valid
	Active	BOOL	Function block is working and currently in an interim state
	Error	BOOL	Indicates an error
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 110
VAR_IN_OUT	CmdVelocity	REAL	Command velocity [rpm]
	CmdAcceleration	REAL	Command acceleration [rad/s ²]
	CmdJerk	REAL	Command jerk [rad/s ³]
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Reference to the master axis It is recommended to create a virtual master axis

Tab. 3-64: Interface description of the MB_ReachVelocityType02

Min./max. and default values

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
StopAcc	BOOL			FALSE	Continuous
SetVelocity	REAL	n.def.	n.def.	0	Continuous between a rising edge at "Execute" and Done = TRUE
Duration	TIME	> 0 s	n.def.	30 s	rising edge at Execute
JerkFactor	REAL	0	100	50	rising edge at Execute
Format	REAL	> 0	n.def.	1000	rising edge at Execute
CycleTime	TIME	> 1 ms	n.def.	10 ms	rising edge at Execute

Tab. 3-65: *Input behavior of the MB_ReachVelocityType02*

Functional description

The acceleration phase starts at a rising edge at "Execute". The acceleration period (duration), the web velocity to be set (SetVelocity) as well as the acceleration profile (JerkFactor) are applied. A special feature is that "Active" is set at a rising edge at "Execute" but that the acceleration phase starts only after one second. That means that the following function blocks (e.g. the register control) can use "Active" to see the beginning of the acceleration phase. After the target velocity has been reached, "Active" is also present for one more second until the processing is completed with "Done".

The acceleration duration (Duration), the web velocity to be traveled to (SetVelocity) and the acceleration profile (JerkFactor) are specified for a virtual master axis, since the axis to be referenced at the master has to be a virtual axis to ensure the desired precision.

To represent the curve characteristics at a real axis, it has to be defined as slave axis synchronously to the virtual master axis.

S-curve

The jerk is the temporal derivation of the acceleration. The JerkFactor indicates the temporal part of the acceleration phase in percent in which the jerk is not zero.

The following figure shows three acceleration phases:

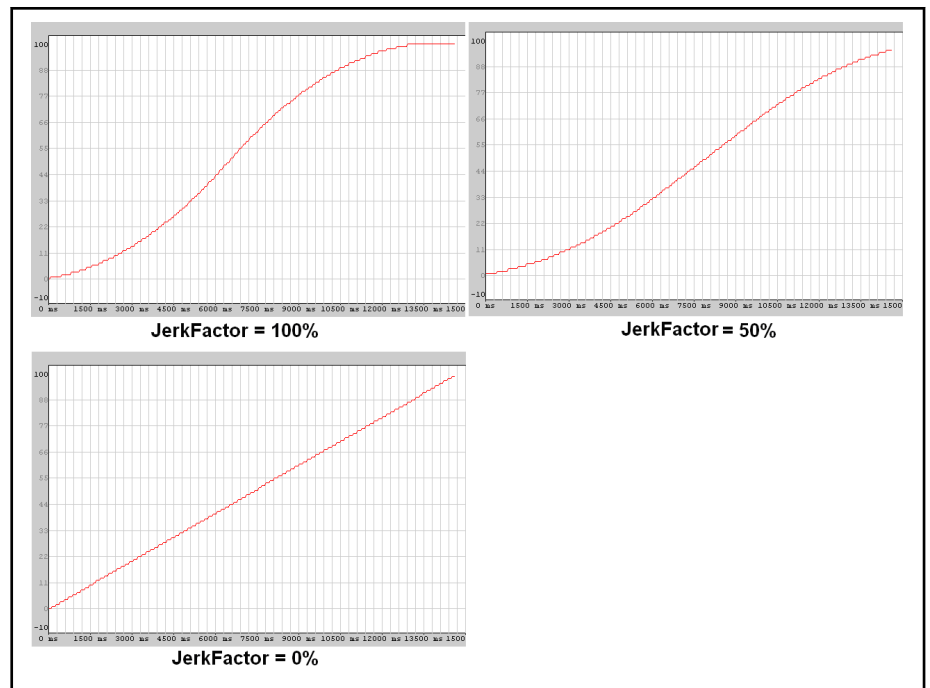


Fig. 3-51: Acceleration phases with different JerkFactor

Changes at SetVelocity

After a target velocity change before reaching the end velocity, "CmdVelocity", "CmdAcceleration" and "CmdJerk" are calculated again by assuming the new target velocity and measuring the current time, velocity and acceleration. The function block allows the target velocity to be modified three times.

The following figure shows the master axis speed in rpm of an acceleration phase from 0 to 200 m/min (SetVelocity) in 30 seconds (Duration) and a format of 1 m with a JerkFactor of 30%. If the velocity is higher than 50 m/min, "SetVelocity" is changed to 300 m/min. After the change, the jerk is calculated again and the remaining acceleration phase shows an S-curve characteristic. The original JerkFactor is not considered anymore after the first change.

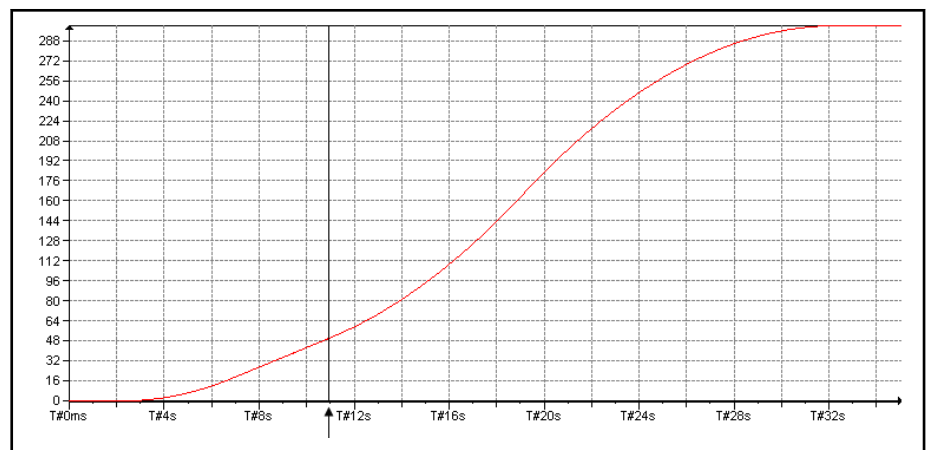


Fig. 3-52: Effects of target velocity changes

Stopping the acceleration

The acceleration phase triggered with "Execute" can be immediately stopped with "StopAcc". The function block immediately stops further acceleration. The "Done" output is set. If the set velocity value "SetVelocity" should still be approached, "StopAcc" has to be deactivated and another rising edge of "Execute" has to be set. The reaching of the "SetVelocity" value is signaled by the "Done" output until a falling edge at "Execute".

The following figure shows the abort of the linear and s-curve characteristic acceleration phase via "StopAcc":

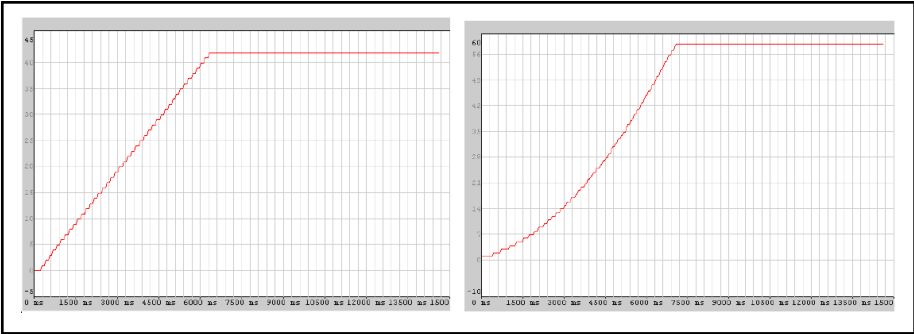


Fig. 3-53: Aborting the acceleration phases (left an abort at 6300 ms, right an abort at 7000 ms)

Troubleshooting The function block generates the following error messages in Additional1/ Additional2 for the table **F_RELATED_TABLE**, 16#0170:

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Error in the state machine
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000000	Invalid printing cylinder format
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000023	Invalid cycle time
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000024	Invalid period
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000025	Invalid jerk factor
INPUT_INVALID_ERROR (16#0001)	16#0000000D	16#00000000	"AxisNo" for "Master" is not in the valid range

Tab. 3-66: Error codes of MB_ReachVelocityType02

3.9 Probe function blocks

3.9.1 Introduction and overview

The MB_InitTouchProbe, MB_TouchProbe, MB_WriteExpectWindow and MB_TouchProbeContinuous function blocks control and manage the probe functionality supported by the Bosch Rexroth IndraDrive drives.

The touch probe functionality of the control hardware is provided via the following function blocks of the ML_PLCOpen library:

Function block	Description
ML_InitTouchProbe	Enables and configures the touch probe in the control
ML_WriteExpectWindow	Configures the expectation window for the touch probe functionality
ML_TouchProbe	Enables the selected touch probe, analyzes the status and provides the measured values after the trigger event

Function block	Description
ML_TouchProbeContinuous	Enables the selected touch probe, analyzes the status and provides the measured values after the trigger event
ML_AbortTrigger	Cancels an active measurement

Tab. 3-67: Function blocks for the touch probe functionality of the control hardware

The probe functionality of the Rexroth IndraDrive drives is provided via the following function blocks of the ML_TechMotionBase library:

Function block	Description
MB_InitTouchProbe, page 120	Enables and configures the touch probe in the drive
MB_WriteExpectWindow, page 134	Configures the expectation window for the touch probe functionality
MB_TouchProbe, page 128	Enables the selected touch probe, analyzes the status and provides the measured values once after the trigger event
MB_TouchProbeContinuous, page 131	Enables the selected touch probe, analyzes the status and provides the measured values continuously after the trigger event
MB_AbortTrigger, page 137	Cancels an active measurement
Data type	
MB_PROBE_EXECUTION_MODE, page 128	Selection whether single-shot or continuous measurement

Tab. 3-68: Function blocks for the touch probe functionality of the Rexroth IndraDrive

The following documentation describes the function blocks for the touch probe functionality of the Rexroth IndraDrive. The function blocks for the touch probe functionality of the control are described in the documentations "MLC 14VRS Functional Description", "Rexroth IndraLogic XLC 14VRS Functional Description" and "MLC/XLC PLCopen Function Blocks and Data Types".

The IndraDrive touch probe functionality permits the detection of positions or of the internal system time in [us] for the instance of a fast, digital input signal. The touch probe function block uses this drive functionality and provides a user interface. The following steps are required to use the touch probe function blocks:

3.9.2 Touch Probe Function Blocks - Components and Parameterization Hardware

The following Bosch Rexroth hardware components are required:

- MLC/XLC
- IndraDrive C or IndraDrive M with advanced or basic performance

The following controls are supported:

- ADVANCED (type code: CSH01.XC-...)
- BASIC sercos (single-axis; type code: CSB01.XN-SE-...)

ML_TechMotionBase.library // MX_TechMotionBase.library

- BASIC PROFIBUS (single-axis; type code: CSB01.XN-PB-...)
- BASIC UNIVERSAL (single-axis; type code: CSH01.XC-...)

Firmware

The following firmware is required and used with the above mentioned Bosch Rexroth hardware components:

- Control firmware 14VRS
- Drive firmware MPx 07VRS or higher

Parameterizing the Touch Probe Function Blocks for MLC/XLC and MLD

The following drive parameterization steps are required in order to use the MB_InitTouchProbe, MB_TouchProbe, and MB_TouchProbeContinuous technology function blocks:

MLC/XLC configuration

The following drive parameter settings are required when using the touch probe functionality in an MLC/XLC:

Open IndraWorks and go online with the relevant project.

- Include the configured touch probe control bits in the signal control word.
- **IndraWorks ► Project ► Motion ► Rea Axes ► Right click on axis: Communication ► Signal control word**
 - Touch probe 1: S-0-0405, bit 0
 - Touch probe 2: S-0-0406, bit 0

Bit	Status	Source Parameter	Bit number
Bit 0	1	S-0-0346: Positioning control word	0
Bit 1	0	S-0-0346: Positioning control word	3
Bit 2	0	S-0-0346: Positioning control word	4
Bit 3	0	S-0-0346: Positioning control word	5
Bit 4	0	S-0-0393: Command value mode	0
Bit 5	0	S-0-0393: Command value mode	1
Bit 6	0	P-0-0088: Control word for synchronous operation mod	0
Bit 7	0	P-0-0088: Control word for synchronous operation mod	1
Bit 8	0	P-0-0155: Synchronization mode	1
Bit 9	0	P-0-0154: Synchronization direction	0
Bit 10	0	P-0-0154: Synchronization direction	1
Bit 11	0	S-0-0000: <reserved>	0
<axis>.UserCmdDataBitA	0	S-0-0405: Probe 1 enable	0
<axis>.UserCmdDataBitB	0	S-0-0000: <empty>	0
<axis>.UserCmdDataBitC	0	S-0-0000: <empty>	0
<axis>.UserCmdDataBitD	0	S-0-0000: <empty>	0

Fig. 3-54: Signal control word

- Include the configured touch probe status bits in the signal status word.
 IndraWorks ► Project ► Motion ► Rea Axes ► Right click on axis: Communication ► Signal status word
 - Touch probe 1, positively detected: S-0-0409, bit 0
 - Touch probe 1, negatively detected: S-0-0410, bit 0
 - Touch probe 2, positively detected: S-0-0411, bit 0
 - Touch probe 2, negatively detected: S-0-0412, bit 0

	Status	Source Parameter	Bit-No.
Bit 0	1	S-0-0330: Message 'n_actual = n_command'	0
Bit 1	1	S-0-0331: Status 'n_feedback = 0'	0
Bit 2	0	S-0-0437: Positioning status word	2
Bit 3	0	S-0-0403: Position feedback value status	1
Bit 4	0	S-0-0419: Positioning command acknowledge	0
Bit 5	0	P-0-0089: Status word for synchronous operating mode	0
Bit 6	0	P-0-0089: Status word for synchronous operating mode	1
Bit 7	1	P-0-0089: Status word for synchronous operating mode	8
Bit 8	0	S-0-0824: Message torque/force command value reac	0
Bit 9	0	P-0-0089: Status word for synchronous operating mode	5
Bit 10	1	S-0-0336: Message In position	0
Bit 11	1	P-0-0089: Status word for synchronous operating mode	6
<axis>.UserActualDataBitA	0	S-0-0409: Probe 1 positive latched	0
<axis>.UserActualDataBitB	0	S-0-0410: Probe 1 negative latched	0
<axis>.UserActualDataBitC	0	S-0-0000: <empty>	0
<axis>.UserActualDataBitD	0	S-0-0000: <empty>	0

Apply

Fig. 3-55: Signal status word

- Include the configured touch probe values in the optional cyclic AT telegram.

IndraWorks ► Project ► Motion ► Rea Axes ► Right click on axis: Communication ► Cyclic sercos Channel

- Touch probe 1, positive edge: S-0-0130
- Touch probe 1, negative edge: S-0-0131
- Touch probe 2, positive edge: S-0-0132
- Touch probe 2, negative edge: S-0-0133
- Touch probe 1, number of marker failures: P-0-0224
- Touch probe 2, number of marker failures: P-0-0225
- Touch probe 1, difference value of the touch probe: P-0-0202
- Touch probe 2, difference value of the touch probe: P-0-0203

Configuration cyclic write access (MDT)			
<axis>.UserCmdDataA	S-0-0000: <empty>		
<axis>.UserCmdDataB	S-0-0000: <empty>		
<axis>.UserCmdDataC	S-0-0000: <empty>		
<axis>.UserCmdDataD	S-0-0000: <empty>		

Configuration cyclic read access (AT)			
<axis>.UserActualDataA	S-0-0130: Probe value 1 positive edge	224,9635	Deg
<axis>.UserActualDataB	S-0-0131: Probe value 1 negative edge	0,0000	Deg
<axis>.UserActualDataC	P-0-0202: Difference probe values 1	135,0365	Deg
<axis>.UserActualDataD	P-0-0224: Probe 1, number of marker failures	0	..

Fig. 3-56: Configuring a cyclic AT telegram

Configuring the MLD-S or MLD-M (master axis)

The following drive parameter settings are required to use the touch probe functionality as master axis in a MLD-S or MLD-M:

- Include the configured touch probe control bits in the signal control word.

IndraWorks ► Project ► MLD ► Right click on: AxisData

- Touch probe 1: S-0-0405, bit 0

- Touch probe 2: S-0-0406, bit 0

Axis #1 (local axis)

Command values

wUserCmdDataBitA_q	S-0-0405 : Probe 1 enable	0
wUserCmdDataBitB_q	S-0-0406 : Probe 2 enable	0
wUserCmdDataBitC_q	S-0-0000 : Free	0
wUserCmdDataBitD_q	S-0-0000 : Free	0
dwUserCmdDataA_q	S-0-0000 : Free	
dwUserCmdDataB_q	S-0-0000 : Free	
dwUserCmdDataC_q	S-0-0000 : Free	
dwUserCmdDataD_q	S-0-0000 : Free	

Actual values

wUserActualDataBitA_i	S-0-0409 : Probe 1 positive latched	0
wUserActualDataBitB_i	S-0-0410 : Probe 1 negative latched	0
wUserActualDataBitC_i	S-0-0000 : Free	0
wUserActualDataBitD_i	S-0-0000 : Free	0
dwUserActualDataA_i	S-0-0130 : Probe value 1 positive edge	
dwUserActualDataB_i	S-0-0131 : Probe value 1 negative edge	
dwUserActualDataC_i	P-0-0202 : Difference probe values 1	
dwUserActualDataD_i	P-0-0224 : Probe 1, number of marker failures	

Apply Configuration

Fig. 3-57: Command value data bits

- Include the configured touch probe status bits in the signal status word.
 - IndraWorks** ► **Right click on sercos III** ► **Cross Communication Drives** ► **Signal Status Word**
 - Touch probe 1, positively detected: S-0-0409, bit 0
 - Touch probe 1, negatively detected: S-0-0410, bit 0
 - Touch probe 2, positively detected: S-0-0411, bit 0
 - Touch probe 2, negatively detected: S-0-0412, bit 0

Bit	Status	Source parameter	Source bit
0	0	P-0-0115 : Device control: status word	12
1	0	S-0-0331 : Status 'n_feedback = 0'	0
2	0	S-0-0403 : Position feedback value status	0
3	0	S-0-0419 : Positioning command acknowledge	0
4	0	S-0-0521 : Status word of position loop	0
5	0	S-0-0521 : Status word of position loop	2
6	0	P-0-0152 : Synchronization completed	0
7	0	P-0-0115 : Device control: status word	7
8	0	S-0-0409 : Probe 1 positive latched	0
9	0	S-0-0410 : Probe 1 negative latched	0
10	0	S-0-0411 : Probe 2 positive latched	0
11	0	S-0-0412 : Probe 2 negative latched	0
12	0	S-0-0409 : Probe 1 positive latched	0
13	0	S-0-0410 : Probe 1 negative latched	0
14	0	S-0-0411 : Probe 2 positive latched	0
15	0	S-0-0412 : Probe 2 negative latched	0

Bits 12 to 15 correspond to the elements
wUserActualDataBitA_i to wUserActualDataBitD_i in the structure MX_AXISDATA

Settings CCD signal control word Apply Configuration

Fig. 3-58: Actual value data bits

- Include the configured touch probe values in the optional cyclic AT telegram.

IndraWorks ► Project ► MLD ► Right click on: AxisData

- Touch probe 1, positive edge: S-0-0130
- Touch probe 1, negative edge: S-0-0131
- Touch probe 2, positive edge: S-0-0132
- Touch probe 2, negative edge: S-0-0133
- Touch probe 1, number of marker failures: P-0-0224
- Touch probe 2, number of marker failures: P-0-0225
- Touch probe 1, difference value of the touch probe: P-0-0202
- Touch probe 2, difference value of the touch probe: P-0-0203
- Include the following parameters for each configured, measured touch probe signal in the actual values data container:
 - TP_ACT_POS_FEEDBACK1 => No additional data is required
 - TP_ACT_POS_FEEDBACK2 => No additional data is required
 - TP_ACT_POSIN_ACT_CYCLE => No additional data is required
 - TP_SYNC_CMD_POS => P-0-0779
 - TP_PROBE_TIME => No additional data is required
 - 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

Axis #1 (local axis)

Command values

wUserCmdDataBitA_q	S-0-0405 : Probe 1 enable	0
wUserCmdDataBitB_q	S-0-0406 : Probe 2 enable	0
wUserCmdDataBitC_q	S-0-0000 : Free	0
wUserCmdDataBitD_q	S-0-0000 : Free	0
dwUserCmdDataA_q	S-0-0000 : Free	
dwUserCmdDataB_q	S-0-0000 : Free	
dwUserCmdDataC_q	S-0-0000 : Free	
dwUserCmdDataD_q	S-0-0000 : Free	

Actual values

wUserActualDataBitA_i	S-0-0409 : Probe 1 positive latched	0
wUserActualDataBitB_i	S-0-0410 : Probe 1 negative latched	0
wUserActualDataBitC_i	S-0-0000 : Free	0
wUserActualDataBitD_i	S-0-0000 : Free	0
dwUserActualDataA_i	S-0-0130 : Probe value 1 positive edge	
dwUserActualDataB_i	S-0-0131 : Probe value 1 negative edge	
dwUserActualDataC_i	P-0-0202 : Difference probe values 1	
dwUserActualDataD_i	P-0-0224 : Probe 1, number of marker failures	

Apply Configuration

Fig. 3-59: Command value data bits

- Set P-0-1367, bit 6 (activate update of the axis data structure, only MLD-M)

Configuring the MLD-M (slave axis)

The following drive parameter settings are required when using the touch probe functionality as slave axis in a MLD-M:

- Include the configured touch probe control bits in the signal control word.

IndraWorks ► Right click on sercos III ► Cross Communication Drives ► Signal Control Word

- Touch probe 1: S-0-0405, bit 0
- Touch probe 2: S-0-0406, bit 0

Bit	Status	Source parameter	Source bit
0	0	S-0-0520 : Control word of axis controller	0
1	0	S-0-0520 : Control word of axis controller	2
2	0	P-0-0155 : Synchronization mode	1
3	0	P-0-0154 : Synchronization direction	0
4	0	P-0-0154 : Synchronization direction	1
5	0	S-0-0000 : Free	0
6	0	S-0-0000 : Free	0
7	0	S-0-0000 : Free	0
8	0	S-0-0000 : Free	0
9	0	S-0-0000 : Free	0
10	0	S-0-0000 : Free	0
11	0	S-0-0000 : Free	0
12	0	S-0-0405 : Probe 1 enable	0
13	0	S-0-0406 : Probe 2 enable	0
14	0	S-0-0000 : Free	0
15	0	S-0-0000 : Free	0

Bits 12 to 15 correspond to the elements
wUserCmdDataBitA_q to wUserCmdDataBitD_q in the structure MX_AXISDATA

Settings CCD signal status word Apply Configuration

Fig. 3-60: Signal control word

- Include the configured touch probe status bits in the signal status word.
IndraWorks ► Right click on sercos III ► Cross Communication Drives ► Signal Status Word
 - Touch probe 1, positively detected: S-0-0409, bit 0
 - Touch probe 1, negatively detected: S-0-0410, bit 0
 - Touch probe 2, positively detected: S-0-0411, bit 0
 - Touch probe 2, negatively detected: S-0-0412, bit 0

ML_TechMotionBase.library // MX_TechMotionBase.library

Bit	Status	Source parameter	Source bit
0	0	P-0-0115 : Device control: status word	12
1	0	S-0-0331 : Status 'h_feedback = 0'	0
2	0	S-0-0403 : Position feedback value status	0
3	0	S-0-0419 : Positioning command acknowledge	0
4	0	S-0-0521 : Status word of position loop	0
5	0	S-0-0521 : Status word of position loop	2
6	0	P-0-0152 : Synchronization completed	0
7	0	P-0-0115 : Device control: status word	7
8	0	S-0-0409 : Probe 1 positive latched	0
9	0	S-0-0410 : Probe 1 negative latched	0
10	0	S-0-0411 : Probe 2 positive latched	0
11	0	S-0-0412 : Probe 2 negative latched	0
12	0	S-0-0409 : Probe 1 positive latched	0
13	0	S-0-0410 : Probe 1 negative latched	0
14	0	S-0-0411 : Probe 2 positive latched	0
15	0	S-0-0412 : Probe 2 negative latched	0

Bits 12 to 15 correspond to the elements
wUserActualDataBitA_i to wUserActualDataBitD_i in the structure MX_AXISDATA

Settings CCD signal control word Apply Configuration

Fig. 3-61: Signal status word

- Include the configured touch probe values in the optional cyclic AT telegram.

IndraWorks ► Right click on sercos III ► Cross Communication Drives ► Cyclic sercos Channel

- Touch probe 1, positive edge: S-0-0130
- Touch probe 1, negative edge: S-0-0131
- Touch probe 2, positive edge: S-0-0132
- Touch probe 2, negative edge: S-0-0133
- Touch probe 1, number of marker failures: P-0-0224
- Touch probe 2, number of marker failures: P-0-0225
- Touch probe 1, difference value of the touch probe: P-0-0202
- Touch probe 2, difference value of the touch probe: P-0-0203
- Include the following parameters in the optional cyclic AT for each configured, measured touch probe signal:
 - TP_ACT_POS_FEEDBACK1 => No additional data is required
 - TP_ACT_POS_FEEDBACK2 => No additional data is required
 - TP_ACT_POSIN_ACT_CYCLE => No additional data is required
 - TP_SYNC_CMD_POS => P-0-0779
 - TP_PROBE_TIME => No additional data is required
 - 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

Fig. 3-62: Configuring a cyclic AT telegram

- Set P-0-1367, bit 6 (activate update of the axis data structure, only MLD-M)

Configuration: Probe1

Touch probe 1	AT container	Control bit	Status bit
Positive edge	S-0-0130	S-0-0405.0	S-0-0409.0
Negative edge	S-0-0131	S-0-0405.0	S-0-0410.0
Positive and negative edge, difference, marker failure	S-0-0130, S-0-0131, P-0-0224 and P-0-0202	S-0-0405.0	S-0-0409.0, S-0-0410.0

Tab. 3-69: Configuration: Probe1

Configuration: Probe2

Probe2	AT container	Control bit	Status bit
Positive edge	S-0-0132	S-0-0406.0	S-0-0411.0
Negative edge	S-0-0133	S-0-0406.0	S-0-0412.0
Positive and negative edge, difference, marker failure	S-0-0132, S-0-0133, P-0-0225 and P-0-0203	S-0-0406.0	S-0-0411.0, S-0-0412.0

Tab. 3-70: Configuration: Probe2

Example configuration of the signal status word, signal control word, and cyclic sercos channel

- IndraDrive Probe1
 - Negative edge
 - Expectation window disabled
 - Marker failure disabled
 - Difference measurement disabled
1. Configuration of a signal control word

	Status	Source Parameter	Bit number
Bit 0	1	S-0-0346: Positioning control word	0
Bit 1	0	S-0-0346: Positioning control word	3
Bit 2	0	S-0-0346: Positioning control word	4
Bit 3	0	S-0-0346: Positioning control word	5
Bit 4	0	S-0-0393: Command value mode	0
Bit 5	0	S-0-0393: Command value mode	1
Bit 6	0	P-0-0088: Control word for synchronous operation mod	0
Bit 7	0	P-0-0088: Control word for synchronous operation mod	1
Bit 8	0	P-0-0155: Synchronization mode	1
Bit 9	0	P-0-0154: Synchronization direction	0
Bit 10	0	P-0-0154: Synchronization direction	1
Bit 11	0	S-0-0000: <reserved>	0
<axis>.UserCmdDataBitA	1	S-0-0405: Probe 1 enable	0
<axis>.UserCmdDataBitB	0	S-0-0000: <empty>	0
<axis>.UserCmdDataBitC	0	S-0-0000: <empty>	0
<axis>.UserCmdDataBitD	0	S-0-0000: <empty>	0

Fig. 3-63: Example of a signal control word

2. Configuration of a signal status word

	Status	Source Parameter	Bit-No.
Bit 0	1	S-0-0330: Message 'n_actual = n_command'	0
Bit 1	1	S-0-0331: Status 'n_feedback = 0'	0
Bit 2	0	S-0-0437: Positioning status word	2
Bit 3	0	S-0-0403: Position feedback value status	1
Bit 4	0	S-0-0419: Positioning command acknowledge	0
Bit 5	0	P-0-0089: Status word for synchronous operating mode	0
Bit 6	0	P-0-0089: Status word for synchronous operating mode	1
Bit 7	0	P-0-0089: Status word for synchronous operating mode	8
Bit 8	0	S-0-0824: Message torque/force command value reac	0
Bit 9	1	P-0-0089: Status word for synchronous operating mode	5
Bit 10	1	S-0-0336: Message In position	0
Bit 11	1	P-0-0089: Status word for synchronous operating mode	6
<axis>.UserActualDataBitA	1	S-0-0410: Probe 1 negative latched	0
<axis>.UserActualDataBitB	0	S-0-0000: <empty>	0
<axis>.UserActualDataBitC	0	S-0-0000: <empty>	0
<axis>.UserActualDataBitD	0	S-0-0000: <empty>	0

Fig. 3-64: Example of a signal status word

3. Configuration of a cyclic sercos channel

Configuration cyclic write access (MDT)

<axis>.UserCmdDataA	S-0-0000: <empty>	
<axis>.UserCmdDataB	S-0-0000: <empty>	
<axis>.UserCmdDataC	S-0-0000: <empty>	
<axis>.UserCmdDataD	S-0-0000: <empty>	

Configuration cyclic read access (AT)

<axis>.UserActualDataA	S-0-0131: Probe value 1 negative edge	0.0000	Deg
<axis>.UserActualDataB	S-0-0000: <empty>		
<axis>.UserActualDataC	S-0-0000: <empty>		
<axis>.UserActualDataD	S-0-0000: <empty>		

Fig. 3-65: Example of a cyclic sercos channel

3.9.3 MB_InitTouchProbe

Brief description The MB_InitTouchProbe function block initializes the touch probe functionality.



In version 14VRS, the "MB_InitTouchProbe (ML_TechBase library)" function block was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-71: Reference table of the MB_Init_TouchProbe

Interface description

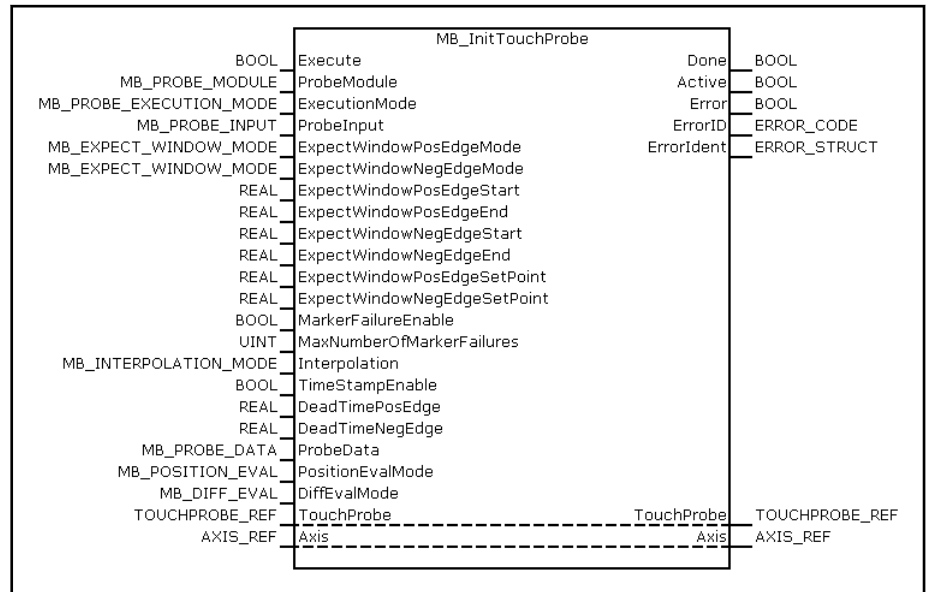


Fig. 3-66: MB_InitTouchProbe function block

I/O type	Name	Data type	Comment
VAR_INPUT	Execute	BOOL	It starts the initialization of the touch probe functionality at a rising edge
	ProbeModule	MB_PROBE_MODULE	Position of the touch probe on the MLC (onboard, function module) is currently not evaluated by the drive probe
	ExecutionMode	MB_PROBE_EXECUTION_MODE	Selection whether single-shot or continuous measurement MB_PROBE_EXECUTION_MODE, page 128
	ProbeInput	MB_PROBE_INPUT	Selects the digital input for the MLC touch probe, is currently not evaluated by the drive probe
	ExpectWindowPosEdgeMode	MB_EXPECT_WINDOW_MODE	Mode of expectation window if the positive edge is evaluated
	ExpectWindowNegEdgeMode	MB_EXPECT_WINDOW_MODE	Mode of expectation window if the negative edge is evaluated
	ExpectWindowPosEdgeStart	REAL	Start of the expectation window if the positive edge is evaluated
	ExpectWindowPosEdgeEnd	REAL	End of the expectation window if the positive edge is evaluated

ML_TechMotionBase.library // MX_TechMotionBase.library

I/O type	Name	Data type	Comment
	ExpectWindowNegEdgeStart	REAL	Start of the expectation window if the negative edge is evaluated
	ExpectWindowNegEdgeEnd	REAL	End of the expectation window if the negative edge is evaluated
	ExpectWindowPosEdgeSetPoint	REAL	Command value for the relative expectation window if the positive edge is evaluated
	ExpectWindowNegEdgeSetPoint	REAL	Command value for the relative expectation window if the negative edge is evaluated
	MarkerFailureEnable	BOOL	Enable marker failure
	MaxNumberOfMarkerFailure	UINT	Threshold for the registered number of marker failure
	Interpolation	MB_INTERPOLATION_MODE	Interpolation for the MLC touch probe, is currently not evaluated by the drive probe
	TimeStampEnable	BOOL	Activate time stamp for MLC probe, is currently not evaluated by the drive probe
	DeadTimePosEdge	REAL	Dead time compensation for the positive edge if evaluated
	DeadTimeNegEdge	REAL	Dead time compensation for the negative edge if evaluated
	ProbeData	MB_PROBE_DATA	Measured touch probe signal
	PositionEvalMode	MB_POSITION_EVAL	Selects the edge to be evaluated
	DiffEvalMode	MB_DIFF_EVAL	Edge selection for the difference measurement
VAR_OUTPUT	Done	BOOL	Initialization completed
	Active	BOOL	The function block is currently active
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 125
VAR_IN_OUT CONSTANT	TouchProbe	TOUCHPROBE_REF	Reference to the signal trigger source
	Axis	AXIS_REF	Reference to the axis

Tab. 3-72: Interface variables of the MB_InitTouchProbe function block

Min./max. and default values

The following table lists the min./max. and default values of the function block input:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
ProbeModule	MB_PROBE_MODULE			ON-BOARD_MODULE	Rising edge at "Execute"
ExecutionMode	MB_PROBE_EXECUTION_MODE			CONTINUOUS	Rising edge at "Execute"

Name	Type	Min. value	Max. value	Default value	Effective
ProbeInput	MB_PROBE_IN- PUT			D_INPUT_1	Rising edge at "Execute"
ExpectWindow- PosEdgeMode	MB_EX- PECT_WIN- DOW_MODE			DISABLED_EXP	Rising edge at "Execute"
ExpectWindow- NegEdgeMode	MB_EX- PECT_WIN- DOW_MODE			DISABLED_EXP	Rising edge at "Execute"
ExpectWindow- PosEdgeStart	REAL	0.0	ModuloValue	0.0	Rising edge at "Execute"
ExpectWindow- PosEdgeEnd	REAL	0.0	ModuloValue	0.0	Rising edge at "Execute"
ExpectWindow- NegEdgeStart	REAL	0.0	ModuloValue	0.0	Rising edge at "Execute"
ExpectWindow- NegEdgeEnd	REAL	0.0	ModuloValue	0.0	Rising edge at "Execute"
ExpectWindow- PosEdgeSetPoint	REAL	0.0	ModuloValue	0.0	Rising edge at "Execute"
ExpectWindow- NegEdgeSetPoint	REAL	0.0	ModuloValue	0.0	Rising edge at "Execute"
MarkerFailureEn- able	BOOL			FALSE	Rising edge at "Execute"
MaxNumberOf- MarkerFailure	UINT			0	Rising edge at "Execute"
TimeStampEna- ble	BOOL			FALSE	Rising edge at "Execute"
DeadTimePo- sEdge	REAL	0.0		0.0	Rising edge at "Execute"
DeadTimeNe- gEdge	REAL	0.0		0.0	Rising edge at "Execute"
ProbeData	MB_PROBE_DA- TA			TP_ACT_POS_F EEDBACK	Rising edge at "Execute"
PositionEvalMode	MB_POSI- TION_EVAL			DISABLED_EVAL	Rising edge at "Execute"
DiffEvalMode	MB_DIFF_EVAL			DISABLED_DIFF	Rising edge at "Execute"

Tab. 3-73: Min./max. and default values for the MB_TouchProbeContinuous in-
puts

Functional description of the MB_InitTouchProbe



The MB_InitTouchProbe function block must be executed in SERCOS phase 4 before the function blocks [MB_TouchProbe](#), [page 128](#), and [MB_TouchProbeContinuous](#), [page 131](#), can be used.

TouchProbe, ProbeModule, ProbeInput

The "TouchProbe" input defines the control and touch probes used. The "CntrlNo" element has to be set to "LOCAL_CNTRL". The "TouchProbeNo" element can either assume the value "TOUCHPROBE_DRIVE_1" or "TOUCHPROBE_DRIVE_2".

The inputs "ProbeModule" and "ProbeInput" are currently not evaluated.

ExecutionMode

At the "ExecutionMode" input, the touch probe function is set either to single-shot measurement or to continuous measurement. If the single-shot measurement is used, the MB_TouchProbe function block has to be used. For a continuous measurement, the MB_TouchProbeContinuous function block has to be used.

ExpectWindow, PositionEvalMode

"ExpectWindow" describes the expectation window of the touch probe. Currently, the expectation window of the drive does not support a separate expectation window for the negative and positive edges at the same time. If only the positive edge is evaluated ("PositionEvalMode"=POSITION_POS_EDGE), the inputs "ExpectWindowPosEdgeMode", "ExpectWindowPosEdgeStart", "ExpectWindowPosEdgeStart" and "ExpectWindowPosEdgeSetPoint" are evaluated.

If only the negative edge is evaluated ("PositionEvalMode"=POSITION_NEG_EDGE), "ExpectWindowNegEdgeMode", "ExpectWindowNegEdgeStart", "ExpectWindowNegEdgeStart" and "ExpectWindowNegEdgeSetPoint" are evaluated.

If both edges are evaluated ("PositionEvalMode"=POSITION_POS_NEG_EDGE), only the inputs "ExpectWindowPosEdgeMode", "ExpectWindowPosEdgeStart", "ExpectWindowPosEdgeEnd", "ExpectWindowPosEdgeSetPoint" for the expectation window of the drive are evaluated. The expectation window applies to both evaluated edges.

DeadTimePosEdge, DeadTimeNegEdge

The inputs "DeadTimePosEdge" and "DeadTimeNegEdge" determine the dead-time compensation of the edges. The "DeadTimePosEdge" input is valid if the positive edge is evaluated. The "DeadTimeNegEdge" input is valid if the negative edge is evaluated.

ProbeData, DiffEvalMode

The "ProbeData" input defines the measured signal of the touch probe function. The following signals can be selected:

ENUM MB_PROBE_DATA (selection)

Signal	Description
TP_ACT_POS_FEEDBACK1	Actual position value Encoder 1
TP_ACT_POS_FEEDBACK2	Actual position value Encoder 2
TP_ACT_POSIN_ACT_CYCLE	Actual position value in the actual value cycle
TP_SYNC_CMD_POS	ELS, synchronous position command value
TP_PROBE_TIME	Time stamp
TP_ACT_MEASURING_ENC_POS	Actual position value of the measuring encoder
TP_RES_MASTER_AXIS_POS	ELS, resulting master axis position

Signal	Description
TP_EFF_MASTER_AXIS_POS	ELS, effective master axis position
TP_CAM_SHAFT_PROF_ACCESS_ANGLE	ELS, cam, access phase
TP_ELS_CMD_POS	ELS, position command value

Tab. 3-74: Possible values of the "ProbeData" input

The "DiffEvalMode" input selects the difference measurement of the touch probe function. Only "STANDARD_MODE" is currently supported.

ModuloValue

The "ModuloValue" output displays the modulo value of the measured signal.

Interpolation

The "Interpolation" input indicates how to interpolate the position values. Currently, the MB_InitTouchProbe function block only supports the values "NO_INTERPOLATION" and "LINEAR_INTERPOLATION". It is not evaluated further.

Time diagram

Timing diagram according to the PLCopen specification.

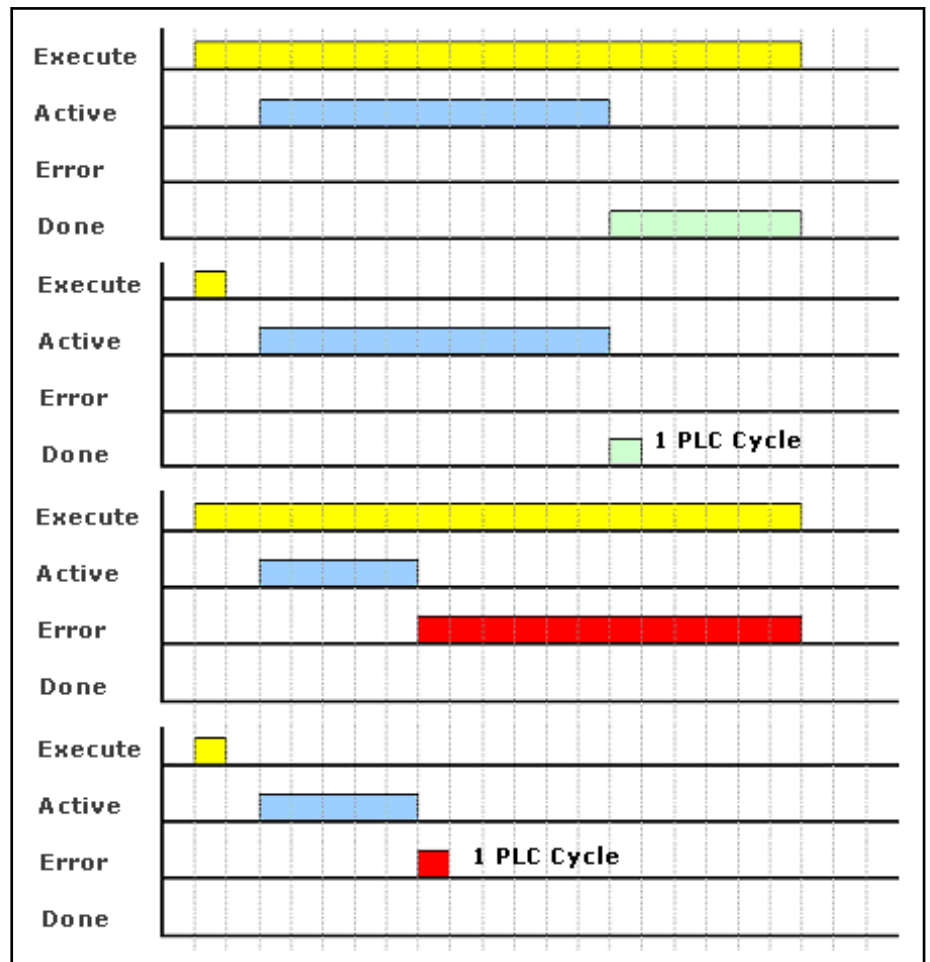


Fig. 3-67: Timing diagram of the MB_InitTouchProbe function block

Troubleshooting

The function block generates the following error messages in Additional1/Additional2 for the "F_RELATED_TABLE" 16#0170:

ML_TechMotionBase.library // MX_TechMotionBase.library

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"AxisNo" for "Axis" is not in the valid range
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Drive is not enabled or drive error
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis (Axis_Ref) was changed while the function block was operating
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0001	The "ProbeModule" input is invalid
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0002	The "ExecutionMode" input is invalid
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0003	The "ProbeInput" input is invalid
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0004	The "DeadTimePosEdge" input is invalid (check whether < 0)
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0005	The "DeadTimeNegEdge" input is invalid (check whether < 0)
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0006	The "ProbeData" input is invalid
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0007	The "PositionEvalMode" input is invalid
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0008	The "DiffEvalMode" input is invalid
INPUT_INVALID_ERROR (16#0001)	16#0404	16#0009	The "MaxNumberOfMarkerFailure" input is invalid (check whether < 0)
INPUT_INVALID_ERROR (16#0001)	16#0404	16#000A	The "TouchProbe.TouchProbeNo" input is invalid
INPUT_INVALID_ERROR (16#0001)	16#0404	16#000B	The "interpolation" input is invalid
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0001	The "DelExpectWindowPosEdgeModeay" input is invalid
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0002	The "ExpectWindowNegEdgeMode" input is invalid
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0003	The "ExpectWindowPosEdgeStart" input is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0004	The "ExpectWindowPosEdgeEnd" input is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0005	The "ExpectWindowNegEdgeStart" input is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0006	The "ExpectWindowNegEdgeEnd" input is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0007	The "ExpectWindowPosEdgeSetPoint" input is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0008	The "ExpectWindowNegEdgeSetPoint" input is not in the valid range
RESOURCE_ERROR (16#0003)	16#0403	16#0001	Required touch probe control bit (S-0-0405, Bit 0) is not configured in the signal control word
RESOURCE_ERROR (16#0003)	16#0403	16#0002	Required touch probe control bit (S-0-0406, Bit 0) is not configured in the signal control word

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0403	16#0003	Required touch probe status bit (S-0-0409, Bit 0) is not configured in the signal status word
RESOURCE_ERROR (16#0003)	16#0403	16#0004	Required touch probe status bit (S-0-0410, Bit 0) is not configured in the signal status word
RESOURCE_ERROR (16#0003)	16#0403	16#0005	Required touch probe status bit (S-0-0411, Bit 0) is not configured in the signal status word
RESOURCE_ERROR (16#0003)	16#0403	16#0006	Required touch probe status bit (S-0-0412, Bit 0) is not configured in the signal status word
RESOURCE_ERROR (16#0003)	16#0403	16#0007	Required touch probe value S-0-0130 is not cyclically configured in the AT
RESOURCE_ERROR (16#0003)	16#0403	16#0008	Required touch probe value S-0-0131 is not cyclically configured in the AT
RESOURCE_ERROR (16#0003)	16#0403	16#0009	Required touch probe value S-0-0132 is not cyclically configured in the AT
RESOURCE_ERROR (16#0003)	16#0403	16#000A	Required touch probe value S-0-0133 is not cyclically configured in the AT
RESOURCE_ERROR (16#0003)	16#0403	16#000B	Required touch probe parameter P-0-0224 is not cyclically configured in the AT
RESOURCE_ERROR (16#0003)	16#0403	16#000C	Required touch probe parameter P-0-0225 is not cyclically configured in the AT
RESOURCE_ERROR (16#0003)	16#0403	16#000D	Required touch probe parameter P-0-0202 is not cyclically configured in the AT
RESOURCE_ERROR (16#0003)	16#0403	16#000E	Required touch probe parameter P-0-0203 is not cyclically configured in the AT
RESOURCE_ERROR (16#0003)	16#0403	16#00XX	P-0-1367, bit 6 of the MLD-M master axis is not set to TRUE. This is required to update the "Axis-Data" structure

Tab. 3-75: Error codes of the MB_InitTouchProbe function block

Parameterized parameters

The MB_InitTouchProbe function block parameterizes the following parameters:

- P-0-0226, Touch probe, extended control word
- S-0-0169, Touch probe control parameter
- S-0-0170, Command touch probe cycle
- S-0-0426, Signal selection, touch probe 1
- S-0-0427, Signal selection, touch probe 2



Configure the cyclic channels manually before the start when using the touch probe function in the drive.

The bits of the signal control word and the signal status word have to be displayed and the measured values S-0-0130, S-0-0131 or P-0-0202 belonging to the selected touch probe function.

3.9.4 MB_PROBE_EXECUTION_MODE

The enumeration type allows you to select whether a single-shot or continuous measurement is to be performed.

Name	Value	Description
CONTINUOUS	0	Touch probe function continuous measurement
SINGLE_SHOT	1	Single-shot touch probe function

Tab. 3-76: ENUM MB_PROBE_EXECUTION_MODE

3.9.5 MB_TouchProbe

Brief description The MB_TouchProbe function block detect an axis position, a master position or time in case of an event trigger. It is a function block for the single-shot measurement.

The function block has to be executed again to detect another signal.

The signal to be measured is defined at the "TouchProbe" input of the function block [MB_InitTouchProbe](#), page 120,.



The function block can only be used if the touch probe function was configured before using the function block [MB_InitTouchProbe](#), page 120,.

It is not sufficient to manually configure the touch probe function!



In version 14VRS, the "MB_TouchProbe (ML_TechBase library)" function block was moved to the "TechMotionBase library".

Target system	Library
MLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-77: MB_TouchProbe reference table

Interface description

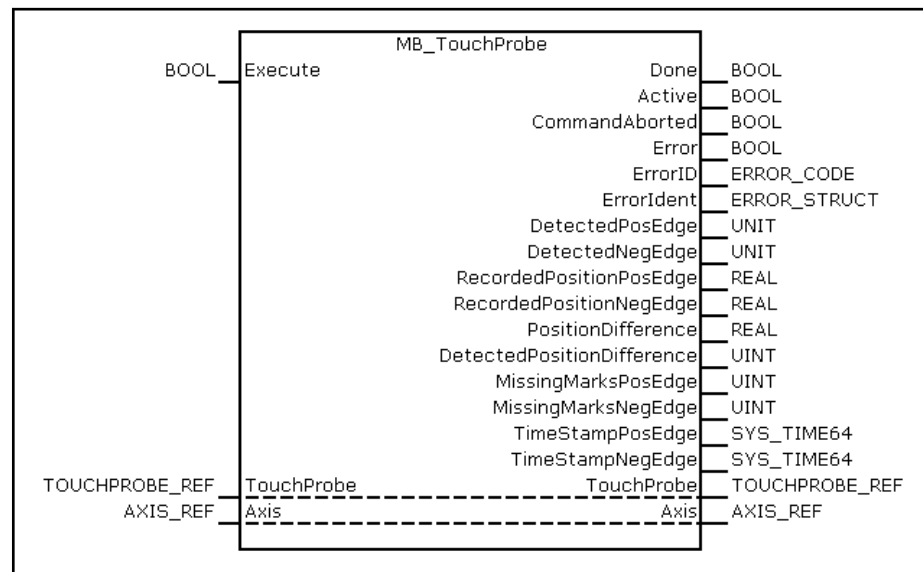


Fig. 3-68: MB_TouchProbe function block

I/O type	Name	Data type	Comment
VAR_INPUT	Execute	BOOL	Starts the touch probe recording at a rising edge
VAR_OUTPUT	Done	BOOL	Trigger event recorded
	Active	BOOL	The function block is currently active
	CommandAborted	BOOL	Command aborted by MB_AbortTrigger
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 130
	DetectedPosEdge	UINT	Positive edge detected
	DetectedNegEdge	UINT	Negative edge detected
	RecordedPositionPosEdge	REAL	Position at which the positive edge occurred (in technical units)
	RecordedPositionNegEdge	REAL	Position at which the negative edge occurred (in technical units)
	PositionDifference	REAL	Indicates the difference between the touch probe values depending on the "DiffEvalMode"
	DetectedPositionDifference	UINT	Difference measurement detected
	MissingMarksPosEdge	UINT	Counter of missing marks if the positive edge is evaluated
	MissingMarksNegEdge	UINT	Counter of missing marks if the negative edge is evaluated
	TimeStampPosEdge	SYS_TIME64	Point in time when the positive edge occurred (only MLC/XLC touch probes)
	TimeStampNegEdge	SYS_TIME64	Point in time when the negative edge occurred (only MLC/XLC touch probes)
VAR_IN_OUT CONSTANT	TouchProbe	TOUCHPROBE_REF	Reference to the signal trigger source
	Axis	AXIS_REF	Reference to the axis

Tab. 3-78: Interface variables of the MB_TouchProbe function block

Default value The following table lists the default value of the function block input:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL	n/a	n/a	FALSE	Continuous

Tab. 3-79: Default value for the MB_TouchProbe input

Functional description of the MB_TouchProbe

The MB_TouchProbe function block executed a single-shot measurement. The function block is enabled at a rising edge at the "Execute" input.

DetectedPosEdge, DetectedNegEdge

If the touch probe function is set to evaluate a positive/negative edge of the input signal, the "DetectedPosEdge"/"DetectedNegEdge" output increases when a positive edge (negative edge) is detected.

RecordedPositionPosEdge, RecordedPositionNegEdge

The outputs "RecordedPositionPosEdge" and "RecordedPositionNegEdge" contain the position at which the positive or negative edge occurred.

PositionDifference, DetectedPositionDifference

The "PositionDifference" output contains, in case of an activated difference measurement, the difference of the two latest measured values of the same touch probe. At each measurement, the "DetectedPositionDifference" output is increased.

The following calculation applies if the drive touch probe function is enabled for the "PositionDifference" output:

$$\text{PositionDifference} = |(S-0-0130) - (S-0-0131)|$$

$$\text{PositionDifference} = |(S-0-0132) - (S-0-0133)|$$

Fig. 3-69: Calculating the position difference for axes

MissingMarksPosEdge, MissingMarksNegEdge

The "MissingMarksPosEdge" or "MissingMarksNegEdge" outputs are only evaluated if the expectation window is active and the marker failure was enabled. They count the marks outside the expectation window. The outputs are set to 0 every time a mark is detected in the expectation window. If the touch probe is set to evaluate both edges, only the "MissingMarksPosEdge" output is valid. Currently, the drive touch probe function does not support individual marker failures for each edge at the same time.

Time diagram

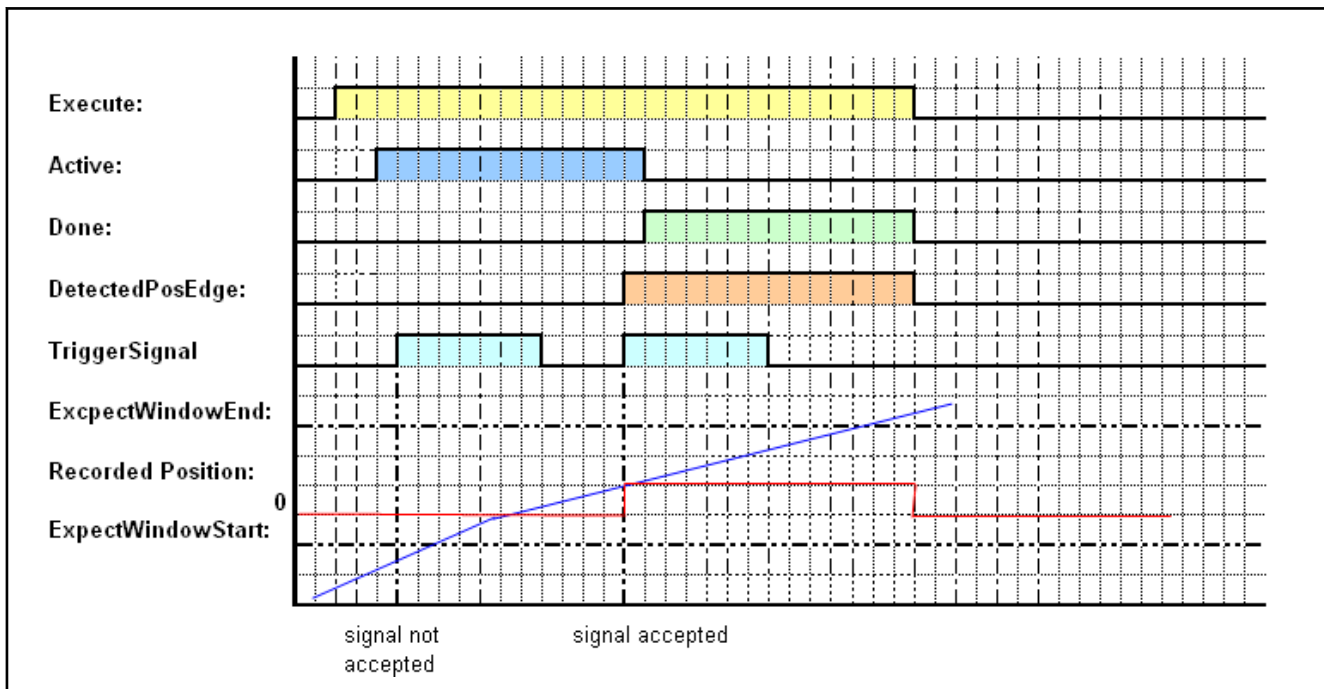


Fig. 3-70: Timing diagram of the MB_TouchProbe function block

Troubleshooting

The function block generates the following error messages in Additional1/Additional2 for the "F_RELATED_TABLE" 16#0170:

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo of "Axis" is not in the valid range
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Drive is not enabled or drive error
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis (Axis_Ref) was changed while the function block was operating
ACCESS_ERROR (16#0003)	16#0406	16#0001	Empty handle for ATPosEdgeVal
ACCESS_ERROR (16#0003)	16#0406	16#0002	Empty handle for ATNegEdgeVal
ACCESS_ERROR (16#0003)	16#0406	16#0003	Empty handle for ATMarkerFailure
ACCESS_ERROR (16#0003)	16#0406	16#0004	Empty handle for ATDiffProbeVal
ACCESS_ERROR (16#0003)	16#0406	16#0005	Empty handle for ATDeadTimeComp
ACCESS_ERROR (16#0003)	16#0406	16#0006	Empty handle for SCWEnableProbe
ACCESS_ERROR (16#0003)	16#0406	16#0007	Empty handle for SSWPosLatched
ACCESS_ERROR (16#0003)	16#0406	16#0008	Empty handle for SSWNegLatched

Tab. 3-80: Error codes of the MB_TouchProbe function block

3.9.6 MB_TouchProbeContinuous

Brief description The MB_TouchProbeContinuous function block detects an axis position, a master position or time in case of an event trigger. It executes a continuous measurement as long as the "Enable" input is TRUE. The signal to be detected is defined at the "TouchProbe" input of the function block [MB_InitTouchProbe](#), page 120,.



The function block can only be used if the touch probe function was configured before using the function block [MB_InitTouchProbe](#), page 120,.

It is not sufficient to manually configure the touch probe function!



In version 14VRS, the "MB_TouchProbeContinuous (ML_TechBase library)" function block was moved to the "TechMotionBase library".

Target system	Library
MLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-81: Reference table of the MB_TouchProbeContinuous

ML_TechMotionBase.library // MX_TechMotionBase.library

Interface description

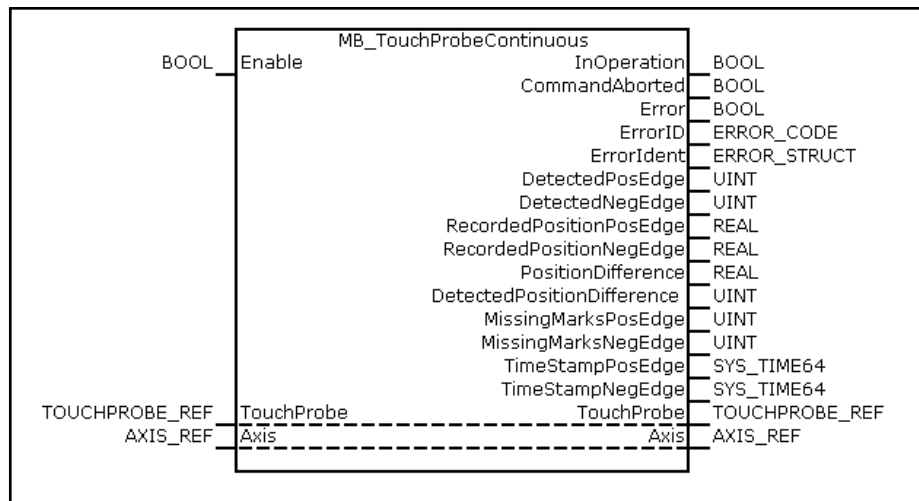


Fig. 3-71: MB_TouchProbeContinuous function block

I/O type	Name	Data type	Comment
VAR_INPUT	Enable	BOOL	Starts the touch probe recording at TRUE
VAR_OUTPUT	InOperation	BOOL	Touch probe in operation
	CommandAborted	BOOL	Command aborted by MB_AbortTrigger
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description
	DetectedPosEdge	UINT	Positive edge detected
	DetectedNegEdge	UINT	Negative edge detected
	RecordedPositionPosEdge	REAL	Position at which the positive edge occurred (in technical units)
	RecordedPositionNegEdge	REAL	Position at which the negative edge occurred (in technical units)
	PositionDifference	REAL	Indicates the absolute difference between the touch probe values
	DetectedPositionDifference	UINT	Difference measurement detected
	MissingMarksPosEdge	UINT	Counter of missing marks for the positive edge if evaluated
	MissingMarksNegEdge	UINT	Counter of missing marks for the negative edge if evaluated
	TimeStampPosEdge	SYS_TIME64	Point in time when the positive edge occurred (only MLC/XLC touch probes)
	TimeStampNegEdge	SYS_TIME64	Point in time when the negative edge occurred (only MLC/XLC touch probes)
VAR_IN_OUT CONSTANT	TouchProbe	TOUCHPROBE_REF	Reference to the signal trigger source
	Axis	AXIS_REF	Reference to the axis

Tab. 3-82: Interface variables of the MB_TouchProbeContinuous function block

Default value The following table lists the default value of the function block input:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous

Tab. 3-83: Default value for the MB_InitTouchProbe input

Functional Description

The MB_TouchProbeContinuous function block is exclusively used for the continuous measurement. The function block is enabled as long as the "Enable" input is TRUE.

DetectedPosEdge, DetectedNegEdge

If the touch probe function is set to evaluate a positive/negative edge of the input signal, the "DetectedPosEdge"/"DetectedNegEdge" output increases when a positive edge (negative edge) is detected.

RecordedPositionPosEdge, RecordedPositionNegEdge

The outputs "RecordedPositionPosEdge" and "RecordedPositionNegEdge" contain the position at which the positive or negative edge occurred.

PositionDifference, DetectedPositionDifference

The "PositionDifference" output contains, in case of an activated difference measurement, the difference of the two latest measured values of the same touch probe. At each measurement, the "DetectedPositionDifference" output is increased.

The following calculation applies if the drive touch probe function is enabled for the "PositionDifference" output:

$$\begin{aligned} \text{PositionDifference} &= |(S-0-0130) - (S-0-0131)| \\ \text{PositionDifference} &= |(S-0-0132) - (S-0-0133)| \end{aligned}$$

Fig. 3-72: Calculating the position difference for axes

MissingMarksPosEdge, MissingMarksNegEdge

The "MissingMarksPosEdge" or "MissingMarksNegEdge" outputs are only evaluated if the expectation window is active and the marker failure was enabled. They count the marks outside the expectation window. The outputs are set to 0 every time a mark is detected in the expectation window. If the touch probe is set to evaluate both edges, only the "MissingMarksPosEdge" output is valid. Currently, the drive touch probe function does not support individual marker failures for each edge at the same time.

Time diagram

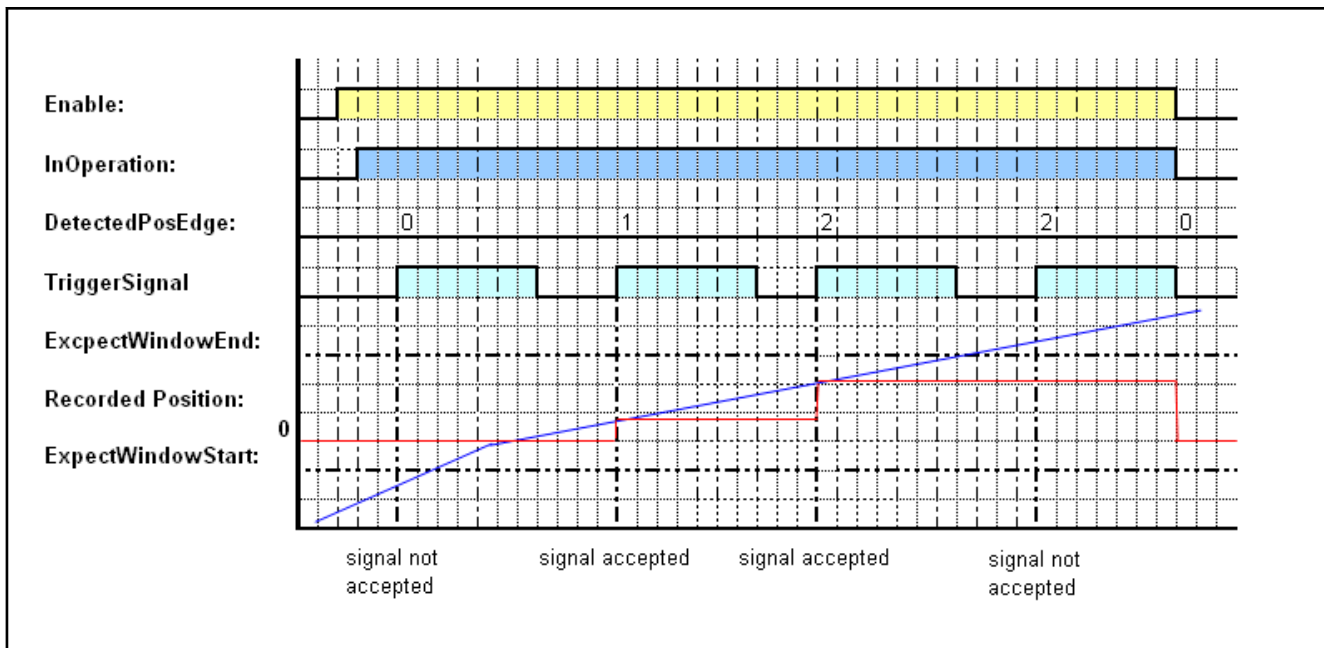


Fig. 3-73: Timing diagram of the MB_TouchProbeContinuous function block

Troubleshooting The function block generates the following error messages in Additional1/Additional2 for the "F_RELATED_TABLE" 16#0170:

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"AxisNo" for "Axis" is not in the valid range
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Drive is not enabled or drive error
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis (Axis_Ref) was changed while the function block was operating
ACCESS_ERROR (16#0003)	16#0406	16#0001	Empty handle for ATPosEdgeVal
ACCESS_ERROR (16#0003)	16#0406	16#0002	Empty handle for ATNegEdgeVal
ACCESS_ERROR (16#0003)	16#0406	16#0003	Empty handle for ATMarkerFailure
ACCESS_ERROR (16#0003)	16#0406	16#0004	Empty handle for ATDiffProbeVal
ACCESS_ERROR (16#0003)	16#0406	16#0005	Empty handle for ATDeadTimeComp
ACCESS_ERROR (16#0003)	16#0406	16#0006	Empty handle for SCWEnableProbe
ACCESS_ERROR (16#0003)	16#0406	16#0007	Empty handle for SSWPosLatched
ACCESS_ERROR (16#0003)	16#0406	16#0008	Empty handle for SSWNegLatched

Tab. 3-84: Error codes of the MB_TouchProbeContinuous function block

3.9.7 MB_WriteExpectWindow

Brief description The MB_WriteExpectWindow function block configures the expectation window for the touch probe functionality.



In version 14VRS, the "MB_WriteExpectWindow (ML_TechBase library)" function block was moved to the "TechMotionBase library".

Target system	Library
MLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-85: Reference table of the MB_WriteExpectWindow

Interface description

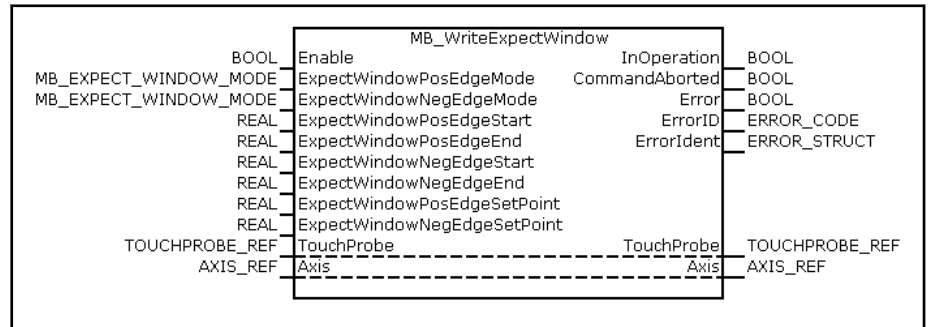


Fig. 3-74: MB_WriteExpectWindow function block

I/O type	Name	Data type	Comment
VAR_INPUT	Enable	BOOL	Enables the function block
	ExpectWindowPosEdgeMode	MB_EXPECT_WINDOW_MODE	Mode of expectation window if the positive edge is evaluated
	ExpectWindowNegEdgeMode	MB_EXPECT_WINDOW_MODE	Mode of expectation window if the negative edge is evaluated
	ExpectWindowPosEdgeStart	REAL	Start of the expectation window if the positive edge is evaluated
	ExpectWindowPosEdgeEnd	REAL	End of the expectation window if the positive edge is evaluated
	ExpectWindowNegEdgeStart	REAL	Start of the expectation window if the negative edge is evaluated
	ExpectWindowNegEdgeEnd	REAL	End of the expectation window if the negative edge is evaluated
	ExpectWindowPosEdgeSetPoint	REAL	Command value for the relative expectation window if the positive edge is evaluated
	ExpectWindowNegEdgeSetPoint	REAL	Command value for the relative expectation window if the negative edge is evaluated
VAR_OUTPUT	InOperation	BOOL	Function block in operation
	CommandAborted	BOOL	Function block aborted
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 137
VAR_IN_OUT CONSTANT	TouchProbe	TOUCHPROBE_REF	Reference to the signal trigger source
	Axis	AXIS_REF	Reference to the axis

Tab. 3-86: Interface variables of the MB_WriteExpectWindow function block

Min./max. and default values The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
ExpectWindow-PosEdgeMode	MB_EXPECT_WINDOW_MODE			DISABLED_EXP	Rising edge at "Enable"
ExpectWindow-NegEdgeMode	MB_EXPECT_WINDOW_MODE			DISABLED_EXP	Rising edge at "Enable"
ExpectWindow-PosEdgeStart	REAL	0.0	ModuloValue	0.0	Continuous
ExpectWindow-PosEdgeEnd	REAL	0.0	ModuloValue	0.0	Continuous
ExpectWindow-NegEdgeStart	REAL	0.0	ModuloValue	0.0	Continuous
ExpectWindow-NegEdgeEnd	REAL	0.0	ModuloValue	0.0	Continuous
ExpectWindow-PosEdgeSetPoint	REAL	0.0	ModuloValue	0.0	Continuous
ExpectWindow-NegEdgeSetPoint	REAL	0.0	ModuloValue	0.0	Continuous

Tab. 3-87: *Min./max. and default values for inputs of the MB_WriteExpectWindow function block*

Functional description of the drive touch probe

Currently, the expectation window of the drive does not support a separate expectation window for the negative and positive edges at the same time. If only the positive edge is evaluated, the inputs "ExpectWindowPosEdgeMode", "ExpectWindowPosEdgeStart", "ExpectWindowPosEdgeStart" and "ExpectWindowPosEdgeSetPoint" are evaluated.

If the negative edge is evaluated, "ExpectWindowNegEdgeMode", "ExpectWindowNegEdgeStart", "ExpectWindowNegEdgeStart" and "ExpectWindowNegEdgeSetPoint" are evaluated.

if both edges are evaluated, only the inputs "ExpectWindowPosEdgeMode", "ExpectWindowPosEdgeStart", "ExpectWindowPosEdgeEnd" and "ExpectWindowPosEdgeSetPoint" are evaluated for the expectation window of the drive. The expectation window applies to both evaluated edges.

Time diagram

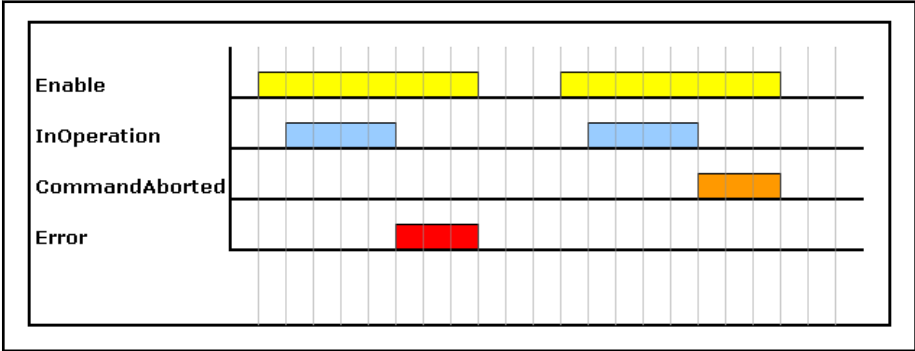


Fig. 3-75: *Timing diagram of the MB_WriteExpectWindow function block*

Troubleshooting The function block generates the following error messages in Additional1/Additional2 for the "F_RELATED_TABLE" 16#0170:

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"AxisNo" for "Axis" is not in the valid range
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Drive is not enabled or drive error
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis (Axis_Ref) was changed while the function block was operating
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0001	The "ExpectWindowPosEdgeMode" input is invalid
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0002	The "ExpectWindowNegEdgeMode" input is invalid
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0003	The "ExpectWindowPosEdgeStart" input is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0004	The "ExpectWindowPosEdgeEnd" input is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0005	The "ExpectWindowNegEdgeStart" input is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0006	The "ExpectWindowNegEdgeEnd" input is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0007	The "ExpectWindowPosEdgeSetPoint" input is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0008	The "ExpectWindowNegEdgeSetPoint" input is not in the valid range
RESOURCE_ERROR (16#0003)	16#0403	16#000F	P-0-1367, bit 6 of the MLD-M master axis is not set to TRUE. This is required to update the "Axis-Data" structure.

Tab. 3-88: Error codes of the MB_WriteExpectWindow function block

3.9.8 MB_AbortTrigger

Brief description The MB_AbortTrigger function block aborts the function block [MB_TouchProbe](#), page 128.



In version 14VRS, the "MB_AbortTrigger (ML_TechBase library)" function block was moved to the "TechMotionBase library".

Target system	Library
MLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-89: Reference table of the MB_AbortTrigger

Interface description

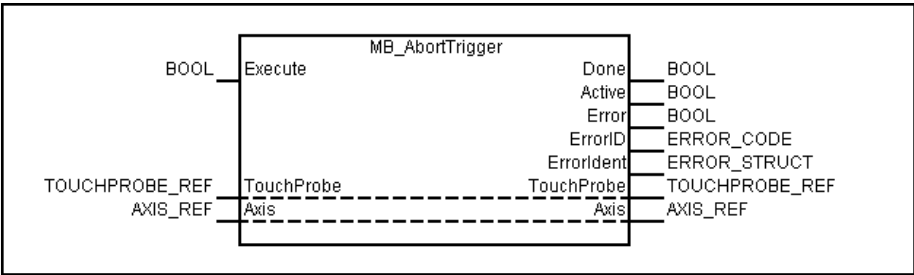


Fig. 3-76: MB_AbortTrigger function block

I/O types	Name	Data type	Comment
VAR_INPUT	Execute	BOOL	Starts the function block at a rising edge
VAR_OUTPUT	Done	BOOL	Touch probe aborted
	Active	BOOL	The function block is currently active
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 139,
VAR_IN_OUT CONSTANT	TouchProbe	TOUCHPROBE_REF	Reference to the signal trigger source
	Axis	AXIS_REF	Reference to the axis

Tab. 3-90: Interface variables of the MB_AbortTrigger function block

Default value The following table lists the default value of the function block input:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous

Tab. 3-91: Default value for the input of the MB_AbortTrigger function block

Functional description The MB_AbortTrigger function block disables the touch probe function. It can only be used together with the MB_TouchProbe function block. To disable the continuous measurement, the "Enable" input of the MB_TouchProbeContinuous function block has to be set to FALSE.

Time diagram Timing diagram according to the PLCopen specification.

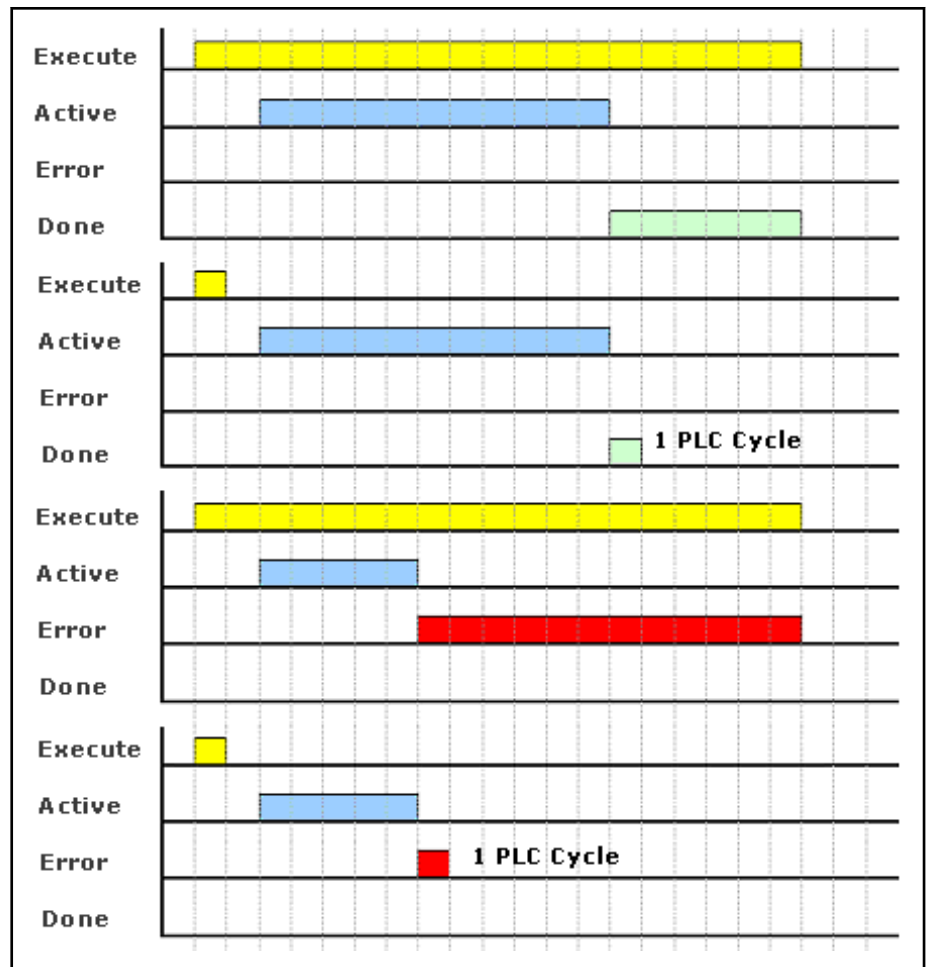


Fig. 3-77: Timing diagram of the MB_AbortTrigger function block

Troubleshooting

The function block generates the following error messages in Additional1/Additional2 for the "F_RELATED_TABLE" 16#0170:

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"AxisNo" for "Axis" is not in the valid range
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Drive is not enabled or drive error
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis (Axis_Ref) was changed while the function block was operating
ACCESS_ERROR (16#0003)	16#0406	16#0006	Empty handle for SCWEnableProbe

Tab. 3-92: Error codes of the MB_AbortTrigger function block

3.10 Cyclic data channels, administration

3.10.1 Introduction and overview

With the Sercos interface, freely configurable cyclic data can be applied. This data is cyclically transferred in each Sercos cycle from the control system to the drives and from the drives to the control. For each axis, four parameters are available in read and write direction. In addition, four bits can be read or

written. The cyclic data is configured in the parameterization mode (PM, Sercos phase 2) and cannot be changed in the operation mode (BB, Sercos phase 4).

In the parameterization mode (PM, Sercos phase 2), a cyclic data channel for a parameter to be transferred can be requested by the `MB_AllocCyclicDataParameter` function block. If the parameter is already applied in the cyclic data channel, the configuration is not changed. Otherwise, it is browsed for a free data channel and the respective parameter is applied.

If the configuration is changed, a cyclic parameter applied once can be enabled again with `MB_FreeCyclicParameter`, page 143. Thus, this channel is available for a new configuration. The channel has to be only explicitly enabled if the data channels are newly configured from the PLC at runtime.

Configuration is also possible in IndraWorks via the AxisData or the cyclic actual and command value dialog. A configuration defined by IndraWorks is used automatically. If some parameters are configured via IndraWorks and other parameters are requested later at runtime with `MB_AllocCyclicParameter`, page 141,, the new parameters are distributed to free channels. `MB_GetCyclicChannelConfig`, page 145, can be used to read out the current configuration of the cyclically applied parameters.

In the operating mode (BB), the function block `MB_GetCyclicParameterHandle`, page 146, searches a cyclically applied parameter in the configuration of one axis and writes all required information to a handle. With this handle, the parameter can be read or it can be written to the parameter in each Sercos cycle using `MB_ReadCyclicParameter`, page 148, and `MB_WriteCyclicParameter`, page 149,. The functions `MB_ReadCyclicRealParameter`, page 149, and `MB_WriteCyclicRealParameter`, page 150, are available for cyclic parameters with decimal places. Errors that occurred while accessing the cyclic data channels can be retrieved via the function `MB_GetLastCyclicParameterError`, page 150,.

The cyclically applied parameters are also available in the AxisData structure of the respective axis. The actual values are specified in the UserActualDataX entries. The command values are written using UserCmdDataX. However, write to the correct data type according to the applied parameter. This is not required when using the function blocks `MB_ReadCyclicParameter` and `MB_WriteCyclicParameter`.

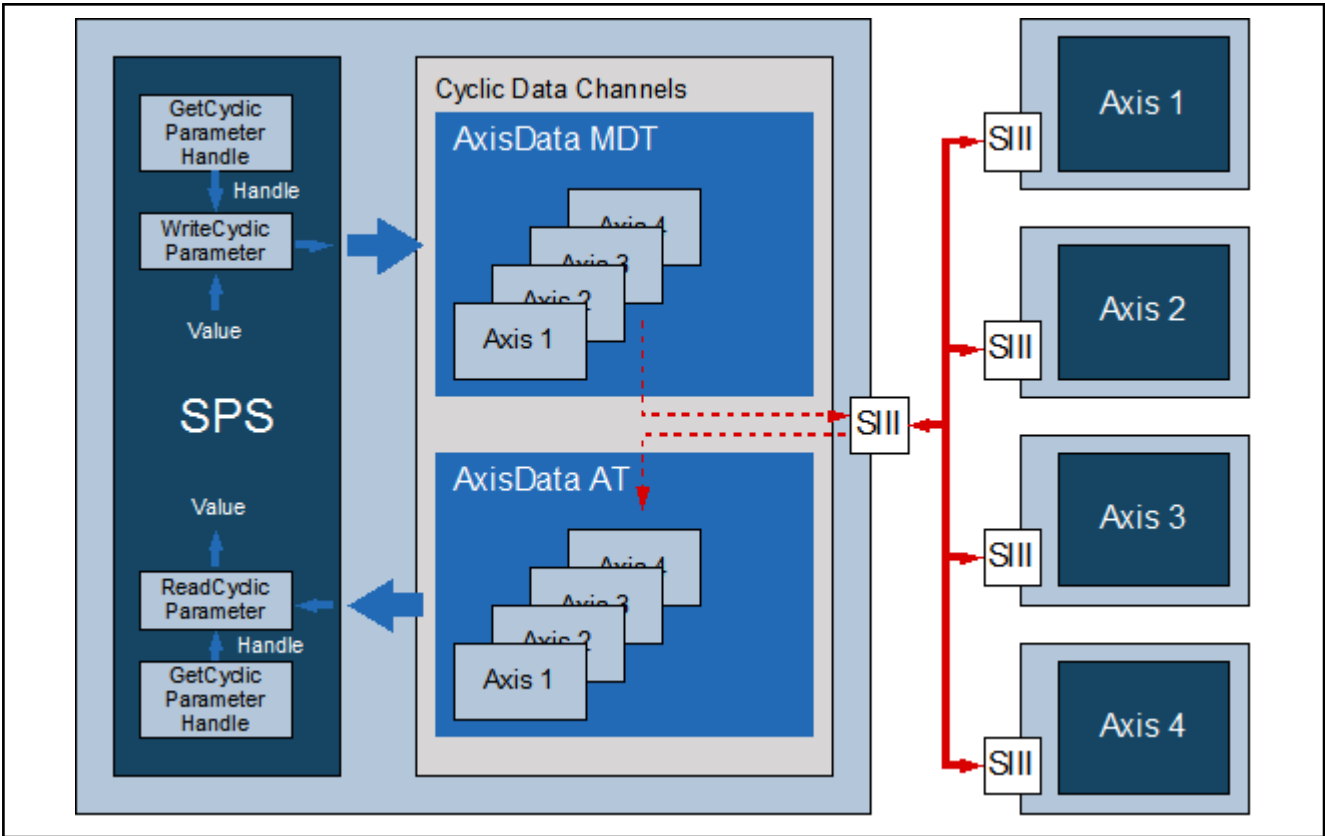


Fig. 3-78: Accessing the cyclic data using the "CyclicParameter" function blocks and functions

3.10.2 MB_AllocCyclicParameter

Brief description

This function block checks the configuration of a parameter to be cyclically read or written to in the cyclic data channel of an axis and configures the cyclic data channel if required.
Assigned parameters are stored in a global data structure.

In version 14VRS, the "MB_AllocCyclicParameter (ML_TechBase library)" function block was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC/XLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-93: Reference table of the MB_AllocCyclicParameter

Interface description

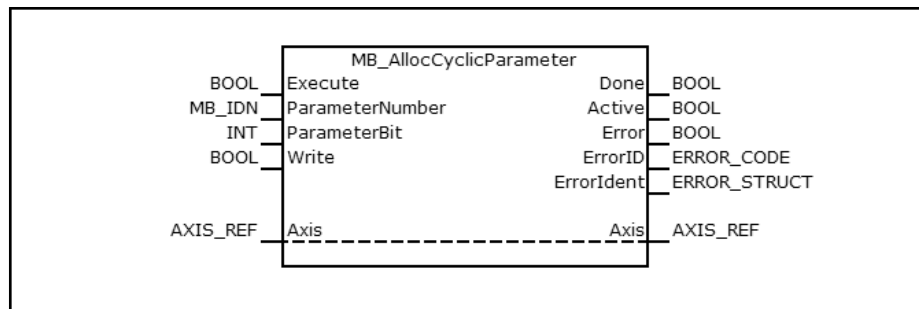


Fig. 3-79: Function block MB_AllocCyclicParameter

I/O type	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Starts the parameter configuration
	ParameterNumber	MB_IDN	Parameter to be applied
	ParameterBit	INT	Internal parameter bit; "CYCLIC_ALLBITS" for the entire parameter
	Write	BOOL	FALSE: Read parameter TRUE: Write parameter
VAR_OUTPUT	Done	BOOL	Parameter was applied
	Active	BOOL	The function block is active
	Error	BOOL	Indicates that an error occurred
	ErrorID	ERROR_CODE	Basic error cause
	ErrorIdent	ERROR_STRUCT	For the error structure with further error division, see Error codes, page 143
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Parameter axis

Tab. 3-94: Interface variables of the MB_AllocCyclicParameter function block

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
ParameterNumber	DINT	0	n.def.	0	Rising edge at "Execute"
ParameterBit	INT	-1	31	CYCLIC_ALLBITS	Rising edge at "Execute"
Write	BOOL			FALSE	Rising edge at "Execute"

Tab. 3-95: Input behavior of the MB_AllocCyclicParameter

Functional description

This function block is generally only operational in Sercos phase 2. At a rising edge at "Execute", the "ParameterNumber", "ParameterBit" and "Write" inputs are accepted. The actual data (UserActualData) and the command data (UserCmdData) are distinguished using the "Write" input. "ParameterBit" specifies the internal bit number of the parameter. If "ParameterBit" is between 0 and 31, a bit data channel is used. If "ParameterBit" is equal to "CYCLIC_ALLBITS", the entire parameter is applied. "ParameterNumber" specifies the parameter to be applied.

In the first step, MB_AllocCyclicParameter checks whether the parameter has already been stored in the current configuration. If this is not the case, it is

searched for a free cyclic channel and the parameter is applied there. Subsequently, the assignment is stored in an internal static variable. If a bit number is specified in "ParameterBit", a free bit channel is preferred. If all bit channels are assigned, a parameter channel is used. After a successful configuration of the cyclic data channel, the output "Done" is set to TRUE for at least one cycle. "Done" remains TRUE as long as the input "Execute" is active. If there is no free cyclic channel available, a corresponding error message is output.

Error behavior If the parameter cannot be applied in the cyclic data channel or if another error occurs, "Error" is set to TRUE.

Error codes The function block generates the following error messages of the "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR	16#000D	AxisNo	AxisRef is not in the valid range
ACCESS_ERROR	16#0010	16#0000	Sercos phase 2 is not active
ACCESS_ERROR	16#1504	16#0000	AxisData structure not enabled
INPUT_INVALID_ERROR	16#F212	16#0000	Parameter type structure not supported
INPUT_INVALID_ERROR	16#F213	16#0000	Parameter type function not supported
INPUT_INVALID_ERROR	16#F214	16#0000	Parameter type not supported
INPUT_INVALID_ERROR	16#F215	16#0000	Cannot be written to the parameter
RESOURCE_ERROR	16#F216	16#0000	No free cyclic channel
INPUT_INVALID_ERROR	16#F217	16#0000	Parameter cannot be configured in the cyclic data channel

Tab. 3-96: Error codes of the MB_AllocCyclicParameter function block

3.10.3 MB_FreeCyclicParameter

Brief description This function block enables an already configured cyclic parameter to be read or written to in the cyclic data channel of an axis. The enabling of the parameter is stored in a global structure.



In version 14VRS, the "MB_FreeCyclicParameter (ML_TechBase library)" function block was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC/XLC 14VRS	ML_TechMotionBase
MOT function package	
MLD (IL2G)	MX_TechMotionBase

Tab. 3-97: Reference table of the MB_FreeCyclicParameter

Interface description

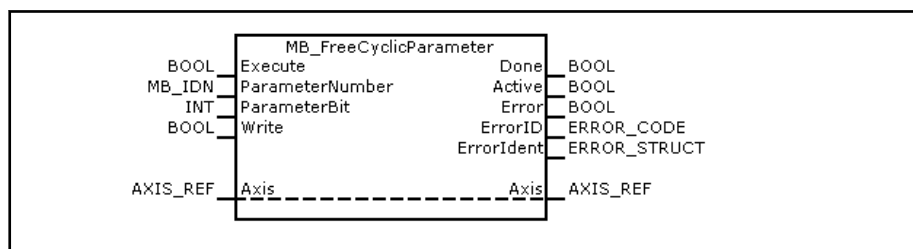


Fig. 3-80: MB_FreeCyclicParameter function block

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Starts enabling the parameter
	ParameterNumber	MB_IDN	Number of the parameter to be enabled
	ParameterBit	INT	Internal parameter bit; "CYCLIC_ALLBITS" for the entire parameter
	Write	BOOL	FALSE: Read parameter TRUE: Write parameter
VAR_OUTPUT	Done	BOOL	Parameter was applied
	Active	BOOL	The function block is active
	Error	BOOL	Indicates that an error occurred
	ErrorID	ERROR_CODE	Basic error cause
	ErrorIdent	ERROR_STRUCT	For the error structure with further error division, see Error codes, page 144
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Parameter axis

Tab. 3-98: Interface variables of the MB_FreeCyclicParameter function block

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
ParameterNumber	MB_IDN	0	n.def.	0	Rising edge at "Execute"
ParameterBit	INT	-1	31	CYCLIC_ALLBITS	Rising edge at "Execute"
Write	BOOL			FALSE	Rising edge at "Execute"

Tab. 3-99: Input behavior of the MB_FreeCyclicParameter function block

Functional description

This function block only operates in sercos phase 2. At a rising edge at "Execute", the "ParameterNumber", "ParameterBit" and "Write" inputs are applied. The actual data (UserActualData) and the command data (UserCmdData) are distinguished using the "Write" input. "ParameterBit" specifies the internal bit number of the parameter. If "ParameterBit" is between 0 and 31, a bit data channel is used. If "ParameterBit" is equal to "CYCLIC_ALLBITS", the entire parameter is applied. "ParameterNumber" specifies the parameter to be enabled.

In the first step, MB_FreeCyclicParameter checks whether the parameter is already in the current configuration. If this is not the case, an error message is output. Subsequently, the enabling is stored in an internal static variable. MB_FreeCyclicParameter does not change the actual configuration of the cyclic data channels. The enabling is only stored in the internal control structure. After a successful enabling of the cyclic data channel, the output "Done" is set to TRUE for at least one cycle. "Done" remains TRUE as long as the input "Execute" is active.

Error behavior

If the parameter is not applied in the cyclic data channel or if another error occurs, "Error" is set to TRUE.

Error codes

The function block generates the following error messages of the "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR	16#000D	16#0000	AxisRef is not in the valid range
ACCESS_ERROR	16#0010	16#0000	sercos phase 2 is not active
ACCESS_ERROR	16#1504	16#0000	AxisData structure not enabled
ACCESS_ERROR	16#F211	16#0000	Parameter not found in the cyclic data channel

Tab. 3-100: Error codes of the MB_FreeCyclicParameter function block

3.10.4 MB_GetCyclicChannelConfig

Brief description The MB_GetCyclicChannelConfig function block reads the configuration of the cyclic data channels of an axis.

Read channels and write channels are read and stored in a data structure.

The number of entries depends on the device and is output in an individual entry.



In version 14VRS, the "MB_GetCyclicChannelConfig (ML_TechBase library)" function block was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC/XLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-101: Reference table of the MB_GetCyclicChannelConfig

Interface description

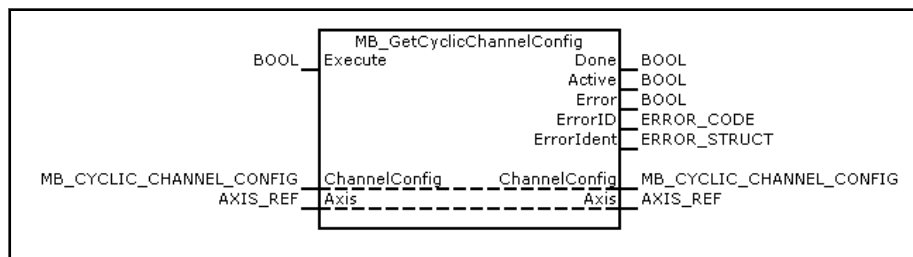


Fig. 3-81: MB_GetCyclicChannelConfig function block

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Starts the parameter configuration
VAR_OUTPUT	Done	BOOL	Parameter was applied
	Active	BOOL	The function block is active
	Error	BOOL	Indicates that an error occurred
	ErrorID	ERROR_CODE	Basic error cause
	ErrorIdent	ERROR_STRUCT	For the error structure with further error division, see Error codes, page 146

I/O type	Name	Data type	Description
VAR_IN_OUT	ChannelConfig	MB_CYCLIC_CHANNEL_CONFIG	Read-out configuration MB_CYCLIC_CHANNEL_CONFIG, page 153
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Parameter axis

Tab. 3-102: Interface variables of the MB_GetCyclicChannelConfig function block

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous

Tab. 3-103: Input behavior of the MB_GetCyclicChannelConfig

Functional description

This function block only operates in sercos phases 2 and 4. Cyclic data channels only exist for real axes. Thus, a function block can only be used for real axes. At a rising edge at "Execute", the created axis is checked and the configuration of the cyclic data channels is read. This operation requires several cycles. During this period, the "Active" output is set. "Active" is reset and "Done" is set after reading has been successfully completed. The configuration is now available at the "ChannelConfig" output. The "ChannelConfig" output is only valid after the "Done" output has been set.

The number of available data channels depends on the axis and is stored in the "...Count" elements of the "ChannelConfig" structure. The elements "ReadChannel", "WriteChannel", "ReadBitChannel" and "WriteBitChannel" contain the assigned parameter numbers. If no parameter is assigned, 0 is entered. The elements "ReadBitChannelBit" and "WriteBitChannelBit" contain the active bit number of the parameter.

Error behavior

If an error occurs during the processing, the outputs "Active" and "Done" are reset and "Error" is set. "ErrorID" and "ErrorIdent" contain more information on the error if the "Error" output is set.

Error codes

The function block generates the following error messages of the "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR	16#000D	16#0000	AxisRef is not in the valid range

Tab. 3-104: Error codes of the MB_GetCyclicChannelConfig function block

3.10.5 MB_GetCyclicParameterHandle

Brief description

This function block searches for a parameter to be cyclically read or written in the cyclic data channel of an axis and stores all required information in a handle.



In version 14VRS, the "MB_GetCyclicParameterHandle (ML_TechBase library)" function was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC/XLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-105: Reference table of the MB_GetCyclicParameterHandle

Interface description

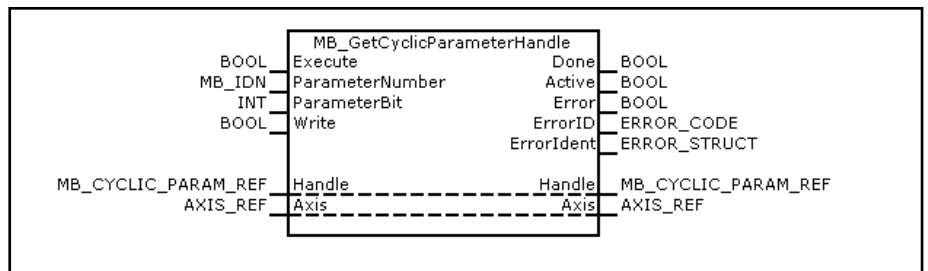


Fig. 3-82: MB_GetCyclicParameterHandle function block

I/O type	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Starts the parameter search
	ParameterNumber	MB_IDN	Number of the cyclically applied parameter
	ParameterBit	INT	Internal bit of the parameter; "CYCLIC_ALLBITS" for the entire parameter
	Write	BOOL	FALSE: Read parameter TRUE: Write parameter
VAR_OUTPUT	Done	BOOL	Parameter was found and "Handle" is valid
	Active	BOOL	The function block is active
	Error	BOOL	It indicates that an error occurred
	ErrorID	ERROR_CODE	Basic error cause
	ErrorIdent	ERROR_STRUCT	For the error structure with further error division, see Error codes, page 148
VAR_IN_OUT	Handle	MB_CYCLIC_PARAM_REF	Contains information on the access to the cyclic parameter MB_CYCLIC_PARAM_REF, page 151
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Parameter axis

Tab. 3-106: Interface variables of the MB_GetCyclicParameterHandle function block

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
ParameterNumber	MB_IDN	0	n.def.	0	Rising edge at "Execute"
ParameterBit	INT	-1	31	CYCLIC_ALLBITS	Rising edge at "Execute"
Write	BOOL			FALSE	Rising edge at "Execute"

Tab. 3-107: Input behavior of the MB_GetCyclicParameterHandle

Functional description

This function block is used in the operating mode (sercos phase 4). After a rising edge at "Execute", the inputs "ParameterNumber", "ParameterBit" and "Write" are read. The "Write" input decides whether the command values of the MDT or whether the actual values of the AT are searched. If "ParameterBit" is "CYCLIC_ALLBITS", the complete parameter is applied. If a bit number is specified in "ParameterBit", it is searched in the bit channel first. If the pa-

parameter is not found in the bit channels and if this parameter is an actual value, the parameter channels are browsed. After a successful processing, "Done" is active and the data in the created "handle" is valid.

Error behavior If the parameter is not found in the cyclic data channel or if another error occurs, "Error" is set to TRUE and "Handle" is not changed.

Error codes The function block generates the following error messages of the "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR	16#000D	16#0000	AxisRef is not in the valid range
ACCESS_ERROR	16#1504	16#0000	AxisData structure not enabled
ACCESS_ERROR	16#F211	16#0000	Parameter not found in the cyclic data channel
INPUT_INVALID_ERROR	16#F212	16#0000	Parameter type structure not supported
INPUT_INVALID_ERROR	16#F213	16#0000	Parameter type function not supported
INPUT_INVALID_ERROR	16#F214	16#0000	Parameter type not supported
INPUT_INVALID_ERROR	16#F215	16#0000	Cannot be written to the parameter

Tab. 3-108: Error codes of the MB_GetCyclicParameterHandle function block

3.10.6 MB_ReadCyclicParameter

Brief description This function reads a cyclic parameter via the "handle" and returns the current content as DINT. Values with decimal places are converted according to the scaling stored in the Sercos attribute. This function can also be used for bit values.

 In version 14VRS, the "MB_ReadCyclicParameter (ML_TechBase library)" function was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC/XLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-109: Reference table of the MB_ReadCyclicParameter

Interface description

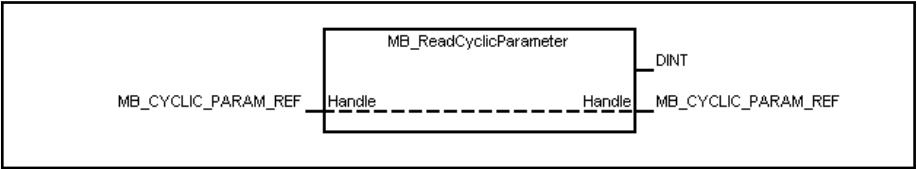


Fig. 3-83: MB_ReadCyclicParameter function

I/O type	Name	Type	Comment
VAR_IN_OUT	Handle	MB_CYCLIC_PARAM_REF	Contains information on the access to the cyclic parameter MB_CYCLIC_PARAM_REF , page 151
Return value		DINT	Parameter value

Tab. 3-110: Interface variables of the MB_ReadCyclicParameter function

3.10.7 MB_ReadCyclicRealParameter

Brief description This function reads a cyclic parameter via the "handle" and returns the current content as REAL. If the data type of the parameter is not REAL, the type is converted. This function cannot be used for bit values.



In version 14VRS, the "MB_ReadCyclicRealParameter (ML_TechBase library)" function was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC/XLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-111: Reference table of the MB_ReadCyclicRealParameter

Interface description

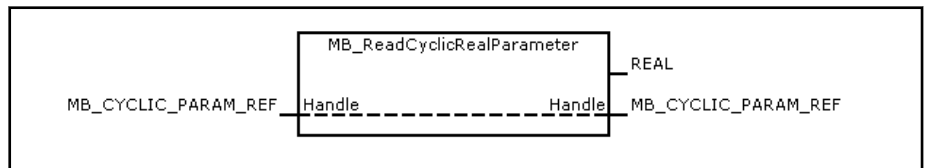


Fig. 3-84: MB_ReadCyclicRealParameter function

I/O type	Name	Type	Comment
VAR_IN_OUT	Handle	MB_CYCLIC_PARAM_REF	Contains information on the access to the cyclic parameter MB_CYCLIC_PARAM_REF , page 151
Return value		REAL	Parameter value

Tab. 3-112: Interface variables of the MB_ReadCyclicRealParameter function

3.10.8 MB_WriteCyclicParameter

Brief description This function writes a cyclic parameter via the "handle". If the target data type has decimal places, the value is converted according to the decimal places specified in the Sercos attribute. This function can also be used for bit values.



In version 14VRS, the "MB_WriteCyclicParameter (ML_TechBase library)" function was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC/XLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-113: Reference table of the MB_WriteCyclicParameter

Interface description

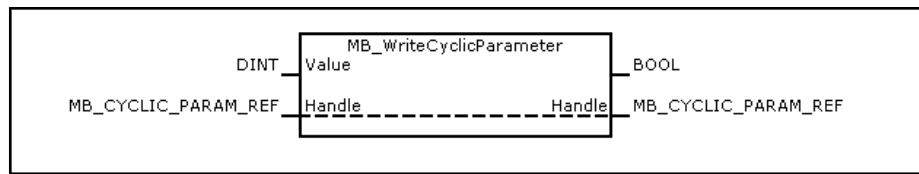


Fig. 3-85: MB_WriteCyclicParameter function

I/O type	Name	Type	Comment
VAR_INPUT	Value	DINT	Value to be written
VAR_IN_OUT	Handle	MB_CYCLIC_PARAM_REF	Contains information on the access to the cyclic parameter MB_CYCLIC_PARAM_REF, page 151
Return value		BOOL	Dummy output without meaning; is always TRUE

Tab. 3-114: Interface variables of the MB_WriteCyclicParameter function

3.10.9 MB_WriteCyclicRealParameter

Brief description This function writes a cyclic parameter via the "handle". Integer target data types without decimal places are rounded. This function cannot be used for bit values.



In version 14VRS, the "MB_WriteCyclicRealParameter (ML_TechBase library)" function was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC/XLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-115: Reference table of the MB_WriteCyclicRealParameter

Interface description

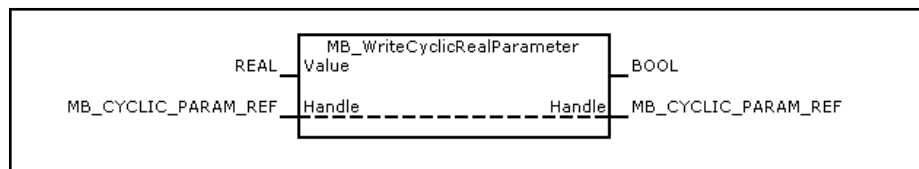


Fig. 3-86: MB_WriteCyclicRealParameter function

I/O type	Name	Type	Comment
VAR_INPUT	Value	REAL	Value to be written
VAR_IN_OUT	Handle	MB_CYCLIC_PARAM_REF	Contains information on the access to the cyclic parameter MB_CYCLIC_PARAM_REF, page 151
Return value		BOOL	Dummy output without meaning; is always TRUE

Tab. 3-116: Interface variables of the MB_WriteCyclicRealParameter function

3.10.10 MB_GetLastCyclicParameterError

Brief description This function returns the last error that occurred when reading or writing a cyclic value for a "handle" and deletes the error.

This function is optional and does not have to be called. However, it is recommended to check the cyclic data communication for occurred errors on a regular basis.



In version 14VRS, the "MB_GetLastCyclicParameterError (ML_TechBase library)" function was moved to the "TechMotionBase library".

Assignment target system/library

Target system	Library
MLC/XLC 14VRS MOT function package	ML_TechMotionBase
MLD (IL2G)	MX_TechMotionBase

Tab. 3-117: Reference table of the MB_GetLastCyclicParameterError

Interface description

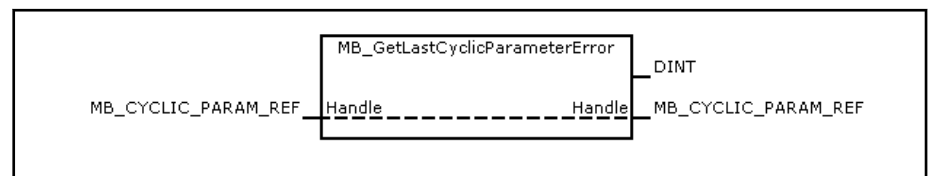


Fig. 3-87: MB_GetLastCyclicParameterError function

I/O type	Name	Type	Comment
VAR_IN_OUT	Handle	MB_CYCLIC_PARAM_REF	Contains information on the access to the cyclic parameter MB_CYCLIC_PARAM_REF , page 151
Return value		DINT	Error code of the last reading or writing action

Tab. 3-118: Interface variables of the MB_GetLastCyclicParameterError function

Error codes The MB_GetLastCyclicParameterError function generates the following errors:

ErrorID	Description
16#0001	Cannot write parameter - parameter only configured for reading
16#0002	Invalid type for bit access
16#0003	Invalid type for parameter access
16#0004	Invalid axis number
16#0005	Invalid data channel
16#0006	Invalid handle

Tab. 3-119: Error codes of the GetLastCyclicParameterError function

3.10.11 MB_CYCLIC_PARAM_REF

Type definitions The following definition describes data types used in the function block [MB_GetCyclicParameterHandle](#), page 146.

The type definitions are included in the library.

The structure "MB_CYCLIC_PARAM_REF" can be read but must not be written. Changing the content can lead to malfunctions. The structure can change in later versions. The content is only to be read for debugging purposes.

The structure "MB_CYCLIC_PARAM_REF" is as follows:

Structure element	Type	Description
ParameterNumber	MB_IDN	FP number of the parameter
AxisNo	OBJECTS	Logical axis number
ParamType	MB_CYCLIC_TYPES	Parameter data type MB_CYCLIC_TYPES, page 152
Bit	INT	Bit number in the parameter
AccessChannel	MB_CYCLIC_CHANNELS	Data channel in which the parameter was found MB_CYCLIC_CHANNELS, page 152
ParamAdr	POINTER TO SM_TYPES	Pointer to the cyclic parameter
BitParamAdr	POINTER TO WORD	Pointer to the cyclic bit parameter
Factor	DINT	Used to convert REAL to DINT
ErrorCode	DWORD	For the error code of the last error, see Error codes, page 151

Tab. 3-120: Definition of the basic elements of the MB_CYCLIC_PARAM_REF structure

3.10.12 MB_CYCLIC_TYPES

The data type "MB_CYCLIC_TYPES" is an enumeration type with the following entries:

Element	Value
tINT	1
tUINT	2
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: ENUM MB_CYCLIC_TYPES

3.10.13 MB_CYCLIC_CHANNELS

The data type "MB_CYCLIC_CHANNELS" is an enumeration type with the following entries:

Element	Value
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
UserActualDataBitA	288
UserActualDataBitB	289
UserActualDataBitC	290
UserActualDataBitD	291
ReadDV	511

Tab. 3-122: ENUM MB_CYCLIC_CHANNELS

3.10.14 MB_CYCLIC_CHANNEL_CONFIG

The "MB_CYCLIC_CHANNEL_CONFIG" structure is described by the [MB_GetCyclicChannelConfig](#), page 145, function block and contains the applied parameters in the cyclic data channel of the axis.

The structure "MB_CYCLIC_CHANNEL_CONFIG" is as follows:

Structure element	Type	Min. value	Max. value	Default value	Description
ReadChannelCount	INT	0	10	0	Number of read channels
WriteChannelCount	INT	0	10	0	Number of write channels
ReadBitChannelCount	INT	0	10	0	Number of bit read channels
WriteBitChannelCount	INT	0	10	0	Number of bit write channels
ReadChannel	ARRAY[1..10] OF MB_IDN	n.def.	n.def.	S-0-0000	Configured parameters of the read channels

ML_TechMotionBase.library // MX_TechMotionBase.library

Structure element	Type	Min. value	Max. value	Default value	Description
WriteChannel	ARRAY[1..10] OF MB_IDN	n.def.	n.def.	S-0-0000	Configured parameters of the write channels
ReadBitChannel	ARRAY[1..10] OF MB_IDN	n.def.	n.def.	S-0-0000	Configured parameters of the bit read channels
ReadBitChannelBit	ARRAY[1..10] OF UINT	0	63	0	Configured bits of the bit read channels
WriteBitChannel	ARRAY[1..10] OF MB_IDN	n.def.	n.def.	S-0-0000	Configured parameters of the bit write channels
WriteBitChannelBit	ARRAY[1..10] OF UINT	0	63	0	Configured bits of the bit write channels

Tab. 3-123: Definition of the MB_CYCLIC_CHANNEL_CONFIG structure

4 ML_TechMotion.library // MX_TechMotion

4.1 Overview on Function Blocks



The library requires the TEC (technology) function package for the control.

The function blocks of the ML_TechMotion

Field of application (folder in the library)	
Function blocks	Description
Cycle-synchronous lock-on and lock-off (CamLock)	
MB_CamLock, page 161	Used to enable the Run, LockOn and LockOff cam profiles calculated and stored by the MB_PrepareCams function block
MB_PrepareCams, page 159	Uses a polynomial 5th order to calculate and generate three cam profiles and to load them to parameters specified by the inputs MC_CAM_ID. The resulting MotionProfiles contain boundary conditions for position and velocity
FlyingShear	
ML_FlyingShear / MX_FlyingShear, page 169	Contains the typical procedure of a "FlyingShear"
MagicBelt	
MB_SmartBeltType01, page 227	Executes a phase correction at a product using a triangular velocity profile
SagControl	
ML_SagControlA, page 223	Contains a sag controller with analog actual value measurement of the sag
ML_SagControlC, page 219	Contains a sag controller with binary minimum and maximum contacts
SmartBelt	
MB_MagicBeltType01, page 182	Contains functionalities to control a chain of a MagicBelt application
Cam tables for technology functions	
MB_PilgrimStepCalcType01, page 250	Calculates a cam table with up to 1,024 data points, the distances of the forward and backward transport as well as stop positions for the pilgrim step operation.
MB_PrintLengthCorrCalcType01, page 260	Calculates a cam table with up to 1,024 data points for the technology function "print length correction" to change the length of a print on the printing material (web or shed)
TensionControl	
MB_TensionControlLoadCellType01, page 266	Depending on the measured value of a load cell, the tension controller controls the tension to the desired command value. The tension controller is parameterized directly via the controller parameter
MB_TensionControlLoadCellMatParType01, page 279	Depending on the measured value of a load cell, the tension controller controls the tension to the desired command value. The tension controller is parameterized using material parameters and controlled system parameters

Tab. 4-1: ML_TechMotion library

_PhaseOut The function blocks mentioned in the following (together with the associated data types) were moved to other libraries for organizational reasons.
This was done partially by modifying the name.

New library: ML_TechMotionBase as from 14.3.0.0

Folder (new)	Name (new)	POU type	Name (old)
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 which have been moved from ML_TechMotion to ML_TechMotionBase

4.2 CamLock - Cycle-synchronous lock-on and lock-off

4.2.1 Introduction and Overview

Starting with firmware MLC 03VRS, MLC controls were supplemented with the availability of CamLock technology function blocks regarding the "Lock On/Lock Off" functionality.

The function block [MB_PrepareCams](#), [page 159](#), uses a polynomial 5th order to calculate, build and download three cam profiles to a slave axis. These cam profiles are designated as Lock On, Lock Off and one-to-one. They allow a synchronized slave axis to disengage (Lock Off) from its master and stop at a predefined position until it is once again synchronized (Lock On) to the master.

The function block [MB_CamLock](#), [page 161](#), is used to activate the Lock On/Lock Off functionality. A user cam profile is also supported.

As long as a slave axis is synchronized to the master, it follows the master using the one-to-one cam.

A phase offset between the master and slave position can be defined.

When the Lock Off cam is enabled (via the MB_CamLock function block), the slave axis transitions off of the master position to a predefined lock off position and comes to a stop.

When the Lock Off cam is disabled (Lock On), the slave axis transitions from its stopped position and re-synchronizes back to the master. Once synchronized, the slave axis switches back to the One-to-One cam and continues following the master input position.

4.2.2 CamLock - Application example

Normally, this functionality is used in packaging machines, provided that the incoming products are spaced equally before they can be packaged.

If the distance is too large, the packaging process in the machine is locked off for one or several cycles by the master until the product is detected. This state is called "No Product, No Seal". As soon as the product is detected, the packaging process is re-synchronized to the master (Lock On) and the packaging of the products continues.

These unequal distances require the slave axis to be accelerated and decelerated in order to synchronize it to the master. The following figure shows a typical packaging machine for filling and sealing:

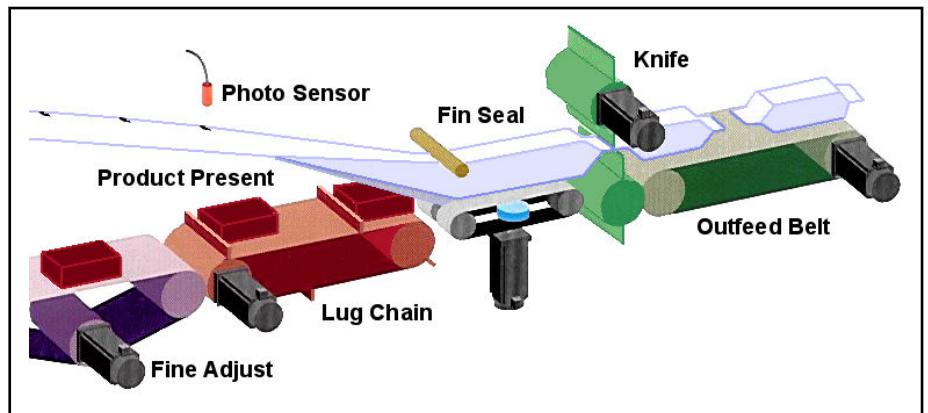


Fig. 4-1: Horizontal version of a packaging machine for filling and sealing

The following diagram shows the processes of processing, slowing down, stopping, accelerating, and re-processing a slave axis follows during the Lock On/Lock Off process.

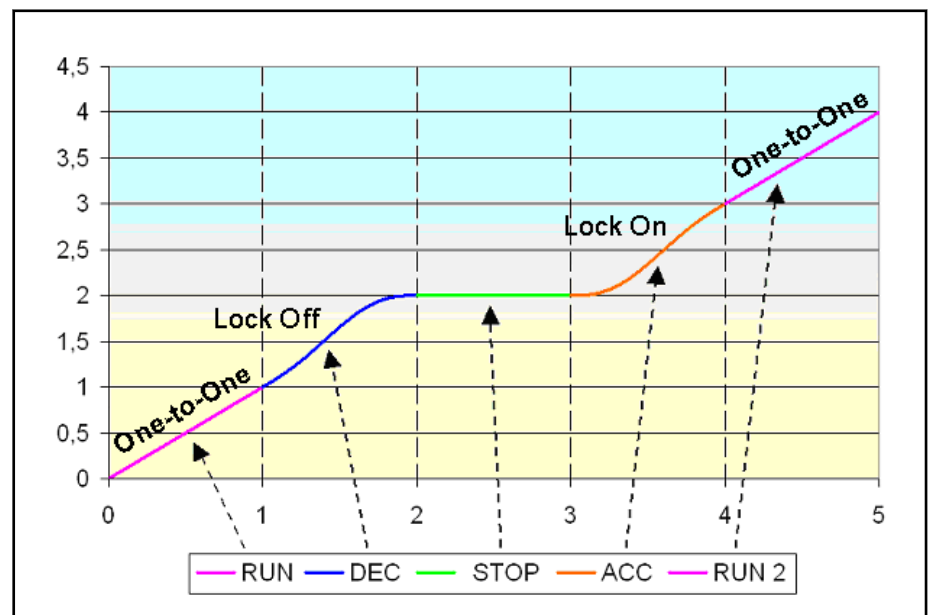


Fig. 4-2: CamLock cam profile

One-to-one cam profile "Synchronized to master"

The one-to-one cam profile is active and synchronized to the master input, if the Lock On/Lock Off functionality is inactive. The Lock On/Lock Off functionality is activated by setting the "Enable" input of the MB_CamLock function block to "High". The Lock Off input has to be set to "Low". In normal working conditions, this cam profile is active and follows the specifications of the master input.

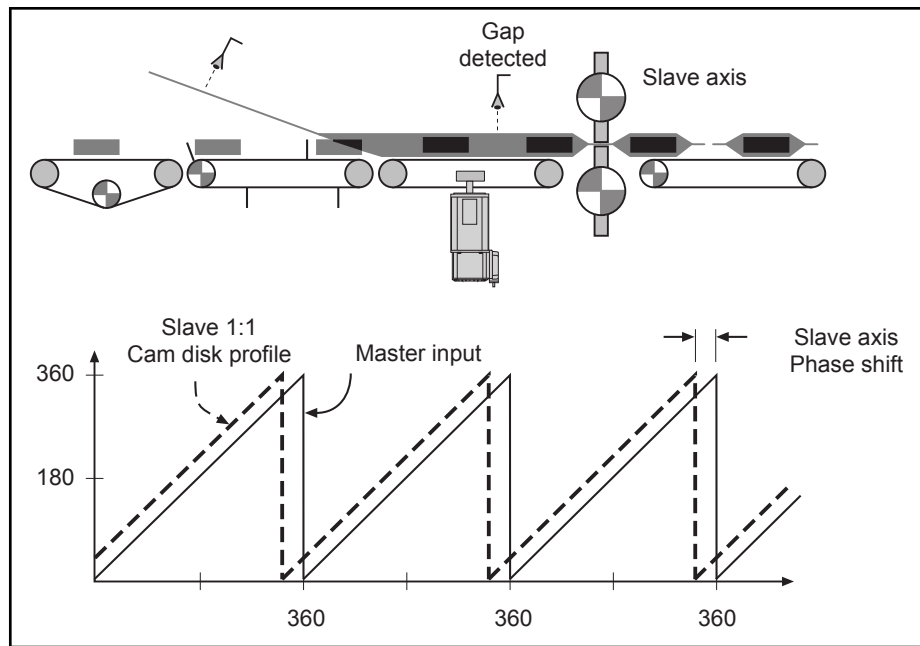


Fig. 4-3: Processing cam active, normal operation of the packaging application

Lock Off cam profile

The Lock Off cam profile slows down the slave axis up to a complete halt via a cycle of the master. The Lock Off cam profile is active as long as both inputs "Enable" and "LockOff" of the MB_CamLock function block are set to "High". After this cycle the velocity of the slave axis is stopped and is only re-started when the Lock On cam profile is active.



Once activated, the Lock Off cam profile is active upon zero passage of the slave axis.

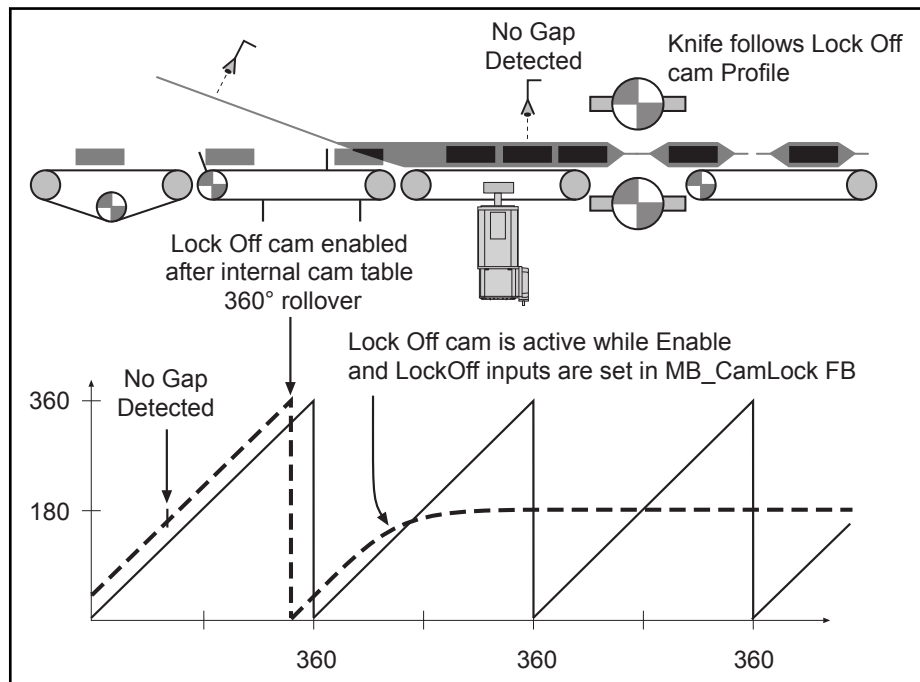


Fig. 4-4: Lock Off cam active, No Product - No Seal

Lock On cam profile The Lock On cam profile accelerates from a stop position until the velocity of the master input is reached during a cycle of the master (360°). After this cycle, the slave axis and the master have the same velocity. The Lock On cam is active as long as the slave axis is synchronized with the master. Afterwards, the slave axis follows the one-to-one cam during normal operation. The Lock On cam profile is active as long as the "Enable" input is set to "High" and the "LockOff" input is set to "Low" at the MB_CamLock function block.



Once activated, the Lock On cam profile is active upon zero passage of the master axis.

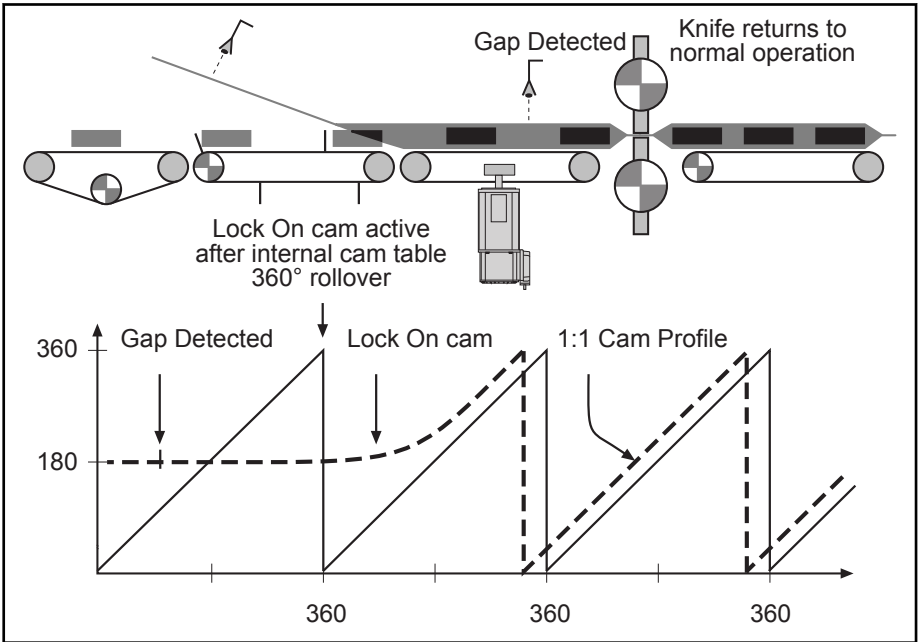


Fig. 4-5: Lock On cam active, product present

4.2.3 MB_PrepareCams

Brief description The function block uses a polynomial 5th order to calculate and to generate 3 cam profiles and to load them to parameters specified by the inputs MC_CAM_ID.

The resulting MotionProfiles contain boundary conditions for position and velocity.

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechMotion

Tab. 4-3: Reference table of the MB_PrepareCams

Interface description

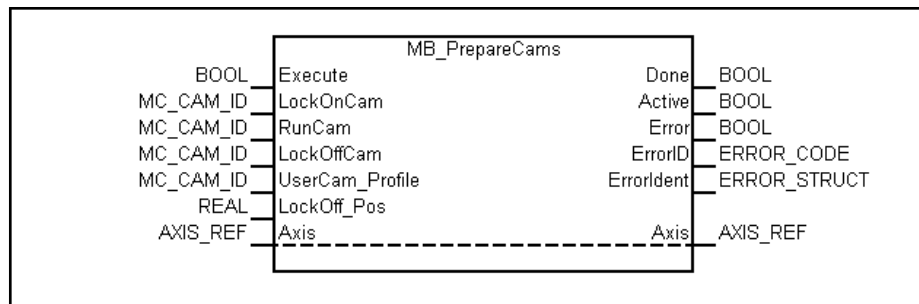


Fig. 4-6: MB_PrepareCams function block



The MC_CAM_ID value only specifies which parameter is used to save the calculated cam profile. The current functionality (e.g., LockOnCam) is determined by the input in the function block.

For an MLC control system, the cam profile is saved in the control parameter C-0-2001 if the value is 1. However, in order to execute the function block MB_PrepareCams without any errors, the axis input for "Axis" is still required.

The calculated cam profiles are stored in control parameters.

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	The positive edge starts the calculation of three cam profiles
	LockOnCam	MC_CAM_ID	Determines the target address of the Lock On cam table. Default value is 4
	RunCam	MC_CAM_ID	Determines the target address of the editing cam table. Default value is 3
	LockOffCam	MC_CAM_ID	Determines the target address of the Lock Off cam table. Default value is 2
	LockOff_Pos	REAL	This input is used to calculate the respective velocity profile
VAR_OUTPUT	Done	BOOL	Three cam profiles have been calculated, built and loaded
	Active	BOOL	Function block is active
	Error	BOOL	Indicates that an error occurred
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Error handling, page 161,
VAR_IN_OUT CONSTANT	Axis	AXIS_REF	Reference to the slave axis for which cam profiles are saved

Tab. 4-4: Interface variables of the MB_PrepareCams function block

Functional description

MLC supports a block of 98 control parameters to store CamLock cam profiles.

Unlike drive parameters, the assignment between the MC_CAM_ID value and the control parameter is continuous.

Starting with control parameter C-0-2001, an MC_CAM_ID value of 1 is saved in the control parameter C-0-2001, an MC_CAM_ID value of 98 in C-0-2098.

The control parameter C-0-2099 cannot be used to save a cam profile. This parameter is reserved to "stop the slave axis".

This function block has to run right after control startup to calculate the three cam profiles and download them to the relevant parameters. Use the "DONE" output to check that this procedure has been completed before the PLC program continues running.



The MB_PrepareCams function block is executed only once at the start of the PLC program and before the start of function block [MB_CamLock, page 161](#).

Error handling

The function block generates the following error messages in Additional1/ Additional2 for the table **F_RELATED_TABLE**:

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR	16#0003	16#0000	Function block was aborted by another function block
RESOURCE_ERROR	16#0004	16#0000	This drive firmware version is not supported
INPUT_RANGE_ERROR	16#13A1	16#0001	The cam-related values were initialized improperly
INPUT_RANGE_ERROR	16#13A1	16#0002	Slave Axis_Ref; the AxisNo is not in the admissible range
INPUT_RANGE_ERROR	16#13A1	16#0003	LockOff_Pos has to be larger than 0 and smaller than 360; default = 180 degrees
CALCULATION_ERROR	16#13A2	16#0000	Calculation of the step width, result = 0
STATE_MACHINE_ERROR	16#0006	16#0000	Invalid state of the state machine

Tab. 4-5: Error codes of the MB_PrepareCams function block

4.2.4 MB_CamLock

Brief description

The MB_CamLock function block is used to enable the Run, Lock On, and Lock Off cam profiles calculated and stored by function block [MB_PrepareCams, page 159](#).

In addition to the activation of the cam profiles, this function block also provides the following functionality:

- Electronic gear ratio
- Direction of synchronization (SyncMode)
- MC_CamIn and MC_CamOut functionality for drive cams
- Master fine adjustment



The function block operation requires the cyclic call in the Motion task. Note the parameterization in [MB_CamLock Parameterization, page 168](#).

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechMotion

Tab. 4-6: Reference table of the MB_CamLock

Interface description

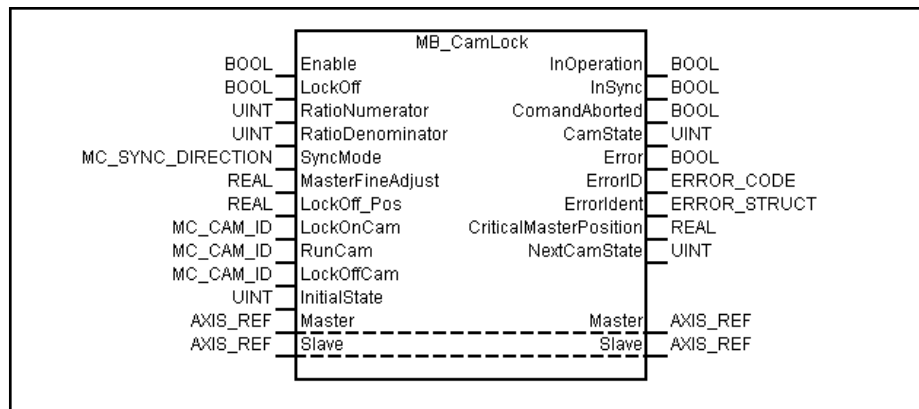


Fig. 4-7: MB_CamLock function block



For MLC control systems, control cams can be used to assign a slave input as virtual axis.

The following table lists the different cam profiles controlled by the MB_CamLock function block:

Cam profile	"Enable" input	"Lock Off" input	Description
Run	High	Low	Normal mode using a 1:1 cam profile
LockOff	High	Low to high	The slave axis switches from the "Run" cam to the "LockOff" cam profile and stops at the position specified in the "LockOff_Pos" input
LockOn	High	High to low	The slave axis accelerates from a stopped position using the "LockOn" cam profile, synchronizes with the master and switches to the "Run" cam

Tab. 4-7: Normal "CamLock" operation

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enables the MB_CamLock function block. Due to an increasing edge, the slave is synchronized dynamically before entering the "Run" state. A falling edge executes a gear lock off
	LockOff	BOOL	TRUE: Locked off FALSE: Locked on If this input is TRUE before setting "Enable", the axis dynamically synchronizes, enters the "Run" state and and locks off (LockOff)
	RatioNumerator	UINT	Electronic gear ratio
	RatioDenominator	UINT	Electronic gear ratio
	SyncMode	MC_SYNC_DIRECTION	0: = shortest distance 1: = positive direction 2: = negative direction
	MasterFineAdjust	REAL	This input is required for the MB_MotionProfile function block

I/O type	Name	Data type	Description
	LockOff_Pos	REAL	The position in degrees where the slave axis stops when locked off from the master. The default values is 180 degrees
	LockOnCam	MC_CAM_ID	Determines the source of the LockOn cam table. Default value is 4
	RunCam	MC_CAM_ID	Determines the source of supply of the "Run" cam table. Default value is 3
	LockOffCam	MC_CAM_ID	Determines the source of supply of the "LockOff" cam table. Default value is 2
	InitialState	UINT	Initial state of the state machine
VAR_OUTPUT	InOperation	BOOL	Is TRUE as long as the function block is enabled
	InSync	BOOL	The "InSync" output is TRUE when the "Run" cam is in synchronized state.
	CommandAborted	BOOL	Command aborted by another function block
	CamState	UINT	Current state of the "CamLock" function block 1: in Dynamic Sync (dynamic synchronization) 2: in Run (running state) 3: in LockOff (locked off state) 4: at Standstill (stop state) 5: in LockOn (locked on state) 6: in Gear Out [gear lock out (continuous motion)]
	Error	BOOL	Indicates that an error occurred
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Error handling, page 167
	CriticalMasterPosition	REAL	Provides the critical master position required for the switching state transition
	NextCamState	UINT	Indicates the next output to be processed
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Real or virtual axis
	Master	AXIS_REF	Real or virtual axis

Tab. 4-8: Interface variables of the MB_CamLock function block

Signal-time diagram

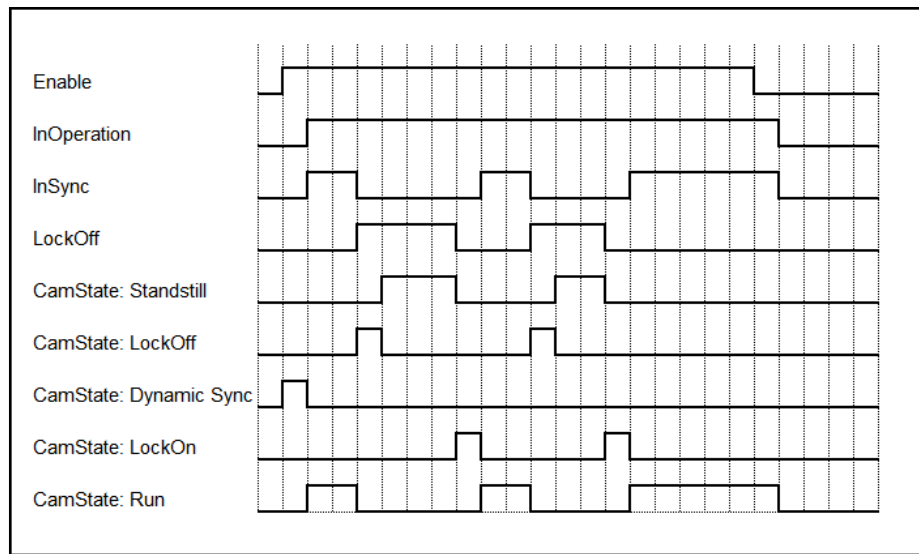


Fig. 4-8: Signal-time diagram of the MB_CamLock

Functional description

If the **Enable** input is set to "High" and the **LockOff** input is set to "Low", the slave axis synchronizes dynamically with the master and switches immediately to the "Run" state. The **CamState** output changes from "Dynamic Sync" (1) to "Run" state (2). The function block remains in the "Run" state (editing a 1:1 cam) until the **LockOff** input is "High".

If the **LockOff** input becomes "High" during editing, the function block executes a "LockOff" cam and stops at the position specified in the **LockOff_Pos** input. During the transition from "Run" to "Stop", the **CamState** output goes from Run (2) to LockOff (3) and then to Standstill (4). The default value for **LockOff_Pos** is 180 degrees. This is half the distance from "LockOn" to "Run" and from "Run" to "LockOff".

At this point, the function block remains in the stop state (standstill) until the **LockOff** input becomes "Low". This completes the cycle. To disable the "CamLock" function block, it is recommended to be in **CamState=4** (stopping state) and to set the "Enable" input to "Low".

If the **Enable** input is set to "Low" in the "Run" state, the slave desynchronizes from the LockOn/LockOff cam functionality and switches to continuous motion.

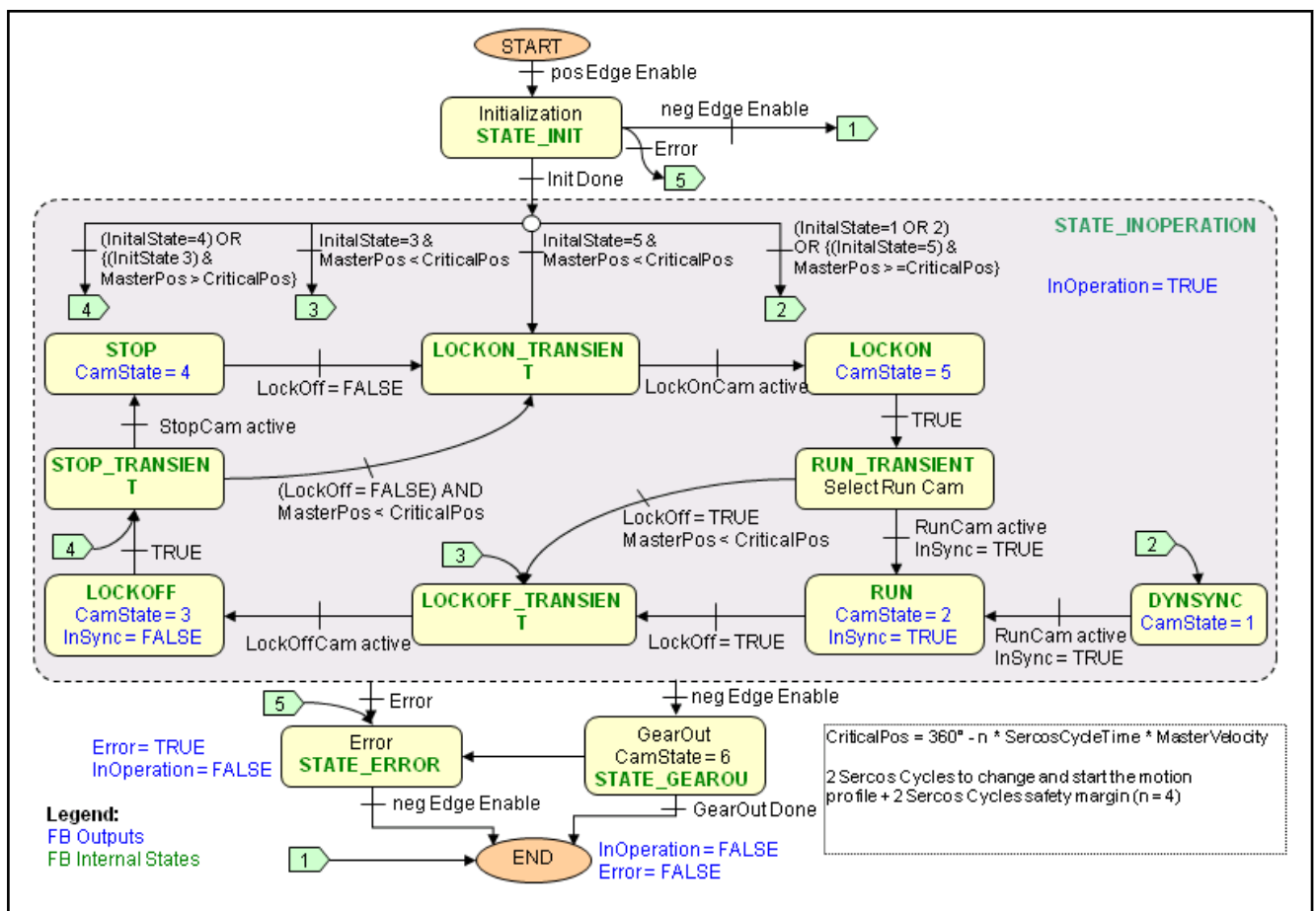
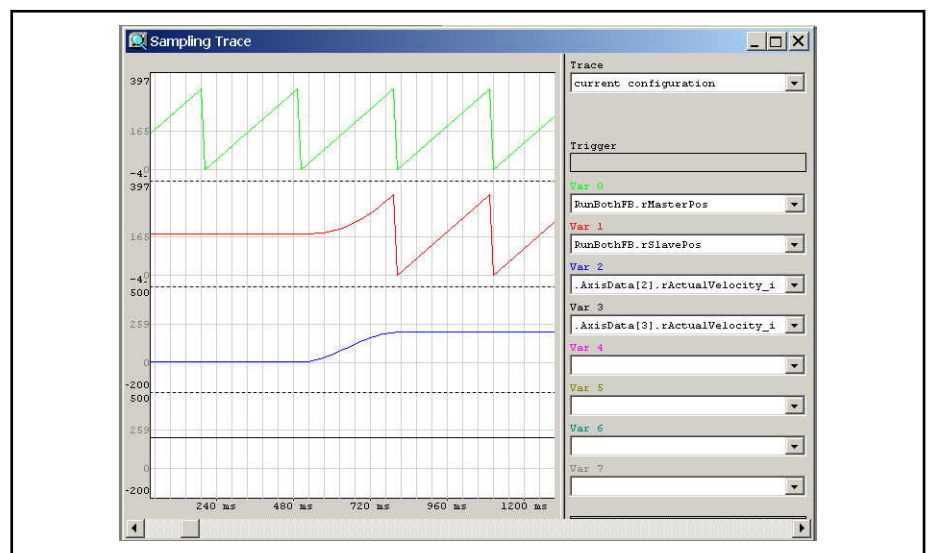


Fig. 4-9: CamLock state machine

LockOn/LockOff sequence examples



Channel 0 (green) = master position
 Channel 1 (red) = slave position
 Channel 2 (blue) = slave velocity
 Master velocity = 200 rpm

Fig. 4-10: "LockOn" profile with a LockOff_Pos of 180 degrees

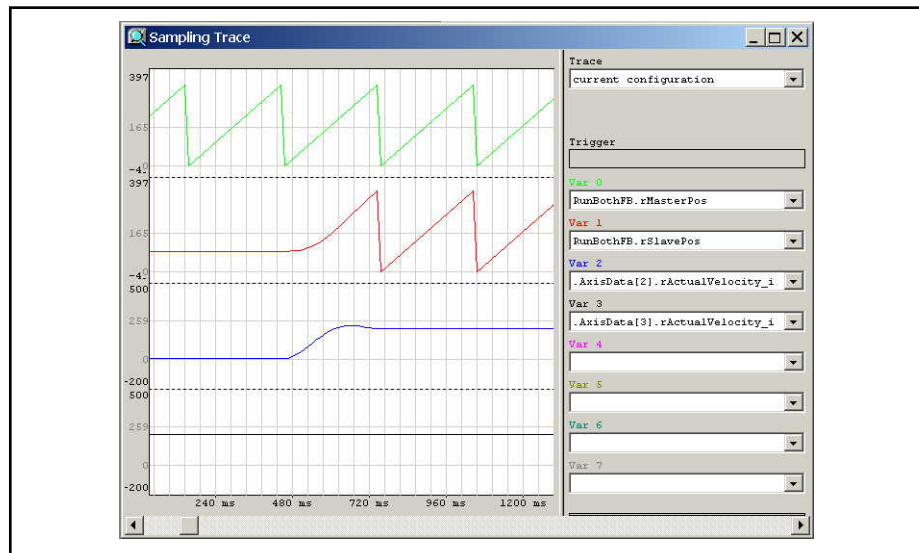
The figure above shows the position and velocity profile during the "LockOn" profile. The LockOff_Pos position is 180 degrees (default). The figure above shows the position and velocity profile during the "LockOff" profile without jumps and overshoots. The "LockOff" position is 180 degrees (default).



Channel 0 (green) = master position
 Channel 1 (red) = slave position
 Channel 2 (blue) = slave velocity
 Master velocity = 200 rpm

Fig. 4-11: "LockOff" profile with a LockOff_Pos of 180 degrees

The figure above shows the position and velocity profile during the "LockOff" profile. The LockOff_Pos position is 180 degrees (default). The figure above shows the position and velocity profile during the "LockOff" profile without jumps and overshoots. The "LockOff" position is 180 degrees (default).

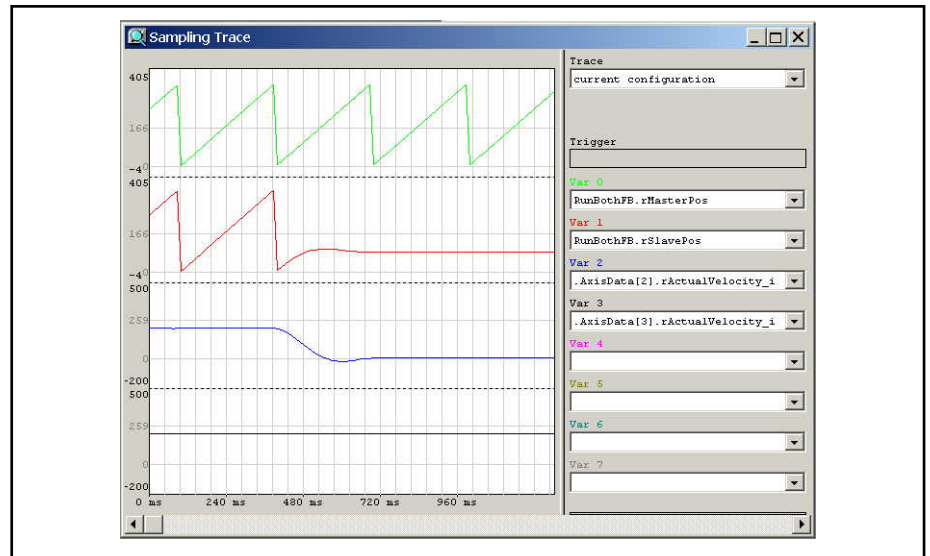


Channel 0 (green) = master position
 Channel 1 (red) = slave position
 Channel 2 (blue) = slave velocity
 Master velocity = 200 rpm

Fig. 4-12: "LockOn" profile with a LockOff_Pos of 90 degrees

The figure above shows the position and velocity profile during the "LockOn" profile if the LockOff_Pos is 90 degrees. The position and velocity profiles are

still ok for a "LockOff" position of 90 degrees (there is a velocity overshoot, but no backward motion).



Channel 0 (green) = master position
 Channel 1 (red) = slave position
 Channel 2 (blue) = slave velocity
 Master velocity = 200 rpm

Fig. 4-13: "LockOff" profile with a LockOff_Pos of 90 degrees

The figure above shows the position and velocity profile during the "LockOff" profile if the LockOff position is 90 degrees. The position and velocity profiles are provided with an overshoot for a "LockOff" position of 90 degrees. This position overshoot causes a backward motion.



It is recommended to use a value of 180 for the LockOff_Pos input (this input has to match for the MB_PrepareCams and the MB_CamLock function blocks).

If a "LockOff" position other than 180 degrees is used, the, the farer from 180 degrees, the more overshoots in position and velocity occur.

Error handling The function block generates the following error messages in Additional1/ Additional2 for the table **F_RELATED_TABLE**:

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR	16#0001	16#0000	Power for the slave axis is not ON
RESOURCE_ERROR	16#0003	16#0000	Function block was aborted by another function block
STATE_MACHINE_ERROR	16#0006	16#0000	Invalid state of the state machine
ACCESS_ERROR	16#0011	16#0000	Control is not in phase BB (Sercos P4)
INPUT_RANGE_ERROR	16#1301	16#0001	LockOff_Pos has to be greater than 0 and lower than 360, default = 180 degrees
INPUT_RANGE_ERROR	16#1301	16#0002	InitialState is not in the range 1...5
INPUT_RANGE_ERROR	16#1301	16#0003	RatioNumerator <= 0
INPUT_RANGE_ERROR	16#1301	16#0004	RatioDenominator <= 0
INPUT_RANGE_ERROR	16#1301	16#0005	SyncMode < 0 OR SyncMode > 2

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR	16#1301	16#0006	LockOnCam < 1 OR LockOnCam > 98
INPUT_RANGE_ERROR	16#1301	16#0007	RunCam < 1 OR RunCam > 98
INPUT_RANGE_ERROR	16#1301	16#0008	LockOffCam < 1 OR LockOffCam > 98
INPUT_RANGE_ERROR	16#1301	16#0009	UserCam_Profile < 1 OR UserCam_Profile > 98
INPUT_RANGE_ERROR	16#1301	16#0010	"CamTable" inputs are equal
ACCESS_ERROR	16#1302	16#0003	The axis modulo value of the slave is not 360°
ACCESS_ERROR	16#1302	16#0004	The master axis modulo value of the slave is not 360°

Tab. 4-9: Error codes MB_CamLock function block

4.2.5 MB_CamLock Parameterization

The function block [MB_CamLock, page 161](#), requires the following parameterization:

1. The "Interpolation in the drive" checkbox has to be deselected when adding a real axis in IndraWorks if this axis is used as slave axis of the MB_CamLock.
2. Master and slave axis have to be set to the modulo format 360°.
3. The drive has to be referenced before using this function block (absolute position preferred).
4. The FB [MB_PrepareCams, page 159](#), has to be executed once to calculate and store the three cams required.
 - LockOnCam, stored in CAM_TABLE_4 by default
 - RunCam, stored in CAM_TABLE_3 by default
 - LockOffCam, stored in CAM_TABLE_2 by default
 - User_Cam_Profile, stored in CAM_TABLE_1 by default)
5. The power has to be added to the drive (drive in "Enabled" state).
6. The MB_CamLock contains time-critical Motion functionality and should thus be called in the MotionTask.
7. The function block only operates for one master axis moving in positive direction.



The CamID 99 is always used as stop Cam.

4.3 FlyingShear

4.3.1 Introduction and Overview

With the application "FlyingShear", a machining station is synchronized with regard to a continuously passing material (like, for example, metal, plastic, etc.), and the machining is executed in the synchronous run of the machining station.

The material position is usually detected via a measuring wheel and transferred via an optional encoder input to the drive controller. Subsequently, the drive controller synchronizes the connected servo motor on the continuously blending through material when the specified cut length was blending through.

The function block [ML_FlyingShear / MX_FlyingShear](#), page 169, is responsible for the synchronization and sets the output "InSync" once the machining station is running synchronously.

In this state, the material can be machined since now the machining station runs synchronously to the material.

After having finished the machining process, the synchronous run can be interrupted with the input "MoveReturn", and the machining station returns to its specified starting position.

For test purposes, a virtual axis can be used instead of the measuring wheel. With this virtual axis, it is then possible to simulate the measuring wheel.



There are different function blocks for the target systems for the "FlyingShear" functionality:

- Target system MLD: MX_FlyingShear function block
- Target system MLC: ML_FlyingShear function block

The following description refers to both function blocks.

For the system-specific settings, refer to

- Target system MLD: See [MX_FlyingShear - Special Features of MLD](#), page 177
- Target system MLC: See [ML_FlyingShear - Special Features of MLC](#), page 176

4.3.2 ML_FlyingShear / MX_FlyingShear

Brief description

The ML_FlyingShear/MX_FlyingShear function block contains the typical procedure of a FlyingShear and performs the following steps once the input "Start" is set to TRUE:

- Move the slave axis to the starting position "ReturnPos" and wait until the axis is in position
- Synchronize the slave axis on the master axis via a Lock On cam profile
- Set the output "InSync" once the slave axis (machining station) was synchronized on the material
- Return the slave axis to its starting position if the corresponding signal is set at the input "MoveReturn"



The function block operation required special boundary condition (e.g. interpolation in the drive, cyclic function block call in the Motion task...).

For the target system MLD, note the commissioning notes in [MX_FlyingShear - Special Features of MLD](#), page 177.

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechMotion
MLD with MPx07/08/17 with MA function package	MX_Technology07.lib

Tab. 4-10: Reference table for ML_FlyingShear / MX_FlyingShear

Interface description

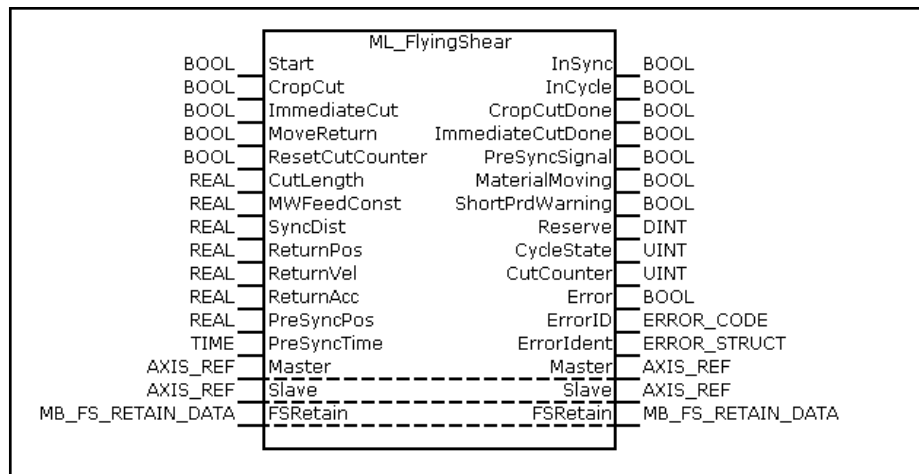


Fig. 4-14: ML_FlyingShear / MX_FlyingShear function block

I/O type	Name	Data type	Explanation
VAR_INPUT	Start	BOOL	Start=TRUE: Starts the FlyingShear function block. Depending on the states of the inputs "ImmediateCut" and "CropCut", the function block performs the following functions: Start=TRUE and positive edge at the input "ImmediateCut": The machining station is synchronized immediately and performs the cutting length (Cutlength) once the immediate cut (ImmediateCut) has been performed. Start=TRUE and positive edge at the input "CropCut": The machining station is synchronized after the cutting length (Cutlength) has passed the machine and performs the cutting length (Cutlength) once the crop cut (CropCut) has been performed. Start=TRUE and no signal "ImmediateCut" and "CropCut": The machining station is synchronized after the cutting length (Cutlength) has passed the machine (measured from the last cut)
	CropCut	BOOL	At a rising edge, the slave axis starts the synchronization process after the cutting length has passed the machine. If a cut cycle is currently performed (InCycle=TRUE), the crop cut is performed in the next cut cycle

I/O type	Name	Data type	Explanation
	ImmediateCut	BOOL	Start = TRUE: This case is intended for an immediate cut on the running material. The immediate cut (ImmediateCut) is activated by a positive edge. The slave axis is synchronized immediately. If a cut is currently performed (InCycle = TRUE), the immediate cut is performed in the next cut cycle. If (Start = FALSE) and ($V_{\text{measuring\_wheel}} < 5 \text{ rpm}$): This case is intended for an immediate cut material standstill. The immediate cut (ImmediateCut) is activated by a positive edge. The slave axis remains in its standstill position and is not synchronized. The output "InSync" is set immediately, and the function block expects the input signal "MoveReturn" to finish the process. After having finished this process, the Flying-Shear is referenced to the material
	MoveReturn	BOOL	Completes the synchronous run of the slave axis with the material and returns the slave axis to its starting position "ReturnPos"
	ResetCutCounter	BOOL	The positive edge resets the cut counter
	Cutlength	REAL	Specified material cutting length ^{*1*2}
	MWFeedconst	REAL	Feed constant of the measuring wheel per revolution ^{*1 *2}
	SyncDist	REAL	Synchronization distance required by the slave axis to synchronize the master axis ^{*1*2} . The shorter the synchronization distance, the higher the synchronization acceleration.
	ReturnPos	REAL	The slave axis travels to this starting position at the beginning and after the synchronization, and the input MoveReturn is set to TRUE ^{*1*2}
	ReturnVel	REAL	The slave axis travels to the starting position "ReturnPos" ^{*1*2} with the velocity ReturnVel
	ReturnAcc	REAL	The slave axis returns to the starting position "ReturnPos" ^{*1*1} with the acceleration ReturnAcc
	PreSyncPos	REAL	To prepare for the synchronous run, the output signal "PreSyncSignal" is set before the synchronous position "PreSyncPos" with the distance given here (TRUE) ^{*1}
	PreSyncTime	TIME	Duration of "PreSyncSignal"
VAR_OUTPUT	InSync	BOOL	The slave axis runs synchronously with the material. It is in its synchronization window (see drive parameter S-0-0228)
	InCycle	BOOL	The slave axis currently performs a cut cycle. An immediate or crop cut is performed in the next cycle (CutCycle = 0)
	CropCutDone	BOOL	The crop cut has been performed

I/O type	Name	Data type	Explanation
	ImmediateCutDone	BOOL	The immediate cut has been performed
	PreSyncSignal	BOOL	First marking before the synchronous position "Pre-SyncPos"
	ShortPrdWarning	BOOL	The slave axis does not have enough time to reach the starting position "ReturnPos" for the next cut. The return is interrupted and the synchronization ramp starts again. This case should be prevented since the drive might be overloaded and the machining station might move to the carriage bed end. Countermeasures: • Increase return velocity (function block input "ReturnVel") • Increase return acceleration (function block input "ReturnAcc") • Reduce synchronization distance (function block input "SyncDist") • Reduce material feed velocity
	MaterialMoving	BOOL	The speed of the material encoder is higher than 5 rpm. An immediate cut in the case of a material standstill is only possible if MaterialMoving=FALSE
	Reserve	DINT	Reserved increments of the synchronization procedure for analysis purposes. Synchronization is not possible if Reserve ≤ 0 -> In this case, the function block returns an error message
	CycleState	UINT	Current cut cycle state: 0: Standstill and waiting phase 1: Acceleration phase 2: Synchronous run phase 3: Return phase
	CutCounter	UINT	With each cut, the cut counter "CutCounter" is increased. With (Start = FALSE) or a positive edge at the input "ResetCutCounter", the counter is reset
	Error	BOOL	Indicates an error. Cleared if Start = FALSE
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Error handling, page 174
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Reference to the master axis
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Reference to the slave axis (machining station)
VAR_IN_OUT	FSRetain	MB_FS_RETAIN_DATA	Reference to the required retain data of this function block MB_FS_RETAIN_DATA, page 176

*1:

Measuring units according to drive scaling, e. g. mm

*2: New input values are transferred with the transition from the synchronous run phase to the return phase. Due to this reason, a new cutting length which is to be activated in the next cycle has to be copied to the function block input before the axis is moved to its return position.

Tab. 4-11: Interface variables of the ML_FlyingShear / MX_FlyingShear function block

Signal-time diagram The following diagram shows the signal flow with immediate cut and stopped material (Start = FALSE):

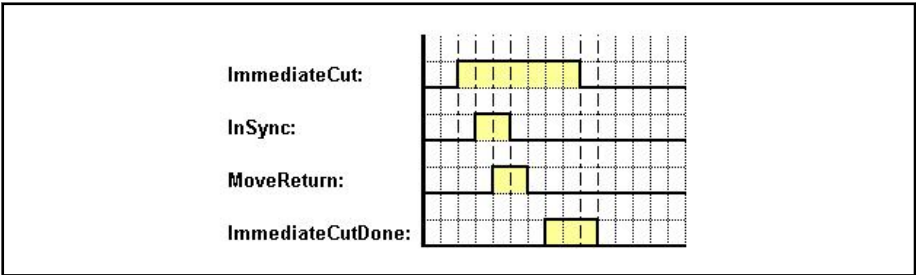


Fig. 4-15: Signal-time diagram: Immediate cut at material standstill

The following chart shows the complete signal flow of the function block "FlyingShear" with continuously passing material and immediate cut:

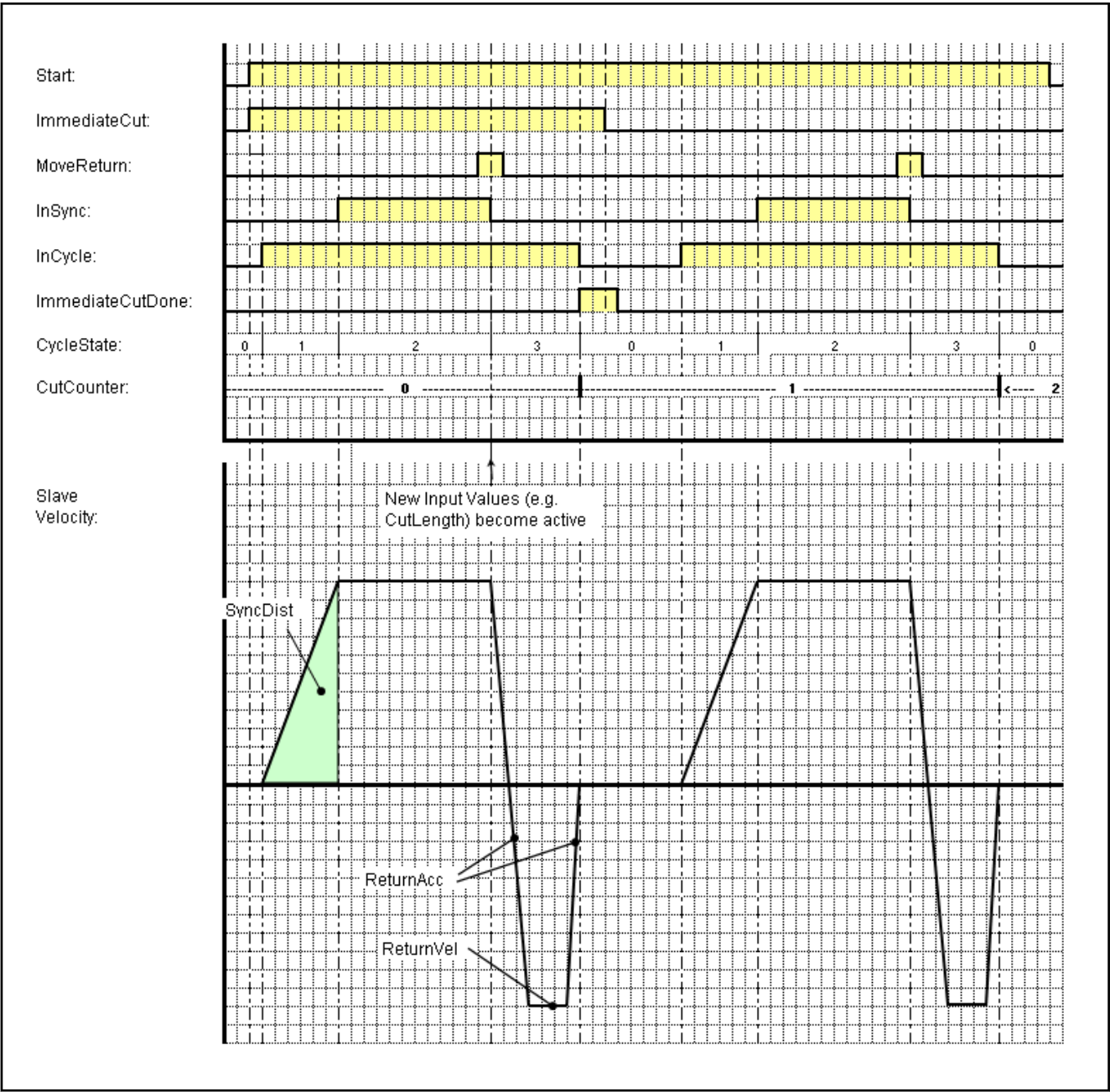


Fig. 4-16: Complete signal flow of the function block "FlyingShear" with immediate cut

Error handling The ML_FlyingShear / MX_FlyingShear function block generates the following error messages in Additional1/Additional2 for the "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Drive not enabled or drive error
ACCESS_ERROR (16#0004)	16#0003	16#0000	The function block was aborted by another function block
ACCESS_ERROR (16#0004)	16#0004	16#0000	Not supported drive firmware

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis (Axis_Ref) was changed during the function block processing
INPUT_RANGE_ERROR (16#0006)	16#0201	16#0001	CutLength <= 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	Synchronous position is too far away. Thus, synchronization is not possible. Reset error and initiate immediate cut
ACCESS_ERROR (16#0004)	16#0203	16#0001	S-0-0048 is not configured in the optional cyclical MDT channel of the slave axis
ACCESS_ERROR (16#0004)	16#0203	16#0002	P-0-0054 is not configured in the optional cyclical MDT channel of the slave axis
ACCESS_ERROR (16#0004)	16#0203	16#0003	P-0-0691 is not configured in the optional cyclical MDT channel of the slave axis
ACCESS_ERROR (16#0004)	16#0203	16#0004	P-0-0692 is not configured in the optional cyclical MDT channel of the slave axis
ACCESS_ERROR (16#0004)	16#0203	16#0005	P-0-0052 is not configured in the optional cyclical AT channel of the slave axis
ACCESS_ERROR (16#0004)	16#0203	16#0006	P-0-0053 is not configured in the optional cyclical AT channel of the slave axis
ACCESS_ERROR (16#0004)	16#0203	16#0007	P-0-0761 is not configured in the optional cyclical AT channel of the slave axis
ACCESS_ERROR (16#0004)	16#0203	16#0008	P-0-1367 bit 6 is not TRUE (the structure Axis-Data is not updated).
OTHER_ERROR (16#00FE)	16#0204	16#0001	The ELS configuration word (P-0-0086) of the slave axis has been configured incorrectly. The configuration has to be: P-0-0086 = 2#xxxx01000
OTHER_ERROR (16#00FE)	16#0204	16#0002	The parameter P-0-0060 of the slave axis is configured incorrectly. The value has to be: P-0-0060=0
OTHER_ERROR (16#00FE)	16#0204	16#0003	The parameter P-0-0693 of the slave axis is configured incorrectly. The value has to be: P-0-0693=0
OTHER_ERROR (16#00FE)	16#0204	16#0004	The parameter P-0-0750 of the slave axis is configured incorrectly. The value has to be: P-0-0750=0
OTHER_ERROR (16#00FE)	16#0204	16#0005	The parameter P-0-0090 of the slave axis is configured incorrectly. Bit 2 of P-0-0090 has to be FALSE

ErrorID	Additional1	Additional2	Description
SYSTEM_ERROR (16#7FFF)	16#0205	16#0000	Immediate cut (in standstill) was requested although the material was moving
OTHER_ERROR (16#00FE)	16#0206	16#0000	The parameter A-0-0200 of the slave axis does not correspond to the logical axis number of the master axis.

Tab. 4-12: Error codes of the ML_FlyingShear / MX_FlyingShear function block

4.3.3 MB_FS_RETAIN_DATA

The MB_FS_RETAIN_DATA structure contains the Retain data required for the ML_FlyingShear / MX_FlyingShear function block.

Type declaration

```

TYPE MB_FS_RETAIN_DATA :
STRUCT
  bCutNotCompleted : BOOL;      // The last cut was not completed
  diMasterSyncPosition : DINT;  // Position where the slave axis should be synchronized
  iRevCounter: INT;             // Revolution counter of the master [-2048...2047]
END_STRUCT
END_TYPE

```

4.3.4 ML_FlyingShear - Special Features of MLC

Required hardware

- IndraDrive C or M
- Additional encoder interface card to read in the measuring wheel position if a measuring encoder is to be used (can be omitted if a virtual or real axis is used)
- Additional encoder to read in the measuring wheel position (as in the IndraDrive project planning manual) if a measuring encoder is to be used (can be omitted if a virtual or real axis is used)



It is recommended to use a high-resolution measuring encoder with a resolution of at least 4096 increments/revolution (possibly sinus encoder). Roughly resolved encoder signals cause low cut accuracies as well as running noises in the synchronous run.

Required firmware

- Drive firmware MPX07 or higher
- The following function packages have to be enabled in the drive:
 - Closed-Loop
 - Synchronization

Required Parameterization

The following parameterization has to be accomplished in the drive to ensure proper functionality of the "ML_FlyingShear" function block.

1. Create the slave axis in IndraWorks
The function block requires the command value generation in the drive. Thus, when creating a slave axis in IndraWorks, the checkbox "Interpolation in the drive" may not be deselected.
2. Loading the parameter file "FlyingShearSettings.par"
By loading the parameter file "FlyingShearSettings.par", important configurations as well as the cams are automatically loaded on the Flying-Shear axis. The parameter file is provided via the IndraWorks MLC installation CD 3 in the path "Add ons\ Tech FB_Parameterfile".
The parameter file contains the following information:
 - 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. Parameterization of the default master axis

Enter the logical axis number of the master axis to the parameter A-0-0200 of the FlyingShear slave axis

Write on the parameter A-0-0200 of the slave axis with the logical axis number of the master axis using the IndraWorks parameter editor.

4. Parameterization of the FlyingShear axis

Parameterization of the FlyingShear axis according to the application.

Scaling should be linear and absolute.

5. Parameterization of the optional encoder

IndraWorks ► Add an encoder axis to the project

Scaling should be set to rotary modulo 360°.

Setting the encoder parameters according to encoder data

6. Ensuring the encoder motion direction

In normal operation, the optional encoder (measuring wheel) has to move in the positive direction. The motion direction can be observed via the IndraWorks dialog of the encoder axis. The motion sense has to be inverted if the encoder moves in the negative direction.

7. Parameterize the synchronization acceleration (P-0-0142) and synchronization velocity (P-0-0143) with preferably high values to ensure a dynamic synchronization process

IndraWorks ► Parameter Editor.

8. Parameterize the synchronous run window (S-0-0228) preferably small to obtain a possibly high cut accuracy

IndraWorks ► Parameter Editor.

Required IndraLogic Steps

- The "ML_FlyingShear" function block contains high priority Motion commands and should therefore be cyclically called in a high priority, cyclical PLC task with a cycle time $\leq 4\text{ms}$ (MotionTask)
- Before starting the function block, MC_Power has to be processed without errors

4.3.5 MX_FlyingShear - Special Features of MLD

Required hardware

- IndraDrive C or M with Advanced control section
- Additional encoder interface card to read in the measuring wheel position (can be omitted if the virtual master axis generator or a real axis is used as master axis)
- Additional encoder to read in the measuring wheel position (as in the IndraDrive project planning manual) if a measuring encoder is to be used (can be omitted if a virtual master axis generator or real axis is used as master axis)

	 It is recommended to use a high-resolution measuring encoder with a resolution of at least 4096 increments/revolution (possibly sinus encoder). Roughly resolved encoder signals cause low cut accuracies as well as running noises in the synchronous run.
Required firmware	• Drive firmware MPx07 or higher • The following function packages have to be enabled in the drive: – Closed-Loop – Synchronization
Required Parameterization	The following parameterization has to be accomplished in the drive to ensure a proper functionality of the "MX_FlyingShear" function block.  The parameter files below are located in the "AddOn" directory of the IndraWorks MLD installation CD in the "Parameter files" folder. 1. Load the parameter file FS_Settings_Local_MPH07.par if the MX_FlyingShear function blocks affects the local MLD axis. The steps 2 and 3 can be omitted in this case. 2. Load the parameter file FS_Settings_MLDM_Slave_MPH07.par to the MLD-M slave axis if the MX_FlyingShear function block should affect this axis. This sets the following parameters: • P-0-0072 ⇒ synchronization cam (polynomial 5th order) • P-0-0092 ⇒ 1:1 cam • P-0-0765 ⇒ 0: Modulo factor measuring encoder • P-0-0750 ⇒ 0: Master axis revolutions for each master axis cycle • P-0-0693 ⇒ 0: Filter time constant of the master axis offset • P-0-0060 ⇒ 0: Filter time constant, additive command position • P-0-0086 ⇒ 0000.0000.0000.1000: Configuration word for synchronous operation mode • P-0-0918 ⇒ 104.8576: Distance virtual master axis • P-0-0916 ⇒ P-0-0758: Signal selection reference value converter • P-0-0917 ⇒ 0000.0000.0000.0001: Control word of the reference value generator • P-0-0756 ⇒ 0000.0000.0000.0010: Scaling virtual reference value generator • P-0-0142 ⇒ 10000: Dynamic synchronization acceleration • P-0-0143 ⇒ 40000: Dynamic synchronization velocity • S-0-0048 ⇒ 0: Additive position command value 3. Load the parameter file FS_Settings_MLDM_Master_MPH07.par in the MLD-M master axis if the MX_FlyingShear should affect the first MLD-M slave axis. The parameter file is provided with the following settings: • P-0-1367 ⇒ 0000.0010.0101.0000: PLC configuration word • P-0-1623 ⇒ Configuration master axis command values • P-0-1624 ⇒ Configuration actual master axis values

- P-0-1625 ⇒ Configuration slave axis command values
- P-0-1626 ⇒ Configuration actual slave axis values
- P-0-0918 ⇒ 104.8576: Distance virtual master axis
- P-0-0916 ⇒ P-0-0758: Signal selection reference value converter
- P-0-0917 ⇒ 0000.0000.0000.0001: Control word of the reference value generator
- P-0-0756 ⇒ 0000.0000.0000.0010: Scaling virtual reference value generator



Due to the presetting in the **FS_Settings_MLDM_Master_MPH07.par** parameter file, the MLD-M master axis triggers the **first** MLD-M slave axis via the FlyingShear function block. Adjust the configuration to the optional cyclic data if another MLD-M slave axis is to be controlled.

The following optional cyclic MDT data (command values) are required and have to be configured accordingly on the MLD-M master axis:

- P-0-0692 ⇒ Additive master axis position, process controller
- P-0-0691 ⇒ Additive position command value, process controller

The following optional cyclic AT data (actual values from the MLD slave axis to the master axis) is required and have to be configured accordingly on the MLD-M master axis:

- P-0-0052 ⇒ Actual position value, measuring encoder
- P-0-0053 ⇒ Master axis position
- P-0-0761 ⇒ Master axis position for slave axis

These setting can be made in IndraWorks by clicking with the right-mouse button on the SERCOS III node and by selecting **Cross Communication Drive: Process data, command values** for MDT and **Cross Communication Drive: Process data, actual values** for the AT configuration.

- Configure the mechanical axis settings of the flying shear (linear axis, no modulo, travel range, range limits etc.) of the given application accordingly.
- Configure the additional second (measuring) encoder.
 - Switch IndraWorks to online mode and open the axis dialog **Measuring Systems ▶ Optional Encoder**
- Ensure that the measuring encoder moves towards the positive direction since the synchronization mechanism can only be used for the positive direction.
 - Switch IndraWorks to online mode and open the axis dialog **Measuring encoder**

Monitor the position of the measuring encoder while the encoder moves in material direction. The encoder position has to be changed to positive direction. The encoder position is only updated in the drive states bb/Ab, AH and AF.



If software limit switches are activated in the drive, "Error" has to be configured as response. The "Warning message" setting can cause a retriggering of the synchronization after an axis standstill.

Relevant parameters for the MX_FlyingShear

The following parameters are relevant to operate the MX_FlyingShear function blocks.

Parameter IDN	Parameter name	Description
P-0-0750	Master axis revolutions per master axis cycle	It specifies the modulo range of the master axis. Always use P-0-0750 = 0
P-0-0765	Modulo factor measuring encoder	It specifies the modulo range of the measuring encoder. Always use P-0-0765 = 0
P-0-0329	Smoothing of actual position value 3 of measuring encoder	This is a time filter constant of the measuring encoder filter. The filter reduce interferences if the flying shear axis runs synchronously to the material
P-0-0142	Synchronization acceleration	This parameter determines the acceleration to be used if the slave axis lacks of time to reach the starting position "ReturnPos" for the next step. The slave axis interrupts the returning motion and accelerates with this value for the next step. The acceleration should be as high as possible
P-0-0143	Synchronization velocity	This parameter determines the velocity to be used if the slave axis lacks of time to reach the starting position "ReturnPos" for the next step. The slave axis interrupts the returning motion and accelerates with this value for the next step. The velocity should be as high as possible
S-0-0228	Synchronization run window position	The MX_FlyingShear sets the input InSync = TRUE as soon as the synchronization distance (SyncDist) was traveled and the actual axis position is in the synchronous run window. The synchronous run window should be small to ensure a high cut accuracy
P-0-0694	Gear ratio fine adjustment, process controller	This gear fine adjustment causes that the flying shear axis moves faster than the material if P-0-0694 > 0%. If P-0-0694 < 0%, the gear fine adjustment causes that the flying shear axis moves slower than the material. A value of <= 0% causes an additional cut inaccuracy
S-0-0193	Positioning jerk	This parameter limits the acceleration change (jerk) in the return motion. S-0-0193=0 deactivates the jerk limitation

Tab. 4-13: Relevant parameters for the MX_FlyingShear

4.4 MagicBelt

4.4.1 Introduction and overview

In the "MagicBelt" application, products are grouped in a nearly continuous stream and typically delivered to secondary packaging (e.g. cardboard box). "MagicBelt" applications are also known as double drag chains, DualBelt or MagicChain. In this application, each chain is controlled with its own individual instance of the MB_MagicBeltType01 function block described below. The function blocks communicate with each other using the applied data structures.

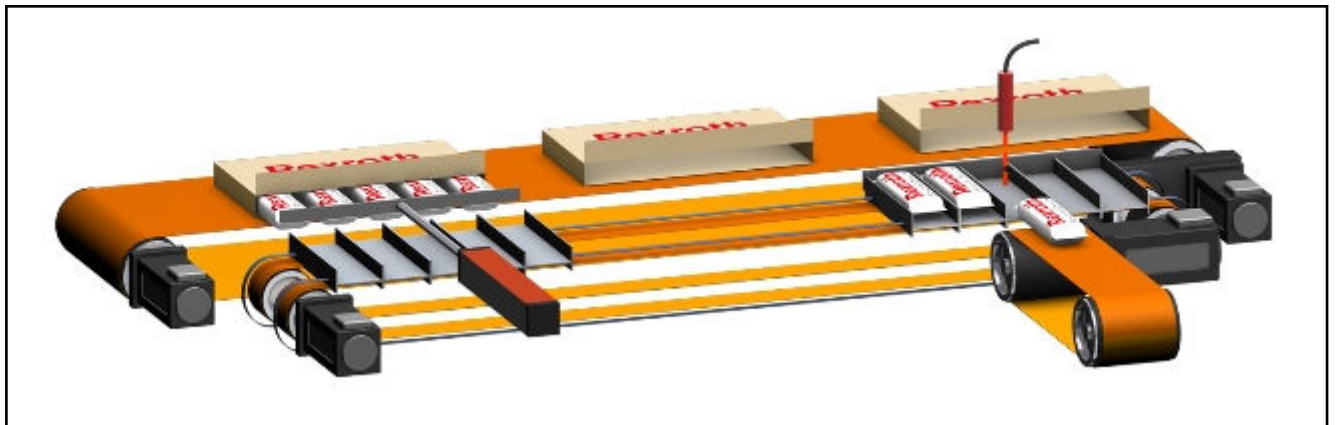


Fig. 4-17: MagicBelt application

Definition of terms

The terms are defined as follows:

- **Chain:** The chain is driven by servo motors. A chain can contain up to 4 trains
- **Train:** The train is used to group products and contains several pockets.
- **Pocket:** A pocket can contain one or more products and is a part of a train.
- **InfeedMaster:** Drive axis (or encoder) of the synchronizing wheel for the synchronizing wheel infeed
- **InfeedAxis:** Virtual axis used internally to execute "Infeed" motions

Definitions and boundary conditions

Several trains are often used per chain, since they enable shorter travel motions and lower velocities. Thus, a "smaller" motor can be used which reduces mechanical wear of chain and motor.

If several trains are mounted on a chain, the following is assumed:

- Distance from one train begin to the next train begin is the same (symmetrical train distribution).
- All trains of one chain have the same pocket design (e.g. number and size of pockets).
- All chains are equipped with the same number of trains.

Thus, the modulo value of a chain drive is defined by the distance between the leading edge of one train and the leading edge of the next train (translatory unit). If there is only one train on the chain, the modulo value is equal to the chain length.

The modulo value is determined using the following equation:

$$\text{ModuloValue} = \frac{\text{Chain Length}}{\text{Trains Per Chain}}$$

Fig. 4-18:

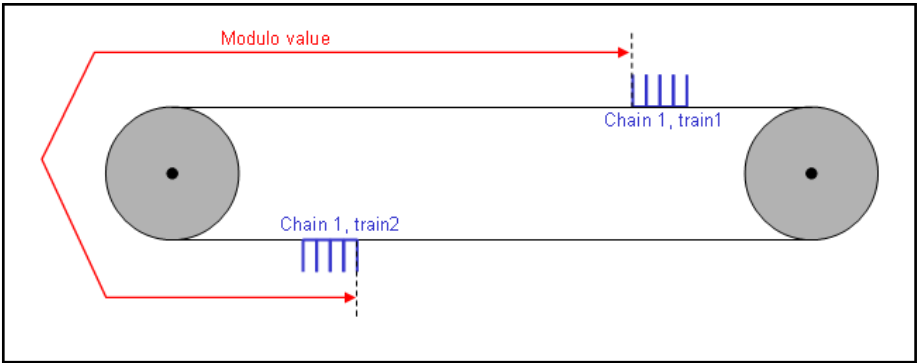


Fig. 4-19: MagicBelt definitions: Modulo value

4.4.2 MB_MagicBeltType01

Brief description The function block contains functionalities to control a chain in a MagicBelt application.

Depending on the operation mode selected, the respective chain is controlled accordingly:

- **Jog mode, MagicBelt, page 203.:** The trains can be jogged individually or synchronously, optionally continuous or incremental.
- **Tune mode, MagicBelt, page 204.:** To determine dynamic limit values (such as the maximum acceleration), the trains are synchronized and continuously traveled by a set distance.
- **Automatic mode, MagicBelt, page 204.:** The trains travel alternately to the loading and unloading position(s).

 The function block operation requires special boundary conditions (e.g. modulo parameterization, cyclic function block call in the "Motion" task...).

Thus, please note [Commissioning notes, MagicBelt, page 218.](#)

Assignment target system/library

Target system	Library
MLC 14VRS	ML_TechMotion
Function package TEC	

Tab. 4-14: Reference table of the MB_MagicBeltType01

- Features**
- Up to three chains each can be managed (three function block instances for chain A,B,C).
 - Up to four trains with equal distances between them can be arranged on one chain.
 - Up to 40 pockets with **different** pocket widths can be used per train. For several pockets **of the same length**, the maximum number of pockets increases accordingly (also refer to the structure element MB_MBLET_INFEED.InfeedStepData)
 - Indexed and synchronized loading are supported
 - Up to ten different unloading positions can be approached.

Interface description

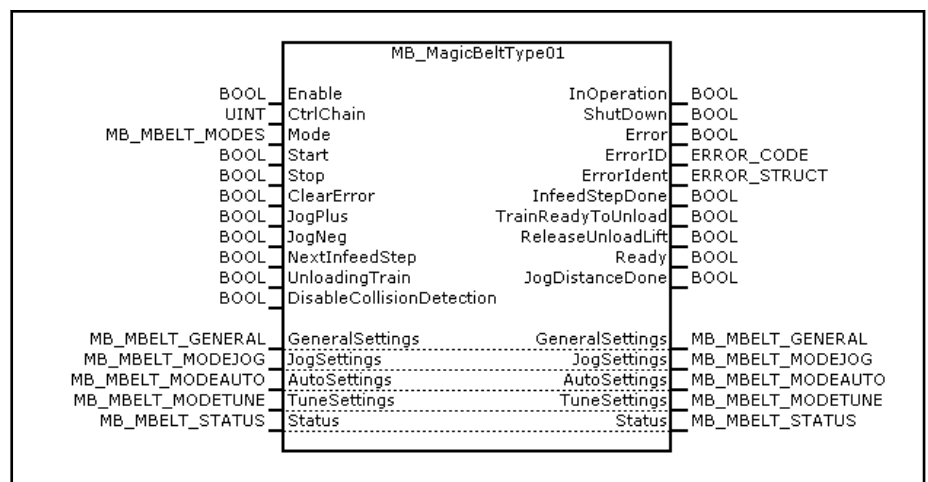


Fig. 4-20: MB_MagicBeltType01 function block

The user program controls the chains using the function block inputs (e.g. Start, Stop...). The configuration data for operation is applied using VAR_IN_OUT data structures (e.g. AutoSettings...).

Additional diagnostic information is available via another "Status" VAR_IN_OUT data structure.

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Processing enabled for function block; state-controlled
	CtrlChain	UINT	Selects the chain number to be controlled: 1: Chain A, 2: Chain B, 3: Chain C
	Mode	MB_MBELT_MODES	Operation mode selection Switching between the operation modes only in "Stopped" state MB_MBELT_MODES, page 189
	Start	BOOL	Positive edge at "Start" starts the axis motion (also refer to "StartType")
	Stop	BOOL	Positive edge at "Stop" stops the axis motion (also refer to "StopType")
	ClearError	BOOL	Positive edge clears a pending error
	JogPlus	BOOL	Jogging the axes in positive direction (Only in active jog mode)
	JogNeg	BOOL	Jogging the axes in negative direction (Only in active jog mode)
	NextInfeedStep	BOOL	For Infeed type "FEED_INDEXED": After a positive edge and after the "StartDelayTime" elapsed, the train is indexed to the next pocket. For Infeed type "FEED_SYNCHRONIZED": A positive edge in the expectation window ("ProductInExpectationWindow" = TRUE) signals a product in the pocket of the "InfeedMaster" Infeed wheel.

I/O type	Name	Data type	Description
	UnloadingTrain	BOOL	As the train is unloaded, the input has to remain set until unloading is completed. The train remains at the unloading position as long as this signal is set. A negative edge results in the train indexed to the next unloading or loading position.
	DisableCollisionDetection	BOOL	Switches off the collision monitoring in order to move the chains from each other after a collision for example
VAR_OUTPUT	InOperation	BOOL	Error-free operation; output data is valid
	ShutDown	BOOL	Defined shutdown (standstill) after a negative edge at "Enable"
	Error	BOOL	Signals incorrect state
	ErrorID	ERROR_CODE	Error diagnostics
	ErrorIdent	ERROR_STRUCT	Detailed error diagnostics Error codes, page 186
	InfeedStepDone	BOOL	Output only for the function block of chain A (CtrlChain = 1 ChainA). Output is only relevant for Infeed variant "FEED_INDEXED". The output is set if the "InfeedStep" was executed. The output is reset with a positive edge at "NextInfeedStep"
	TrainReadyToUnload	BOOL	Train is at unloading position and ready to be unloaded
	ReleaseUnloadLift	BOOL	Unloader enabled
	Ready	BOOL	Only output for a function block with "CtrlChain" = 1 (ChainA): The output is set if the incoming product stream from MagicBelt can be accepted; see also product jam monitoring
	JogDistanceDone	BOOL	The output is set if the specified distance was covered during incremental jogging. The output is reset if "JogPlus"/"JogNeg" is reset
VAR_IN_OUT	GeneralSettings	MB_MBELT_GENERAL	General configuration of the "MagicBelt" application MB_MBELT_GENERAL, page 189
	JogSettings	MB_MBELT_MODEJOG	Jog mode settings MB_MBELT_MODEJOG, page 191
	AutoSettings	MB_MBELT_MODEAUTO	Automatic mode settings MB_MBELT_MODEAUTO, page 193
	TuneSettings	MB_MBELT_MODETUNE	Tune mode settings MB_MBELT_MODETUNE, page 192
	Status	MB_MBELT_STATUS	Status information on the chains MB_MBELT_STATUS, page 201

Tab. 4-15: Interface variables of the MB_MagicBeltType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
CtrlChain	UINT	1	3	1	Pos. Edge at "Enable"
Mode	MB_MBELT_MODES	MODE_AUTO	MODE_JOG	MODE_AUTO	Continuous
Start	BOOL			FALSE	Continuous in the "Auto" and "Tune" mode
Stop	BOOL			FALSE	Continuous in the "Auto" and "Tune" mode
ClearError	BOOL			FALSE	Continuous in the "Error" mode
JogPlus	BOOL			FALSE	Continuous in the "Jog" mode
JogNeg	BOOL			FALSE	Continuous in the "Jog" mode
NextInfeedStep	BOOL			FALSE	Continuous in the "Auto" mode
UnloadingTrain	BOOL			FALSE	Continuous in the "Auto" mode
DisableCollision-Detection	BOOL			FALSE	Continuous

Tab. 4-16: Min./max. and default values of the MB_MagicBeltType01 function block

Data structures used

Data structures used

Structure // substructure	Purpose
MB_MBELT_MODES, page 189	Operation modes of MagicBelt function block
MB_MBELT_GENERAL, page 189	Operation mode-comprehensive settings (e.g. chain length)
MB_MBELT_MODEJOG, page 191	Settings for the jog mode (e.g. jog velocity)
MB_MBELT_MODETUNE, page 192	Setting for the tune mode (e.g. tune velocity)
MB_MBELT_MODEAUTO, page 193	Settings for the auto mode (e.g. tune velocity)
MB_MBELT_STARTTYPE, page 193	Start type (e.g. START_AT_BEGIN)
MB_MBELT_STOPTYPE, page 194	Stop type (e.g. STOP_IMMEDIATE)
MB_MBELT_INFEED, page 194	Settings to load in auto mode
MB_MBELT_FEEDTYPE, page 197	Loading procedure (indexed, synchronized, with velocity)
MB_MBELT_INFEEDSTEPS, page 197	Chain-specific chain data to load in automatic mode
MB_MBELT_OUTFEED, page 199	Settings to unload in auto mode
MB_MBELT_MOVE, page 200	Settings to travel between Infeed ↔ Outfeed in auto mode
MB_MBELT_STATUS, page 201	Chain-comprehensive status information

Functional description With "Enable" = TRUE, the function block starts and the selected operation mode selected at "Mode" is enabled after initialization. The chain is controlled according to the selected operation mode. Switching between operation modes is only allowed in the safe "Stopped" state when the chains are at standstill. The operation modes and their functionalities are described below.

The following inputs are independent of the operation modes and always effective:

- Enable
- Mode
- CtrlChain
- ClearError
- DisableCollisionDetection

The following outputs are independent of the operation modes and always effective:

- InOperation
- Error
- ErrorID
- ErrorIdent
- Shutdown

The following inputs/outputs are independent of the operation modes and always effective:

- GeneralSettings
- Status



All "Chain" axes have to be referenced (using Enable=TRUE) when the function block starts. If this condition is not met, the function block reports an error.

Error codes The following error codes in the "F_RELATED_TABLE" are generated by the function block:

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR	16#0001	16#0000	No power connected to chain drive
RESOURCE_ERROR	16#0003	16#0000	The function block was aborted by another function block
STATE_MACHINE_ERROR	16#0006	16#0000	Invalid state of the internal state machine
RESOURCE_ERROR	16#0011	16#0000	Control not ready for operation
INPUT_INVALID_ERROR	16#1A00	16#0001	The remaining positioning time is ≤ 0 ms Reason: InfeedLoadTime is greater than the time resulting from MaxProductRate. Debugging: Reduce InfeedLoadTime and/or MaxProductRate.
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

ErrorID	Additional1	Additional2	Description
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 is not in the value range
INPUT_INVALID_ERROR	16#1A00	16#000A	ChainAxisRef is not in the valid value range
INPUT_INVALID_ERROR	16#1A00	16#000B	StopRamp too small. StopRamp has to be higher than or equal to all other given accelerations
INPUT_INVALID_ERROR	16#1A00	16#000C	InfeedType is invalid
INPUT_INVALID_ERROR	16#1A00	16#000D	FirstInfeedPosition is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#000E	BeginReducedVel1 is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#000F	EndReducedVel1 is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0010	BeginReducedVel2 is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0011	EndReducedVel2 is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0012	MaxProductRate is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0013	MinDistanceToOutfeed is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0014	MaxInfeedVelocity is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0015	MaxInfeedAcceleration is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0016	MaxInfeedJerk is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0017	StartDelayTime is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0018	InfeedLoadTime is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0019	InfeedMotionLaw is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#001A	ProdExpectationWindowStart is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#001B	ProdExpectationWindowEnd is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#001C	AdditionalSteps is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#001D	InfeedStepData is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#001E	FirstOutfeedPosition is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#001F	MaxOutfeedVelocity is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0020	MaxOutfeedAcceleration is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0021	MaxOutfeedJerk is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0022	OutfeedDistances is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0023	MaxMoveVelocity is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0024	MaxMoveAcceleration is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0025	MaxMoveJerk is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0026	LoadTrainAcceleration is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0027	ReducedVelocity is not in the valid range

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR	16#1A00	16#0028	JogAxisSelection is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0029	JogVelocity is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#002A	JogAcceleration is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#002B	JogDistance is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#002C	TrainDistance is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#002D	SafetyDistance is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#002E	Mode input is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#002F	Expectation window end "ProdExpectationWindowEnd" is too big
INPUT_INVALID_ERROR	16#1A00	16#0030	InfeedMaster AxisRef2 is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0031	CatchUpAcceleration is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0032	Infeed step exceeds the maximum acceleration "MaxInfeedAcceleration"
INPUT_INVALID_ERROR	16#1A00	16#0033	LoadTrainDeceleration is not in the valid range
INPUT_INVALID_ERROR	16#1A00	16#0034	WaitDistBlockedOutfeed is not in the valid range
RESOURCE_ERROR	16#1A01	16#0000	One or multiple chain drives are not referenced
RESOURCE_ERROR	16#1A02	16#0000	Outfeed position is locked
SYSTEM_ERROR	16#1A03	16#0000	Collision detected with train at front
SYSTEM_ERROR	16#1A04	16#0000	Infeed acceleration is higher than "Move" acceleration
RESOURCE_ERROR	16#1A05	16#0000	Falling edge at "TrainUnload", but "UnloadLift" is not "retracted" or "extended"
RESOURCE_ERROR	16#1A06	16#0000	Modulo value of the chain drive or InfeedMaster is not set correctly
RESOURCE_ERROR	16#1A07	16#0000	Consecutive error caused by leading chain
SYSTEM_ERROR	16#1A08	16#xxxx	Unexpected state of another chain xxxx: Chain number
SYSTEM_ERROR	16#1A09	16#xxxx	Invalid internal array index xxxx: Array index
RESOURCE_ERROR	16#1A0A	16#xxxx	UnloadLiftLowPos AND UnloadLiftHighPos are set at the same point in time xxxx = Outfeed index
RESOURCE_ERROR	16#1A0B	16#xxxx	Resulting error caused by another chain xxxx = Chain number with error cause
RESOURCE_ERROR	16#1A0C	16#xxxx	Repositioning is required (START_AT_LAST_STEP), but unloading device is blocked xxxx = Chain number with error cause
SYSTEM_ERROR	16#1A0D	16#0000	Invalid internal step width
SYSTEM_ERROR	16#1A0E	16#0000	Internally calculated collision train is invalid

ErrorID	Additional1	Additional2	Description
SYSTEM_ERROR	16#1A0F	16#0000	The next "Infeed" step was triggered, but the previous "Infeed" step was not yet completed and a REST_IN_REST Infeed motion law was selected
RESOURCE_ERROR	16#1A10	16#0001	Position scaling of the chain drive is invalid
RESOURCE_ERROR	16#1A10	16#0002	Velocity scaling for the chain drive is invalid
RESOURCE_ERROR	16#1A10	16#0003	Acceleration scaling for the chain drive is invalid
RESOURCE_ERROR	16#1A10	16#0004	Position scaling for the InfeedAxis is invalid
RESOURCE_ERROR	16#1A10	16#0005	Velocity scaling for the InfeedAxis is invalid
RESOURCE_ERROR	16#1A10	16#0006	Acceleration scaling for the InfeedAxis is invalid
RESOURCE_ERROR	16#1A10	16#0007	Position scaling for the InfeedMaster is invalid
RESOURCE_ERROR	16#1A10	16#0008	Velocity scaling for the InfeedMaster is invalid
RESOURCE_ERROR	16#1A10	16#0009	Acceleration scaling for the InfeedMaster is invalid
RESOURCE_ERROR	16#1A11	16#0000	The execution of the function block is higher than the Motion cycle time (C-0-0400)

Tab. 4-17: Error codes of the MB_MagicBeltType01 function block

4.4.3 Data types, MagicBelt

MB_MBELT_MODES

The "MB_MBELT_MODES" enumeration contains the operation modes that can be selected for the MagicBelt function block:

Element	Value	Description
MODE_AUTO	16#100	Auto mode
MODE_TUNE	16#300	Tune mode
MODE_JOG	16#D00	Jog mode

Tab. 4-18: MB_MBELT_MODES enumeration type elements

MB_MBELT_GENERAL

The "MB_MBELT_GENERAL" contains chain-comprehensive and operation mode-comprehensive settings:

Structure element	Type	Default value	Effective	Description
ChainLength	REAL	4000.0	Pos. edge "Enable"	Chain length [mm]; Minimum = 1 Max = undefined
NoOfChains	UINT	2	Pos. edge "Enable"	Number of chains Minimum = 1 Maximum = 3
TrainsPerChain	UINT	1	Pos. edge "Enable"	Trains per chain Minimum = 1 Maximum = 4

Structure element	Type	Default value	Effective	Description
DischargeBackwards	BOOL	FALSE	Continuous	Determines the direction of motion in which the trains are discharged. FALSE: Travels across 2/3 of the belt length in positive direction TRUE: Travels across 2/3 of the belt length in negative direction
AllowNegDirection	BOOL	FALSE	Continuous	Allows the train motions in negative direction
TrainDistance	REAL	5.0	Continuous	A train that lags behind catch up with a train that is ahead with this distance (application in Automatic mode) [mm]. The "TrainDistance" has to be specified greater than the "SafetyDistance", since the collision monitoring is triggered otherwise. Minimum = 0.0mm Maximum = Modulo value/4
SafetyDistance	REAL	0.5	Continuous	Safety distance of the collision monitoring [mm] Minimum = -10 mm Maximum = 100 mm
StopRamp	REAL	10000.0	Continuous	Stop ramp deceleration in case of error reactions or stop with the stop variant "STOP_IMMEDIATE". The stop ramp has to be higher than or equal to the maximum value of all other accelerations (MaxMoveAcceleration; MaxInfeedAcceleration; MaxOutfeedAcceleration). Minimum = see description above Maximum = undefined
StartAtLastStepWindow	REAL	100	Continuous	In the start option "StartAtLastStep", the current axis positions are compared with the positions saved after reaching standstill. Only if the position difference for all axes is less than the range specified, the procedure continues according to the start option "StartAtLastStep". If at least one axis is located outside this window, the procedure continues with the start option "Start_At_Begin". Minimum: 0 mm Maximum: ModuloValue
Feedrate	UINT	100	Continuous	All velocities in the "Auto Mode" are reduced accordingly if the "FeedrateOverride" < 100% is specified. Minimum: 10% Maximum: 100%

Structure element	Type	Default value	Effective	Description
InfeedMaster	AXIS_REF	NO_OBJECT	Pos. edge "Enable"	Defines the master axis for synchronous loading with wheel infeed. Only necessary if synchronous loading is selected.
InfeedAxis	AXIS_REF	NO_OBJECT	Pos. edge "Enable"	For the virtual "Infeed axis" used for the Infeed motion
DummyMaster	AXIS_REF	NO_OBJECT	Pos. edge "Enable"	Apply a dummy axis for the internal "FlexProfile" positioning. The axis type and settings are not relevant. The axis is not actively controlled and can be used for other purposes on the application side. It is required for a valid axis reference
ChainAxes	ARRAY [1..3] OF AXIS_REF		Pos. edge "Enable"	Axis assignment for the chains ChainAxes[1] = Chain A ChainAxes[2] = Chain B ChainAxes[3] = Chain C (optional)

Tab. 4-19: Interface description and default values of the MB_MBELT_GENERAL data structure

MB_MBELT_MODEJOG

The data structure "MB_MBELT_MODEJOG" contains the chain-comprehensive data for the Jog mode:

Structure element	Type	Default value	Effective	Description
Incremental	BOOL	FALSE	Pos. edge Jog-Plus/JogNeg	TRUE: With a positive edge at "JogPlus"/"JogNeg", the "JogDistance" is traveled either in positive or in negative direction. FALSE: Travel in positive or negative direction as long as "JogPlus"/"JogNeg" is enabled.
SingleAxisJog	BOOL	FALSE	Pos. edge Jog-Plus/JogNeg	FALSE: All trains are jogged synchronously. TRUE: Only the train selected by "JogAxisSelection" is jogged.
JogAxisSelection	UINT	1	Pos. edge Jog-Plus/JogNeg	Axis selection for single-axis jogging 1: Chain A, 2: Chain B, 3: Chain C
JogVelocity	REAL	100.0	Pos. edge Jog-Plus/JogNeg	Jog acceleration
JogAcceleration	REAL	1000.0	Pos. edge Jog-Plus/JogNeg	Jog accelerations
JogDistance	REAL	2.0	Pos. edge Jog-Plus/JogNeg	Jog distance at incremental jogging

Tab. 4-20: Interface description and default values of the MB_MBELT_MODEJOG data structure

MB_MBELT_MODETUNE

The data structure "MB_MBELT_MODETUNE" contains the chain-comprehensive data for the Tune mode.

Structure element	Type	Default value	Effective	Description
PosDirectionOnly	BOOL	FALSE	At motion start	TRUE: It is traveled relatively in positive direction with "TuneDistance". FALSE: "TuneDistance" travels alternately forward and backward from the initial position.
MotionLaw	MC_STEP_MODE	X_FIT_SINE_TRAPEZE_ALIM	At motion start	Specifies the motion law for a "Tune" positioning. <i>Available motion laws:</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 For detailed information on the motion laws and the MC_STEP_MODE data type, refer to the FlexProfile documentation.
TuneVelocity	REAL	1000.0	At motion start	Maximum positioning velocity
TuneAcceleration	REAL	100.0	At motion start	Maximum positioning acceleration
TuneJerk	REAL	200.0	At motion start	Tune jerk limitation (only effective for motion laws with limited jerk)
TunePosTime	REAL	500	At motion start	<i>Tune positioning time, effective for all motion laws except for:</i> • X_FIT_VEL_TRAPEZE_ALIM • X_FIT_SINE_TRAPEZE_ALIM • X_FIT_ACC_TRAPEZE_JLIM
TuneDistance	REAL	10.0	At motion start	Travel distance per positioning

Tab. 4-21: Interface description and default values of the MB_MBELT_MODETUNE data structure

MB_MBELT_MODEAUTO

The data structure "MB_MBELT_MODEAUTO" contains chain-comprehensive data for the Automatic mode:

Structure element	Type	Default value	Description
StartType	MB_BELT_START-TYPE	START_AT_BEGIN	Defines the start behavior for Automatic mode MB_MBELT_STARTTYPE, page 193
StopType	MB_BELT_STOPTYPE	STOP_CYCLE_COMPLETE	Defines the stop behavior for Automatic mode MB_MBELT_STOPTYPE, page 194 Application at positive "Stop" or negative edge at "Enable"
Infeed	MB_BELT_INFEED		Contains data to load the trains MB_MBELT_INFEED, page 194
Outfeed	MB_BELT_OUTFEED		Contains data to unload the trains MB_MBELT_OUTFEED, page 199
MoveTo	MB_BELT_MOVE		Contains data for the motion between loading and unloading MB_MBELT_MOVE, page 200

Tab. 4-22: Interface description and default values of the MB_MBELT_MODEAUTO data structure

MB_MBELT_STARTTYPE

The "MB_MBELT_STARTTYPE" enumeration contains the possible start variants for the Automatic mode:

Element	Value	Description
START_AT_BEGIN	1	Defined start at reset position
START_WITH_TRAIN_UNLOAD	2	Defined start at reset position with upstream train unloading
START_AT_LAST_STEP	3	Start with the last (interrupted) state (e.g. after error or stop)

Tab. 4-23: Element of the MB_MBELT_STARTTYPE enumeration type

MB_MBELT_STOPTYPE

The "MB_MBELT_STOPTYPE" enumeration contains the possible stop variants for the Automatic mode:

Element	Value	Description
STOP_IMMEDIATE	1	Immediate stop with the ramp MB_MBELT_GENERAL.StopRamp
STOP_CYCLE_COMPLETE	2	Active product cycle (Infeed Step) is still executed. Afterwards, it stops (defined standstill). All trains not synchronized with the InfeedAxis (e.g. for motions between the "Outfeed" positions) are stopped immediately
STOP_IMMEDIATE_SYNC	3	This stop variant is only useful for applications with synchronized infeed. This stop variants acts like STOP_IMMEDIATE. Only the axes in a synchronous relation with the InfeedMaster remain synchronized. When the external InfeedMaster decelerates, these axes decelerate as well. The deceleration of the InfeedMaster has to be ensured outside of the function block

Tab. 4-24: Element of the MB_MBELT_STOPTYPE enumeration type

MB_MBELT_INFEED

The "MB_MBELT_INFEED" data structure contains the comprehensive chain data for loading in the Automatic mode:

Structure element	Type	Default value	Effective	Description
InfeedType	MB_MBELT_FEEDTYPE	FEED_INDEXED	Pos. edge "Enable"	Loading procedure (indexed, synchronized, with velocity) MB_MBELT_FEEDTYPE, page 197
StartDelayTime	TIME	T#50ms	Pos. edge at "NextInfeed-Step"	Only relevant for indexed loading: Delay time (pos. edge "NextInfeedStep"--> starting motion) for indexing Minimum = T#0ms Maximum = T#1h
InfeedLoadTime	TIME	T#10ms	Pos. edge at "NextInfeed-Step"	Only relevant for indexed loading: Time required for train loading. In this period, the train remains at standstill so that the product is not damaged during loading. The input is not effective for the motion laws: X_FIT_ACC_TRAPEZE_JLIM X_FIT_SINE_TRAPEZE_ALIM X_FIT_VEL_TRAPEZE_ALIM Since, in this case, the specified a, v, j maximum values are used

Structure element	Type	Default value	Effective	Description
MaxProductRate	UINT	200	Pos. edge at "NextInfeed-Step"	Only relevant for indexed loading: Maximum number of products per minute that are to be indexed in [1/min] (product stream). The input is not effective for the motion laws: X_FIT_ACC_TRAPEZE_JLIM X_FIT_SINE_TRAPEZE ALIM X_FIT_VEL_TRAPEZE ALIM Since, in this case, the specified a, v, j maximum values are used
MotionLaw	MC_STEP_MODE	VELOCITY_IN_VELOCITY_MOD_SINE	Pos. edge at "NextInfeed-Step"	Only relevant for indexed loading: Specifies the motion law for an Infeed step (for a detailed description of the motion laws refer to the FlexProfile documentation). <i>The following motion laws are recommended for MagicBelt applications:</i> • VELOCITY_IN_VELOCITY_MOD_SINE: Modified sinusoid/low jerk, but high acceleration peak • X_VELOCITY_IN_VELOCITY_TRAPEZE ALIM: Modified acceleration trapezoid/strongly affected by jerk but with lowest acceleration peaks • X_FIT_ACC_TRAPEZE_JLIM: Jerk-limited motion. The "MaxProductRate" input is not effective for this motion law, since the a, v, j maximum values are used. Attention: For "REST_IN_REST_XXX" motion laws, the previous Infeed step has to be completed before a new Infeed step is started. The error Add1 = 16#1A0F is output if this condition is not met
MinDistance-ToOutfeed	REAL	400.0	Continuous	Minimum distance between a train in the "Infeed" state and another train in the "Outfeed" state. If the distance does not meet the lower limit, the "Ready" output is reset to prevent a product jam. For details, refer to the product jam monitoring
MaxInfeedVelocity	REAL	300000.0	Continuous	Maximum velocity of the Infeed step: <i>Only effective for motion laws:</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

Structure element	Type	Default value	Effective	Description
MaxInfeedAcceleration	REAL	5000.0	Continuous	Maximum acceleration for Infeed step: <i>Only effective for motion laws:</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	Continuous	Infeed step maximum jerk: <i>Only effective for the motion law:</i> • X_FIT_ACC_TRAPEZE_JLIM
ProductBuffer	UINT	3	Pos. edge "Enable"	Only for synchronized loading: Number of products between product detection and product ejection Minimum: 0 Maximum: 64
ProdExpectationWindowStart	REAL	30.0	Continuous	Only for synchronized loading: Start of the product expectation window. The "NextInfeedStep" is evaluated in the expectation window. Minimum = 0° Maximum = ProdExpectationWindowEnd
ProdExpectationWindowEnd	REAL	180.0	Continuous	Only for synchronized loading: End of the product expectation window. The "NextInfeedStep" is evaluated in the expectation window. Minimum = ProdExpectationWindowStart Maximum = 360°
FirstInfeedPosition	REAL	500.0	Continuous	First (absolute) loading position Minimum = 0 Maximum = ModuloValue

Structure element	Type	Default value	Effective	Description
AdditionalSteps	UINT	0	Continuous	With "AdditionalSteps", a train that is already completely loaded can be moved one or more steps ahead of the next train before it travels to the "Outfeed" position (e.g. until hold-down is reached).
InfeedStepData	ARRAY [1..3, 1..40] OF MB_MBELT_INFEEDSTEPS		Continuous	List with the size of the pockets and number of products per pocket for all chains. "InfeedStepData"[1,x] = chain A "InfeedStepData"[2,x] = chain B "InfeedStepData"[3,x] = chain C (optional) For synchronized loading, it is always assumed that there is one product per pocket MB_MBELT_INFEEDSTEPS, page 197

Tab. 4-25: Interface description and default values of the MB_MBELT_INFEED data structure

MB_MBELT_FEEDTYPE

The "MB_MBELT_FEEDTYPE" enumeration contains the possible train loading and unloading procedures:

Element	Value	Description
FEED_INDEXED	1	Indexed loading
FEED_SYNCHRONIZED	2	Synchronized loading

Tab. 4-26: Elements of the MB_MBELT_FEEDTYPE enumeration type

MB_MBELT_INFEEDSTEPS

The "MB_MBELT_INFEEDSTEPS" data structure contains the specific chain data for loading in Automatic mode:

Structure element	Type	Default value	Effective	Description
PocketWidth	REAL	50.0	Continuous	Distance from the leading edge of the current pocket to the leading edge of the following pocket. If several pockets are of the same size, the number of pockets of the same size can be specified using the "PocketQuantity" element
PocketQuantity	UINT	1	Continuous	Number of pockets of the same size -1 = Identifier for group formation (see example) 0 = Identifier for train end (see example)
ProductsPerPocket	UINT	0	Continuous	Number of products per pocket (only relevant for indexed loading)

Tab. 4-27: Interface description and default values of the MB_MBELT_INFEEDSTEPS data structure

Example:

Example 1: Pockets of the same size (also refer to the figure below):

The train consists of ten pockets of the same size with a pocket width of 50 mm with one product each

```
ChainA_InfeedData[1].PocketWidth = 50 mm
ChainA_InfeedData[1].PocketQuantity = 10
ChainA_InfeedData[1].ProductsPerPocket = 1
ChainA_InfeedData[2].PocketWidth = don't mind
ChainA_InfeedData[2].PocketQuantity = 0 <- Train end ID
ChainA_InfeedData[2].ProductsPerPocket = don't mind
```

Example:

Example 2: Pockets of different sizes (also refer to the figure below):

The train consists of three different pockets (40 mm, 80 mm, 120 mm) with (1,2,3) products per pocket

```
ChainA_InfeedData[1].PocketWidth = 40 mm
ChainA_InfeedData[1].PocketQuantity = 1
ChainA_InfeedData[1].ProductsPerPocket = 1
ChainA_InfeedData[2].PocketWidth = 80 mm
ChainA_InfeedData[2].PocketQuantity = 1
ChainA_InfeedData[2].ProductsPerPocket = 2
ChainA_InfeedData[3].PocketWidth = 120 mm
ChainA_InfeedData[3].PocketQuantity = 1
ChainA_InfeedData[3].ProductsPerPocket = 3
ChainA_InfeedData[4].PocketWidth = don't mind
ChainA_InfeedData[4].PocketQuantity = 0 <- Train end ID
ChainA_InfeedData[4].ProductsPerPocket = don't mind
```

Example:

Example 3: Pocket group (also refer to the figure below):

The train consists of a pocket group with 5 * 3 different pockets (40 mm, 80 mm, 120 mm) with (1,2,3) products per pocket

```
ChainA_InfeedData[1].PocketWidth = 40 mm
ChainA_InfeedData[1].PocketQuantity = -1 < group ID
ChainA_InfeedData[1].ProductsPerPocket = 1
ChainA_InfeedData[2].PocketWidth = 80 mm
ChainA_InfeedData[2].PocketQuantity = -1 < group ID
ChainA_InfeedData[2].ProductsPerPocket = 2
ChainA_InfeedData[3].PocketWidth = 120 mm
ChainA_InfeedData[3].PocketQuantity = 5 <-group end ID (value > 0) with the
number of groups
ChainA_InfeedData[3].ProductsPerPocket = 3
ChainA_InfeedData[4].PocketWidth = don't mind
ChainA_InfeedData[4].PocketQuantity = 0 <- train end ID
ChainA_InfeedData[4].ProductsPerPocket = don't mind
```

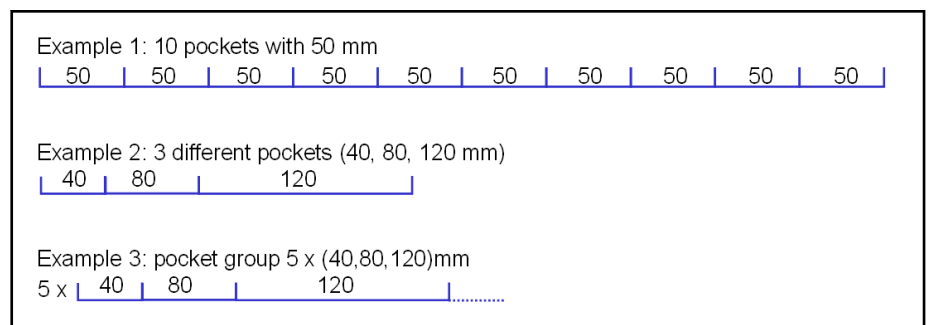


Fig. 4-21: Train configurations of the examples shown above

MB_MBELT_OUTFEED

The "MB_MBELT_OUTFEED" data structure contains the comprehensive chain data for unloading in the Automatic mode:

Structure element	Type	Default value	Effective	Description
MotionLaw	MC_STEP_MODE	X_FIT_SINE_TRAPEZE_ALIM	Pos. edge at "UnloadingTrain"	Specifies the motion law for the motion to the next "Outfeed" position (for a detailed description of the motion laws, refer to the FlexProfile documentation) Acceleration-limited motion (X_FIT_VEL_TRAPEZE_ALIM) Jerk-limited motion (X_FIT_ACC_TRAPEZE_JLIM) Acceleration-limited sine (X_FIT_SINE_TRAPEZE_ALIM)
MaxOutfeedVelocity	REAL	60000.0	Continuous	Maximum velocity to travel to the next unloading position
MaxOutfeedAcceleration	REAL	5000.0	Continuous	Maximum acceleration to travel to the next unloading position
MaxOutfeedJerk	REAL	0.0	Continuous	Maximum jerk to travel to the next unloading position Only effective for the motion law = X_FIT_ACC_TRAPEZE_JLIM
FirstOutfeedPosition	REAL	2000.0	Continuous	First (absolute) unloading position
WaitDistBlockedOutfeed	REAL	0.0	Continuous	When moving to an occupied "Outfeed" position, the train stops at the entered distance to the train in front. No waiting position is approached with WaitDistBlockedOutfeed = 0 (reaction as up to version 12)

Structure element	Type	Default value	Effective	Description
EnableUnload-Handshake	BOOL	FALSE	Continuous	FALSE: Unloading handshake disabled. For applications without collision risk between the unloading device and the train (e.g. unloading robots) TRUE: Unloading handshake enabled. For applications with collision risk between the unloading device and the train. "UnloadLiftLowPos" and "UnloadLiftHighPos" have to be set accordingly (e.g. unloaders)
OutfeedDistances	ARRAY [1..3, 2..10] OF REAL	0.0	Continuous	This list is only necessary if more than one unloading position is to be approached. In this case, this list can be used to specify the distances from one unloading position to the following unloading position. Each negative edge at the "UnloadingTrain" input causes an indexing to the next unloading position. The end of the list is reached if the value 0 is entered. "OutfeedDistances"[1,x] = chain A "OutfeedDistances"[2,x] = chain B "OutfeedDistances"[3,x] = chain C (optional)
UnloadLiftLowPos	ARRAY [1..10] OF BOOL	TRUE	Continuous	Unloader position limit switch/TRUE = unloader retracted
UnloadLiftHighPos	ARRAY [1..10] OF BOOL	TRUE	Continuous	Unloader position limit switch/TRUE = unloader extended

Tab. 4-28: Interface description and default values of the MB_MBELT_OUT-FEED data structure

MB_MBELT_MOVE

The "MB_MBELT_MOVE" data structure contains the comprehensive chain data for traveling between the loading and unloading positions:

Structure element	Type	Default value	Effective	Description
MaxMoveVelocity	REAL	300000.0	At the "Move" motion start	Maximum velocity to travel between loading and unloading stations and for synchronization
MaxMoveAcceleration	REAL	10000.0	At the "Move" motion start	Maximum acceleration/deceleration to travel between loading and unloading stations
MaxMoveJerk	REAL	0.0	At the "Move" motion start	Maximum jerk to travel between loading and unloading stations
LoadTrainAcceleration	UINT	100	At the "Move" motion start	During the movement from the loading station to the first unloading station, it is possible to reduce the acceleration using the "LoadTrainAcceleration" input. The effective acceleration is calculated as follows: $\text{MaxMoveAcceleration} * \text{LoadTrainAcceleration} [\%] / 100 [\%]$

Structure element	Type	Default value	Effective	Description
LoadTrainDeceleration	UINT	0	At the "Move" motion start	During the movement from the loading station to the first unloading station, it is possible to reduce the deceleration using the "LoadTrainDeceleration" input. The effective deceleration is calculated as follows: $\text{MaxMoveAcceleration} * \text{LoadTrainDeceleration} [\%] / 100[\%]$ In the special case "LoadTrainDeceleration = 0", the "LoadTrainAcceleration" input also has an impact on the deceleration
BeginReducedVel1	REAL	0.0	At the "Move" motion start	From this position onwards (the leading edge of the train) the train travels at reduced velocity. This can reduce the centrifugal forces generated by changing directions in curves
EndReducedVel1	REAL	0.0	At the "Move" motion start	The velocity limitation is cancelled from this position onwards (the leading edge of the train)
BeginReducedVel2	REAL	0.0	At the "Move" motion start	From this position onwards (the leading edge of the train) the train travels at reduced velocity. This can reduce the centrifugal forces generated by changing directions in curves
EndReducedVel2	REAL	0.0	At the "Move" motion start	The velocity limitation is cancelled from this position onwards (the leading edge of the train)
ReducedVelocity	REAL	100000.0	At the "Move" motion start	Velocity limitation to reduce the centrifugal forces in curves
AddaptiveSyncRamp	BOOL	TRUE	At the "Move" motion start	Enables an adaptive synchronization ramp
CatchUpAcceleration	REAL	100.0	At the "Move" motion start	Maximum acceleration of the synchronization motion to synchronize to the Infeed. Specification in % of the MaxMoveAcceleration

Tab. 4-29: Interface description and default values of the MB_MBELT_MOVE data structure

MB_MBELT_STATUS

The " MB_MBELT_STATUS" data structure contains status information on the train:

Structure element	Type	Description
ChainStates	ARRAY [1..3] OF MB_MBELT_MODES	Current actual state of the chains MB_MBELT_MODES, page 189
PocketsToBeLoaded	UINT	A train in the "Auto Infeed" state outputs the number of pockets to be loaded in the train via this structure element. Thus, it can be calculated at a higher level for example how long the loading is likely to last
ActInfeedStep	UINT	A train in the "Auto Infeed" state outputs information on the "Infeed" step currently processed via this structure element.

Structure element	Type	Description
ProductInExpectationWindow	BOOL	This output is used in the "FEED_SYNCHRONIZED" Infeed mode if the "InfeedMaster" axis is located in the expectation window. The expectation window is defined by the structure elements "MB_BELT_INFEED", "ProdExpectationWindow-Start" and "MB_MBELT_INFEED" and "ProdExpectation-WindowEnd"
AllChainsHomed	BOOL	All chains are homed
Error	BOOL	Chain-comprehensive error flag
ErrorSourceChain	UINT	Error-causing chain number
MeasuredInfeedStepRate	REAL	Measured InfeedStepRate [min-1] (determined from the temporal gap of the "NextInfeedStep" edges)
MaxProdExpectationWindowEnd	REAL	Maximum possible end position of the expectation window "ProdExpectationWindowEnd" calculated from the current velocity of the InfeedMaster axis [°]
ChainInAutoInfeed	UINT	Chain number located in the "AUTO_INFEED" state
SupressedInfeedSteps	UINT	Number of the suppressed InfeedSteps. Incremented in the loading variant "FEED_INDEXED" if a "NextInfeedStep" edge is applied and if the Infeed train cannot execute the Infeed Step (for diagnostics purposes)
StartAtLastStepPossible	ARRAY [1..3] OF BOOL	Is TRUE if the start option "START_AT_LAST_STEP" is feasible for the respective chain. Index 1 applies to chain A Index 2 applies to chain B Index 3 applies to chain C In the "STATE_MBELT_AUTO_STOPPED" status, the bits for the chains are formed according to the following criteria. The bit for the respective chain is set if: The absolute position difference between the current train position and the position after stopping (Status.RestoreData) is less than GeneralSettings.StartAtLastStepWindow AND The status at stop (Status.RestoreData) was an automatic status AND <i>The status at stop was not:</i> • STATE_MBELT_AUTO_STOPPING or • STATE_MBELT_AUTO_STOPPED or • STATE_MBELT_AUTO_STARTING AND (EnableUnloadHandshake = FALSE OR The absolute position difference between the current train position and the position after stopping is less than 2mm OR all unloading devices UnloadLiftHighPos are retracted)
MaxNoOfInfeedSteps	UINT	Maximum number of Infeed steps for the current train at the Infeed position

Structure element	Type	Description
FirstMasterChain	UINT	Number of chains that reach the FirstInfeedPosition first (for internal purposes only)
OutfeedPosOccupied	ARRAY [1...10] OF BOOL	Set for every "Outfeed" position if it is occupied by a train or the unloader
OccupiedChain	ARRAY [1...10] OF UINT	Chain number which occupies the "Outfeed" position
ChainCmdStates	ARRAY [1...3] OF MB_MBELT_STATES	Calculated target state (for internal purposes only)
RestoreData	ARRAY [1...3] OF MB_MBELT_RESTORE	Information saved at the stop point in time to restore the most recent state at the "StartType" = "START_AT_LAST_STEP" (only for internal purposes)

Tab. 4-30: Interface description and default values of the MB_MBELT_STATUS data structure

4.4.4 Operation modes, MagicBelt

Operation modes, MagicBelt, introduction and overview

The function block [MB_MagicBeltType01](#), page 182, includes functionalities to control a chain of a MagicBelt application.

Depending on the operation mode selected, the respective chain is controlled accordingly:

- [Jog mode, MagicBelt](#), page 203.: The trains can be jogged individually or synchronously, optionally continuous or incremental.
- [Tune mode, MagicBelt](#), page 204.: To determine dynamic limit values (such as the maximum acceleration), the trains are synchronized and continuously traveled by a set distance.
- [Automatic mode, MagicBelt](#), page 204.: The trains travel alternately to the loading and unloading position(s).



The function block operation requires special boundary conditions (e.g. modulo parameterization, cyclic function block call in the "Motion" task...).

Thus, please note [Commissioning notes, MagicBelt](#), page 218.

Jog mode, MagicBelt

Jog mode In jog mode, the following inputs/outputs are active:

Function block inputs:

- JogPlus
- JogNeg

Function block outputs:

- JogDistanceDone

Function block inputs/outputs:

- JogSettings

When setting up the machine, the trains can be jogged continuously or incrementally with the jog mode. After enabling the jog mode, the trains are at standstill (Stopped state). To prevent collisions, jogging the axes is locked with the optional unloader.

Thus, jogging is only possible if:

- The function block input "UnloadingTrain" is reset AND
- The optional unloaders are retracted, i.e. the unloader limit switch signals are: "MB_MBELT_OUTFEED.UnloadLiftLowPos[1..10]" = TRUE

If the above named condition is met, the trains can be jogged in positive or negative direction using the "JogPlus" or "JogNeg" inputs.

The following options are possible:

- The "MB_MBELT_MODEJOG.Incremental" structure element can switch between incremental and continuous jogging (default setting = continuous jogging).
- The "MB_MBELT_MODEJOG.SingleAxisJog" structure element can enable the single-axis jogging. The train to be jogged is selected using the "MB_MBELT_MODEJOG.JogAxisSelection" structure element (default setting = synchronized jogging).

Tune mode, MagicBelt

"Tune" mode In "Tune" mode, the following inputs/outputs are enabled:

Function block inputs:

- Start
- Stop

Function block inputs/outputs:

- TuneSettings

When setting up the machine, the optimum dynamic settings/motion laws can be determined with the "Tune" mode. In permanent start/stop operation, the trains are moved either in one direction or bipolar according to the selected motion law. As in the jog mode, starting the axes is locked with the optional unloader (for a detailed description, refer to the jog mode).

After a positive edge at "Start", the chains are synchronized and travel continuously and cyclically over the distance "MB_MBELT_MODETUNE.TuneDistance". Depending on the "MB_MBELT_MODETUNE.PosDirectionOnly" setting, the trains travel either continuously in positive direction or alternate forward and backward. During the motion, the dynamic settings (e.g. "MB_MBELT_MODETUNE.TuneAcceleration" can be modified in order to determine the maximum acceleration values for example.

Automatic mode, MagicBelt

Automatic mode, MagicBelt, general information

"Auto" mode In "Auto" mode, the following inputs/outputs are active:

Function block inputs:

- Start
- Stop
- NextInfeedStep
- UnloadingTrain

Function block outputs:

- InfeedStepDone
- TrainReadyToUnload
- ReleaseUnloadLift
- Ready

Function block inputs/outputs:

- AutoSettings
- Status

In "Automatic" mode, the trains are moved alternately between the loading and unloading positions to accept and group the almost continuous product stream at the loading position. If the train is fully loaded, the unloading positions are approached to allow an unloader or an unloading robot to remove the product groups. After unloading, the respective train approaches the loading position again.

Start, automatic mode, MagicBelt

The positive "Start" edge enables the motion of the trains. Depending on the start option ("AutoSettings.StartType"), it is started as follows:

START_AT_BEGIN:

The train closest to the "MB_MBELT_INFEED.FirstInfeedPosition" is the leading train and therefore the master. It is also considered whether the trains may only move in positive direction or bipolar (can be set via "MB_MBELT_GENERAL.AllowNegDirection").

With the train distance "MB_MBELT_GENERAL.TrainDistance" all trains catch up to the leading train. The leading train travels to the "MB_MBELT_INFEED.FirstInfeedPosition" and all other trains follow synchronously. After the "FirstInfeedPosition" is reached the train is loaded train begins.

START_WITH_TRAIN_UNLOAD:

The procedure is similar to START_AT_BEGIN. Only an additional train unloading motion is executed at about 2/3 * "MB_MBELT_GENERAL.ChainLength" before the movement to the "FirstInfeed" position to unload the trains safely. The direction of the unloading motion can be affected by the structure element "MB_MBELT_GENERAL.DischargeBackwards". The unloading motion is executed with the maximum velocity "MB_MBELT_MOVE.MaxMoveVelocity".

START_AT_LAST_STEP:

The most recent state after an error or stop can be restored with this start option. This start option first checks if all axis positions are still located in the adjustable window MB_MBELT_GENERAL.StartAtLastStepWindow (compared to the position saved after stop). If at least one of the axes is located outside this window, the "START_AT_BEGIN" procedure is used. If all axes are located in this window, all axes are repositioned in the next step to the latest active position before stop or error. Subsequently, the most recent state before stop or error is restored. To complete the restoration, the axes are repositioned if necessary.

Thus, this start option can only be executed if the following boundary conditions are met:

- Position difference between the current actual position and the latest position of the chain axes at standstill is in the "MB_MBELT_GENERAL.StartAtLastStepWindow". If this condition is not met, START_AT_BEGIN is executed
- The last states saved before the interruption of all chain axes correspond to the "Auto" mode. If this condition is not met, START_AT_BEGIN is executed
- If an unloading device is available (EnableUnloadHandshake=TRUE) and at least one unloader is extended (UnloadLiftHighPos[n] = TRUE) and a repositioning of more than 2 mm at least one axis is required,

"START_AT_LAST_STEP" cannot be executed, since mechanical damage can occur. In this case an error is output

Repositioning is carried out as follows:

- Chains are synchronized (relative synchronization)
- Only if three chains are configured: Chain C is positioned at the last valid position saved. All other chains follow synchronously. Continues when chain C reached the target position
- Chain B is positioned at the last valid position saved Chain C (if present) remains at standstill. Chain A follows synchronously. Continues when chain B reached the target position
- Chain A is positioned at the last valid position saved. All other chains remain at standstill. Continues when chain A reached the target position
- The latest state and all necessary internal variables are restored to reach the state before the interruption

Loading, automatic mode, MagicBelt

All settings for the loading are specified using the structure element "MB_MBELT_MODEAUTO.Infeed".

In principle, indexed loading of the train is differentiated from synchronized loading. With indexed loading, the train moves along the loading position using the function block input "NextInfeedStep" until all pockets of the train are filled.

With synchronized loading, the train is synchronized with the product feed during the loading procedures (product feed with an Infeed wheel for example). In general, this loading procedure is better for the products so that higher index rates are possible. With this procedure, the pocket width is always assumed to be the same. If a product feed pocket is empty, the train has to synchronously skip a pocket width while loading (compensation for empty section).

The train loading procedure is determined by the structure element "MB_MBELT_MODEAUTO.Infeed.InfeedType".

Indexed loading, automatic mode, MagicBelt

With indexed loading, all elements in the data structure "MB_MBELT_MODEAUTO.Infeed" are effective (exception: ProductBuffer and ProductExpectationWindow are not effective).

With indexed loading (AutoSettings.Infeed.InfeedType = FEED_INDEXED), the products arrive on conveyor belts with more or less the same distance between them with a minimum space. Even if the distances seem to be the same, one or more products can be missing. On the way to the pocket, the product triggers a sensor signal to indicate that the train can move on. This signal is applied at the function block input "NextInfeedStep".

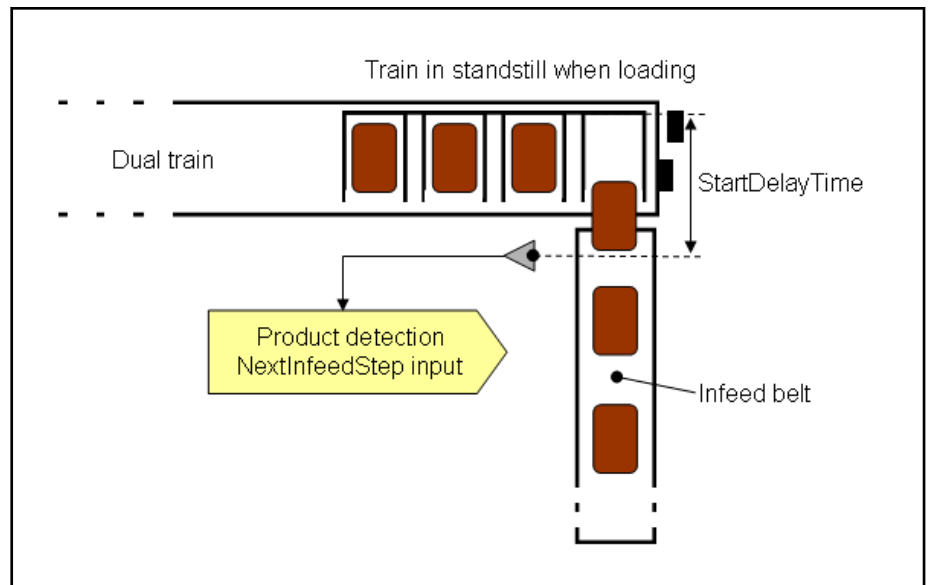


Fig. 4-22: Indexed product feed/loading at standstill from the front

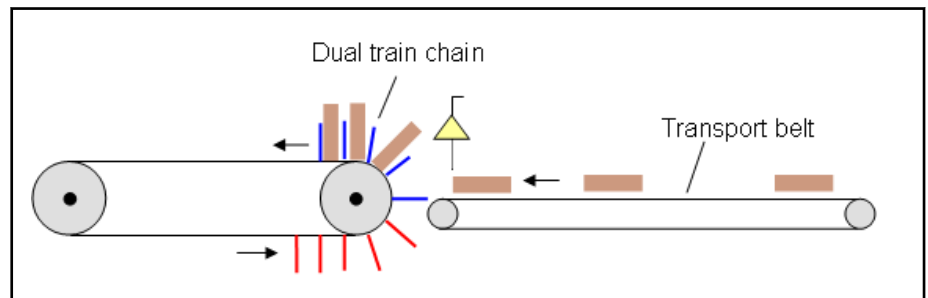


Fig. 4-23: Indexed product feed/loading at standstill from the side

The transport distance can be set from the detection of the sensor signal up to the start of the indexing motion via the adjustable time "AutoSettings.Infeed.StartDelayTime". After the indexing is completed, the function block output "InfeedStepDone" is set for acknowledgement. The pocket width and the products per pocket can vary from pocket to pocket and is specified by the structure element "MB_MBELT_MODEAUTO.Infeed.InfeedStepData".

The feed motion is determined by the following structure elements (see also the following figure):

- "MB_MBELT_MODEAUTO.Infeed.MotionLaw" specifies the motion law
- "MB_MBELT_MODEAUTO.Infeed.StartDelayTime" specifies the time delay until the Infeed motion starts
- "MB_MBELT_MODEAUTO.Infeed.InfeedLoadTime" specifies the time that the train is to remain in standstill (while it is loaded)
- "MB_MBELT_MODEAUTO.Infeed.MaxProductRate" specifies the maximum number of products per minute

The selected motion law "MB_MBELT_MODEAUTO.Infeed.MotionLaw" is very important for the executed Infeed motion.

The following has to be noted:

- "VELOCITY_IN_VELOCITY..." MotionProfiles should be used, since they also allow the train to move forward with velocity (if the previous Infeed step is not yet fully completed, for example). In special cases, all "Rest-in-Rest" FlexProfile motion laws can be used if it is certain that the train to be loaded reaches the standstill after each Infeed step.

The recommended Infeed motion laws are:

- Modified sinoid (low jerk, but high acceleration peaks) ("VELOCITY_IN_VELOCITY_MOD_SINE")
- Modified acceleration trapezoid (strongly affected by jerk but with lowest acceleration peaks) ("X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM")
- A jerk-limited motion is executed with the "X_FIT_ACC_TRAPEZE_JLIM" motion law. The "MB_MBELT_MODEAUTO.Infeed.MaxProductRate" input is not effective for this motion law since the a, v, j maximum values are used.



The distance between the product sensor and the loading station should be less than one product length. In "StartDelayTime", all other positive edges at "NextInfeedStep" are hidden.

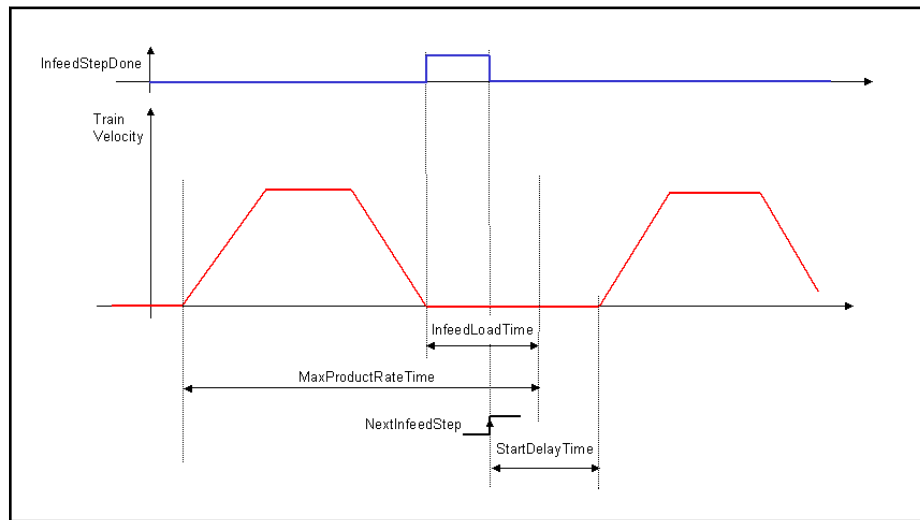


Fig. 4-24: Overview and mode of operation of "InfeedLoadTime", "StartDelayTime" and "MaxProductRate"

MaxMoveVelocity, MaxMoveAcceleration, MaxMoveJerk setting

X_FIT_ACC_TRAPEZE_JLIM

VELOCITY_IN_VELOCITY_MOD_SINE

The loaded train has to leave the Infeed "faster" than the lagging train moves at the Infeed. Thus, the input data MaxMoveVelocity, MaxMoveAcceleration, MaxMoveJerk have to be set as follows:

If the motion law **X_FIT_ACC_TRAPEZE_JLIM** is used for the Infeed motion, the input values **MaxMove...** have to be set as follows:

- MaxMoveVelocity $\geq$ MaxInfeedVelocity
- MaxMoveAcceleration $\geq$ MaxInfeedAcceleration
- MaxMoveJerk $\geq$ MaxInfeedJerk

If the motion law **VELOCITY_IN_VELOCITY_MOD_SINE** is used for the Infeed motion, the input values **MaxMove...** have to be set as follows:

Setting MaxMoveVelocity:

$$\text{MaxMoveVelocity} \left[\frac{\text{mm}}{\text{min}} \right] \geq 1,76 \cdot \frac{S_{\text{max\_infeed}} [\text{mm}]}{\frac{1}{\text{MaxProductRate} [\text{min}^{-1}]} - \frac{\text{InfeedLoadTime} [\text{ms}]}{60000 \left[\frac{\text{ms}}{\text{min}} \right]}}$$

Fig. 4-25: Calculation of MaxMoveVel
S_{max_infeed} = Maximum Infeed step width

Setting MaxMoveAcceleration:

$$\text{MaxMoveAcceleration} \left[\frac{\text{mm}}{\text{s}^2} \right] \geq 5,53 \cdot \frac{S_{\text{max\_infeed}}[\text{mm}]}{\left(\frac{60 \left[\frac{\text{s}}{\text{min}} \right]}{\text{MaxProductRate}[\text{min}^{-1}]} - \frac{\text{InfeedLoadTime}[\text{ms}]}{1000 \left[\frac{\text{ms}}{\text{s}} \right]} \right)^2}$$

Fig. 4-26: Calculation of MaxMoveAcceleration
Smax_infeed = Maximum Infeed step width

Setting MaxMoveJerk:

$$\text{MaxMoveJerk} \left[\frac{\text{mm}}{\text{s}^3} \right] \geq 69,47 \cdot \frac{S_{\text{max\_infeed}}[\text{mm}]}{\left(\frac{60 \left[\frac{\text{s}}{\text{min}} \right]}{\text{MaxProductRate}[\text{min}^{-1}]} - \frac{\text{InfeedLoadTime}[\text{ms}]}{1000 \left[\frac{\text{ms}}{\text{s}} \right]} \right)^3}$$

Fig. 4-27: Calculation of MaxMoveJerk
Smax_infeed = Maximum Infeed step width

X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM

If the motion law **X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM** is used for the Infeed motion, the input values **MaxMove...** have to be set as follows:

Setting MaxMoveVelocity:

$$\text{MaxMoveVelocity} \left[\frac{\text{mm}}{\text{min}} \right] \geq 60 \left[\frac{\text{s}}{\text{min}} \right] \cdot \left(\frac{t_{\text{move}} \cdot a_{\text{max}}}{2} - \sqrt{\left(\frac{t_{\text{move}} \cdot a_{\text{max}}}{2} \right)^2 - a_{\text{max}} \cdot S_{\text{max\_infeed}}[\text{mm}]} \right)$$

$$a_{\text{max}} = \text{MaxMoveAcceleration} \left[\frac{\text{mm}}{\text{s}^2} \right]$$

$$t_{\text{move}}[\text{s}] = \frac{60 \left[\frac{\text{s}}{\text{min}} \right]}{\text{MaxProductRate}[\text{min}^{-1}]} - \frac{\text{InfeedLoadTime}[\text{ms}]}{1000 \left[\frac{\text{ms}}{\text{s}} \right]}$$

Fig. 4-28: Calculation of MaxMoveVel
Smax_infeed = Maximum Infeed step width

Setting MaxMoveAcceleration and MaxMoveJerk:

- MaxMoveAcceleration ≥ MaxInfeedAcceleration
- MaxMoveJerk = 0 (no jerk limitation)

Indexed loading with infeed in moving train, automatic mode, Magi-cBelt

When values for "MB_MBELT_MODEAUTO.Infeed.StartDelayTime" are specified accordingly, the train can also move during loading. Thus, applications in which the train is loaded while moving can be implemented (see following figure).

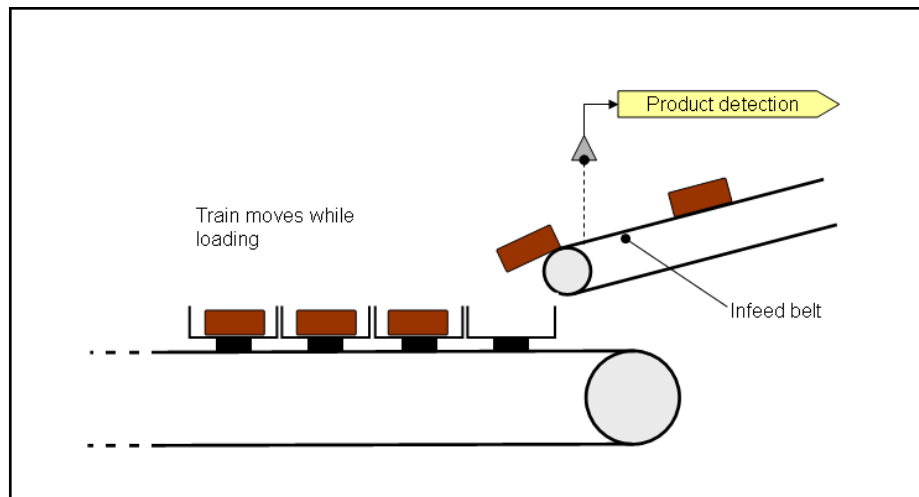


Fig. 4-29: Indexed product feed/loading when the train is moving



For the "FEED_INDEXED" loading variant, the product stream is typically specified for the system using the "NextInfeedStep" input. The train has to follow the specified product cycle. Typically, a train always has to be available at the "Infeed" to accept the products.

However, there are also applications in which the available products are collected and the train "requests" the products (example: carton transport applications). The MagicBelt function block specifies the index rate via the "InfeedStepDone" output. It is not mandatory for these application that a train is at the loading position. These applications can also be implemented with the MagicBelt function block. The following procedures are recommended for these use cases:

If the AND operation "InfeedStepDone" AND "Ready" = TRUE, a train is at the "Infeed" and requests a product. The product (e.g. a cardboard box) should be moved to the train pocket. When loading is completed, the train can move on with a positive edge at "NextInfeedStep". If the loading time is nearly constant, the "InfeedLoadTime" can be set accordingly and the "NextInfeedStep" input can be wired as follows:

NextInfeedStep := Ready AND InfeedStepDone AND ProductsAvailable

With this assignment, the train automatically moves as long as products are available (ProductsAvailable=TRUE). The "ProductsAvailable" signal is calculated in the application program.

Infeed with constant velocity, automatic mode, MagicBelt

Generally, all "VELOCITY_IN_VELOCITY..." Infeed motion laws (such as X_VELOCITY_IN_VELOCITY_TRAPEZE_ALIM) allow continuing motion in the next "Infeed Step" if the train is still moving based on the previous "Infeed Step" (i.e. velocity > 0).

If the specified time intervals of the positive edges at the "NextInfeedStep" input are shorter than the specified maximum product rate "MB_MBELT_INFEED.MaxProductRate", the train no longer reaches the standstill (prerequisite: MB_MBELT_INFEED.InfeedLoadTime = 0). When the "NextInfeedStep" time intervals become progressively shorter, the velocity drops also become progressively smaller. If the positive edges at the "NextInfeedStep" input are

applied at double the product rate "MB_MBELT_INFEED.MaxProductRate" (or higher), the train travels at a constant velocity.

Example:

MB_MBELT_MODEAUTO.Infeed.MaxProductRate = 60 products/min. --> 1 product is equivalent to 1 s --> The train at the "Infeed" moves at a constant velocity if a positive edge is applied every 5 s at the "NextInfeedStep".

The structure element "MB_MBELT_STATUS.MeasuredInfeedStepRate" can check the frequency of the positive edges at the "NextInfeedStep" input for diagnostic purposes. In the example mentioned above, NextInfeedStep = 120 is displayed. The train to be loaded moves at a constant velocity that corresponds to 120 products/minute.

Synchronous loading, automatic mode, MagicBelt

With synchronized loading, the following elements in the data structure MB_MBELT_MODEAUTO.Infeed are effective:

- ProductBuffer
- ProductExpectationWindowStart
- ProductExpectationWindowEnd
- AdditionalSteps
- MinDistanceToOutfeed

With synchronous loading, the train is synchronized with the product feed while loading. In this case, the product feed is carried out using a continuously rotating Infeed wheel ("MB_MBELT_GENERAL.InfeedMaster") which moves one master cycle (= 360°) with each product. A product sensor is wired to the "NextInfeedStep" input and detects whether a product is in the Infeed wheel compartment. The production detection is enabled if the product is located in the expectation window (ProdExpectationWindowStart ... ProdExpectationWindowEnd) with regard to the position of the InfeedMaster. The detected product is entered in a shift register, which indexes with each master cycle. With each master cycle, the train travels by one pocket length if a product is in the Infeed wheel compartment. If no product is in the compartment, the train skips it accordingly. Thus, no empty pocket can result. If there is an empty pocket, the train cannot remain exactly at the "InfeedPosition" since the deceleration path has to be travelled. The train stops after half a compartment (synchronous skipping). Several products can be between the product detection and the train loading. The number of these products is specified for the function block via the "MB_MBELT_MODEAUTO.Infeed.ProductBuffer". With synchronous loading, different pocket sizes can be set. One product per pocket is always assumed. The optional adjustable train distance is added to the last Infeed step on the pocket width to compensate the distance.



In the special case "MB_MBELT_MODEAUTO.INFEED.PRODUCTBUFFER" = 0, the product detection is located directly in front of the train loading area. In this case, the product is included in the same master cycle and the motion law required for the next cycle is prepared. The preparation of the motion law for the next master cycle requires three Sercos cycles. The end of the expectation window "MB_MBELT_INFEED.ProdExpectationWindowEnd" may not become greater than the calculated output value "MB_MBELT_STATUS.MaxProdExpectationWindowEnd". With the specified Sercos cycle time and maximum velocity of the InfeedMaster, this value can also be calculated externally with the following formula:

$$\varphi_{\text{MaxProdExpectationWindowEnd}} [^\circ] = 360^\circ - 3 \cdot t_{\text{Sercos Cycle}} [\text{ms}] \cdot v_{\text{InfeedMaster}} [\text{min}^{-1}] \cdot 0,006$$

Fig. 4-30: MB_MBELT_STATUS.MaxProdExpectationWindowEnd calculation

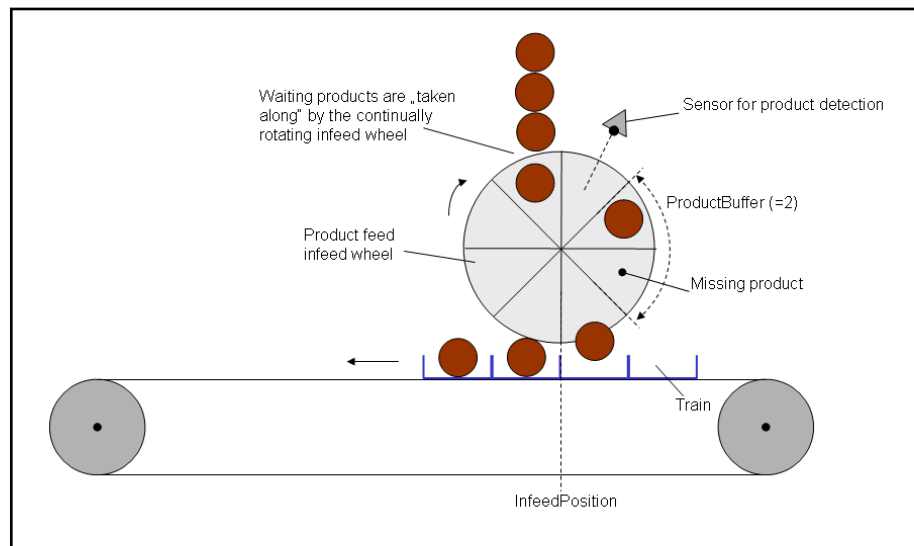


Fig. 4-31: Synchronous product feed

Example of synchronous product feed (temporal sequence):

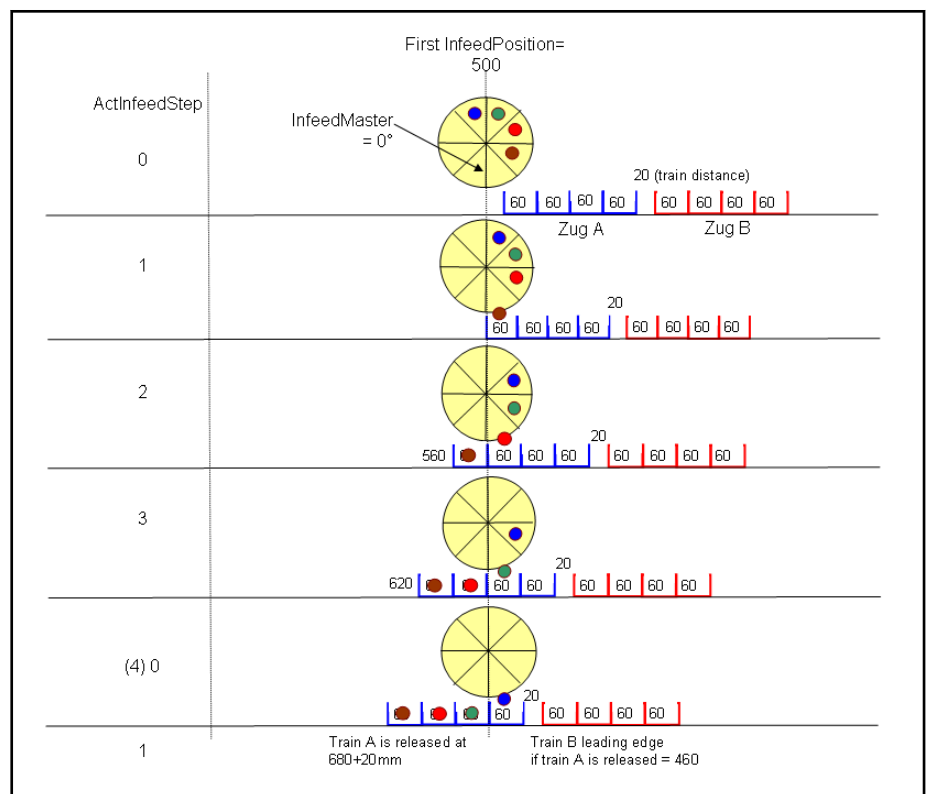


Fig. 4-32: Synchronous product feed (temporal sequence)

Motions between the loading and unloading station, automatic mode, MagicBelt

After the train is loaded or unloaded, it has to approach the next station as quickly as possible. The motions between the loading and unloading stations are specified using the "MB_MBELT_MOVE" data structure. The effective acceleration and deceleration for a loaded train can be reduced using the structure elements "MB_MBELT_MOVE.LoadTrainAcceleration" and "MB_MBELT_MOVE.LoadTrainDeceleration"

Furthermore, the velocity can be reduced in the areas with curves to reduce the resulting centrifugal forces.



The "MB_MBELT_MOVE" velocity/acceleration has to be set greater than or equal to the maximum Infeed velocity/acceleration. Otherwise, there is a risk of collision if a train is released from the loading motion. If the settings are incorrect, the dynamic monitoring reports an error in order to avoid collisions.

Unloading, automatic mode, MagicBelt

All settings for unloading are specified using the structure element "MB_MBELT_MODEAUTO.Outfeed".

To unload, the train travels to the first absolute unloading position ("MB_MBELT_MODEAUTO.Outfeed.FirstOutfeedPosition"). The function block sets the output "TrainReadyToUnload" when the train reaches the "Outfeed position".

A negative edge at the function block input "UnloadingTrain" results in the train moving to the next unloading or loading position. Further unloading positions can be specified using the structure element "MB_MBELT_MODEAUTO.Outfeed.OutfeedDistances". The distances specified correspond to the

distance to the latest unloading position. If a distance = 0 (default) is specified, the "FirstInfeedPosition" is approached again after a negative edge at the function block input "UnloadingTrain".

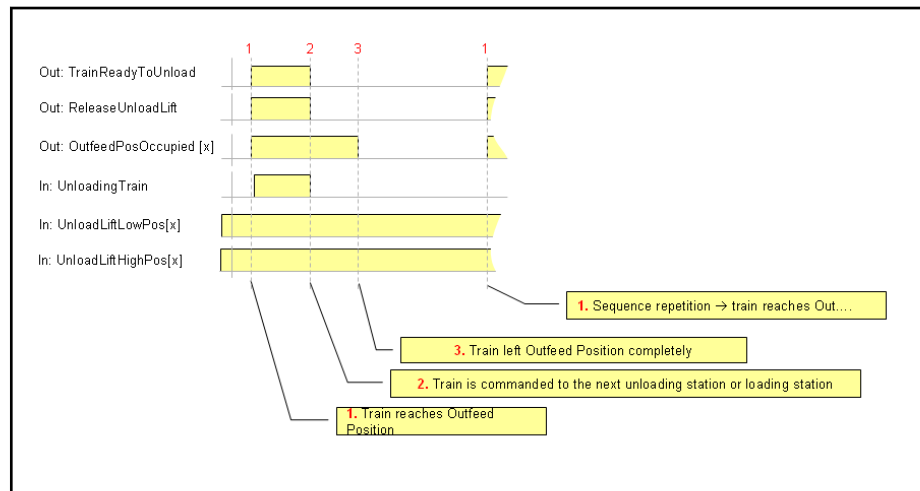


Fig. 4-33: Signal-time diagram when unloading without unload handshake



For 13VRS and higher: When moving to an occupied "Outfeed" position, the train stops at a waiting position. The train starts moving again as soon as the "Outfeed" position is not occupied anymore. The waiting position is specified by the structure element "MB_MBELT_MODEAUTO.Outfeed.WaitDistBlockedOutfeed". The desired distance to the train in front occupying the "Outfeed" position is specified here.

No waiting position is approached when setting MB_MBELT_MODEAUTO.Outfeed.WaitDistBlockedOutfeed = 0 (reaction as up to version 12).

If the train is unloaded by an unloader, the motion of the train and the unloader have to be locked to avoid collisions between train and unloader. The unloading handshake is enabled using the structure element "MB_MBELT_MODEAUTO.Outfeed.EnableUnloadHandshake".

In the simplest case, the train does not start moving to the next unloading or loading position until the unloader has been retracted. The corresponding handshake and the signal-time diagram for this case are shown in the following figure.

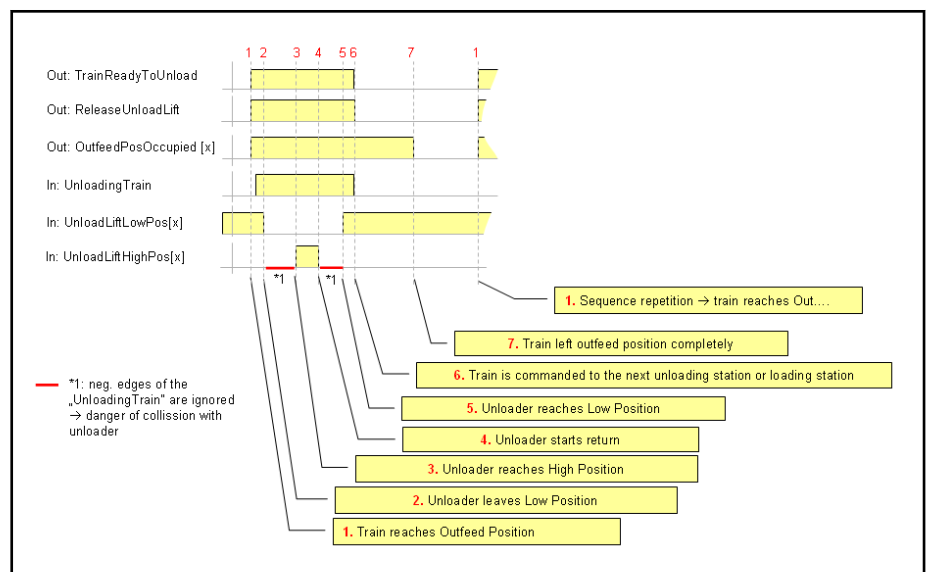


Fig. 4-34: Signal-time diagram to unload with handshake/continuing train traveling when the unloader is retracted

In another use case, the train is indexed already, even though the unloader is fully extended. In this case, the unloader may only be retracted when the train has travelled completely beyond the unloader's penetration area. The unloading position remains occupied until the unloader is completely retracted. The corresponding handshake and the signal-time diagram for this case are shown in the following figure.

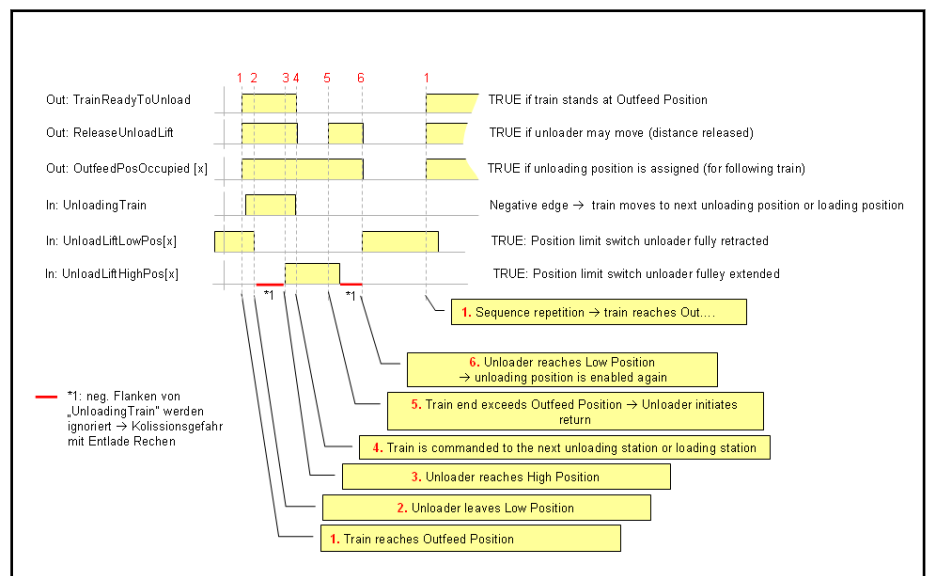


Fig. 4-35: Signal-time diagram when unloading with handshake/continuing train traveling when the unloader is extended

Stop, automatic mode, MagicBelt

With a positive edge at the "Stop" input, the train is stopped according to the specified stop variant ("MB_MBELT_MODEAUTO. StopType"). With the synchronous Infeed variant, it is recommended that the chains are stopped by decelerating the InfeedMaster and subsequently stopping the chains using StopType = STOP_IMMEDIATE.

STOP_IMMEDIATE:

For this stop variant, all trains and the InfeedAxis are decelerated immediately using the "MB_MBELT_GENERAL.StopRamp" ramp (e.g. for fatal disturbances).

STOP_CYCLE_COMPLETE:

For this stop variant, the active product cycle (Infeed step) is still executed. Afterwards, it is stopped (defined standstill). All trains not synchronized with the InfeedAxis (e.g. for motions between the "Outfeed" positions) are stopped immediately

STOP_IMMEDIATE_SYNC:

This stop variant is only useful for applications with synchronized Infeed. This stop variant acts like STOP_IMMEDIATE. Only the axes in a synchronous relation with the InfeedMaster remain synchronized. When the external InfeedMaster decelerates, these axes decelerate as well. The deceleration of the InfeedMaster has to be ensured outside the function block.

Error reaction

In case of an error, all axes are stopped immediately using "MB_MBELT_GENERAL.StopRamp" if they can still follow the command values (procedure as for "STOP_IMMEDIATE"). Furthermore, the "Error" output is set and error codes are output at the corresponding function block outputs. The error is acknowledged by a rising edge at the "ClearError" input. Depending on the start type selected, the Automatic mode can resume at the place of interruption ("MB_MBELT_STARTTYPE" = START_AT_LAST_STEP) after the error is cleared or from the defined reset position ("MB_MBELT_STARTTYPE" = START_AT_BEGIN).

Monitoring, automatic mode, MagicBelt

Product jam monitoring

Unplanned decelerations (e.g. during unloading) can cause situations in which the continuous product stream can no longer be accepted and a product jam occurs. Such situations have to be detected as early as possible to reduce or interrupt the product stream.

Thus, product jam monitoring is enabled in the first train (train A) that sets the "Ready" output when the following condition occurs:

- There is a train at the loading station

AND

*If there is **no** unloading device ("MB_MBELT_INFEED.EnableUnloadHandshake" = FALSE):*

- The distance between a train just loaded (train in "Infeed" state) and other trains at the "Outfeed" positions is greater than "MB_MBELT_INFEED.MinDistanceToOutfeed"

*If there is **an** unloading device ("MB_MBELT_INFEED.EnableUnloadHandshake" = TRUE):*

- The distance between a train just loaded (train in "Infeed" state) and "Outfeed" positions marked as occupied (see "MB_MBELT_STATUS.OutfeedPosOccupied") is greater than "MB_MBELT_INFEED.MinDistanceToOutfeed". If an ("Outfeed" position is occupied, a fixed limit is set at the ("Outfeed" position – train length) position. If the train is in the "Infeed" state and has the distance "MB_MBELT_INFEED.MinDistanceToOutfeed" to this limit, the "Ready" output is set.

If the conditions are not met, the "Ready" output is reset. If the "Ready" output is reset, the upstream station should interrupt the product stream (such as discharging products if "Ready" = FALSE)

If the product stream persists even if "Ready"=FALSE, the monitoring feature checks whether the train in the "Infeed" state would collide with the train at

the occupied "Outfeed" position(s). To do this, the "Stop" ramp is used with the current train velocity and deceleration ("MB_MBELT_GENERAL.StopRamp"). If a collision is detected, the corresponding train is immediately stopped with the "Stop" ramp and an error is output.

The cases with and without unloading device are distinguished to determine the train distances:

- **Without** unloading device ("MB_MBELT_INFEED.EnableUnloadHandshake" = FALSE), the distance between a train that was just loaded (train in "Infeed" state) and other trains at the "Outfeed" positions is monitored
- **With** unloading device, the distance between a train that was just loaded (train in "Infeed" state) and "Outfeed" positions marked as occupied is monitored. If an ("Outfeed" position is occupied, a fixed limit is set at the ("Outfeed" position – train length) position. If the "Infeed" train travels beyond this limit with its deceleration ramp "MB_MBELT_GENERAL.StopRamp", a collision is detected and handled

Collision monitoring The collision monitoring is enabled automatically after the function block start and runs cyclically on an ongoing basis. The collision monitoring consists of two steps:

In operating states with large distances to the train at the front (such as traveling in an empty section), the train distance has to be greater than the deceleration path to the train in front + the safety distance "MB_MBELT_GENERAL.SafetyDistance". The collision monitoring signals when this condition is not met.

In operating states with very low distances to the train in front (such as catching up to the train in front), the train distance has to be greater than the safety distance MB_MBELT_GENERAL.SafetyDistance. The collision monitoring signals when this condition is not met.



Due to system conditions, the collision monitoring is based on position command values. Train overruns caused by poorly parameterized position control can cause collisions.

The collision monitoring can be disabled using the input **DisableCollisionDetection** = TRUE. However, this function should be used exclusively for moving trains away from each other after a collision. As soon as the trains have traveled away from each other, the collision monitoring is to be enabled again.

Dynamic monitoring Set the "MagicBelt" application so that the motion dynamics (such as acceleration, velocity) while traveling between "In- and Outfeed" positions is set to greater or equal values with regard to the "Infeed or Outfeed" motion. Only this ensures that a train that executes an "Infeed or Outfeed" motion does not collide with a slower train in front. The function block generates an error and decelerates all axes with the "MB_MBELT_GENERAL.StopRamp" ramp if the dynamic data does not correspond to the rules mentioned above.

Monitoring unloading When products are removed with a gripper, products can remain in the train if an error occurs. Incomplete product removal has thus - if necessary - to be detected outside the "MagicBelt" function block. If such a case is detected externally, the chains can be stopped using the "Stop" input. Following a positive edge at "Start", the operation continues.

Troubleshooting In the event of an error, all trains are always stopped with ramp. It does not make a difference which train caused the error.

Restart, automatic mode, MagicBelt

After troubleshooting, the error can be deleted via a positive edge at the **ClearError** input. With a positive edge at "Start", the trains start according to the specified start option. With the start option "START_AT_LAST_STEP", the state before the interruption can be restored and a train that is still loaded can travel to the unloading position for example.

4.4.5 Commissioning notes, MagicBelt

The following has to be considered when commissioning the "MB_MagicBelt-Type01" function block:

- One individual instance of the function block has to be called per chain.
- The function block "MB_MagicBeltType01" has to be called cyclically in a **sercos-synchronous task**. The task watchdog has to be switched on (sensitivity = 1)
- The axes of the chain drive **Chain A, B, (C)** have to be created using IndraWorks.

Observe the following parameterization:

- Axis type: Real or virtual (for testing)
- When the drive is added, "Interpolation in the drive" **has to be de-selected** (command position value generation on the control)
- Scaling type "**Translatory, Modulo, Unit [mm]**" has to be set.
- The modulo value has to be set as follows: **Modulo value = Chain-Length/TrainsPerChain**
- The **InfeedAxis** has to be created using IndraWorks.

Observe the following parameterization:

- Axis type: **Virtual**
- Scaling type "**Translatory, Modulo, Unit [mm]**" has to be set.
- The modulo value has to be set as follows: **Modulo value = Chain-Length/TrainsPerChain**
- The **dummy master** axis has to be created in IndraWorks if no other axis is available. All axis types are allowed. No special parameterization is required, since the axis is only passively used
- For synchronous loading, the **InfeedMaster** axis has to be created with IndraWorks as well.

Observe the following parameterization:

- In principle, all axis types (real, virtual, encoder...) can be used.
- Scaling type "**translatory, rotary**" has to be set.
- Modulo value = **360°**
- The axes created in IndraWorks have to be assigned to the corresponding axes (e.g. "ChainAxes") via the data structure [MB_MBELT_GENER-AL, page 189](#), (via AxisRef).

4.5 Sag Control

4.5.1 Introduction and Overview

Sag controls are used to decouple system components.

The following belt system is a typical use case.

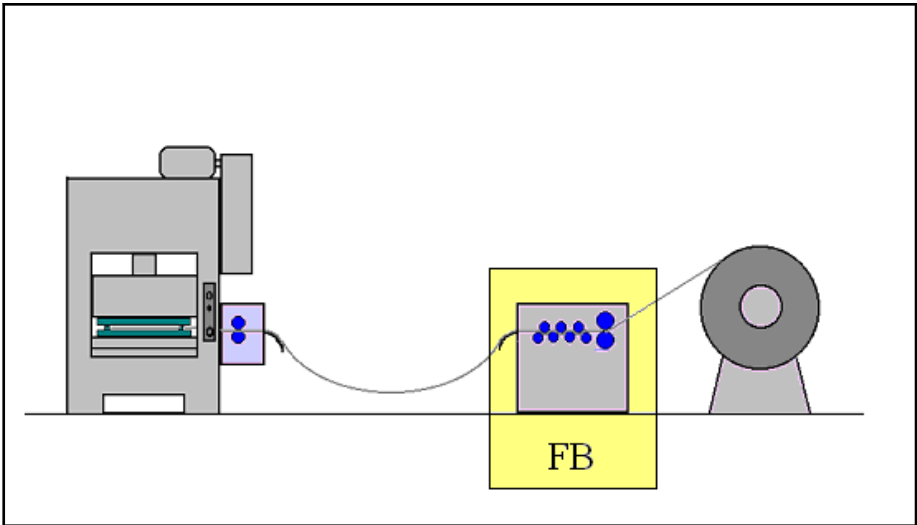


Fig. 4-36: Belt system, concept sketch

The sag is located between the punching machine feed rolls and the up-stream feed system. It is used to decouple both motion systems (start/stop or continuously). The sag depends on the roll feed data.

In normal operating mode one area of the belt system runs continuously at an average velocity v specified by the operator.

The other system area runs in start-stop operation. While the sheet is formed, this part of the system goes to stop mode and the sag accumulates. When the material is tracked for the next process, the sag contents are removed again. In one work cycle, the sag always move between the min./max. contacts.

Prerequisites for using the function block for sag control:

When setting up the belt system, the new sheet belt has to be fed from the coil through the straightener to the punching machine such that a sag already exists between the min./max. contacts (or min./max. analog values).

- [ML_SagControlC](#), page 219, controls the sag with min./max. contacts.
- [ML_SagControlA](#), page 223, controls the sag with analog switch signals.

4.5.2 ML_SagControlC

Brief description The "ML_SagControlC" function block contains a sag controller with binary minimum and maximum contacts.

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechMotion

Tab. 4-31: Reference table of the ML_SagControlA

Interface description

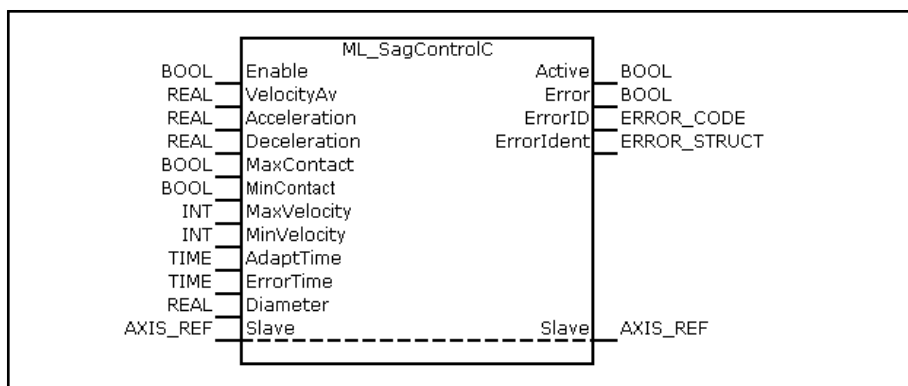


Fig. 4-37: ML_SagControlC function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Activates/deactivates the closed-loop velocity control
	VelocityAv	REAL	Average velocity of the belt system
	Acceleration	REAL	Acceleration for velocity changes
	Deceleration	REAL	Deceleration for velocity changes
	MaxContact	BOOL	Maximum contact of the sag
	MinContact	BOOL	Minimum contact of the sag
	MaxVelocity	INT	Maximum velocity in % based on the average velocity of the belt system
	MinVelocity	INT	Minimum velocity in % based on the average velocity of the belt system
	AdaptTime	TIME	Adaptation time (Ta)
	ErrorTime	TIME	Error time (Te)
	Diameter	REAL	Coil diameter ⇒ Evaluation of the velocity in %. If the input is not set, a default value of 100% is set
VAR_OUTPUT	Active	BOOL	Sag control is running
	Error	BOOL	Indicates an error. "Enable" = FALSE clears the error
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description depending on the error table Error handling, page 221
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Slave axis

Tab. 4-32: Interface variables of the ML_SagControlC function block

Functional description

The input "Enable" activates/deactivates the closed-loop velocity control. The first active velocity is the average velocity of the system specified by the operator.



If the "Enable" input is reset, the drive to be controlled is set in stop mode.

The function block for the sag control function exclusively affects the drive for material feed. This is an attempt to keep the sag in a defined window (min./

max. value) by adjusting the current velocity of the belt system-based on each deviation, with an optional sensor for detecting the coil diameter.

When the minimum contact is reached, the velocity of the material feed system increases to the maximum velocity associated with the function block to build up a greater sag. If movement away from the minimum contact towards the maximum contact occurs again in the error reaction time " T_e ", the greater velocity is used for operation for the rest of the adaptation time " T_a " so that the operation continues at the average velocity specified. If the sag becomes so big that the maximum contact responds, the velocity is reduced to the minimum velocity applied to the function block. If it moves away from the maximum contact in the error reaction time " T_e ", the drive continues to operate at the reduced velocity for the remaining time " T_a ". Afterwards, the operation continues at the specified average velocity (see v-t diagram).

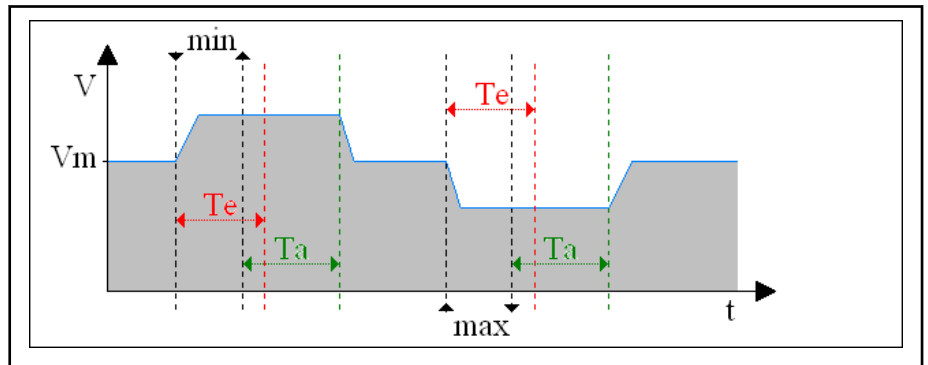


Fig. 4-38: Correct function block behavior

Error handling

If the minimum contact is pending for a period longer than " T_e ", an error is generated and the operation continues at the same velocity. This state remains until the contact is left again. Afterwards, the drive still operates at the increased velocity for the rest of the adaptation time " T_a " and then at the average velocity specified. However, if the maximum contact is pending for the period " T_e ", velocity is reduced to zero. If the maximum contact switches again, the drive remains stopped for the period " T_a " and then accelerates again to the specified average velocity " V_m " and continues to operate at this velocity.

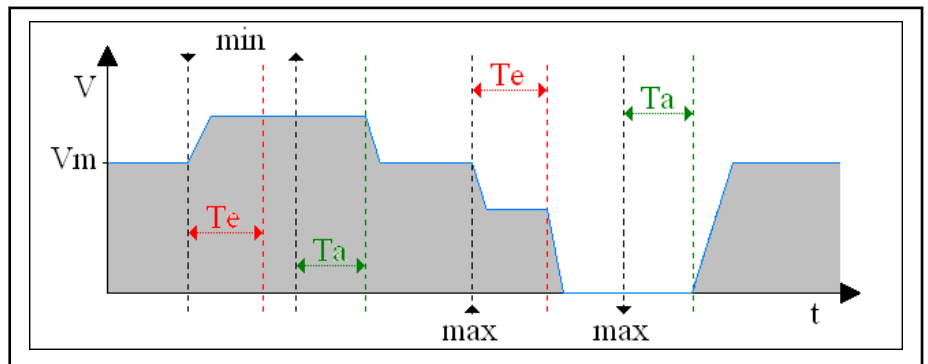


Fig. 4-39: Incorrect function block behavior

The function block generates the following error messages in Additional1/Additional2 for the table **F_RELATED_TABLE**, 16#0170:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Inputs outside of the valid range: Velocity <= 0 Acceleration <= 0 Deceleration <= 0 Max. velocity. <= Min. velocity Adaptation time <= 0 Error reaction time <= 0 Coil diameter <= 0
SYSTEM_ERROR (16#7FFF)	16#0501	16#0000	Velocity adaptation failed, adaptation time and error time have elapsed, maximum contact is still present
SYSTEM_ERROR (16#7FFF)	16#0502	16#0000	Velocity adaptation failed, adaptation time and error time have elapsed, minimum contact is still present
SYSTEM_ERROR (16#7FFF)	16#0503	16#0000	Maximum and minimum contacts are present at the same time. Contact problem ?
SYSTEM_ERROR (16#7FFF)	16#0504	16#0000	The predefined velocity could not be reached. Check velocity specifications

Tab. 4-33: Error codes of the ML_SagControlC function block

Firmware, software and hardware requirements

The following firmware, software and hardware is required:

- Hardware - IndraDrive C or M
- Firmware - Drive firmware (function package - Closed-Loop)
- Software - IndraWorks Engineering 14VRS

Application example 1

In the simplest form 2 contacts, 1 minimum and 1 maximum contact are available as closed-loop control parameters.

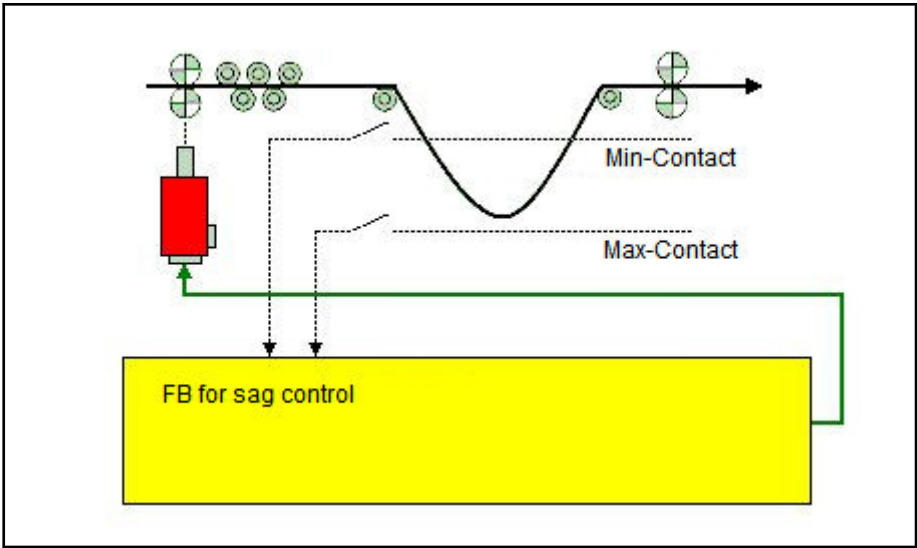


Fig. 4-40: Application 1, ML_SagControlC

In normal operation mode, the belt system runs at an average velocity "Vm" specified by the operator. The function block for the sag control now attempts to keep the sag in a defined window (min./max. contact) by adjusting the current velocity of the belt system-based on each deviation.

Application example 2

In this variant, the drive to be controlled is the coil itself. The decreasing coil diameter is used as a controlled variable in addition to the min/max contacts.

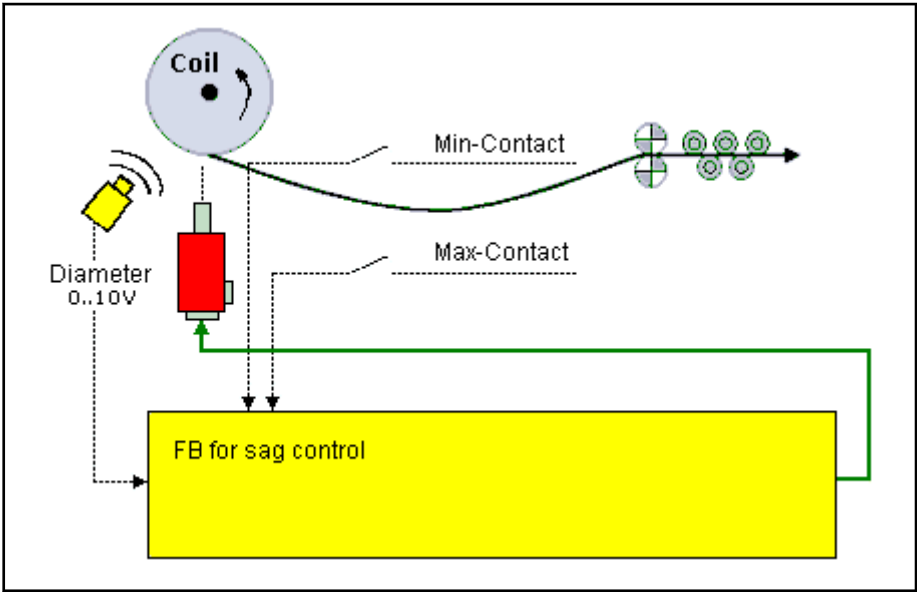


Fig. 4-41: Application 2, ML_SagControlC

4.5.3 ML_SagControlA

Brief description The function block "ML_SagControlA" contains a sag controller with an analog actual value measurement for the sag.

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechMotion

Tab. 4-34: Reference table of the ML_SagControlA

Interface description

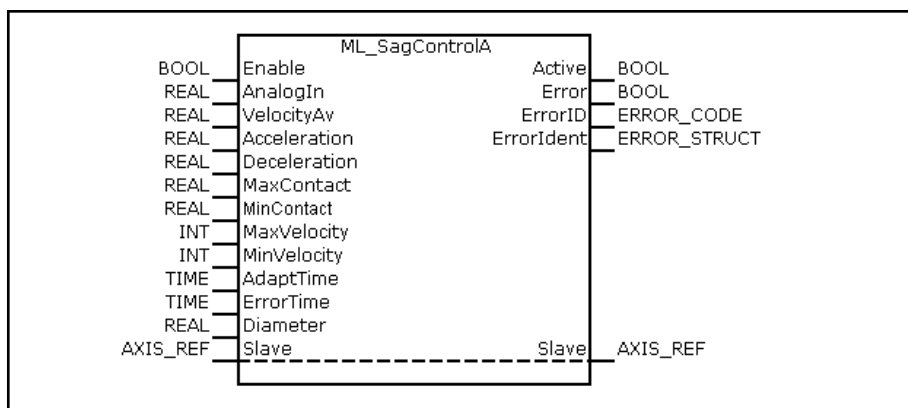


Fig. 4-42: ML_SagControlA function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Activates/deactivates the closed-loop velocity control
	AnalogIn	REAL	Analog value for sag (0 - 10)
	VelocityAv	REAL	Average velocity of the belt system
	Acceleration	REAL	Acceleration for velocity changes
	Deceleration	REAL	Deceleration for velocity changes
	MaxContact	REAL	Maximum contact of the sag
	MinContact	REAL	Minimum contact of the sag
	MaxVelocity	INT	Maximum velocity in % based on the average velocity of the belt system
	MinVelocity	INT	Minimum velocity in % based on the average velocity of the belt system
	AdaptTime	TIME	Adaptation time (Ta)
	ErrorTime	TIME	Error time (Te)
	Diameter	REAL	Coil diameter → Evaluation of the velocity in %. If the input is not set, a default value of 100% is set
VAR_OUTPUT	Active	BOOL	Sag control is running
	Error	BOOL	Indicates an error. "Enable" = FALSE clears the error
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description depending on the error table Error handling, page 225
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Slave axis

Tab. 4-35: Interface variables of the ML_SagControlA function block

Functional description

The "Enable" input activates/deactivates the closed-loop velocity control. The first active velocity is the average velocity of the system specified by the operator.



If the "Enable" input is reset, the drive to be controlled is set in stop mode.

The function block for the sag control function exclusively affects the drive for material feed. This is an attempt to keep the sag in a defined window (sensor analog value) by adjusting the current velocity of the belt system-based on each deviation, with an optional sensor for detecting the coil diameter.

When the minimum contact is reached, the velocity of the material feed system increases to the maximum velocity associated with the function block to build up a greater sag. If movement away from the minimum contact towards the maximum contact occurs again in the error reaction time " T_e ", the greater velocity is used for operation for the rest of the adaptation time " T_a " so that the operation continues at the average velocity specified. If the sag becomes so big that the maximum contact responds, the velocity is reduced to the minimum velocity applied to the function block. If it moves away from the minimum contact in the error reaction time " T_e ", the drive continues to operate at the reduced velocity for the remaining time " T_a ". Afterwards, the operation continues at the specified average velocity (see v-t diagram).

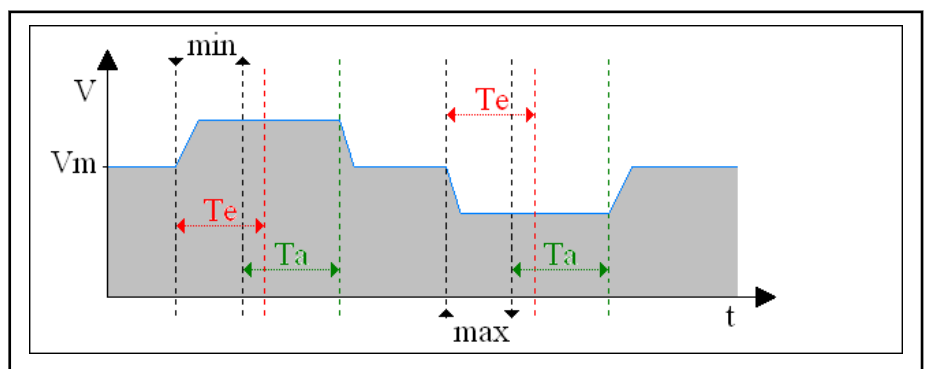


Fig. 4-43: Correct function block behavior

Error handling

If the minimum contact is pending for a period longer than " T_e ", an error is generated and the operation continues at the same velocity. This state remains until the contact is left again. Afterwards, the drive still operates at the increased velocity for the rest of the adaptation time " T_a " and then at the average velocity specified. However, if the maximum contact is pending for the period " T_e ", velocity is reduced to zero. If the maximum contact switches again, the drive remains stopped for the period " T_a " and then accelerates again to the specified average velocity " V_m " and continues to operate at this velocity.

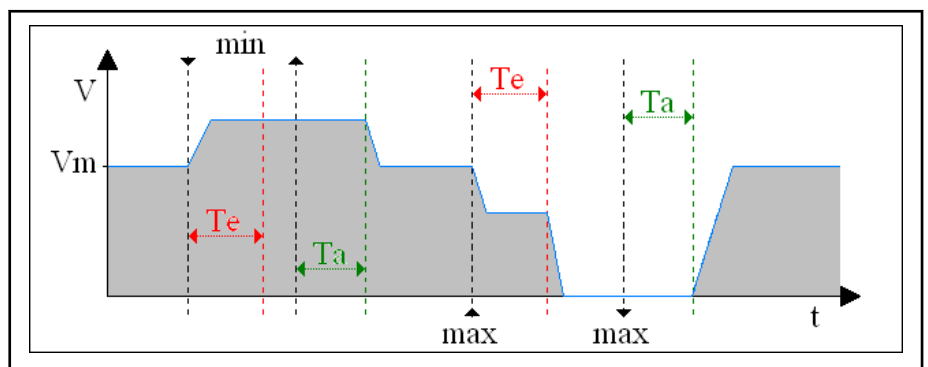


Fig. 4-44: Incorrect function block behavior

The function block generates the following error messages in Additional1/Additional2 for the table **F_RELATED_TABLE**, 16#0170:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#0002	16#0000	Inputs outside of the valid range: Velocity ≤ 0 Acceleration ≤ 0 Deceleration ≤ 0 Max. velocity. $\leq$ Min. velocity Adaptation time ≤ 0 Error reaction time ≤ 0 Coil diameter ≤ 0
SYSTEM_ERROR (16#7FFF)	16#0501	16#0000	Velocity adaptation failed, adaptation time and error time have elapsed, maximum contact is still present
SYSTEM_ERROR (16#7FFF)	16#0502	16#0000	Velocity adaptation failed, adaptation time and error time have elapsed, minimum contact is still present
SYSTEM_ERROR (16#7FFF)	16#0504	16#0000	The predefined velocity could not be reached. Check velocity specifications

Tab. 4-36: Error codes of the ML_SagControlA function block

Firmware, software and hardware requirements

The following firmware, software and hardware is required:

- Hardware - IndraDrive C or M
- Firmware - Drive firmware (function package - Closed-Loop)
- Software - IndraWorks Engineering 14VRS

Application example 1

In this variant, a sensor reproduces the sag position as an analog input (0-10V) which is added to the closed-loop velocity control.

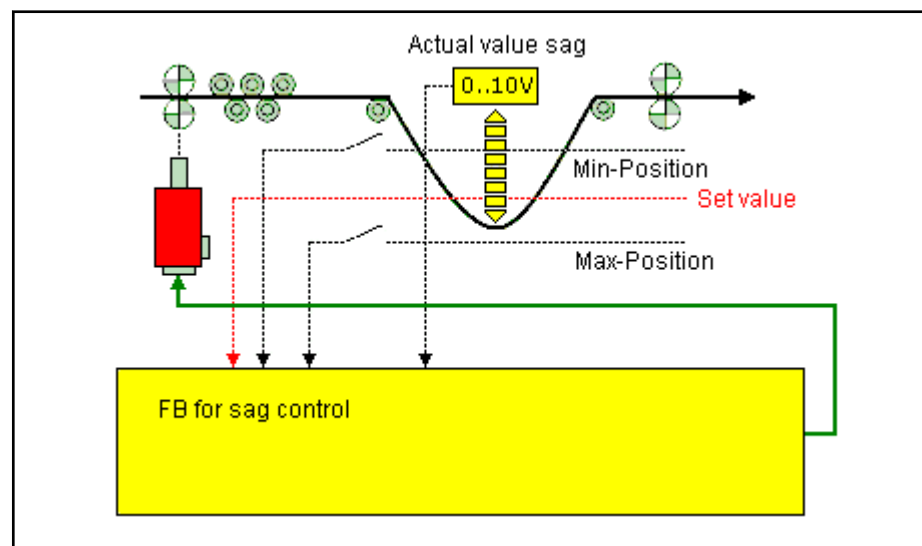


Fig. 4-45: Application 1, ML_SagControlA

In normal operation mode, the belt system runs at an average velocity V_m specified by the operator. The function block for the sag control now attempts

to keep the sag in a defined window (min./max. position) by adjusting the current velocity of the belt system-based on each deviation.

Application example 2

In this variant, the drive to be controlled is the coil itself. The decreasing coil diameter is used as a controlled variable in addition to the analog input.

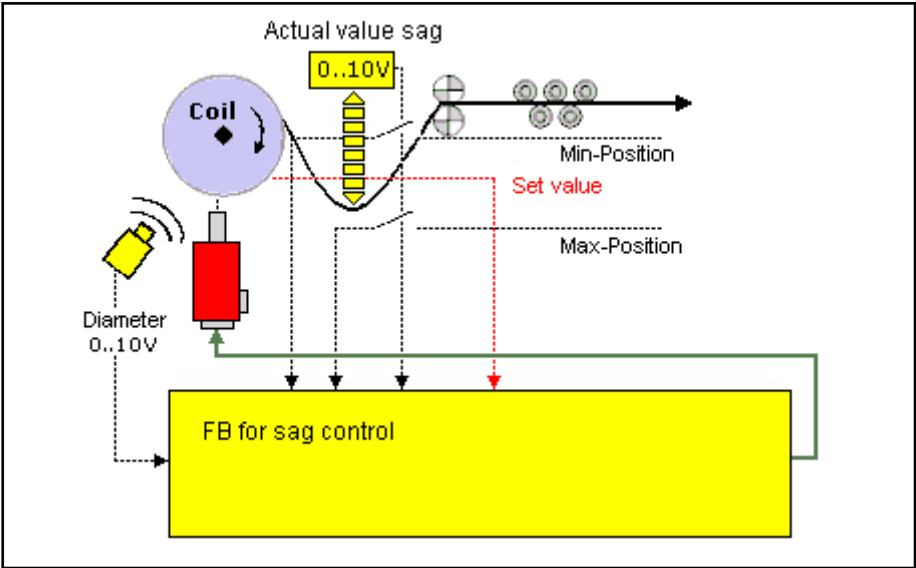


Fig. 4-46: Application 2, ML_SagControlA

4.6 SmartBelt

4.6.1 MB_SmartBeltType01

Brief description The MB_SmartBeltType01 function block measures the phase offset between the position of a product and the desired master position. If a phase offset is detected, a phase correction is applied to the position of the product, using a triangular velocity shape.

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechMotion
MLD with MPx07/08/17 with MA function package	MX_Packaging_07.lib

Tab. 4-37: Reference table of the MB_SmartBeltType01

Interface description

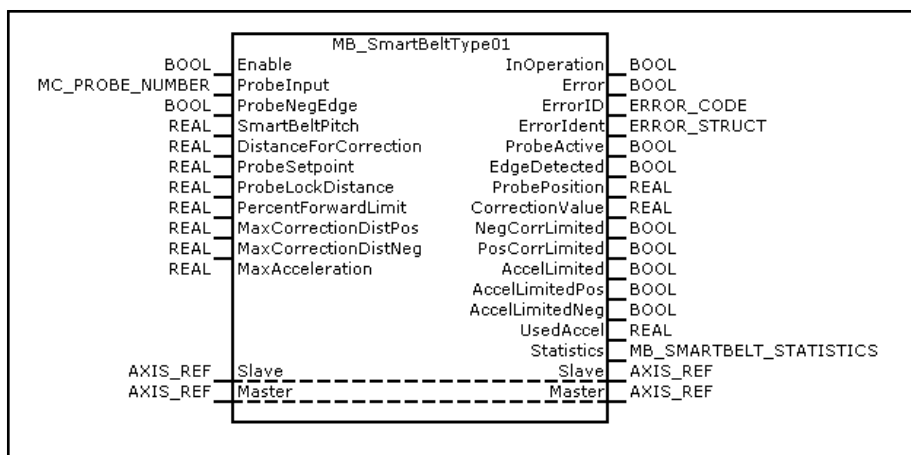


Fig. 4-47: MB_SmartBeltType01 Function Block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Rising edge enables function block
	ProbeInput	MC_PROBE_NUMBER	Selection of input for probe on drive (1= Probe1, 2= Probe2). Is read at the rising edge at the "Enable" input.
	ProbeNegEdge	BOOL	FALSE: Positive edge of touch probe is evaluated TRUE: Negative edge of touch probe is evaluated. Is read at the rising edge at the "Enable" input.
	SmartBeltPitch	REAL	Effective SmartBelt length (inches [in] or millimeters [mm]). Is read at the rising edge at the "Enable" input.
	DistanceForCorrection	REAL	Distance left on the belt for correction. This is usually the distance from where the touch probe is mounted to the end of the belt. Is read at the rising edge at the "Enable" input.
	ProbeSetpoint	REAL	Command value of probe (inches [in] or millimeters [mm]) Read cyclically
	ProbeLockDistance	REAL	Distance within which touch probe events are ignored (inches [in] or millimeters [mm]). If a touch probe event is detected, the next touch probe event is not considered until the belt has traveled the distance in "ProbeLockDistance". Read cyclically

I/O type	Name	Data type	Description
	PercentForwardLimit	REAL	The percentage of the total product cycle that determines the distribution between forward and backward correction. Example: A value of 70.0 for a product cycle length of 100mm would correct up to 70mm in forward direction and up to 30mm in backward direction. Read cyclically
	MaxCorrectionDistPos	REAL	Maximum correction distance for positive correction (inches [in] or millimeters [mm]). Read cyclically
	MaxCorrectionDistNeg	REAL	Maximum correction distance for negative correction (inches [in] or millimeters [mm]). Read cyclically
	MaxAcceleration	REAL	Maximum allowed acceleration in units of drive (in/s ² , mm/s ²). Read cyclically
VAR_OUTPUT	InOperation	BOOL	SmartBelt is operating
	Error	BOOL	Indicates an error. Clear error with "Enable" = FALSE
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Error handling, page 232
	ProbeActive	BOOL	SmartBelt touch probe functionality active. Waiting for event
	EdgeDetected	BOOL	Touch probe event detected
	ProbePosition	REAL	Captured position in drive units based on "SmartBelt-Pitch"
	CorrectionValue	REAL	Correction value in drive units based on "SmartBelt-Pitch"
	NegCorrLimited	BOOL	Negative correction is limited to distance set in "Max-CorrectionDistanceNeg"
	PosCorrLimited	BOOL	Positive correction is limited to distance set in "Max-CorrectionDistancePos"
	AccelLimited	BOOL	Correction is limited to acceleration set in "MaxAcceleration"
	AccelLimitedPos	BOOL	Positive acceleration is limited
	AccelLimitedNeg	BOOL	Negative acceleration is limited
	UsedAccel	REAL	Calculated acceleration used during phase correction (in/s ² or mm/s ²)
	Statistics	MB_SMART-BELT_STATISTICS	Structure containing different statistical data MB_SMARTBELT_STATISTICS, page 233

I/O type	Name	Data type	Description
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Address of SmartBelt axis
	Master	AXIS_REF	Address of master axis

Tab. 4-38: Interface variables of the MB_SmartBeltType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value
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	Modulo value of the master axis	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: Overview on min./max. values and initial values of the inputs

Functional description**Detailed description of some input and output variables**

The measurement units have to be equal for all entries (inches [in] or millimeters [mm]). If the "SmartBeltPitch" is for example specified in mm, the "DistanceForCorrection" also has to be in mm and the "MaxAcceleration" in mm/s².

"PercentForwardLimit" determines the distribution of the forward and backward correction. If the product should only be corrected in forward direction for example, a value of 100.0 has to be entered. That compensates all possible position errors by a forward correction. This value indicates a percentage distribution to the front or to the back.

The maximum measured error for a product can range from (-0.5 * product cycle) to (+0.5 * product cycle). The term product cycle is identical to the term "CamShaftDistance" or corresponds to the product length plus the gap between the products. Refer to the following figure as an example:

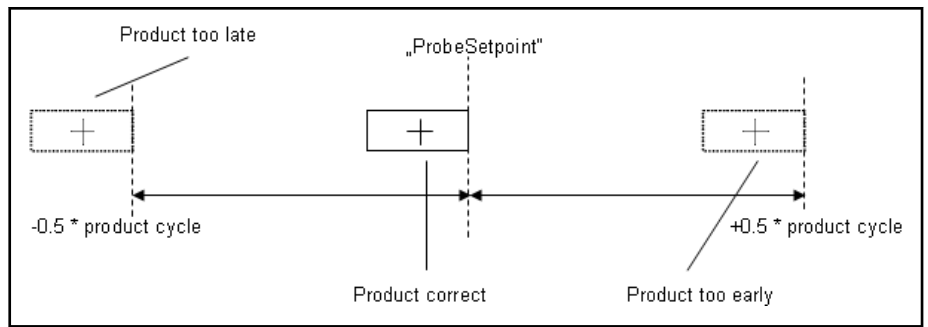


Fig. 4-48: Meaning of the correction in positive or negative direction

The product can now be corrected in either the positive or negative direction. Thus, allowing the correction of the complete product cycle. "PercentForwardLimit" specifies how the measured error is corrected in relation to the product cycle length. If "PercentForwardLimit" is set to 50%, half of the error is corrected forward and half backward.

If "PercentForwardLimit" is set to 70%, corrections are made as follows:

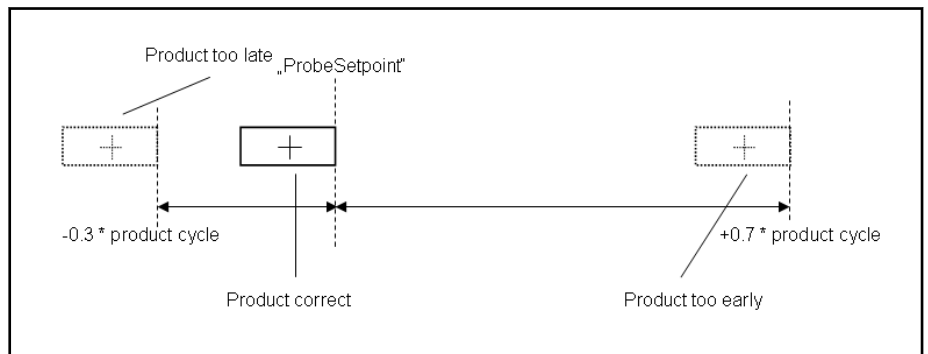


Fig. 4-49: PercentForwardLimit

"MaxCorrectionDistancePos" and "MaxCorrectionDistanceNeg" limit the amount of absolute distance that can be compensated in either direction. Together with "PercentForwardLimit", combinations can be created to compensate for all errors or just fractional errors in either direction. For example, to correct 80% of all errors in the positive direction (correcting no more than 60mm in the positive direction and no more than 20mm in the negative direction of the length of a complete product cycle) configure the inputs as follows:

"PercentForwardLimit" = 80

"MaxCorrectionDistancePos" = 60

"MaxCorrectionDistanceNeg" = -20

Explanation: For a 200mm long product cycle, with "PercentForwardLimit" = 80, 80% (160 mm) is corrected forward and 20% (40mm) backward. "MaxCorrectionDistancePos" = 60 limits the available 160mm to only 60mm while "MaxCorrectionDistanceNeg" = -20 limits the allowed backward correction to -20 mm.

The overall scenario is described in the following:

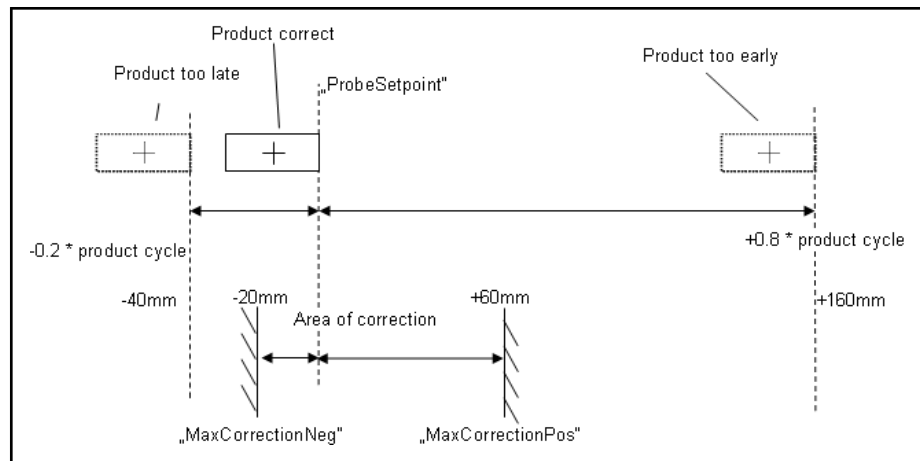


Fig. 4-50: *MaxCorrectionDistancePos and MaxCorrectionDistanceNeg*

An error outside the limits of "MaxCorrectionDistancePos" and "MaxCorrectionDistanceNeg" is limited to "MaxCorrectionDistancePos" and "MaxCorrectionDistanceNeg". The remainder is corrected on the next belt. Thus, it is possible to distribute the corrections on all belts so that the first belt does not have the main correction load.

The higher the correction distance, the higher the used acceleration. Alternatively, the maximum acceleration can also be limited using "MaxAcceleration". Both maximum acceleration and maximum correction distance are active at the same time with the lower of the two restrictions being applied.

Time requirements

The function block requires up to 400 PLC cycles until initialization is finished. The exact amount depends on the load on the control.

Error handling

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0006)	16#0012	16#0000	Axis without power
RESOURCE_ERROR (16#0006)	16#0001	x	No power is added to the drive "x" indicates the address of the erroneous axis
ACCESS_ERROR (16#0004)	16#0003	16#0000	The internally used function block MB_TouchProbe-Continuous was interrupted by another function block
ACCESS_ERROR (16#0004)	16#0003	16#0001	The internally used function block MB_PhasingSlave was interrupted by another function block
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Error in state machine
INPUT_INVALID_ERROR (16#0001)	16#000D	x	Axis (master or slave axis) is not provided with a valid address. x indicates the address of the incorrect axis
INPUT_INVALID_ERROR (16#0001)	16#0F02	16#0010	SmartbeltPitch outside valid range (0..10000)
INPUT_INVALID_ERROR	16#0F02	16#0011	ProbeInput invalid (PROBE1, PROBE2)
INPUT_INVALID_ERROR	16#0F02	16#0012	DistanceForCorrection invalid (0..SmartBeltPitch)
INPUT_INVALID_ERROR	16#0F02	16#0013	MaxCorrectionDistPos invalid (0..SmartBeltPitch)

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR	16#0F02	16#0014	MaxCorrectionDistNeg invalid (-SmartBeltPitch..0)
INPUT_INVALID_ERROR	16#0F02	16#0015	Axis is not an IndraDrive axis (Pack profile not allowed)
INPUT_INVALID_ERROR	16#0F02	16#0017	Scaling of slave is not the preferred scaling
INPUT_INVALID_ERROR	16#0F02	16#0018	Scaling of master is not the preferred scaling
INPUT_INVALID_ERROR	16#0F02	16#0019	Acceleration value too small (>0)
INPUT_INVALID_ERROR	16#0F02	16#001A	Slave does not have modulo scaling
INPUT_INVALID_ERROR	16#0F02	16#001B	Master does not have modulo scaling
INPUT_INVALID_ERROR	16#0F02	16#001C	Settings in P-0-0154 (MLD)/A-0-2793 (MLC) have to be set to "ShortestPath"
INPUT_INVALID_ERROR	16#0F02	16#001D	Input ProbeSetpoint is out of limits (0..Master Modulo)
INPUT_INVALID_ERROR	16#0F02	16#001F	Input ProbeLockDistance is out of limits (0..SmartBeltPitch)
INPUT_INVALID_ERROR	16#0F02	16#0020	PercentForwardLimit is out of limits (0..100.0)
INPUT_INVALID_ERROR	16#0F02	16#0021	MaxCorrectionDistPos+ ABS(MaxCorrectionDistNeg) is greater than SmartBeltPitch
INPUT_INVALID_ERROR	16#0F02	16#0022	Modulo value of the slave axis has to be at least twice the size of the SmartBeltPitch
DEVICE_ERROR (16#0008)	16#0F06	16#0000	Scaling of velocity and scaling of position data is not the same

Tab. 4-40: Error codes of the MB_SmartBeltType01 function block

In addition, the function block can issue all the following errors of sub function blocks:

- MB_GetCyclicParameterHandle
- MB_PhasingSlave
- MB_ReadParameter
- MB_ReadRealParameter
- MB_ReadStringParameter
- MB_InitTouchProbe
- MB_TouchProbeContinuous

Refer to the relevant documentation for error codes of the respective function block.

4.6.2 MB_SMARTBELT_STATISTICS

This structure contains statistical data that can be used for monitoring and assessing the application. The individual elements are listed in the following table:

Structure element	Type	Default value	Description
CurRate	REAL	0.0	Product rate based on distance between current and previous product
NbrOnBelt	UINT	0	Number of products currently on the belt. A product is considered "on the belt", when it has traveled less than a whole SmartBelt length since registration
MaxGap	REAL	0.0	Maximum gap between two consecutive products in units based on "SmartBeltPitch"
MinGap	REAL	"SmartBeltPitch"	Minimum gap between two consecutive products in units based on "SmartBeltPitch"
TotalNbr	UINT	0	Total number of products registered by the SmartBelt

Tab. 4-41: MB_SMARTBELT_STATISTICS Structure

All elements in the MB_SMARTBELT_STATISTICS structure are set to their default values on the rising edge of the "Enable" input.



The values of the elements might not be accurate if products are not registered correctly.

The "CurRate" value is calculated based on time measurements. The resolution for the time is 1ms. For high product rates, the calculated rates may vary in the range of 1ms resolution. I.e., when a product rate of 400 products per minute is used, one product takes $1 / (400 / 60) = 150$ ms. Assuming a resolution of 1ms, we could measure values in the range of 149ms, 150ms and 151ms, resulting in product rates of 403, 400 and 397 respectively.

The values for "MaxGap" and "MinGap" are calculated based on the distance traveled by the master. This is especially noticeable on the first belt when it is accelerated or decelerated. In this case, the SmartBelt cover a lot more/less distance than the master axis.

4.6.3 Introduction to the SmartBelt use

SmartBelt use

SmartBelts distribute the products correctly before they are processed. Products are provided by a feeding unit to the SmartBelts where the phase is adjusted. Subsequently, a processing unit further handles the products (e.g., a flow wrapper).

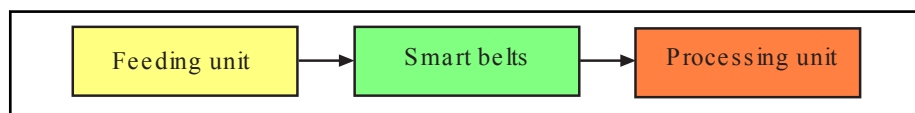


Fig. 4-51: Example a flow wrapper

Depending on the application, different situations can arise:

- Feeding units provide products with the correct distance to each other but the processing unit cannot synchronize to the feeding unit when starting or stopping. In this case, SmartBelts can correct the gap between products
- Feeding units provide products from a storage on demand. No SmartBelts are necessary
- Feeding units provide products with a random, but small variation in the gap between the products. SmartBelts can be used

- Feeding units provide products with a random variation in the gap between the products. SmartBelts can be used together with following additional measures:
 - Velocity of the feeding unit can be controlled to adjust a constant product rate
 - Velocity of the processing unit can be controlled to adjust to the current product rate

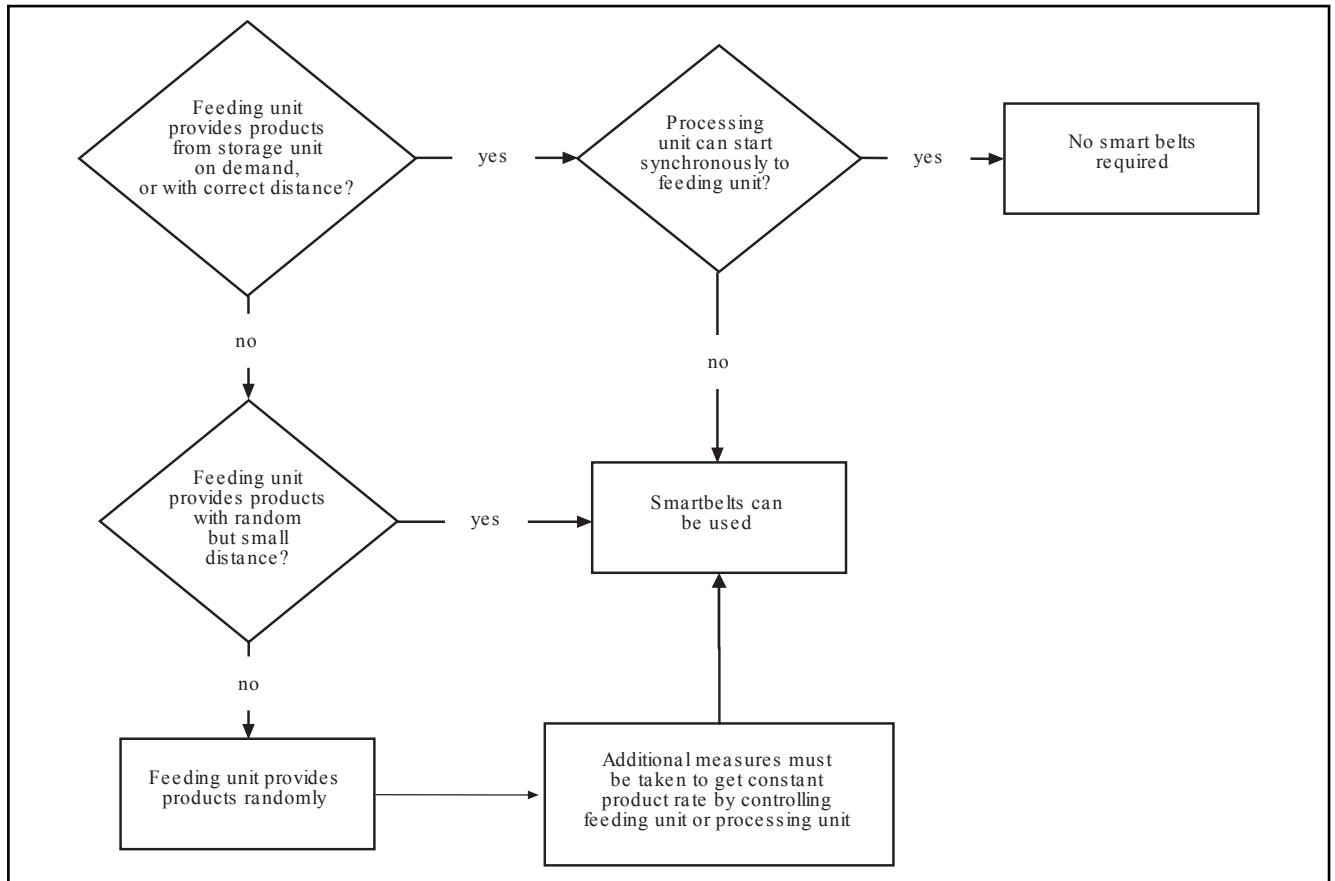


Fig. 4-52: Use of SmartBelts

Accumulating conveyor

Using an accumulating conveyor increases the order of the products. In the following figure, belt 2 is an accumulating conveyor. Its length (L2) has to be longer than the length of the products on the previous belt (L1). Otherwise, the products from belt 1 push the products on belt 2.

The velocity on the accumulating conveyor has to be less than the velocity on belt 1 in order to accumulate products.

Belt 3 has a higher velocity compared to the accumulating conveyor in order to pull the products apart by a predefined distance.

Gaps between the products on the accumulating conveyor result in larger gaps on belt 3. However, they are not necessarily a multiple of the desired distance. The velocity of the accumulating conveyor can be controlled in order to avoid this situation.

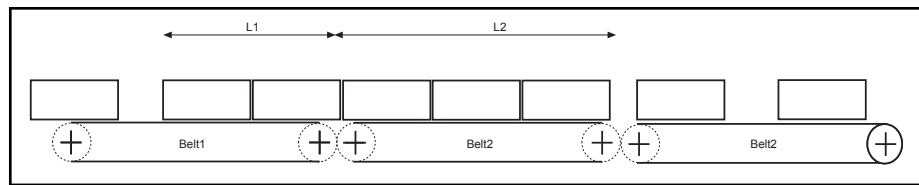


Fig. 4-53: Accumulating conveyor

Degree of order

Assessment using product cycles and gaps

The degree of order is critical to the number of SmartBelts necessary. The higher the order, the lower the number of SmartBelts. The degree of order directly affects the amount of correction for the SmartBelts. It is calculated with the following formula:

The following figure should explain the various parameters:

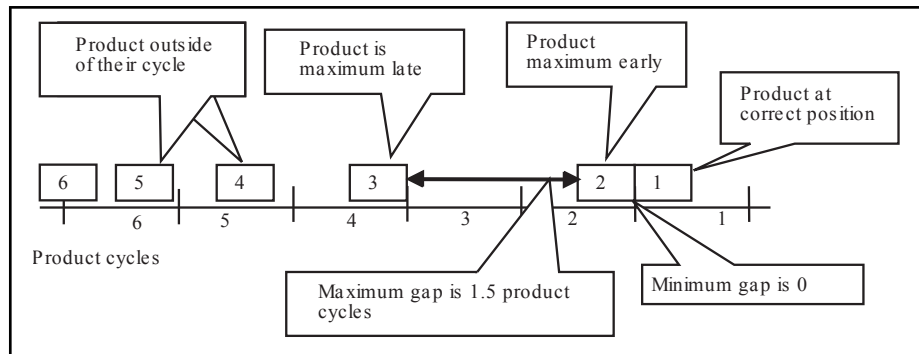


Fig. 4-54: Assessment using product cycles and gaps

The general assumption is that the leading edge of the product is point of reference. It does not matter if the leading, trailing edge or the center of the product is considered when making the following conclusions, as long as the same point of reference is used.

The figure above, product 1 is exactly where it should be and would not need any further correction. Product 2 is too close to product 1, but is still in the product cycle. It has a minimum gap of 0 to its predecessor. Product 3 is too late, but also in the product cycle. Products 4 and 5 are not in their respective product cycles while product 6 is in its product cycle.

The amount of correction is calculated using the following variables:

- Product length (L_{Prod})
- Nominal product gap at the given velocity (L_{gap})
- Nominal product cycle at the given velocity (L_{abs})
- Minimum gap between products (L_{Gapmin})
- Maximum gap between products (L_{Gapmax})
- Subsequent products outside the product cycle (N_{Abs})

The following example values apply to the above figure:

L_{Prod} : 40 mm (assumed)

L_{gap} : 50 mm (assumed)

L_{cycl} : 90mm ($= L_{Prod} + L_{gap}$)

$L_{gap min}$: 0 mm (gap between products 1 and 2)

$L_{gap max}$: 1.5 product cycle = $1.5 * L_{cycl} = 135\text{mm}$ (gap between products 2 and 3)

N_{gap} : 2 (gap between products 4 and 5)

The necessary amount of correction for the SmartBelts is calculated as follows:

$$\text{Correction} = \min \left(L_{\text{gap,max}}, \left(1.5 * L_{\text{gap,1}} - L_{\text{prod}} + L_{\text{gap}} * (N_{\text{gap}} - 1) - (N_{\text{gap}} * L_{\text{gap,min}}) \right) \right)$$

For this example, the following values apply:

$$\begin{aligned} \text{Correction} &= \min (135 \text{ mm}, (1.5 * 90 \text{ mm} - 40 \text{ mm} + 50 \text{ mm} * (2 - 1) - (2 * 0 \text{ mm}))) \\ &= \min (135 \text{ mm}, 145 \text{ mm}) \\ &= 135 \text{ mm} \end{aligned}$$

This is example, 135mm would be the amount of distance the SmartBelts would have to travel in order to correct product position in both directions.

Assessment using product rate and duration

A different method to estimate whether the correction capability is sufficient is to take a look at the product rate. The rate should be constant, but varies with time around an average value. The duration of that deviation and the size of the deviation can also be used to determine if a SmartBelt can support the application.

To calculate the deviation in time, use the following formula:

$$t = \frac{1}{\frac{R}{60} * \frac{\text{Deviation}[\%]}{100}}$$

Where R is the intended product rate in [1/min] and "Deviation" is the percentage by which the product rate is not the intended one.

Example:

If the intended product rate is 200 products per minute, and deviations occur which are 10% in size. This would mean that there can be 180 to 220 products per minute. The above formula results in a time of :

$$t = \frac{1}{\frac{200}{60} * \frac{10\%}{100}} = 3s$$

In case the product rate deviates from the nominal rate for less than 3 seconds, the SmartBelt can support the application. Otherwise, either the products are not corrected on the belt or empty places occur.

Friction

The connection between friction and possible acceleration is explained as follows:

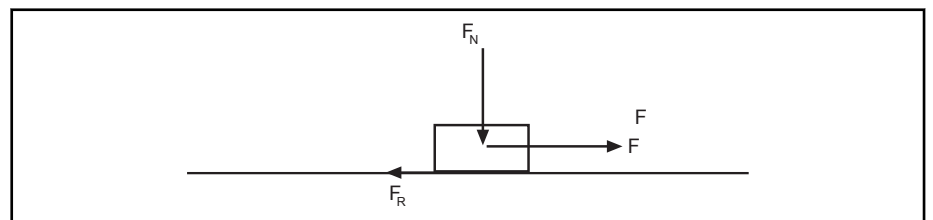


Fig. 4-55: Friction

Static friction is defined as:

$$F_R = \mu_H F_N$$

with

$$F_N = F_G = m \cdot g$$

and substituting the later by the first one

$$F_R = \mu_H \cdot m \cdot g$$

The necessary force for acceleration is:

$$F = m \cdot a$$

Dynamic friction starts, when $F = F_R$

or substituted

$$\mu_H \cdot m \cdot g = m \cdot a_{\max}$$

solved for $a_{\max}$:

$$a_{\max} = \mu_H \cdot g$$

$a_{\max}$ is the maximum acceleration, μ_H is the friction coefficient and g is the acceleration of gravity.

μ_H can also be visualized in a different way as the following figure shows. μ_H is the equivalent of $\tan \phi$. In other words, an object with a friction of 1 would start to slide on a plane at a 45° angle.

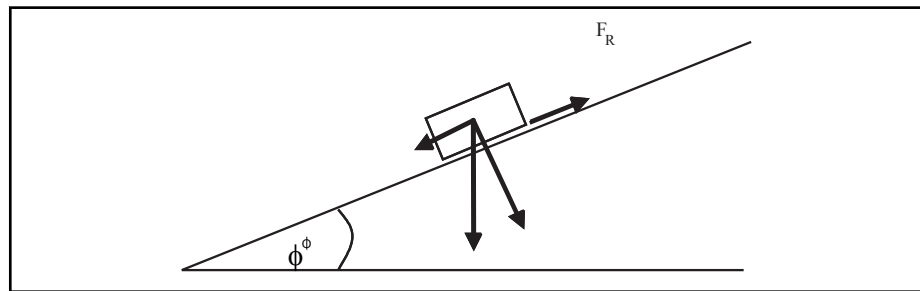


Fig. 4-56: Relation between friction coefficient and angle

4.6.4 Project planning

General information

This section describes the settings and calculations that has to be performed when using the MB_SmartBeltType01 function block.

Mechanical requirements for belt and machine

The belt should have a length that is approximately the size of the product plus the intended gap (product cycle). The belt length is the physical length of the whole belt, not the circumference. There is no benefit if the belt is significantly longer or shorter than the product cycle. In case of a longer belt, multiple products remain on a belt. Thus, causing the last product to wait for the first one to leave the belt. In case of shorter belts, there is less time and higher accelerations to create gaps between the products.

The touch probe for each belt should be connected to the corresponding drive input. Note which touch probe input is being used. Mount the touch probe as close as possible to the start of the belt in case the positive edge is detected. However, the product has to have its center of mass already on the belt. For symmetric products this is half of the product length. The touch probe should be mounted at a location that is half the product length plus half the axis diameter of the belt (as shown in).

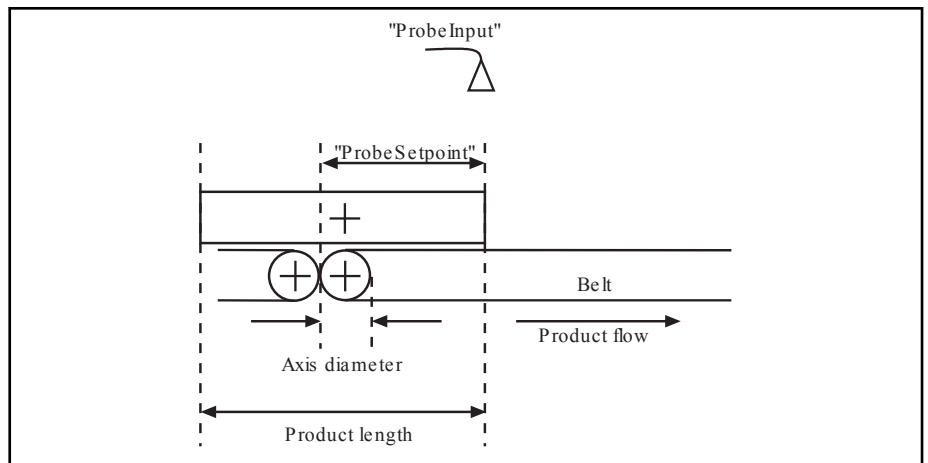


Fig. 4-57: Touch probe with detection of positive edge

If the negative edge of the product is used, similar logic applies. In this case, the touch probe should be located at a position half the product length minus half the axis diameter of the belt (before the belt) as shown in .

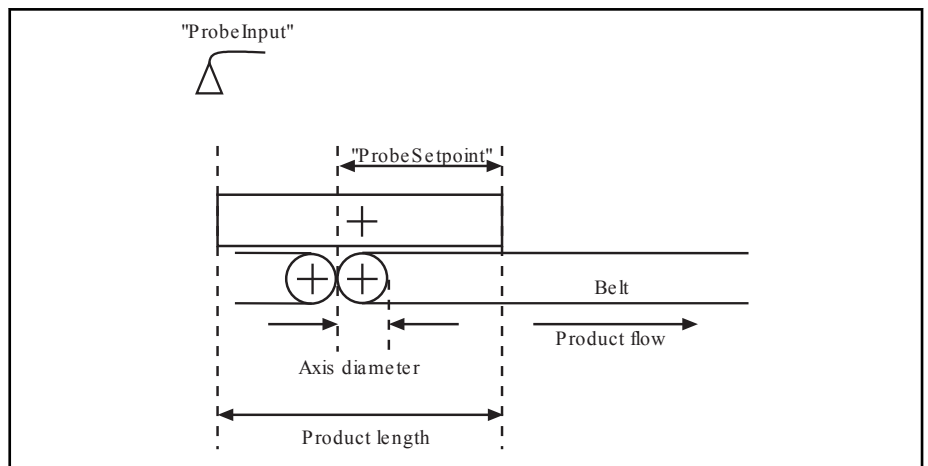


Fig. 4-58: Touch probe with detection of negative edge

For multiple products with different formats running on the same SmartBelt, the touch probe should be mounted for registration with the negative edge. In this case, the smallest product determines the location of the touch probe. In this case, the compensation capability of larger products is reduced.

Drive requirements

The drive has to have a function package that supports synchronous operation modes and touch probe inputs.

For MLC the following requirements apply:

- Only IndraDrives with interpolation in the drive can be used. The drive firmware has to be at least MPx04, V22 or higher. MPx03 is not supported

For MLD the following requirements apply:

- The drive firmware has to be at least MPx05 or higher. MPx03 and MPx04 are not supported.
- The MA package has to be activated

Drive parameter settings

It is recommended to load base parameters before configuring the drive. In this section, the differences to the default parameter are listed.

Before using the function block, some additional settings have to be made. Since the function block uses the touch probe functionality, the touch probe configuration has to be set properly. Depending on the touch probe input that is used and the edge for triggering, the following settings are necessary for the optional cyclic channel. In order to have different options at runtime, multiple settings can be configured. For example, place S-0-0130 and S-0-0131 as well as S-0-0409 and S-0-0410 in the cyclic read channel. This allows the use of either positive or negative edge at runtime.



The following parameters are set in the control to allow cyclic parameters to be transmitted to the drive. Additional parameters, set in the drive, are required for the touch probe functionality.

The IndraWorks drive parameter editor can be used to make the parameter settings.

MLC parameter settings

Desired configuration	Cyclic read access	Signal control word
Touch probe 1, positive edge	S-0-0130, S-0-0409	S-0-0405, bit 0
Touch probe 1, negative edge	S-0-0131, S-0-0410	S-0-0405, bit 0
Touch probe 2, positive edge	S-0-0132, S-0-0411	S-0-0406, bit 0
Touch probe 2, negative edge	S-0-0133, S-0-0412	S-0-0406, bit 0

MLD parameter settings

Desired configuration	Actual values	Command values
Touch probe 1, positive edge	S-0-0130 S-0-0409	S-0-0405, bit 0
Touch probe 1, negative edge	S-0-0131 S-0-0410	S-0-0405, bit 0
Touch probe 2, positive edge	S-0-0132 S-0-0411	S-0-0406, bit 0
Touch probe 2, negative edge	S-0-0133 S-0-0412	S-0-0406, bit 0

Touch probe functionality in the drive

The following parameters are set in the drive to enable the drive touch probe functionality:

Parameters	Value	Description
P-0-0300	S-0-0401 (touch probe 1) or S-0-0402 (touch probe 2)	Digital input assignment of touch probe functionality
P-0-0301	0 (Bit 0)	Bit assignment for the touch probe functionality
S-0-0426	P-0-0775	Master position for touch probe measurement

S-0-0169	Bit 0 to 3	Activation of positive or negative edge detection for touch probe 1 and 2
P-0-0226	Bit 9 = TRUE	Touch probe function is automatically enabled upon entering phase 4 in the drive

In addition to application-specific settings, the following parameters have to be set.

Hardware-independent settings

Parameters	Value	Description
P-0-0072 P-0-0092 P-0-0780 P-0-0781 P-0-0782 P-0-0783 P-0-0784 P-0-0785	Linear Cam with 1:1 relationship between master and slave. Only one parameter (Cam) has to be set. CamBuilder software can create and download a cam	The corresponding cam number (CamTableID) has to be referenced in the MC_CamIn function block when establishing synchronous operating mode
S-0-0121	Input revolutions of load gear	Mechanical gear, input revolutions
S-0-0122	Output revolutions of load gear	Mechanical gear, output revolutions
S-0-0123	Linear motion per revolution	Feed constant for the drive

MLC axis parameter settings

Parameters	Value	Description
A-0-0045	Twice the belt length	Modulo value of the axis
A-0-0059	Bit 0 = TRUE, bit 1 and 2 = FALSE, Bit 3 = FALSE, bit 7 = TRUE	Select linear, modulo and preferred scaling
A-0-0059	Bit 4 = FALSE for units in millimeters [mm] Bit 4 = TRUE for units in inches [in]	Selecting units of belt
A-0-2790	High enough for the application	Synchronization velocity
A-0-2791	High enough for the application	Synchronization acceleration
A-0-0032	High enough for the application	Positive velocity limit
A-0-0033	High enough for the application	Negative velocity limit
A-0-0034	High enough for the application	Acceleration limit value

The master axis used for the belt has to have a modulo and preferred scaling as well (A-0-0059: Bit 3 = FALSE, bit 7 = TRUE) as well as rotary or linear scaling. One master revolution converts to one product cycle (product length plus gap between products) for a rotary master. In case of a linear master, the modulo value has to correspond to the product cycle. If one master cycle corresponds to multiple/fractions of the product cycle, the gear ratio has to be set accordingly. For example, if one master revolution corresponds to three products, then the inputs for the gear should be as follows:

- "RatioNumerator"/"InputRevolution" should be set to 1
- "RatioDenominator"/"OutputRevolution" should be set to 3

MLD parameter settings

Parameters	Value	Description
S-0-0103	Twice the belt length	Modulo value of the axis
S-0-0076	Bit 0 = TRUE, bit 1 and 2 = FALSE, Bit 3 = FALSE, bit 7 = TRUE	Select linear, modulo and preferred scaling
S-0-0076	Bit 4 = FALSE for units in millimeters [mm] Bit 4 = TRUE for units in inches [in]	Selecting units of belt
P-0-0143	High enough for the application	Synchronization velocity
P-0-0142	High enough for the application	Synchronization acceleration
S-0-0038	High enough for the application	Positive velocity limit
S-0-0039	High enough for the application	Negative velocity limit
S-0-0138	High enough for the application	Acceleration limit value

The master for the belt has to have as well a modulo and preferred scaling (P-0-0756 in case of an axis generator), a rotary or linear scaling. One master revolution converts to one product cycle (product length plus gap between products) for a rotary master. In case of a linear master, the modulo value has to correspond to the product cycle. If one master cycle corresponds to multiple/fractions of the product cycle, the gear ratio has to be set accordingly in the PLC program. For example, if one master revolution corresponds to three products, then the inputs for the gear should be "RatioNumerator"/"InputRevolution" should be set to 1 and "RatioDenominator"/"OutputRevolution" should be set to 3.

PLC programming

The axis to be operated with the SmartBelt has to be in a synchronous operation mode. The following code fragments show how to set it.

Structured text:

Program:

```

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=>
);

```

Function block diagram:

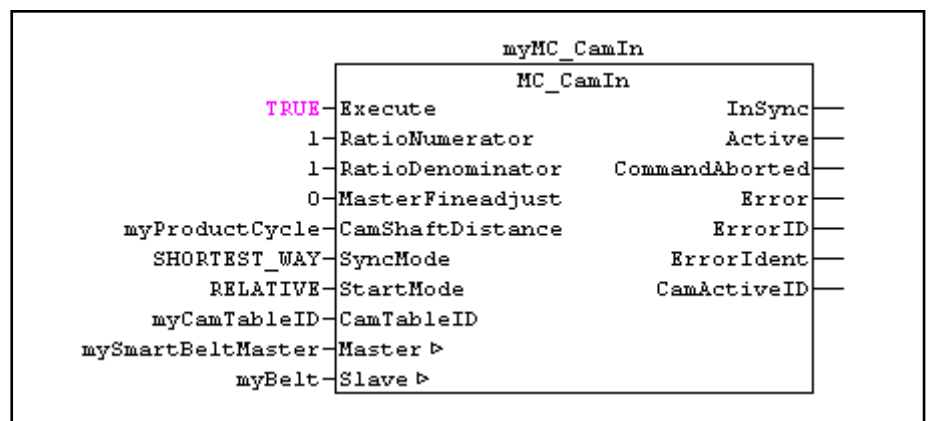


Fig. 4-59: MC_CamIn settings for use with SmartBelt

When programming the **AxisInterface**, the operating mode for the drive has to be set as well to "CamIn". Refer to the following window for an example:

Structured text:

Program:

```
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;
```

The synchronization distance has to be set to SHORTEST_WAY to guarantee that the SmartBelt moves forward and backward.

The starting mode has to be set to RELATIVE. This allows the user to synchronize the SmartBelt to the master in standstill mode without the belt moving.

The "CamShaftDistance" input has to be set to the product cycle of the application. This is the length of the product plus the gap between the products. Given this length, the SmartBelt determines whether to correct forward or backward. Even if there no 1:1 relationship between the master axis and SmartBelt exists, this value has to be set for the product cycle. In this case, use the gear to choose the relationship between master axis and SmartBelt.

The cam ID used in the drive can be chosen freely but has to be set accordingly in the "CamTableID" element. The cam has to be downloaded to the drive before using it. The cam has to be in the ratio of 1:1. It can be created with the CamBuilder software. To reduce the download time, the number of cam points can be reduced to the minimum possible points.

The gear ratio used is 1:1 which follows a 1:1 ratio between the master cycle and the product cycle. If this ratio is different, the gear ratio has to be set accordingly.

Specification of the "SmartBeltPitch"

The "SmartBeltPitch" indicates the physically effective length of the SmartBelt. The SmartBelt length should approximately be equal to the product

length + gap between the products (= product cycle). The touch probe is mounted at the beginning of the SmartBelt and detects either the beginning or the end of the product. The SmartBelt function block assumes that the product has to move further at least by the "SmartBeltPitch" after detected by the touch probe and before leaving the belt. Only then, the following product can be corrected. This can be seen in the following. The product cycle is a bit smaller than the length of the SmartBelt. In the exemplary case, product 1 moves up to the dashed line before the following product (2) can be corrected.

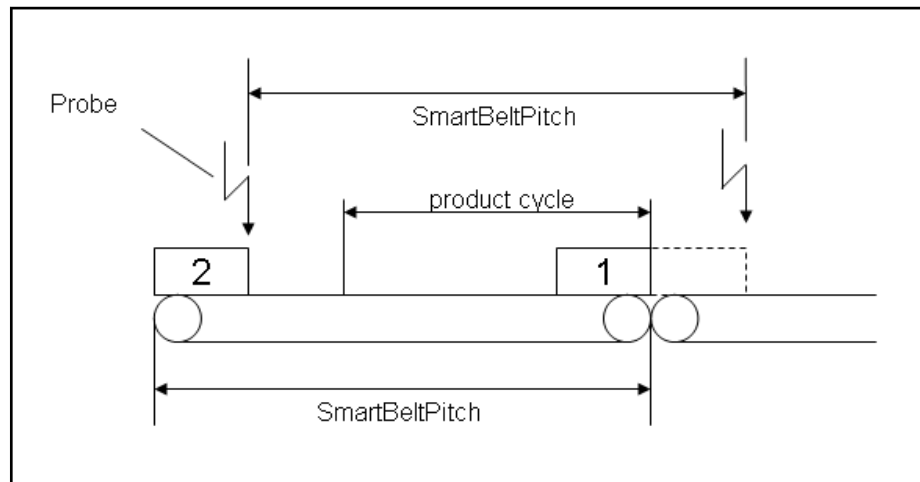


Fig. 4-60: SmartBelt with physical length approximately corresponds to the product cycle

These assumptions are correct as long as the touch probe is mounted at the beginning of the SmartBelt in the area that does not exceed the product length (see following figure). In this case, the SmartBelt is slightly longer than two product cycles. Starting from the second belt, a product cycle A is also drawn, but backwards seen from the touch probe of the second belt. This shows that the following product is always still on the first belt when the first product had traveled the "SmartBeltPitch" length. Subsequently, the second product can be corrected.

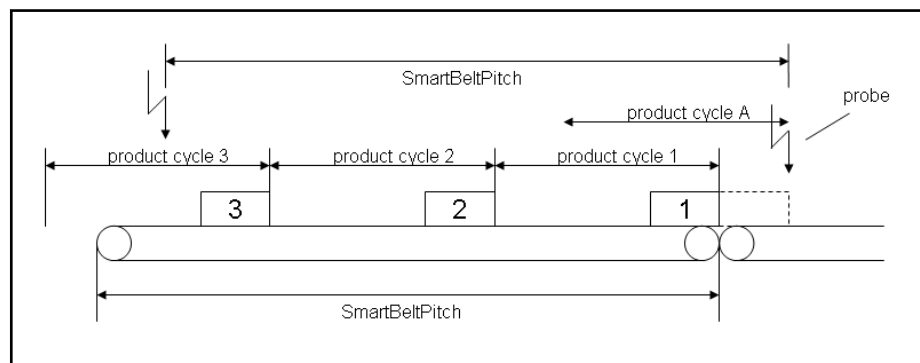


Fig. 4-61: SmartBelt with physical length corresponding to a multiple of the product cycle

If the touch probe is not mounted at the beginning of the SmartBelt, the SmartBelt does not operate correctly since it wants to keep on moving a complete "SmartBeltPitch" length after the product detection. In this case, the product left the belt long before the following product is corrected. In the case drawn, the following product would also have left the belt since the product cycle A is completely on the following belt. The following product cannot be corrected anymore.

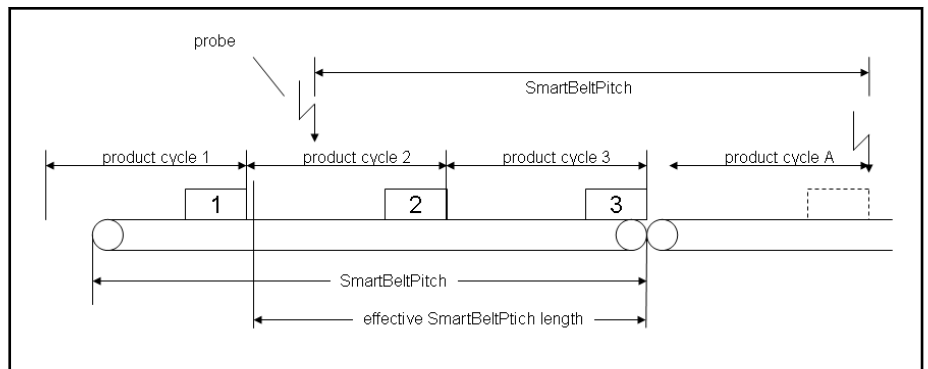


Fig. 4-62: SmartBelt with touch probe not mounted at the beginning of the belt

To ensure this, the physical length may not be specified but the effective length of the SmartBelt has to be specified. This corresponds to the distance from the end of the belt to the touch probe + one product length. This distance is also drawn in the figure. The difference between the effective length and the physical length is the area at the beginning of the SmartBelt in which the product is on the belt but not yet detected by the touch probe. Thus, it is not corrected in this area and is thus irrelevant for the function block.



To calculate the "ProbeSetpoint" in the following, the physical length of the SmartBelt and not the effective length has to be used!

Calculating the "ProbeSetpoint"

The "ProbeSetpoint" input identifies the desired master position at the moment the product passes the touch probe of a SmartBelt. This point can be calculated by using the distance between the touch probe location and the end of all SmartBelts, i.e. where the product has to be aligned. This is illustrated in the following figure:

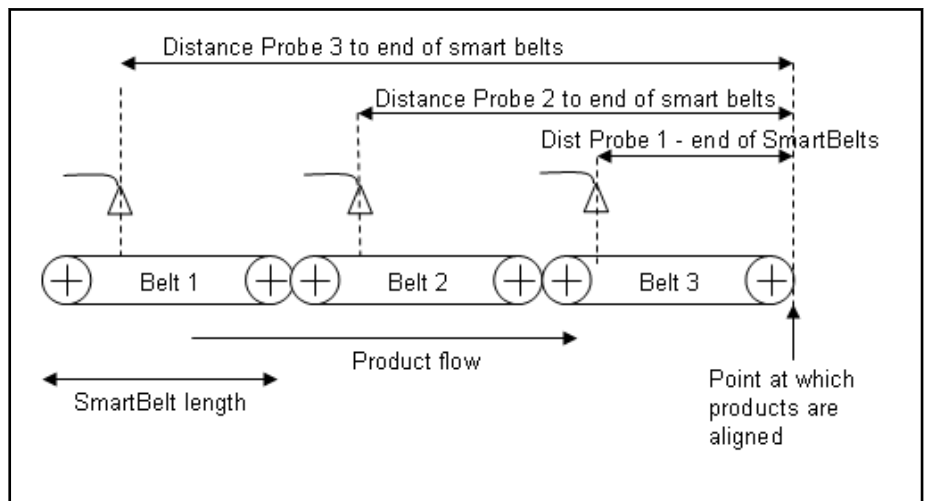


Fig. 4-63: Measurements to calculate "ProbeSetpoint"

First, measure all the distances from the touch probes to the end of all SmartBelts, i.e. where the products are aligned. The following formula can be used to calculate the value for the **positive edge detection** on the touch probe input:

$$\text{ProbeSetpoint}_k = L_{\text{cycl}} - \left(\left(L_{\text{SmartBelt}_n} - L_{\text{ProbePos}_n} \right) + \sum_{n=1}^{k-1} L_{\text{SmartBelt}_n} \right) \text{MOD} (L_{\text{cycl}})$$

Fig. 4-64: Formula for the positive edge detection

The following formula applies to the **negative edge detection** of the touch probe input:

$$\text{ProbeSetpoint}_k = L_{\text{cycl}} - \left(\left(L_{\text{SmartBelt}_n} - L_{\text{ProbePos}_n} - L_{\text{Prod}} \right) + \sum_{n=1}^{k-1} L_{\text{SmartBelt}_n} \right) \text{MOD} (L_{\text{cycl}})$$

Fig. 4-65: Formula for the negative edge detection

The following variables are used in the calculations:

- L_{cycl} : Length of the product cycle (length of product (L_{prod}) plus length of gap)
- k : number of SmartBelts used
- L_{ProbePos} : touch probe position at a specified SmartBelt (measured from the beginning of the respective SmartBelt)
- $L_{\text{SmartBelt}}$: Length of the SmartBelt

Assuming that three SmartBelts are used, each 150mm long, where the touch probe is mounted 30mm from the beginning of the SmartBelt, the product cycle is 120mm and the positive edge of the product is detected. The following "ProbeSetpoint" values can be calculated as follows:

- 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 limitations and behavior

Limitations

The function block is intended for phase corrections of products on one belt. The products have to be located on the belt with a gap in between to identify each product properly. If the gap is too narrow, the second product is not detected that leads to a false correction.

The limit values for acceleration and/or maximum and minimum distances have to be set to values that can be reached. Otherwise, the products start to slide on the belt. This results in incorrect corrections.

Ideally, only one product should be on the belt at any time. If two or more products are on one belt, the second one is not corrected until the first one leaves the belt. The remaining distance for the correction is thus lower. In addition, the second product participates in the correction of the first product. Both values are taken into account when calculating the correction amount for the second product. However, it is not possible that the second product is corrected in the same way as it would if alone on a belt.

Two or more products on one belt

When two products are on a belt, the second product is not corrected, until the first one is off the belt. In this case, "off the belt" means the first product has traveled one "SmartBeltPitch" from the registration point. In the following figure, the dotted product is considered "off the belt" since it has traveled one whole "SmartBeltPitch" since its registration on belt 1 (looking at the leading edge of the product). This value is specified.

The second product has less time/distance to make its corrections. The remaining distance consists of the "DistanceForCorrection" minus the already traveled distance. Having this value, all variables (velocity, acceleration and time) can be calculated for the second product.

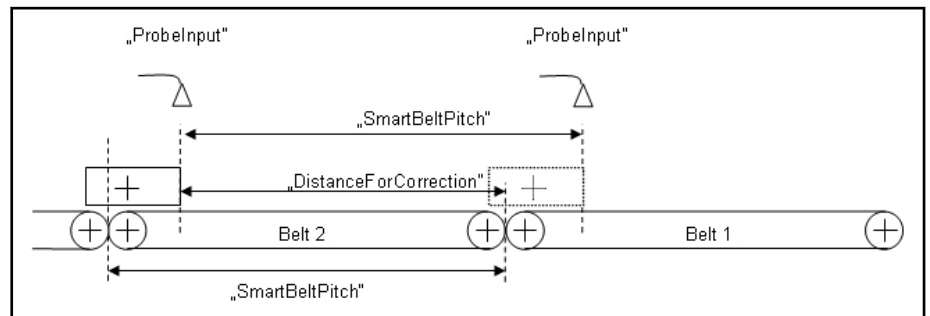


Fig. 4-66: Two or more products on one belt

An additional problem is that the second product has to be always corrected backwards. The correction is calculated based on the settings in "PercentForwardLimit", "MaxCorrectionDistancePos" and "MaxCorrectionDistanceNeg". Now, if the second product is on the belt as the first product leaves the belt, the algorithm calculates a new correction value for the second product. This could push the second product forward. However, this is not possible since the first product is there. In this case, a complete product cycle is subtracted in order to move the product to the next empty spot. Any backward motion is limited to "MaxCorrectionDistanceNeg".

Multiple products per master cycle

If multiple or fractional products correspond to one master cycle, additional settings have to be made. The gear ratio defines the relationship between the cycles.

For example, if one master revolution corresponds to three products, the inputs "RatioNumerator"/"InputRevolution" have to be set to 1 and "RatioDenominator"/"OutputRevolution" have to be set to 3. (Refer to [PLC programming, page 242](#)).



It is important to note that the "CamShaftDistance" input cannot be used to change the ratio between the master axis and the SmartBelt. If "CamShaftDistance" is not set to the product cycle, false corrections result!

4.6.6 Tips and tricks

Correction distance

The previous sections in this chapter provided details on the configuration of the SmartBelt functionality. This section provides additional information, so dependencies can be seen and improvements to an already running system can be made.

The SmartBelt uses the data of the input "DistanceForCorrection" to calculate the distance for correcting the gap between products. This is normally the distance between the touch probe position and the belt end. This is shown for product A in the figure below. This is safer, since the product can only be corrected as long as it is completely on the belt.

Product B has a longer distance for the correction. The center of mass is at the same position as the center of the SmartBelt axis. In this case, the chances of slippage are still low.

At product C is the center of mass between the two adjacent SmartBelts. In this case, slippage already occurs.

The value of "DistanceToCorrect" can be increased from case A to C in order to increase the amount of correction while monitoring slippage. It depends on the application, mainly friction and diameter of the SmartBelt axis, how much the value of "DistanceToCorrect" can be increased. Using this information, the amount of correction can be optimized.

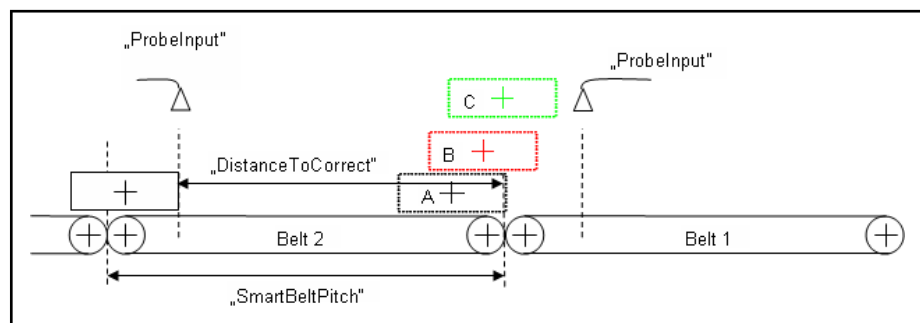


Fig. 4-67: Different distances for correction

Slippage

Slippage occurs when the product cannot accelerate as quickly as the SmartBelt. In this case, an error in the amount of correction necessary is caused. The SmartBelt function block requires the maximum acceleration as an input to determine how much correction is possible and to limit the acceleration accordingly. However, often the possible acceleration is not known. The basic interrelationship between possible acceleration and friction is given in [Friction, page 237](#), and allows for an estimation of a possible value.

A method does exist to determine if slippage occurs. Each SmartBelt is provided with the output "CorrectionValue" that indicates the correction amount of the SmartBelt. It also has output "ProbePosition" that identifies the measured position of the product. Now, the SmartBelt should be configured in a way that the correction is done in a forward direction ("PercentForwardLimit" set equal to 100% and "MaxCorrectionDistPos" equal to the SmartBelt length).

Afterwards, multiple products are feed to the SmartBelt in such a way that their position has the maximum offset compared to the master. In this case, the output "AccelLimited" is set to TRUE to indicate that the maximum acceleration had been used (if the output is not set to TRUE, the input "MaxAcceleration" for the maximum acceleration has to be reduced until the output becomes TRUE).

Once the maximum acceleration had been reached, output "ProbePosition" should be noted as well as "CorrectionValue". Even the output "ProbePosition" of the SmartBelt following the first SmartBelt should be noted. Now, the simple rule is that the "ProbePosition" value of the first SmartBelt minus "CorrectionValue" of the first SmartBelt should match the value of the "ProbePosition" value of the second SmartBelt. If the values differ significantly, the as-

sumption is that slippage occurred. In this case, the input "MaxAcceleration" has to be reduced until there is no slippage anymore.

In case the values match, the acceleration can even be increased to get more correction capability. This can lead up to a point, where the maximum acceleration is never reached. In other words: The complete correction can be done on one belt.

4.6.7 Common mistakes and misunderstandings

Question	Answer:
My drive does not synchronize or takes very long to synchronize	When the drive is switched to linear mode, the units change, but the default values are kept the same. The acceleration and velocity for synchronizing are usually very small, slowing synchronization down. Increase the values for A-0-2790 and A-0-2791 (MLC) or P-0-0142 and P-0-0143 (MLD)
Error F2002028 (MLC) or F2028 (MLD)"Excessive control deviation" occurs permanently.	Set the parameters A-0-0032, A-0-0033 and A-0-0034 (MLC) or S-0-0038, S-0-0039 and S-0-0138 (MLD) to proper values. The default values for those parameters fit rotary scaling, but with linear scaling, they are usually too small and need adjustment
The function block is configured to correct 100%. However, the product is not in the correct position after coming from the belt.	There can be multiple reasons: First, the phase difference might have been too large to be corrected in one belt length. In this case, the outputs "NegCorrLimited", "PosCorrLimited" or "AccelLimited" are set to TRUE. Second, if the values "MaxCorrectionDistancePos", "MaxCorrectionDistanceNeg" or "MaxAcceleration" are set too high, the products might not be able to physically follow the belt and start sliding. This condition is not indicated by the status outputs. Measures can be taken to increase the friction (vacuum belts) or more belts have to be used to compensate for the errors on multiple belts. Third, if two or more products are on the belt, refer to the next question
When there are two or more products on the belt, the second or third product is not corrected properly	The second product has to wait until the first product has left the belt. Subsequently, it is corrected. If the remaining distance is too small, the required correction cannot be achieved
If two products are located end-to-end on the belt, the second product is not corrected	If the products are located end-to-end, the touch probe function cannot detect the second product. The SmartBelt only sees one product. It has to be ensured that the process preceding the SmartBelts separates the products with a minimum distance which enables the touch probe functionality to detect the next product
Everything is configured, but the Smart-Belt does not perform any correction and no error is present	Set bit 9 of parameter P-0-0226 in the drive and switch to phase 2 and then to phase 4 again. This automatically enables the touch probe functionality. Otherwise, the touch probe functionality is configured but not enabled. This leads to no touch probe measurements and hence, the SmartBelt does not recognize any product

4.7 Cam tables for technology functions

4.7.1 Introduction and overview

The function blocks create cam tables. The function block [MB_PilgrimStepCalcType01](#), [page 250](#), calculates a cam table with up to 1,024 data points, the distances of the forward and backward transport as well as stop positions for the pilgrim step operation.

The function block [MB_PrintLengthCorrCalcType01](#), [page 260](#), calculates a cam table with up to 1,024 data points for the technology function "print

length correction" to change the length of a print on the printing material (web or shed).

4.7.2 MB_PilgrimStepCalcType01

Brief description The function block calculates a cam table with up to 1,024 data point elements as well as the distances of the forward and backward transport as well as stop positions for the step-back operation. It considers the possibility that the synchronous range of the printing cylinder can be bigger than the format length to be printed. In case of offset print works, the plate length for example is specified due to the mechanical fixture and therefore always constant. The printing format on the printing plate can be shorter.

The cam calculated by the function block operates the infeed roll and the out-feed roll of the printing press.

Assignment target system/library

MLC/XLC 14VRS	ML_TechMotion
Function package TEC	

Tab. 4-42: Reference table of the MB_PilgrimStepCalcType01

Interface description

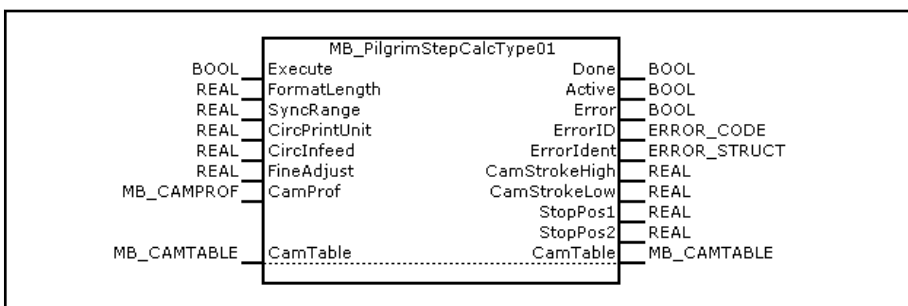


Fig. 4-68: MB_PilgrimStepCalcType01 function block

I/O type	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Function block processing enabled (once, edge-controlled)
	FormatLength	REAL	Length of the format to be printed [mm]
	SyncRange	REAL	Length of the synchronous range [mm]
	CircPrintUnit	REAL	Circumference of the printing cylinder [mm]
	CircInfeed	REAL	Circumference of the draw rollers [mm]
	FineAdjust	REAL	Fine adjustment [%] – defines the desired path tension
	CamProf	MB_CAMPROF	Cam profile selection MB_CAMPROF , page 97
VAR_OUTPUT	Done	BOOL	Processing completed without errors, output data valid
	Active	BOOL	Processing not yet completed, output data invalid
	Error	BOOL	Processing completed with error, output data invalid, error outputs valid
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, it contains a detailed error description (see Troubleshooting , page 260)

I/O type	Name	Type	Comment
	CamStrokeHigh	REAL	Stroke P-0-0073 outfeed / P-0-0093 infeed P-0-0093 is specified via the function block MC_CamIn / input "CamShaftDistance" P-0-0073 is specified via the function block MB_WriteRealParameter
	CamStrokeLow	REAL	Stroke P-0-0073 infeed / P-0-0093 outfeed P-0-0093 is specified via the function block MC_CamIn / input "CamShaftDistance" P-0-0073 is specified via the function block MB_WriteRealParameter Acceleration used related to the input "MaxAcc" in [%]
	StopPos1	REAL	Point of reversal 1 [°] with velocity = 0
	StopPos2	REAL	Point of reversal 2 [°] with velocity = 0
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements MB_CAMTABLE, page 97

Tab. 4-43: Interface variables of the MB_PilgrimStepCalcType01 function block

Min./max. and default values

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
FormatLength	REAL	>0.0	n.def.	100.0	Rising edge at "Execute"
SyncRange	REAL	>0.0	n.def.	100.0	Rising edge at "Execute"
CircPrintUnit	REAL	>0.0	n.def.	100.0	Rising edge at "Execute"
CircInfeed	REAL	>0.0	n.def.	100.0	Rising edge at "Execute"
FineAdjust	REAL	0.0	100.0	1.0	Rising edge at "Execute"
CamProf	MB_CAMPROF			CAM_PROF_STANDARD	Rising edge at "Execute"

Tab. 4-44: Min./max. and default values of the MB_PilgrimStepCalcType01 function block

The enumeration MB_CAMPROF, ML_TechMotionBase, selects between different cam profiles.



When selecting the cam profile "CAMPROF_STANDARD", the linear path has to be disabled since it is a profile without gear reduction. The linear path for the profiles "CAMPROF_ADDITIVE" and "CAMPROF_ADDITIVE_SEGMENTABLE" has to be ticked and the reduction has to be set to "1", since they are profiles with gear reduction.

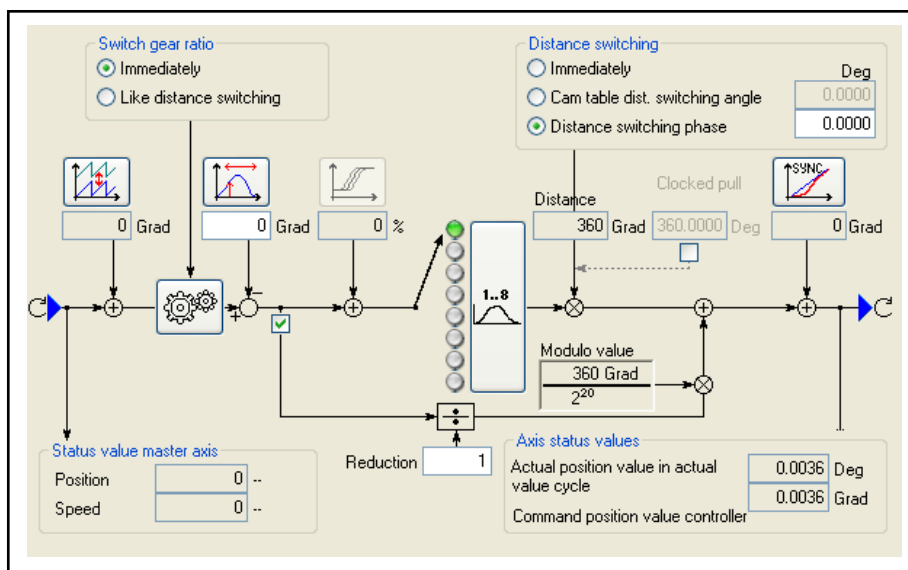


Fig. 4-69: Enabling the linear path

The structure MB_CAMTABLE, ML_TechMotionBase, is provided with the "CamTable" array containing the calculated cam points and the "NumberOfElements" variable which shows the number of valid cam points of the "CamTable" array.

Functional description

If not the complete circumference of a printing cylinder is covered with a printing plate or printing block, printed and non-printed parts alternate in continuous operation mode. To avoid these printing gaps, the web is decelerated by feed rolls, retracted by a defined distance and accelerated again to synchronous velocity so that the new print immediately follows the old one.

The compensation motion calculated by the function block is characterized by a continuity up to the acceleration.

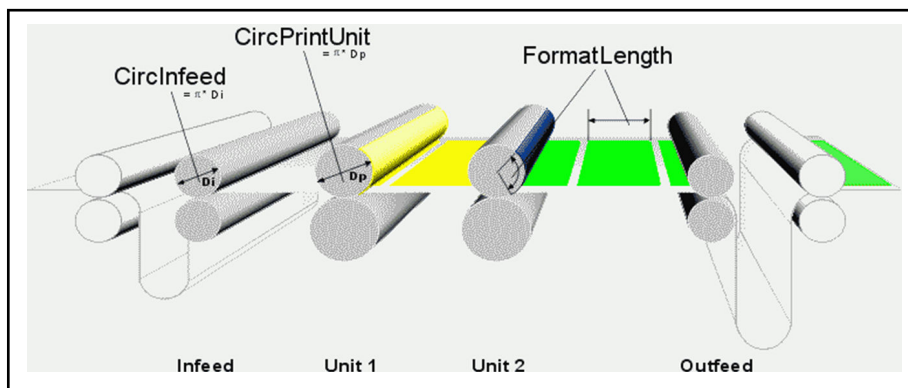


Fig. 4-70: Schematic diagram of the step-back

Prerequisites to use the function block

- The function block can only be used for axes with interpolation in the drive
- The drive function "Clocked pull roll" (see drive documentation P-0-0086) has to be switched on by the user

The MB_PilgrimStepCalcType02 calculates a cam table with up to 1,024 data point elements and both the distance factors "CamStrokeHigh" and "CamStrokeLow" and the stop positions "StopPos1" and "StopPos2" only once after the function block was enabled for processing using "Execute". The func-

tion block uses the mechanical specification of the circumference of the infeed or outfeed roll "CircInfeed", the circumference of the printing cylinder "CircPrintUnit", the printing plate length "FormatLength" on the printing cylinder and the total length of the synchronous range "SyncRange". The number of data point elements of a cam table is set by the "NumberOfElements" element of the "CamTable" structure.

It is assumed that the circumference of the infeed roll and the circumference of the outfeed roll are identical (CircInfeed).

According to the advancing specified by the fine adjustment "FineAdjust", two distances are calculated to keep the web tension constant. The higher distance is for the roll located in the front with regard to the transportation direction. To calculate the distances, the function block uses the following formulae:

$$CamStrokeLow = 360^\circ * \frac{FormatLength}{CircInfeed}$$

The "CamStrokeLow" values applies to the parameter P-0-0073 of the infeed roll and for the parameter P-0-0093 of the outfeed roll

Fig. 4-71: *CamStrokeLow formula*

$$CamStrokeHigh = 360^\circ * \frac{FormatLength}{CircInfeed} * \left(1 + \frac{FineAdjust}{100} \right)$$

The "CamStrokeHigh" values applies to the parameter P-0-0073 of the outfeed roll and for the parameter P-0-0093 of the infeed roll

Fig. 4-72: *CamStrokeHigh formula*

The outputs "StopPos1" and "StopPos2" contain master axis positions where cams are provided with a point of reversal to change the format at one of these positions during a control or to change the distance with the register controller.

The cam table to be calculated is stored in the "CamTable" structure with scaling in per cent. The validity of the cam elements is signaled by the "Done" output. The cam table can be loaded to the control system or drive using the MB_LoadCamData, ML_TechMotionBase, function block.

If an error occurs while processing the function block, this is signaled by the "Error" output. The elements of the "CamTable" structure as well as the outputs "CamStrokeHigh" and "CamStrokeLow", "StopPos1" and "StopPos2" are not updated in case of error.

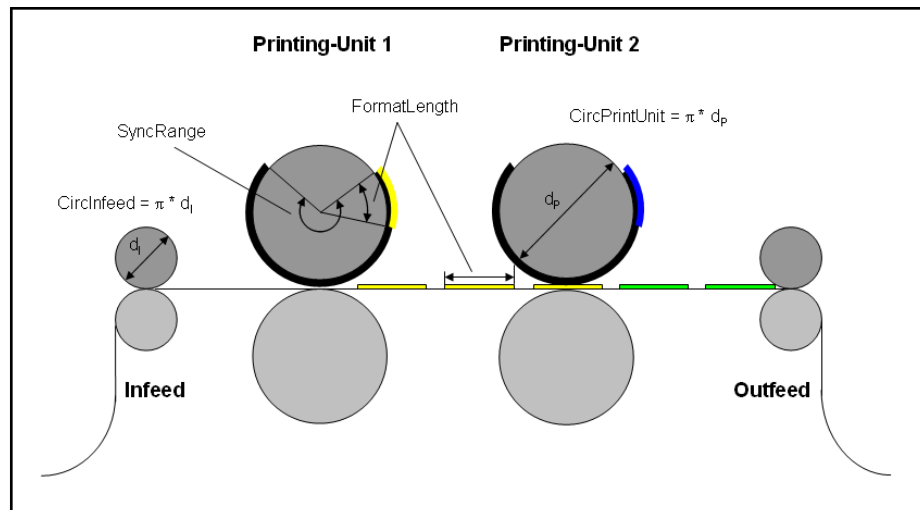


Fig. 4-73: Process diagram ($\text{FormatLength} < \text{SyncRange}$)

Cam characteristics

The cam table of the MB_PilgrimStepCalcTyp01 function block provides the following characteristics:

- The function block is based on the MSV_CamTableType04 function block and also applies to those cases with a synchronous range of the printing cylinder bigger than the printing range
- The compensation motion for the back-transport of the web characterized by the cam table is constant up to the acceleration (s , v , a) and shows a jump in the jerk (j)
- The compensation motion begins with the first table element, then, a linear inclination which is phase-synchronous to the printing cylinder
- The algorithm to calculate the cam data points is divided into several cycles, i.e. calls the instance to avoid a considerable effect on the PLC cycle time
- The calculated distances of the function block provide the following characteristics:
 - The distances depend on the printing format and the circumference of the infeed roll
 - The distance "CamStrokeHigh" is provided for the parameter P-0-0093 of the infeed roll and the parameter P-0-0073 of the outfeed roll, the distance "CamStrokeLow" is provided for the parameter P-0-0073 of the infeed roll and the parameter P-0-0093 of the outfeed roll

To generate the web tension, the distance for the forward transport of the outfeed roll and the distance for the back-transport of the infeed roll exceed the distance for the forward transport of the infeed roll and the distance for the back-transport of the outfeed roll by the fine adjustment.

There are three different types of cams:

- **FormatLength = SyncRange = CircPrintUnit**
The printing plate or printing block covers the complete synchronous zone of the cylinder and the synchronous zone covers the complete circumference of the printing cylinder. In this case, there is no compensation motion. If "SyncRange" is equal to "CircPrintUnit", "FormatLength" has to be provided with the same value, since no compensation motion

can be executed. If "FormatLength" is lower than "SyncRange", the function block cancels the processing outputting an error.

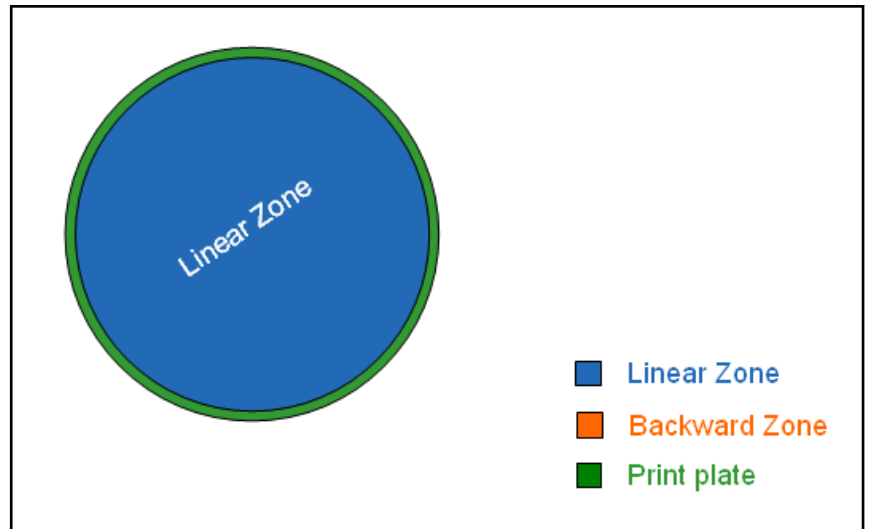
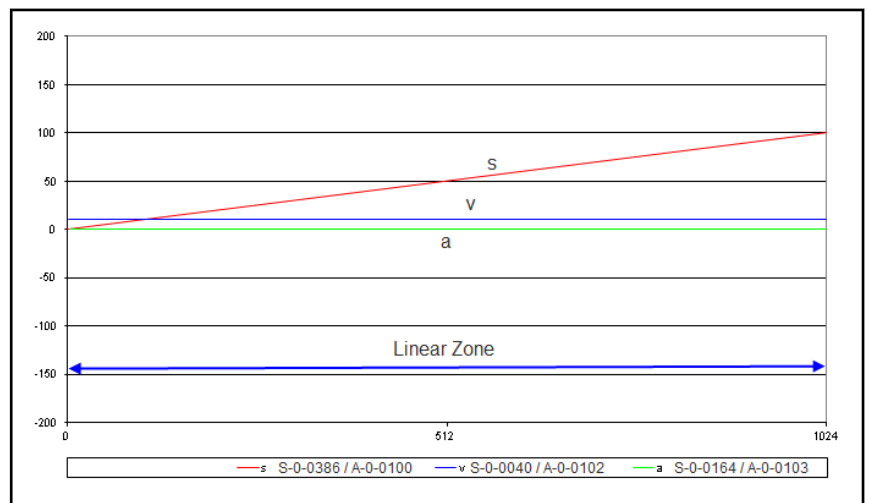


Fig. 4-74: *MB_PilgrimStepCalcType01* with *FormatLength* = *SyncRange* = *CircPrintUnit*



FormatLength = 2000.0; *SyncRange* = 2000.0; *CircPrintUnit* = 2000.0; *CircInFeed* = 2000.0

Fig. 4-75: *MB_PilgrimStepCalcType01* MotionProfile

- **FormatLength = SyncRange**

The printing plate or printing block covers the complete synchronous zone of the cylinder. The complete remaining cylinder area can be used for compensation motion.

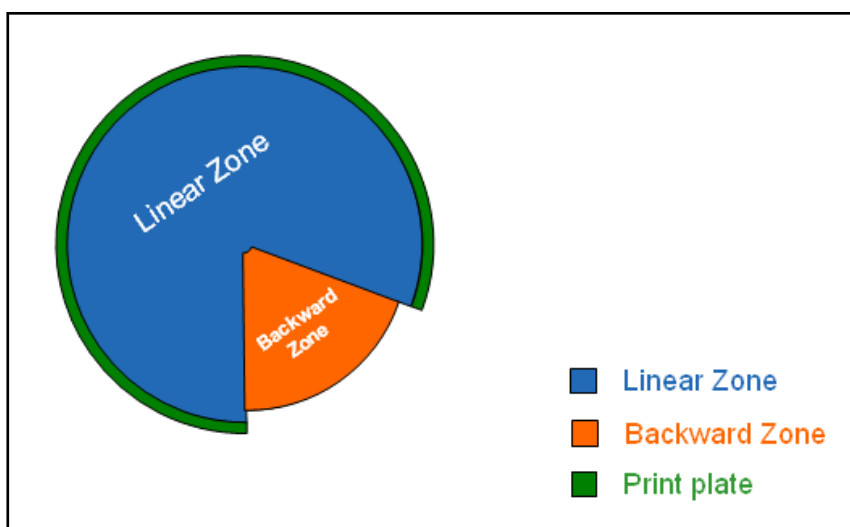
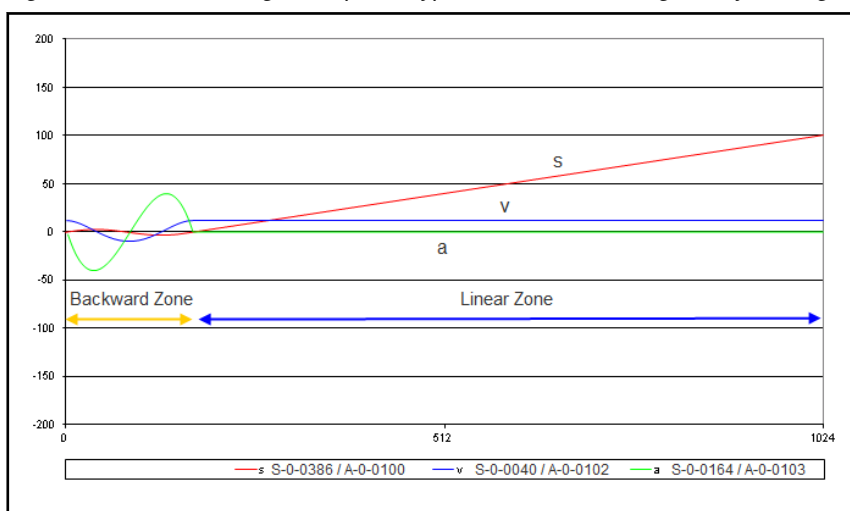
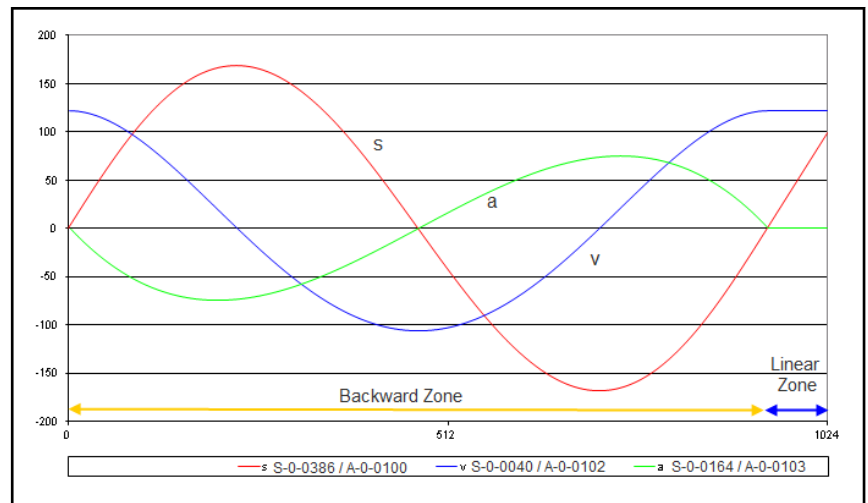


Fig. 4-76: *MB_PilgrimStepCalcType01* with *FormatLength* = *SyncRange*



FormatLength = 500.0; *SyncRange* = 500.0; *CircPrintUnit* = 600.0; *CircInFeed* = 600.0

Fig. 4-77: *MB_PilgrimStepCalcType01* MotionProfile



FormatLength = 200.0; SyncRange = 200.0; CircPrintUnit = 2500.0; CircInFeed = 2500.0

Fig. 4-78: MB_PilgrimStepCalcType01 MotionProfile

- **FormatLength < SyncRange**

The printing plate or printing block only partially covers the synchronous zone of the cylinder. This has to move up to the end of the synchronous range with synchronous velocity. Only if the cylinder has no contact with the web anymore, the material web can be decelerated, retracted and again accelerated. Here, the additional distance of the non-printing synchronous range has to be retracted.

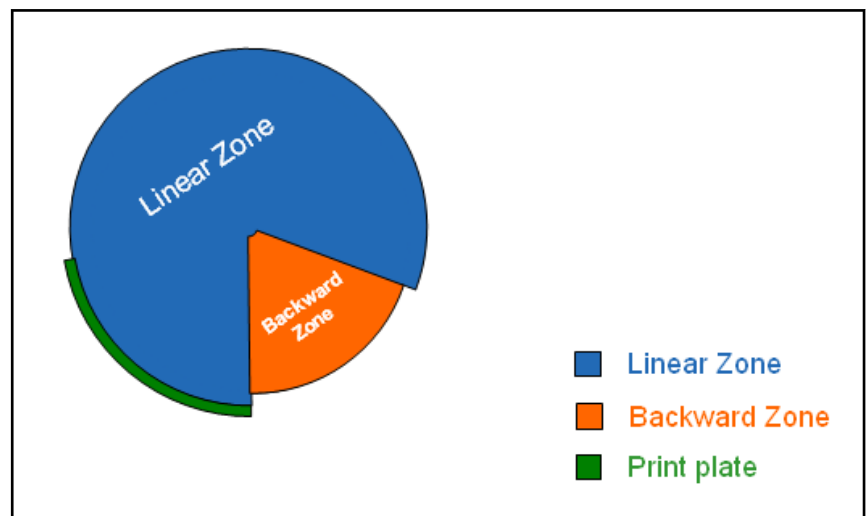
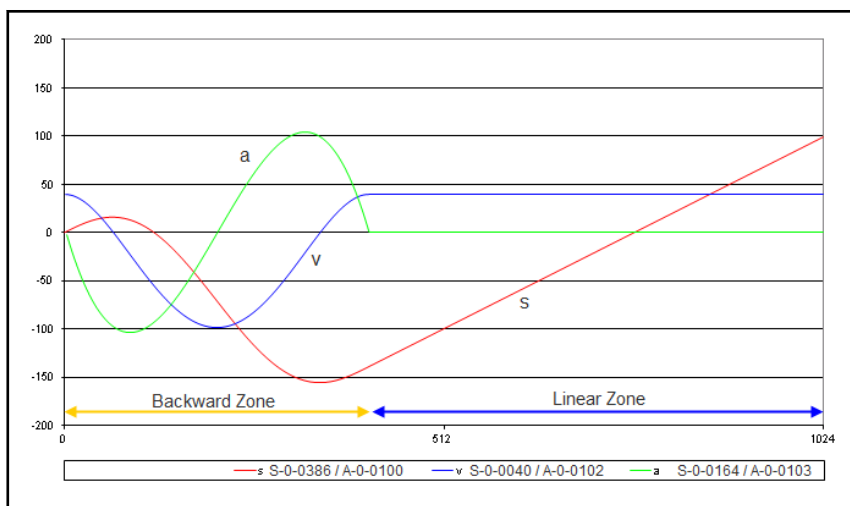
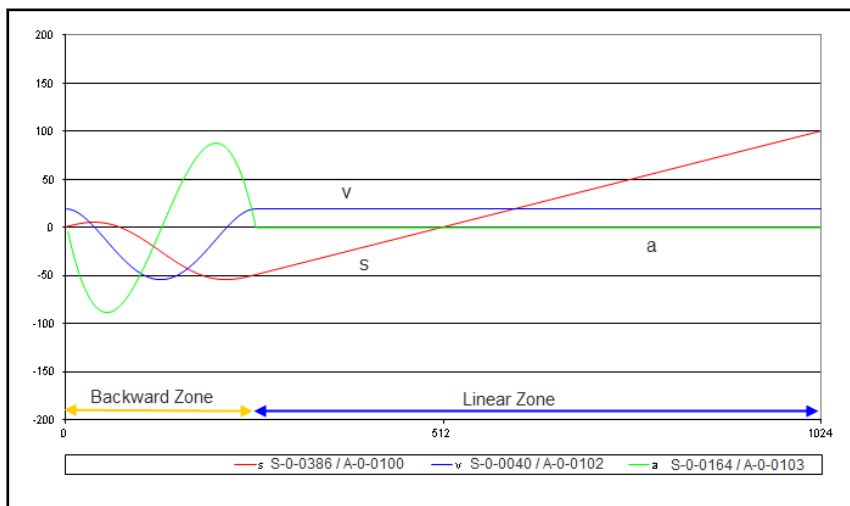


Fig. 4-79: MB_PilgrimStepCalcType01 with FormatLength < SyncRange



FormatLength = 500.0; SyncRange = 1200.0; CircPrintUnit = 2000.0; CircInFeed = 3000.0

Fig. 4-80: MB_PilgrimStepCalcType01 MotionProfile

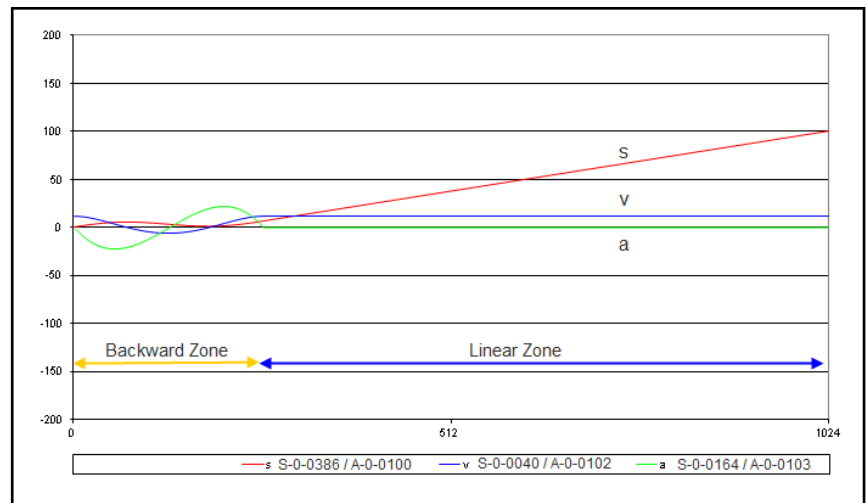


FormatLength = 1000.0; SyncRange = 1500.0; CircPrintUnit = 2000.0; CircInFeed = 3000.0

Fig. 4-81: MB_PilgrimStepCalcType01 MotionProfile

- **FormatLength > SyncRange**

For special productions, the format length may also be larger than the SyncRange of a printing cylinder.



FormatLength = ; SyncRange = ; CircPrintUnit = ; CircIn-Feed =

Fig. 4-82: MB_PilgrimStepCalcType01 MotionProfile

Use case special product printed with two printing units:



Fig. 4-83: Standard product (format length 350 mm):



Fig. 4-84: Special product (format length 700 mm):

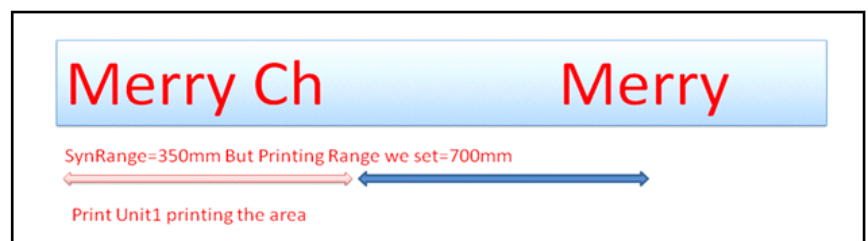


Fig. 4-85: Printing unit 1 for printing the red area



Fig. 4-86: Printing unit 2 for printing the blue area

Troubleshooting The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR	16#2200	16#0001	"NumberOfElements" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#2260	16#0001	"FormatLength" input is < minimum
INPUT_RANGE_ERROR	16#2260	16#0002	"SyncRange" input is < minimum
INPUT_RANGE_ERROR	16#2260	16#0003	"CircPrintUnit" input is < minimum
INPUT_RANGE_ERROR	16#2260	16#0004	"CircInfeed" input is < minimum
INPUT_RANGE_ERROR	16#2260	16#0005	"FineAdjust" input is < minimum or > maximum
INPUT_INVALID_ERROR	16#2261	16#0001	Address of the "CamTable" array is zero
INPUT_INVALID_ERROR	16#2261	16#0001	"FormatLength" input is < minimum
INPUT_INVALID_ERROR	16#2261	16#0002	"SyncRange" input is < minimum

Tab. 4-45: Error codes FB MB_PilgrimStepCalcType01

4.7.3 MB_PrintLengthCorrCalcType01

Brief description The function block calculates a cam table with up to 1,024 data point elements for the technology function printing length correction.

Assignment: Target system/library

MLC 14VRS	ML_TechMotion
TEC function package	

Tab. 4-46: Reference table of the MB_PrintLengthCorrCalcType01

Interface description

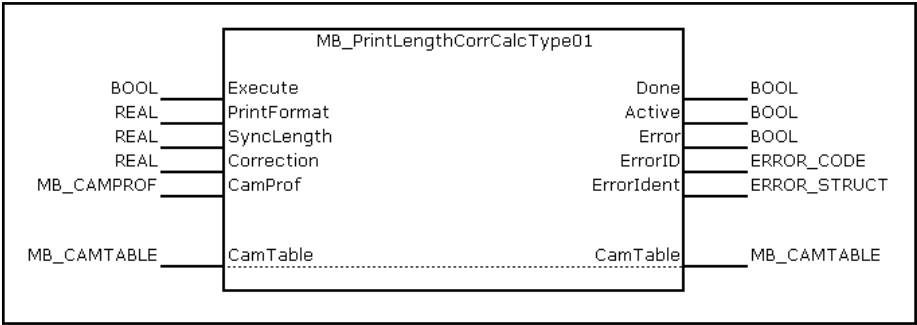


Fig. 4-87: MB_PrintLengthCorrCalcType01 function block

I/O type	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Processing of the function block enabled (once, edge-controlled)
	PrintFormat	REAL	Expected nominal printing format [mm]
	SyncLength	REAL	Synchronous format length, i.e. circumference of the printing cylinder [mm]
	Correction	REAL	Printing correction to compensate the deviation [mm]
	CamProf	MB_CAMPROF	Cam profile selection MB_CAMPROF, page 97
VAR_OUTPUT	Done	BOOL	Processing completed without errors, output data is valid
	Active	BOOL	Processing not yet completed, output data invalid
	Error	BOOL	Processing completed with error, output data invalid, error outputs valid
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, it contains a detailed error description (see Error handling, page 265)
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements MB_CAMTABLE, page 97

Tab. 4-47: Interface variables of the MB_PrintLengthCorrCalcType01 function block

Min./max. and default values

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
PrintFormat	REAL	>0.0	n.def.	1000.0	Rising edge at "Execute"
SyncLength	REAL	>0.0	n.def.	1500.0	Rising edge at "Execute"
Correction	REAL	-9999.9	9999.9	100.0	Rising edge at "Execute"
CamProf	MB_CAMPROF			CAM_PROF_STANDARD	Rising edge at "Execute"

Tab. 4-48: Min./max. and default values of the MB_PrintLengthCorrCalcType01

The enumeration MB_CAMPROF, ML_TechMotionBase selects between different cam profiles.



When selecting the cam profile "CAMPROF_STANDARD", the linear path has to be disabled since it is a profile without gear reduction. The linear path for the profiles "CAMPROF_ADDITIVE" and "CAMPROF_ADDITIVE_SEGMENTABLE" has to be ticked and the reduction has to be set to "1", since they are profiles with gear reduction.

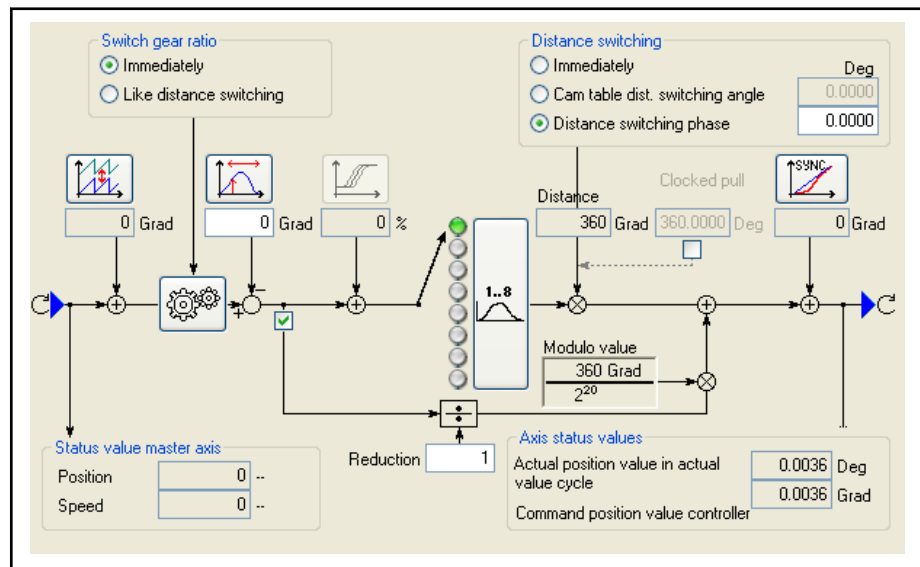


Fig. 4-88: Enabling the linear path

The structure MB_CAMTABLE, ML_TechMotionBase is provided with the "CamTable" array containing the calculated cam points and the variable "NumberOfElements" which shows the number of valid cam points of the "CamTable" array.

Functional description

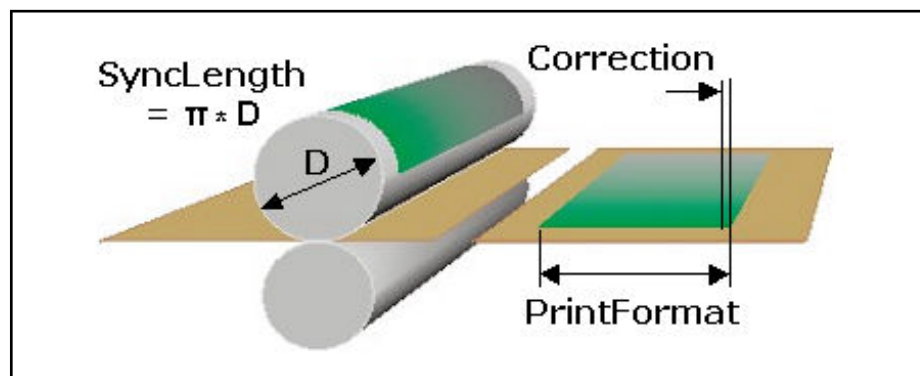


Fig. 4-89: Schematic diagram of the printing length correction

The printing length correction changes the length of a printing on the printing material (material web or shed). Therefore, the velocity of the printing roll to the printing material is increased or decreased during the printing contact of the printing material. In the following printing-free area, the position of the printing cylinder is corrected in a way that the next print can be started again in register.

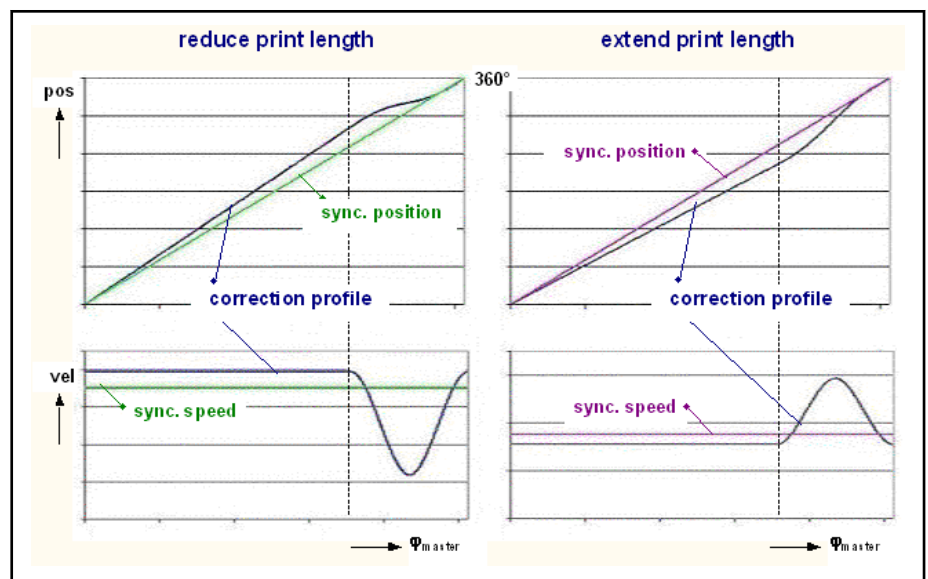


Fig. 4-90: Profile of the printing length correction

With the specification of the formats "PrintFormat" and "SyncLength" and the correction "Correction", the "MB_PrintLengthCorrCalcType01" function block non-recurrently calculates a cam table with 1,024 data point elements after enabling the processing with "Execute" based on the input values. The number of data point elements of a cam table is set by the "NumberOfElements" element of the "CamTable" structure.

The sum of printing format and printing correction may not exceed the circumference of the printing roll. The necessary positive and negative acceleration values increase with regard to the decreasing print-free zone.

- A positive correction value increases the distance traveled by the printing cylinder with the same master axis position and thus reduces the printing format

The cam table to be calculated is stored in the "CamTable" structure with scaling in per cent. The validity of the cam elements is signaled by the "Done" output. The cam table can be loaded to the control or drive using the MB_LoadCamData, ML_TechMotionBase function block.

If an error occurs while processing the function block, this is signaled by the "Error" output. The elements of the "CamTable" structure are not updated in case of error.

Cam characteristics

The cam table of the ML_PrintLengthCorrCalcType01 provides the following characteristics:

- The MotionProfile characterized by the cam table is constant up to the acceleration (s,v,a) and shows a jump in the jerk (j)
- The correction of the printing length, i.e. the lengthening or shortening of the printing length format is based on an increase or reduction of the velocity of the printing cylinder during printing and the concluding compensation motion
- The printing format, i.e. the printing plate, may only correspond to a part of the synchronous format length, i.e. between two prints, there has to be a print-free zone which is used for the compensation motion

- The algorithm to calculate the cam data points is divided into several cycles, i.e. calls the function block instance to avoid a considerable effect on the PLC cycle time

MotionProfiles with different assignments of the "Correction" input

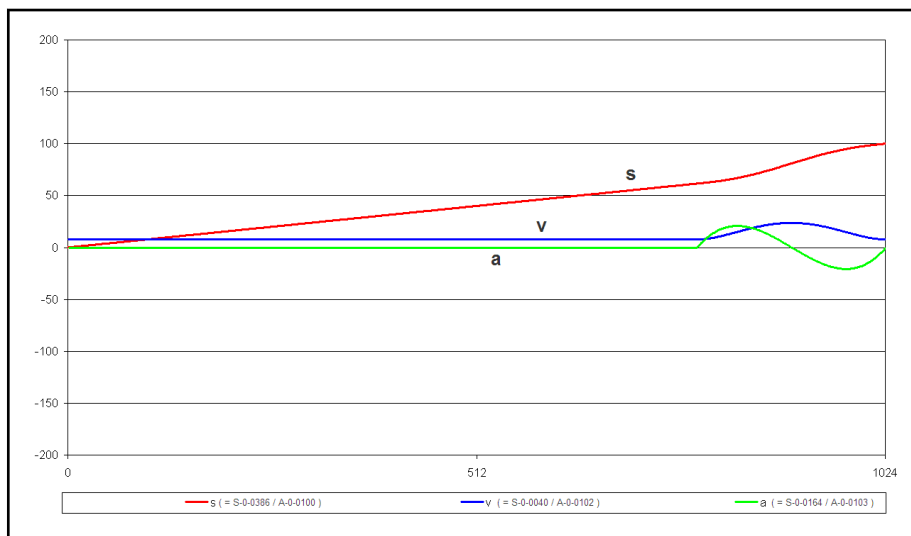


Fig. 4-91:

MB_PrintLengthCorrCalcType01 MotionProfile

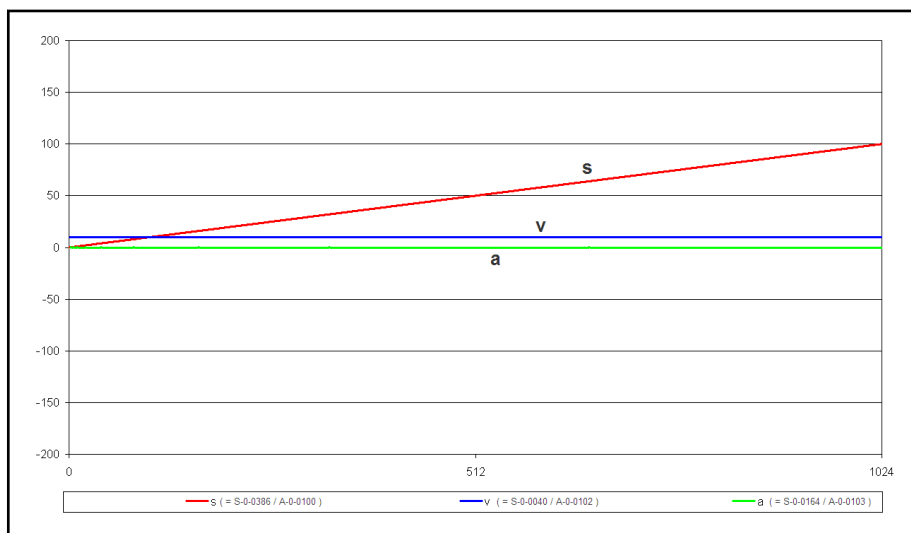
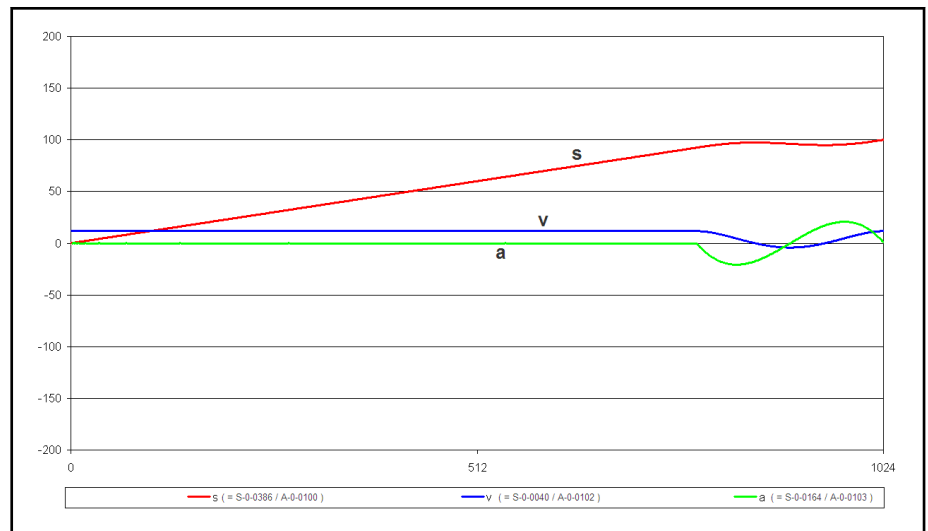


Fig. 4-92:

MB_PrintLengthCorrCalcType01 MotionProfile



SyncLength: 1.3 * PrintLength; Correction: 200

Fig. 4-93: MB_PrintLengthCorrCalcType01 MotionProfile

Error handling The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2.

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR	16#2200	16#0001	"NumberOfElements" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#2240	16#0001	"PrintFormat" input is < minimum
INPUT_RANGE_ERROR	16#2240	16#0002	"SyncLength" input is < minimum
INPUT_RANGE_ERROR	16#2240	16#0003	"Correction" input is < minimum or > maximum
INPUT_INVALID_ERROR	16#2201	16#0001	Address of the "CamTable" array is zero
INPUT_INVALID_ERROR	16#2241	16#0001	Invalid calculation: (PrintFormat + Correction) > SyncLength or SyncLength <= PrintFormat

Tab. 4-49: Error codes of the MB_PrintLengthCorrCalcType01 function block

4.8 Tension controller

4.8.1 Introduction and overview

Depending on the measured value of a load cell, the tension controllers ([MB_TensionControlLoadCellType01](#), page 266, and [MB_TensionControlLoadCellMatParType01](#), page 279) control the tension to the command value. The actual value has to be provided to the controller and is normally measured via an analog or field bus input.

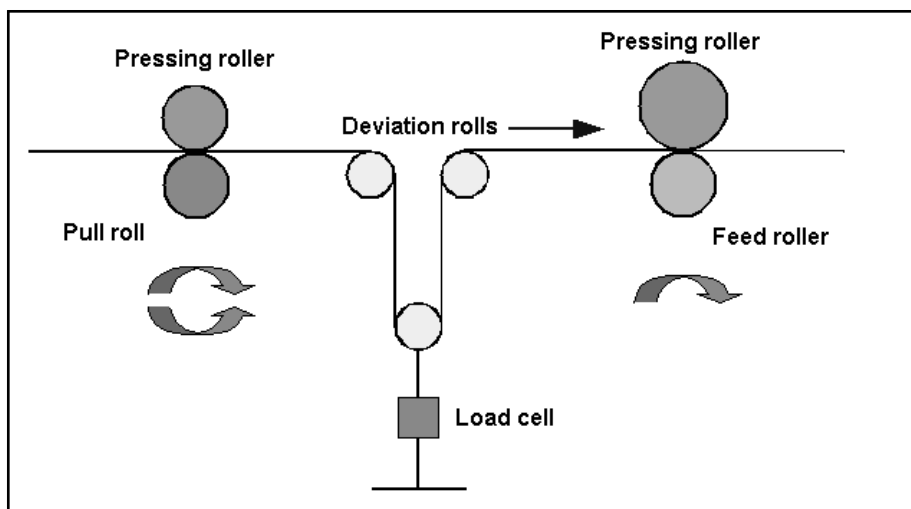


Fig. 4-94: Schematic diagram for the closed-loop tension control

The tension controller is designed as PI controller. The controller output affects the gear fine adjustment of the slave axis (draw roller).

For the standstill control, an additional P-controller has been integrated. The controller output in standstill acts on the additive master axis position of the slave axis (draw roller).

With the "MB_TensionControlLoadCellType01" tension controller, the dynamic control loop response can be parameterized by specifying the controller parameters, with the "MB_TensionControlLoadCellMatParType01" tension controller by specifying material and path parameters.

The slave axis can either be operated in the velocity-synchronous or in the phase-synchronous operation mode.

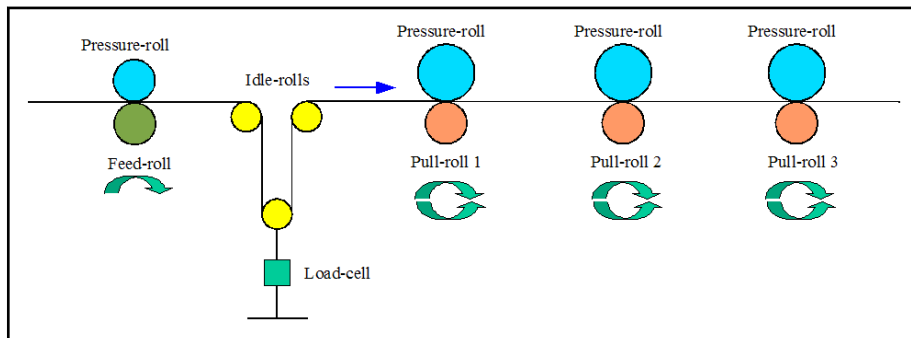


Fig. 4-95: Multi-axis tension controller

4.8.2 MB_TensionControlLoadCellType01

Brief description

Depending on the measured value of a load cell, the tension controller controls the tension to the command value.

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechMotion
MLD with MPx07/08/17 with MA function package	MX_Technology_07.lib
MLD MPx20 and higher with MA function package	MX_TechMotion

Tab. 4-50: Reference table of the MB_TensionControlLoadCellType01



MLC 12VRS or higher also supports slave axes with interpolation on the control.



The "ResponseCurve" input is declared in the RIL_LoopControl library. Therefore, the RIL_LoopControl library has to additionally be integrated in the library manager if the "ResponseCurve" input is used.

Interface description

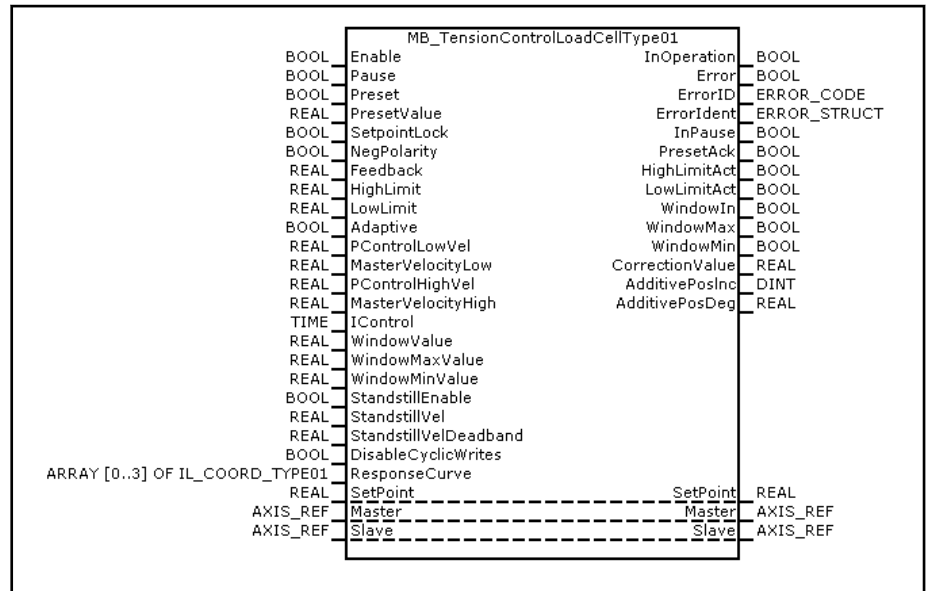


Fig. 4-96: MB_TensionControlLoadCellType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	If this input is set, the function block operates
	Pause	BOOL	If this input is set, the tension controller is paused and the I-value of the output value is frozen. The "InPause" output signalizes an active "Pause"
	Preset	BOOL	At the rising edge of this input, the control output is set to the value parameterized in "PresetValue" and, if successful, the "PresetAck" output is set
	PresetValue	REAL	The controller output is set to this value if the "Preset" input is set
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output. The value is only evaluated at a rising edge at "Enable"
	SetpointLock	BOOL	At a rising edge at the input, the current actual tension value at "Feedback" is applied as new command value for the tension controller
	NegPolarity	BOOL	If this input is set, the control variable "CorrectionValue" and "AdditivePosInc" are provided with a negative sign
	Feedback	REAL	Current actual value of the tension

I/O type	Name	Data type	Description
	HighLimit	REAL	Maximum value of the controller output. If the control variable calculated by the controller is greater than this value, the control variable is limited and the I-value of the controller is not summed up anymore
	LowLimit	REAL	Minimum value of the controller output. If the control variable calculated by the controller is lower than this value, the control variable is limited and the I-value of the controller is not summed up anymore
	Adaptive	BOOL	If this input is set, an adaptive P-gain is used for the closed-loop control
	PControlLowVel	REAL	P-gain of the controller if "Adaptive" = FALSE Otherwise, the value corresponds to the P-gain at low master axis velocities
	MasterVelocityLow	REAL	Only effective if "Adaptive" = TRUE. Master axis velocity up to which "PControlLowVel" is used
	PControlHighVel	REAL	P-gain of the controller at high master axis velocities. If "Adaptive" = FALSE, this input has no effect
	MasterVelocityHigh	REAL	Only effective if "Adaptive" = TRUE. Master axis velocity from which "PControlHighVel" is used
	IControl	TIME	Integral action time of the I-controller If the value is set to 0 s, the I-controller is not effective
	WindowValue	REAL	Process variable window
	WindowMaxValue	REAL	Maximum value of the process variable
	WindowMinValue	REAL	Minimum value of the process variable
	StandstillEnable	BOOL	If this input is set, the additive velocity specified in "StandstillVel" is effective
	StandstillVel	REAL	Additive velocity for standstill control
	StandstillVelDeadband	REAL	This value determines the controller deviation value. Below this value, the tension velocity for the standstill control is reduced
	ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	Characteristics to scale the control deviation in the "In-Velocity Control" branch IL_COORD_TYPE01, RIL_LoopControl.library
VAR_OUTPUT	InOperation	BOOL	The tension controller operates and the outputs are valid
	Error	BOOL	Indicates an error If "Enable" = FALSE, the error is cleared
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description according to error table (see Troubleshooting, page 278)

I/O type	Name	Data type	Description
	InPause	BOOL	"Pause" is active
	PresetAck	BOOL	If this output is active, the "Preset" input was analyzed and the "CorrectionValue" output applied the "Preset-Value" value
	HighLimitActive	BOOL	This output is set if the control variable calculated by the controller is greater than the value specified in "HighLimit"
	LowLimitActive	BOOL	This output is set if the control variable calculated by the controller is lower than the value specified in "Low-Limit"
	WindowIn	BOOL	If TRUE is set, the process variable is in the parameterized window
	WindowMax	BOOL	If TRUE is set, the process variable exceeded the parameterized maximum value
	WindowMin	BOOL	If TRUE is set, the process variable is below the parameterized minimum value
	CorrectionValue	REAL	Control variable of the tension controller
	AdditivePosInc	DINT	Additive master axis position in increments per Sercos cycle. It is used for the real slave axis with interpolation in the drive (target parameter: P-0-0692) [-]
	AdditivePosDeg	REAL	Additive master axis position in degrees per Sercos cycle. It is used for the slave axis with interpolation on the control system (e.g. for virtual axes or real axes with interpolation on the control. Target parameter: A-0-2600) [°/ms]
VAR_IN_OUT	Setpoint	REAL	Tension command value This value is generally set by the user but can also be changed by the function block (see "SetpointLock")
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Reference to the master axis
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Reference to the slave axis

Tab. 4-51: Interface variables of the MB_TensionControlLoadCellType01 function block

Min./max. and default values The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Preset	BOOL			FALSE	Continuous
DisableCyclic-Writes	BOOL			FALSE	Rising edge at "Enable"
SetPointLock	BOOL			FALSE	Continuous

Name	Type	Min. value	Max. value	Default value	Effective
NegPolarity	BOOL			FALSE	Continuous
Feedback	REAL	n.def.	n.def.	0.0	Continuous
PresetValue	REAL	n.def.	n.def.	0.0	Rising edge at "Preset"
HighLimit	REAL	>LowLimit	n.def.	200.0	Continuous
LowLimit	REAL	-95.0	<HighLimit	-50	Continuous
Adaptive	BOOL			FALSE	Continuous
PControlLowVel	REAL	>0.0	n.def.	1.0	Continuous
MasterVelocity-Low	REAL	>0.0	<=MasterVelocity-High	1.0	Continuous
PControlHighVel	REAL	>0.0	n.def.	1.0	Continuous
MasterVelocity-High	REAL	>=MasterVelocity-Low	n.def.	1.0	Continuous
IControl	TIME	0 s	n.def.	0 s	Continuous
WindowValue	REAL	0.0	100.0	0.0	Continuous
WindowMaxValue	REAL	>=WindowMinValue	100.0	0.0	Continuous
WindowMinValue	REAL	0.0	<=WindowMaxValue	0.0	Continuous
StandstillEnable	BOOL			FALSE	Continuous
StandstillVel	REAL	0.0	n.def.	0.0	Continuous
StandstillDead-band	REAL	0.0	n.def.	1.0	Continuous
ResponseCurve	ARRAY [0..3] OF IL_CO-ORD_TYPE01	See RIL_Loop-Control	See RIL_Loop-Control	See data structure	Rising edge at "Enable"

Tab. 4-52: Input behavior of the MB_TensionControlLoadCellType01

Functional description

The following explains the individual functions of the function block inputs in detail. In general, all outputs are continuously analyzed if the "Enable" input is set. That means that changes made to the inputs are immediately effective without toggling the "Enable" input again.

An exception is the "PresetValue" input which is only applied at a rising edge at the "Preset" input. Likewise, the "DisableCyclicWrites" input is only applied at a rising edge at the "Enable" input.

The complete mode of operation of the tension controller is shown in the following flowcharts. Depending on the "StandstillEnable" input, either the "In-Velocity" path or "Standstill Control" is effective.

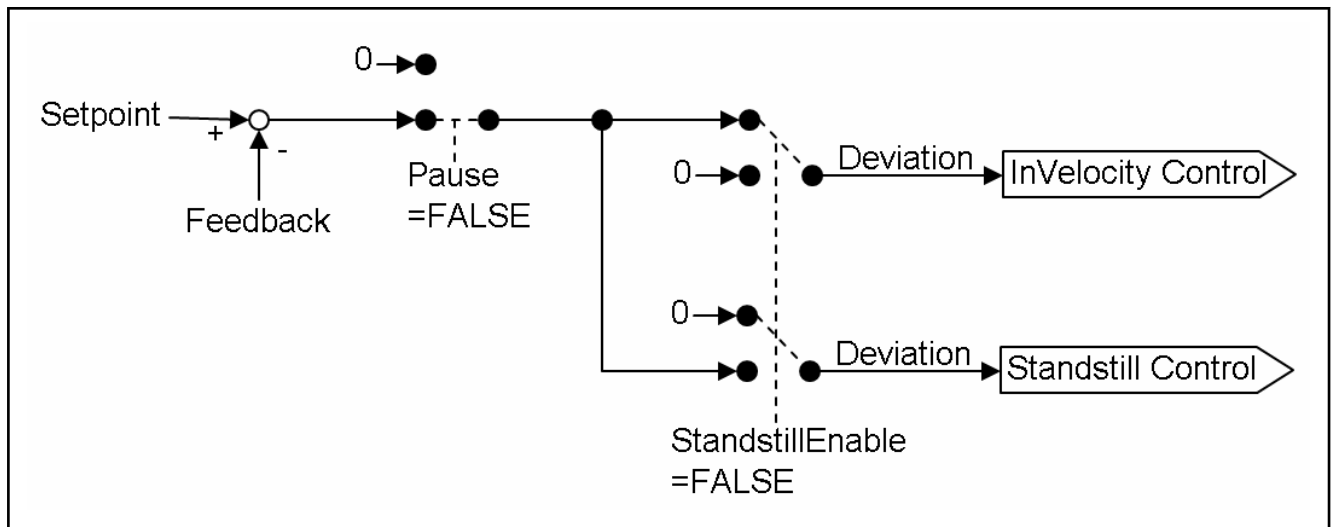


Fig. 4-97: Flowchart of the tension controller with branch for the control at standstill/in velocity

The "InVelocity Control" path is effective if the "StandstillEnable" input is not set. The tension is controlled via the parameter "Gear ratio fine adjustment" P-0-0694 (slave axis with interpolation in the drive/IPO Drive) or A-0-2605 (slave axis with interpolation on the control/IPO Control). It is directly written on the parameters if the "DisableCyclicWrites" input is not set. For real axes with interpolation in the drive, the configuration of the parameter P-0-0694 in the optional cyclic Sercos data (MDT) is thus required.

The control variable is additionally output via the "CorrectionValue" output.

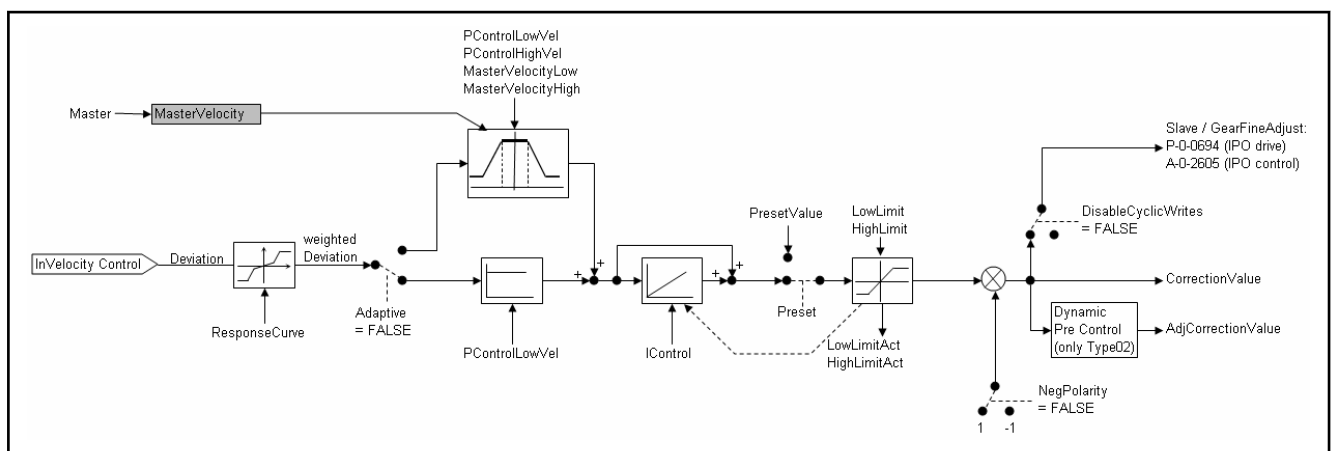


Fig. 4-98: Flowchart of the tension controller for the "InVelocity Control" branch

For the standstill control, another path can be activated to which the additive master axis position (P-0-0692) can be written as long as "DisableCyclicWrites" is inactive. At the same time, the value is always available via the "AdditivePosInc" output.

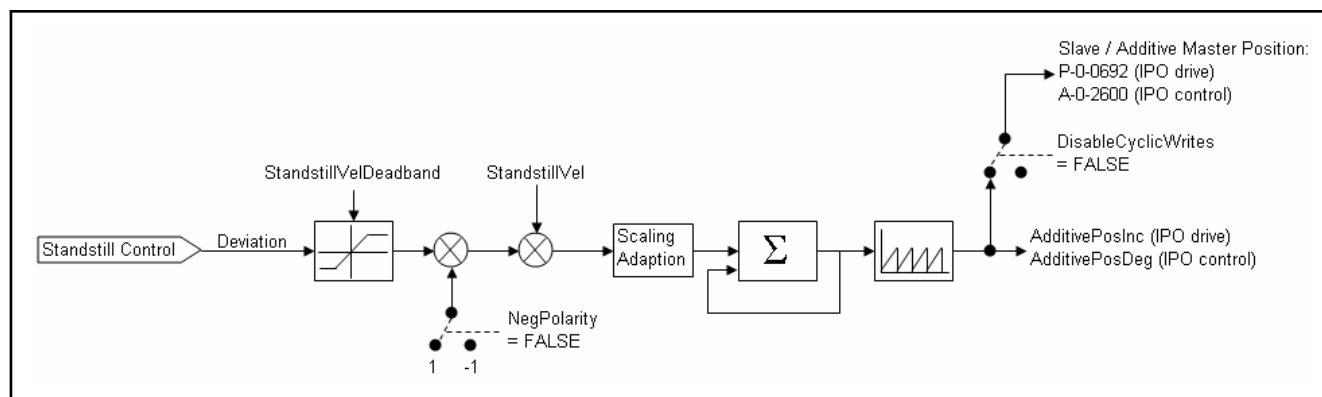


Fig. 4-99: Flowchart of the tension controller for the closed-loop control at standstill (Standstill Control)

To ensure user-friendliness, the flowchart does not contain the "WindowValue", "WindowMaxValue" or "WindowMinValue" limitations and their status displays.

Commissioning

If "DisableCyclicWrites" is inactive, P-0-0694 (Gear ratio fine adjustment) and P-0-0692 (Additive master axis position) parameters have to be applied to the cyclic command value channel of the slave axis. At a rising edge at "Enable", the cyclic data channels are checked. If one of the two parameters is not found, an error message is returned at the error outputs of the function block. If "DisableCyclicWrites" is enabled, the configuration of the cyclic command value data channels is not checked.

The axes used have to be in synchronous mode before the "Enable" input is activated. This requires the execution by the PLCopen function blocks "GearIn" or "GearInPos".

If the tension controller is used in a drive-integrated MLD, the AxisData structure has to be enabled in the drive.

Preset/PresetValid

A positive edge at the "Preset" input specifies an initial value for the control variable using the "PresetValue" variable. At the same time, the internal variables of the controller are set in a way that the controller is approached without any jerk. "Preset" acts only on the gear fine adjustment in the path "InVelocity Control".

The "Preset" input is only evaluated if "Enable" is TRUE. At a rising edge at "Preset", the new value is applied at "PresetValue" and "PresetAck" is set. A falling edge at "Preset" or "Enable" deletes "PresetAck". The "Preset" inputs dominates "Pause". That means that the value at "PresetValue" is applied at the output even though "Pause" is set. If "Preset" is set at the rising edge at "Enable", "PresetValue" is immediately applied as control variable.



Execute the "Preset" function in standstill (StandstillEnable=TRUE). Otherwise, velocity jumps can result due to the setting of the "ControlValue" to the "PresetValue". The value of the jump is proportional to the velocity.

The "Preset" input is only effective at the rising edge.

DisableCyclicWrites

The "DisableCyclicWrites" input determines whether the values calculated by the function block are to be written directly via the cyclic channel (FALSE) or whether they are only available at the function block output (TRUE).

Thus, it is possible to cascade several function blocks if "DisableCyclicWrites" is set to TRUE. Subsequently, only the outputs ("CorrectionValue"

Adaptive, PControlLowVel, PControlHighVel, MasterVelocityLow, MasterVelocityHigh

and "AdditivePosInc") can be used and modified later on. However, the user still has to write the values to the optional cyclic channel.

If "DisableCyclicWrites" is set to TRUE, it is not checked whether the optional cyclic parameters are also configured.

The P-gain can either be specified as constant or adaptive P-gain. Via the "Adaptive" input, either an adaptive or a constant P-gain is selected. Setting "Adaptive" to TRUE selects the adaptive P-gain.

If a constant P-gain is selected, only the value in "PControlLowVel" is effective for the P-gain.

If an adaptive P-gain is selected, the "PControlLowVel", "PControlHighVel", "MasterVelocityLow" and "MasterVelocityHigh" inputs have to be assigned. The characteristic curve looks as follows:

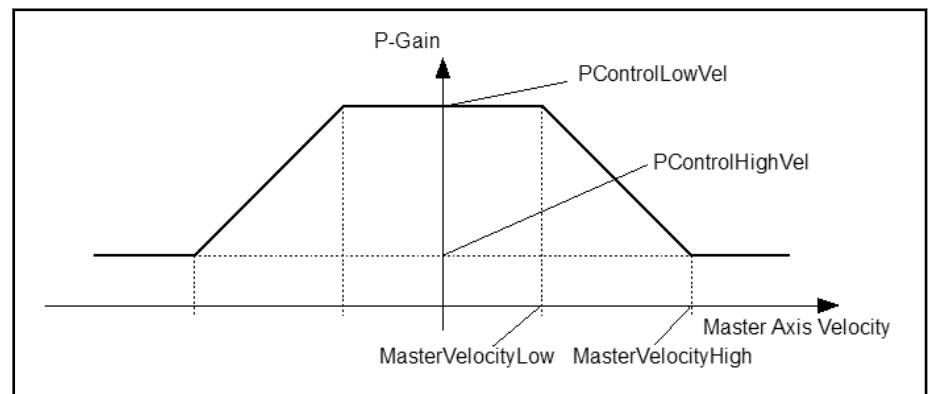


Fig. 4-100: Characteristic curve for the adaptive P-gain

The individual inputs describe the characteristic curve for positive master axis velocities. The characteristic curve for negative master axis velocities results from the mirroring at the axis of the P-gain. This means that only symmetric characteristics are possible. The P-gain between the master axis velocities "MasterVelocityLow" and "MasterVelocityHigh" is linearly interpolated. This results in some conditions for the value ranges of the inputs:

- The values for "MasterVelocityLow" and "MasterVelocityHigh" have to be positive
- The values for "MasterVelocityLow" and "MasterVelocityHigh" may not be identical
- The "MasterVelocityLow" value has to be lower than the "MasterVelocityHigh" value

If one of these conditions is not met, an error is generated at the function block output.

SetpointLock Setting the "SetpointLock" input applies the current actual tension value as the new command value for the tension controller. For example, the function allows the command value of the tension to obtain a reasonable initial value that reflects the current state of the machine.

Pause The "Pause" input allows the tension controller to pause. If this input is set to TRUE, the control deviation is set to "0", i.e. the control output is frozen at the current I-value. If "Pause" is active, the "InPause" output is set.

NegPolarity The "NegPolarity" input affects the control direction. Depending on the position of the draw roller with regard to the direction the material moves and the position of the feed roll, the control direction has to be either positive or negative. If this input is FALSE, the control direction is positive and a positive control deviation results in a positive control motion. The input also affects the standstill control.

Feedback	This input represents the actual value of the tension. It can be read via an analog input at the drive or at the Inline I/O for example. The user is responsible for a reasonable scaling.
PresetValue	Setting the "Preset" input specifies an initial value for the control variable using the "PresetValue" variable. At the same time, the internal variables of the controller are set in a way that the controller is approached without any jerk.
HighLimit, HighLimitAct	This input limits the control variable of the controller to a maximum value. If the limitation occurs, the I-value of the controller is no longer summed up ("Anti-reset-windup"). The maximum value is determined by the maximum value for the gear ratio fine adjustment defined in the drive. If a limitation occurs, the "HighLimitAct" output is set.
LowLimit, LowLimitAct	This input limits the control variable of the controller to a minimum value. If the limitation occurs, the I-value of the controller is no longer summed up ("Anti-reset-windup"). The minimum value is -95 irrespective of possible lower values in the drive (also refer to Gear ratio fine adjustment, page 276). If a limitation occurs, the "LowLimitAct" output is set.
PControlLowVel	This value corresponds to the proportional gain of the tension controller if the "Adaptive" input is not active. Otherwise, it represents the lower gain of an adaptive P-gain.
WindowValue, WindowIn	Use this value to define a symmetric window containing the current actual tension value. The window is defined as follows:

$$\left(SetPoint - |SetPoint| \cdot \frac{WindowValue}{100\%} \right) \leq Feedback \leq \left(SetPoint + |SetPoint| \cdot \frac{WindowValue}{100\%} \right)$$

Fig. 4-101: WindowValue, WindowIn

If this condition is met, the "WindowIn" output is set. If this condition is not met, the output is not set and there is no limitation.

WindowMaxValue, WindowMax	This value can be used to set the upper value for the actual tension value. The value is evaluated as follows:

$$Feedback > \left(SetPoint + |SetPoint| \cdot \frac{WindowMaxValue - 100}{100} \right)$$

Fig. 4-102: WindowMaxValue, WindowMax

If this condition is met, the "WindowMax" output is set. If this condition is not met, the output is not set and there is no limitation.

WindowMinValue, WindowMin	This value can be used to specify a lower value for the actual tension value. The value is evaluated as follows:

$$Feedback < \left(SetPoint + |SetPoint| \cdot \frac{WindowMinValue - 100}{100} \right)$$

Fig. 4-103: WindowMinValue, WindowMin

If this condition is met, the "WindowMin" output is set. If this condition is not met, the output is not set and there is no limitation.

StandstillEnable	Set the "StandstillEnable" input to additively add the velocity specified in "StandstillVel" to the axis to be controlled. This allows a closed-loop tension control at standstill as well.

	 At lower velocities, the influence of the gear ratio fine adjustment decreases continuously. There is also an open control loop at master axis standstill, since the gear ratio fine adjustment does no influence anymore. Thus, the "StandstillEnable" input has to be set when reaching the standstill window in order not to increase the I-value up to the limit in the "InVelocity Control" path by the open control loop.
	 For axes with interpolation on the control, the "AdditivePosInc" output is given in modulo format 2^{20} (12VRS or higher).
	 If this input is used, ensure that the tension controller is only used in the Sercos-synchronous task. If this is not the case, the motor cannot run smoothly or oscillation around the standstill position can result.
StandstillVel	This value indicates the additive velocity sent to the drive when setting the "StandstillEnable" input. The units for the velocity correspond to the scaling set in the drive.
	 At standstill, the velocity is generated in jumps. Since the velocity has to be reached in one Sercos cycle, the values specified cannot be indefinitely high. If the value selected for the standstill velocity is too high, the drive can oscillate. The maximum value depends on the scaling of the velocity, the maximum acceleration set and the mass connected to the drive. To avoid big velocity jumps, a filter time constant can be defined in the drive parameter P-0-0693. This smoothes the jump.
ResponseCurve	Using the "ResponseCurve" input, the control deviation can be scaled. Use this functionality to specify a smaller scaling of the control deviation for smaller control deviations. This results in a lower proportional gain and thus causes a smoother control behavior in a steady state. When parameterizing the input, the controller parameters should first be set without evaluation. Subsequently, a lower scaling should be specified. This allows to globally set the controller parameters first using the system dynamics, and then, a local optimization is ensured (e.g. for the steady state). The local optimization is parameterized with a characteristic curve with up to four data points. Negative deviations are processed by point reflecting the curve at the origin. This results in a total of nine linear sections. The section s5 runs through the origin. The sections s1 and s9 have a slope of 1 and each run into positive or negative infinity. If the input is not assigned, a 1-to-1 characteristic curve is used as default value. That means that it is not scaled.

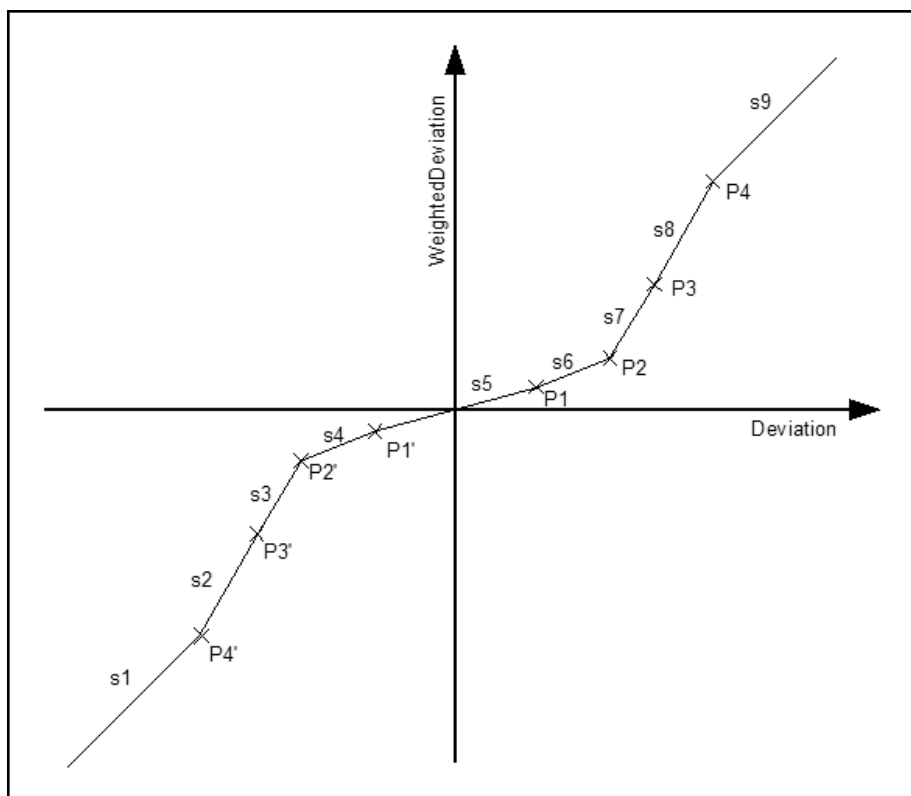


Fig. 4-104: Scaling the control deviation using the "ResponseCurve" input

Points P1 to P4 are arranged in an array made up of IL_COORD_TYPE01. If fewer than four points are required, the X-coordinates of the points not required (beginning with the last point) are set to 0. The X-values have to be positive and rise in a positive direction. It is permitted to use the same point twice.

Background information

Gear ratio fine adjustment

The tension controller generally operates with velocities greater than 0. The gear ratio fine adjustment is an appropriate tool to maintain the tension. However, there are also some boundary conditions to be explained.

The gear ratio fine adjustment is internally limited to a minimum of -95%. If a lower value is entered, the function block outputs an error.

The reason for this limitation is the calculated output value defined as follows for a given input value:

$$\text{Output value} = \text{Input value} * \left(1 + \frac{\text{Gear fine adjust}}{100} \right)$$

Based on the equation, it can be seen that - when the gear ratio fine adjustment approaches 100% - the output value approaches 0. Thus, the control loop practically opens since a modification of the input value does not cause any response at the output. It is even worse at a gear ratio fine adjustment of less than -100%, since the sign of the output value reverses. In the control loop, positive feedback occurs that can result in an unstable system.

StandstillEnable, StandstillVel, StandstillVelDeadband

The standstill control can be executed using the "StandstillEnable" and "StandstillVel" inputs. This adds an additive velocity. To achieve better control quality, it is reduced by the command value. This can be executed with a modified signum function provided with the following transfer function:

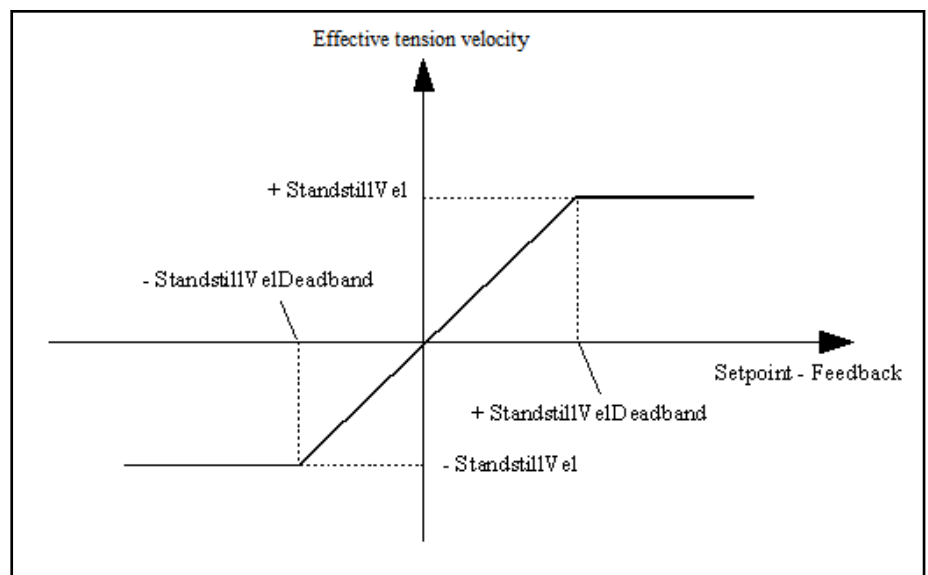


Fig. 4-105: Modified signum function to limit the added "StandstillVel"

The control deviation is relativized in the "Setpoint" range. If the "Feedback" approaches the "Setpoint", the closed-loop control becomes "more sensitive" than a simple two-position controller. The value responsible for the limitation can be set at the "StandstillVelDeadband" input. The values are provided with the same units as the "Setpoint" input or the "Feedback" input. There is a linear interpolation between the two extreme values "+StandstillVel" and "-StandstillVel" so that the transfer function above results for a control deviation and a tension velocity.

Subsequently, the factor determined is multiplied by the specified "StandstillVel". The velocity is processed as speed with the scaling set in the drive. If the web velocity should be entered instead of the speed, calculate it before.

To reverse the direction of motion for the standstill control, set the "NegPolarity" input.

Subsequently, the result is multiplied by a correction factor. This correction factor results from the gear ratio. However, the gear ratio fine adjustment is not considered. The gear ratio fine adjustment is generally very small and therefore not important. The gear ratio is only applied at a rising edge at the "Enable" input.

The flowchart shows that the velocity specified is converted to increments. A few rules are to be considered:

- The specified velocity cannot be unlimited, since the drive has to be able to reach the velocity in one Sercos cycle. Thus, only very small values are allowed. The value size depends on the scaling and the mass connected to the motor. The calculated increments are specified via parameter P-0-0692. The increments calculated are specified using the parameter P-0-0692. This parameter can also be smoothed using the filter (P-0-0693) available in the drive and thus allows larger jumps.
- The calculated increments are added before the gear. The gear ratio (A-0-2720 and A-0-2721 or P-0-0156 and P-0-0157) is included in the calculation. The gear ratio fine adjustment (P-0-0083 and P-0-0694) is not considered since it is generally very small. This can cause that the axis moves at a velocity different from the velocity set.
- The units are scaled. In addition, the velocity value entered is not checked against the maximum and minimum values. This can result in very large values for the number of increments per Sercos cycle. If the

calculated intermediate value exceeds the maximum values by $\pm$ half of a modulo value, the values are automatically limited. This results in a less additive velocity than specified at the "StandstillVel" input.

- At very low velocities, it has to be considered as well that the current velocity can deviate from the specified velocity. This can be caused due to the increments granulated per Sercos cycle: For example, corresponds to 1 increment corresponds to a velocity of 0.03 rpm, only velocities that are multiples of 0.03 rpm can be set. If this is not the case, it is rounded up to the next value.
- The calculated increments are processed with each Sercos cycle. Thus, when using the standstill control, the function block has to be processed in the Sercos-synchronous task. If this is not the case, the drive moves one Sercos cycle by the calculated number of increments and in the following Sercos cycles it comes to standstill again which results in a jerky motion. The filter in P-0-0693 can also be helpful.

Troubleshooting The function block generates the following error messages in Additional1/Additional2 for the table **F_RELATED_TABLE**, 16#0170:

ErrorID	Additional1	Additional2	Description
ACCESS_ERROR (16#0004)	16#00000A00	16#00000001	The parameter P-0-0694 is not parameterized in the optional cyclic telegram of the slave axis
ACCESS_ERROR (16#0004)	16#00000A00	16#00000002	The parameter P-0-0692 is not parameterized in the optional cyclic telegram of the slave axis
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000001	The master axis velocity in the lower operating point ("MasterVelocityLow") is < 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000002	The master axis velocity in the upper operating point ("MasterVelocityHigh") is < 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000003	The master axis velocity in the lower operating point ("MasterVelocityLow") and upper operating point ("MasterVelocityHigh") are provided with the same values
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000004	The value of the master axis velocity in the lower operating point ("MasterVelocityLow") is greater than the value in the upper operating point ("MasterVelocityHigh")
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000005	The value of the "LowLimit" input is lower than the minimum allowed (-95.0)
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000006	The value of the inputs "PGainLowVel" or "PGainHighVel" is lower than or equal to 0
DEVICE_ERROR (16#0008)	16#00000A02	16#00000007	No power is added to the drive specified in "Slave". A closed-loop tension control is therefore not possible
DEVICE_ERROR (16#0008)	16#00000A02	16#00000008	The drive specified in the "Slave" is not in a synchronous mode
RESOURCE_ERROR (16#0003)	16#00001503	16#00000000	AxisData is not active in the control. This error occurs only on the MLD

Tab. 4-53: Error codes of the MB_TensionControlLoadCellType01 function block

Special features for the MLD

Interpolation in the drive has to be activated in the drive for the tensile stress controller axis when using the MLD-S control with a superordinate MLC control (**Real axes** ► **Corresp. axis** ► **Properties**). This setting does not exist when using the MLD-M control.

The parameters "P-0-0694 Gear ratio fine adjustment, process loop" and "P-0-0692 Additive master axis position, process loop" have to be entered into the cyclic channel between the control and the drive of the corresponding drive ("UserCmdDataX")

The real axes are parameterized in the parameterization mode in IndraWorks (the dialog is located in the project tree below the MLD node **AxisData**).

The "AxisData structure" of the drive has to be parameterized ("P-0-1367 PLC configuration, bit 6 is set" or dialog "PLC configuration").

Axis reference for the master axis:

For technology function blocks, the following is recommended as axis reference for the master axis:

- Use axis reference "**VmAxis1**" for a virtual master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 20 (VMA_1).
- Use axis reference "**RmAxis1**" for a real master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 23 (RMA_1).

Additionally, the logic address of the measuring encoder has to be entered in parameter "P-0-1820.0.2 CCD: Source of measuring encoder position feedback value".

For detailed information, refer to Functional Description MLD.

4.8.3 MB_TensionControlLoadCellMatParType01

Brief description

Depending on the measured value of a load cell, the MB_TensionControlLoadCellMatParType01 function block controls the web tension to the command value. Compared to the function block [MB_TensionControlLoadCellType01](#), [page 266](#), this variant offers the following benefits.

- Controller parameterization by machine and material data
- Individual controller design (command value or disturbance response)
- Customized control loop dynamics (aperiodic or periodic)
- Automatic adaptation of the controller parameters

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechMotion

Tab. 4-54: MB_TensionControlLoadCellMatParType01 reference table



The "ResponseCurve" input is declared in the RIL_LoopControl library. Therefore, the RIL_LoopControl library has to additionally be integrated in the library manager if the "ResponseCurve" input is used.

Interface description

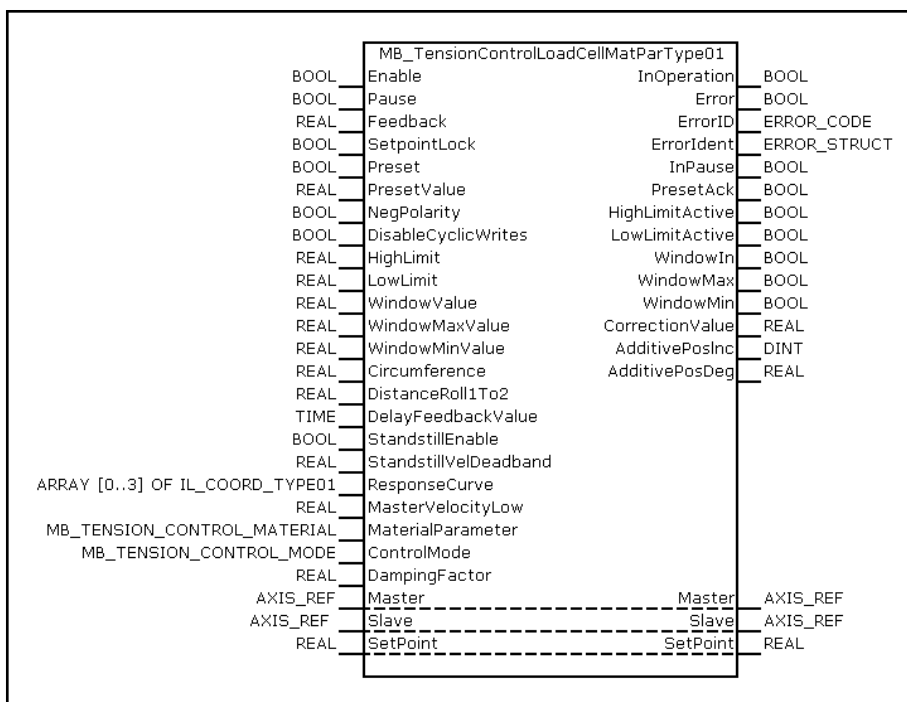


Fig. 4-106: MB_TensionControlLoadCellMatParType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	If this input is set, the function block operates
	Pause	BOOL	If this input is set, the tension controller is paused and the output value is frozen. The "InPause" output signalizes an active "Pause"
	Feedback	REAL	Current actual value of the tension [N]*
	SetpointLock	BOOL	With a rising edge at this input, the current "Feedback" actual tension value is applied as new "Setpoint" command value for the tension controller
	Preset	BOOL	With a rising edge at this input, the control output is set to the value parameterized in "PresetValue" and, if successful, the "PresetAck" output is set
	PresetValue	REAL	Value to which the "CorrectionValue" controller output is set if the "Preset" input is set [%]
	NegPolarity	BOOL	If the input is set, the control direction is inverted
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output.
	HighLimit	REAL	Maximum value of the controller output [%]. If the control variable calculated by the controller is greater than this value, the control variable is limited and the I-value of the controller is not summed up anymore
	LowLimit	REAL	Minimum value of the controller output [%]. If the control variable calculated by the controller is lower than this value, the control variable is limited and the I-value of the controller is not summed up anymore

I/O type	Name	Data type	Description
	WindowValue	REAL	Process parameter window (relative) [%]
	WindowMaxValue	REAL	Maximum value of the process parameter (absolute) [%]
	WindowMinValue	REAL	Minimum value of the process parameter (absolute) [%]
	Circumference	REAL	Axis circumference of the master axis [mm]*
	DistanceRoll1To2	REAL	Web length between master axis and slave axis [mm]*
	DelayFeedbackValue	TIME	Delay time T_u of the web tension control system. Total of all peripheral time constants. E.g. motor dynamics, filter times, sensors, motion cycle time.
	StandstillEnable	BOOL	StandstillEnable = FALSE: "InVelocity Control" branch is active. Control variable: Gear ratio fine adjustment StandstillEnable = TRUE: "Standstill control" branch is active. Control variable: Master axis position, additive
	StandstillVelDeadband	REAL	Maximum effective control deviation in the standstill control [N]*. The tensioning velocity is limited according to this control deviation
	ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	Characteristics to scale the control deviation in the "In-Velocity Control" branch
	MasterVelocityLow	REAL	Master axis velocity below which the controller adaptation is limited [1/min]
	MaterialParameter	MB_TENSION_CONTROL_MATERIAL	Structure with material-specific data (type, width, thickness, grammage and modulus of elasticity) MB_TENSION_CONTROL_MATERIAL, page 292
	ControlMode	MB_TENSION_CONTROL_MODE	Selection of the control loop response. Optimized for command value or disturbance response MB_TENSION_CONTROL_MODE, page 293
	DampingFactor	REAL	This input affects the control loop response [%]. It can be steplessly selected between a slow aperiodic and a fast periodic response.
VAR_OUTPUT	InOperation	BOOL	The tension controller operates and the outputs are valid
	Error	BOOL	Indicates that an error occurred
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description according to error table (see Troubleshooting, page 290)
	InPause	BOOL	"Pause" is active
	PresetAck	BOOL	If this output is active, the "Preset" input was analyzed and the "CorrectionValue" output applied the "Preset-Value" value

I/O type	Name	Data type	Description
	HighLimitActive	BOOL	This output is set if the control variable calculated by the controller is greater than the value specified in "HighLimit"
	LowLimitActive	BOOL	This output is set if the control variable calculated by the controller is lower than the value specified in "LowLimit"
	WindowIn	BOOL	If TRUE is set, the process variable is in the parameterized window
	WindowMax	BOOL	If TRUE is set, the process variable exceeded the parameterized maximum value
	WindowMin	BOOL	If TRUE is set, the process variable is below the parameterized minimum value
	CorrectionValue	REAL	Control variable of the tension controller [%]
	AdditivePosInc	DINT	Additive master axis position in increments. It is used for the real slave axis with interpolation in the drive (target parameter: P-0-0692) [-]
	AdditivePosDeg	REAL	Additive master axis position in degrees. It is used for the slave axis with interpolation on the control system (e.g. for virtual axes or real axes with interpolation on the control. Target parameter: A-0-2600) [°]
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Reference to the master axis
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Reference to the slave axis
VAR_IN_OUT	Setpoint	REAL	Tension command value [N]*. This value is generally set by the user but can also be changed by the function block (see "SetpointLock")

*

The units [N] and [mm] are obligatory if the internal material database (E [N/mm²]) is used. If user-defined modulus of elasticity are used, you may use any unit such as [kg] and/or [ft]. Ensure that the selected units are always used.

Tab. 4-55: Interface variables of the MB_TensionControlLoadCellMatParType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL	FALSE	TRUE	FALSE	Continuous
Pause	BOOL	FALSE	TRUE	FALSE	Continuous
Feedback	REAL	n.def.	n.def.	0.0	Continuous
SetpointLock	BOOL	FALSE	TRUE	FALSE	Continuous
Preset	BOOL	FALSE	TRUE	FALSE	Continuous
PresetValue	REAL	-95.0	n.def.	0.0	Rising edge at "Preset"
NegPolarity	BOOL	FALSE	TRUE	FALSE	Continuous

Name	Type	Min. value	Max. value	Default value	Effective
DisableCyclic-Writes	BOOL	FALSE	TRUE	FALSE	Rising edge at "Enable"
HighLimit	REAL	>LowLimit	n.def.	200.0	Continuous
LowLimit	REAL	-95.0	<HighLimit	-50	Continuous
WindowValue	REAL	n.def.	n.def.	0.0	Continuous
WindowMaxValue	REAL	>=WindowMinValue	n.def.	0.0	Continuous
WindowMinValue	REAL	n.def.	<=WindowMaxValue	0.0	Continuous
Circumference	REAL	>0.0	n.def.	100.0	Rising edge at "Enable"
DistanceRoll1To2	REAL	>0.0	n.def.	1000.0	Rising edge at "Enable"
DelayFeedback-Value	TIME	n.def.	n.def.	T#100ms	Continuous
StandstillEnable	BOOL	FALSE	TRUE	FALSE	Continuous
StandstillVel-Deadband	REAL	>0.0	n.def.	1.0	Continuous
ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	See RIL_Loop-Control	See RIL_Loop-Control	See RIL_Loop-Control	Rising edge at "Enable"
MasterVelocity-Low	REAL	>0.0	n.def.	1.0	Continuous
MaterialParameter	MB_TENSION_CONTROL_MATERIAL	See data structure	See data structure	See data structure	Continuous
ControlMode	MB_TENSION_CONTROL_MODE	See data structure	See data structure	See data structure	Continuous
DampingFactor	REAL	20.0	200.0	100.0	Continuous
Master	AXIS_REF	see RIL_CommonTypes	see RIL_CommonTypes	see RIL_CommonTypes	Rising edge at "Enable"
Slave	AXIS_REF	see RIL_CommonTypes	see RIL_CommonTypes	see RIL_CommonTypes	Rising edge at "Enable"
Setpoint	REAL	n.def.	n.def.	0.0	Continuous

Tab. 4-56: MB_TensionControlLoadCellMatParType01 input behavior

Functional description

The mode of action of the tension controller is shown in the following flowcharts. Depending on the "StandstillEnable" input, either the "InVelocity Control" or "Standstill Control" path is effective.

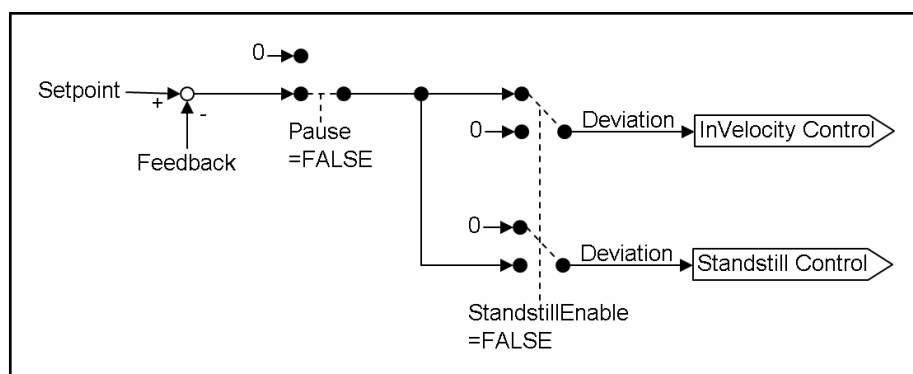


Fig. 4-107: Flowchart of the tension controller with branch for the control at standstill/in velocity

The "InVelocity Control" path is effective if the "StandstillEnable" input is not set. The tension is controlled via the parameter "Gear ratio fine adjustment" P-0-0694 (slave axis with interpolation in the drive/IPO Drive) or A-0-2605 (slave axis with interpolation on the control/IPO Control).

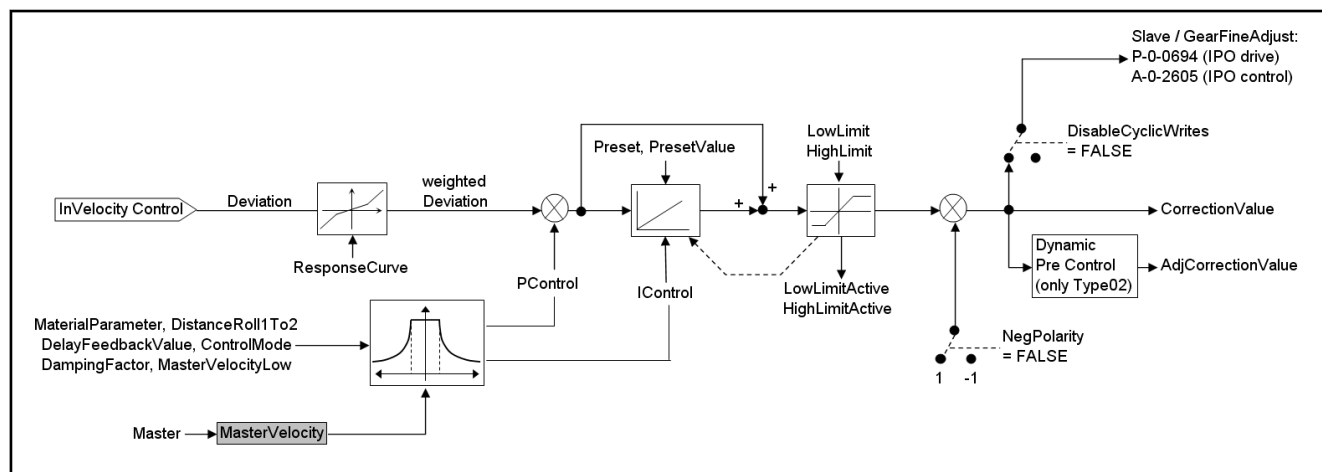


Fig. 4-108: Flowchart of the tension controller for the (InVelocity Control) branch

For the control during standstill, the "Standstill Control" path may be activated. This path is effective if the "StandstillEnable" input is set. The tension is controlled via the "additive master axis position" P-0-0692 (IPO Drive) or A-0-2600 (IPO Control) parameter.

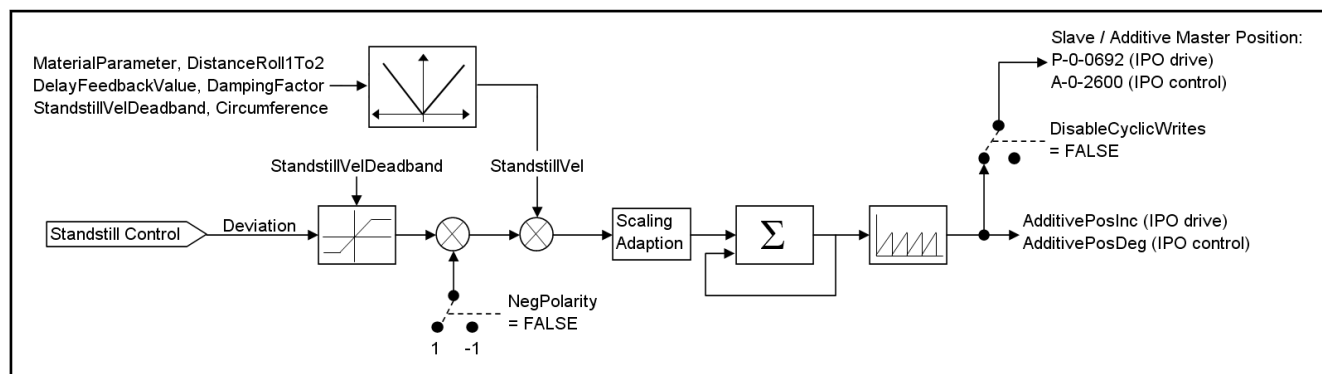


Fig. 4-109: Flowchart of the tension controller for the closed-loop control at standstill (Standstill Control)

To ensure user-friendliness, the flowcharts do not contain the "WindowValue", "WindowMaxValue" or "WindowMinValue" limitations and their status displays.

Commissioning If "DisableCyclicWrites" is inactive and a real slave axis with interpolation is used in the drive, the P-0-0694 ("InVelocity Control") and P-0-0692 ("Standstill Control") parameters have to be applied to the cyclic command value channel of the slave axis. At a rising edge at "Enable", the cyclic data channels are checked. If one of the two parameters is not found, an error message is returned at the error outputs of the function block. If "DisableCyclicWrites" is enabled, the configuration of the cyclic command value data channels is not checked.

The following explains the individual function block inputs in detail.

In general, all inputs are continuously analyzed if the "Enable" input is set. That means that changes made to the inputs are immediately effective without toggling the "Enable" input again. The exception to this rule are the "PresetValue", "DisableCyclicWrites", "Circumference", "DistanceRoll1To2", "ResponseCurve", "Master" and "Slave" inputs. In this connection, the "PresetValue" input will only be accepted in case of a rising edge at the "Preset" input and the "DisableCyclicWrites", "Circumference", "DistanceRoll1To2", "ResponseCurve", "Master" and "Slave" inputs only in case of a rising edge at the "Enable" input.

Pause, InPause The "Pause" input allows the tension controller to pause. If the input is set to TRUE, the current controller control signal will be frozen. If "Pause" is active, the "InPause" output is set.



If no tension control is used at standstill (StandstillEnable is always FALSE), the "Pause" input has to be set at machine standstill.

Feedback This input represents the actual value of the tension. It can be read via an analog input at the drive or at the Inline I/O for example. The user is responsible for a reasonable scaling.

SetpointLock Setting the "SetpointLock" input applies the current actual tension value as the new "Setpoint" command value for the tension controller. For example, the function allows the command value of the tension to obtain a reasonable initial value that reflects the current state of the machine.

Preset, PresetValue, PresetAck A positive edge at the "Preset" input specifies an initial value for the "CorrectionValue" control variable using the "PresetValue" variable. At the same time, the internal variables of the controller are set in a way that the controller is approached without any jerk. "Preset" acts only on the gear fine adjustment in the path "InVelocity Control".

The "Preset" input is only evaluated if "Enable" is TRUE. At a rising edge at "Preset", the new value is applied at "PresetValue" and "PresetAck" is set. A falling edge at "Preset" or "Enable" deletes "PresetAck". The "Preset" inputs dominates "Pause". That means that the value at "PresetValue" is applied at the output even though "Pause" is set. If "Preset" is set at the rising edge at "Enable", "PresetValue" is immediately applied as control variable.



The "Preset" function is to be executed in standstill (StandstillEnable=TRUE). Otherwise, velocity jumps can occur due to the setting of the "CorrectionValue" to the "PresetValue". The value of the jump is proportional to the velocity.

The "Preset" input is only effective at the rising edge.

NegPolarity The "NegPolarity" input affects the control direction. Depending on the position of the draw roller ("slave" axis) with regard to the direction the material moves and the position of the feed roll ("master" axis), the control direction has to be either positive or negative. If this input is FALSE (positive control

	direction), a positive control deviation leads to a positive control motion. The input also affects the standstill control.
DisableCyclicWrites	The "DisableCyclicWrites" input determines whether the values calculated by the function block are to be written directly via the cyclic channel (FALSE) or whether they are only available at the function block output (TRUE). If "DisableCyclicWrites" is set to TRUE, it is possible to pick up and modify the "CorrectionValue", "AdditivePosInc" and "AdditivePosDeg" controller outputs without cyclically applying them to the command value channel of the slave axis. Apart from that, it will in this case not be checked whether the optional cyclic parameters have been configured. Consequently, it is in this case necessary to independently write the values to the optional cyclic channel.
HighLimit, HighLimitAct	This input limits the control variable of the controller to a maximum value. If the limitation occurs, the I-value of the controller is no longer summed up ("Anti-reset-windup"). The maximum value is determined by the maximum value for the gear ratio fine adjustment defined in the drive. If a limitation occurs, the "HighLimitAct" output is set.
LowLimit, LowLimitAct	This input limits the control variable of the controller to a minimum value. If the limitation occurs, the I-value of the controller is no longer summed up ("Anti-reset-windup"). The minimum value is -95 irrespective of possible lower values in the drive (also refer to Gear ratio fine adjustment, page 290). If a limitation occurs, the "LowLimitAct" output is set.
WindowValue, WindowIn	With the "WindowValue" value, a symmetric window can be defined containing the current actual tension value. The window is defined as follows:

$$\left(\text{SetPoint} - |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right) \leq \text{Feedback} \leq \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right)$$

Fig. 4-110: WindowValue, WindowIn

If this condition is met, the "WindowIn" output is set. If the condition is not met, the output is not set. There is only a Boolean output, however no limitation at all.

WindowMaxValue, WindowMax	This value can be used to set the upper value for the actual tension value. The value is evaluated as follows:

$$\text{Feedback} > \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowMaxValue} - 100}{100} \right)$$

Fig. 4-111: WindowMaxValue, WindowMax

If this condition is met, the "WindowMax" output is set. If the condition is not met, the output is not set. There is only a Boolean output, however no limitation at all.

WindowMinValue, WindowMin	This value can be used to specify a lower value for the actual tension value. The value is evaluated as follows:

$$\text{Feedback} < \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowMinValue} - 100}{100} \right)$$

Fig. 4-112: WindowMinValue, WindowMin

If this condition is met, the "WindowMin" output is set. If the condition is not met, the output is not set. There is only a Boolean output, however no limitation at all.

Circumference, DistanceRoll1To2

The "Circumference" and "DistanceRoll1To2" inputs are used for the internal calculation of the control parameters. They describe the individual, structural conditions of the plant. The "Circumference" input describes the circumference of the "master" axis and the "DistanceRoll1To2" input the material web length between the "master" and the "slave" axis.

DelayFeedbackValue

The "DelayFeedbackValue" input is used for the internal calculation of the controller parameters. It describes the delay time T_u of the web tension control system. So this value corresponds to the total of peripheral time constants such as motor dynamics, filter times, sensors, motion cycle time.

StandstillEnable, StandstillVel-Deadband

Set the "StandstillEnable" input to add an additive velocity to the axis to be controlled in order to control the web tension. As long as the "StandstillVel-Deadband" limitation is not enabled, this speed is directly proportional to the control difference.



At lower velocities, the influence of the gear ratio fine adjustment decreases continuously. There is also an open control loop at master axis standstill, since the gear ratio fine adjustment does no influence anymore. Thus, the "StandstillEnable" input has to be set when reaching the standstill window in order not to increase the I-value up to the limit in the "InVelocity Control" path by the open control loop.



For axes with interpolation on the control system, the "AdditivePosInc" output is given in modulo format 2^{20} .

The "StandstillVelDeadband" control signal limitation corresponds to a limitation of the control difference and thus – due to the proportional controller – also a limitation of the control signal. The value is provided with the same units as the "Setpoint" input or the "Feedback" input. If the difference between "Setpoint" and "Feedback" is larger than the value specified at the "StandstillVelDeadband" input, the tension velocity will be limited. This tension velocity limitation is e.g. relevant for a newly inserted, non-tensioned material web. Accordingly, this input is to be set so that control differences relevant with regard to the control technology can be corrected without limitation and control differences caused by the process are considered by the limited tension velocity. A "StandstillVelDeadband" of 5% to 25% of the "Setpoint" should be sufficient for the most processes.



At standstill, the axis velocity is generated in jumps. Since the velocity has to be reached in one Sercos cycle, the values specified cannot be indefinitely high and the tension controller has to be used in the Sercos-synchronous task. The maximum value depends on the maximum acceleration of the drive and the connected mass. In order to prevent large velocity jumps, you can on the one hand increase the filter time constant in the P-0-0693 drive parameter (4 times the Sercos cycle time is recommended) and/or on the other hand limit the maximum control signal by means of the "StandstillVelDeadband" standstill window.

To reverse the direction of motion for the standstill control, set the "NegPolarity" input.

ResponseCurve

Using the "ResponseCurve" input, the control deviation can be scaled. Use this functionality to specify a smaller scaling of the control deviation for smaller control deviations. This results in a lower proportional gain and thus causes a smoother control behavior in a steady state. When parameterizing the

input, the controller parameters should first be set without evaluation. Subsequently, a lower scaling should be specified. This allows to globally set the controller parameters first using the system dynamics, and then, a local optimization is ensured (e.g. for the steady state).

The local optimization is parameterized with a characteristic curve with up to four data points. Negative deviations are processed by point reflecting the curve at the origin. This results in a total of nine linear sections. The section s5 runs through the origin. The sections s1 and s9 have a slope of 1 and each run into positive or negative infinity. If the input is not assigned, a 1-to-1 characteristic curve is used as default value. That means that it is not scaled.

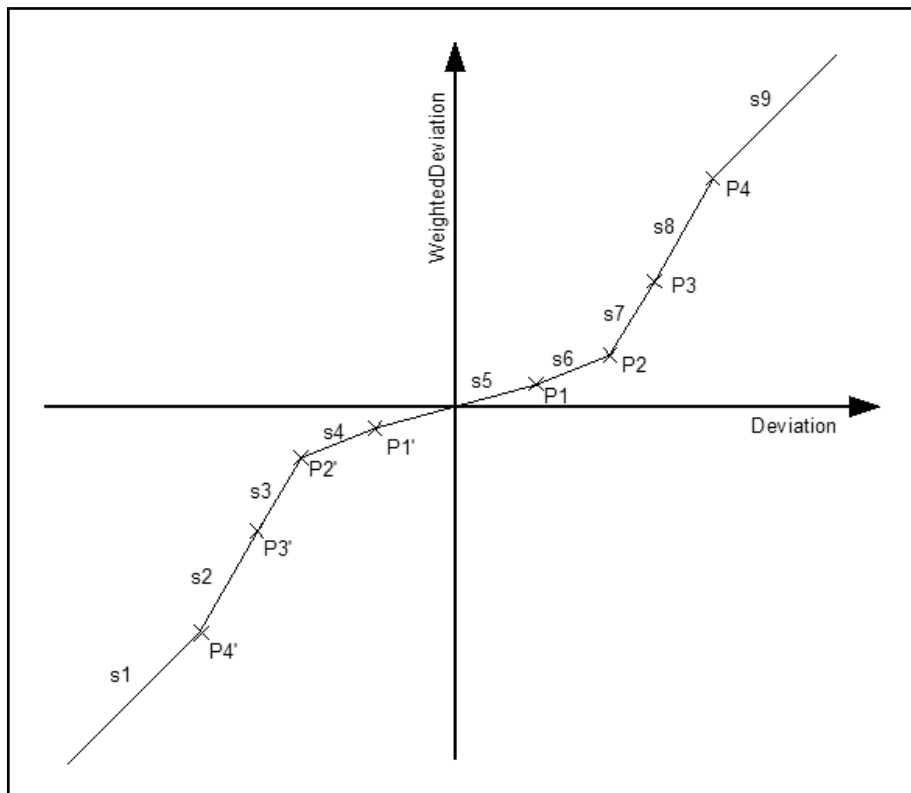


Fig. 4-113: Scaling the control deviation using the "ResponseCurve" input

Points P1 to P4 are arranged in an array made up of IL_COORD_TYPE01. If fewer than four points are required, the X-coordinates of the points not required (beginning with the last point) are set to 0. The X-values have to be positive and rise in a positive direction. It is permitted to use the same point twice.

MasterVelocityLow

The "MasterVelocityLow" input serves to limit the controller parameters. Due to the speed-dependent controlled system response, speed-dependent controller parameters result. Depending on the desired control loop response, either the controller gain and/or the controller reset time is speed-dependent. The following substitutionally represents the dependency of the controller gain from the web velocity in a graphical form.

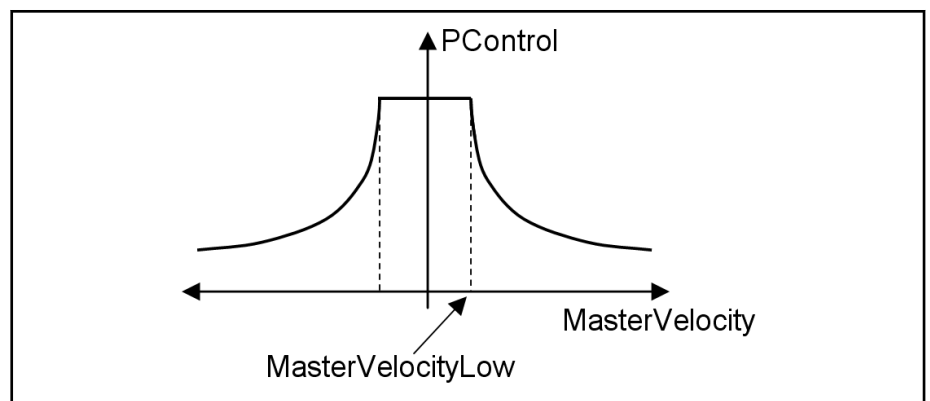


Fig. 4-114: Dependency of the controller gain from the web velocity

As it has already been explained in the section "StandstillEnable", the influence of the gear ratio fine adjustment decreases continuously at lower velocities. This behavior is directly reflected in the controller gain. In order to prevent unnaturally large controller parameters, the "MasterVelocityLow" input serves to limit the controller parameters in case of "low" web velocities. In most cases, the value of this input should lie below the smallest production speed and above the switchover speed in the standstill path. This ensures unlimited control loop dynamics above the "MasterVelocityLow" velocity and at the same time, the controller parameters are limited in case of "low" web velocities.

MaterialParameter

The "MaterialParameter" input structure is used for the internal calculation of the control parameters. This structure contains information regarding the individual web material. The individual elements of this structure describe the type, the width, the thickness and/or the grammage and the modulus of elasticity of the web material. For web materials consisting of paper, the grammage "WebGrammage" and for other web materials the material thickness "WebThickness" is analyzed. When selecting a given web material "MaterialParameter.MaterialType", the modulus of elasticity is taken from the internal database. Accordingly, the "MaterialParameter.MaterialEModule" input will only be analyzed if a user-defined material type "CUSTOM_PAPER" or "CUSTOM_NON_PAPER" is selected. The following two exemplary circuits are to clarify this:

Example:

Paper from block-internal database

```
MaterialParameter.MaterialType := OFFSET_PAPER;
MaterialParameter.WebWidth := 500;
MaterialParameter.WebGrammage := 80;
```

Example:

User-defined web material (e.g. foil)

```
MaterialParameter.MaterialType := CUSTOM_NON_PAPER;
MaterialParameter.WebWidth := 500;
MaterialParameter.WebThickness := 200;
MaterialParameter.MaterialEModule := 650;
```

If the web material used is not listed in the internal database and if there is no information regarding the modulus of elasticity, it may be determined in a standstill measurement.

$$\Delta F \approx A \cdot E \cdot \frac{\Delta L}{L} \Rightarrow E \approx \frac{L \cdot \Delta F}{A \cdot \Delta L}$$

Fig. 4-115: Hooke's law

You can see from this Hooke's formula that by means of a measured force difference, due to a given elongation (change in length due to an additive master axis position), the individual modulus of elasticity [N/mm²] can be determined at standstill. Apart from that, you can also directly determine the product of modulus of elasticity [N/mm²] and cross-sectional area [mm²], and by standardization of the cross-sectional area ("MaterialType:=CUSTOM_NON_PAPER", "WebWidth:=1[mm]", "WebThickness:=1000[μm]") to A=1[mm²], it may be specified as modified modulus of elasticity [N].

ControlMode You can use the "ControlMode" input to determine the primary control loop response. The web tension controller may either be optimized for command value or for disturbance response. In case of fixed value control of the web tension, "CONTROL_MODE_OPT_DISTURBANCE" is to be assigned to this input. As a result, the internal controller parameters will be determined so that faults are perfectly corrected. The other way round, "CONTROL_MODE_OPT_COMMAND" is to always be assigned to this input in case of sequential control. As a result, the internal controller parameters will be determined so that the controlled variable perfectly follows the reference variable.

DampingFactor You can use the "DampingFactor" input to determine the control loop dynamics. It acts on both branches, standstill control and control during web operation. The control loop dynamics may either be set aperiodically (without overshoot) or periodically (with overshoot). For exactly controlled system parameters, a "DampingFactor" of 100% corresponds exactly to an aperiodic boundary case and a "DampingFactor" of 120% corresponds to a transient oscillation with approximately 20% overshoot. In addition, the control loop response can be decelerated accordingly with a "DampingFactor" of up to 20% or can be accelerated with a "DampingFactor" of 200% respectively. If there are controlled system parameters ("Circumference", "DistanceRoll1To2", "DelayFeedbackValue", "MaterialParameter") or a system response deviating from the real behavior (e.g. location- and time-dependent humidity and temperatures), the desired dynamic control loop response can be corrected via this input.

Background information

Gear ratio fine adjustment The tension controller generally operates with velocities greater than "0". The gear ratio fine adjustment is an appropriate tool to control the tension.

The gear ratio fine adjustment is internally limited to a minimum of -95%. If a lower value is entered, the function block outputs an error.

Reason for this limitation is the calculated output value defined as follows at a given input value:

$$\text{Output value} = \text{Input value} * \left(1 + \frac{\text{Gear fine adjust}}{100} \right)$$

Fig. 4-116: Output value

It can be seen from this formal description that a gear ratio fine adjustment of less than "-100%" results in an inversion of the direction of rotation.

Troubleshooting The function block generates the following error messages in Additional1/ Additional2 for the table **F_RELATED_TABLE**, 16#0170:

ErrorID	Additional1	Additional2	Description
ACCESS_ERROR (16#0004)	16#00000A00	16#00000001	The parameter P-0-0694 is not parameterized in the optional cyclic telegram of the slave axis
ACCESS_ERROR (16#0004)	16#00000A00	16#00000002	The parameter P-0-0692 is not parameterized in the optional cyclic telegram of the slave axis
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000001	The master axis velocity in the lower operating point ("MasterVelocity-Low") is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000005	The value of the "LowLimit" input is lower than the minimum allowed (-95.0)
DEVICE_ERROR (16#0008)	16#00000A02	16#00000005	The Sercos cycle time (C-0-0503) is less than or equal 0
DEVICE_ERROR (16#0008)	16#00000A02	16#00000006	The gear ratio of the controlled axis is inadmissible (A-0-2720, A-0-2721)
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000002	The value specified at the "Control-Mode" input is not contained in the "MB_TENSION_CONTROL_MODE" data type
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000003	The value specified at the "MaterialParameter.MaterialType" input is not contained in the "MB_TENSION_CONTROL_MATERIAL_TYPE" data type.
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000007	The "StandstillVelDeadband" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000009	The "MaterialParameter.WebWidth" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000010	The "MaterialParameter.WebThickness" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000011	The "MaterialParameter.MaterialE-Modul" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000012	The "MaterialParameter.WebGrammage" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000013	The "DampingFactor" input is less than 20 or higher than 200
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000014	The "Circumference" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000015	The "DistanceRoll1To2" input is less than or equal 0

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	State machine of the function block is in an invalid status
RESOURCE_ERROR (16#0003)	16#00001503	16#00000000	AxisData is not active in the control. This error occurs only on the MLD

Tab. 4-57: Error codes of the MB_TensionControlLoadCellMatParType01 function block

4.8.4 MB_TENSION_CONTROL_MATERIAL

The following table lists the characteristics of the MB_TENSION_CONTROL_MATERIAL structure.

Name	Type	Min. value	Max. value	Default value	Effective	Description
MaterialType	MB_TENSION_CONTROL_MATERIAL_TYPE, page 292	CUSTOM_PAPER	BOPP_FILM	OFFSET_PAPER	Continuous	Type of web material
WebWidth	REAL	>0.0	n.def.	500	Continuous	Width of web material [mm]*
WebThickness	REAL	>0.0 (if MaterialType is not paper)	n.def.	0	Continuous (if "MaterialType" is not paper)	Thickness of web material [μm]*
WebGrammage	REAL	>0.0 (if MaterialType is paper)	n.def.	80	Continuous (if "MaterialType" is paper)	Grammage of web material [g/m²]
MaterialEModule	REAL	>0.0 (if MaterialType is CUSTOM_...)	n.def.	0	Continuous	Modulus of elasticity of web material [N/mm²]*

* The units [mm] and [μm] are obligatory when using the internal material database (E [N/mm²]). When using a user-defined modulus of elasticity, the basic unit is [N/mm²]. However, any units can be used, e.g. [kg/ft²]. Ensure that the selected units are always used consistently.

Tab. 4-58: Structural description of the MB_TENSION_CONTROL_MATERIAL

4.8.5 MB_TENSION_CONTROL_MATERIAL_TYPE

The following table lists the characteristics of the enumeration type MB_TENSION_CONTROL_MATERIAL_TYPE.

Enumeration element	Number	Description
CUSTOM_PAPER	1	User-defined paper
PACKING_PAPER	2	Packing paper
OFFSET_PAPER	3	Offset paper (default)
METALLIZED_PAPER	4	Metallized paper
CUSTOM_NON_PAPER	5	User-defined no paper
PE_FILM	6	Polyethylene foil (PE film)
PET_FILM	7	Polyethylene terephthalate foil (PET film)

Enumeration element	Number	Description
PP_FILM	8	Polypropylene film (PP film)
OPP_FILM	9	Oriented polypropylene film (OPP film)
BOPP_FILM	10	Biaxially oriented polypropylene film (BOPP film)

Tab. 4-59: Enumeration type MB_TENSION_CONTROL_MATERIAL_TYPE

4.8.6 MB_TENSION_CONTROL_MODE

The following table lists the characteristics of the enumeration type MB_TENSION_CONTROL_MODE.

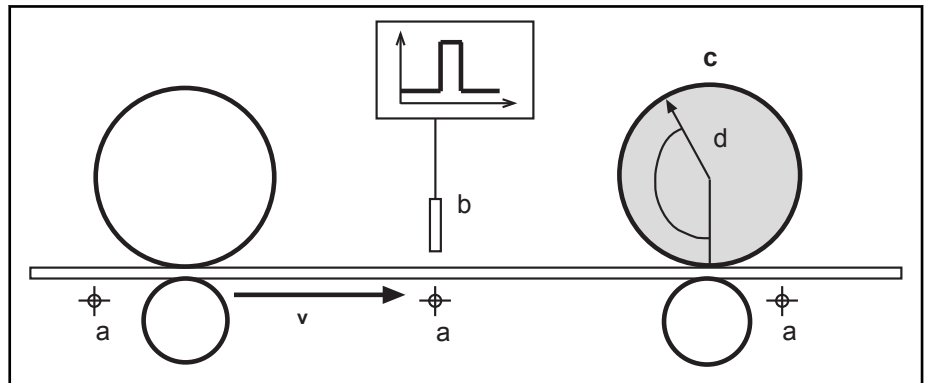
Enumeration element	Number	Description
CONTROL_MODE_OPT_DISTURBANCE	1	The function block optimizes the disturbance response of the "Feedback" value to the "Setpoint" value (default)
CONTROL_MODE_OPT_COMMAND	2	The function block optimizes the command value response of the "Feedback" value to the "Setpoint" value

Tab. 4-60: Enumeration type MB_TENSION_CONTROL_MODE

5 ML_TechRegi.library // MX_TechRegi.lib

5.1 Introduction and overview

Registration marks printed on a paper web or a paper sheet are detected using a digital input and then read in by the register controller as a position measuring value (position measurement). The position offset, velocity of the drive or the master axis drive parameters are adjusted by the register controller function blocks.



- a:** Registration mark
b: Register touch probe sensor
c: Printing cylinders
d: Print ϕ

Fig. 5-1: Measuring principle with web printing

The register controller controls the position of material in terms of the position of a reference axis. This can be the position of a sheet related to the printing cylinder. The closed-loop control is conducted with the help of an electrical input signal at the reference axis. The electrical input signal is provided by the optical measurement of a registration mark on the material.



The library requires the **TEC (technology) function package** of the control system

ML_TechRegi (MLC)/ MX_TechRegi (MLD) function blocks

Area of application (folder in library)			
Function block	MLC	MLD MPx07/08/17 (IL1G)	MLD MPx20 (IL2G)
Editing measured touch probe signals (RegiMeasuring)			
MB_RegiMeasuringType01, page 296	X	X	
MB_RegiMeasuringType02, page 298	X	X	
MB_RegiMeasuringType03, page 301	X	X	X
Quality evaluation of printed images (RegiQuality)			
IL_RegiQualityType01, page 305	X	X	
Write a control value to the drive (RegiRegulate)			

MB_RegiRegulateType01, page 308	X	X	
MB_RegiRegulateType02, page 313	X	X	X
MB_RegiRegulateType03, page 321	X	X	
MB_RegiRegulateType04, page 327	X		
Locking a value for a register controller (RegiSetpointLock)			
MB_SetpointLockType01, page 336	X	X	X
Register controller function block (RegisterController)			
MB_RegisterControllerType02, page 338	X	X	
MB_RegisterControllerType04, page 347	X	X	X
Register controller function block (RegisterControllerSide)			
MB_RegisterControllerSideType01, page 366	X	X	

Tab. 5-1: Overview on function block support by the library

5.2 Measuring function blocks for register controllers

5.2.1 Introduction and overview

This section describes the measuring function blocks developed to create customized register controller function blocks.

5.2.2 MB_RegiMeasuringType01

Brief description The MB_RegiMeasuringType01 function block processes the measured touch probe signal.

Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechRegi
MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-2: Reference table of the MB_RegiMeasuringType01

Interface description

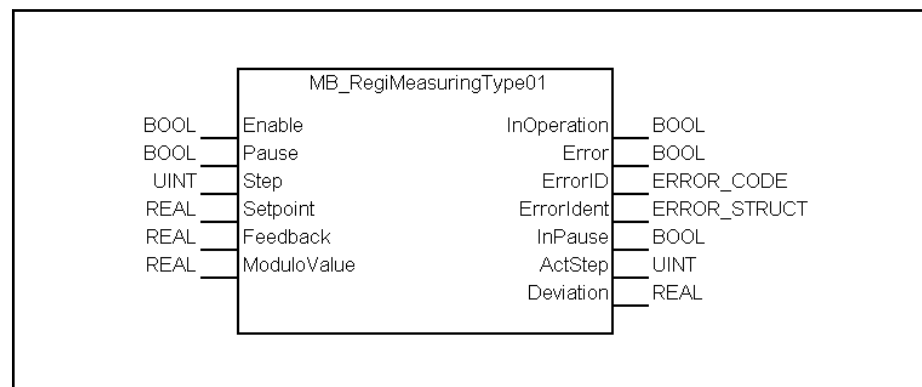


Fig. 5-2: MB_RegiMeasuringType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enables the function block when set to TRUE
	Pause	BOOL	The "Pause" input is only evaluated while the "Enable" input is active
	Step	UINT	New values are calculated when the "Step" input changes
	Setpoint	REAL	Command value
	Feedback	REAL	Measured value
	ModuloValue	REAL	Modulo value of the measured axis
VAR_OUTPUT	InOperation	BOOL	Measuring function block is operating
	Error	BOOL	Indicates that an error occurred during the function block operation
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Error handling, page 298
	InPause	BOOL	"Pause" is active. The "Deviation" output maintains its value
	ActStep	UINT	Step value, incrementing with every new value at the "Deviation" output
	Deviation	REAL	"Setpoint" - "Feedback"

Tab. 5-3: Interface variables of the MB_RegiMeasuringType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Step	UINT	0	65.535	0	Continuous. Starts at a rising edge at "Enable" and at a falling edge at "Pause"
Setpoint	REAL	0	ModuloValue	0.0	Change of "Step"
Feedback	REAL	0	ModuloValue	0.0	Change of "Step"
ModuloValue	REAL	0	n.def.	0.0	Rising edge at "Enable"

Tab. 5-4: Min./max. and default values of the inputs

Timing diagram

Timing diagram according to the PLCopen specification.

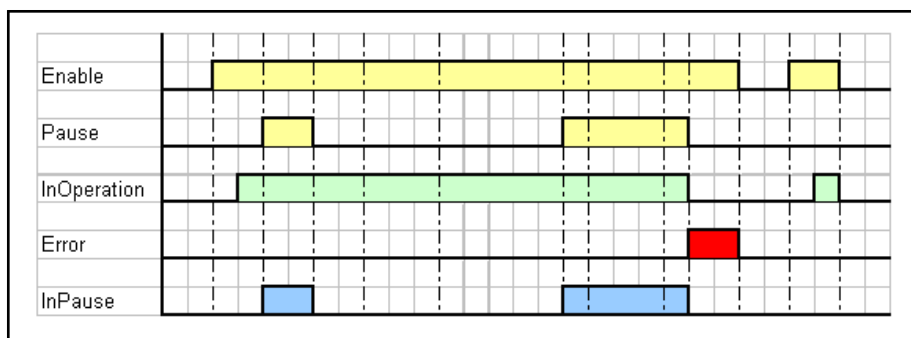


Fig. 5-3: Timing diagram of the MB_RegiMeasuringType01

Functional description The MB_RegiMeasuringType01 function block subtracts the "Setpoint" and "Feedback" inputs ("Setpoint" - "Feedback"). The result of the subtraction is the value of the "Deviation" output.

Pause The "Pause" input is only evaluated as long as the "Enable" input is active. The "ActStep" output is updated even if the "Pause" input is active. While "Pause" is active, the "InPause" output is true and the "Deviation" output maintains its value.

Error handling The function block generates the following error messages in Additional1/Additional2 using the "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis ("Axis_Ref") was changed while the function block was operating
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0002	Input "Setpoint" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0003	Input "Setpoint" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0004	Input "Feedback" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0005	Input "Feedback" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0006	Input "ModuloValue" < 0

Tab. 5-5: Error codes of the MB_RegiMeasuringType01 function block

5.2.3 MB_RegiMeasuringType02

Brief description The MB_RegiMeasuringType02 function block processes the measured touch probe signal.

Target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechRegi
MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-6: Reference table of the MB_RegiMeasuringType02

Interface description

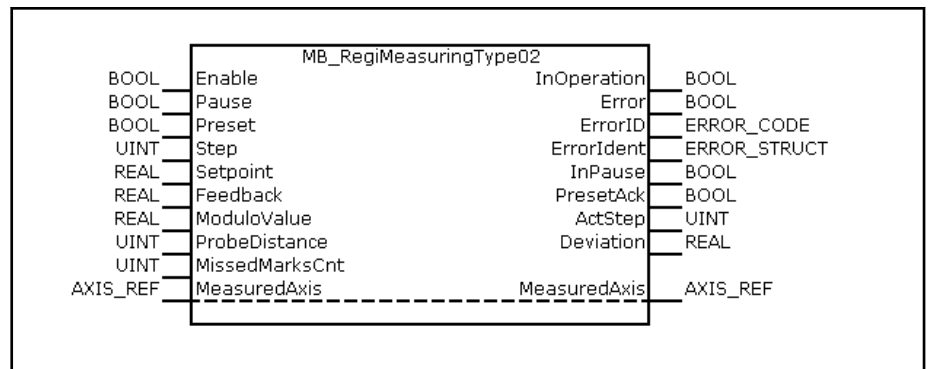


Fig. 5-4: MB_RegiMeasuringType02 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enables the function block when set to TRUE
	Pause	BOOL	The "Pause" input is only evaluated while the "Enable" input is active
	Preset	BOOL	The "Preset" input is only evaluated when "Enable" is TRUE. A positive edge of this input forces the shift register memory to "0".
	Step	UINT	New values are calculated when the "Step" input changes
	Setpoint	REAL	Command value
	Feedback	REAL	Measured value
	ModuloValue	REAL	Modulo value of the measured axis
	ProbeDistance	UINT	The distance between mark sensor and measuring (= controlled) axis in master axis revolutions (format lengths)
	MissedMarksCnt	UINT	Counter for marker failures if an expectation window is used
VAR_OUTPUT	InOperation	BOOL	Measuring function block is operating
	Error	BOOL	Indicates that an error occurred during the function block operation
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 301
	InPause	BOOL	"Pause" is active. The "Deviation" output maintains its value
	PresetAck	BOOL	"Preset" function is done
	ActStep	UINT	Step value, incrementing with every new value on the output "Deviation"
	Deviation	REAL	"Setpoint" - "Feedback" delayed if shift register is active
VAR_IN_OUT CONSTANT	MeasuredAxis	AXIS_REF	Reference to the measured axis

Tab. 5-7: Interface variables of the MB_RegiMeasuringType02 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Preset	BOOL			FALSE	"Enable" TRUE (continuous)
Step	UINT	0	65.535	0	Continuous. Starts at a rising edge at "Enable" and at a falling edge at "Pause"
Setpoint	REAL	0	ModuloValue	0.0	Change of "Step"
Feedback	REAL	0	ModuloValue	0.0	Change of "Step"
ModuloValue	REAL	0	n.def.	0.0	Rising edge at "Enable"
ProbeDistance	UINT	0	30	0	Continuous
MissedMarksCnt	UINT	0	65.535	0	Continuous

Tab. 5-8: Min./max. and default values of the inputs

Time diagram

Timing diagram according to the PLCopen specification.

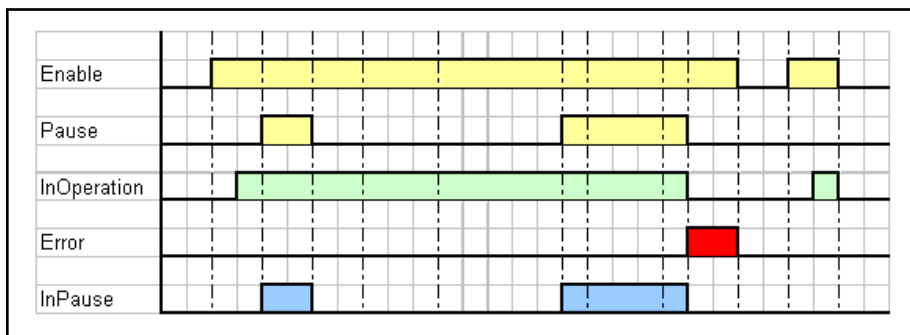


Fig. 5-5: Timing diagram of the MB_RegiMeasuringType02

Functional description

The MB_RegiMeasuringType02 function block subtracts the "Setpoint" and "Feedback" inputs ("Setpoint" - "Feedback"). The result of the subtraction is the value of the "Deviation" output when the shift register is deactivated. If the shift register is activated, it is delayed by the number of "ProbeDistance".

Pause

The "Pause" input is only evaluated as long as the "Enable" input is active. The "ActStep" output is updated even if the "Pause" input is active. While "Pause" is active, the "InPause" output is true and the "Deviation" output maintains its value.

Preset

The shift register can be pre written with the "Preset" input. In order to pre write the shift register memory, "Enable" must be active. A zero value is written to the memory on a rising edge of the "Preset" input if "Enable" is active. If successful, the "PresetAck" output is set to TRUE.

ProbeDistance

The "ProbeDistance" input is the distance between the mark sensor and the measuring (= controlled) axis in master axis revolutions (format lengths). The maximum number of format length is 30. If "ProbeDistance" is 0, the shift register is deactivated. The "Deviation" output is set to "Setpoint" - "Feedback".

If "ProbeDistance" is not 0, the shift register is activated. This means that the values at the "Deviation" output are delayed by the number of master axis revolutions (format lengths) on the "ProbeDistance" input.

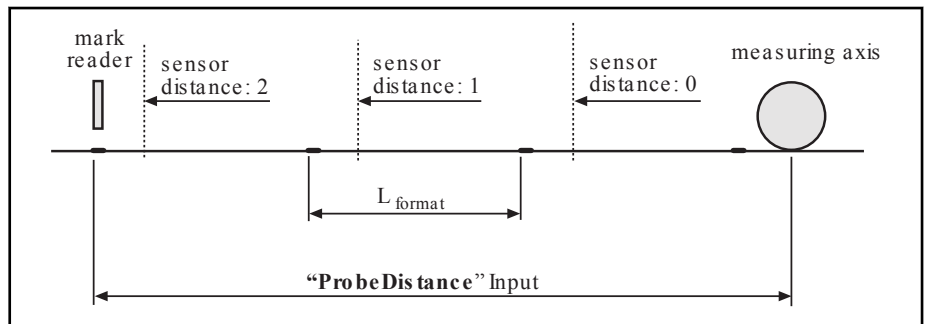


Fig. 5-6: Setting the "ProbeDistance" input

MissedMarkCnt This is the marker failure monitoring counter from the touch probe function when the expectation window is enabled. If a mark is missed, a zero is written to the shift register if activated. In the event of a deactivated shifting register, the zero is written to the "Deviation" output.

Troubleshooting The function block generates the following error messages in Additional1/ Additional2 using the "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis ("Axis_Ref") was changed while the function block was operating
RESOURCE_ERROR (16#0003)	16#0012	16#0003	"AXIS_REF.CntrlNo" is not in the valid range (axes of remote controls are invalid)
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0001	"ProbeDistance" input is not in the valid range ($0 \leq \text{"ProbeDistance"} \leq 30$)
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0002	Input "Setpoint" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0003	Input "Setpoint" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0004	Input "Feedback" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0005	Input "Feedback" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0006	Input "ModuloValue" < 0

Tab. 5-9: Error codes of the MB_RegiMeasuringType02 function block

5.2.4 MB_RegiMeasuringType03

Brief description The MB_RegiMeasuringType03 function block processes the measured touch probe signal.

Target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechRegi

MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib (IL1G)
MLD with MPx20 with MA function package	MX_TechRegi (IL2G)

Tab. 5-10: Reference table of the MB_RegiMeasuringType03

Interface description

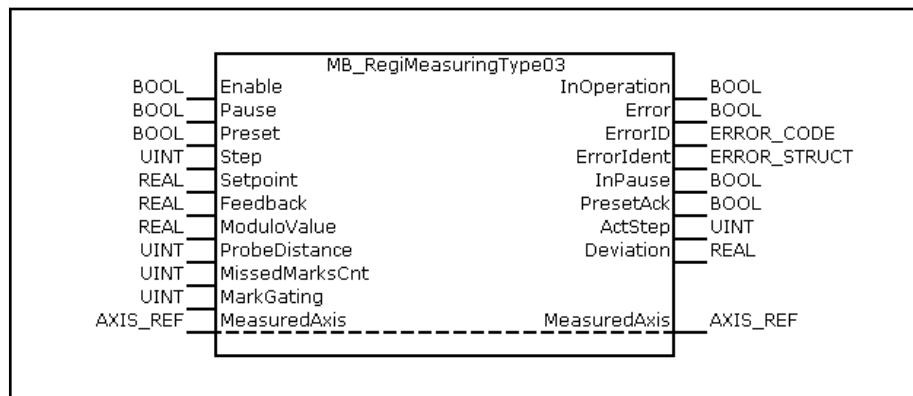


Fig. 5-7: MB_RegiMeasuringType03 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enables the function block when set to TRUE
	Pause	BOOL	The "Pause" input is only evaluated while the "Enable" input is active
	Preset	BOOL	The "Preset" input is only evaluated when "Enable" is TRUE. A positive edge of this input forces the shift register memory to "0".
	Step	UINT	New values are calculated when the "Step" input changes
	Setpoint	REAL	Command value
	Feedback	REAL	Measured value
	ModuloValue	REAL	Modulo value of the measured axis
	ProbeDistance	UINT	The distance between the mark sensor and the measuring axis in master axis revolutions (format lengths)
	MissedMarksCnt	UINT	Missed mark counter if using a expectation window
	MarkGating	UINT	Every nth mark is used for registration
VAR_OUTPUT	InOperation	BOOL	Measuring function block is operating
	Error	BOOL	Indicates that an error occurred during the function block operation
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 305
	InPause	BOOL	"Pause" is active. The "Deviation" output maintains its value
	PresetAck	BOOL	"Preset" function is done

I/O type	Name	Data type	Description
	ActStep	UINT	Step value, incrementing with every new value at the "Deviation" output
	Deviation	REAL	"Setpoint" - "Feedback" delayed if shift register is active
VAR_IN_OUT CONSTANT	MeasuredAxis	AXIS_REF	Reference to the measured axis

Tab. 5-11: Interface variables of the MB_RegiMeasuringType03 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Preset	BOOL			FALSE	"Enable" TRUE (continuous)
Step	UINT	0	65.535	0	Continuous Starts at a rising edge at "Enable" and at a falling edge at "Pause"
Setpoint	REAL	0	ModuloValue	0.0	Change of "Step"
Feedback	REAL	0	ModuloValue	0.0	Continuous
ModuloValue	REAL	0	N. def.	0.0	Rising edge at "Enable"
ProbeDistance	UINT	0	60	0	Continuous
MissedMarksCnt	UINT	0	65.535	0	Continuous
MarkGating	UNIT	0	65.535	1	Rising edge at "Enable". "0" disables mark gating (identical to "1")

Tab. 5-12: Min./max. and default values of the inputs

Time diagram

Timing diagram according to the PLCopen specification.

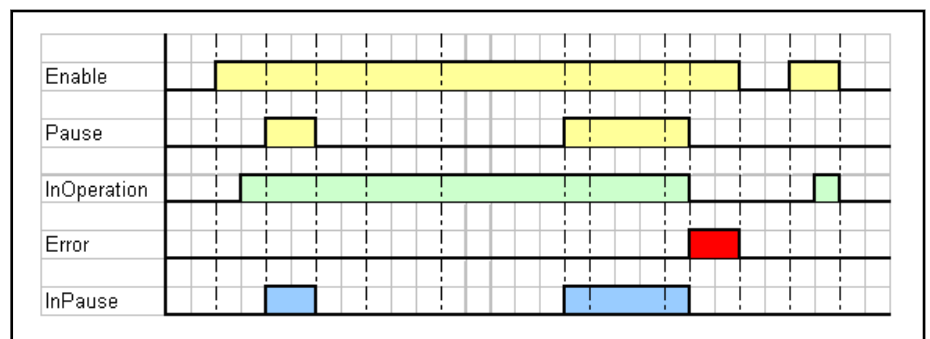


Fig. 5-8: Timing diagram of the MB_RegiMeasuringType03

Functional description

The MB_RegiMeasuringType03 function block subtracts the "Setpoint" and "Feedback" inputs ("Setpoint" - "Feedback"). The result of the subtraction is the value of the "Deviation" output when the shift register is deactivated. If the

shift register is activated, it is delayed by the number of "ProbeDistance". Only every nth mark is for the registration with the mark gating feature of the function block.

Pause The "Pause" input is only evaluated as long as the "Enable" input is active. The "ActStep" output is updated even if the "Pause" input is active. While "Pause" is active, the "InPause" output is true and the "Deviation" output maintains its value.

Preset The shift register can be pre written with the "Preset" input. In order to pre write the shift register memory, "Enable" must be active. A zero value is written to the memory on a rising edge of the "Preset" input if "Enable" is active. If successful, the "PresetAck" output is set to TRUE.

ProbeDistance The "ProbeDistance" input is the distance between the mark sensor and the measuring (= controlled) axis in master axis revolutions (format lengths). The maximum number of format length is 60. If "ProbeDistance" is 0, the shift register is deactivated. The "Deviation" output is set to "Setpoint" – "Feedback".

If "ProbeDistance" is not 0, the shift register is activated. This means that the values at the "Deviation" output are delayed by the number of master axis revolutions (format lengths) on the "ProbeDistance" input.

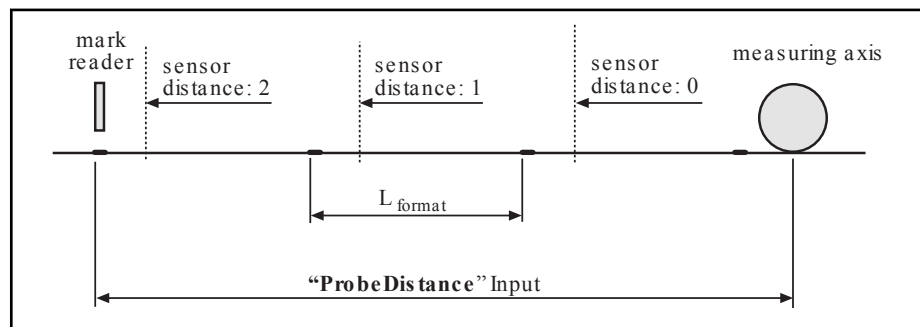


Fig. 5-9: Setting the "ProbeDistance" input



When using the "ProbeDistance", the previously entered deviations into the shift register have to be adjusted each by 100% (P_Control = 1 / I_Control = 0).

That means: The new deviation entered is only the delta of the sum already entered into the shift register.

MissedMarkCnt This is the marker failure monitoring counter from the touch probe function when the expectation window is enabled. If a mark is missed, a zero is written to the shift register if activated. In the event of a deactivated shifting register, the zero is written to the "Deviation" output.

MarkGating This feature allows the user to control the registration only every nth mark by gating invalid marks. This can be used e.g. if the marks on the web were printed with a printing plate where the printed image is displayed several times. In this case, the mark distances between subsequent marks is not constant. For example, the mark distances between every nth consecutive mark is constant. The RegiMeasuring function block only uses every nth mark for the register control.

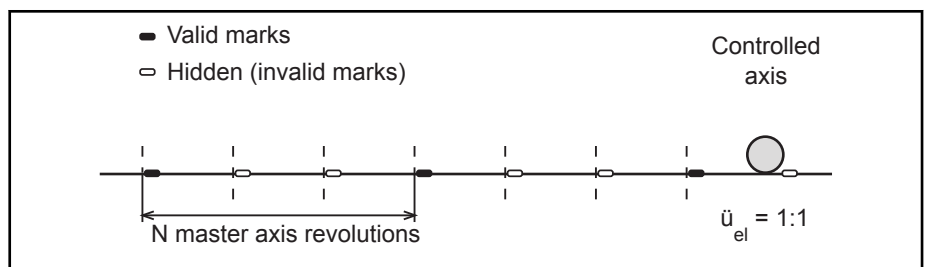


Fig. 5-10: Mark gating

Troubleshooting The function block generates the following error messages in Additional1/ Additional2 using the "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis ("Axis_Ref") was changed while the function block was operating
RESOURCE_ERROR (16#0003)	16#0012	16#0003	"AXIS_REF.CntrlNo" is not in the valid range (axes of remote controls are invalid)
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0001	"ProbeDistance" input is not in the valid range ($0 \leq \text{"ProbeDistance"} \leq 30$)
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0002	Input "Setpoint" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0003	Input "Setpoint" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0004	Input "Feedback" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0005	Input "Feedback" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0006	Input "ModuloValue" < 0
INPUT_RANGE_ERROR (16#0006)	16#0814	16#0007	Input "MarkGating" < 0

Tab. 5-13: Error codes of the MB_RegiMeasuringType03 function block

5.3 Register quality function block

5.3.1 Introduction and overview

The register quality function block is part of the libraries ML_TechRegi (MLC) and MX_TechRegi.lib (MLD).

5.3.2 IL_RegiQualityType01

Brief description This function block is event-triggered and evaluates up to 16 register deviations. With two limit values, the printed images are divided into "good, medium and bad" quality and summed up. The quality of the printed image depends on the color mark.

Target system	Libraries
MLC 14VRS TEC function package	ML_TechRegi
MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-14: Reference table of the IL_RegiQualityType01

Interface description

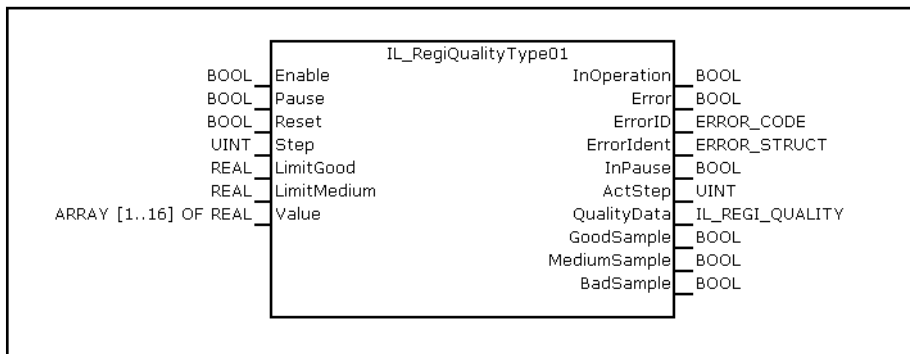


Fig. 5-11: IL_RegiQualityType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Calculation of data in each cycle when "Enable" = TRUE
	Pause	BOOL	The output data is not changed when Pause = TRUE
	Reset	BOOL	Function block reset
	Step	UINT	Current calculation step
	LimitGood	REAL	Max. absolute value [mm] of the register deviation to classify the quality of the printed image as "good"
	LimitedMedium	REAL	If the absolute value of the register deviation [mm] is between "LimitGood" and "LimitMedium", the printed image is classified as "medium" quality. A register deviation greater than "LimitMedium" is considered as "bad" quality
	Value	ARRAY [1..16] OF REAL	It contains register deviations of a printed image (up to 16 register marks)
VAR_OUTPUT	InOperation	BOOL	Processing without errors, output data is valid
	Error	BOOL	Error (see ErrorID and ErrorIdent)
	ErrorID	ERROR_CODE	Error Description
	ErrorIdent	ERROR_STRUCT	Detailed error description Error codes, page 308
	InPause	BOOL	Indicates that the "Pause" input is enabled and that the values do not change at the outputs
	ActStep	UINT	Last processed step
	QualityData	IL_REGI_QUALITY	Contains the calculated quality data for the printing process IL_REGI_QUALITY, page 308

I/O type	Name	Data type	Description
	GoodSample	BOOL	Is TRUE if the current printed image is of "good quality"
	MediumSample	BOOL	Is TRUE if the current printed image is of "medium" quality
	BadSample	BOOL	Is TRUE if the current printed image is of "bad quality"

Tab. 5-15: Interface variables of the IL_RegiQualityType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Reset	BOOL			FALSE	Continuous
Step	UINT	0	65535	0	Continuous
LimitGood	REAL	0.0	LimitMedium	1.0	Rising edge at "Enable"
LimitMedium	REAL	LimitGood	n.def	2.0	Rising edge at "Enable"
Value	ARRAY [1..16] OF REAL	n.def	n.def	0.0	If "Step" is changed

Tab. 5-16: Min./max. and default values of the IL_RegiQualityType01 inputs

Functional description

The register errors of the printed image are an important characteristic for the quality of the printing process. The largest register error of a color mark is the most important criteria for the quality of a printed image.

Based on the register deviation, the printed image is subdivided into three quality categories ("Good", "Medium", "Bad"). The quality is assessed with the inputs "LimitGood" and "LimitMedium". If the inputs "LimitGood" and "LimitMedium" are equal, the printed image is only subdivided into two quality categories ("Good", "Bad"). If the same value is specified for "LimitGood" and "LimitMedium", it is only classified in "good" and "bad" (refer to the following figure).

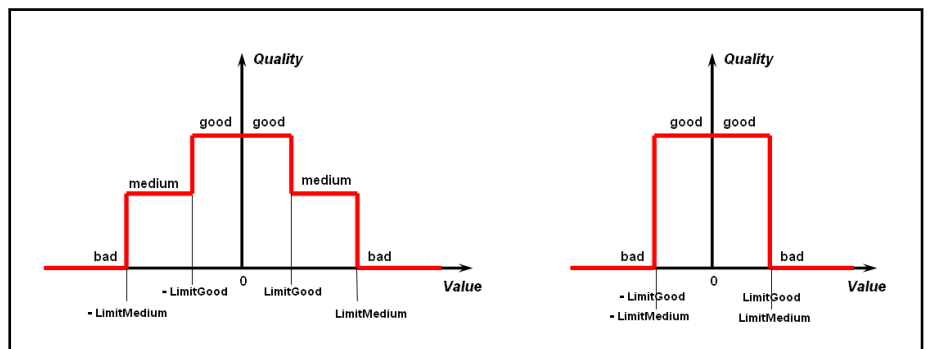


Fig. 5-12: Quality assessment principle for the IL_RegiQualityType01

The output data structure "QualityData" sums up the printed images of the three quality categories. The structure also contains the statistical percentages for each quality category. All values are set to "0" at a rising edge of the "Reset" input.

It is processed continuously if the **Enable** input is set. The "ActStep" output indicates that a new data package with measured values was received and evaluated. The Boolean outputs "Good/Medium/BadSample" indicate additionally the quality of the current printed image. If the "Pause" input is set, the current state is frozen, the "InPause" output is set and there is no further evaluation.

Error codes The function block uses the error table **F_RELATED_TABLE**. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
INPUT_INVALID_ERROR	16#0817	16#0001	"LimitGood" value < 0.0
INPUT_INVALID_ERROR	16#0817	16#0002	"LimitMedium" value < "LimitGood"

Tab. 5-17: Error codes of the IL_RegiQualityType01 function block

5.3.3 IL_REGI_QUALITY

The "IL_REGI_QUALITY" structure makes it possible to record the number and the percentage of good, average, and poor quality.

This structure supports the function block [IL_RegiQualityType01](#), page 305.

Name	Type	Description
GoodSampleCount	UDINT	Number of printed images of "good" quality
MediumSampleCount	UDINT	Number of printed images of "medium" quality
BadSampleCount	UDINT	Number of printed images of "bad" quality
GoodSampleRatio	REAL	Percentage of printed images with "good" quality
MediumSampleRatio	REAL	Percentage of printed images with "medium" quality
BadSampleRatio	REAL	Percentage of printed images with "bad" quality

Tab. 5-18: Definition of the IL_REGI_QUALITY structure

5.4 Regulating function blocks for register controllers

5.4.1 Introduction and overview

This section describes how the function blocks writing a control value to the drive can be used to create customized register controller function blocks.

5.4.2 MB_RegiRegulateType01

Brief description The MB_RegiRegulateType01 function block writes a control value to the drive.

Target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechRegi
MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-19: Reference table of the MB_RegiRegulateType01

Interface description

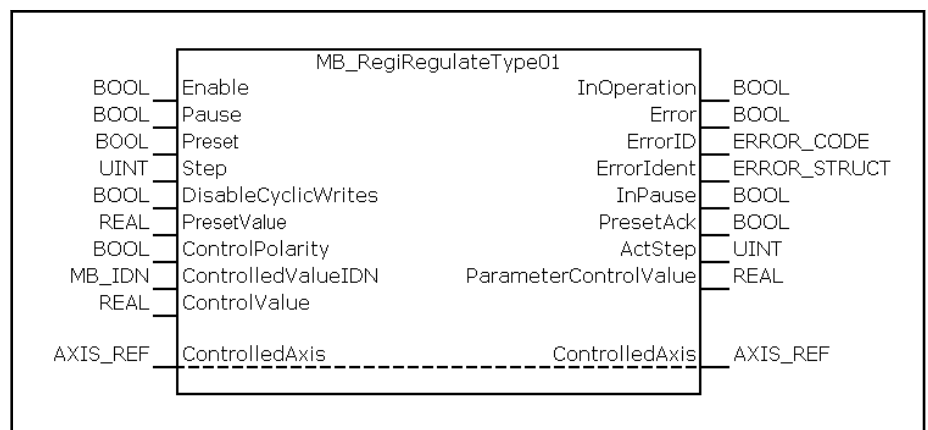


Fig. 5-13: MB_RegiRegulateType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enables the function block when set to TRUE
	Pause	BOOL	The "Pause" input is only evaluated while the "Enable" input is active
	Preset	BOOL	The "Preset" input is only evaluated when "Enable" is TRUE. A positive edge of this input forces "ParameterControlValue" to "PresetValue"
	Step	UINT	Processing step
	DisableCyclicWrites	BOOL	Disables cyclic writing to the selected parameter in "ControlledValueIDN" if TRUE
	PresetValue	REAL	Value transferred to the "ParameterControlValue" output at a rising edge of the "Preset" input
	ControlPolarity	BOOL	The polarity of the control value is inverted when this input is TRUE
	ControlledValueIDN	MB_IDN	Sercos IDN of the controlled parameter (e.g. P-0-0691). The following IDNs are supported: IndraMotion MLD: P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 IndraMotion 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	Calculated control value from IL_PIType02 for example
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Indicates an error. Clear error with "Enable" = FALSE
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 312
	InPause	BOOL	"Pause" is active. The "ParameterControlValue" output maintains its value

I/O type	Name	Data type	Description
	PresetAck	BOOL	"Preset" is done. The "Preset" value is copied to the "ParameterControlValue" output
	ActStep	UINT	Step value, incrementing with every new value at the output "ParameterControlValue"
	ParameterControlValue	REAL	The register controller function block calculates a value based on the selected parameter in the input "ControlledValueIDN" and stores the result in the output "ParameterControlValue". This value is written to the selected parameter input "ControlledValueIDN" when the input "DisableCyclicWrites" = FALSE
VAR_IN_OUT CONSTANT	ControlledAxis	AXIS_REF	Reference to the controlled axis

Tab. 5-20: Interface variables of the MB_RegiRegulateType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Preset	BOOL			FALSE	Continuous
Step	UINT	0	65.535	0	Continuous. Starts at a rising edge at "Enable" and at a falling edge at "Pause"
DisableCyclicWrites	BOOL			FALSE	Rising edge at "Enable"
PresetValue	REAL	N. def.	N. def.	0.0	Rising edge at "Preset"
ControlPolarity	BOOL			FALSE	Continuous
ControlledValueIDN	MB_IDN	N. def.	N. def.	0	Rising edge at "Enable"
ControlValue	REAL	N. def.	N. def.	0.0	Continuous
ModuloValue	REAL	0.0	N. def.	0.0	Rising edge at "Enable"

Tab. 5-21: Overview on the min./max. values and default values of the inputs

Time diagram

Timing diagram according to the PLCopen specification.

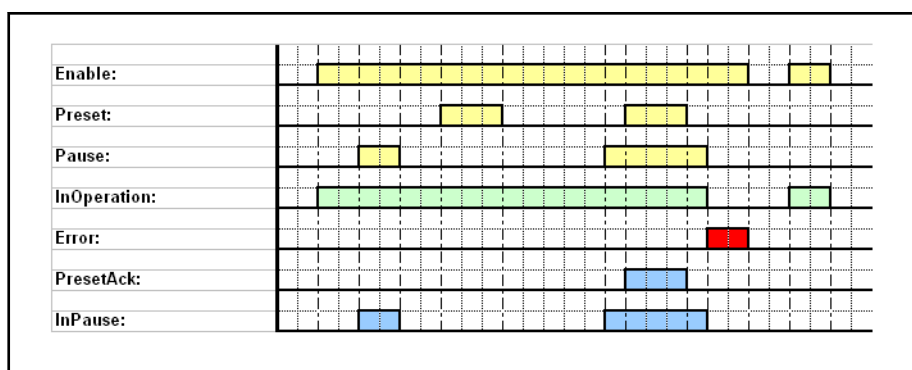


Fig. 5-14: Timing diagram of the MB_RegiRegulateType01

Functional description

The function block writes control values to a parameter in an axis. Depending on the target parameter, "ControlledValueIDN", the function block includes an integration of the "ParameterControlValue" input. The following "ControlledValueIDN" values are integrated:

- 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 The "Pause" input is only evaluated as long as the "Enable" input is active. The "ActStep" output is updated even if the "Pause" input is active. As long as "Pause" is active, the "InPause" output is TRUE and the "ParameterControlValue" output maintains its value.

Preset The "PresetAck" output is active if the value in "PresetValue" is taken into account. The "Preset" input has priority over "Pause". The value in "PresetValue" is taken into account on a rising edge of "Preset" if the "Enable" input is active. If successful, the "PresetAck" output is set TRUE and the value in "PresetValue" is written to the controlled parameter.

DisableCyclicWrites Disables cyclic writing to the controlled parameter (Input "ControlledValueIDN") from the function block.

Normal mode The "RegiRegulateType" only calculates a new value when the "Step" input changes. The controlled axis has to be a modulo axis.

Required parameterization**MLC parameterization**

If a **real axis with interpolation in the drive** is used as the controlled axis ("ControlledAxis" input), the parameter used in the "ControlledValueIDN" input has to be one of the following supported P-parameters:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 and P-0-0695



Additionally, this P-parameter has to be configured in the MDT of the cyclic data channel in the drive.

If a **real axis with interpolation in the control** is used as a **virtual axis**, the parameter used in the "ControlledValueIDN" input has to be one of the following supported P-parameters:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

MLD parameterization

The parameter used as controlled parameter for the "ControlledValueIDN" (P-parameter) input has to be configured in the "AxisData" structure. Thus,

the axis data structure has to be activated. The following P-parameters are supported:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

For the MLD firmware MPx20 (IL2G) and later, axis references which are not included in the "AxisData" structure are also supported at the "MeasuredAxis" (> VMA1).

In this case, the correction window is not determined using the current position of "MeasuredAxis", but via P-0-0761 "Master axis position for slave axis" of the local axis.

To provide the required position information in P-0-0761 "Master axis position for slave axis", the master axis generator has to be configured respectively.

Parameters involved:

- P-0-0760 "Virtual master axis, position control word"
- P-0-0916 "Master axis format converter signal selection"

See "Virtual master axis generator" in the documentation IndraDrive MPx-20 Functions, Application Description, DOK-INDRV*-MP*-20VRS*-APRS-EN-P.

Troubleshooting The function block generates the following error messages in Additional1/ Additional2 using the "F_RELATED_TABLE", 16#0170:
INPUT_RANGE_ERROR (16#0006)

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Drive is not enabled or drive error
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis ("AXIS_REF") was changed while the function block was operating
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"AXIS_REF" is not in the valid range
RESOURCE_ERROR (16#0003)	16#0012	16#0003	"AXIS_REF.CntrlNo" is not in the valid range (axes of remote controls are invalid)
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000A	S-0-0103 or A-0-0045 of the controlled axis is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000B	S-0-0091 or A-0-0032 of the controlled axis is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000C	P-0-0750 of the controlled axis is not in the valid range
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0001	Selected "ControlledValueIDN" is not supported
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00A0	The type of the parameter "ControlledValueIDN" to be controlled does not match the controlled axis (A-parameter for the axis with interpolation in the drive)

INPUT_INVALID_ERROR (16#0001)	16#0812	16#00B0	The type of the parameter "ControlledValueIDN" to be controlled does not match the controlled axis (P-parameter for the axis with interpolation in the control)
ACCESS_ERROR (16#0003)	16#0815	16#0002	Required parameter P-0-0061 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0003)	16#0815	16#0003	Required parameter P-0-0690 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0003)	16#0815	16#0004	Required parameter P-0-0691 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0003)	16#0815	16#0005	Required parameter P-0-0692 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0003)	16#0815	16#0006	Required parameter P-0-0694 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0003)	16#0815	16#0007	Required parameter P-0-0695 is not configured in the optional cyclic MDT data of the controlled axis

Tab. 5-22: Error codes of the MB_RegiRegulateType01 function block

5.4.3 MB_RegiRegulateType02

Brief description The MB_RegiRegulateType02 function block writes a control value to the drive. If the correction window is active, the control deviations are corrected in one parameterized correction window.

Target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechRegi
MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib (IL1G)
MLD with MPx20 with MA function package	MX_TechRegi (IL2G)

Tab. 5-23: Reference table of the MB_RegiRegulateType02

Interface description

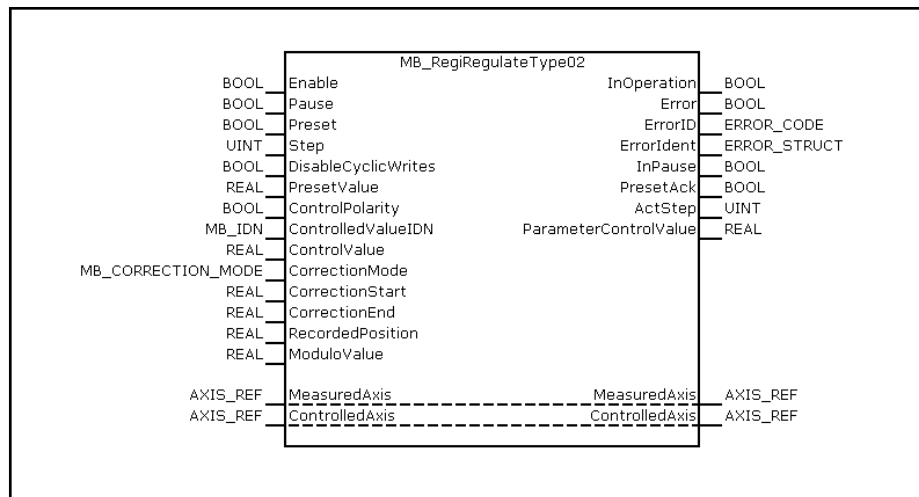


Fig. 5-15: MB_RegiRegulateType02 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enables the function block when set to TRUE
	Pause	BOOL	The "Pause" input is only evaluated while the "Enable" input is active
	Preset	BOOL	The "Preset" input is only evaluated when "Enable" is TRUE. A positive edge of this input forces "ParameterControlValue" to "PresetValue"
	Step	UINT	Processing step
	DisableCyclicWrites	BOOL	Disables cyclic writing to the selected parameter in "ControlledValueIDN" if TRUE
	PresetValue	REAL	Value transferred to the "ParameterControlValue" output at a rising edge of the "Preset" input
	ControlPolarity	BOOL	The polarity of the control value is inverted when this input is TRUE
	ControlledValueIDN	MB_IDN	Sercos IDN of the controlled parameter (e.g. P-0-0691). The following IDNs are supported: 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	Calculated control value from IL_PIType02 for example
	CorrectionMode	MB_CORRECTION_MODE	Properties of the correction window MB_CORRECTION_MODE, page 335
	CorrectionStart	REAL	Start of the correction window
	CorrectionEnd	REAL	End of the correction window
	RecordedPosition	REAL	Measured value
	ModuloValue	REAL	Modulo value of the measured axis
VAR_OUTPUT	InOperation	BOOL	Function block is operating

I/O type	Name	Data type	Description
	Error	BOOL	Indicates an error. Clear error with "Enable" = FALSE
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 319
	InPause	BOOL	"Pause" is active. The "ParameterControlValue" output maintains its value
	PresetAck	BOOL	"Preset" is done. The "Preset" value is copied to the "ParameterControlValue" output
	ActStep	UINT	Step value, incrementing with every new value at the output "ParameterControlValue"
	ParameterControlValue	REAL	The register controller function block calculates a value based on the selected parameter in the input "ControlledValueIDN" and stores the result in the output "ParameterControlValue". This value is written to the selected parameter input "ControlledValueIDN" when the "DisableCyclicWrites" input = FALSE
VAR_IN_OUT CONSTANT	MeasuredAxis	AXIS_REF	Reference to the measured axis
	ControlledAxis	AXIS_REF	Reference to the controlled axis

Tab. 5-24: Interface variables of the MB_RegiRegulateType02 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Preset	BOOL			FALSE	Continuous
Step	UINT	0	65.535	0	Continuous. Starts at a rising edge at "Enable" and at a falling edge at "Pause"
DisableCyclicWrites	BOOL			FALSE	Rising edge at "Enable"
PresetValue	REAL	N. def.	N. def.	0.0	Rising edge at "Preset"
ControlPolarity	BOOL			FALSE	Continuous
ControlledValueIDN	MB_IDN	N. def.	N. def.	0	Rising edge at "Enable"
ControlValue	REAL	N. def.	N. def.	0.0	Continuous
CorrectionMode	MB_CORRECTION_MODE	0	4	CORR_REL	Continuous
CorrectionStart	REAL	0.0	ModuloValue	0.0	Continuous

Name	Type	Min. value	Max. value	Default value	Effective
CorrectionEnd	REAL	0.0	ModuloValue	0.0	Continuous
RecordedPosition	REAL	0.0	ModuloValue	0.0	Change of the "Step" input
ModuloValue	REAL	0.0	N. def.	0.0	Rising edge at "Enable"

Tab. 5-25: Min./max. and default values of the inputs

Time diagram Timing diagram according to the PLCopen specification.

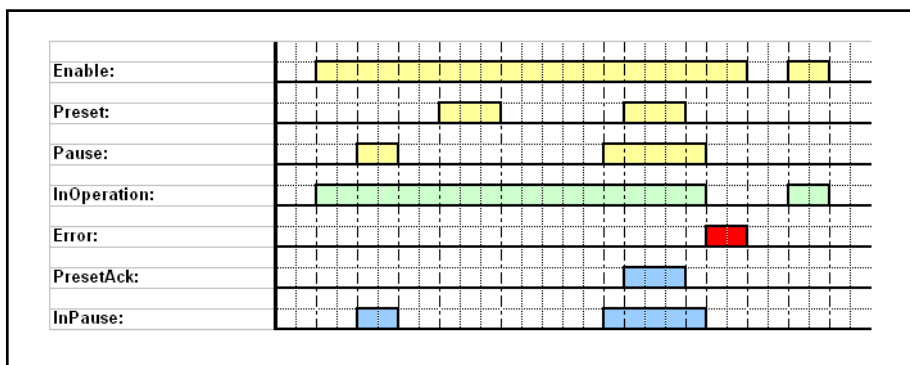


Fig. 5-16: Timing diagram of the MB_RegiRegulateType02

Functional description

The function block writes control values to a parameter in an axis. Depending on the target parameter, "ControlledValueIDN", the function block includes an integration of the "ParameterControlValue" input. The following "ControlledValueIDN" values are integrated:

- 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 The "Pause" input is only evaluated as long as the "Enable" input is active. The "ActStep" output is updated even if the "Pause" input is active. As long as "Pause" is active, the "InPause" output is TRUE and the "ParameterControlValue" output maintains its value.

Preset The "PresetAck" output is active if the value in "PresetValue" is taken into account. The "Preset" input has priority over "Pause". The value in "PresetValue" is taken into account on a rising edge of "Preset" if the "Enable" input is active. If successful, the "PresetAck" output is set TRUE and the value in "PresetValue" is written to the controlled parameter.

DisableCyclicWrites Disables cyclic writing to the controlled parameter (Input "ControlledValueIDN") from the function block.

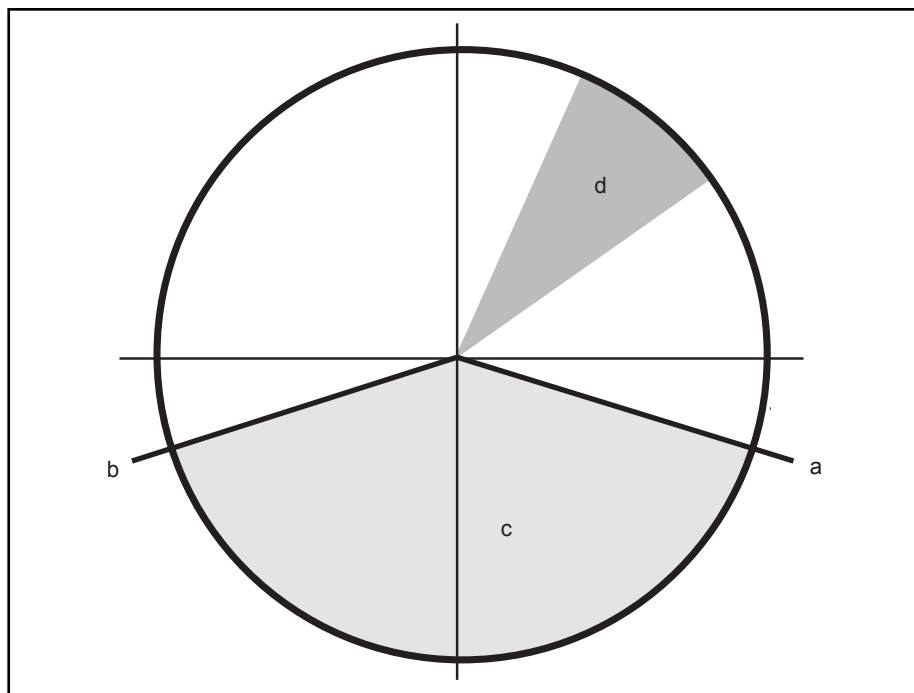
Correction window The correction window is an angular range relating to the measuring axis. The register controller controls the control deviations within this window. The position and size of the correction window are set with the help of the following inputs:

The "CorrectionMode" input sets the correction window either absolute or relative. If the "CorrectionMode" is set to CORR_WIN_DISABLED, the correction window is inactive.

The "CorrectionStart" input defines the start of the correction window.

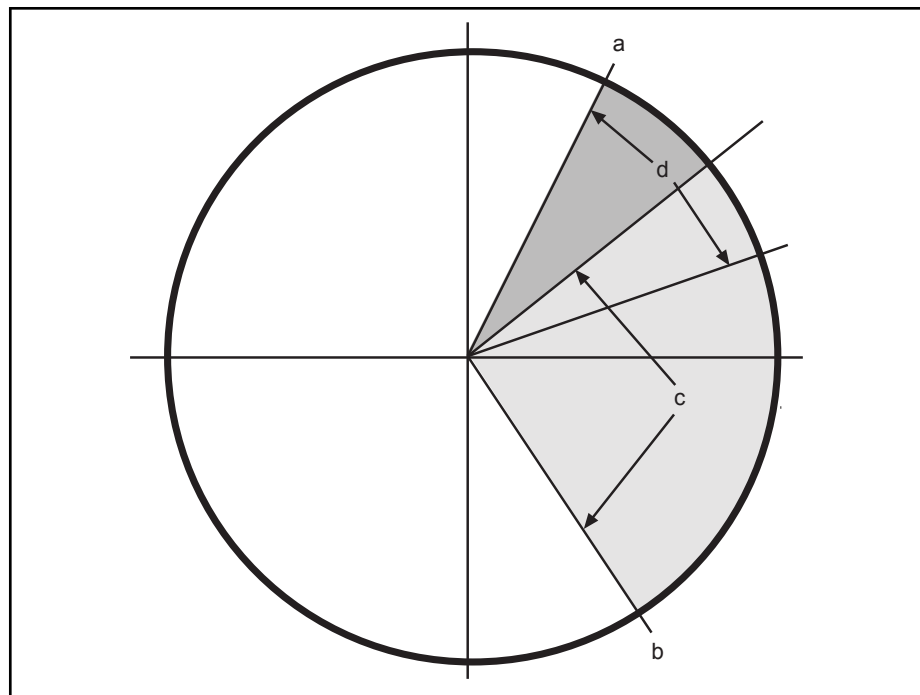
The "CorrectionEnd" input defines the end of the correction window.

The correction window supports only a positive modulo range (e.g., 0...+360 degrees) of the measuring axis.



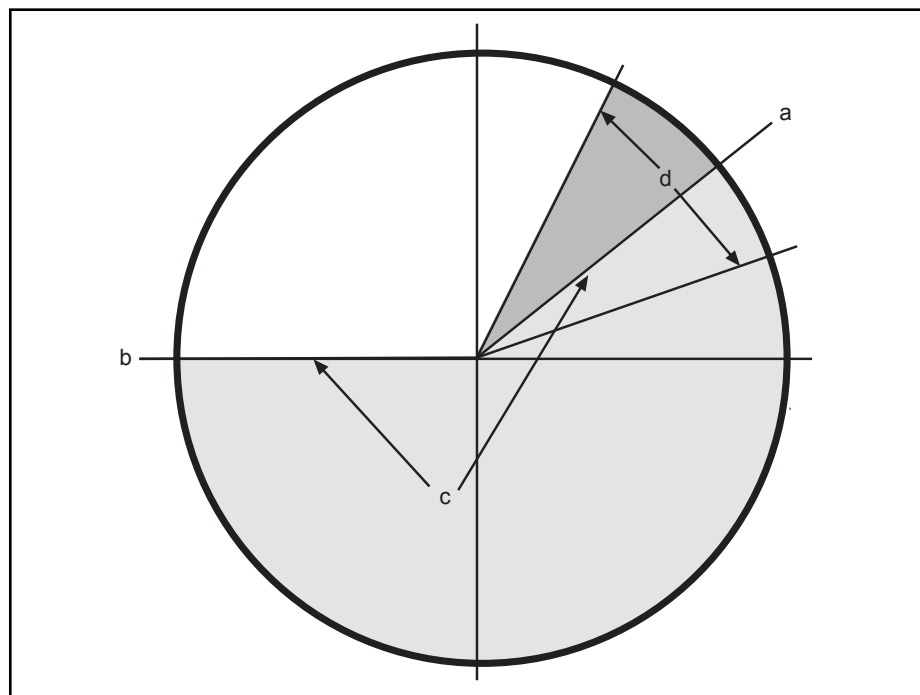
- a:** Start correction window
- b:** End correction window
- c:** Correction window
- d:** Expectation window

Fig. 5-17: Correction window absolute



- a:** Start correction window = measuring end
b: Relative end of correction window to measuring end
c: Correction window
d: Expectation window

Fig. 5-18: Correction window relative



- a:** Start correction window = measuring end
b: End of correction window = perm. angle
c: Correction window
d: Expectation window

Fig. 5-19: Correction window relative start absolute end

Normal mode	The RegiRegulateType02 function block only calculates a new value if the "Step" input changes. The controlled and measured axis must be modulo axes.
Required parameterization	MLC parameterization If a real axis with interpolation in the drive is used as the controlled axis ("ControlledAxis" input), the parameter used in the "ControlledValueIDN" input has to be one of the following supported P-parameters: • P-0-0690, P-0-0691, P-0-0692, P-0-0694 and P-0-0695  Additionally, this P-parameter has to be configured in the MDT of the cyclic data channel in the drive. If a real axis with interpolation in the control is used as a virtual axis, the parameter used in the "ControlledValueIDN" input has to be one of the following supported P-parameters: • A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620 MLD parameterization The parameter used as controlled parameter for the "ControlledValueIDN" (P-parameter) input has to be configured in the "AxisData" structure. Thus, the axis data structure has to be activated. The following P-parameters are supported: • P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695 For the MLD firmware MPx20 (IL2G) and later, axis references which are not included in the "AxisData" structure are also supported at the "MeasuredAxis" (> VMA1). In this case, the correction window is not determined using the current position of "MeasuredAxis", but via P-0-0761 "Master axis position for slave axis" of the local axis. To provide the required position information in P-0-0761 "Master axis position for slave axis", the master axis generator has to be configured respectively. Parameters involved: • P-0-0760 "Virtual master axis, position control word" • P-0-0916 "Master axis format converter signal selection" See "Virtual master axis generator" in the documentation IndraDrive MPx-20 Functions, Application Description, DOK-INDRV*-MP*-20VRS*-APRS-EN-P.
Data type MB_CORRECTION_MODE	Modes of the correction window. <i>Program:</i> <pre data-bbox="596 1617 1442 1814"> TYPE MB_CORRECTION_MODE: (CORR_WIN_DISABLED :=0, (* correction window disabled *) CORR_ABS :=1, (* correction window is absolute *) CORR_REL :=2, (* correction window is relative *) CORR_REL_ABS :=3, (* correction window is relative start and absolute end *) CORR_ABS_REL :=4); (* correction window is absolute start and relative end *) END_TYPE </pre>
Troubleshooting	The function block generates the following error messages in Additional1/Additional2 using the "F_RELATED_TABLE", 16#0170: INPUT_RANGE_ERROR (16#0006)

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Drive is not enabled or drive error
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis ("AXIS_REF") was changed while the function block was operating
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"AXIS_REF" is not in the valid range
RESOURCE_ERROR (16#0003)	16#0012	16#0003	"AXIS_REF.CntrlNo" is not in the valid range (axes of remote controls are invalid)
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0001	Length of the correction window < 0.25% of the value at the "Modulo-Value" input
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0002	Input "CorrectionStart" < 0
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0003	Input "CorrectionEnd" < 0
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0004	Value < 0 input "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0005	Value < 0 input "RecordedPosition"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0006	Input "CorrectionStart" > "Modulo-Value"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0007	Input "CorrectionEnd" > "Modulo-Value"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0008	Input "RecordedPosition" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0009	Correction path > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000A	S-0-0103 or A-0-0045 of the controlled axis is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000B	S-0-0091 or A-0-0032 of the controlled axis is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000C	P-0-0750 of the controlled axis is not in the valid range
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0001	Selected "ControlledValueIDN" is not supported
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0002	Input "CorrectionMode" is invalid
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0003	"CorrectionStart" = "CorrectionEnd"
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00A0	The type of the parameter "ControlledValueIDN" to be controlled does not match the controlled axis (A-parameter for the axis with interpolation in the drive)

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00B0	The type of the parameter "ControlledValueIDN" to be controlled does not match the controlled axis (P-parameter for the axis with interpolation in the control)
ACCESS_ERROR (16#0003)	16#0815	16#0002	Required parameter P-0-0061 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0003)	16#0815	16#0003	Required parameter P-0-0690 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0003)	16#0815	16#0004	Required parameter P-0-0691 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0003)	16#0815	16#0005	Required parameter P-0-0692 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0003)	16#0815	16#0006	Required parameter P-0-0694 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0003)	16#0815	16#0007	Required parameter P-0-0695 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0003)	16#0815	16#0009	The required parameter P-0-0753 is not configured in the optional cyclic AT data of the axis measured.

Tab. 5-26: Error codes of the MB_RegiRegulateType02 function block

5.4.4 MB_RegiRegulateType03

Brief description The MB_RegiRegulateType03 function block writes a control value to the drive using a parameterized correction ramp.

Target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechRegi
MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-27: Reference table of the MB_RegiRegulateType03

Interface description

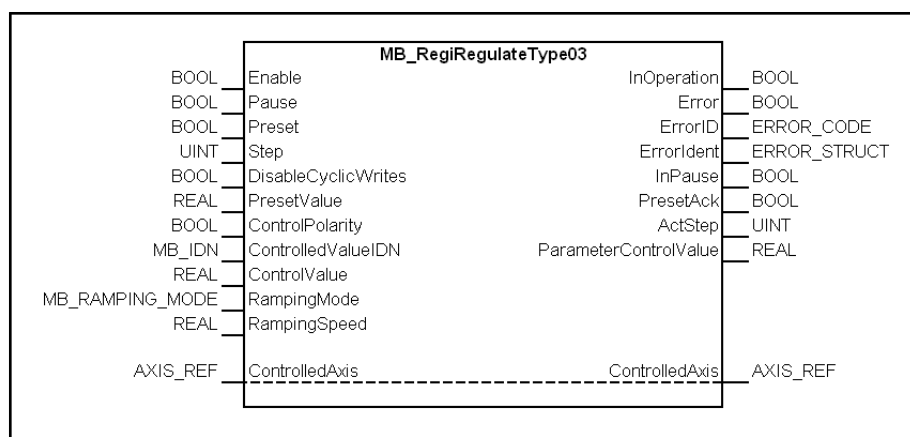


Fig. 5-20: MB_RegiRegulateType03 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enables the function block when set to TRUE
	Pause	BOOL	The "Pause" input is only evaluated while the "Enable" input is active
	Preset	BOOL	The "Preset" input is only evaluated when "Enable" is TRUE. A positive edge of this input forces "ParameterControlValue" to "PresetValue"
	Step	UINT	Processing step
	DisableCyclicWrites	BOOL	Disables cyclic writing to the selected parameter in "ControlledValueIDN" if TRUE
	PresetValue	REAL	Value transferred to the "ParameterControlValue" output at a rising edge of the "Preset" input
	ControlPolarity	BOOL	The polarity of the control value is inverted when this input is TRUE
	ControlledValueIDN	MB_IDN	Sercos IDN of the controlled parameter (e.g. P-0-0691). The following IDNs are supported: 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	Calculated control value from IL_PIType02 for example
	RampingMode	MB_RAMPING_MODE	Specification of the ramping velocity type. MB_RAMPING_MODE, page 335
	RampingSpeed	REAL	This value uses a ramping gradient
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Indicates an error. Clear error with "Enable" = FALSE
	ErrorID	ERROR_CODE	Brief error description

I/O type	Name	Data type	Description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 326
	InPause	BOOL	"Pause" is active. The "ParameterControlValue" output maintains its value
	PresetAck	BOOL	"Preset" is done. The "Preset" value is copied to the "ParameterControlValue" output
	ActStep	UINT	Step value, incrementing with every new value at the output "ParameterControlValue"
	ParameterControlValue	REAL	The register controller function block calculates a value based on the selected parameter in the input "ControlledValueIDN" and stores the result in the output "ParameterControlValue". This value is written to the selected parameter input "ControlledValueIDN" when the "DisableCyclicWrites" input = FALSE
VAR_IN_OUT CONSTANT	ControlledAxis	AXIS_REF	Reference to the controlled axis

Tab. 5-28: Interface variables of the MB_RegiRegulateType03 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Preset	BOOL			FALSE	Continuous
Step	UINT	0	65.535	0	Continuous. Starts at a rising edge at "Enable" and at a falling edge at "Pause"
DisableCyclicWrites	BOOL			FALSE	Rising edge at "Enable"
PresetValue	REAL	N. def.	N. def.	0.0	Rising edge at "Preset"
ControlPolarity	BOOL			FALSE	Continuous
ControlledValueIDN	MB_IDN	N. def.	N. def.	0	Rising edge at "Enable"
ControlValue	REAL	N. def.	N. def.	0.0	Continuous
RampingMode	MB_RAMPING_MODE	0.0	1	RMPING_CONS	Rising edge at "Enable"
RampingSpeed	REAL	0.0	N. def.	1.0	Continuous

Tab. 5-29: Min./max. and default values of the inputs

Time diagram

Timing diagram according to the PLCopen specification.

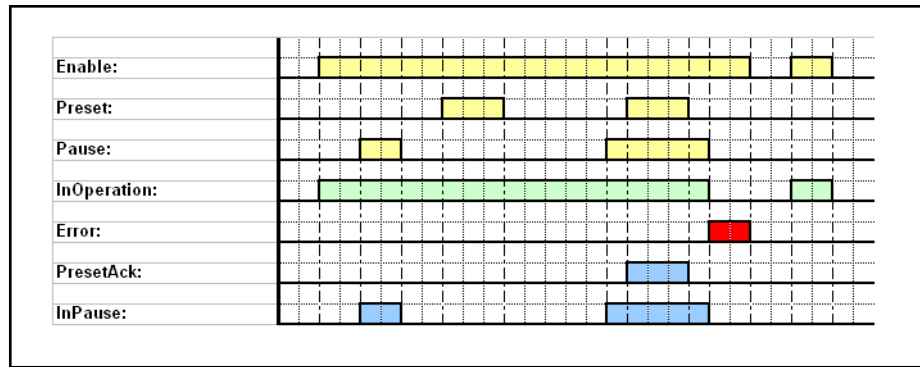


Fig. 5-21: Timing diagram of the MB_RegiRegulateType03

Functional description

The function block writes control values to a parameter in an axis. Depending on the parameter, "ControlledValueIDN", the function block includes the integration of the "ParameterControlValue" input. The following "ControlledValueIDN" values are integrated:

- 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 The "Pause" input is only evaluated as long as the "Enable" input is active. The "ActStep" output is updated even if the "Pause" input is active. As long as "Pause" is active, the "InPause" output is TRUE and the "ParameterControlValue" output maintains its value.

Preset The "PresetAck" output is active if the value in "PresetValue" is taken into account. The "Preset" input has priority over "Pause". The value in "PresetValue" is taken into account on a rising edge of "Preset" if the "Enable" input is active. If successful, the "PresetAck" output is set TRUE and the value in "PresetValue" is written to the controlled parameter.

DisableCyclicWrites Disables cyclic writing to the controlled parameter (Input "ControlledValueIDN") from the function block.

Ramping mode There are two different ramping modes. One mode is constant ramping with the gradient from the "RampingSpeed" ("RAMPING_CONSTANT") input. The other mode depends on the change of the "Step" input. The faster the input "Step" changes, the steeper the ramping of the "ParameterControlValue" ("RAMPING_ADAPTIVE").

- RAMPING_CONSTANT:

In the mode, the change of the "ParameterControlValue" output depends on the "RampingSpeed" input. For example, if the "RampingSpeed" input is set to 1, ramping of the "ParameterControlValue" output is 1 degree/second for the parameter P-0-0691.

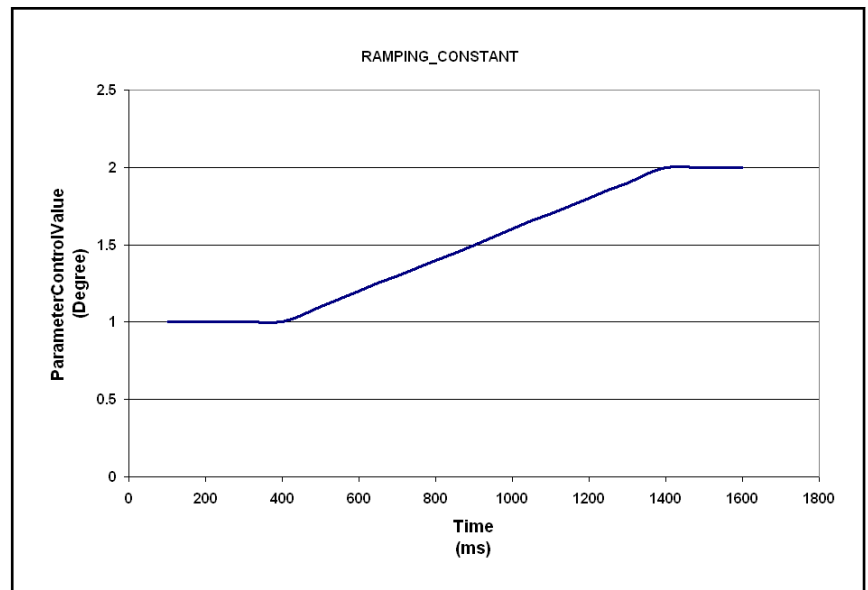


Fig. 5-22: Ramping constant

- **RAMPING_ADAPTIVE:**

In the mode, the change of the "ParameterControlValue" depends on the time difference between the changes of a "Step" input. For example, a change every 0.6(0.3) seconds at the "Step" input results in ramping the "ParameterControlValue" output by 1(2) degree/second for the parameter P-0-0691.

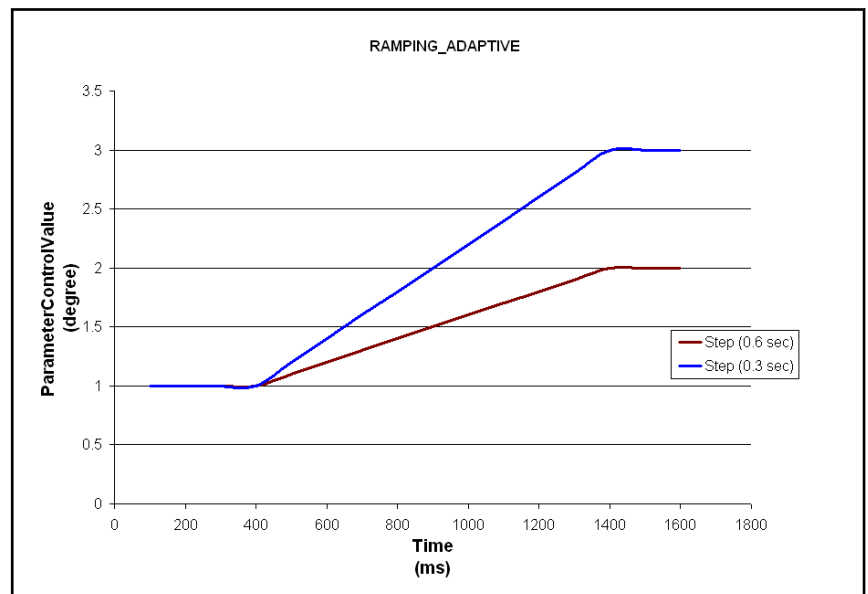


Fig. 5-23: Adaptive ramping

Ramping velocity	The ramping velocity is only valid with the ramping mode RAMPING_CONSTANT.
Normal mode	The "RegiRegulateType" only calculates a new value when the "Step" input changes. The controlled axis has to be a modulo axis.
Required parameterization	MLC parameterization

If a **real axis with interpolation in the drive** is used as the controlled axis ("ControlledAxis" input), the parameter used in the "ControlledValueIDN" input has to be one of the following supported P-parameters:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 and P-0-0695



Additionally, this P-parameter has to be configured in the MDT of the cyclic data channel in the drive.

If a **real axis with interpolation in the control** is used as a **virtual axis**, the parameter used in the "ControlledValueIDN" input has to be one of the following supported P-parameters:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

MLD parameterization

The parameter used as controlled parameter for the "ControlledValueIDN" (P-parameter) input has to be configured in the "AxisData" structure. Thus, the axis data structure has to be activated. The following P-parameters are supported:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

For the MLD firmware MPx20 (IL2G) and later, axis references which are not included in the "AxisData" structure are also supported at the "MeasuredAxis" (> VMA1).

In this case, the correction window is not determined using the current position of "MeasuredAxis", but via P-0-0761 "Master axis position for slave axis" of the local axis.

To provide the required position information in P-0-0761 "Master axis position for slave axis", the master axis generator has to be configured respectively.

Parameters involved:

- P-0-0760 "Virtual master axis, position control word"
- P-0-0916 "Master axis format converter signal selection"

See "Virtual master axis generator" in the documentation IndraDrive MPx-20 Functions, Application Description, DOK-INDRV*-MP*-20VRS*-APRS-EN-P.

Troubleshooting

The function block generates the following error messages in Additional1/Additional2 using the "**F_RELATED_TABLE**", 16#0170:

INPUT_RANGE_ERROR (16#0006)

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Drive is not enabled or drive error
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis ("AXIS_REF") was changed while the function block was operating
RESOURCE_ERROR (16#0003)	16#0012	16#0003	"AXIS_REF.CntrlNo" is not in the valid range (axes of remote controls are invalid)
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000A	S-0-0103 or A-0-0045 of the controlled axis is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000B	S-0-0091 or A-0-0032 of the controlled axis is not in the valid range

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#0811	16#000C	P-0-0750 of the controlled axis is not in the valid range
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0001	Selected "ControlledValueIDN" is not supported
INPUT_INVALID_ERROR (16#0001)	16#0812	16#000C	Value < 0 for the "RampingSpeed" input
INPUT_INVALID_ERROR (16#0001)	16#0812	16#000D	Modulo value of the controlled axis is 0 or could not be read
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00A0	The type of the parameter "ControlledValueIDN" to be controlled does not match the controlled axis (A-parameter for the axis with interpolation in the drive)
INPUT_INVALID_ERROR (16#0001)	16#0812	16#00B0	The type of the parameter "ControlledValueIDN" to be controlled does not match the controlled axis (P-parameter for the axis with interpolation in the control)
ACCESS_ERROR (16#0004)	16#0815	16#0002	Required parameter P-0-0061 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0004)	16#0815	16#0003	Required parameter P-0-0690 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0004)	16#0815	16#0004	Required parameter P-0-0691 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0004)	16#0815	16#0005	Required parameter P-0-0692 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0004)	16#0815	16#0006	Required parameter P-0-0694 is not configured in the optional cyclic MDT data of the controlled axis
ACCESS_ERROR (16#0004)	16#0815	16#0007	Required parameter P-0-0695 is not configured in the optional cyclic MDT data of the controlled axis

Tab. 5-30: Error codes of the MB_RegiRegulateType03 function block

5.4.5 MB_RegiRegulateType04

Brief description The MB_RegiRegulateType04 function block writes a control value to up to eight drives. If the correction window is active, the control deviations are corrected in one parameterized correction window.

Target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechRegi

Tab. 5-31: Reference table of the MB_RegiRegulateType04

Interface description

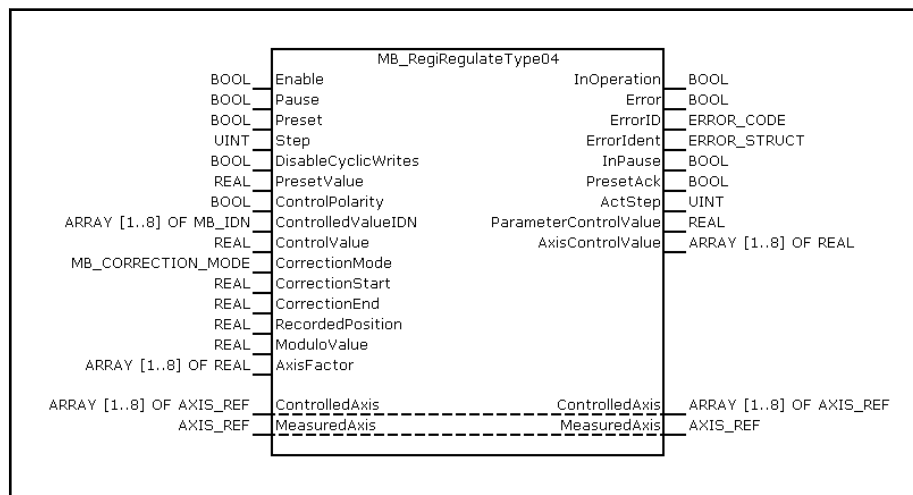


Fig. 5-24: MB_RegiRegulateType04 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enables the function block when set to TRUE
	Pause	BOOL	The "Pause" input is only evaluated while the "Enable" input is active
	Preset	BOOL	The "Preset" input is only evaluated when "Enable" is TRUE. A positive edge of this input forces "ParameterControlValue" to "PresetValue"
	Step	UINT	Processing step
	DisableCyclicWrites	BOOL	Disables cyclic writing to the selected parameter in "ControlledValueIDN" if TRUE
	PresetValue	REAL	Value transferred to the "ParameterControlValue" output at a rising edge of the "Preset" input
	ControlPolarity	BOOL	The polarity of the control value is inverted when this input is TRUE
	ControlledValueIDN	ARRAY [1..8] OF MB_IDN	Sercos IDN of the controlled parameter (e.g. P-0-0691). The following IDNs are supported: 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	Calculated control value from IL_PIType02 for example
	CorrectionMode	MB_CORRECTION_MODE	Properties of the correction window MB_CORRECTION_MODE , page 335

I/O type	Name	Data type	Description
	CorrectionStart	REAL	Start of the correction window
	CorrectionEnd	REAL	End of the correction window
	RecordedPosition	REAL	Measured value
	ModuloValue	REAL	Modulo value of the measured axis
	AxisFactor	ARRAY [1..8] OF REAL	Factor to archive an axis-dependent scaling. The "ParameterControlValue" value is multiplied with this axis-dependent factor (ARRAY) and is distributed to every controlled axis
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Indicates an error. Clear error with "Enable" = FALSE
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 333
	InPause	BOOL	"Pause" is active. The "ParameterControlValue" and "AxisControlValue" outputs maintain their values.
	PresetAck	BOOL	"Preset" is done. The "Preset" value is copied to the "ParameterControlValue" output
	ActStep	UINT	Step value, incrementing with every new value at the output "ParameterControlValue"
	ParameterControlValue	REAL	The register controller function block calculates a value based on the selected parameter in the input "ControlledValueIDN" and stores the result in the output "ParameterControlValue". This value is written to the selected parameter input "ControlledValueIDN" when the "DisableCyclicWrites" input = FALSE
	AxisControlValue	ARRAY [1..8] OF REAL	Calculated value of the register controller function block written to the "ControlledValueIDN" input. The "AxisControlValue" is the scaled "ParameterControlValue". Used for diagnostic purposes or for further operation if "DisableCyclicWrites" = TRUE
VAR_IN_OUT CONSTANT	ControlledAxis	ARRAY [1..8] OF AXIS_REF	Reference to the controlled axes
	MeasuredAxis	AXIS_REF	Reference to the measured axis

Tab. 5-32: Interface variables of the MB_RegiRegulateType04 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Preset	BOOL			FALSE	Continuous

Name	Type	Min. value	Max. value	Default value	Effective
Step	UINT	0	65.535	0	Continuous. Starts at a rising edge at "Enable" and at a falling edge at "Pause"
DisableCyclic-Writes	BOOL			FALSE	Rising edge at "Enable"
PresetValue	REAL	N. def.	N. def.	0.0	Rising edge at "Preset"
ControlPolarity	BOOL			FALSE	Continuous
ControlledValueIDN	ARRAY [1..8] OF MB_IDN	N. def.	N. def.	0	Rising edge at "Enable"
ControlValue	REAL	N. def.	N. def.	0.0	Continuous
CorrectionMode	MB_CORRECTION_MODE	0	4	CORR_REL	Continuous
CorrectionStart	REAL	0.0	ModuloValue	0.0	Continuous
CorrectionEnd	REAL	0.0	ModuloValue	0.0	Continuous
RecordedPosition	REAL	0.0	ModuloValue	0.0	Change of the "Step" input
ModuloValue	REAL	0.0	N. def.	0.0	Rising edge at "Enable"
AxisFactor	ARRAY [1..8] OF REAL	N. def.	N. def.	1.0	Continuous

Tab. 5-33: Min./max. and default values of the inputs

Time diagram

Timing diagram according to the PLCopen specification.

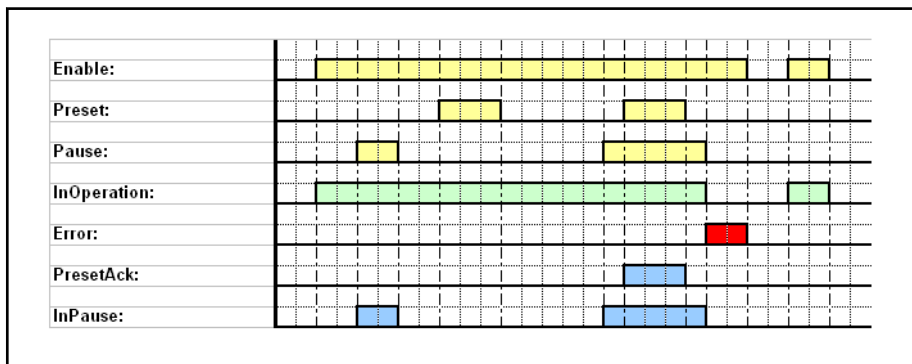


Fig. 5-25: Timing diagram of the MB_RegiRegulateType04

Functional description

The function block writes control values to a parameter in an axis. Depending on the parameter, "ControlledValueIDN", the function block includes the integration of the "ParameterControlValue" input. This integration is done for the following ControlledValueIDN values:

- 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

The "Pause" input is only evaluated when the "Enable" input is active. The "ActStep" output is updated even if the "Pause" input is active. As long as

"Pause" is active, the "InPause" output is TRUE and the "ParameterControlValue" and "AxisControlValue" outputs maintain their values.

Preset The "PresetAck" output is active if the value in "PresetValue" is taken into account. The "Preset" input has priority over "Pause". The value in "PresetValue" is taken into account on a rising edge of "Preset" if the "Enable" input is active. If successful, the "PresetAck" output is set TRUE and the value in "PresetValue" is written to the controlled parameter.

DisableCyclicWrites Disables cyclic writing to the controlled parameter (Input "ControlledValueIDN") from the function block.

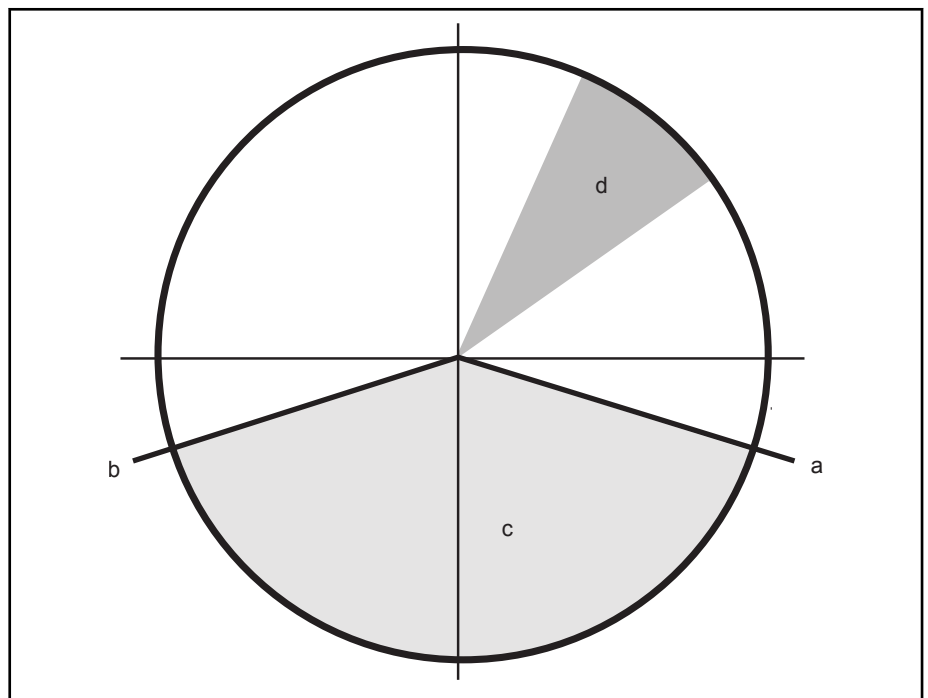
Correction window The correction window is an angular range relating to the measuring axis. The register controller controls the control deviations within this window. The position and size of the correction window are set with the help of the following inputs:

The "CorrectionMode" input sets the correction window either absolute or relative. If the "CorrectionMode" is set to CORR_WIN_DISABLED, the correction window is inactive.

The "CorrectionStart" input defines the start of the correction window.

The "CorrectionEnd" input defines the end of the correction window.

The correction window supports only a positive modulo range (e.g., 0...+360 degrees) of the measuring axis.



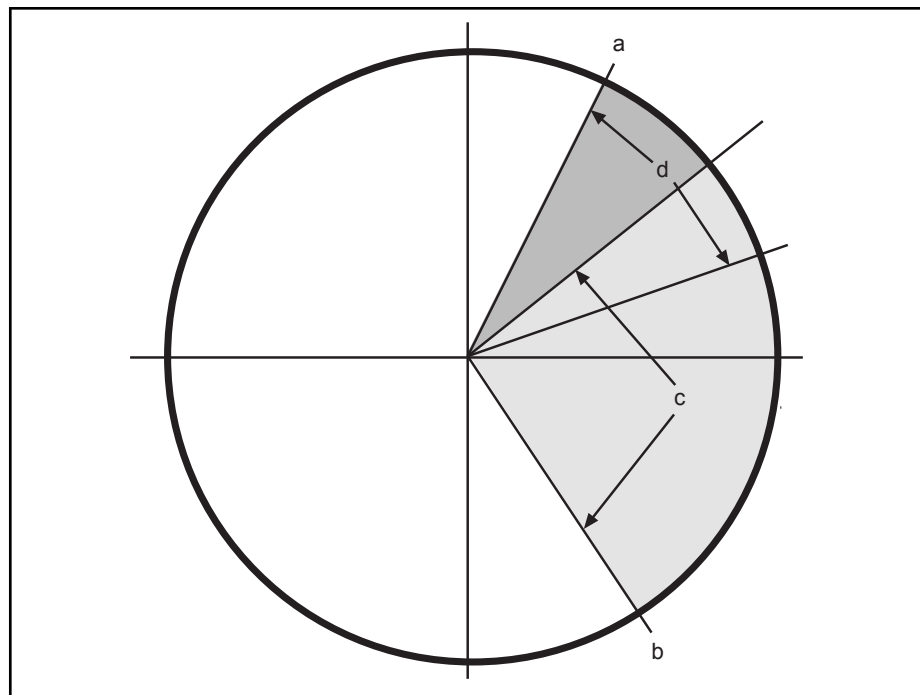
a: Start correction window

b: End correction window

c: Correction window

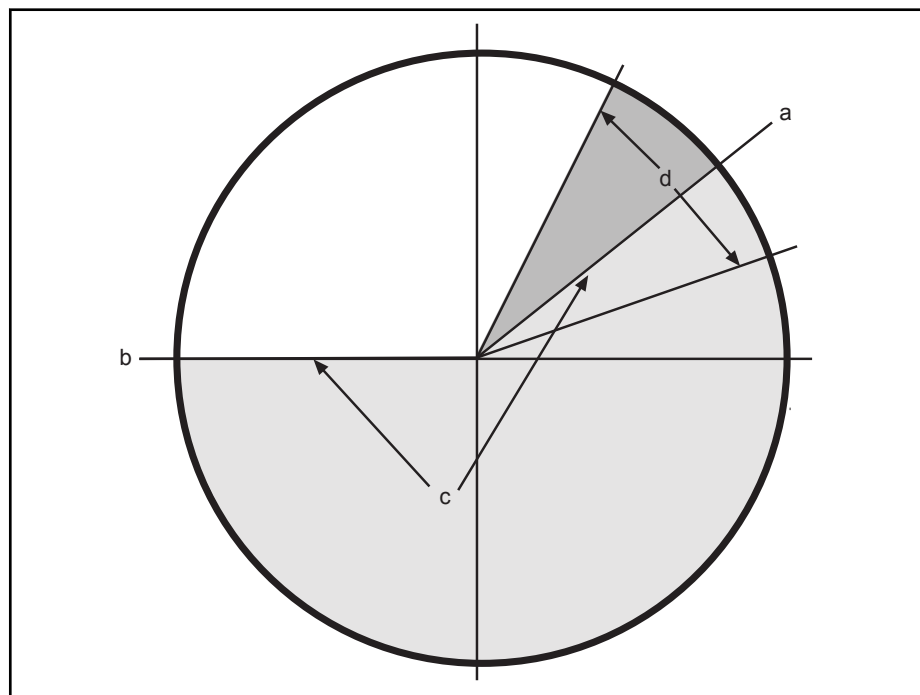
d: Expectation window

Fig. 5-26: Correction window absolute



- a:** Start correction window = measuring end
b: Relative end of correction window to measuring end
c: Correction window
d: Expectation window

Fig. 5-27: Correction window relative



- a:** Start correction window = measuring end
b: End of correction window = perm. angle
c: Correction window
d: Expectation window

Fig. 5-28: Correction window relative start absolute end

- Multiple axes** The calculated control values are written to the respective parameters ("ControlValueIDN") of up to 8 controlled axes. The axes are referenced with the "ControlledAxis[x]" input.
- Individual Control Value Scaling** The control value of each individual axis can be scaled with the "AxisFactor[x]" input.
- Normal mode** The RegiRegulateType04 function block only calculates a new value if the "Step" input changes. The controlled and measured axis must be modulo axes.

Required parameterization MLC parameterization

If a **real axis with interpolation in the drive** is used as the controlled axis ("ControlledAxis" input), the parameter used in the "ControlledValueIDN" input has to be one of the following supported P-parameters:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 and P-0-0695



Additionally, this P-parameter has to be configured in the MDT of the cyclic data channel in the drive.

If a **real axis with interpolation in the control** is used as a **virtual axis**, the parameter used in the "ControlledValueIDN" input has to be one of the following supported P-parameters:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

MLD parameterization

The parameter used as controlled parameter for the "ControlledValueIDN" (P-parameter) input has to be configured in the "AxisData" structure. Thus, the axis data structure has to be activated. The following P-parameters are supported:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

For the MLD firmware MPx20 (IL2G) and later, axis references which are not included in the "AxisData" structure are also supported at the "MeasuredAxis" (> VMA1).

In this case, the correction window is not determined using the current position of "MeasuredAxis", but via P-0-0761 "Master axis position for slave axis" of the local axis.

To provide the required position information in P-0-0761 "Master axis position for slave axis", the master axis generator has to be configured respectively.

Parameters involved:

- P-0-0760 "Virtual master axis, position control word"
- P-0-0916 "Master axis format converter signal selection"

See "Virtual master axis generator" in the documentation IndraDrive MPx-20 Functions, Application Description, DOK-INDRV*-MP*-20VRS*-APRS-EN-P.

- Troubleshooting** The function block generates the following error messages in Additional1/Additional2 using the "**F_RELATED_TABLE**", 16#0170:
- INPUT_RANGE_ERROR (16#0006)

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Drive is not enabled or drive error
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0009	16#0000	Selected axis (AXIS_REF) was changed while the function block is operating
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AXIS_REF is not in the valid range
RESOURCE_ERROR (16#0003)	16#0012	16#0003	"AXIS_REF.CntrlNo" is not in the valid range (axes of remote controls are invalid)
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0001	Length of the correction window < 0.25% of the value at the "Modulo-Value" input
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0002	Input "CorrectionStart" < 0
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0003	Input "CorrectionEnd" < 0
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0004	Value < 0 input "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0005	Value < 0 input "RecordedPosition"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0006	Input "CorrectionStart" > "Modulo-Value"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0007	Input "CorrectionEnd" > "Modulo-Value"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0008	Input "RecordedPosition" > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0009	Correction path > "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0x10	S-0-0103 or A-0-0045 of the controlled axis [x] is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0x20	S-0-0091 or A-0-0032 of the controlled axis [x] is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0811	16#0x30	P-0-0750 of the controlled axis [x] is not in the valid range
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0x01	Selected "ControlledValueIDN[x]" is not supported
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0002	Input "CorrectionMode" is invalid
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0003	"CorrectionStart" = "CorrectionEnd"
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0xA0	The type of the parameter to be controlled "ControlledValueIDN[x]" does not match the controlled axis "Controlled Axis[x]" (A-parameter for the axis with interpolation in the drive)
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0xB0	The type of the parameter to be controlled "ControlledValueIDN[x]" does not match the controlled axis "Controlled Axis[x]" (P-parameter for the axis with interpolation in the control)

ErrorID	Additional1	Additional2	Description
ACCESS_ERROR (16#0003)	16#0815	16#0x02	Required parameter P-0-0061 is not configured in the optional cyclic MDT data of the controlled axis[x]
ACCESS_ERROR (16#0003)	16#0815	16#0x03	Required parameter P-0-0690 is not configured in the optional cyclic MDT data of the controlled axis[x]
ACCESS_ERROR (16#0003)	16#0815	16#0x04	Required parameter P-0-0691 is not configured in the optional cyclic MDT data of the controlled axis[x]
ACCESS_ERROR (16#0003)	16#0815	16#0x05	Required parameter P-0-0692 is not configured in the optional cyclic MDT data of the controlled axis[x]
ACCESS_ERROR (16#0003)	16#0815	16#0x06	Required parameter P-0-0694 is not configured in the optional cyclic MDT data of the controlled axis[x]
ACCESS_ERROR (16#0003)	16#0815	16#0x07	Required parameter P-0-0695 is not configured in the optional cyclic MDT data of the controlled axis[x]
ACCESS_ERROR (16#0003)	16#0815	16#0009	The required parameter P-0-0753 is not configured in the optional cyclic AT data of the axis measured.

Tab. 5-34: Error codes of the MB_RegiRegulateType04 function block

5.4.6 MB_CORRECTION_MODE

The enumeration type "MB_CORRECTION_MODE" contains possible forms of correction windows for the function blocks [MB_RegiRegulateType02](#), page 313, and [MB_RegiRegulateType04](#), page 327.

Name	Value	Description
CORR_WIN_DISABLED	0	Correction window is deactivated
CORR_ABS	1	Absolute correction window [CorrectionStart .. CorrectionEnd].
CORR_REL	2	Relative correction window [Start = "RecordedPosition" + "CorrectionStart" .. End = "RecordedPosition" + "CorrectionEnd"]
CORR_REL_ABS	3	Relative start... absolute end correction window [Start = "RecordedPosition" + "CorrectionStart" .. End = "CorrectionEnd"]
CORR_ABS_REL	4	Absolute start correction window... relative end correction window [Start = "CorrectionStart" .. End = "RecordedPosition" + "CorrectionEnd"]

Tab. 5-35: ENUM MB_CORRECTION_MODE

5.4.7 MB_RAMPING_MODE

The enumeration type "MB_RAMPING_MODE" permits for function block [MB_RegiRegulateType03](#), page 321, selection from the following ramping modes:

Name	Value	Description
RAMPING_CONSTANT	0	Ramping velocity is constant
RAMPING_ADAPTIVE	1	Ramping velocity is calculated based on the time difference in the "Step" input

Tab. 5-36: ENUM MB_RAMPING_MODE

5.5 Register setpoint lock function block

5.5.1 Introduction and overview

The register setpoint lock function block is part of the libraries ML_TechRegi (MLC) and MX_TechRegi.lib (MLD).

5.5.2 MB_SetpointLockType01

Brief description The MB_RegiSetpointLockType01 function block latches a value for a register controller.

Target system	Library
MLC 14VRS Function package TEC	ML_TechRegi
MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib (IL1G)
MLD with MPx20 with MA function package	MX_TechRegi (IL2G)

Tab. 5-37: Reference table of the MB_SetpointLockType01

Interface description

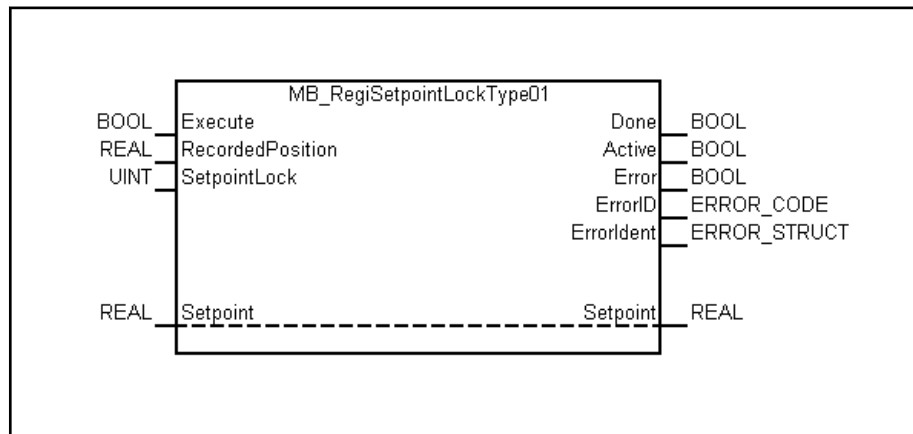


Fig. 5-29: MB_SetpointLockType01 function block

I/O type	Name	Data type	Description
VAR_IN	Execute	BOOL	The register setpoint lock function block becomes active at a rising edge
	RecordedPosition	REAL	The applied position is copied to the "Setpoint" input when "SetpointLock" changes
	SetpointLock	UINT	The value has to increase if the value in "Recorded-Position" is locked. At 65.535, the value becomes 0 (modulo type value). Typically, an "actStep" output of a technology function block is used as input

I/O type	Name	Data type	Description
VAR_OUT	Done	BOOL	The function block is completed and the value in the "RecordPosition" input is copied to "Setpoint"
	Active	BOOL	Function block is active
	Error	BOOL	Indicates that an error occurred during the function block operation
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 338
VAR_IN_OUT	Setpoint	REAL	Setpoint

Tab. 5-38: Interface variables of the MB_RegiSetpointLockType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
RecordedPosition	REAL			0.0	Change of "Setpoint-Lock"
SetpointLock	UINT	0	65.535	0	Continuous. Initialized at a rising edge of "Execute"

Tab. 5-39: Min./max. and default values of the inputs

Time diagram

The following diagram shows the signal-time diagram of the MB_RegiSetpointLockType01 function block.

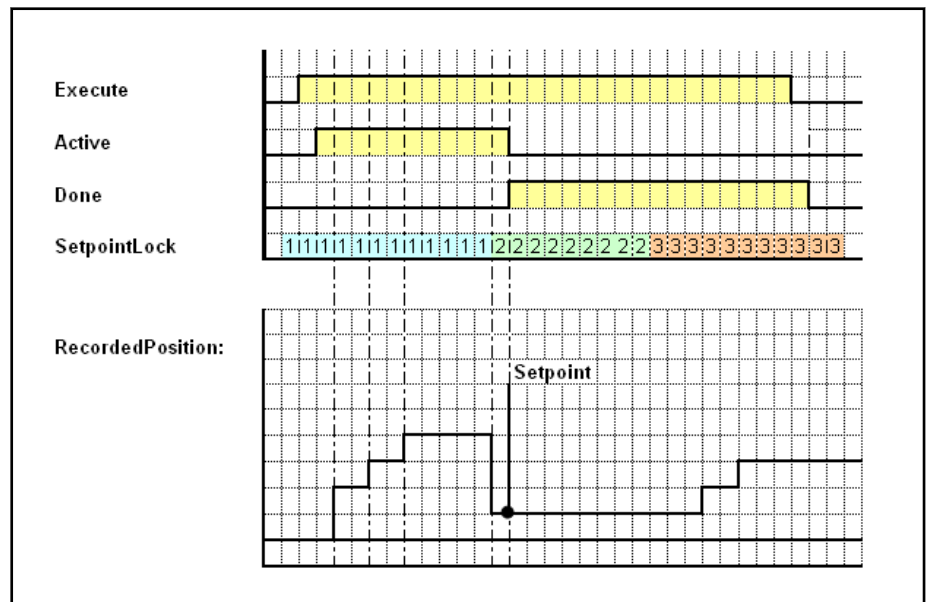


Fig. 5-30: Timing diagram of the MB_RegiSetpointLockType01

Functional Description

The function block sets "Setpoint" to the next value at the "RecordedPosition" input. The next occurrence is detected when the "SetpointLock" input changes. This can be used to set a new command value for a register controller for example.

Troubleshooting The function block generates the following error messages in Additional1/ Additional2 using the **F_RELATED_TABLE**, 16#0170:

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state

Tab. 5-40: Error codes of the MB_RegiSetpointLockType01 function block

5.6 Register Controller Function Block

5.6.1 Introduction and overview

The register controller function block is part of the libraries ML_TechRegi (MLC) and MX_TechRegi.lib (MLD).

5.6.2 MB_RegisterControllerType02

Brief description The MB_RegisterControllerType02 register controller function block supports the following functionality:

- Calculation of a correction value based on the measured value and the command value using a P- or PI-control loop
- Preset function
- Pause function
- Min./max. limitation of the calculated control value
- Correction window function
- Lock setpoint on demand

Target system	Library
MLC 14VRS Function package TEC	ML_TechRegi
MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-41: Reference table of the MB_RegisterControllerType02

Interface description

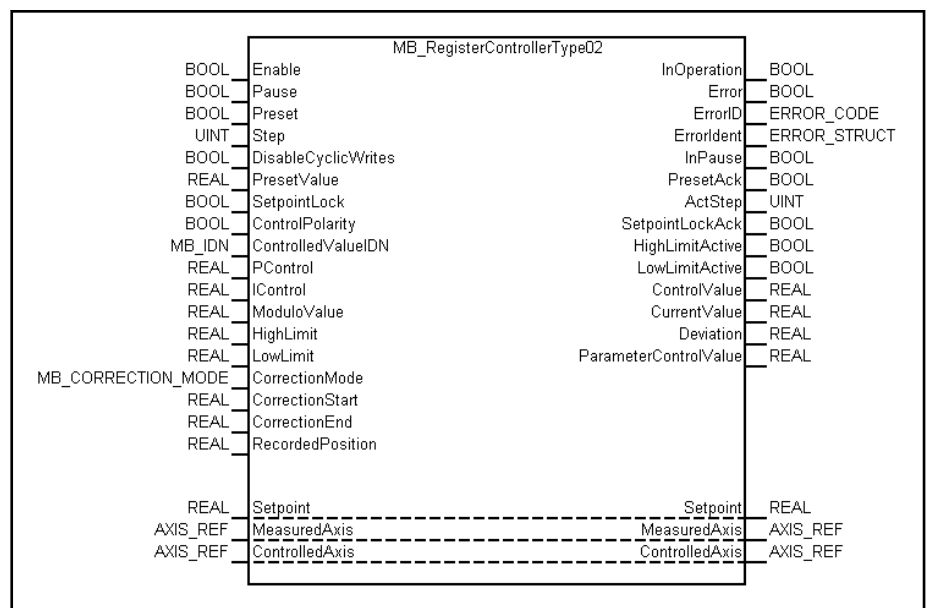


Fig. 5-31: MB_RegisterControllerType02 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enables the register controller
	Pause	BOOL	The "Pause" input is only evaluated when the "Enable" input is active. The "ControlValue" and "ParameterControlValue" outputs are constant when "Pause" is active
	Preset	BOOL	The "Preset" input is only evaluated when "Enable" is TRUE. A positive edge of this input forces the control loop to "PresetValue". In addition, the "Preset" input sets the "ParameterControlValue" to the "PresetValue"
	Step	UINT	Processing step
	DisableCyclicWrites	BOOL	Disables cyclic writing to the selected parameter in "ControlledValueIDN" if TRUE
	PresetValue	REAL	Value applied to the "ControlValue" and "ParameterControlValue" outputs at the rising edge of the "Preset" input
	SetpointLock	BOOL	Enables command value application (setpoint lock) If TRUE, the next measured mark is latched
	ControlPolarity	BOOL	The polarity of the control value is inverted when this input is TRUE
	ControlledValueIDN	MB_IDN	Sercos IDN of the controlled parameter (e.g. P-0-0691). The following IDNs are supported: 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	Proportional gain of the PI-controller

I/O type	Name	Data type	Description
	IControl	REAL	Integral action time T_n of the PI-controller
	ModuloValue	REAL	Modulo value of the measured axis
	HighLimit	REAL	Maximum value for "ControlValue"
	LowLimit	REAL	Minimum value for "ControlValue"
	CorrectionMode	MB_CORRECTION_MODE	Properties of the correction window MB_CORRECTION_MODE, page 335
	CorrectionStart	REAL	Start of the correction window
	CorrectionEnd	REAL	End of the correction window
	RecordedPosition	REAL	Measured value of axis position
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Indicates an error. Clear error with "Enable" = FALSE
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 347
	InPause	BOOL	"Pause" is active. The "ControlValue" and "ParameterControlValue" outputs maintain their values
	PresetAck	BOOL	"Preset" is done. The "Preset" value is copied to the "ControlValue" and "ParameterControlValue" outputs
	ActStep	UINT	Step value, incrementing with every new value at the "ControlValue" output
	SetpointLockAck	BOOL	"Setpoint" is locked. Current value in "RecordedPosition" is copied to "Setpoint"
	HighLimitActive	BOOL	Upper limit is active. "ControlValue" would be higher, but is limited to "HighLimit"
	LowLimitActive	BOOL	Lower limit is active. "ControlValue" would be lower, but is limited to "LowLimit"
	ControlValue	REAL	Control value calculated by the register controller
	CurrentValue	REAL	The currently recorded position used in the register controller
	Deviation	REAL	"Deviation" = ("Setpoint" - "CurrentValue") of the register controller
	ParameterControlValue	REAL	The register controller calculates a value based on the selected parameter in the "ControlledValueIDN" input and stores the result in the "ParameterControlValue" output. This value is written to the selected parameter input "ControlledValueIDN" when the "DisableCyclicWrites" input = FALSE
VAR_IN_OUT	Setpoint	REAL	Command value. This value is compared to "RecordedPosition"

I/O type	Name	Data type	Description
VAR_IN_OUT CONSTANT	MeasuredAxis	AXIS_REF	Reference to the measured axis
VAR_IN_OUT CONSTANT	ControlledAxis	AXIS_REF	Reference to the controlled axis

Tab. 5-42: Interface variables of the MB_RegisterControllerType02 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Preset	BOOL			FALSE	"Enable" TRUE (continuous)
Step	UINT	0	65.535	0	Continuous. Starts at a rising edge at "Enable" and at a falling edge at "Pause"
DisableCyclicWrites	BOOL			FALSE	Rising edge at "Enable"
SetpointLock	BOOL			FALSE	Continuous
PresetValue	REAL	N. def.	N. def.	0.0	Rising edge at "Preset"
ControlPolarity	BOOL			FALSE	Continuous
ControlledValueIDN	MB_IDN	N. def.	N. def.	0	Rising edge at "Enable"
Setpoint	REAL	N. def.	N. def.	0.0	Change of "Step"
PControl	REAL	0.0	N. def.	1.0	Continuous
IControl	REAL	0.0	N. def.	0.0	Continuous
ModuloValue	REAL	0.0	N. def.	0.0	Rising edge at "Enable"
HighLimit	REAL	LowLimit	N. def.	0.0	Continuous
LowLimit	REAL	N. def.	HighLimit	0.0	Continuous
CorrectionMode	MB_CORRECTION_MODE			CORR_REL	Continuous
CorrectionStart	REAL	0.0	ModuloValue	0.0	Continuous
CorrectionEnd	REAL	0.0	ModuloValue	0.0	Continuous
RecordedPosition	REAL	N. def.	N. def.	0.0	Change of "Step"

Tab. 5-43: Overview on the min./max. values and default values of the inputs

Time diagram

The following diagram shows the signal-time characteristics of MB_RegisterControllerType02 including the "Pause" and "Preset" functions.

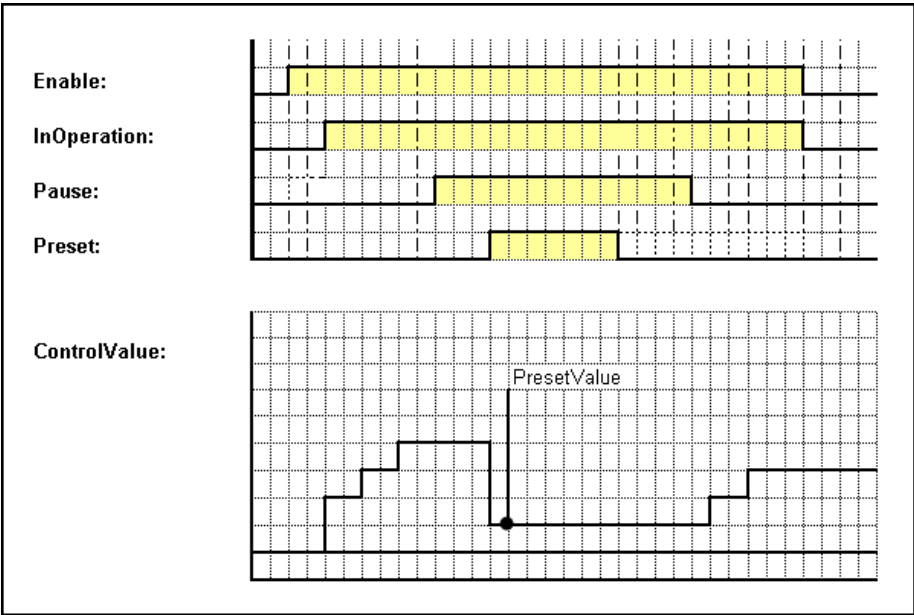


Fig. 5-32: Timing diagram of the MB_RegisterControllerType02 function block with the functions "Pause" and "Preset"

Functional description A sensor detects marks, perforations, cuts or pasted joints on the material and provides a (binary) signal to the drive.

The touch probe feature of the drive determines the edge of the sensor signal and latches the following positions, depending on the signal selection of the touch probe:

Actual drive position parameters	S-0-0051, S-0-0053, P-0-0753
Master axis position parameters	P-0-0052, P-0-0227, P-0-0775, P-0-0776, P-0-0778

The touch probe function records positional data with a resolution of 0.5 μ s. The sensor has to provide a 24V signal with a rise time within the μ s range. The sensor specific delay time can be compensated using the "DeadTimePosEdge" or "DeadTimeNegEdge" inputs of the MB_InitTouchProbe function block.

The register controller calculates the control deviation between the measured and setpoint positions and determines the required correction value once a new edge of the sensor signal is detected.

Changes of the correction value become active via the trapezoidal profile, velocity ramp, PT1-filter or instantaneously, depending on the selected control parameter ("ControlledValueIDN") input of the function block. The following table shows the available control parameters ("ControlledValueIDN") as well the resulting MotionProfile of all available drive operation modes with synchronization.

Parameter Name	Controlled-ValueIDN	Motion Behavior in case the parameter value is changed	Available in the following Operation Modes			
			Velocity-Synchronization	Phase-Synchronization	Cam Shaft	Electronic Motion Profile
Angle offset begin of profile	P-0-0061 A-0-2730	Changes of P-0-0061 become active via ramp using P-0-0158 (velocity). If P-0-0158=0, changes of P-0-0061 become active instantaneously.	No	No	Yes	Yes
Additive velocity command value, process loop	P-0-0690 A-0-2615	Changes of P-0-0690 become active instantaneously	Yes	No	No	No
Additive position command value, process loop	P-0-0691 A-0-2610	If P-0-0155 Bit0 = 0: Changes of P-0-0691 become active via PT1 filter using P-0-0060 (filter time constant). If P-0-0060=0, changes of P-0-0691 become active instantaneously. If P-0-0155 Bit0 = 1: Changes of P-0-0691 become active instantaneously	Yes	Yes	Yes	Yes
Additive master axis position, process loop	P-0-0692 A-0-2600	Changes of P-0-0692 is smoothed with a 1st order filter using P-0-0693(filter time constant) and then added to the sum of P-0-0054, Additive master axis position and P-0-0053, Master axis position or P-0-0052, Actual position value of measuring encoder.	Yes	Yes	Yes	Yes
Gear ratio fine adjust, process loop	P-0-0694 A-0-2605	Changes of P-0-0694 become active instantaneously	Yes	Yes	Yes	Yes
Angle offset begin of profile, process loop	P-0-0695 A-0-2620	Changes of P-0-0695 become active via PT1 filter P-0-0696 (filter time constant). If P-0-0696=0, changes of P-0-0695 become active instantaneously.	No	No	Yes	Yes

Fig. 5-33: Supported control parameter and behavior

The following "ControlledValueIDN" values are integrated:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Initialization During initialization, the "Setpoint" and "RecordedPosition" inputs must have valid data.

PresetValue For a trouble-free start of the controller outputs "ControlValue" and "ParameterControlValue", set the "Preset" input before activating the "Enable" input. During this period, the "Setpoint" and "RecordedPosition" inputs have to have valid data. In addition, the upper limits and lower limits are monitored to en-

sure a limitation of the "ControlValue" output if the current value for "PresetValue" is not in the valid range of "HighLimit" or "LowLimit".

The "PresetAck" output is active if the value in the "PresetValue" input is copied to the "ControlValue" and "ParameterControlValue" outputs. The "Preset" input has a higher priority than the "Pause" input. The value in "PresetValue" is copied to the respective outputs at the rising edge of the "Preset" input if "Enable" is active. If successful, the "PresetAck" output is set to TRUE.

Pause The "Pause" input is only evaluated as long as the "Enable" input is active. In this case, the deviation ("Setpoint" – "RecordedPosition") is set to 0. Therefore, the outputs "ControlValue" and "ParameterControlValue" do not change anymore. This can be used to maintain the values of the "ControlValue" and "ParameterControlValue" outputs. The "ActStep" output is updated even if the "Pause" input is active. As long as the "Pause" input is active, the "In-Pause" output is TRUE.

Lock Setpoint A manual to Automatic mode change is achieved by latching to the next register mark that is detected. For example, when a new web is pulled into position and the register mark is detected, the "SetpointLock" input is set to TRUE. The value in "Setpoint" is overwritten with the value in "RecordedPosition" the next time the "Step" input changes.

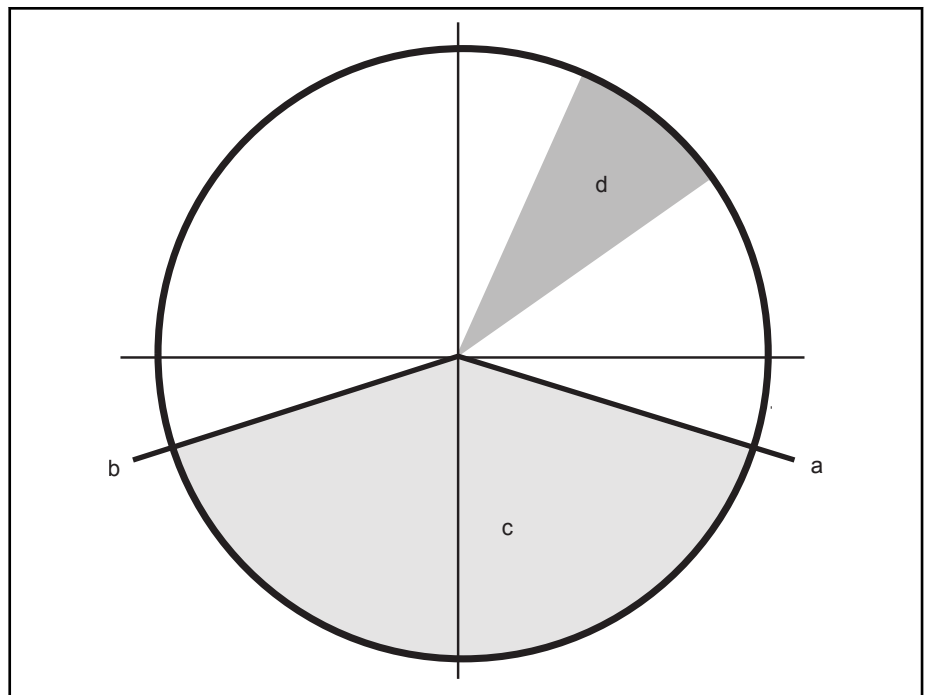
Correction window The correction window is an angular range relating to the measuring axis. The register controller controls the control deviations within this window. The position and size of the correction window are set with the help of the following inputs:

The "CorrectionMode" input sets the correction window either absolute or relative. If the "CorrectionMode" is set to CORR_WIN_DISABLED, the correction window is inactive.

The "CorrectionStart" input defines the start of the correction window.

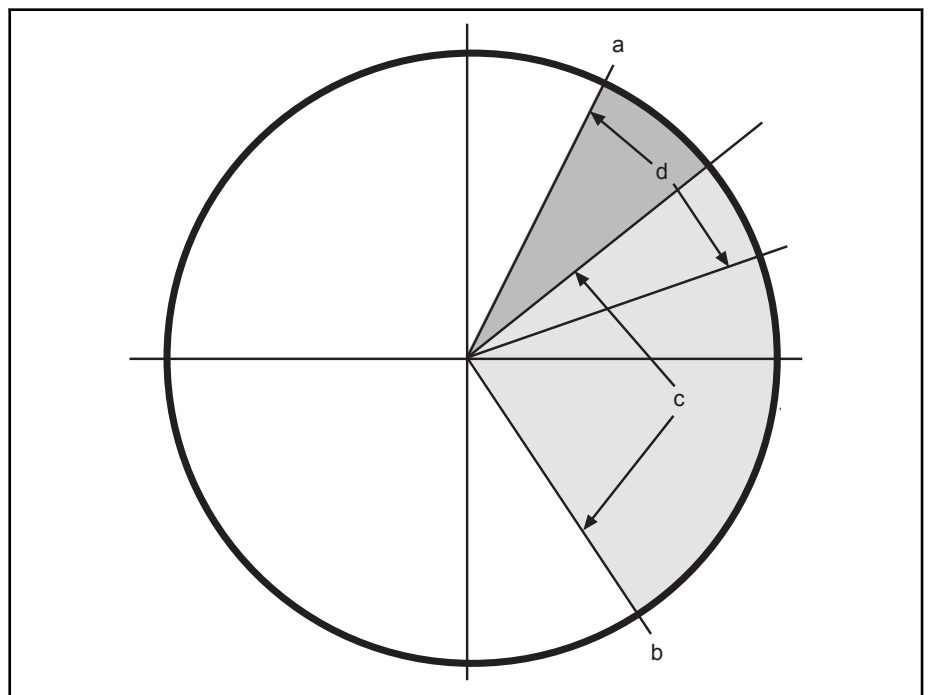
The "CorrectionEnd" input defines the end of the correction window.

The correction window supports only a positive modulo range (e.g., 0...+360 degrees) of the measuring axis.



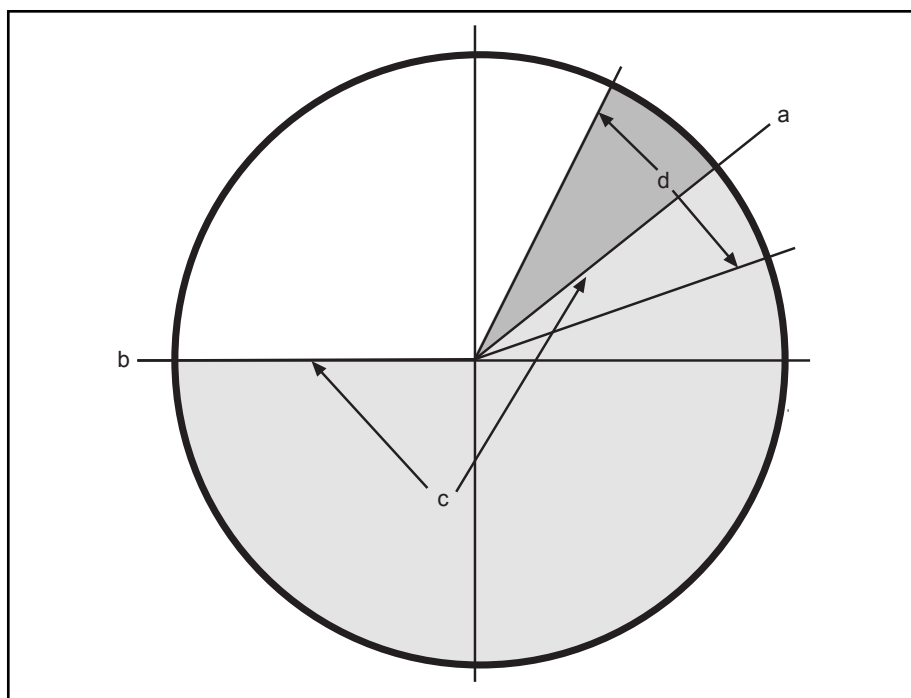
- a:** Start correction window
b: End correction window
c: Correction window
d: Expectation window

Fig. 5-34: *Correction window absolute*



- a:** Start correction window = measuring end
b: Relative end of correction window to measuring end
c: Correction window
d: Expectation window

Fig. 5-35: *Correction window relative*



- a:** Start correction window = measuring end
b: End of correction window = perm. angle
c: Correction window
d: Expectation window

Fig. 5-36: Correction window relative start absolute end

Normal mode

The RegisterController only calculates a new value if the "Step" input changes. The "PControl" and "IControl" controller parameters can be changed continuously.

If the value in the "ControlValue" output exceeds or falls below the values of outputs "HighLimit" or "LowLimit", then "ControlValue" is limited to one of the following limit values:

- If the value in "ControlValue" is positive, then the value in "HighLimit" is used.
- If the value in "ControlValue" is negative, then the value in "LowLimit" is used.

If the register controller has an integral action, then an anti-reset wind-up is carried out and the integral action is no longer summed. In this case the "HighLimitActive" and "LowLimitActive" outputs are set to TRUE.

Required parameterization

MLC parameterization

If a **real axis with interpolation in the drive** is used as the controlled axis ("ControlledAxis" input), the parameter used in the "ControlledValueIDN" input has to be one of the following supported P-parameters:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 and P-0-0695



Additionally, this P-parameter has to be configured in the MDT of the cyclic data channel in the drive.

If a **real axis with interpolation in the control** is used as a **virtual axis**, the parameter used in the "ControlledValueIDN" input has to be one of the following supported P-parameters:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

MLD parameterization

The parameter used as controlled parameter for the "ControlledValueIDN" (P-parameter) input has to be configured in the "AxisData" structure. Thus, the axis data structure has to be activated. The following P-parameters are supported:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

For the MLD firmware MPx20 (IL2G) and later, axis references which are not included in the "AxisData" structure are also supported at the "MeasuredAxis" (> VMA1).

In this case, the correction window is not determined using the current position of "MeasuredAxis", but via P-0-0761 "Master axis position for slave axis" of the local axis.

To provide the required position information in P-0-0761 "Master axis position for slave axis", the master axis generator has to be configured respectively.

Parameters involved:

- P-0-0760 "Virtual master axis, position control word"
- P-0-0916 "Master axis format converter signal selection"

See "Virtual master axis generator" in the documentation IndraDrive MPx-20 Functions, Application Description, DOK-INDRV*-MP*-20VRS**-APRS-EN-P.

Troubleshooting

The function block outputs the error messages of the following function blocks used.

- [MB_RegiMeasuringType01, page 296](#)
- Error codes of the IL_PIType02, RIL_LoopControl function block
RIL_LoopControl library (see documentation "Rexroth IndraWorks 14VRS Basic Libraries IndraLogic 2G", DOK-IL*2G*-BASLIB**V14-LIRS-EN-P, R911343919)
- [MB_RegiRegulateType02, page 313](#)
- [MB_SetpointLockType01, page 336](#)

The function block generates the following error messages in Additional1/Additional2 using the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"Axis_Ref" is not in the valid range (measured/controlled)
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: Error codes of the MB_RegisterControllerType02 function block

5.6.3 MB_RegisterControllerType04

Brief description

The MB_RegisterControllerType04 register controller function block supports the following functionality:

- Calculation of a correction value based on the measured value and the command value using a P- or PI-control loop
- Preset function
- Pause function
- Min./max. limitation of the calculated control value

- Correction window function
- Shift register
- Dead band Kp characteristic curve
- PT1 element
- Lock setpoint on demand
- Mark gating function

Target system	Library
MLC 14VRS Function package TEC	ML_TechRegi
MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib (IL1G)
MLD with MPx20 with MA function package	MX_TechRegi (IL2G)

Tab. 5-45: Reference table of the MB_RegisterControllerType04

Interface description

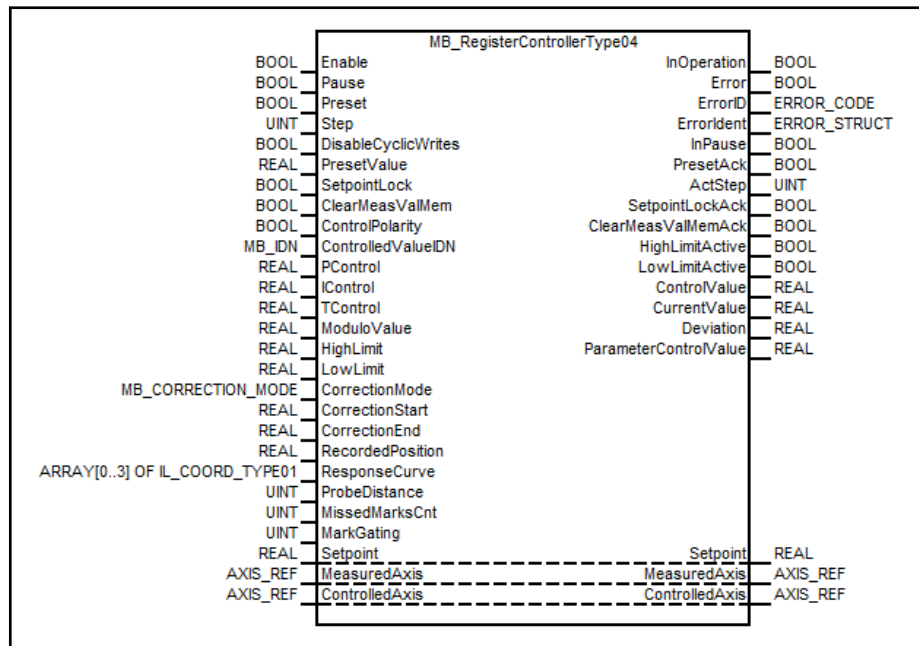


Fig. 5-37: MB_RegisterControllerType04 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enables the register controller
	Pause	BOOL	The "Pause" input is only evaluated when the "Enable" input is active. The "ControlValue" and "ParameterControlValue" outputs are constant when "Pause" is active
	Preset	BOOL	The "Preset" input is only evaluated when "Enable" is TRUE. A positive edge of this input forces the control loop to "PresetValue". In addition, the "Preset" input sets the "ParameterControlValue" to the "PresetValue"

I/O type	Name	Data type	Description
	Step	UINT	Processing step
	DisableCyclicWrites	BOOL	Disables cyclic writing to the selected parameter in "ControlledValueIDN" if TRUE
	PresetValue	REAL	Value applied to the "ControlValue" and "Parameter-ControlValue" outputs at the rising edge of the "Preset" input
	SetpointLock	BOOL	Enables command value application (setpoint lock) If TRUE, the next measured mark is latched
	ClearMeasValMem	BOOL	Clear the memory for the measured values used by the "touch probe distance" function
	ControlPolarity	BOOL	The polarity of the control value is inverted when this input is TRUE
	ControlledValueIDN	MB_IDN	Sercos IDN of the controlled parameter (e.g. P-0-0691). The following IDNs are supported: 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	Proportional gain of the PI-controller
	IControl	REAL	Integral action time T_n of the PI-controller
	TControl	REAL	Filter constant for the input "RecordedPosition"
	ModuloValue	REAL	Modulo value of the measured axis
	HighLimit	REAL	Maximum value for "ControlValue"
	LowLimit	REAL	Minimum value for "ControlValue"
	CorrectionMode	MB_CORRECTION_MODE	Properties of the correction window MB_CORRECTION_MODE, page 335
	CorrectionStart	REAL	Start of the correction window
	CorrectionEnd	REAL	End of the correction window
	RecordedPosition	REAL	Position where edge occurred (in technical units)
	ResponseCurve	ARRAY[0..3] OF IL_COORD_TYPE01	Characteristic curve
	ProbeDistance	UINT	The distance between marks sensor and controlled axis in format lengths (master axis revolutions, formats),
	MissedMarksCnt	UNIT	Missed mark counter when using an expectation window
	MarkGating	UINT	Every nth mark is used for the register
VAR_OUTPUT	InOperation	BOOL	Register controller is operating
	Error	BOOL	Indicates an error. Clear error with "Enable" = FALSE
	ErrorID	ERROR_CODE	Brief error description

I/O type	Name	Data type	Description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 359
	InPause	BOOL	"Pause" is active. The "ControlValue" and "ParameterControlValue" outputs maintain their values
	PresetAck	BOOL	"Preset" is done. The "Preset" value is copied to the "ControlValue" and "ParameterControlValue" outputs
	ActStep	UINT	Step value, incrementing with every new value at the "ControlValue" output
	SetpointLockAck	BOOL	"Setpoint" is locked. Current value in "RecordedPosition" is copied to "Setpoint"
	ClearMeasValMemAck	BOOL	Clearing the memory of the measured values completed
	HighLimitActive	BOOL	Upper limit is active. "ControlValue" would be higher, but is limited to "HighLimit"
	LowLimitActive	BOOL	Lower limit is active. "ControlValue" would be lower, but is limited to "LowLimit"
	ControlValue	REAL	Control value calculated by the register controller
	CurrentValue	REAL	Current value used in the register controller. It is equal to the input "RecordedPosition" if "ProbeDistance" and "MarkGating" are disabled
	Deviation	REAL	"Deviation" = ("Setpoint" - "CurrentValue") of the register controller
	ParameterControlValue	REAL	The register controller calculates a value based on the selected parameter in the "ControlledValueIDN" input and stores the result in the "ParameterControlValue" output. This value is written to the selected parameter input "ControlledValueIDN" when the "DisableCyclicWrites" input = FALSE
VAR_IN_OUT	Setpoint	REAL	Command value. This value is compared to "RecordedPosition"
VAR_IN_OUT CONSTANT	MeasuredAxis	AXIS_REF	Reference to the measured axis
VAR_IN_OUT CONSTANT	ControlledAxis	AXIS_REF	Reference to the controlled axis

Tab. 5-46: Interface variables of the MB_RegisterControllerType04 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous

Name	Type	Min. value	Max. value	Default value	Effective
Preset	BOOL			FALSE	"Enable" TRUE (continuous)
Step	UINT	0	65.535	0	Continuous. Starts at a rising edge at "Enable" and at a falling edge at "Pause"
DisableCyclicWrites	BOOL			FALSE	Rising edge at "Enable"
SetpointLock	BOOL			FALSE	Continuous
ClearMeasValMem	BOOL			FALSE	Continuous
PresetValue	REAL	N. def.	N. def.	0.0	Rising edge at "Preset"
ControlPolarity	BOOL			FALSE	Continuous
ControlledValueIDN	MB_IDN	N. def.	N. def.	0	Rising edge at "Enable"
Setpoint	REAL	N. def.	N. def.	0.0	Change of "Step"
PControl	REAL	0.0	N. def.	1.0	Continuous
IControl	REAL	0.0	N. def.	0.0	Continuous
TControl	REAL	0.0	N. def.	0.0	Continuous
ModuloValue	REAL	0.0	N. def.	0.0	Rising edge at "Enable"
HighLimit	REAL	LowLimit	N. def.	0.0	Continuous
LowLimit	REAL	N. def.	HighLimit	0.0	Continuous
CorrectionMode	MB_CORRECTION_MODE			CORR_REL	Continuous
CorrectionStart	REAL	0.0	ModuloValue	0.0	Continuous
CorrectionEnd	REAL	0.0	ModuloValue	0.0	Continuous
RecordedPosition	REAL	N. def.	N. def.	0.0	Change of "Step"
ResponseCurve	ARRAY[0..3] OF IL_COORD_TYPE_01	N. def.	N. def.	0.0	Rising edge at "Enable"
ProbeDistance	UINT	0	30	0	Continuous
MissedMarksCnt	UINT	0	65.535	0	Continuous. Initialization at a rising edge at "Enable" and falling edge at "Pause"
MarkGating	UINT	0	65.535	1	Rising edge at "Enable". "0" disables mark gating (identical to "1")

Tab. 5-47: Min./max. and default values of the inputs

Time diagram The following diagram shows the signal-time diagram of MB_RegisterControllerType04 including the "Pause" and "Preset" functions.

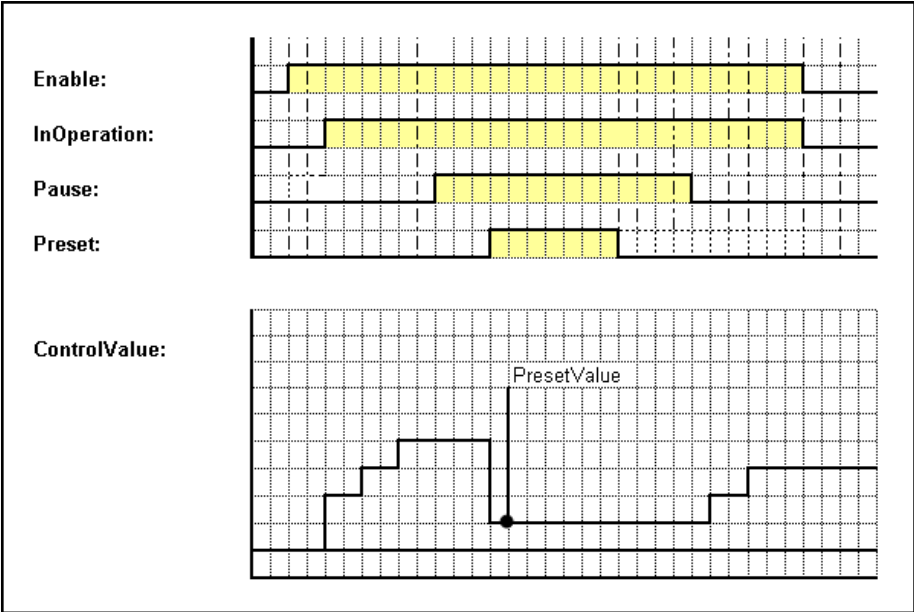


Fig. 5-38: Timing diagram of the MB_RegisterControllerType04 function block with the functions "Pause" and "Preset"

Functional description A sensor detects marks, perforations, cuts or pasted joints on the material and provides a (binary) signal to the drive.

The touch probe feature of the drive determines the edge of the sensor signal and latches the following positions, depending on the signal selection of the touch probe:

Actual drive position parameters	S-0-0051, S-0-0053, P-0-0753
Master axis position parameters	P-0-0052, P-0-0227, P-0-0775, P-0-0776, P-0-0778

The touch probe function records positional data with a resolution of 0.5 μ s. The sensor has to provide a 24V signal with a rise time within the μ s range. The sensor specific delay time can be compensated using the "DeadTimePosEdge" or "DeadTimeNegEdge" inputs of the MB_InitTouchProbe function block.

The register controller calculates the control deviation between the measured and setpoint positions and determines the required correction value once a new edge of the sensor signal is detected.

Changes of the correction value become active via the trapezoidal profile, velocity ramp, PT1-filter or instantaneously, depending on the selected control parameter ("ControlledValueIDN") input of the function block. The following table shows the available control parameters ("ControlledValueIDN") as well the resulting MotionProfile of all available drive operation modes with synchronization.

Parameter Name	Controlled-ValueIDN	Motion Behavior in case the parameter value is changed	Available in the following Operation Modes			
			Velocity-Synchronization	Phase-Synchronization	Cam Shaft	Electronic Motion Profile
Angle offset begin of profile	P-0-0061 A-0-2730	Changes of P-0-0061 become active via ramp using P-0-0158 (velocity). If P-0-0158=0, changes of P-0-0061 become active instantaneously.	No	No	Yes	Yes
Additive velocity command value, process loop	P-0-0690 A-0-2615	Changes of P-0-0690 become active instantaneously	Yes	No	No	No
Additive position command value, process loop	P-0-0691 A-0-2610	If P-0-0155 Bit0 = 0: Changes of P-0-0691 become active via PT1 filter using P-0-0060 (filter time constant). If P-0-0060=0, changes of P-0-0691 become active instantaneously. If P-0-0155 Bit0 = 1: Changes of P-0-0691 become active instantaneously	Yes	Yes	Yes	Yes
Additive master axis position, process loop	P-0-0692 A-0-2600	Changes of P-0-0692 is smoothed with a 1st order filter using P-0-0693(filter time constant) and then added to the sum of P-0-0054, Additive master axis position and P-0-0053, Master axis position or P-0-0052, Actual position value of measuring encoder.	Yes	Yes	Yes	Yes
Gear ratio fine adjust, process loop	P-0-0694 A-0-2605	Changes of P-0-0694 become active instantaneously	Yes	Yes	Yes	Yes
Angle offset begin of profile, process loop	P-0-0695 A-0-2620	Changes of P-0-0695 become active via PT1 filter P-0-0696 (filter time constant). If P-0-0696=0, changes of P-0-0695 become active instantaneously.	No	No	Yes	Yes

Fig. 5-39: Supported control parameter and behavior

The following "ControlledValueIDN" values are integrated:

- P-0-0061, A-0-2730
- P-0-0691, A-0-2610
- P-0-0692, A-0-2600
- P-0-0695, A-0-2620

Initialization During initialization, the "Setpoint" and "RecordedPosition" inputs must have valid data.

PresetValue For a trouble-free start of the controller outputs "ControlValue" and "ParameterControlValue", set the "Preset" input before activating the "Enable" input. During this period, the "Setpoint" and "RecordedPosition" inputs have to have valid data. In addition, the upper limits and lower limits are monitored to en-

sure a limitation of the "ControlValue" output if the current value for "PresetValue" is not in the valid range of "HighLimit" or "LowLimit".

The "PresetAck" output is active if the value in the "PresetValue" input is copied to the "ControlValue" and "ParameterControlValue" outputs. The "Preset" input has a higher priority than the "Pause" input. The value in "PresetValue" is copied to the respective outputs at the rising edge of the "Preset" input if "Enable" is active. If successful, the "PresetAck" output is set to TRUE.

Pause The "Pause" input is only evaluated as long as the "Enable" input is active. In this case, the deviation ("Setpoint" – "RecordedPosition") is set to 0. Therefore, the outputs "ControlValue" and "ParameterControlValue" do not change anymore. This can be used to maintain the values of the "ControlValue" and "ParameterControlValue" outputs. The "ActStep" output is updated even if the "Pause" input is active. As long as the "Pause" input is active, the "In-Pause" output is TRUE.

Lock Setpoint A manual to Automatic mode change is achieved by latching to the next register mark that is detected. For example, when a new web is pulled into position and the register mark is detected, the "SetpointLock" input is set to TRUE. The value in "Setpoint" is overwritten with the value in "RecordedPosition" the next time the "Step" input changes.

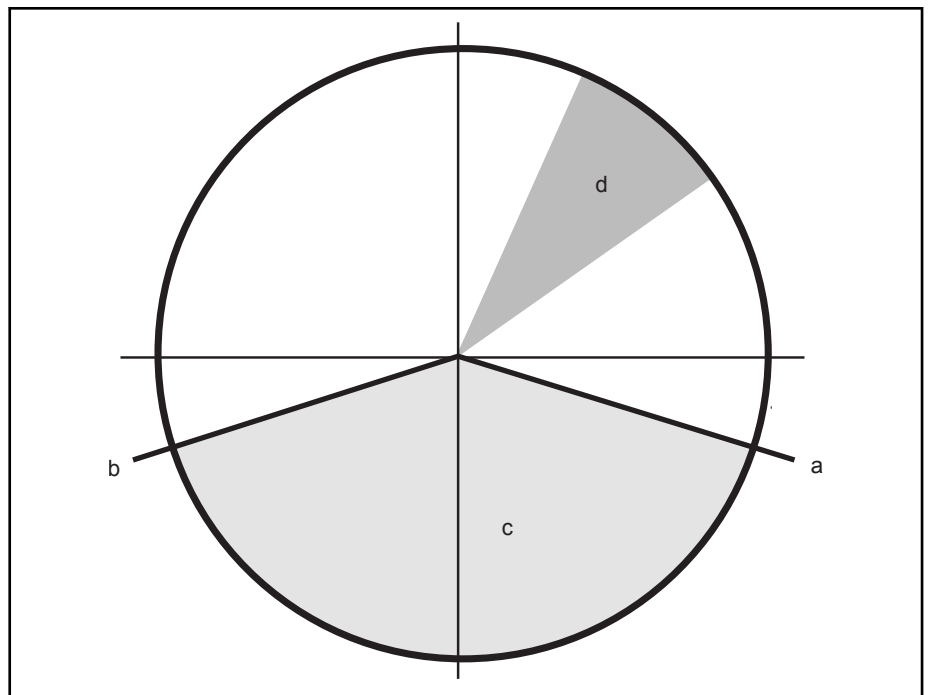
Correction window The correction window is an angular range relating to the measuring axis. The register controller controls the control deviations within this window. The position and size of the correction window are set with the help of the following inputs:

The "CorrectionMode" input sets the correction window either absolute or relative. If the "CorrectionMode" is set to CORR_WIN_DISABLED, the correction window is inactive.

The "CorrectionStart" input defines the start of the correction window.

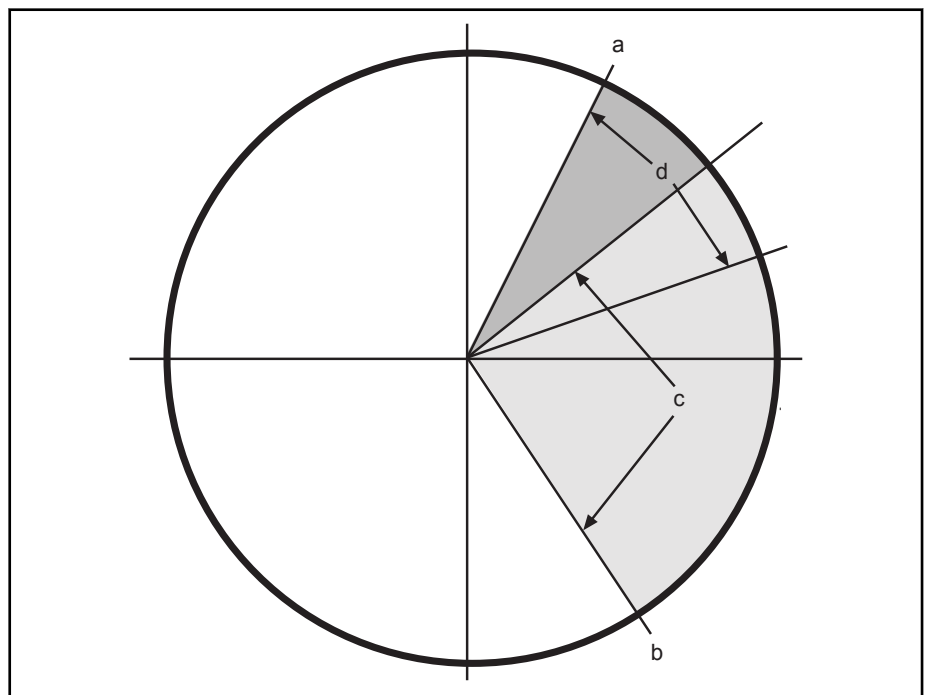
The "CorrectionEnd" input defines the end of the correction window.

The correction window supports only a positive modulo range (e.g., 0...+360 degrees) of the measuring axis.



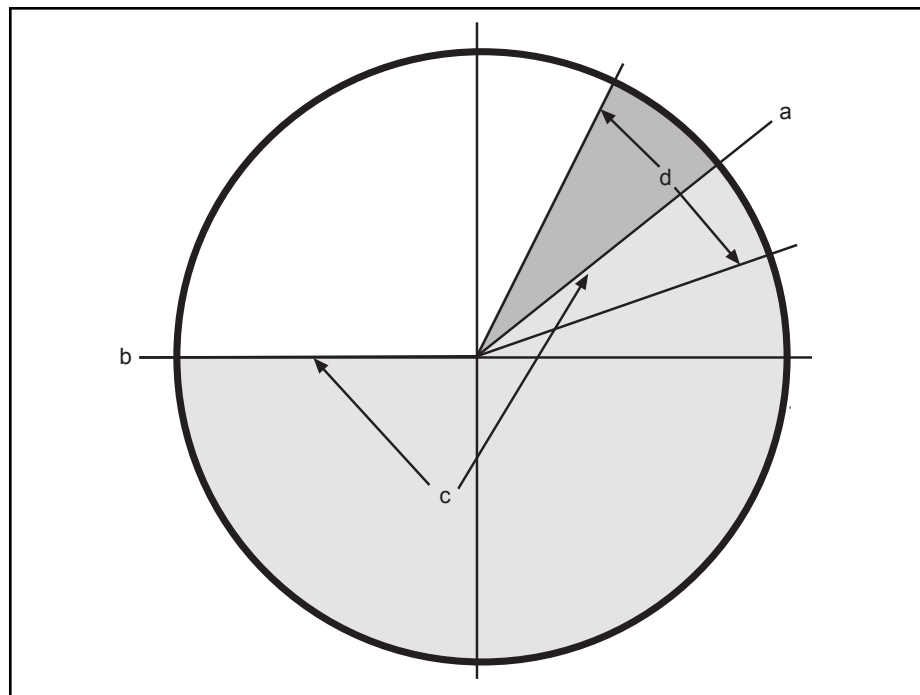
- a:** Start correction window
b: End correction window
c: Correction window
d: Expectation window

Fig. 5-40: Correction window absolute



- a:** Start correction window = measuring end
b: Relative end of correction window to measuring end
c: Correction window
d: Expectation window

Fig. 5-41: Correction window relative



- a:** Start correction window = measuring end
b: End of correction window = perm. angle
c: Correction window
d: Expectation window

Fig. 5-42: Correction window relative start absolute end

Smoothing The "TControl" input is the attenuation constant for the "RecordedPosition" input. If set to 0, smoothing is inactive.

ProbeDistance In the simplest case, the measuring axis is a format cylinder (= controlled axis) and the distance of the sensor to the measuring axis is less than the format length. Measurement and correction relate to the same product. Many times it is necessary to mount the mark reader (sensor) further away from the measuring axis, e.g., hazardous areas in printing facilities. If the distance is greater than the format length, then the measured register error cannot be corrected until the product has reached the format cylinder. The correction must be delayed by as many formats as there are formats between the controlled axis and the sensor. The "ProbeDistance" input is the distance between the mark reader (sensor) and the controlled axis in format lengths (master axis revolutions, formats). If set to 0, the distance of the sensor to the measuring axis is less than the format length.

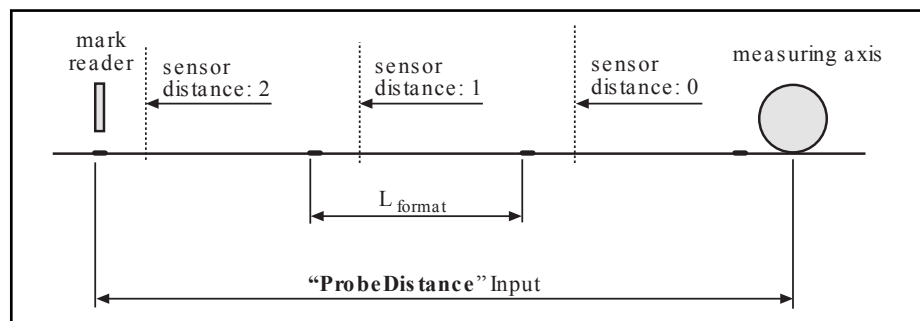


Fig. 5-43: Probe distance



When using the "ProbeDistance", the previously entered deviations into the shift register have to be adjusted each by 100% ($P\_Control = 1 / I\_Control = 0$).

That means: The new deviation entered is only the delta of the sum already entered into the shift register.

Clearing the internal memory of measured values

The touch probe distance feature contains an internal memory that stores measured values (like a shift register). This internal memory can be cleared (set to "0") using the "ClearMeasValMem" input.

ResponseCurve

Using the "ResponseCurve" input, the control deviation can be scaled. Use this functionality to specify a smaller scaling of the control deviation for smaller control deviations. This results in a lower proportional gain and thus causes a smoother control behavior in a steady state. When parameterizing the input, the controller parameters should first be set without evaluation. Subsequently, a lower scaling should be specified. This allows to globally set the controller parameters first using the system dynamics, and then, a local optimization is ensured (e.g. for the steady state).

The local optimization is parameterized with a characteristic curve with up to four data points. Negative deviations are processed by point reflecting the curve at the origin. This results in a total of nine linear sections. The section s5 runs through the origin. The sections s1 and s9 have a slope of 1 and each run into positive or negative infinity. If the input is not assigned, a 1-to-1 characteristic curve is used as default value. That means that it is not scaled.

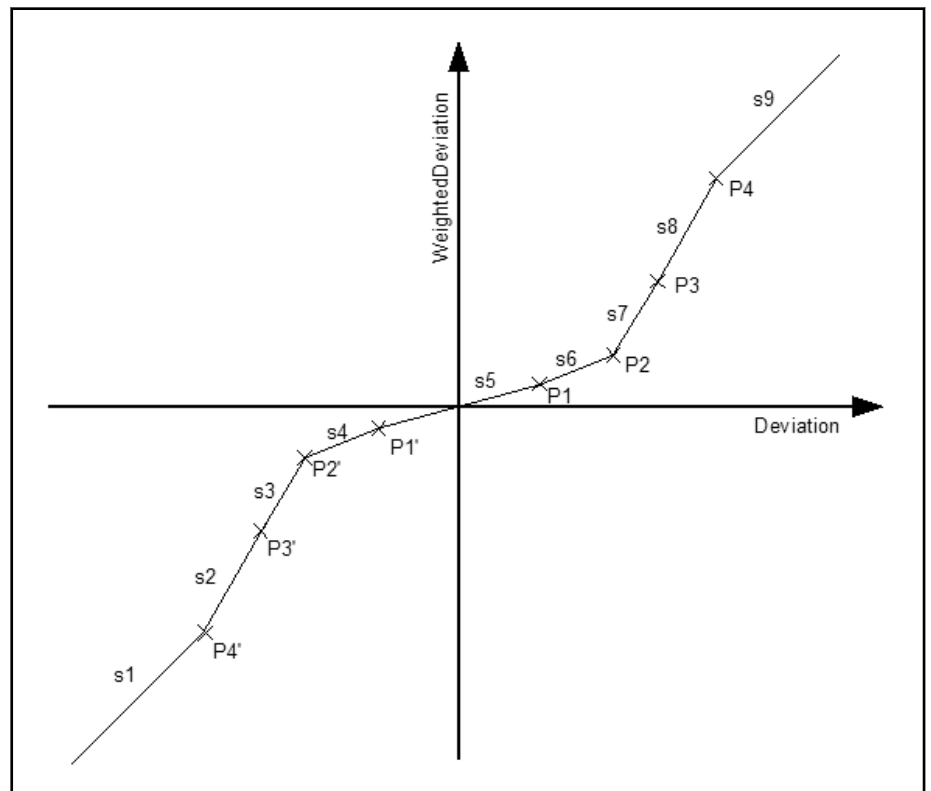


Fig. 5-44: Scaling the control deviation using the "ResponseCurve" input

Points P1 to P4 are arranged in an array made up of `IL_COORD_TYPE01`. If fewer than four points are required, the X-coordinates of the points not required (beginning with the last point) are set to 0. The X-values have to be

positive and rise in a positive direction. It is permitted to use the same point twice.

MarkGating

With this feature, the register deviation can only be determined at every n th mark, but the control step can be executed at each mark. This can be used e.g. if the marks on the web were printed with a print medium where the printed image is displayed several times. In this case, the mark distances between subsequent marks is not constant. In this situation, the mark distances between consecutive marks are not constant. The mark distances between every n th consecutive mark is constant. The register controller uses only every n th mark to determine the register deviation, but executes a control step at each mark to adjust the register deviation across " n " control steps.

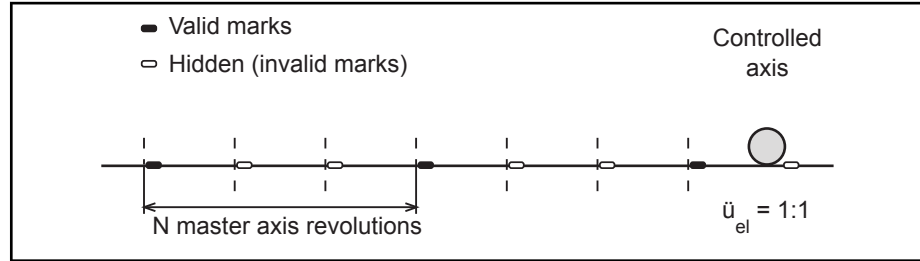


Fig. 5-45: MarkGating



As the register deviation is adjusted across " n " control steps, use the value $1/n$ for the proportional gain of the controller.

Normal mode

The RegisterController only calculates a new value if the "Step" input changes. The "PControl" and "IControl" controller parameters can be changed continuously.

If the value in the "ControlValue" output exceeds or falls below the values of outputs "HighLimit" or "LowLimit", then "ControlValue" is limited to one of the following limit values:

- If the value in "ControlValue" is positive, then the value in "HighLimit" is used.
- If the value in "ControlValue" is negative, then the value in "LowLimit" is used.

If the register controller has an integral action, then an anti-reset wind-up is carried out and the integral action is no longer summed. In this case the "HighLimitActive" and "LowLimitActive" outputs are set to TRUE.

Required parameterization

MLC parameterization

If a **real axis with interpolation in the drive** is used as the controlled axis ("ControlledAxis" input), the parameter used in the "ControlValueIDN" input has to be one of the following supported P-parameters:

- P-0-0690, P-0-0691, P-0-0692, P-0-0694 and P-0-0695



Additionally, this P-parameter has to be configured in the MDT of the cyclic data channel in the drive.

If a **real axis with interpolation in the control** is used as a **virtual axis**, the parameter used in the "ControlValueIDN" input has to be one of the following supported P-parameters:

- A-0-2730, A-0-2615, A-0-2610, A-0-2600, A-0-2605, A-0-2620

MLD parameterization

The parameter used as controlled parameter for the "ControlledValueIDN" (P-parameter) input has to be configured in the "AxisData" structure. Thus, the axis data structure has to be activated. The following P-parameters are supported:

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

For the MLD firmware MPx20 (IL2G) and later, axis references which are not included in the "AxisData" structure are also supported at the "MeasuredAxis" (> VMA1).

In this case, the correction window is not determined using the current position of "MeasuredAxis", but via P-0-0761 "Master axis position for slave axis" of the local axis.

To provide the required position information in P-0-0761 "Master axis position for slave axis", the master axis generator has to be configured respectively.

Parameters involved:

- P-0-0760 "Virtual master axis, position control word"
- P-0-0916 "Master axis format converter signal selection"

See "Virtual master axis generator" in the documentation IndraDrive MPx-20 Functions, Application Description, DOK-INDRV*-MP*-20VRS**-APRS-EN-P.

Troubleshooting

The function block outputs the error messages of the following function blocks used.

- [MB_RegiMeasuringType03, page 301](#)
- Error codes of the IL_PIType02, RIL_LoopControl function block
RIL_LoopControl library (see documentation "Rexroth IndraWorks 14VRS Basic Libraries IndraLogic 2G", DOK-IL*2G*-BASLIB**V14-LIRS-EN-P, R911343919)
- [MB_RegiRegulateType02, page 313](#)
- [MB_SetpointLockType01, page 336](#)

The function block generates the following error messages in Additional1/Additional2 using the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	"Axis_Ref" is not in the valid range (measured)
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: Error codes of the MB_RegisterControllerType04 function block

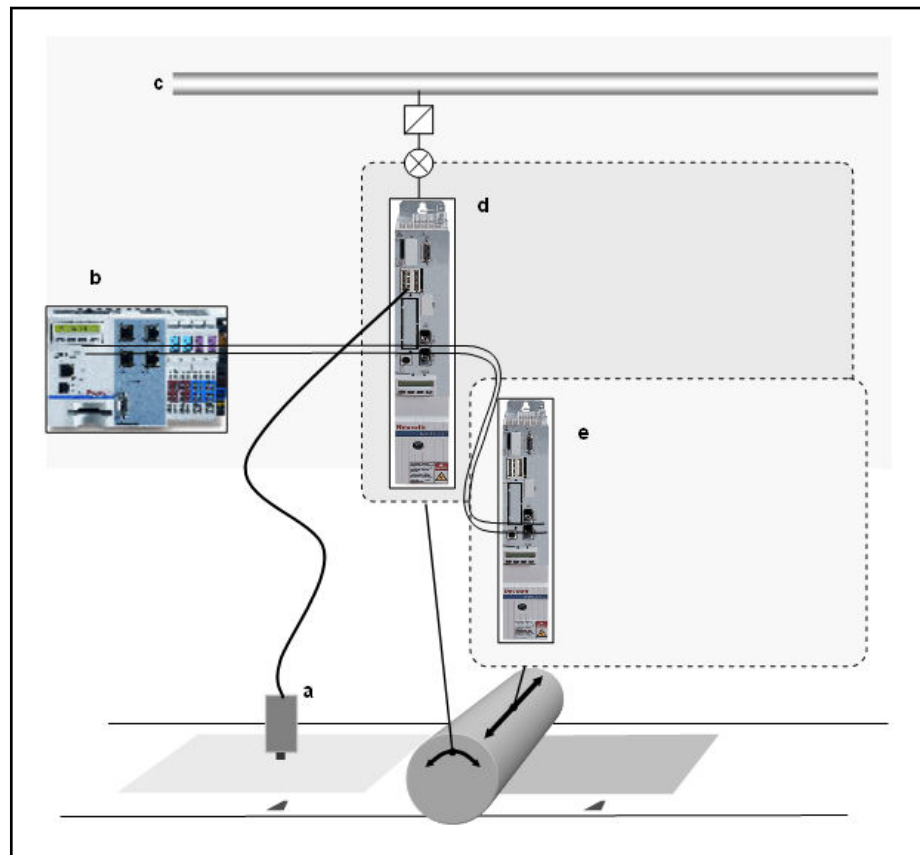
5.6.4 Configuring the Register Controller

Overview

The MB_RegisterControllerType04 function block is used for paper, printing, packaging and film applications which use synchronized drives (such as phase or cam synchronization) to feed material through a machine.

In these applications, variations in the characteristics of the material, material slippage and the production process affect the accuracy of the material position. A sensor determines the current position of registration marks, mounted or printed on the material and the register controller measures the deviation to the command value (setpoint). It calculates the required correction value

using a control loop. Using this method, the registration marks remain precisely on their command value and do not drift. The following figure shows an example of the implementation of a register controller:



a Sensor
b: MLC control (closed-loop)
c Master axis (virtual axis)
d Register control, touch probe function, phase synchronization

Fig. 5-46: Example of a register controller on a MLC-based system

Using the Register Control

Overview

The fastest option to use the register control functionality in an application is to combine the register controller function block with a touch probe function block. The touch probe function block provides the actual value to the register controller. The register controller calculates the correction value based on the actual deviation between the command value and the actual value.

The example in this section shows the MB_RegisterControllerType04 register controller function block with the MB_TouchProbeContinuous function block.

Configuring the Touch Probe Function Block

The following touch probe signals have to be configured using the MB_Init-TouchProbe function block:

1. Drive touch probe 1
2. Positive touch probe 1
3. Marker failure enabled
4. Expected window enabled



For a description of the touch probe function block, refer to [MB_InitTouchProbe, page 120](#).

Configuring the Touch Probe and RegisterControllerType04 Function Blocks

The following figure shows the connection of the [MB_TouchProbeContinuous, page 131](#), and [MB_RegisterControllerType04, page 347](#), function blocks when triggering the touch probe function using a positive edge.

As part of the [MB_InitTouchProbe, page 120](#), function block configuration, the marker failure and the expectation window are enabled. Thus, the counter for marker failures have to be connected as well.

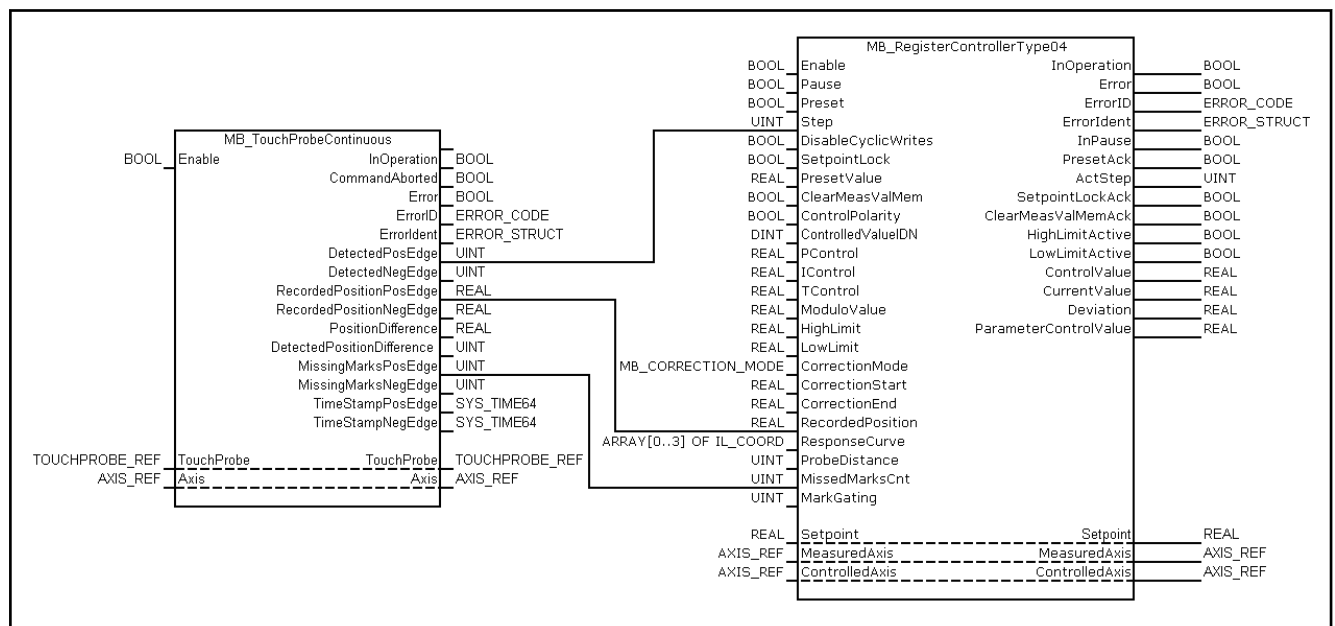


Fig. 5-47: *MB_TouchProbeContinuous and MB_RegisterControllerType04 interface*

Refer to the relevant function block descriptions for a complete description of the inputs and outputs.

The following signal connections are described in the figure above:

1. "DetectedPosEdge" to "Step"
2. "RecordedPositionPosEdge" to "RecordedPosition"
3. "MissingMarksPosEdge" to "MissedMarksCnt"

This connection is only necessary if the expectation window and the counter for marker failures are used.

Parameterizing the Register Controller

Axis Scaling

To use the register control functionality, both axes (measured axis, controlled axis) have to be set to rotary modulo format. The IndraWorks menu selection is as follows:

- For the MLD, select **IndraWorks** ► **Axis** ► **Scaling/Mechanical System** ► **Scaling/Units**.
- For the MLC, select **IndraWorks** ► **Motion** ► **Virtual Axes/Real Axes** ► **Axis** ► **Scaling/Mechanical System** ► **Scaling/Units**.

The following figure is an example:

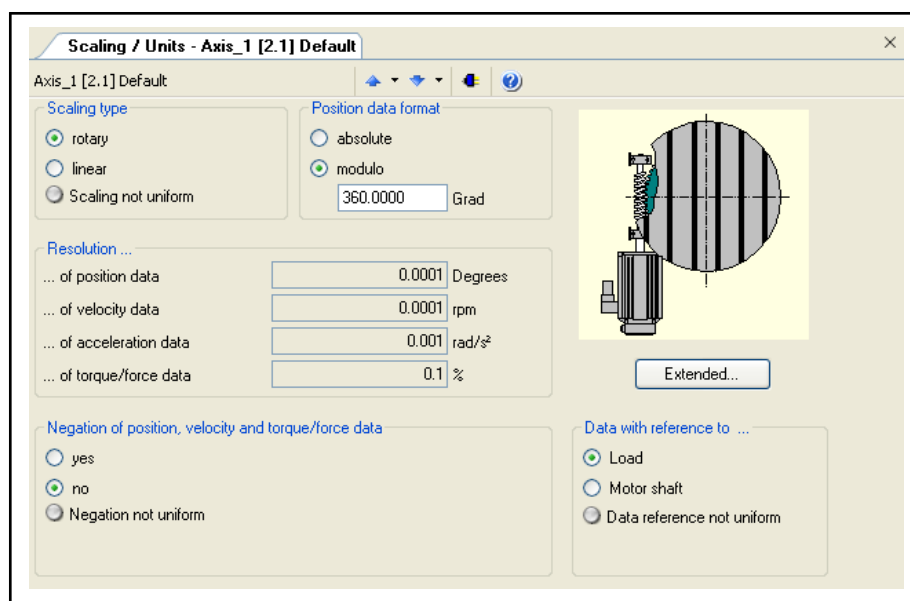


Fig. 5-48: Setting the axis scaling

Configuring the MLC Master Data Telegram (MDT)

The parameter used for the "ControlledValueIDN" input has to be configured in the MDT of the controlled axis. From the relevant IndraWorks project, select **IndraWorks** ► **Motion**, right click on the controlled axis and select **Communication** ► **Cyclical sercos Channel**.

The following figure is an example:

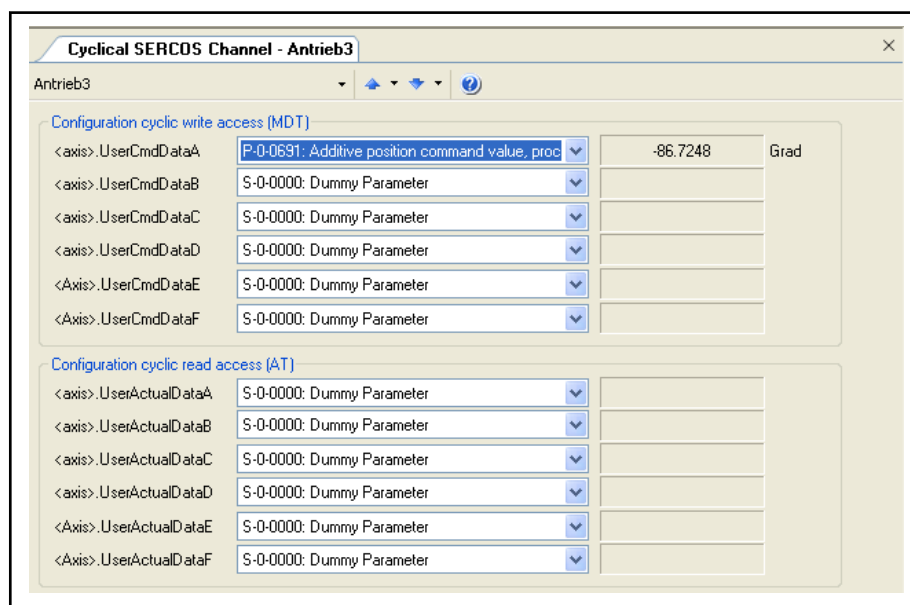


Fig. 5-49: MDT configuration of the controlled axis (MLC)

MLD Axis Data Configuration

The parameter used for the "ControlledValueIDN" input has to be configured in the AxisData structure of axis. From the relevant IndraWorks project, select **IndraWorks** ► **MLD** ► **Configuration** and click on **AxisData structure support**.

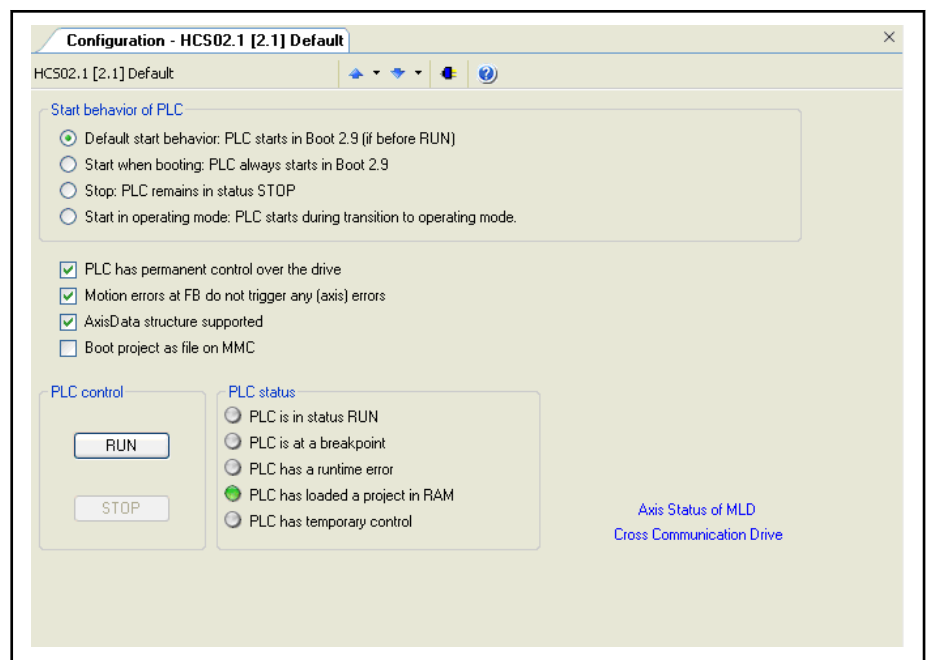


Fig. 5-50: MLD configuration

Subsequently, select **IndraWorks** ► **MLD** ► **Configuration** ► **AxisData** and select the parameter used for the "ControlledValueIDN" input.

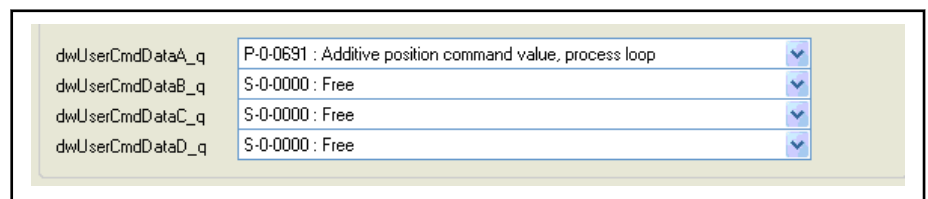


Fig. 5-51: AxisData configuration

5.6.5 Register Controller - Use Cases

Punching in Label Printing

The following example describes a punching process on a label printing machine:

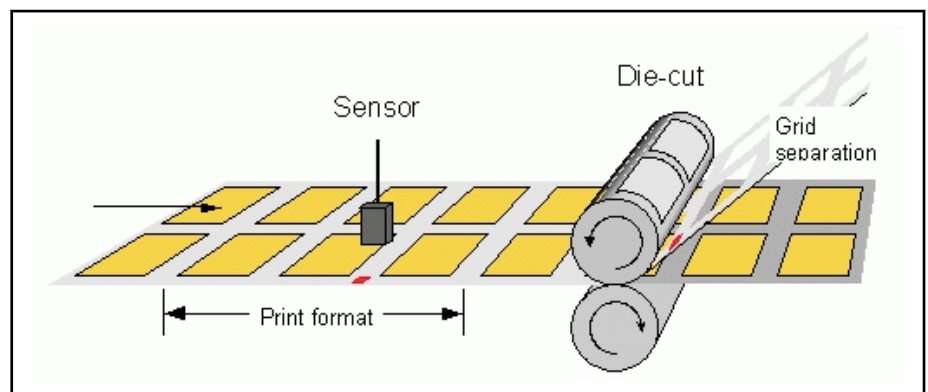


Fig. 5-52: Closed-loop register control of a punching device

Labels are printed on a composite material including a backing foil and a self-adhesive top layer. The unprinted grid is stamped and rewound.

The task of the register controller is to adjust the punching cylinder on the printed web.

The punching cylinder runs in phase-synchronized mode. The adjustment with regard to the printed web is accomplished by adjusting the phase offset. The sensor is connected with the drive of the punching cylinder. The angular position (the actual position value) of the punching cylinder is determined via the sensor signal.

The positioning command for the register controller corresponds to the cylinder position which is exactly aligned to the product. The correction value is calculated based on the difference between the reference value and the actual value and by adding this value to the current phase offset.

The most important settings for the touch probe, the register controller and the drive operation mode of this application are:

- Measured value (touch probe function) = Actual position value encoder 1 (S-0-0051)
- Controlled value (ControlledValueIDN) = additive position command value (P-0-0691)
- Drive operation mode = phase synchronization

Closed-Loop Insetter Control

The following figure shows a closed-loop insetter control in a material infeed application:

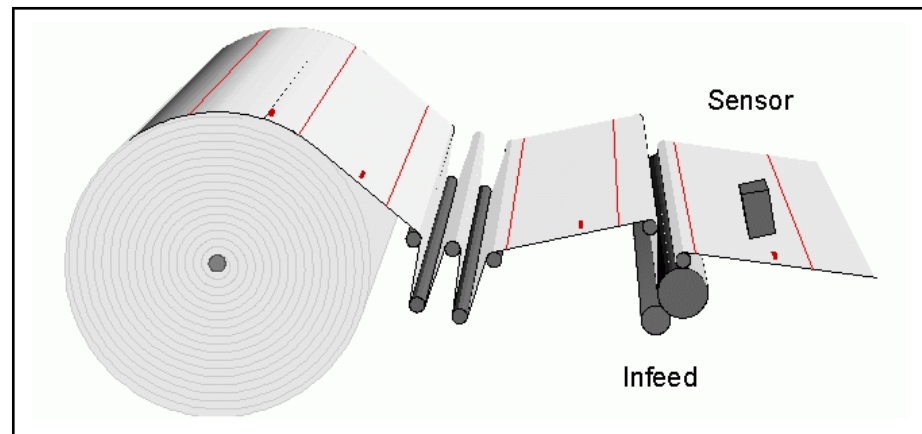


Fig. 5-53: Closed-loop insetter control of the material infeed

The material infeed unit transports preprinted material to a printing press. After the preprinting and drying process, the web has shrunk, i. e. the format length does not exactly correspond to the nominal format. The web has to be elongated again for further processing.

The task of the register controller is to control the velocity of the material infeed rollers so that the preprinted web is elongated to the nominal paper format. This implies that one paper format is transported during each revolution of the master axis.

The positioning command of the register controller is related to the master axis.

The material infeed roller runs in the velocity synchronization mode. The velocity of the infeed roller is controlled via the gear ratio fine adjustment. The elongation of the web is correct if the printed mark is always detected in the same master axis position by the mark reader. The master axis position is detected by the sensor signal. The mark reader is connected with the touch probe input of the material infeed roller drive.

The most important settings for the touch probe, the register controller and the drive operation mode of this application are:

- Measured value (touch probe function) = resulting master axis position (P-0-0775)
- Controlled value (ControlledValueIDN) = gear ratio fine adjustment (P-0-0694)
- Drive operation mode = speed synchronization

Flow Wrapper

The following figure shows a sidesseal axis (foil infeed axis) of a flow wrapper application:

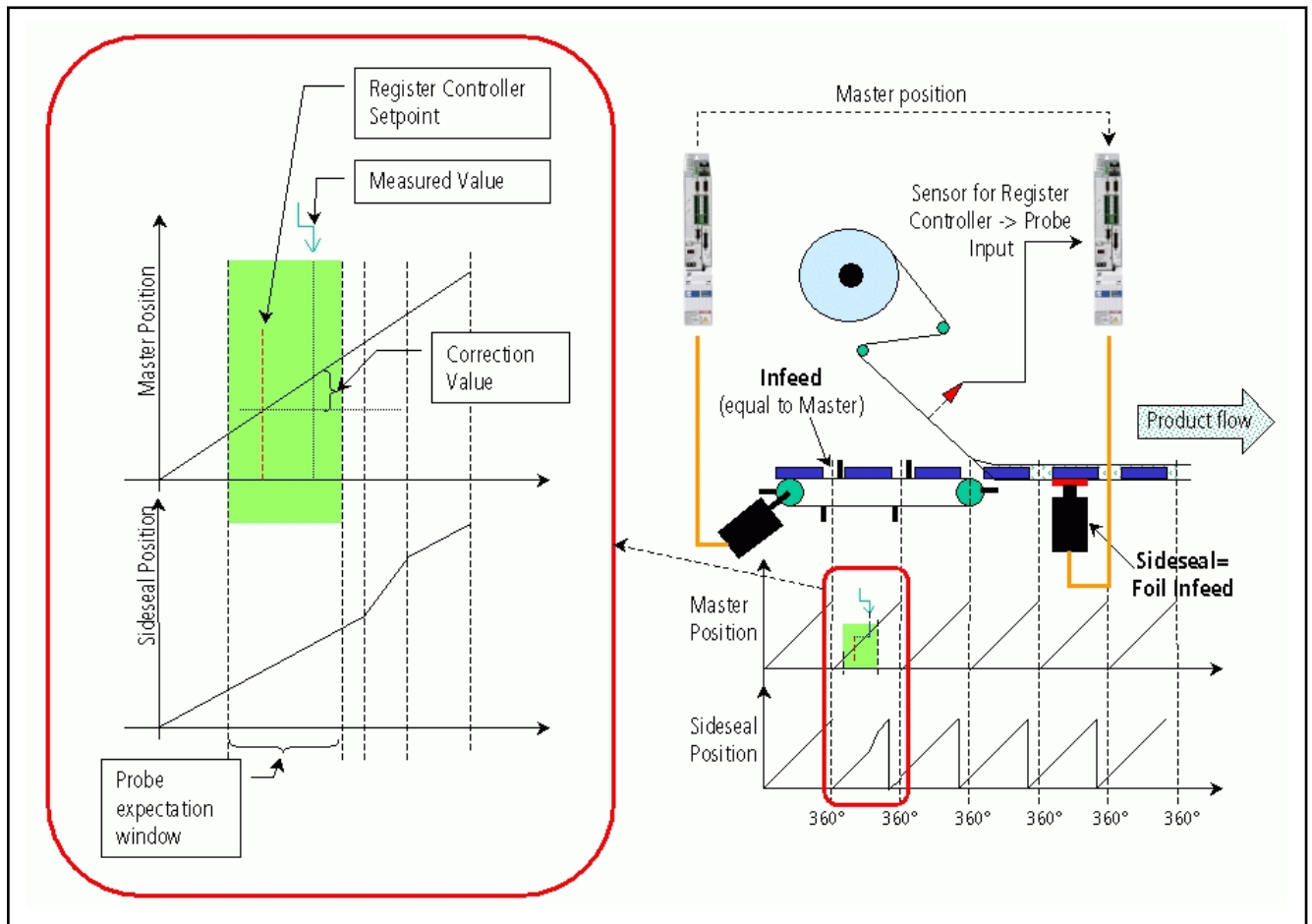


Fig. 5-54: Foil infeed axis of a flow wrapper application

In this application, the product is fed into the packaging machine with the help of an infeed belt. The foil infeed axis (sidesseal axis) runs synchronously with the infeed belt and pulls in the foil. During this process, the correct alignment of the foil to the product as well as an equal product length has to be ensured.

Slippage between the material and the infeed rollers as well as inaccuracies of the material can result in positioning errors between the servo motor and the foil. Thus, register marks are printed on the material allowing a correction of positioning errors.

Prints, pictures and dirt on the product can result in unwanted signals at the sensor input of the printed mark. Thus, an expectation window for the printed mark signal is required. The drive touch probe function is only active in the expectation window to measure the printed mark position. Furthermore, it is required to detect missing marks in the expectation window as well.

The correction motion is proportional to the difference between the measured value and the command value. Due to mechanical limits of the machine, the correction motion has to be limited to a user-defined value.

The register controller has to adjust the foil infeed axis to the printed foil to adjust the foil to the product position.

The foil infeed axis runs in the phase synchronization mode. The register controller calculates an additive position command value for the foil infeed axis by which the foil is correctly readjusted.

The most important settings for the touch probe, the register controller and the drive operation mode of this application are:

- Measured value (touch probe function) = resulting master axis position (P-0-0775)
- Controlled value (ControlledValueIDN) = additive position command value (P-0-0691)
- Drive operation mode = phase synchronization

5.7 Side register controller function block

5.7.1 Introduction and overview

The side register controller function block is part of the libraries ML_TechRegi (MLC) and MX_TechRegi.lib (MLD).

5.7.2 MB_RegisterControllerSideType01

Brief description The MB_RegisterControllerSideType01 register controller function block supports the following functionality:

- Calculation of a correction value based on the measured values and the command values using a P- or PI-control loop
- Preset function
- Pause function
- Min./max. limitation of the calculated control value
- Setpoint lock feature
- Dead band K_p characteristic curve
- PT1 element

Target system	Library
MLC 14VRS Function package TEC	ML_TechRegi
MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-49: Reference table of the MB_RegisterControllerSideType01

Interface description

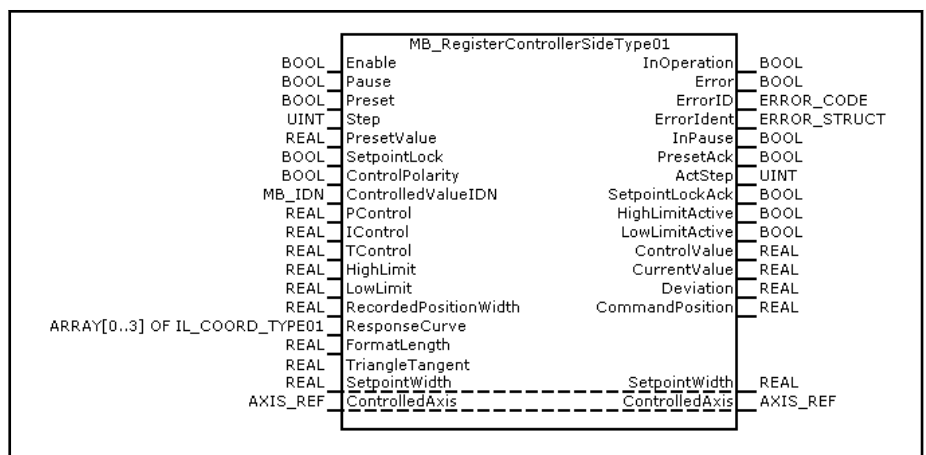


Fig. 5-55: MB_RegisterControllerSideType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	The function block is enabled as long as the input is TRUE
	Pause	BOOL	The "Pause" input is only evaluated when the "Enable" input is active. The "ControlValue" and "CommandPosition" outputs are constant while "Pause" is active
	Preset	BOOL	The "Preset" input is only evaluated when "Enable" is TRUE. A positive edge of this input forces the control loop to "PresetValue"
	Step	UINT	Processing step
	PresetValue	REAL	Value applied to the "ControlValue" output at the rising edge at the "Preset" input
	SetpointLock	BOOL	Enables command value application (setpoint lock) If TRUE, the next measured mark is latched
	ControlPolarity	BOOL	The polarity of the control value is inverted when this input is TRUE
	ControlledValueIDN	MB_IDN	The following control IDN is supported: S-0-0258
	PControl	REAL	Proportional gain of the PI-controller. If set to 0, the proportional part of the controller is disabled and the proportional gain is set internally to 1 to work as an I-controller
	IControl	REAL	Integral action time T_n of the PI-controller. If set to 0, the integral part of the PI-controller is disabled
	TControl	REAL	Filter constant for the "RecordedPositionWidth" input
	HighLimit	REAL	Maximum value for "ControlValue"
	LowLimit	REAL	Minimum value for "ControlValue"
	RecordedPosition-Width	REAL	Width of the measured mark (in technical units)
	ResponseCurve	ARRAY[0..3] OF IL_COORD_TYPE01	Characteristic curve

I/O type	Name	Data type	Description
	FormatLength	REAL	Format length in mm
	TriangleTangent	REAL	Value of $\tan \alpha = (\text{opposite leg length} / \text{adjacent leg length})$
VAR_OUTPUT	InOperation	BOOL	Side register controller is operating
	Error	BOOL	Indicates an error Clear error with "Enable" = FALSE
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 372
	InPause	BOOL	"Pause" is active. The "ControlValue" and "Command" outputs maintain their values
	PresetAck	BOOL	"Preset" is done. The "Preset" value is copied to the "ControlValue" output
	ActStep	UINT	Step value, incrementing with every new value at the "ControlValue" output. Note: The process controller is paused and the "Act-Step" output still increments with every new value at the "Step" input.
	SetpointLockAck	BOOL	Setpoint is locked. The current value in "RecordedPositionWidth" is copied to "SetpointWidth"
	HighLimitActive	BOOL	Upper limit is active. "ControlValue" would be higher, but is limited to "HighLimit"
	LowLimitActive	BOOL	Lower limit is active. ControlValue would be lower, but is limited to "LowLimit"
	ControlValue	REAL	Control value calculated by the register controller
	CurrentValue	REAL	Recorded position currently used in the register controller side. Note: Converted "RecordedPositionWidth" value.
	Deviation	REAL	Difference of the command value side (converted "SetpointWidth") - "CurrentValue" of the side register controller (mm)
	CommandPosition	REAL	Calculated command position for the drive
VAR_IN_OUT	SetpointWidth	REAL	Setpoint width value (in technical units)
VAR_IN_OUT CONSTANT	ControlledAxis	AXIS_REF	Reference to the controlled axis

Tab. 5-50: Interface variables of the MB_RegisterControllerSideType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous

Name	Type	Min. value	Max. value	Default value	Effective
Preset	BOOL			FALSE	"Enable" TRUE (continuous)
Step	UINT	0	65335	0	Continuous. Starts at a rising edge at "Enable" and at a falling edge at "Pause"
PresetValue	REAL	n.def.	n.def.	0.0	Rising edge at "Preset"
SetpointLock	BOOL	n.def.	n.def.	FALSE	Continuous
ControlPolarity	BOOL			FALSE	Continuous
ControlledValueIDN	MB_IDN	n.def.	n. def.	0	Rising edge at "Enable"
PControl	REAL	0.0	n.def.	1.0	Continuous
IControl	REAL	0.0	n.def.	0.0	Continuous
TControl	REAL	0	n.def.	0.0	Continuous
HighLimit	REAL	LowLimit	n.def.	0.0	Continuous
LowLimit	REAL	n.def.	HighLimit	0.0	Continuous
RecordedPositionWidth	REAL	n.def.	n.def.	0.0	Change of "Step"
ResponseCurve	ARRAY[0..3] OF IL_COORD	n.def.	n.def.	0.0	Continuous
FormatLength	REAL	0.0	n.def.	0.0	Rising edge at "Enable"
TriangleTangent	REAL	0.0	n.def.	1.0	Rising edge at "Enable"

Tab. 5-51: Min./max. and default values of the inputs

Time diagram

The following diagram shows the signal-time diagram of the MB_Register-ControllerType04 function block including the "Pause" and "Preset" functions.

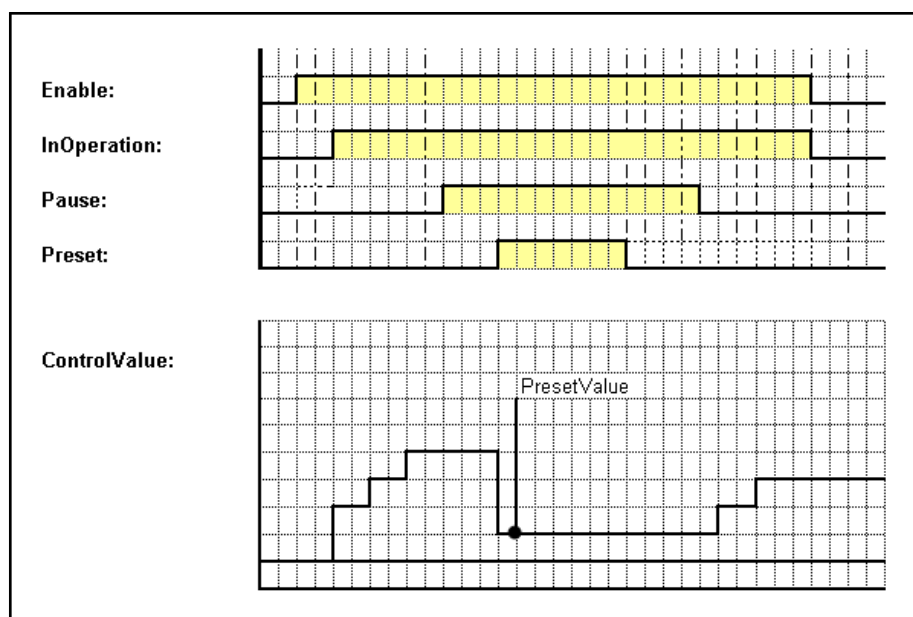


Fig. 5-56: Timing diagram of the MB_RegisterControllerSideType01 with the functions "Pause" and "Preset"

Functional description

- Initialization** During initialization, the "SetpointWidth" and "RecordedPositionWidth" inputs have to have valid data.
- PresetValue** For a trouble-free start of the controller outputs "ControlValue" and "CommandPosition", set the "Preset" input before activating the "Enable" input. During this period, the "SetpointWidth" and "RecordedPositionWidth" inputs have to have valid data. In addition, the upper limits and lower limits are monitored to ensure a limitation of the "ControlValue" output if the current value for "PresetValue" is not in the valid range of "HighLimit" or "LowLimit".
- The "PresetAck" output is active if the value in the "PresetValue" input is copied to the "ControlValue" output. The "Preset" input has a higher priority than the "Pause" input. The value in "PresetValue" is copied to the respective output at the rising edge at the "Preset" input if "Enable" is active. If successful, the "PresetAck" output is set to TRUE.
- Pause** The "Pause" input is only evaluated when the "Enable" input is active. In this case, the deviation ("Setpoint" - "RecordedPosition") is set to 0. Therefore, the outputs "ControlValue" and "CommandPosition" do not change anymore. This can be used to maintain the values of the "ControlValue" and "CommandPosition" outputs. The "ActStep" output is updated even if the "Pause" input is active. As long as the "Pause" input is active, the "InPause" output is TRUE.
- Smoothing** The "TControl" input is the attenuation constant for the "RecordedPositionWidth" input. If set to 0, smoothing is inactive.
- ResponseCurve** Using the "ResponseCurve" input, the control deviation can be scaled. Use this functionality to specify a smaller scaling of the control deviation for smaller control deviations. This results in a lower proportional gain and thus causes a smoother control behavior in a steady state. When parameterizing the input, the controller parameters should first be set without evaluation. Subsequently, a lower scaling should be specified. This allows to globally set the controller parameters first using the system dynamics, and then, a local optimization is ensured (e.g. for the steady state).

The local optimization is parameterized with a characteristic curve with up to four data points. Negative deviations are processed by point reflecting the curve at the origin. This results in a total of nine linear sections. The section s5 runs through the origin. The sections s1 and s9 have a slope of 1 and each run into positive or negative infinity. If the input is not assigned, a 1-to-1 characteristic curve is used as default value. That means that it is not scaled.

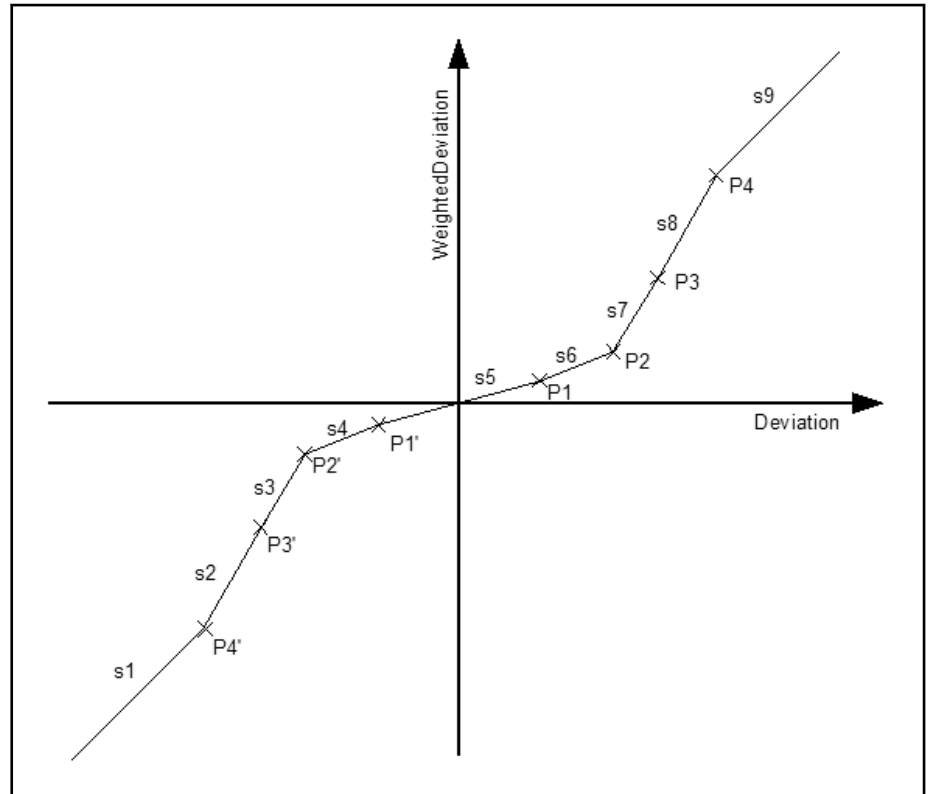


Fig. 5-57: *Scaling the control deviation using the "ResponseCurve" input*

Points P1 to P4 are arranged in an array made up of IL_COORD_TYPE01. If fewer than four points are required, the X-coordinates of the points not required (beginning with the last point) are set to 0. The X-values have to be positive and rise in a positive direction. It is permitted to use the same point twice.

FormatLength	Format length (master axis revolutions, formats) in mm.
TriangleTangent	The mark length between opposite leg and adjacent leg ($\tan \alpha$).

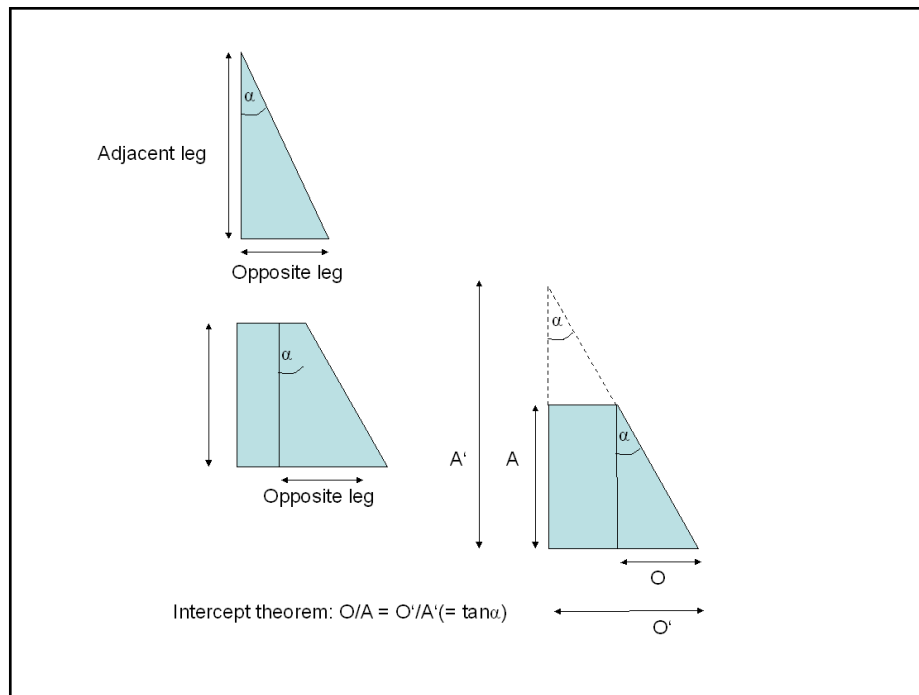


Fig. 5-58: Right triangle mark and conical frustum mark

Normal mode

The register controller side only calculates a new value if the "Step" input changes. The "PControl" and "IControl" and "TControl" controller parameters can be changed permanently.

If the value in the "ControlValue" output exceeds or falls below the values of outputs "HighLimit" or "LowLimit", "ControlValue" is limited to one of the following limit values:

- If the value in "ControlValue" is positive, the value in "HighLimit" is used
- If the value in "ControlValue" is negative, the value in "LowLimit" is used

If the register controller side has an integral action time, an anti-reset wind-up is carried out and the integral action is no longer summed. In this case, the "HighLimitActive" and "LowLimitActive" outputs are set to TRUE.

Troubleshooting

The function block outputs the error messages of the following function blocks used

- [Error codes of the MB_RegiMeasuringType01 function block, page 298](#)
- IL_PType02, RIL_LoopControl
RIL_LoopControl library (see documentation "Rexroth IndraWorks 14VRS Basic Libraries IndraLogic 2G", DOK-IL*2G*-BASLIB**V14-LIRS-EN-P, R911343919)
- [Error codes of the MB_RegiRegulateType01 function block, page 312](#)
- [Error codes of the MB_RegiSetpointLockType01 function block, page 338](#)

The function block generates the following error messages in Additional1/ Additional2 using the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
RESOURCE_ERROR (16#0003)	16#0004	16#0000	Drive firmware is not supported
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	Invalid function block state

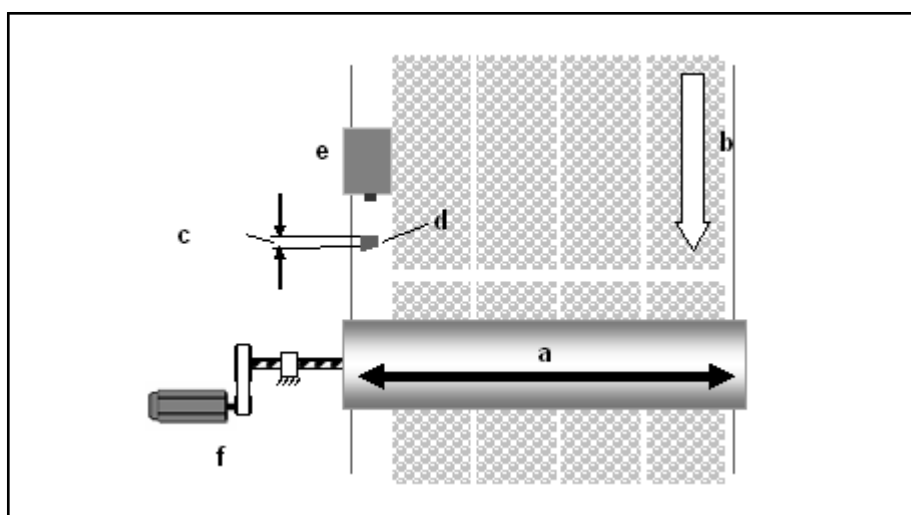
ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#000D	16#0000	AxisNo of "ControlledAxis" is not in the valid range
RESOURCE_ERROR (16#0003)	16#0001	16#0000	Axis has no power "ControlledAxis"
RESOURCE_ERROR (16#0003)	16#0009	16#0000	"Axis_Ref" has changed "ControlledAxis"
INPUT_INVALID_ERROR (16#0001)	16#0812	16#0001	Selected "ControlledValueIDN" is not supported
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	Input "SetpointWidth" < 0
INPUT_RANGE_ERROR (16#0006)	16#0816	16#0002	Input "RecordedPositionWidth" < 0
INPUT_RANGE_ERROR (16#0006)	16#0816	16#0003	Input "TriangleTangent" is not in valid range
INPUT_RANGE_ERROR (16#0006)	16#0816	16#0004	Input "FormatLength" is not in the valid range
INPUT_INVALID_ERROR (16#0001)	16#0C01	16#0001	"LowLimit" > "HighLimit"
INPUT_INVALID_ERROR (16#0001)	16#0C01	16#0002	Controller type selection is not supported. Check the inputs "IControl" and "PControl"
INPUT_RANGE_ERROR (16#0006)	16#0C02	16#0001	"TControl" input is invalid
INPUT_RANGE_ERROR (16#0006)	16#0C02	16#0003	"IControl" input is invalid
INPUT_RANGE_ERROR (16#0006)	16#0C02	16#0004	"PControl" input is invalid
INPUT_INVALID_ERROR (16#0001)	16#0C05	16#0001	X-value in "ResponseCurve" not in ascending order
INPUT_INVALID_ERROR (16#0001)	16#0C05	16#0002	More than one value for one x-position in "ResponseCurve"
INPUT_INVALID_ERROR (16#0001)	16#0C05	16#0003	Negative x-value in "ResponseCurve"
ACCESS_ERROR (16#0004)	16#0815	16#0008	The controlled axis is a MLD-M slave axis, but the parameter P-0-1367, bit 6 is not TRUE

Tab. 5-52: Error codes of MB_RegisterControllerSideType01 function block

5.7.3 Side Register Configuration

Overview The side register is a function used for the axial adjustment of rotating cylinders. A typical example is the adjustment of a printing cylinder for a preprinted web.

The triangle mark is the axial reference on the printed image. A optical sensor registers the incoming and outgoing edge of the mark. The difference between these two edges is the width of the mark. This value is the actual value of the side register. The difference between the actual width of the mark and setpoint is converted into the target position of the servo motor.

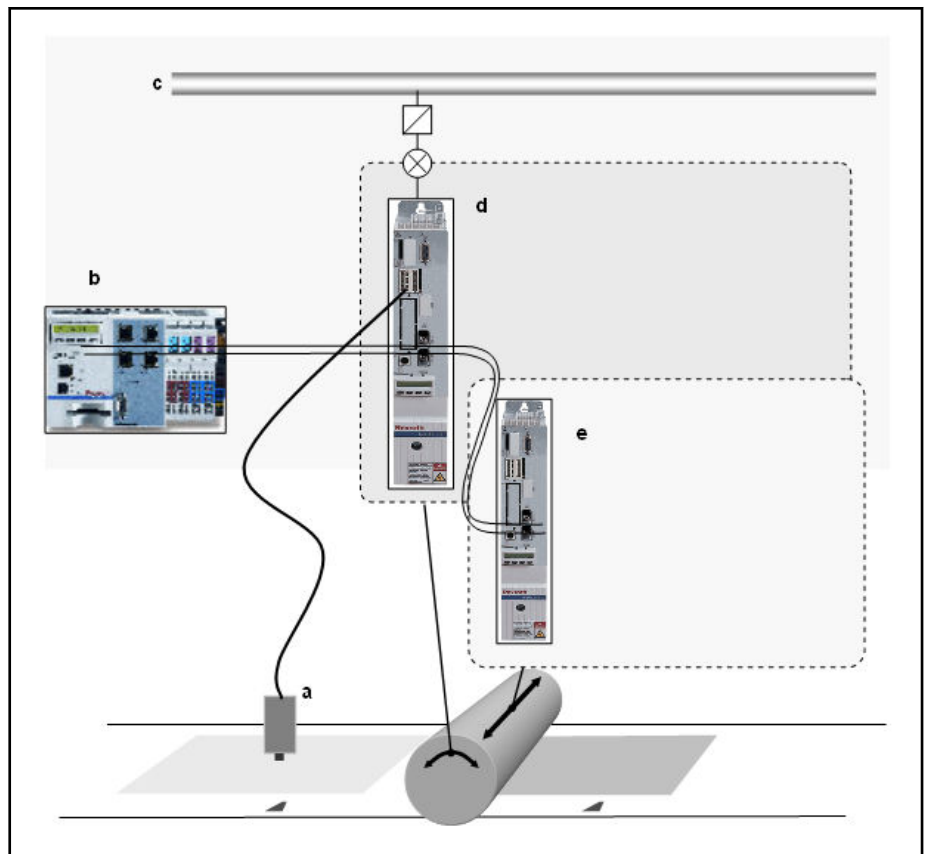


- a:** Side register
- b:** Web direction
- c:** Measured value
- d:** Registration mark
- e:** Sensor
- f:** Servo motor

Fig. 5-59: Side Register Configuration

Using Side Register Control

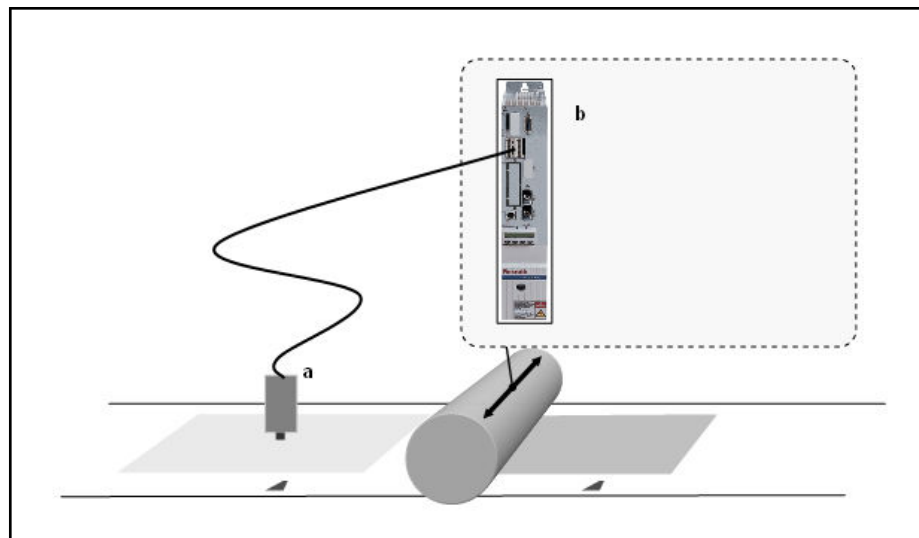
The following picture shows a possible configuration of a side register controller for a MLC system.



- a:** Sensor
- b:** MLC
- c:** Master axis
- d:** Register controller, touch probe function, phase synchronization
- e:** Side register controller, absolute positioning

Fig. 5-60: Register controller and side register controller overview on the MLC

The following figure shows the configuration of a side register controller for an MLD system.



- a:** Sensor
b: MLD: probe function, side register controller, absolute positioning

Fig. 5-61: Side register controller overview on MLD



The axial axis has to be an absolute linear axis.

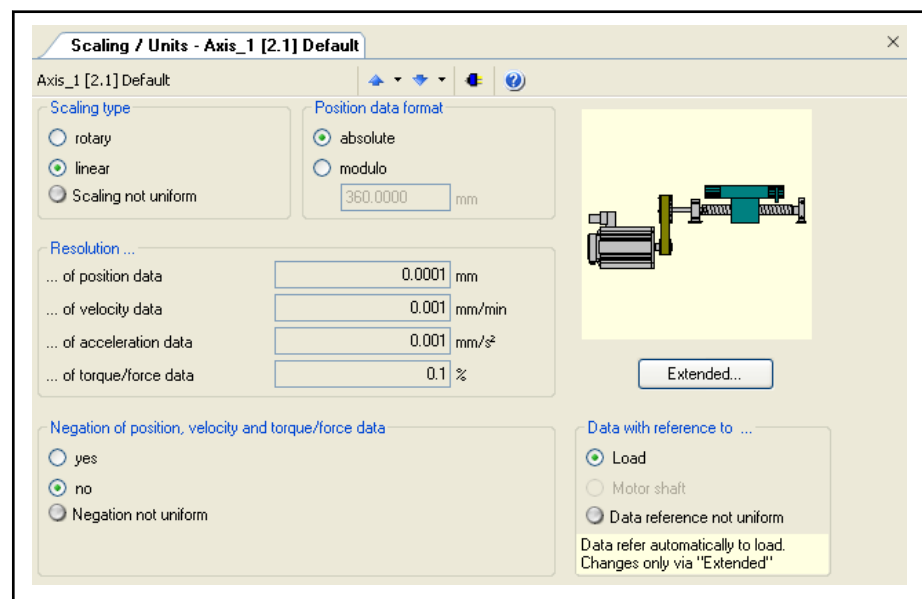


Fig. 5-62: Scaling of the controlled axis of the side register controller

Example configuration

The fastest way to use the side register controller functionality in an application is to combine the touch probe function blocks and the AxisInterface. The touch probe function block provides the actual value to side register. The side register controller calculates the target position for the axis. The side register has no internal functionality to write the position to a position command parameter of the axis. Therefore, the AxisInterface writes the command position to respective parameter of the axis.

This example shows the side register controller [MB_RegisterControllerSide-Type01](#), page 366, with the MB_TouchProbeContinuous, ML_TechMotion-Base.library function block and with the function blocks for the axis interface.

1. Output "PositionDifference" to input the "RecordedPositionWidth"

2. Output "DetectedPositionDifference" to input "Step"

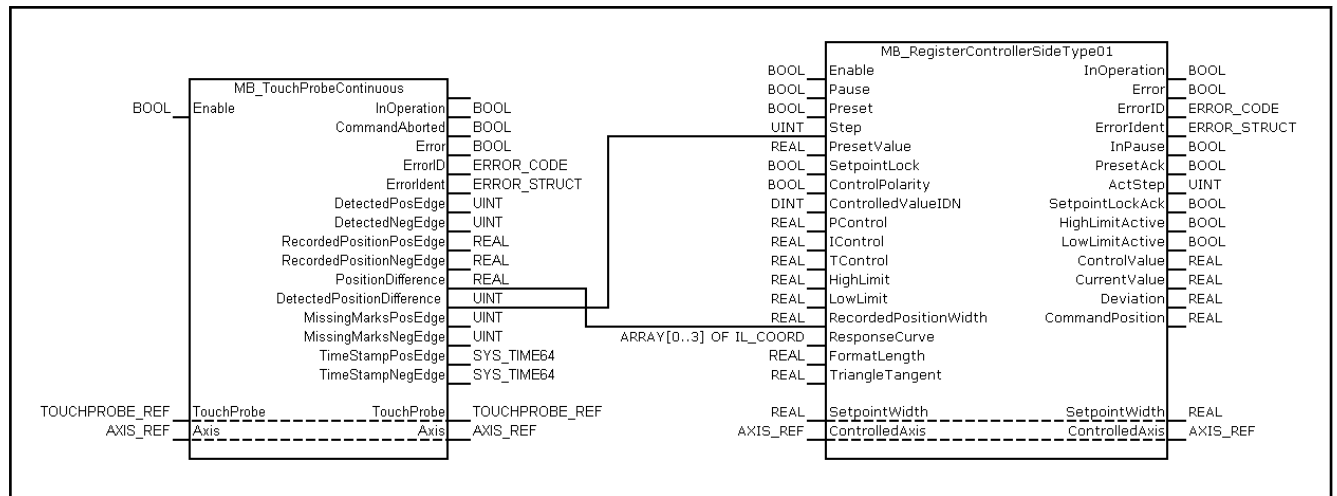


Fig. 5-63: Schematic representation of the function blocks MB_TouchProbeContinuous and MB_RegisterControllerSideType01

3. Assign "CommandPosition" to the array arAxisCtrl[ControlledAxis].Pos-Mode.Position



AxisInterface Settings

- Operating Mode : Absolute positioning
- The following command activates the operating mode
`arAxisCtrl[ ].Admin._OpMode := ModePosAbs;` or `arAxisCtrl[ ].Admin.MODE_POS_ABS := TRUE;`
- EnableCyclicScanning has to be activated

For more information, refer to the manual "XLC/MLC xxVRS Generic Application Template" in the chapter "ML_TechInterface.library".

xx - Version

Program:

```
fbMB_RegisterControllerSideType01(
    Enable:= bEnableRegi,
    Pause:= bPauseRegi,
    Preset:= bPresetRegi,
    Step:= uiDetectedPositionDifference,
    PresetValue:= bPresetValueRegi,
    SetpointLock:= bSetpointLock,
    ControlPolarity:= bControlPolarity,
    ControlledValueIDN:= diControlledValueIDN,
    PControl:= rPControl,
    IControl:= rIControl,
    TControl:= rTControl,
```

ML_TechRegi.library // MX_TechRegi.lib

```

    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);

(* Position command to operate the side register control *)

```

```
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 Introduction and Overview

General Information Technology function blocks (Tech FB) extend the basic functionality of the target system MLC and provide application-specific functionalities. The library described here contains functionalities for crank arm and bell-crank kinematics.

The function blocks described are provided for the target system MLC via the "ML_TechCrank" library.

This documentation describes the functionality as well as the inputs and outputs of the technology function blocks.



The library requires the TEC (technology) function package for the control.

Requirements Technology function blocks of the library "ML_TechCrank" require the firmware support of the target system MLC. Specific requirements of the technology function blocks are documented in the chapter of the respective function blocks.

Overview on function blocks The following table provides an overview on the available function blocks.

Fields of application and function blocks of ML_TechCrank

Field of application (folder in the library)	
Function blocks	Description
Bell-crank kinematics (BellCrank/Basics)	
MB_BellCrankData, page 386	Calculates the y-positions and crank angles Phi at the top and bottom pole
MB_YvirtToYmech, page 388	Converts the virtual translatory position to the mechanical position
Bell-crank kinematics (BellCrank/CamCalculation)	
MB_CamTableBellCrank, page 390	Calculates a cam for transformation for a bell-crank drive
MB_CamTableBellCrankSuperimposed, page 394	Superimposes a user cam with the transformation can for a superimposed cam.
Bell-crank kinematics (BellCrank/Transformers)	
MB_PhiToYvirt, page 398	Converts the crank angle Phi to a mechanical and virtual translatory position
MB_YvirtToPhi, page 400	Converts the virtual translatory position to the crank angle Phi
MB_YvirtToPhiPoly5, page 402	Converts the virtual translatory position to the crank angle Phi using a substitute polynomial 4th order
MB_YVirtToWorkRange, page 404	Returns the working range of crank 1
Crank kinematics (Crank/Crank_Cam)	
MB_CamTableCrank, page 410	Calculates the cam for transforming a translatory virtual master axis position into rotary values (crank angle)
MB_CamTableCrankSuperimposed, page 415	Superimposes a user cam with the transformation can for a superimposed cam.
Crank kinematics (Crank/Transformers)	

MB_MasterToPhi, page 421	Outputs crank angles calculated using the master axis position and the superimposed cam
MB_PhiToXvirt, page 420	Converts crank angle (Phi) into virtual translatory position values (Xvirt) to be used by the PLC program
MB_XvirtToXmech, page 422	Converts virtual translatory position values (Xvirt) to mechanic translatory position values (Xmech)

Tab. 6-1: ML_TechCrank_Library

6.2 Bell crank kinematics

6.2.1 Introduction and overview

Bell-crank kinematics drive forming tools for thermoforming machines or pressing machines for example. The following figure shows the mechanical structure of the bell-crank kinematics used below.

In case of this kinematics, turning crank 1 results in a translatory movement in y direction. The crank is driven by the servo motor. The measuring system is attached to the servo motor and thus works in rotary units.

For the bell-crank kinematics shown, command and actual values (such as position, velocity) are specified respectively displayed in translatory units, although the measuring system works in a rotary manner. The Tech function blocks described below conduct the corresponding transformations.

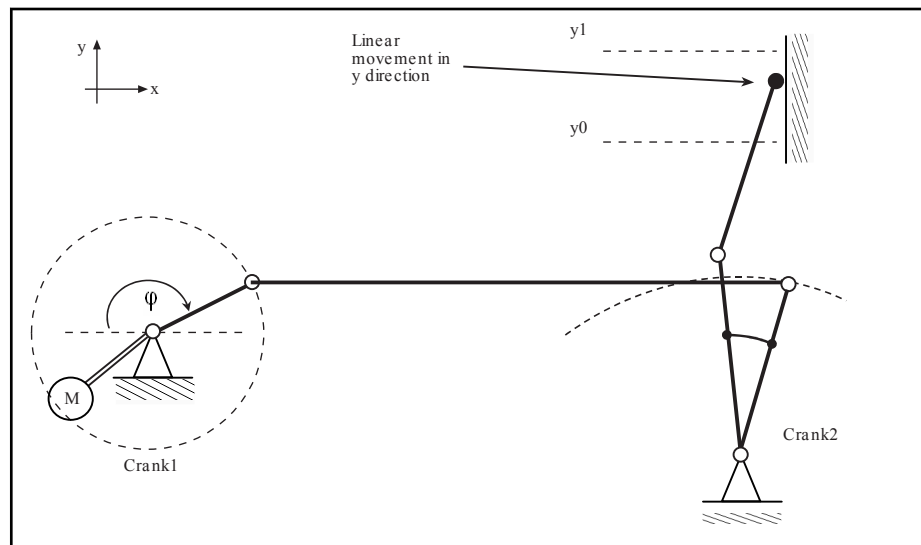


Fig. 6-1: Bell crank kinematics

6.2.2 General Definitions

Basic Variables The following illustration shows the relation between crank and translatory coordinates, as well as the importance of the individual mechanical parameters.

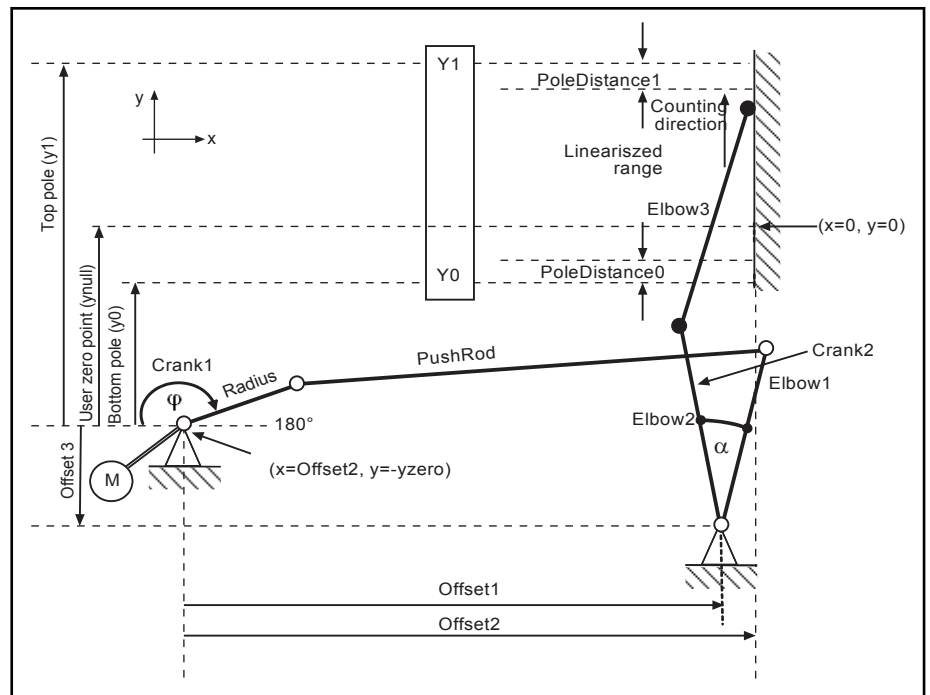


Fig. 6-2: Relation between crank and translatory coordinates at the bell-crank kinematics

Counting Direction

Basically, the following counting directions apply to the arrangement shown (top figure):

- Counting direction of the crank angle:
Clockwise, starting on the left with 0° and ending with 360°
- Counting mode of the translatory y-position:
Increasing numerical values upwards

Mechanical (Y_{mech}) and Virtual (Y_{virt}) Translatory Position

A mechanical translatory position (Y_{mech}) typically can be achieved via two different angular positions of crank 1. Thus, the mechanical translatory position (Y_{mech}) cannot be assigned clearly to one crank angle. In order to solve this ambiguity, the virtual translatory position (Y_{virt}) is introduced, which continues counting in positive direction even if the direction of the translatory axis is reversed.

Using the zero point (Y_{zero}) the user can specify where the mechanical zero point and thus the modulo overflow as well is located in the travel range. In doing so, the modulo value of Y_{virt} has to correspond to the double travel range of the translatory axis. For this, the traversing range (S_v) is defined as distance between top (Y₁) and bottom (Y₀) pole.

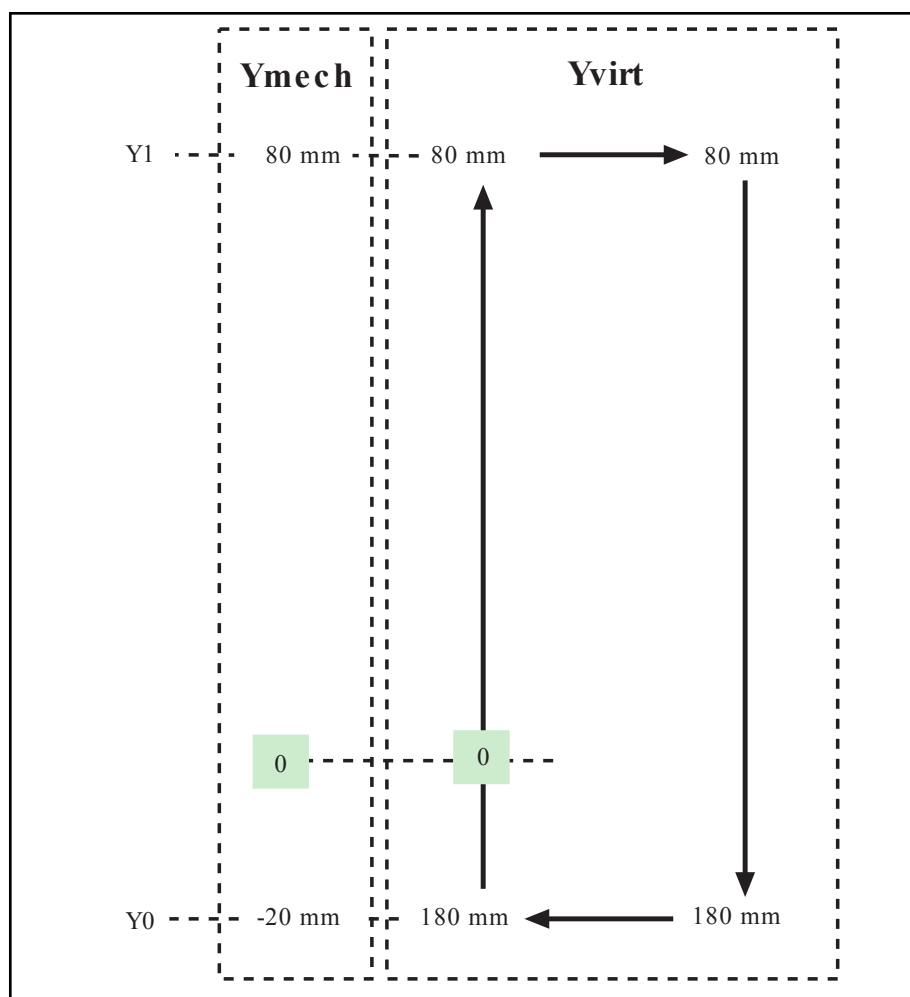


Fig. 6-3: Relation between Y_{virt} and Y_{mech}

Basic Principle of the Cam Transformation

The cams calculated by the function blocks of the bell-crank kinematics transform a master axis position into a crank angle of the bell-crank drive. Basically, the following two use case are distinguished:

- In the first case the real crank axis is coupled to a translatorily scaled virtual axis using a transformation cam.

In doing so, the virtual axis can be operated in all available operation modes in user-friendly translatory units (e.g. mm, mm/s) and the crank follows according to the transformation cam. In doing so, the necessary transformation cam is calculated by the "MB_CamTableBellCrank" function block. For the target system MLD, the virtual axis does not support any synchronous operation modes. That is why this use case is only reasonable to a limited extent.

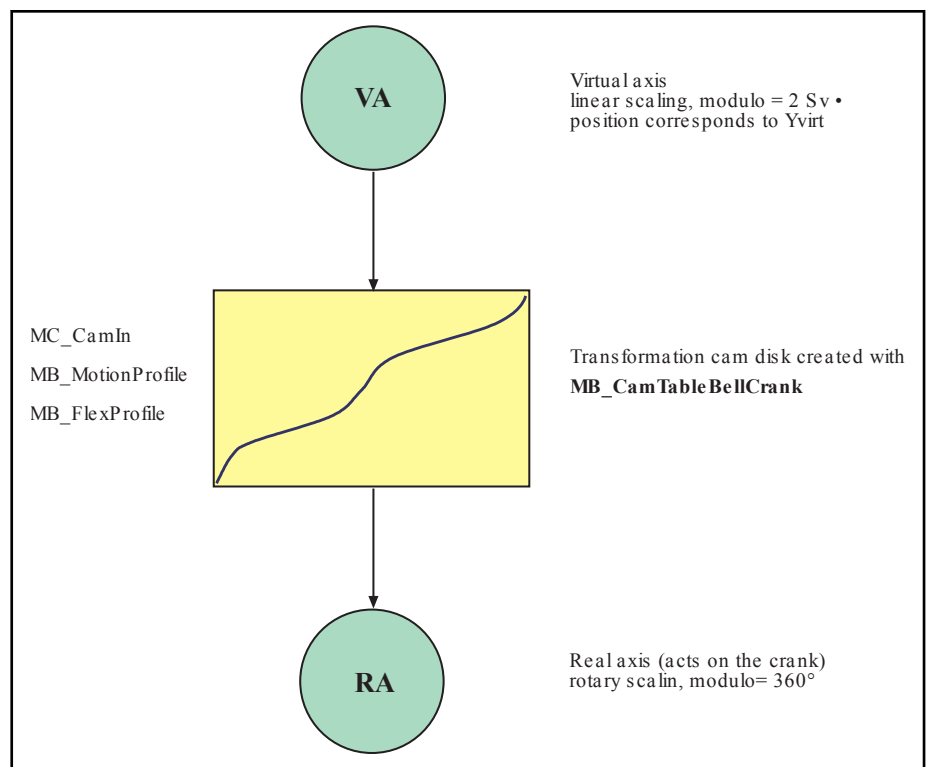


Fig. 6-4: Mode of operation of the transformation cam with a translatable virtual axis

- In the second case, the real crank axis is coupled to any master axis with freely selectable modulo scaling using a superimposed user cam. The user cam describes the movement of the translatable axis in relation to the master axis, without taking the non-linear transmission behavior of the kinematics. Thus, the "MB_CamTableBellCrankSuperimposed" function block superimposes the user cam with the transformation equations of the kinematics and calculates the superimposed cam. With this, the bell-crank drive with a user cam can be coupled to any master axis with freely selectable modulo value. This use case is particularly reasonable for the target system MLD, as the bell-crank drive can be coupled to a master axis. For the target system MLC this use case example has a lower significance. However, an additional virtual axis is not required.

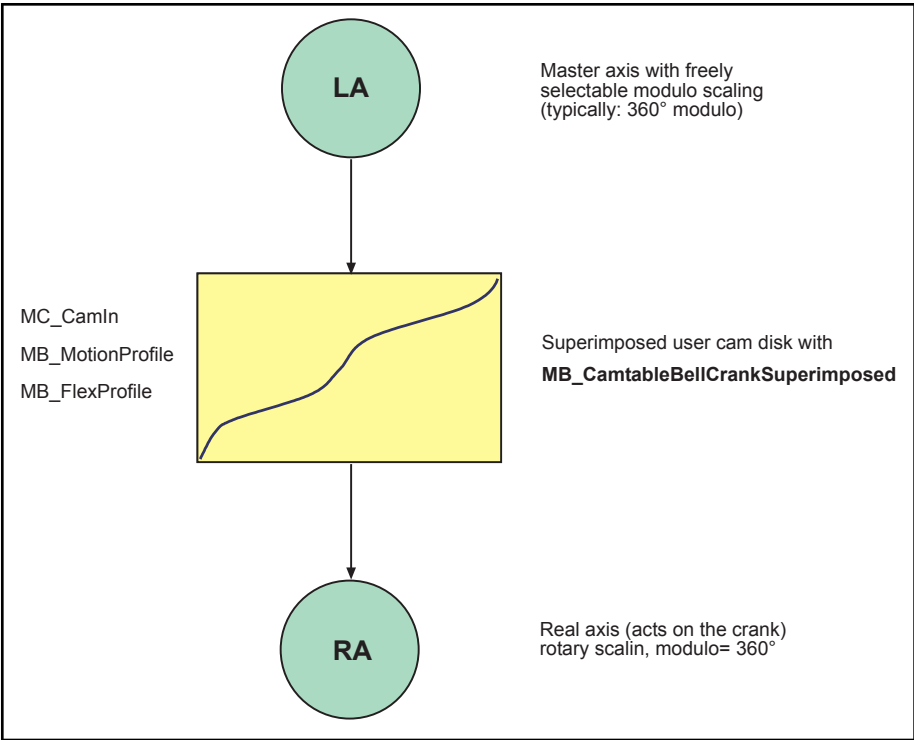


Fig. 6-5: Mode of operation of the superimposed to any master axis

Simplifications and Basic Conditions

- The following simplifications are specified and have to be met. The mechanical kinematic data (such as radius, pushrod) have to be selected accordingly.
- Both cranks reach their dead center simultaneously at the top pole (Y1). Thus, radius and pushrod as well as elbow2 and elbow3 form a straight line at the top pole
 - The second crank moves only in working range 1. In working area 1, the mechanical position of the y-axis moves in positive direction if the second crank turns clockwise. However, the first crank can move in both working areas and turn endlessly in one direction

6.2.3 MB_BellCrankData

Functional description The "MB_BellCrankData" function block calculates the y-positions (Y0, Y1) and the crank angles Phi (Phi0, Phi1) at the bottom and top poles as well as the maximum travel range (Sv) of the y-axis.

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechCrank
MLD with MPx08 and MLD with MPx17 with MA function package	MX_Packaging_07.lib

Tab. 6-2: Reference table of the MB_BellCrankData

Interface description

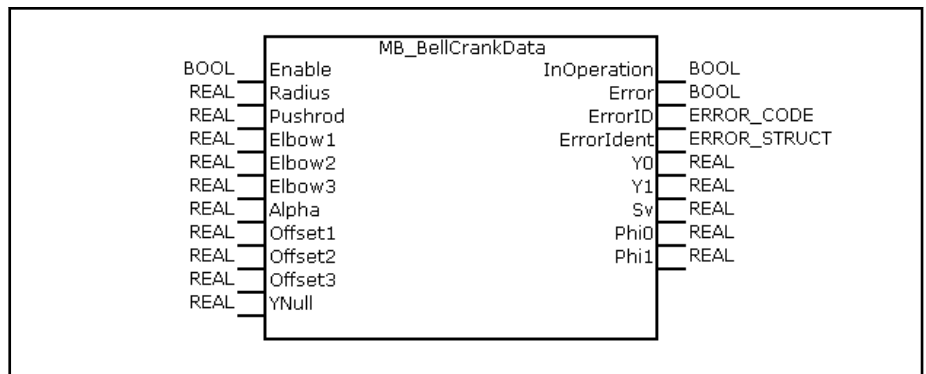


Fig. 6-6: MB_BellCrankData function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Calculation of data in each cycle as long as Enable = TRUE
	Radius	REAL	Length of crank 1 in [mm]
	Pushrod	REAL	Length of the pushrod in [mm]
	Elbow1	REAL	Length of the bell crank 1 [mm]
	Elbow2	REAL	Length of the bell crank 2 [mm]
	Elbow3	REAL	Length of the bell crank 3 [mm]
	Alpha	REAL	Angle between the bell cranks 1 and 2 [°]
	Offset1	REAL	X-axis offset between crank and bell crank pivot point [mm]
	Offset2	REAL	X-axis offset between crank pivot point and translatory motion [mm]
	Offset3	REAL	Y-axis offset between crank and bell crank pivot point [mm]
	Yzero	REAL	Distance crank pivot point to zero position of the translatory motion [mm]
VAR_OUTPUT	InOperation	BOOL	Processing without errors, output data is valid
	Error	BOOL	Error (see ErrorID and ErrorStruct)
	ErrorID	ERROR_CODE	Error Description
	ErrorIdent	ERROR_STRUCT	Detailed error description Error handling, page 387
	Y0	REAL	Mechanical y-position of the bottom pole [mm]
	Y1	REAL	Mechanical y-position of the top pole [mm]
	Sv	REAL	Maximum travel range of the y-axis = ABS(Y1-Y0) [mm]
	Phi0	REAL	Angle Phi of crank 1 at the bottom pole [°]
	Phi1	REAL	Angle Phi of crank 1 at the top pole [°]

Tab. 6-3: Interface variables of the MB_BellCrankData function block

Error handling

The function block generates the following error messages in Additional1/ Additional2 of the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
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	Yzero < (Y0+Yzero)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Yzero > ((Y0+Yzero)+2*travel range)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Internal calculation error, see 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: Error codes of the MB_BellCrankData function block

6.2.4 MB_YvirtToYmech

Functional description The MB_YvirtToYmech function block converts the virtual translatory position (Yvirt) to the mechanical translatory position (Ymech). Ymech can be used for display purposes for example.

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechCrank
MLD with MPx07/08/17 with MA function package	MX_Packaging_07.lib

Tab. 6-5: Reference table of the MB_YvirtToYmech

Interface description

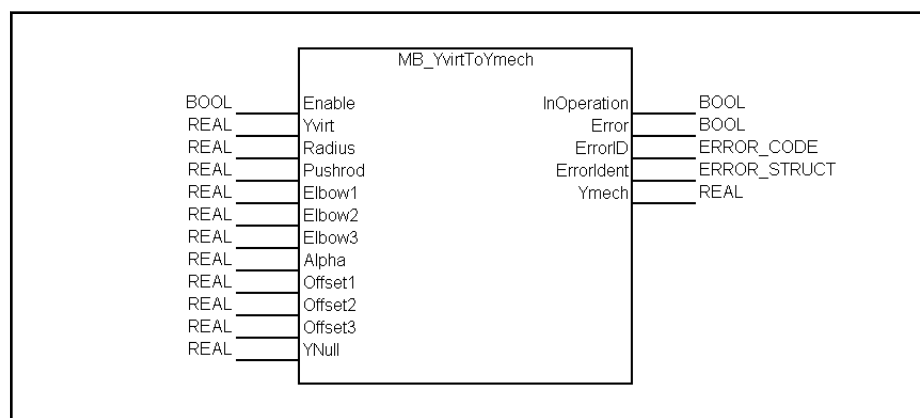


Fig. 6-7: MB_YvirtToYmech function block

I/O type	Name	Type	Comment
VAR_INPUT	Enable	BOOL	The function block calculates as long as Enable=TRUE
	Yvirt	REAL	Virtual translatory position in [mm]
	Radius	REAL	Length of crank in [mm]
	Pushrod	REAL	Length of the pushrod in [mm]
	Elbow1	REAL	Length of the bell crank 1 [mm]
	Elbow2	REAL	Length of the bell crank 2 [mm]
	Elbow3	REAL	Length of the bell crank 3 [mm]
	Alpha	REAL	Angle between the bell cranks 1 and 2 [°]
	Offset1	REAL	X-axis offset between crank and bell crank pivot point [mm]
	Offset2	REAL	X-axis offset between crank pivot point and translatory motion [mm]
	Offset3	REAL	Y-axis offset between crank and bell crank pivot point [mm]
	Yzero	REAL	Distance crank pivot point to zero position of the translatory motion [mm]
VAR_OUTPUT	InOperation	BOOL	Output value (Ymech) valid
	Error	BOOL	Error (see ErrorID and ErrorStruct)
	ErrorID	ERROR_CODE	Error Description
	ErrorIdent	ERROR_STRUCT	Detailed error description Error handling, page 389
	Ymech	REAL	Mechanical translatory position (Ymech) [mm]

Tab. 6-6: Interface variables of the MB_YvirtToYmech

Error handling The function block generates the following error messages in Additional1/ Additional2 of the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
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	Yzero < (Y0+Yzero)

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Yzero > ((Y0+Yzero)+2*travel range)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Internal calculation error, see 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: Error codes of the MB_YvirtToYmech function block

6.2.5 MB_CamTableBellCrank

Functional description

The MB_CamTableBellCrank function block calculates the necessary transformation cam (with 1,024 points) for a bell-crank drive. The transformation cam converts translatory units to the crank angle Phi. Thus, the rotary crank drive can be coupled to a translatory scaled virtual axis using the transformation cam.

The virtual axis can be moved in translatory units and the crank drive follows the transformation cam.

To travel through the poles as well, the distance to the poles (PoleDistance0 und PoleDistance1) can be defined. In the PoleDistance, a compensation function (approximation polynomial of 5th grade) approximating the bell-crank drive and limiting the resulting drive dynamics. During transition to and from the linearized range, position, velocity and acceleration are constant.

The function block provides the calculated cam to the PLC program via the input/output "CamTable".



Refers to the application of the point table using **FlexProfile**:

Depending on the function block input data, the start value of the calculated point table is <> 0%.

When enabling the FlexProfile, the start value = 0 is applied.

Thus, enable the FlexProfile using the input:

SlaveOffset = (360°/100%)* first point of the point table [%]

Thus, the correct start value and the crank angle become active.

The table values are present in "ARRAY OF DINT" format and are calculated for six decimal places. The above mentioned calculation has thus to be divided by 1.000.000.

Example code:

```
fbFlexProfile.SlaveOffset:=
    3.6*DINT_TO_REAL(arCamTable[0])/1000000;
```



The lower the inputs “PoleDistance0” and “PoleDistance1”, the higher the increase of the calculated cam table in the area around the poles.

Thus, the velocity increases at the pole area. This can result in the control error F5113002, Velocity too high for cycletime.

Troubleshoot the error F5113002 as follows

- Reduce Motion cycle time (if possible)
- Reduce master axis velocity (if possible)
- Increase PoleDistance0 and PoleDistance1 inputs

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechCrank
MLD with MPx08 and MLD with MPx17 without MA function package	MX_Packaging_07.lib

Tab. 6-8: Reference table of the MB_CamTableBellCrank

Interface description

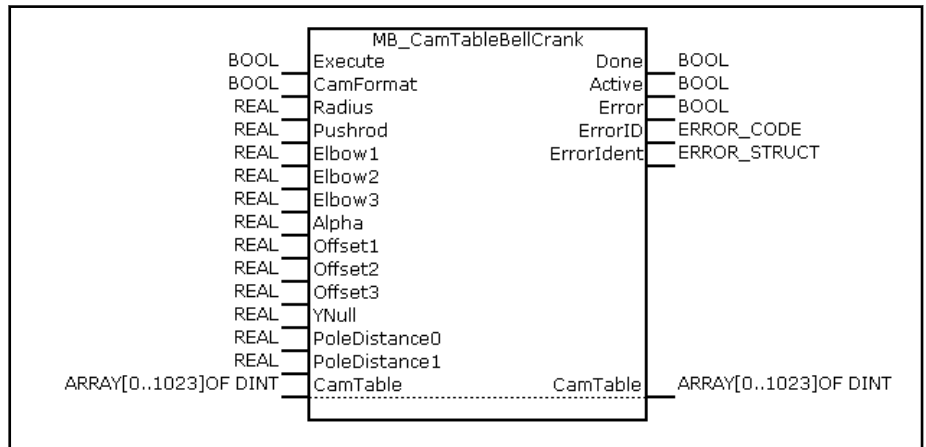


Fig. 6-8: MB_CamTableBellCrank function block

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Positive edge starts calculating the transformation cam
	CamFormat	BOOL	TRUE: New cam format -> last table point corresponds to first table point
	Radius	REAL	Length of crank 1 in [mm]
	Pushrod	REAL	Length of the pushrod in [mm]
	Elbow1	REAL	Length of the bell crank 1 [mm]
	Elbow2	REAL	Length of the bell crank 2 [mm]
	Elbow3	REAL	Length of the bell crank 3 [mm]
	Alpha	REAL	Angle between the bell cranks 1 and 2 [°]
	Offset1	REAL	X-axis offset between crank and bell crank pivot point [mm]

I/O type	Name	Data type	Description
	Offset2	REAL	X-axis offset between crank pivot point and translatory motion [mm]
	Offset3	REAL	Y-axis offset between crank and bell crank pivot point [mm]
	Yzero	REAL	Distance crank pivot point to zero position of the translatory motion [mm]
	PoleDistance0	REAL	Distance from the bottom pole Y0 (in [mm]) to travel through the pole with limited drive dynamics. In this area, a polynomial 5th order is used approximating the bell-crank kinematics and limiting the necessary drive dynamics
	PoleDistance1	REAL	Distance from the top pole Y1 (in [mm]) to travel through the pole with limited drive dynamics. In this area, a polynomial 5th order is used approximating the bell-crank kinematics and limiting the necessary drive dynamics
VAR_OUTPUT	Done	BOOL	Calculation complete -> cam table valid
	Active	BOOL	Function block is processed
	Error	BOOL	Error (see ErrorID and ErrorStruct)
	ErrorID	ERROR_CODE	Error description
	ErrorStruct	ERROR_STRUCT	Detailed error description Troubleshooting, page 393
VAR_IN_OUT	CamTable	ARRAY [0..1023] OF DINT	Array with the calculated transformation cam data

Tab. 6-9: Interface variables of the MB_CamTableBellCrank function block

Necessary boundary conditions and requirements

- Both cranks reach their dead center simultaneously at the top pole (Y1). Thus, radius and pushrod as well as elbow2 and elbow3 form a straight line at the top pole
- The second crank moves only in working range 1. In working area 1, the mechanical position of the y-axis moves in positive direction if the second crank turns clockwise. However, the first crank can move in both working areas and turn endlessly in one direction
- The axis used for the cam (as master) has to be translatory and modulo-scaled.
The modulo value is to be set as follows:
Modulo value = 2 * traversing range (Sv).
For this, the traversing range (Sv) is defined as distance between top (Y1) and bottom (Y0) pole.
The traversing range (Sv) can be determined using the function block [MB_BellCrankData, page 386](#).
- The crank drive has to be scaled rotary, modulo 360°

Use case of MB_CamTableBell-Crank

Using the following sequence, the translatory axis can be traveled in single-axis mode:

- Calculation of the transformation cam using the MB_CamTableBell-Crank function block

- Load calculated transformation cam to the crank drive (e.g. with MB_WriteListParameter if not already loaded)
- Activating power (MC_Power)
- Position the absolute data reference if no absolute value encoder is used. Therefore, the drive command "Drive-controlled referencing" C0600 is used
- Positioning the virtual axis on the translatory position resulting from the crank angle. The resulting translatory position "Yvirt" can be edited in the crank angle (Phi) using the function block [MB_PhiToYvirt](#), [page 398](#). Afterwards, the virtual axis is set to the determined position
- Activating the cam operation mode at the crank drive (e.g. using MC_CamIn, with CamShaftDistance=360,1:1 gear and selection of the transformation cam calculated and loaded before). The crank drive does not perform any synchronizing motion, as the virtual axis already is in "Synchronous position"
- Now, the virtual axis can be traveled in single-axis mode using the function blocks MC_MoveAbsolute and MC_MoveVelocity or AxisInterface (the crank axis follows the transformation cam). The virtual axis can be operated in a synchronous operation mode as well



The crank drive performs a dynamic synchronizing motion when the operation mode is activated, unless the virtual axis was set to the crank position before.

Troubleshooting

The function block generates the following error messages in Additional1/ Additional2 of the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
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 > traversing range
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	Yzero < (Y0+Yzero)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Yzero > ((Y0+Yzero)+2*traversing range)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Internal calculation error, see 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: Error codes of the MB_CamTableBellCrank function block

6.2.6 MB_CamTableBellCrankSuperimposed

Functional description

This function block superimposes the specified user cam (CamInput) with the transformation cam and provides the result via the input/output "CamOutput". The transformation cam is calculated in the function block according to the same rules as in the function block [MB_CamTableBellCrank](#), page 390,.

The principle of superimposition is shown in [Principle of the cam superimposition](#), page 394,. The user cam has to contain 1,024 data points and defines the motion of the translatory axis related to the master axis (without taking the bell-crank kinematics into account).

The table value 100% corresponds to the motion

$2 * \text{traversing range} = \text{modulo value of the virtual translatory axis}$.

The calculated superimposed cam has to be loaded on the drive and used via the cam operation mode to be used by the PLC program.



Depending on the specified end point of the user cam, the following cases are distinguished:

- If the end point of the user cam is near 100%, crank 1 does not reverse the direction (crank 1 "rotates") -> energy-saving procedure, as natural motion of the crank is used.
- If the end point of the user cam is 0%, the translatory axis moves forwards and backwards with reversal of direction of crank 1 (as shown in [Principle of the cam superimposition](#), page 394,.)

Advanced note

From MLC03VRS onwards, the virtual axis used supports synchronous operation modes. Thus, superimposing the user cam with the transformation cam typical for the MLC03 applications is only required if an additional virtual axis should not be used.

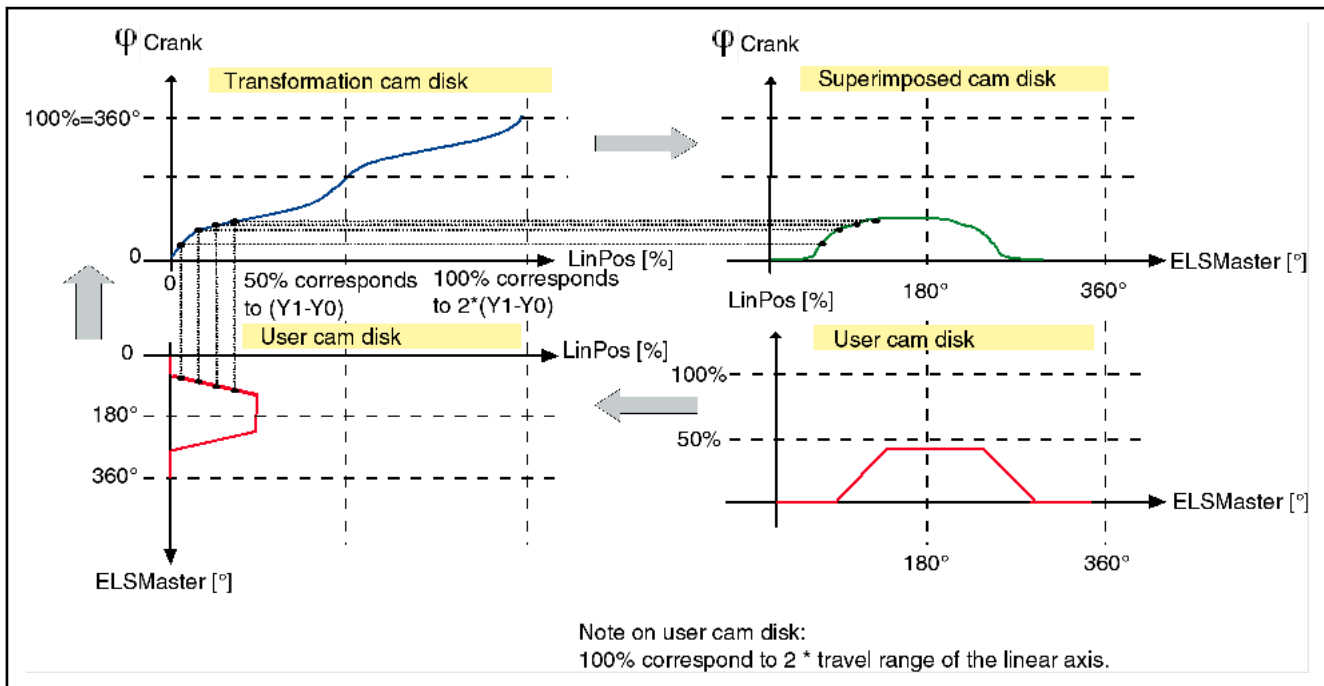


Fig. 6-9: Principle of the cam superimposition



Refers to the application of the point table using **FlexProfile**:

Depending on the function block input data, the start value of the calculated point table is $\neq 0\%$.

When enabling the FlexProfile, the start value = 0 is applied.

Thus, enable the FlexProfile using the input:

SlaveOffset = (360°/100%)* first point of the point table [%]

Thus, the correct start value and the crank angle become active.

The table values are present in "ARRAY OF DINT" format and are calculated for six decimal places. The above mentioned calculation has thus to be divided by 1.000.000.

Example code:

```
fbFlexProfile.SlaveOffset:=
    3.6*DINT_TO_REAL(arCamTable[0])/1000000;
```



The lower the inputs "PoleDistance0" and "PoleDistance1", the higher the increase of the calculated cam table in the area around the poles.

Thus, the velocity increases at the pole area. This can result in the control error F5113002, Velocity too high for cycletime.

Troubleshoot the error F5113002 as follows

- Reduce Motion cycle time (if possible)
- Reduce master axis velocity (if possible)
- Increase PoleDistance0 and PoleDistance1 inputs

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechCrank
MLD with MPx08 and MLD with MPx17 with MA function package	MX_Packaging_07.lib

Tab. 6-11: Reference table of the MB_CamTableBellCrankSuperimposed

Interface description

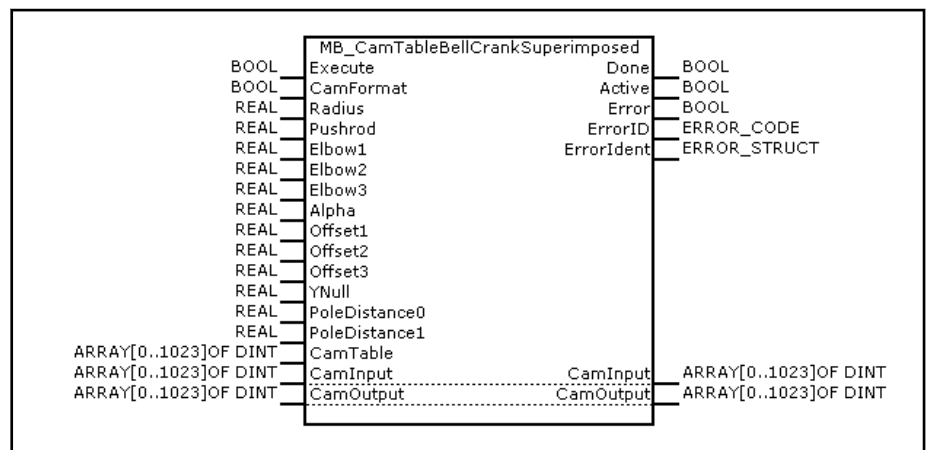


Fig. 6-10: Function block MB_CamTableBellCrankSuperimposed

I/O type	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Positive edge starts calculation of the superimposed cam
	CamFormat	BOOL	TRUE: New cam format -> last table point corresponds to first table point
	Radius	REAL	Length of crank 1 in [mm]
	Pushrod	REAL	Length of the pushrod in [mm]
	Elbow1	REAL	Length of the bell crank 1 [mm]
	Elbow2	REAL	Length of the bell crank 2 [mm]
	Elbow3	REAL	Length of the bell crank 3 [mm]
	Alpha	REAL	Angle between the bell cranks 1 and 2 [°]
	Offset1	REAL	X-axis offset between crank and bell crank pivot point [mm]
	Offset2	REAL	X-axis offset between crank pivot point and translatory motion [mm]
	Offset3	REAL	Y-axis offset between crank and bell crank pivot point [mm]
	Yzero	REAL	Distance crank pivot point to zero position of the translatory motion [mm]
	PoleDistance0	REAL	Distance from the bottom pole Y0 (in [mm]) to travel through the pole with limited drive dynamics. In this area, a polynomial 5th order is used approximating the bell-crank kinematics and limiting the necessary drive dynamics
	PoleDistance1	REAL	Distance from the top pole Y1 (in [mm]) to travel through the pole with limited drive dynamics. In this area, a polynomial 5th order is used approximating the bell-crank kinematics and limiting the necessary drive dynamics
	CamTable	ARRAY [0..1023] OF DINT	
VAR_OUTPUT	Done	BOOL	Calculation complete -> output cam table valid
	Active	BOOL	Function block is processed
	Error	BOOL	Error (see ErrorID and ErrorStruct)
	ErrorID	ERROR_ CODE	Error description
	ErrorIdent	ERROR_ STRUCT	Detailed error description Troubleshooting, page 398
VAR_IN_OUT	CamInput	ARRAY [0..1023] OF DINT	Array with the data of the user cam to specify the translatory motion
	CamOutput	ARRAY [0..1023] OF DINT	Array with data of the calculated superimposed cam to actuate the crank axis

Tab. 6-12: Interface variables of the MB_CamTableBellCrankSuperimposed function block



to use the superimposed cam

If the translatory axis is to be synchronized to the superimposed cam, the procedure is:

- The function block is provided with the user cam via input/output "CamInput" (e.g. by reading a drive cam, a file...)
- The superimposed cam is calculated by the MB_CamTable-BellCrankSuperimposed function block
- Upon superimposition of the cam, the user program loads the superimposed cam "CamOutput" to the drive or the control system (MB_WriteListParameter)
- Activation of Power (MC_Power) at the crank drive
- Position the absolute data of the crank axis if no absolute value encoder is used. Therefore, the drive command C0600 "Drive-controlled referencing" has to be used
- The current crank position and the position command value from the superimposed cam can be different. Thus, the cam operation mode (e.g. via MC_CamIn, with CamShaftDistance= 360, 1:1 gearbox and selection of the superimposed cam) can be activated using the following 2 options:
 - Activation of the cam operation mode at the crank drive without prior position comparison -> The crank drive synchronizes dynamically
 - Crank axis is set to "Synchronous position" before activating the cam operation mode. Therefore, the crank angle from the superimposed cam can be determined using the function "MB_MasterToPhi". Before activating the synchronous operation mode, the crank drive can be set to this position. Afterwards, the superimposed cam can be activated without synchronizing motion



To scale the axes

- The master axis used has to be modulo-scaled (translatory or rotary).
- The axis to the crank drive has to be scaled, rotary, modulo 360°

Necessary boundary conditions and requirements

- Both cranks reach their dead center simultaneously at the top pole (Y1). Thus, radius and pushrod as well as elbow2 and elbow3 form a straight line at the top pole
- The second crank moves only in working range 1. In working area 1, the mechanical position of the y-axis moves in positive direction if the second crank turns clockwise. However, the first crank can move in both working areas and turn endlessly in one direction
- The axis used for the cam (as master) has to be translatory and modulo-scaled.

The modulo value is to be set as follows:

Modulo value = 2 * traversing range (Sv).

For this, the traversing range (Sv) is defined as distance between top (Y1) and bottom (Y0) pole.

The traversing range (Sv) can be determined using the function block [MB_BellCrankData](#), page 386,.

- The crank drive has to be scaled rotary, modulo 360°

Troubleshooting The function block generates the following error messages in Additional1/ Additional2 of the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
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 > traversing range
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	Yzero < (Y0+Yzero)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Yzero > ((Y0+Yzero)+2*traversing range)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Internal calculation error, see 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: Error codes of the MB_CamTableBellCrankSuperimposed function block

6.2.7 MB_PhiToYvirt

Functional description

The function block MB_PhiToYvirt converts the crank angle (Phi) to the mechanical and virtual translatory position (Ymech, Yvirt).

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechCrank
MLD with MPx07/08/17 with MA function package	MX_Packaging_07.lib

Tab. 6-14: Reference table of the MB_PhiToYvirt

Interface description

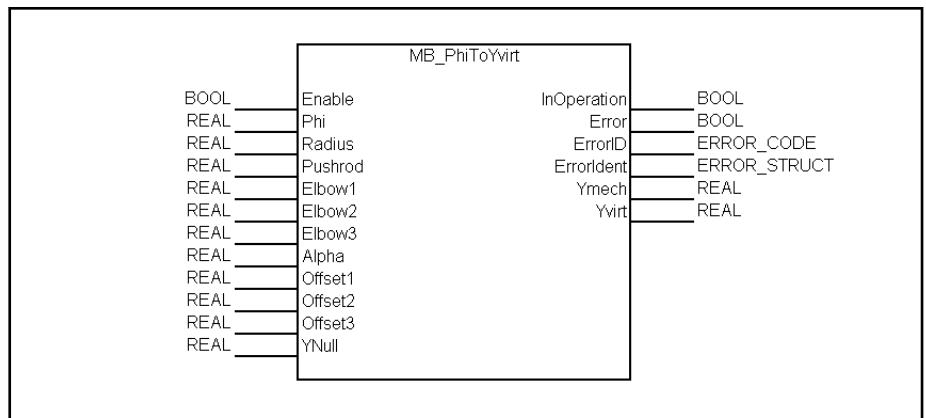


Fig. 6-11: Function block MB_PhiToYvirt

I/O type	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Calculation of Yvirt in every cycle as long as Enable=TRUE
	Phi	REAL	Crank position in [°]
	Radius	REAL	Length of crank in [mm]
	Pushrod	REAL	Length of the pushrod in [mm]
	Elbow1	REAL	Length of the bell crank 1 [mm]
	Elbow2	REAL	Length of the bell crank 2 [mm]
	Elbow3	REAL	Length of the bell crank 3 [mm]
	Alpha	REAL	Angle between the bell cranks 1 and 2 [°]
	Offset1	REAL	X-axis offset between crank and bell crank pivot point [mm]
	Offset2	REAL	X-axis offset between crank pivot point and translatory motion [mm]
	Offset3	REAL	Y-axis offset between crank and bell crank pivot point [mm]
	Yzero	REAL	Distance crank pivot point to zero position of the translatory motion [mm]
VAR_OUTPUT	InOperation	BOOL	Calculation complete -> Yvirt valid
	Error	BOOL	Error (see ErrorID and ErrorStruct)
	ErrorID	ERROR_CODE	Error Description
	ErrorIdent	ERROR_STRUCT	Detailed error description Error handling, page 399
	Ymech	REAL	Mechanical translatory position [mm]
	Yvirt	REAL	Virtual translatory position [mm]

Tab. 6-15: Interface variables of the MB_PhiToYvirt function block

Error handling

The function block generates the following error messages in Additional1/ Additional2 of the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
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	Yzero < (Y0+Yzero)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Yzero > ((Y0+Yzero)+2*travel range)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Internal calculation error, see 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: Error codes of the MB_PhiToYvirt function block

6.2.8 MB_YvirtToPhi

Functional description

The MB_YvirtToPhi function block converts the virtual translatory position (Yvirt) in the crank angle (Phi).

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechCrank
MLD with MPx08 and MLD with MPx17 with MA function package	MX_Packaging_07.lib

Tab. 6-17: Reference table of the MB_YvirtToPhi

Interface description

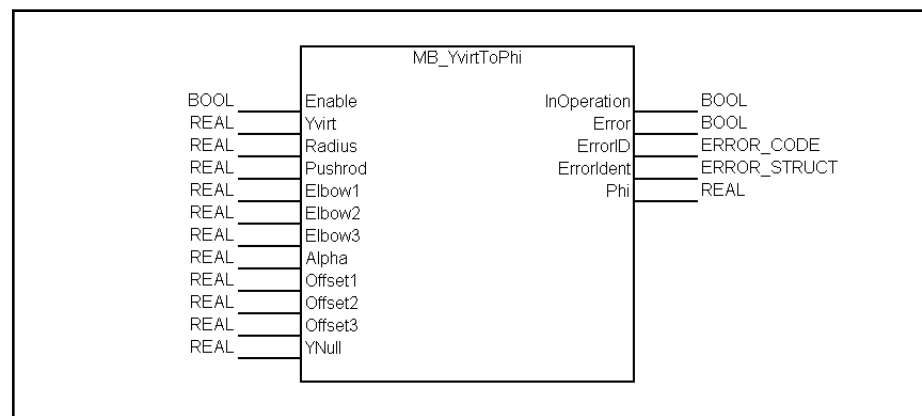


Fig. 6-12: MB_YvirtToPhi function block

I/O type	Name	Type	Description
VAR_INPUT	Enable	BOOL	Calculation of Phi in every cycle as long as Enable=TRUE
	Yvirt	REAL	Virtual translatory position [mm]
	Radius	REAL	Length of crank in [mm]
	Pushrod	REAL	Length of the pushrod in [mm]
	Elbow1	REAL	Length of the bell crank 1 [mm]
	Elbow2	REAL	Length of the bell crank 2 [mm]
	Elbow3	REAL	Length of the bell crank 3 [mm]
	Alpha	REAL	Angle between the bell cranks 1 and 2 [°]
	Offset1	REAL	X-axis offset between crank and bell crank pivot point [mm]
	Offset2	REAL	X-axis offset between crank pivot point and translatory motion [mm]
	Offset3	REAL	Y-axis offset between crank and bell crank pivot point [mm]
	Yzero	REAL	Distance crank pivot point to zero position of the translatory motion [mm]
VAR_OUTPUT	InOperation	BOOL	Calculation complete -> Phi valid
	Error	BOOL	Error (see ErrorID and ErrorStruct)
	ErrorID	ERROR_CODE	Error Description
	ErrorIdent	ERROR_STRUCT	Detailed error description Error handling, page 401
	Phi	REAL	Crank angle in [°]

Tab. 6-18: Interface variables of the MB_YvirtToPhi function block

Error handling The function block generates the following error messages in Additional1/ Additional2 of the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
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	Yzero < (Y0+Yzero)

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Yzero > ((Y0+Yzero)+2*travel range)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Internal calculation error, see 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: Error codes of the MB_YvirtToPhi function block

6.2.9 MB_YvirtToPhiPoly5

Functional description The MB_YvirtToPhiPoly5 function block converts the virtual translatory position (Yvirt) to the crank angle (Phi). In the PoleDistance, a compensation function (approximation polynomial of 5th grade) approximating the bell-crank drive and limiting the resulting drive dynamics. During transition in and from the linearized range, position, velocity and acceleration at crank 1 are constant.

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechCrank
MLD with MPx08 and MLD with MPx17 with MA function package	MX_Packaging_07.lib

Tab. 6-20: Reference table of the MB_YvirtToPhiPoly5

Interface description

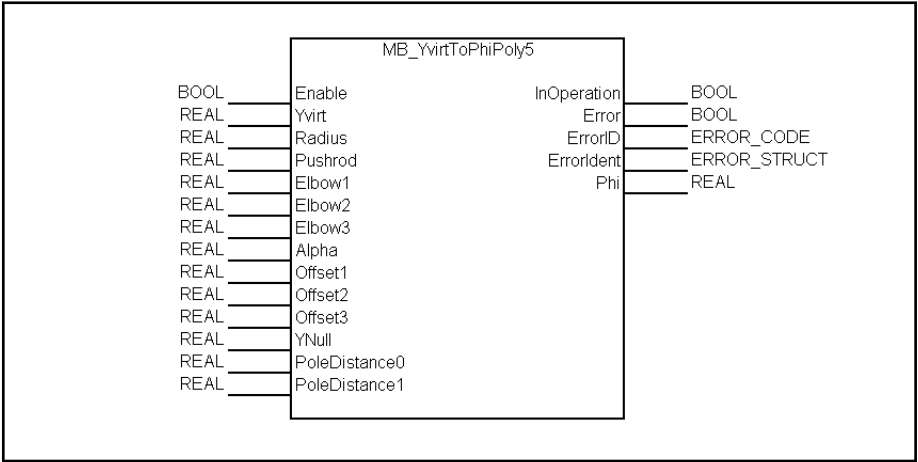


Fig. 6-13: MB_YvirtToPhiPoly5 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Calculation of Phi in every cycle as long as Enable=TRUE
	Yvirt	REAL	Virtual translatory position [mm]
	Radius	REAL	Length of crank in [mm]
	Pushrod	REAL	Length of the pushrod in [mm]
	Elbow1	REAL	Length of the bell crank 1 [mm]
	Elbow2	REAL	Length of the bell crank 2 [mm]
	Elbow3	REAL	Length of the bell crank 3 [mm]
	Alpha	REAL	Angle between the bell cranks 1 and 2 [°]
	Offset1	REAL	X-axis offset between crank and bell crank pivot point [mm]
	Offset2	REAL	X-axis offset between crank pivot point and translatory motion [mm]
	Offset3	REAL	Y-axis offset between crank and bell crank pivot point [mm]
	Yzero	REAL	Distance crank pivot point to zero position of the translatory motion [mm]
	PoleDistance0	REAL	Distance from the bottom pole Y0 (in [mm]), in order to pass the pole with limited drive dynamics. In this area, a polynomial 5th order is used approximating the bell-crank kinematics and limiting the necessary drive dynamics
	PoleDistance1	REAL	Distance from the top pole Y1 (in [mm]) to travel through the pole with limited drive dynamics. In this area, a polynomial 5th order is used approximating the bell-crank kinematics and limiting the necessary drive dynamics
VAR_OUTPUT	InOperation	BOOL	Calculation complete -> Phi valid
	Error	BOOL	Error (see ErrorID and ErrorStruct)
	ErrorID	ERROR_CODE	Error Description
	ErrorIdent	ERROR_STRUCT	Detailed error description Error handling, page 403
	Phi	REAL	Crank angle in [°]

Tab. 6-21: Interface variables of the MB_YvirtToPhiPoly5 function block

Error handling The function block generates the following error messages in Additional1/ Additional2 of the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
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

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0007	PoleDistance1 <= 0
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0008	PoleDistance0+PoleDistance1 > travel range
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	Yzero < (Y0+Yzero)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#000E	Yzero > ((Y0+Yzero)+2*travel range)
CALCULATION_ERROR (16#0007)	16x1002	16#xxyy	Internal calculation error, see 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: Error codes of the MB_YvirtToPhiPoly5 function block

6.2.10 MB_YVirtToWorkRange

Functional description

The function MB_YVirtToWorkRange provides the working range (1 or 2) of crank 1 as return value resulting from the virtual translatory position. In working range 1 the mechanical position of the y-axis moves in a positive direction if crank 1 rotates clockwise. In working range 2 the mechanical position of the y-axis moves in a negative direction if crank 1 rotates clockwise. Internally, this function is used by other function blocks.

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechCrank
MLD with MPx08 and MLD with MPx17 with MA function package	MX_Packaging_07.lib

Tab. 6-23: Reference table of the MB_YVirtToWorkRange

Interface description

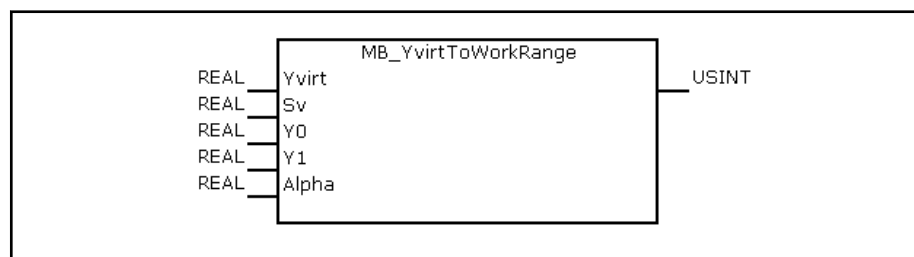


Fig. 6-14: MB_YVirtToWorkRange function

I/O type	Name	Data type	Description
VAR_INPUT	Yvirt	REAL	Virtual translatory position [mm]
	Sv	REAL	Maximum travel range of the y-axis = $ABS(Y1-Y0)$ [mm]
	Y0	REAL	Mechanical y-position of the bottom pole [mm]
	Y1	REAL	Mechanical y-position of the top pole [mm]
	Alpha	REAL	Angle between 1st and 2nd Elbow[°]
RETURN		USINT	Provides the working range (1 or 2) of crank 1

Tab. 6-24: Interface variables of the MB_YVirtToWorkRange function

Error handling The function does not generate any errors.

6.3 Crank kinematics

6.3.1 Introduction and Overview

The functions and function blocks described in this section convert translatory into rotary motions (crank angle) and vice versa, and are designed for crank kinematics applications.

Crank kinematics often are used to drive cross sealing shoes in bagging machines or forming and punching tools in thermoforming machines. The translatory carriage in this kinematics is moved by the rotary motion of a crank. The axis of the servo motor driving the crank is located opposite and displaced of the axis of the translatory slide.

The crank kinematics shown below outputs command and actual values (position, velocity etc.) in translatory units, even though the measurement system outputs in rotary units. Therefore, command and actual values have to be converted from translatory to rotary units and vice versa.

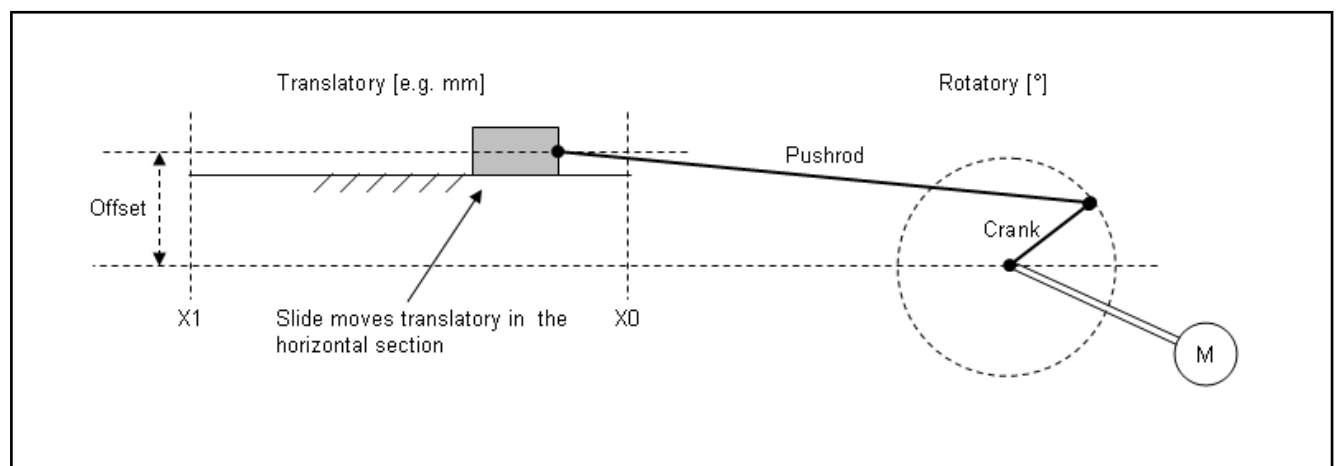


Fig. 6-15: Single-axis crank kinematics with offset crank axis

The following functions and function blocks are supported:

Function	Description
MB_MasterToPhi , page 421	Outputs crank angles calculated using the master axis position and the superimposed cam

Tab. 6-25: Crank kinematics function

Function block	Description
MB_CamTableCrank, page 410	Calculates the cam for transforming a translatory virtual master axis position into rotary values (crank angle)
MB_CamTableCrankSuperimposed, page 415	Superimposes a user cam with the transformation cam for a superimposed cam.
MB_PhiToXvirt, page 420	Converts crank angle (Phi) into virtual translatory position values (Xvirt) to be used by the PLC program
MB_XvirtToXmech, page 422	Converts virtual translatory position values (Xvirt) to mechanic translatory position values (Xmech)

Tab. 6-26: Crank kinematics function blocks

Typical use cases are:

Operation Mode: Single Axis

The term single-axis mode relates to the position or velocity-controlled operation of the translatory axis. In the present application, a virtual master axis (with translatory scaling) is moved by position or velocity control function blocks (e.g. MC_MoveAbsolute, MC_MoveRelative or MC_MoveVelocity).

The virtual master axis position is converted from translatory position values into crank angle values by the transformation cam. The transformation cam is calculated by function block [MB_CamTableCrank, page 410](#), and activated with the MC_CamIn function block. The crank axis follows the virtual master axis (issuing translatory units), at which the transformation cam converts the translatory position values into crank angle values. For detailed information, see .

Synchronous Operation with Cam

In this application, a translatory axis with a user-defined cam profile follows a master axis. The user-defined cam exclusively relates to the translatory axis. Thus, the transfer behavior of the crank does not have to be taken into account in the user-defined cam.

In contrast to that, the user-defined cam is superimposed with a transformation cam by function block [MB_CamTableCrankSuperimposed, page 415](#), and the resulting cam is transferred to the drive by the PLC program.

Reversed Transformation

In both the use cases above, translatory units are converted into rotary units (command value generation). Rotary units have to be converted into translatory units to ensure that the PLC can subsequently process position and velocity commands. The function block [MB_PhiToXvirt, page 420](#), is used for this purpose.

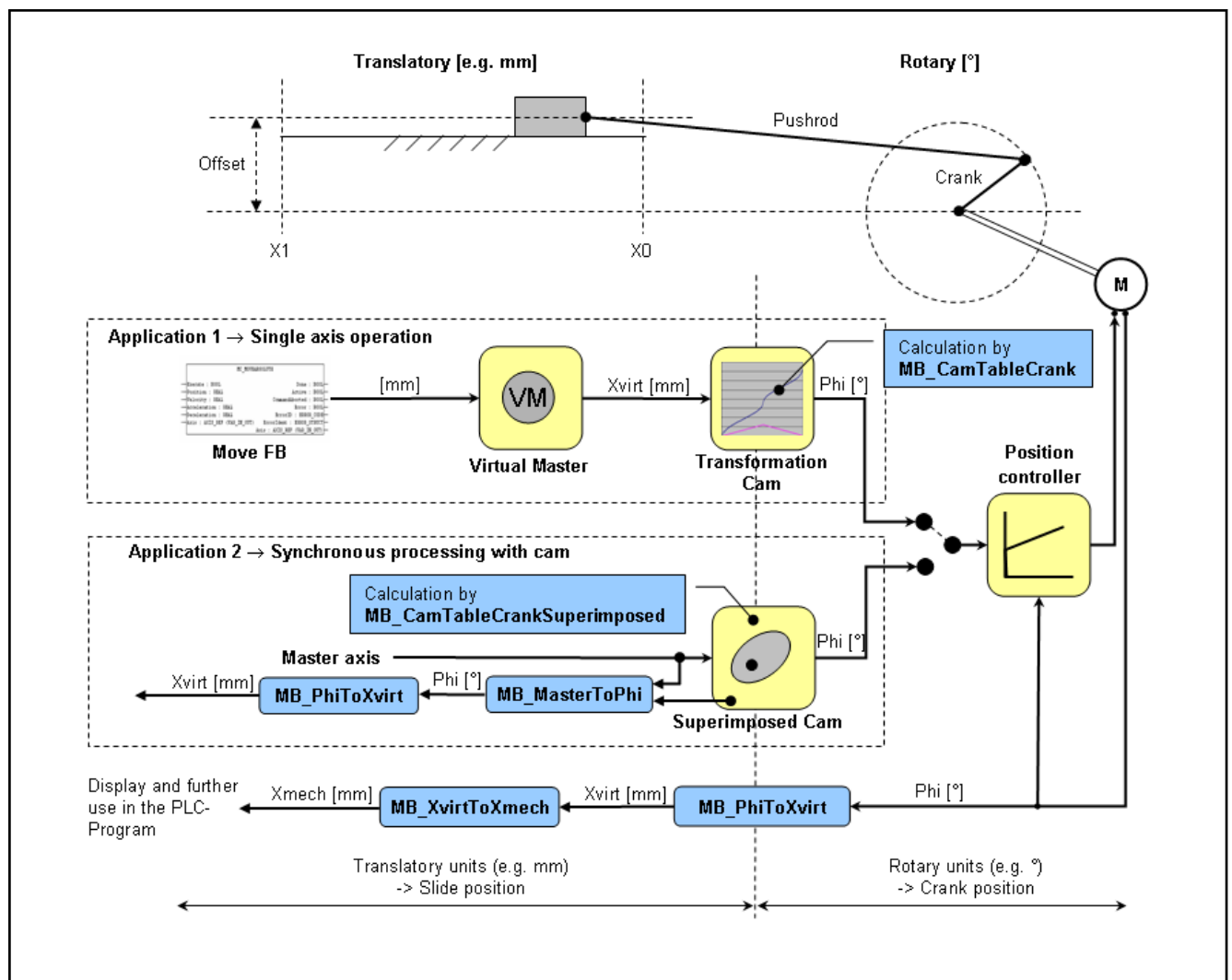


Fig. 6-16: Schematic representation of the crank kinematics technology function blocks

6.3.2 General Definitions

Definition of the Basic Variables in the Crank Kinematics

The following figures show the relations between crank and translatory coordinates as well as the importance of different mechanical parameters.

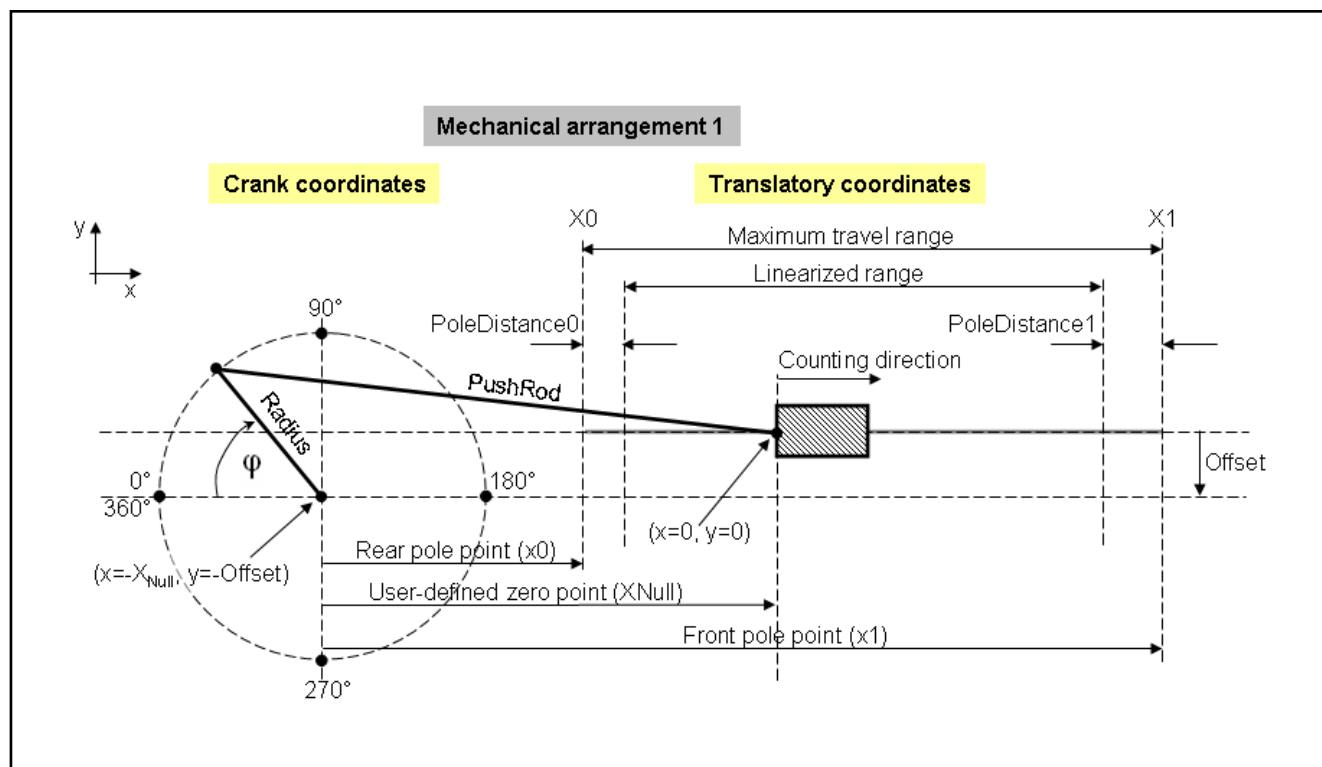


Fig. 6-17: Mechanical arrangement 1: Relation of crank and translatory coordinates in the crank kinematics

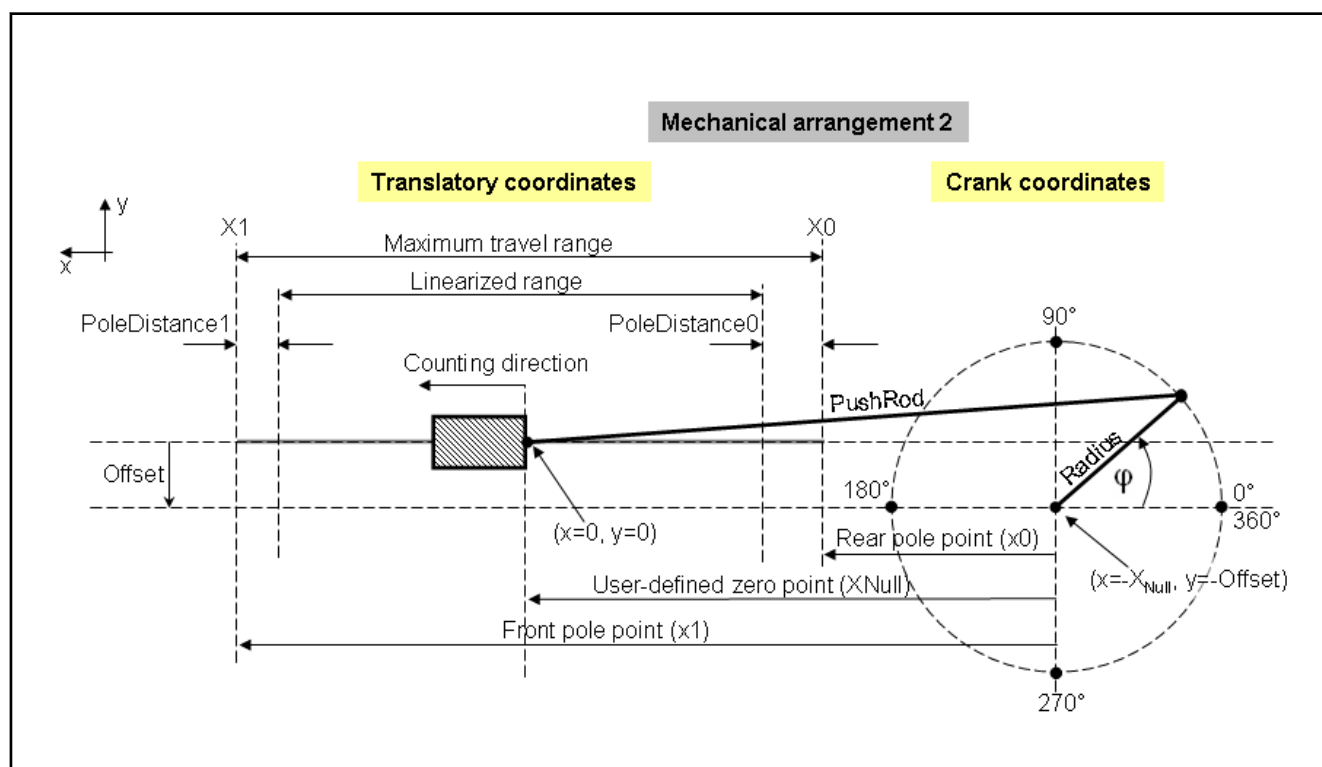


Fig. 6-18: Mechanical arrangement 2: Relation of crank and translatory coordinates in the crank kinematics

Variable/term	Definition
Xzero	Distance between crank axis and slide zero point [mm]
PoleDistance0 PoleDistance1	Distance from the pole X0 / X1 (a compensation function is used outside the linearized range)
Maximum traversing range	Travel range of the translatory coordinate system = X1-X0 .
Linearized range	In the linearized range, no approximation polynomial is effective
Transformation cam	Converts translatory (mm) coordinates to crank angles (degrees).

Tab. 6-27: Variables and terms

Counting Direction

The following counting directions apply in general for the mechanical arrangements shown in [fig. 6-17 "Mechanical arrangement 1: Relation of crank and translatory coordinates in the crank kinematics"](#) on page 408 and [fig. 6-18 "Mechanical arrangement 2: Relation of crank and translatory coordinates in the crank kinematics"](#) on page 408:

- Counting direction of the crank angle:
 - Mechanical arrangement 1: Clockwise rotation, starting on the left with 0° and ending with 360°
 - Mechanical arrangement 2: Counterclockwise, starting on the right with 0° and ending with 360°
- Counting direction of the translatory slide:
 - Mechanical arrangement 1: Increasing numerical values to the right
 - Mechanical arrangement 2: Increasing numerical values to the left

Mechanical (Xmech) and Virtual (Xvirt) Translatory Position

Mechanically, the slide (translatory axis) can move between the rear (X0) and the front (X1) pole. Between these two limits, the "maximum travel range" is situated. Every mechanical translatory position (Xmech) corresponds to two different crank angles. Thus, the virtual translatory position (Xvirt) assigns a unique translatory position to every crank angle. The virtual translatory position thus moves in positive direction, even though the mechanical position (Xmech) reverses the direction by exceeding the pole. The modulo overflow of Xvirt is defined by the user via the zero point (Xzero). Furthermore, the modulo value of Xvirt corresponds to the double travel range of the crank kinematics. The travel range is determined as follows:

$$\text{Travelrange} = (\text{FrontPolePoint} - \text{RearPolePoint})$$

$$\text{Travelrange} = \sqrt{(\text{Radius} + \text{PushRod})^2 - \text{Offset}^2} - \sqrt{(\text{Radius} - \text{PushRod})^2 - \text{Offset}^2}$$

Fig. 6-19: Travel range equation for the crank kinematics

The following figure shows Xvirt and Xmech in an example:

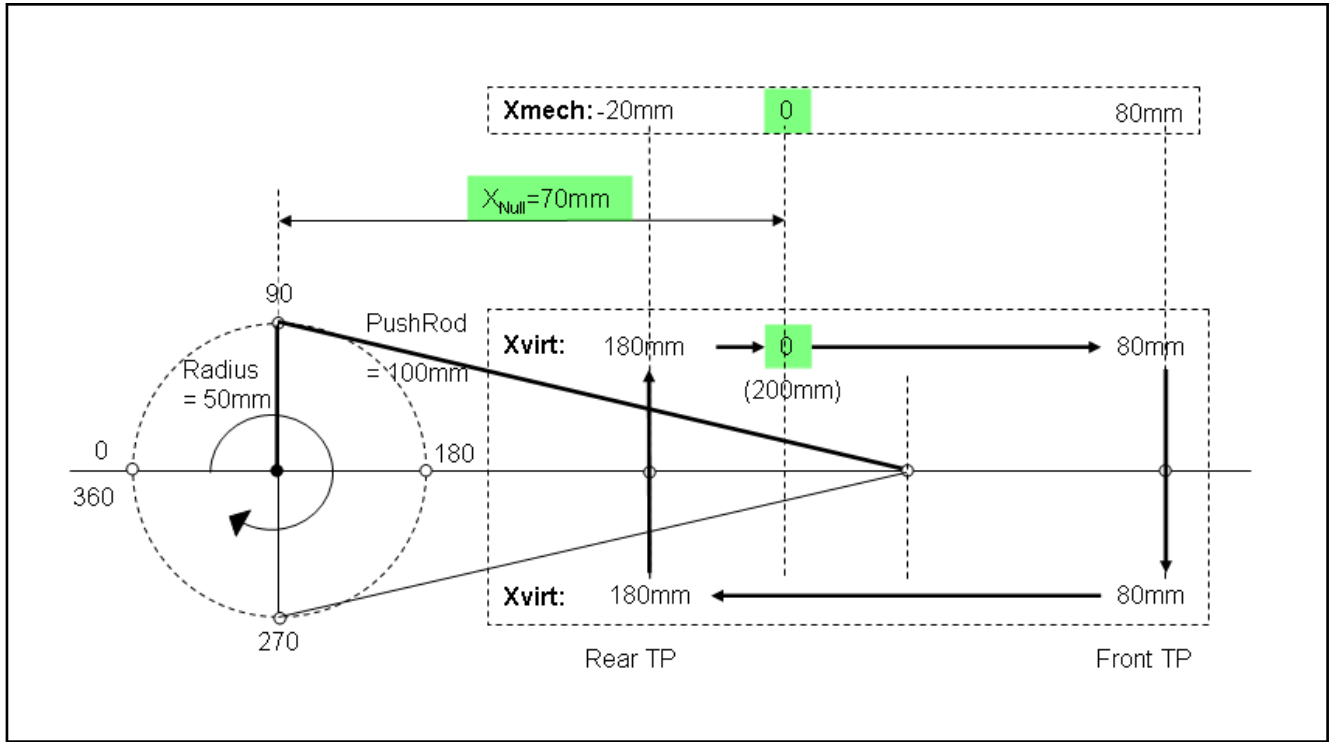


Fig. 6-20: Counting direction of the slide position (Xvirt and Xmech)

6.3.3 MB_CamTableCrank

Brief description The function block MB_CamTableCrank calculates a transformation cam (with 1,024 data points) based on crank-specific input values. This transformation cam converts translatory position values into rotary position values (crank angle) to be used in a crank kinematics. It allows to couple a translatory virtual master axis with a rotary crank drive. For detailed information, see [Introduction and Overview, page 405](#).

The virtual axis in translatory units moves in the single-axis mode while the drive of the transformation cam follows.

A motion extending the linear area can be executed by specifying a value for "PoleDistance0" and "PoleDistance1". In the "PoleDistance", a compensation function (approximation polynomial 5th grade) approximating the crank drive and limiting the resulting drive dynamics. When going in and from the linearized position range, velocity and acceleration are constant.

The function block provides the calculated transformation cam using VAR_IN_OUT "CamTable". The PLC program has to load the generated cam table to the drive (e.g. via MB_WriteListParameter) to be used (e.g. by the function block MC_CamIn).



Refers to the application of the point table using **FlexProfile**:

Depending on the function block input data, the start value of the calculated point table is $\neq 0\%$.

When enabling the FlexProfile, the start value = 0 is applied.

Thus, enable the FlexProfile using the input:

SlaveOffset = (360°/100%)* first point of the point table [%]

Thus, the correct start value and the crank angle become active.

The table values are present in "ARRAY OF DINT" format and are calculated for six decimal places. The above mentioned calculation has thus to be divided by 1.000.000.

Example code:

```
fbFlexProfile.SlaveOffset:=
    3.6*DINT_TO_REAL(arCamTable[0])/1000000;
```



The lower the inputs "PoleDistance0" and "PoleDistance1", the higher the increase of the calculated cam table in the area around the poles.

Thus, the velocity increases at the pole area. This can result in the control error F5113002, Velocity too high for cycletime.

Troubleshoot the error F5113002 as follows

- Reduce Motion cycle time (if possible)
- Reduce master axis velocity (if possible)
- Increase PoleDistance0 and PoleDistance1 inputs

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechCrank
MLD with MPx07/08/17 with MA function package	MX_Packaging_07.lib

Tab. 6-28: Reference table of the MB_CamTableCrank

Interface description

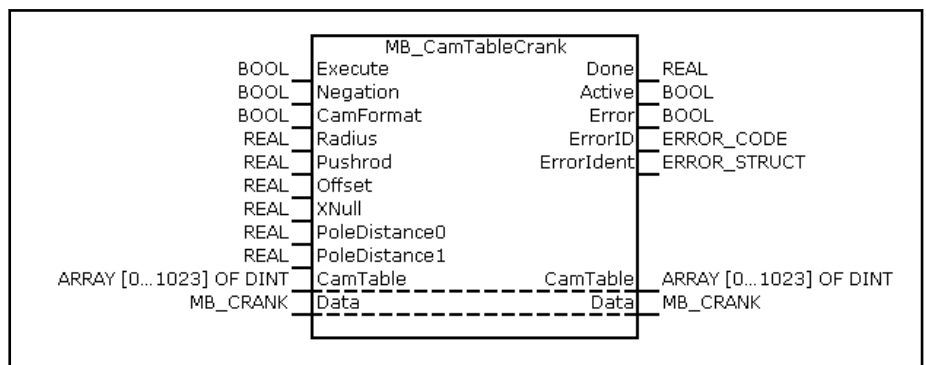


Fig. 6-21: MB_CamTableCrank function block

I/O type	Name	Data type	Explanation
VAR_INPUT	Execute	BOOL	Calculation of the transformation cam is triggered by a positive edge.
	Negation	BOOL	Inverts the direction of counting of a translatory axis
	CamFormat	BOOL	TRUE = new format (last point = first point = 360°) FALSE = old format (last point = 360° - d)
	Radius	REAL	Crank length in [mm]
	Pushrod	REAL	Length of the pushrod in [mm]
	Offset	REAL	Offset of the slide level regarding the crank center [mm]
	Xzero	REAL	Distance between crank center and slide zero point [mm]
	PoleDistance0	REAL	Distance between the rear pole X0 [mm] and the linearized range, in which the movement is influenced by the limited drive dynamics
	PoleDistance1	REAL	Distance between the front pole X1 [mm] and the linearized range, in which the movement is influenced by the limited drive dynamics
VAR_OUTPUT	Done	BOOL	Calculation finished, cam table data and output data is valid
	Active	BOOL	Function block is being executed.
	Error	BOOL	Error (see ErrorID and ErrorStruct)
	ErrorID	ERROR_ CODE	Error description
	ErrorStruct	ERROR_ STRUCT	Detailed error description Troubleshooting, page 413
VAR_IN_OUT	CamTable	ARRAY[0..1023] OF DINT	Array with the data of the calculated transformation cam.
	Data	MB_CRANK	Structure with internally calculated data of the crank kinematics. The data is calculated in this function block and transferred to other function blocks MB_CRANK, page 414

Tab. 6-29: Interface variables of the MB_CamTableCrank function block

Boundary condition The following boundary conditions have to be met to prevent PLC error messages:

Boundary condition	Reason
$Rod \geq Radius + Offset $	If this boundary condition is not met, the translatory slide cannot travel mechanically on a constant y-coordinate
$(Xzero \geq \text{rear TP}) \ \& \ (Xzero \leq \text{rear TP} + 2 \times \text{traversing range})$	The zero point has to be in the traversing range

Tab. 6-30: Boundary conditions

To traverse the translatory slide in single-axis mode, make the following configuration in IndraWorks:

- Translatory scaling of the virtual axis with a modulo value of (2 x traversing range).

- Rotary scaling of the crank drive (real axis) with a modulo value of 360°.



To configure the scaling factors, IndraWorks has to be in online mode and the drive has to be in parameterizing mode.

Use case *The following sequential chart applies to the single-axis mode:*

1. The MB_CamTableCrank function block calculates the transformation cam.
2. The calculated transformation cam is written to the crank drive by the MB_WriteListParameter function block.

The function block MB_WriteListParameter expects an ARRAY OF REAL for transmission, but the function block MB_CamTableCrank generates an ARRAY OF DINT. Thus, before using the MB_WriteListParameter, the ARRAY OF DINT has to be converted into an ARRAY OF REAL.

The following code example shows the conversion.

```
FOR i := 0 TO 1023 DO
  arRealCamTable[i] := DINT_TO_REAL(arDintCamTable[i])/1000000;
END_FOR
```

3. Using the MC_Power function block, the crank drive is activated.
4. The crank drive is referenced with the MC_Home function block. This step is necessary, if there is no absolute encoder at the crank drive engine.
5. Input of the actual crank drive position in the function block [MB_PhiToXvirt, page 420.](#)
The actual crank drive position is converted from crank angle degrees to a virtual translatory value in mm.
6. To move the virtual master axis to the actual position of the crank drive, the output value Xvirt of the function block MB_PhiToXvirt has to be transferred to the MC_MoveAbsolute function block for the virtual axis. After this sequence, there is a virtual axis in the actual position of the crank drive and the crank drive is synchronized with the master axis without having executed a synchronization motion.
7. Switching the crank drive to the cam operating mode via MC_CamIn. Setting the CamShaftDistance = 360, gear ratio to 1:1, selecting the transformation cam and the virtual master axis as master axis. The crank drive does not move for synchronization with the virtual master axis, as the virtual master axis has been set to the crank position in step 6. If step 6 has not been conducted, the crank drive conducts a dynamic synchronization motion.
8. Now, the virtual axis can be moved in single-axis mode using the function blocks MC_MoveAbsolute and MC_MoveVelocity (the crank axis now follows the master axis and in doing so uses the transformation cam).

Troubleshooting The function block generates the following error messages in Additional1/Additional2 of the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0001	Pushrod length too small. The following condition has to be fulfilled: Pushrod > radius + ABS(offset)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0002	Xzero is too small. The following condition has to be fulfilled: Xzero > SQRT((Pushrod radius)^2-off-set^2)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0003	Xzero too large. The following condition has to be fulfilled: Xzero < SQRT((Pushrod radius)^2-off-set^2)+2*traversing range
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 > traversing range
CALCULATION_ERROR (16#0007)	16#1001	16#0005	Internal calculation error: (Xmech+Xzero)^2+offset^2 < 0
CALCULATION_ERROR (16#0007)	16#1001	16#0006	Internal calculation error: (Radius^2+(Xmech+Xzero)^2+offset^2+Pushrod^2)/2*radius*SQRT(rVar1) > 1
CALCULATION_ERROR (16#0007)	16#1001	16#0007	Internal calculation error: Xmech+Xzero = 0

Tab. 6-31: Error codes of the MB_CamTableCrank function block

6.3.4 MB_CRANK

The structure MB_Crank is responsible for internal data exchange between function blocks and functions.

The content of the structure MB_CRANK is calculated by the function blocks [MB_CamTableCrank](#), page 410, and [MB_CamTableCrankSuperimposed](#), page 415, and is applicable to all remaining relevant function blocks and functions.

Type declaration

```

TYPE MB_CRANK:
STRUCT
  X0: REAL;      (* Rear pole position of the linear slide in linear units *)
  X1: REAL;      (* Front pole position of the linear slide in linear units *)

  X01: REAL;     (* Entry position in linearized range at the rear pole position *)
  X02: REAL;     (* Exit position out of linearized range at the rear pole position *)
  X11: REAL;     (* Entry position in linearized range at the front pole position *)
  X12: REAL;     (* Exit position out of linearized range at the front pole position *)

  Sv: REAL;      (* Travel range of the translatory axis = ABS(X1-X0) *)

  Phi0: REAL;    (* Rear pole position of the crank in rotary units *)
  Phi1: REAL;    (* Front pole position of the crank in rotary units *)

  Phi01: REAL;   (* Entry angle in linearized range at the rear pole position *)
  Phi02: REAL;   (* Exit angle out of linearized range at the rear pole position *)
  Phi11: REAL;   (* Entry angle in linearized range at the front pole position *)
  Phi12: REAL;   (* Exit angle out of linearized range at the front pole position *)
END_STRUCT
END_TYPE

```

6.3.5 MB_CamTableCrankSuperimposed

Brief description The function block MB_CamTableSuperimposed superimposes the user-defined cam (CamInput) on the transformation cam and return the resulting superimposed cam via the "CamOutput" output.

The transformation cam is calculated in the function block, comparable to the MB_CamTableCrank function block [MB_CamTableCrank, page 410](#).

The superimposition principle is shown in [fig. 6-22 "Principle of the cam superimposition" on page 416](#). The user defined cam has to contain 1,024 data points and has to define the movement of the translatory axis in relation to the master axis (irrespective of the crank kinematics).

The 100 percent value of the table corresponds to the motion

(2 x traversing range = modulo value of the virtual translatory axis).

The calculated superimposed cam has to be written on the crank drive using the MB_WriteListParameter function block and enabled by the PLC program via the function block MC_CamIn.

Notes for users

Depending on the corresponding end point of the user-defined cam, the following cases are distinguished:

- If the end point of the user-defined cam is near 100%, the crank does not reverse the direction (but rotations continually in the same direction). Energetically, this is the most economic procedure, as the natural movement of the crank is used.
- If the end point of the user-defined cam is 0%, a see-saw movement of the translatory axis takes place, during which the crank changes direction continually (see figure below).

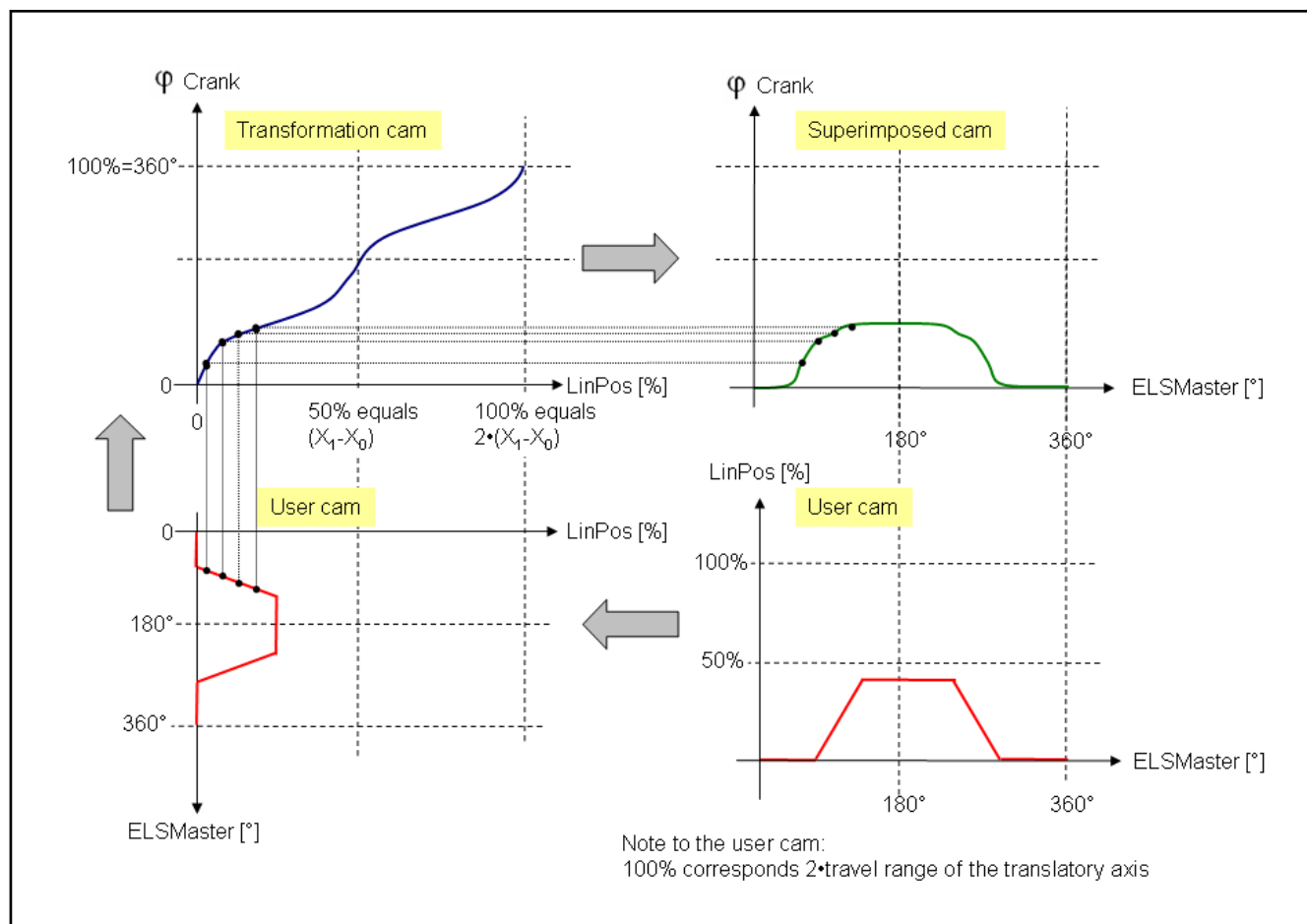


Fig. 6-22: Principle of the cam superimposition



Refers to the application of the point table using **FlexProfile**:

Depending on the function block input data, the start value of the calculated point table is $\neq 0\%$.

When enabling the FlexProfile, the start value = 0 is applied.

Thus, enable the FlexProfile using the input:

SlaveOffset = $(360^\circ/100\%) \cdot \text{first point of the point table } [\%]$

Thus, the correct start value and the crank angle become active.

The table values are present in "ARRAY OF DINT" format and are calculated for six decimal places. The above mentioned calculation has thus to be divided by 1.000.000.

Example code:

```
fbFlexProfile.SlaveOffset:=
    3.6*DINT_TO_REAL(arCamTable[0])/1000000;
```




The lower the inputs "PoleDistance0" and "PoleDistance1", the higher the increase of the calculated cam table in the area around the poles. Thus, the velocity increases at the pole area. This can result in the control error "F5113002, Command velocity too high for cycle time".

Perform the following measures to troubleshoot the error F5113002:

- Reduce Motion cycle time (if possible)
- Reduce master axis velocity (if possible)
- Increase PoleDistance0 and PoleDistance1 inputs

Assignment target system/library

Target system	Library
MLC 14VRS	ML_TechCrank
Function package TEC	
MLD with MPx07/08/17 with MA function package	MX_Packaging_07.lib

Tab. 6-32: Reference table of the MB_CamTableCrankSuperimposed

Interface description

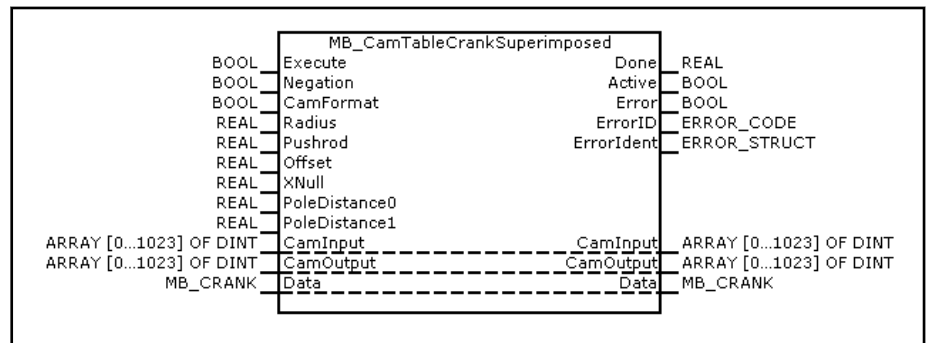


Fig. 6-23: MB_CamTableCrankSuperimposed function block

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Calculation of the superimposed cam disk is triggered by a positive pitch.
	Negation	BOOL	Inverts the direction of counting of a translatory axis
	CamFormat	BOOL	TRUE = new format (last point = first point = 360°) FALSE = old format (last point = 360° - d)
	Radius	REAL	Crank length in [mm]
	Pushrod	REAL	Length of the pushrod in [mm]
	Offset	REAL	Offset of the slide level regarding the crank center [mm]
	Xzero	REAL	Distance between crank center and slide zero point [mm]
	PoleDistance0	REAL	Distance between the rear pole X0 [mm] and the linearized range, in which the movement is influenced by the limited drive dynamics

I/O type	Name	Data type	Description
	PoleDistance1	REAL	Distance between the back pole X1 [mm] and the linearized range, in which the movement is influenced by the limited drive dynamics
VAR_OUTPUT	Done	BOOL	Calculation finished, cam table data and output data is valid
	Active	BOOL	Function block is being executed.
	Error	BOOL	Error (see ErrorID and ErrorStruct)
	ErrorID	ERROR_CODE	Error description
	ErrorStruct	ERROR_STRUCT	Detailed error description Troubleshooting, page 419
VAR_IN_OUT	CamInput	ARRAY[0..1023] OF DINT	Array with data of the user-defined cam to preset the translatory motion profile
	CamOutput	ARRAY[0..1023] OF DINT	Array with data of the calculated superimposed cam to activate the crank.
	Data	MB_CRANK	Structure with internally calculated data of the crank kinematics. The data is calculated in this function block and used by other function blocks MB_CRANK, page 414

Tab. 6-33: Interface variables of the MB_CamTableCrankSuperimposed function block

Boundary condition The following boundary conditions have to be met to prevent PLC error messages:

Boundary condition	Reason
$Rod \geq Radius + Offset $	If this boundary condition is not met, the translatory slide cannot travel mechanically on a constant y-coordinate
$(Xzero \geq rear\ TP) \ \& \ (Xzero \leq rear\ TP + 2 \times traversing\ range)$	The zero point has to be in the traversing range

Tab. 6-34: Boundary conditions

To traverse the translatory slide in single-axis mode, make the following configuration in IndraWorks:

- Translatory scaling of the virtual axis with a modulo value of (2 x traversing range).
- Rotary scaling of the crank drive (real axis) with a modulo value of 360°.

In order to be able to travel the translatory slide in synchronous mode with a specified, user-defined cam, the following configuration has to be conducted in IndraWorks:

- Rotary scaling of the crank drive (real axis) with a modulo value of 360°.



To configure the scaling factors, IndraWorks has to be in online mode and the drive has to be in parameterizing mode.

Use case: Superimposed cam

The following sequential chart applies to a cam-synchronous application:

1. Input of a user-defined cam using the input CamInput of the function block (e.g. by reading a drive cam or reading out from a file, etc.)

2. Output of a superimposed cam by the function block MB_CamTableCrankSuperimposed. It was calculated with the superimposition of a user-defined cam in a transformation cam (see [fig. 6-22 "Principle of the cam superimposition" on page 416](#)).
3. The calculated superimposed cam is written to the crank drive by the MB_WriteListParameter function block.
4. The crank drive is activated using the MC_Power function block.
5. The crank drive is referenced with the MC_Home function block. This step is required if no absolute encoder is used.
6. The position command value of the superimposed cam and the current crank drive position can differ. There are two options to add the cam operation mode (via MC_CamIn, with the setting values CamShaftDistance=360, gear ratio 1:1 and the selection of the superimposed cam):
 - Adding the cam operation mode at the crank drive without previous position calibration. Subsequently, the crank drive executes a dynamic synchronization.
 - Crank drive synchronization with the master axis by entering the master axis position and the superimposed cam in the MB_MasterToPhi function block. Before switching to synchronous mode, the crank drive has to be positioned with the MC_MoveAbsolute function block in a way that it corresponds to the output position of the MB_MasterToPhi function block.

Troubleshooting The function block generates the following error messages in Additional1/ Additional2 of the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0001	Pushrod length too small. The following condition has to be fulfilled: $\text{Pushrod} > \text{radius} + \text{ABS}(\text{offset})$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0002	Xzero is too small. The following condition has to be fulfilled: $\text{Xzero} > \text{SQRT}((\text{Pushrod radius})^2 - \text{offset}^2)$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0003	Xzero too large. The following condition has to be fulfilled: $\text{Xzero} < \text{SQRT}((\text{Pushrod radius})^2 - \text{offset}^2) + 2 * \text{traversing range}$
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	$\text{PoleDistance0} + \text{PoleDistance1} > \text{traversing range}$
CALCULATION_ERROR (16#0007)	16#1001	16#0005	Internal calculation error: $(\text{Xmech} + \text{Xzero})^2 + \text{offset}^2 < 0$
CALCULATION_ERROR (16#0007)	16#1001	16#0006	Internal calculation error: $(\text{Radius}^2 + (\text{Xmech} + \text{Xzero})^2 + \text{offset}^2 + \text{Pushrod}^2) / 2 * \text{radius} * \text{SQRT}(\text{rVar1}) > 1$
CALCULATION_ERROR (16#0007)	16#1001	16#0007	Internal calculation error: $\text{Xmech} + \text{Xzero} = 0$

Tab. 6-35: Error codes of the MB_CamTableCrankSuperimposed function block

6.3.6 MB_PhiToXvirt

Brief description The MB_PhiToXvirt function block converts crank angle values (Phi) into virtual translatory position values (Xvirt).

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechCrank
MLD with MPx08 and MLD with MPx17 with MA function package	MX_Packaging_07.lib

Tab. 6-36: Reference table of the MB_PhiToXvirt

Interface description

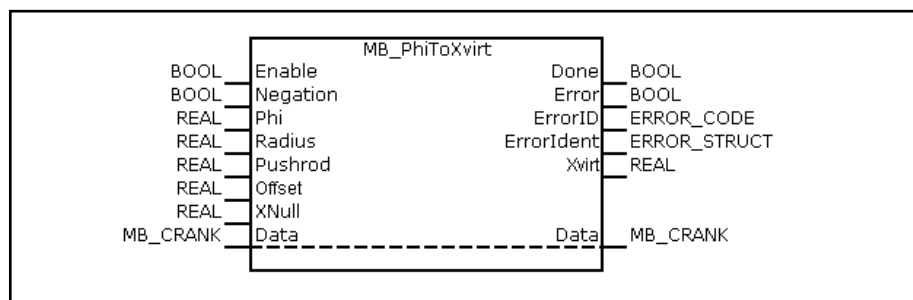


Fig. 6-24: MB_PhiToXvirt function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Calculation of Xvirt in every cycle, while Enable=TRUE
	Negation	BOOL	Inverts the counting direction of the translatory axis
	Phi	REAL	Crank position in [°]
	Radius	REAL	Crank length in [mm]
	Pushrod	REAL	Length of the pushrod in [mm]
	Offset	REAL	Offset of the slide level regarding the crank center [mm]
	Xzero	REAL	Distance between crank center and slide zero point [mm]
VAR_OUTPUT	Done	BOOL	Calculation complete, Xvirt valid
	Error	BOOL	Error (see ErrorID and ErrorStruct)
	ErrorID	ERROR_CODE	Error Description
	ErrorStruct	ERROR_STRUCT	Detailed error description Error handling, page 420
	Xvirt	REAL	Virtual translatory position (Xvirt)
VAR_IN_OUT	Data	MB_CRANK	Structure with the internal data of the crank kinematics (to be calculated before MB_CamTableCrank or MB_CamTableCrankSuperimposed) MB_CRANK, page 414

Tab. 6-37: Interface variables of the MB_PhiToXvirt function block

Error handling The function block generates the following error messages in Additional1/ Additional2 of the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0001	Pushrod length too small. The following condition has to be fulfilled: $\text{Pushrod} > \text{radius} + \text{ABS}(\text{offset})$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0002	Xnull is too small. The following condition has to be fulfilled: $\text{Xnull} > \text{SQRT}((\text{Pushrod radius})^2 - \text{offset}^2)$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0003	Xnull is too high. The following condition has to be fulfilled: $\text{Xnull} < \text{SQRT}((\text{Pushrod radius})^2 - \text{offset}^2) + 2 * \text{traveling range}$
CALCULATION_ERROR (16#0007)	16#1001	16#0008	Internal calculation error: $\text{Pushrod}^2 - (\text{radius} * \text{SIN}(\text{Phi}) - \text{offset})^2 < 0$

Tab. 6-38: Error codes of the MB_PhiToXvirt function block

6.3.7 MB_MasterToPhi

Brief description The function MB_MasterToPhi provides the crank angle (Phi) calculated based on the master axis position (master value) and the (superimposition) cam. The superimposed cam needs to be calculated prior to this function being called up (using [MB_CamTableCrankSuperimposed](#), page 415).

The output value of the MB_MasterToPhi function is used to position the crank drive in such a way before switching to synchronous mode that the position corresponds to the position of the master axis.

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechCrank
MLD with MPx07/08/17 with MA function package	MX_Packaging_07.lib

Tab. 6-39: Reference table of the MB_MasterToPhi

Interface description

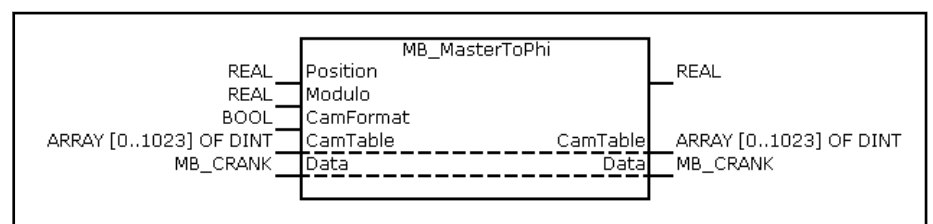


Fig. 6-25: Function MB_MasterToPhi

I/O type	Name	Data type	Description
VAR_INPUT	Position	REAL	Master axis position
	Modulo	REAL	Modulo value of the master axis
	CamFormat	BOOL	TRUE = new format (last point = first point = 360°) FALSE = old format (last point = 360° - d)
Return value		REAL	Crank angle (Phi)

I/O type	Name	Data type	Description
VAR_IN_OUT	CamTable	ARRAY[0..1023] OF DINT	Array with the data of the superimposed cam.
	Data	MB_CRANK	Structure with the internal data of the crank kinematics (to be calculated before MB_CamTableCrank or MB_CamTableCrankSuperimposed) MB_CRANK, page 414

Tab. 6-40: Interface variables of the MB_MasterToPhi function

6.3.8 MB_XvirtToXmech

Brief description The MB_XvirtToXmech function block converts the virtual translatory position (Xvirt) into the mechanical translatory position (Xmech) for the slide. Xmech values can be used for display purposes (e.g. on an HMI device).

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	ML_TechCrank
MLD with MPx08 and MLD with MPx17 with MA function package	MX_Packaging_07.lib

Tab. 6-41: Reference table of the MB_XvirtToXmech

Interface description

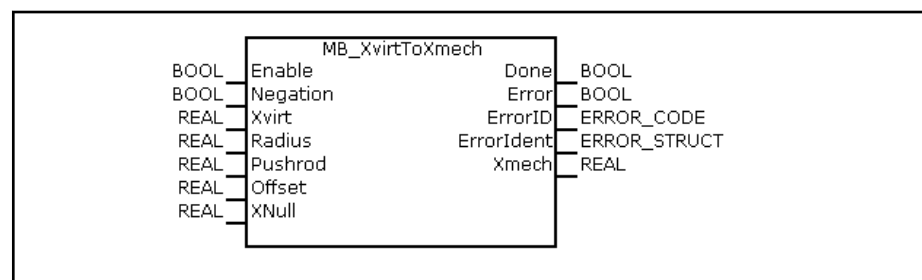


Fig. 6-26: MB_XvirtToXmech function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	The function block calculates as long as "Enable" is TRUE
	Negation	BOOL	Inverts the count direction of the translatory axis
	Xvirt	REAL	Virtual translatory position in [mm]
	Radius	REAL	Crank length in [mm]
	Pushrod	REAL	Length of the pushrod in [mm]
	Offset	REAL	Offset of the slide level regarding the crank center [mm]
	Xzero	REAL	Distance between crank center and slide zero point [mm]

I/O type	Name	Data type	Description
VAR_OUTPUT	Done	BOOL	Output value "Phi" valid
	Error	BOOL	Error (see ErrorID and ErrorStruct)
	ErrorID	ERROR_CODE	Error Description
	ErrorStruct	ERROR_STRUCT	Detailed error description Error handling, page 423
	Xmech	REAL	Mechanical translatory position (Ymech) of the carriage

Tab. 6-42: Interface variables of the MB_XvirtToXmech function block

Error handling

The function block generates the following error messages in Additional1/Additional2 of the table "F_RELATED_TABLE", 16#0170:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0001	Pushrod length too small. The following condition has to be fulfilled: Pushrod > radius + ABS(offset)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0002	Xnull is too small. The following condition has to be fulfilled: $X_{null} > \sqrt{(\text{Pushrod radius})^2 - \text{offset}^2}$
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0003	Xnull is too high. The following condition has to be fulfilled: $X_{null} < \sqrt{(\text{Pushrod radius})^2 - \text{offset}^2} + 2 \cdot \text{traveling range}$

Tab. 6-43: Error codes of the MB_XvirtToXmech function block

7 ML_TechTensionAdvanced.library

7.1 Introduction and overview

The ML_TechTensionAdvanced library provides function blocks for multi-axis tension controllers with decoupling network.



The use of the function blocks of the ML_TechTensionAdvanced library is **subject to license**.

The respective license FWS-IM*MLC*-TEC-NNVRS-NN-TENS*ADV to use the function blocks has to be purchased.

For each control on which the function blocks are used, one license is required each (runtime license) which only permits its use on this control system.

The function blocks of the ML_TechTensionAdvanced

Area of application (folder in library)	
Function block	Description
Parameter identification for tension controllers with simple parameterization	
MB_TensionIdentModelParamType01, page 425	Function block for the online identification of the relevant controlled system parameters
Multi-axis tension controller	
MB_TensionControlLoadCellType02, page 433	Multi-axis tension controller function block with dynamic pre-control for up to eight axes
MB_TensionControlLoadCellMatParType02, page 448	Multi-axis tension controller function block with simple parameterization and dynamic precontrol for up to eight axes
Decoupling network for multi-axis tension controller	
MB_TensionDecouplingNetworkType01, page 464	Decoupling network function block for multi-axis tension controller

Tab. 7-1: Function blocks of the ML_TechTensionAdvanced library (subject to license)

7.2 MB_TensionIdentModelParamType01

Brief description Using the resulting web tension due to system excitation, the MB_TensionIdentModelParamType01 identification block determines the individual controlled system parameters of a web tension control system.

These parameters identified online (controlled system gain K_s , delay time T_u and compensation time T_g) may subsequently be forwarded to the MB_TensionControlLoadCellMatParType01, ML_TechMotion.library and [MB_TensionControlLoadCellMatParType02, page 448](#), function blocks using the structure MB_Tension_CONTROL_MATERIAL (K_s) and the parameters DelayFeedbackValue (T_u) and DistanceRollToRoll ($\sim T_g$).

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechTensionAdvanced

Tab. 7-2: MB_TensionIdentModelParamType01 reference table



The use of the function block is subject to license (for details, refer to [Introduction and overview, page 425](#)).

Interface description

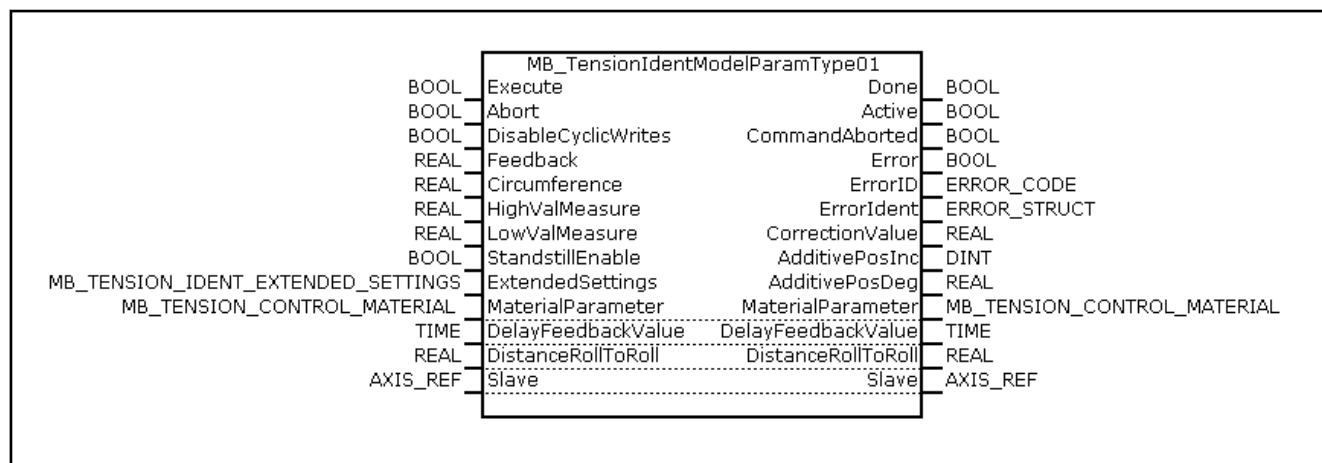


Fig. 7-1: MB_TensionIdentModelParamType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Processing of function block enabled (once, edge-controlled).
	Abort	BOOL	Abort of the ongoing identification.
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output.
	Feedback	REAL	Current actual value of the tension [N]*.
	Circumference	REAL	Axis circumference of the slave axis [mm]*.
	HighValMeasure	REAL	Maximum value which the web tension is to reach during the identification [N]*.
	LowValMeasure	REAL	Minimum value which the web tension is to reach during the identification [N]*.
	StandstillEnable	BOOL	StandstillEnable = FALSE: Identification during web transport, control variable: Gear ratio fine adjustment. StandstillEnable = TRUE: Identification at standstill, control variable: Master axis position, additive.
	ExtendedSettings	MB_TENSION_IDENT_EXTENDED_SETTINGS	Structure containing individual adjustment options for the identification. MB_TENSION_IDENT_EXTENDED_SETTINGS, page 432

I/O type	Name	Data type	Description
VAR_OUTPUT	Done	BOOL	Processing completed without errors, output data valid.
	Active	BOOL	Processing active, output data invalid.
	CommandAborted	BOOL	Processing aborted using the Abort input.
	Error	BOOL	Processing completed with errors. If "Enable" = FALSE, the error is cleared.
	ErrorID	ERROR_CODE	Short error description
	ErrorIdent	ERROR_STRUCT	Detailed error description according to error table (see Error codes, page 431).
	CorrectionValue	REAL	Control signal during web transport (StandstillEnable = FALSE) [%].
	AdditivePosInc	DINT	Control signal during standstill (Standstill = TRUE). Additive master axis position in increments. It is used for the real slave axis with interpolation in the drive (target parameter: P-0-0692) [-].
	AdditivePosDeg	REAL	Control signal during standstill (Standstill = TRUE). Additive master axis position in degrees. It is used for the slave axis with interpolation on the control system (e.g. for virtual axes or real axes with interpolation on the control. Target parameter: A-0-2600) [°].
VAR_IN_OUT	MaterialParameter	MB_TENSION_CONTROL_MATERIAL	Structure with material-specific data (type, width, thickness, grammage and modulus of elasticity). MB_TENSION_CONTROL_MATERIAL, page 292
	DelayFeedbackValue	TIME	Delay time T_u of the web tension control system. Total of all peripheral time constants. E.g. motor dynamics, filter times, sensors, motion cycle time.
	DistanceRollToRoll	REAL	Material web length between the limiting clamping units [mm]*.
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Reference to the slave axis.

*

If user-defined modulus of elasticity are used, you may use any unit such as [N], [kg] and/or [mm], [ft]. Ensure that the selected units are always used.

Tab. 7-3: Interface variables of the MB_TensionIdentModelParamType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL	FALSE	TRUE	FALSE	Continuous
Abort	BOOL	FALSE	TRUE	FALSE	Continuous
DisableCyclicWrites	BOOL	FALSE	TRUE	FALSE	rising edge at "Execute"
Feedback	REAL	n.def.	n.def.	0.0	Continuous
Circumference	REAL	>0.0	n.def.	100.0	rising edge at "Execute"

Name	Type	Min. value	Max. value	Default value	Effective
HighValMeasure	REAL	>Low-ValMeasure	n.def.	0.0	rising edge at "Execute"
LowValMeasure	REAL	0.0	<High-ValMeasure	0.0	rising edge at "Execute"
StandstillEnable	BOOL	FALSE	TRUE	TRUE	rising edge at "Execute"
ExtendedSettings	MB_TENSION_IDENT_EXTENDED_SETTINGS	See "Data structure"	See "Data structure"	See "Data structure"	rising edge at "Execute"
MaterialParameter	MB_TENSION_CONTROL_MATERIAL	See "Data structure"	See "Data structure"	See "Data structure"	rising edge at "Execute"
DelayFeedbackValue	TIME	T#25ms	n.def.	User-defined	rising edge at "Execute"
DistanceRollToRoll	REAL	>0.0	n.def.	User-defined	rising edge at "Execute"
Slave	AXIS_REF	see RIL_CommonTypes	see RIL_CommonTypes	see RIL_CommonTypes	rising edge at "Execute"

Tab. 7-4: MB_TensionIdentModelParamType01 input behavior

Functional description

A web section is always limited by two clamping units.

To control the web tension, you typically use, however, only one of these clamping units.

As system excitation for identification, the MB_TensionIdentModelParamType01 function block will either connect a position to the selected control signal at standstill ("StandstillEnable" := TRUE) or a circumferential velocity during web transport ("StandstillEnable" := FALSE). Using the tension change resulting from this excitation, the function block is able to determine the individual controlled system parameters.

Depending on axis type and operating mode ("StandstillEnable"), corresponding parameters will be written unless cyclic writing has been disabled ("DisableCyclicWrites" := FALSE).

During the web transport, the gear ratio fine adjustment is used as control signal (P-0-0694 at interpolation in the drive or A-0-2605 at interpolation on the control system). If the master axis is at standstill, the draw roller is moved via the additive master axis position, since in this case, the gear ratio fine adjustment does not influence the tension. Depending on the axis type, it is written on the respective parameter (P-0-0692 at interpolation in the drive or A-0-2600 at interpolation on the control).

Commissioning

If "DisableCyclicWrites" is inactive, the two parameters P-0-0694 (Gear ratio fine adjustment) and P-0-0692 (Additive master axis position) have to be applied to the cyclic command value channel of the slave axes if a real slave axis with interpolation in the drive is used. At a rising edge at "Execute", the cyclic data channels are checked. If one of the two parameters is not found, an error message is returned at the error outputs of the function block. If "DisableCyclicWrites" is enabled, the configuration of the cyclic command value data channels is not checked.

For slave axes with interpolation on the control (e.g. virtual axis), the cyclic data channels do not have to be configured.



- The slave axis used has to be in the synchronous operation mode before the "Execute" input is enabled. This can be executed by the AxisInterface or the "MC_GearIn" or "MB_GearInPos" PLCopen function blocks.
- The master axis has to be scaled to modulo 360 degrees

The following explains the individual functions of the function block inputs in detail. In general, all inputs will only be analyzed if there is a rising edge at the "Execute" input.

Exceptions to this rule are the "Abort" and "Feedback" inputs which are continuously accepted. That means that changes made to these inputs are immediately effective without toggling the "Execute" input again.

Abort

Using a rising edge at the "Abort" input, an already started identification can be aborted early. It is e.g. useful if for technical or applicative reasons the web tension resulting from the excitation does not arrive at the "Feedback" input of the function block.

DisableCyclicWrites

The "DisableCyclicWrites" input determines whether the values calculated by the function block are to be written directly via the cyclic channel (FALSE) or whether they are only available at the function block output (TRUE).

If "DisableCyclicWrites" is set to TRUE, it is not checked whether the optional cyclic parameters are also configured.

Feedback

This input represents the actual value of the tension. It can be read via an analog input at the drive or at the Inline I/O for example. The user is responsible for a reasonable scaling.

Circumference

The "Circumference" input refers to the circumference of the "Slave" axis and is used for the internal conversion between an angle adjustment and a distance change or an angular velocity and a circumferential velocity.

During standstill of the plant ("StandstillEnable" := TRUE), this circumference and the web length ("DistanceRollToRoll") are used to establish a directly proportional relationship between an angle adjustment and an elongation adjustment.

$$\Delta \varepsilon(t) = \frac{\text{Circumference}}{\text{DistanceRollToRoll}} \cdot \frac{A-0-2720}{A-0-2721} \cdot \frac{\Delta \varphi(t)}{360^\circ}$$

Fig. 7-2: Conversion between angle and elongation adjustment

HighValMeasure, LowValMeasure

The "HighValMeasure" and "LowValMeasure" inputs allow the user to specify the minimum and maximum permissible web tension during the identification. Following a symmetric excitation there will – assuming a linear system – also be a symmetric reaction of the web tension around the stationary state. Accordingly, the maximum admissible deflection against the stationary web tension and the specified limits will be calculated in the initialization phase of the function block using the following formula:

$$\Delta F = \min \left[\text{HighValMeasure} - F_{\text{stat}}, F_{\text{stat}} - \text{LowValMeasure} \right]$$

Fig. 7-3: Maximum deflection of the web tension around the stationary state

So during the identification, both limits will only be touched in case of symmetric specification of the limits around the stationary value. Otherwise, the web tension will only reach the limit with the shorter distance to the stationary value.

StandstillEnable Setting the "StandstillEnable" input allows for the differentiation between an identification during standstill ("StandstillEnable":=TRUE) and an identification during the material web transport ("StandstillEnable":=FALSE).

During standstill of the plant, you can here determine both, the controlled system gain Ks (MaterialEModul*WebThickness*WebWidth) and the delay time Tu ("DelayFeedbackValue").

Afterwards, the controlled system compensation time Tg and/or the proportional, effective web section length ("DistanceRollToRoll") and the controlled system gain Ks resulting from the processing/climatic influence can be determined in the working point.

ExtendedSettings The "ExtendedSettings" input structure allows for the identification to individual requirements of the user.

1. The "ExtendedSettings.NumMeasureData" input specifies the number of final, redundant measurements for the mean value formation.
2. The "ExtendedSettings.AccuracyFactor" input specifies the minimum precision from which the final mean value formation starts. Apart from that, the mean value formation will be repeated if one value exceeds this precision.
3. The "ExtendedSettings.TimeFactor" input specifies the relation between the excitation frequency and the controlled system reaction. For adequate excitation of the system states, the selected excitation frequency is decelerated by this factor as compared to the controlled system time constant to be expected. In order to reduce the identification time, this relation can be reduced.
4. The "ExtendedSettings.ExcitationFactor" input specifies the maximum increase in the excitation amplitude per iteration cycle. If an identification is started without a prior information, this value can be increased in order to reduce the identification time.
5. The "ExtendedSettings.EModuleSaftyFactor" input specifies the reduction of the initial excitation which is calculated on the basis of the given modulus of elasticity. If, for example, an identification is started with the exact values, the excitation amplitude without safety factor would theoretically result in the system reaction exactly touching the "HighValMeasure" and/or "LowValMeasure" limits. If the effective modulus of elasticity is now increased due to thermal or hygroscopic influence, the specified limits would – without reduction – already be exceeded and/or undershot upon initial excitation.
6. The "ExtendedSettings.ErrorFactor" input specifies the additional tension range which may maximally occur in addition to the "HighValMeasure" and "LowValMeasure" limits without the function block canceling the process with an error message. The maximum admissible limits may be calculated according to the following formula.

$$\begin{aligned} \text{MaxValue} &= (\text{HighValMeasure} - F_{\text{stat}}) \cdot (1 + \text{ErrorFactor} / 100\%) + F_{\text{stat}} \\ \text{MinValue} &= (\text{LowValMeasure} - F_{\text{stat}}) \cdot (1 + \text{ErrorFactor} / 100\%) + F_{\text{stat}} \end{aligned}$$

Fig. 7-4: Limits of the web tension before the function block cancels the process with an error message

MaterialParameter The "MaterialParameter" structure contains material-specific information and is used by the web tension controllers MB_TensionControlLoadCellMatParType01 and MB_TensionControlLoadCellMatParType02 for the individual calculation of the controller parameters.

([MB_TensionControlLoadCellMatParType01](#), page 279, and [MB_TensionControlLoadCellMatParType02](#), page 448)

The individual elements of this structure describe the type, the width, the thickness and/or the grammage and the modulus of elasticity of the web material.

For web materials consisting of paper, the grammage "WebGrammage" and for other web materials the material thickness "WebThickness" is considered.

When selecting a defined web material, the initial modulus of elasticity is taken from the internal database. When specifying a user-defined modulus of elasticity, the "MaterialParameter.MaterialEModule" input is used as initial value.

After successful identification, the function block will switch over to the corresponding user-defined material type "CUSTOM_PAPER" or "CUSTOM_NON_PAPER", if necessary, and output the identified modulus of elasticity via the "MaterialParameter.MaterialEModule" variable.

DelayFeedbackValue

The "DelayFeedbackValue" input/output is used by the MB_TensionControlLoadCellMatParType01 and MB_TensionControlLoadCellMatParType02 web tension controllers for the internal calculation of the controller parameters.

The input/output describes the delay time T_u of the web tension control system. So this value corresponds to the total of peripheral time constants such as motor dynamics, filter times, sensors, motion cycle time.

DistanceRollToRoll

The "DistanceRollToRoll" input/output is used by the MB_TensionControlLoadCellMatParType01 and MB_TensionControlLoadCellMatParType02 web tension controllers for the internal calculation of the controller parameters.

The input/output is proportional to the compensation time T_g of the web tension control system. It describes the individual material web length between the two limiting clamping units.

Error codes

The function block generates the following error messages in Additional1/Additional2 for the table F_RELATED_TABLE, 16#0170:

ErrorID	Additional1	Additional2	Description
ACCESS_ERROR (16#0004)	16#0A00	16#0001	The parameter P-0-0694 is not parameterized in the optional cyclic telegram of the slave axis
ACCESS_ERROR (16#0004)	16#0A00	16#0002	The parameter P-0-0692 is not parameterized in the optional cyclic telegram of the slave axis
DEVICE_ERROR (16#0008)	16#0A02	16#0004	The StandstillEnable input is FALSE and the slave axis velocity = 0.
DEVICE_ERROR (16#0008)	16#0A02	16#0005	The motion cycle time (C-0-0400) is smaller or equal to 0
DEVICE_ERROR (16#0008)	16#0A02	16#0006	The gear ratio of the controlled axis is inadmissible (A-0-2720, A-0-2721)
INPUT_INVALID_ERROR (16#0001)	16#0A04	16#0003	The value specified at the "MaterialParameter.MaterialType" input is not contained in the "MB_TENSION_CONTROL_MATERIAL_TYPE" data type.
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0009	The "MaterialParameter.WebWidth" input is less than or equal 0

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0010	The "MaterialParameter.WebThickness" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0011	The "MaterialParameter.MaterialEModul" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0012	The "MaterialParameter.WebGrammage" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0014	The "Circumference" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0015	The "DistanceRollToRoll" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0016	The value of the "LowValMeasure" input is larger than or equal to the "HighValMeasure" input
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0017	The "LowValMeasure" input is less than 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0018	The "ExtendedSettings.NumMeasureData" input = 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0019	The "ExtendedSettings.AccuracyFactor" input is less than 1 or larger than 100
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0020	The "ExtendedSettings.TimeFactor" input is less than 10 or larger than 100
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0021	The "ExtendedSettings.ExcitationFactor" input is less than 10 or larger than 1000
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0022	The "ExtendedSettings.EModuleSaftyFactor" is less than 0
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0023	The "ExtendedSettings.ErrorFactor" input is less than 10 or greater than 100
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0024	The Feedback input is either smaller than "LowValMeasure" or greater than "HighValMeasure"
INPUT_RANGE_ERROR (16#0006)	16#0A05	16#0025	The Feedback input has exceeded or fallen below an absolute limit "LowValMeasure" or "HighValMeasure" including ExtendedSettings.ErrorFactor
STATE_MACHINE_ERROR (16#0005)	16#0006	16#0000	State machine of the function block is in an invalid status

Tab. 7-5: Error codes of the MB_TensionIdentModelParamType01 function block

7.3 MB_TENSION_IDENT_EXTENDED_SETTINGS

The following table lists the elements of the MB_TENSION_IDENT_EXTENDED_SETTINGS structure.

Name	Type	Min. value	Max. value	Default value	Description
NumMeasureData	USINT	1	n.def.	3	Number of iterations of the final mean value formation [-].
AccuracyFactor	REAL	1.0	100.0	20.0	The final measurements may maximally vary within this percentage value range [%].
TimeFactor	USINT	10	100	25	Relation between excitation frequency and controlled system dynamics [-].
ExcitationFactor	UINT	10	1000	100	Maximum increase in the excitation amplitude per iteration cycle [%].
EModuleSaftyFactor	UINT	0	n.def.	100	Percentage reduction of the initial excitation due to the given modulus of elasticity [%].
ErrorFactor	UINT	10	100	50	Tension range which may occur in addition to the "HighValMeasure" and "LowValMeasure" limits [%].

Tab. 7-6: *MB_TENSION_IDENT_EXTENDED_SETTINGS structure description*

7.4 MB_TensionControlLoadCellType02

Brief description Depending on the measured value of a load cell, the multi-axis tension controller controls the tension of up to eight axes to the desired command value.

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechTensionAdvanced

Tab. 7-7: *Reference table of the MB_TensionControlLoadCellType02*

	The use of the function block is subject to license (for details, refer to Introduction and overview, page 425).
	MLC 12VRS or higher additionally supports slave axes with interpolation on the control.
	The "ResponseCurve" input is declared in the RIL_LoopControl library. Therefore, the RIL_LoopControl library has to additionally be integrated in the library manager if the "ResponseCurve" input is used.

Interface description

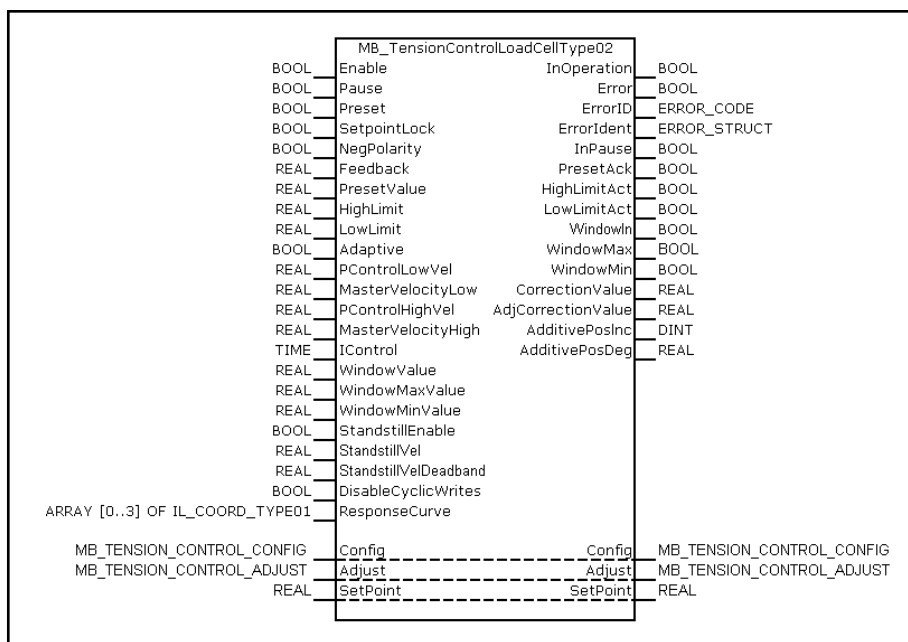


Fig. 7-5: MB_TensionControlLoadCellType02 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	If this input is set, the function block operates
	Pause	BOOL	If this input is set, the tension controller is paused and the I-value of the output value is frozen. The "InPause" output signalizes an active "Pause"
	Preset	BOOL	At the rising edge of this input, the control output is set to the value parameterized in "PresetValue" and, if successful, the "PresetAck" output is set
	SetpointLock	BOOL	At a rising edge at the input, the current actual tension value at "Feedback" is applied as new command value for the tension controller
	NegPolarity	BOOL	If this input is set, the control variables "CorrectionValue" and "AdditivePosInc"/"AdditivePosDeg" are provided with a negative sign
	Feedback	REAL	Current actual value of the tension
	PresetValue	REAL	The controller output is set to this value if the "Preset" input is set
	HighLimit	REAL	Maximum value of the controller output. If the control variable calculated by the controller is greater than this value, the control variable is limited and the I-value of the controller is not summed up anymore
	LowLimit	REAL	Minimum value of the controller output. If the control variable calculated by the controller is lower than this value, the control variable is limited and the I-value of the controller is not summed up anymore
	Adaptive	BOOL	If this input is set, an adaptive P-gain is used for the closed-loop control

I/O type	Name	Data type	Description
	PControlLowVel	REAL	P-gain of the controller if "Adaptive" = FALSE. Otherwise, the value corresponds to the P-gain at low master axis velocities
	MasterVelocityLow	REAL	Only effective if "Adaptive" = TRUE. Master axis velocity up to which "PControlLowVel" is used
	PControlHighVel	REAL	P-gain of the controller at high master axis velocities. If "Adaptive" = FALSE, this input is not effective
	MasterVelocityHigh	REAL	Only effective if "Adaptive" = TRUE. Master axis velocity from which "PControlHighVel" is used
	IControl	TIME	Integral action time of the I-controller. If the value is set to 0, the I-controller is not effective
	WindowValue	REAL	Process variable window in [%]
	WindowMaxValue	REAL	Maximum value of the process variable in [%]
	WindowMinValue	REAL	Minimum value of the process variable in [%]
	StandstillEnable	BOOL	StandstillEnable = FALSE: "InVelocity Control" branch is active. Control variable: Gear ratio fine adjustment StandstillEnable = TRUE: "Standstill control" branch is active. Control variable: Master axis position, additive The additive velocity specified in "StandstillVel" is effective
	StandstillVel	REAL	Additive velocity for standstill control
	StandstillVelDeadband	REAL	This value determines the controller deviation value. Below this value, the tension velocity for the standstill control is reduced
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output. The value is only evaluated at a rising edge at "Enable"
	ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	Characteristics to scale the control deviation in the "In-Velocity Control" branch IL_COORD_TYPE01, RIL_LoopControl.library
VAR_OUTPUT	InOperation	BOOL	The tension controller operates and the outputs are valid
	Error	BOOL	Indicates an error. If "Enable" = FALSE, the error is cleared
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description according to error table (see Error codes, page 447)
	InPause	BOOL	"Pause" is active

I/O type	Name	Data type	Description
	PresetAck	BOOL	If this output is active, the "Preset" input was analyzed and the "CorrectionValue" output applied the "Preset-Value" value
	HighLimitAct	BOOL	This output is set if the control variable calculated by the controller is greater than the value specified in "HighLimit"
	LowLimitAct	BOOL	This output is set if the control variable calculated by the controller is lower than the value specified in "Low-Limit"
	WindowIn	BOOL	If TRUE is set, the process variable is in the parameterized window
	WindowMax	BOOL	If TRUE is set, the process variable exceeded the parameterized maximum value
	WindowMin	BOOL	If TRUE is set, the process variable is below the parameterized minimum value
	CorrectionValue	REAL	Control variable of the tension controller
	AdjCorrectionValue	REAL	Control variable of the tension controller with dynamic precontrol
	AdditivePosInc	DINT	Additive master axis position in increments for each Sercos cycle. It is used for the real slave axis with interpolation in the drive (target parameter: P-0-0692)
	AdditivePosDeg	REAL	12VRS or higher: Additive master axis position in degrees for each Sercos cycle. It is used for the slave axis with interpolation on the control (e.g. for virtual axes or real axis with interpolation on the control. Target parameter: A-0-2600)
VAR_IN_OUT	Config	MB_TENSION_CONTROL_CONFIG	Configuration of the tension controller (axes, distribution of tension) MB_TENSION_CONTROL_CONFIG, page 462
	Adjust	MB_TENSION_CONTROL_ADJUST	Structure for the slave axis settings at runtime MB_TENSION_CONTROL_ADJUST, page 463
	Setpoint	REAL	Tension command value This is generally set by the user but can also be changed by the function block (SetpointLock, page 443)

Tab. 7-8: Interface variables of the MB_TensionControlLoadCellType02 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Preset	BOOL			FALSE	Continuous
SetPointLock	BOOL			FALSE	Continuous

Name	Type	Min. value	Max. value	Default value	Effective
NegPolarity	BOOL			FALSE	Continuous
Feedback	REAL	n.def.	n.def.	0.0	Continuous
PresetValue	REAL	n.def.	n.def.	0.0	Rising edge at "Preset"
HighLimit	REAL	>LowLimit	n.def.	0.00001	Continuous
LowLimit	REAL	-95.0	<HighLimit	-0.00001	Continuous
Adaptive	BOOL			FALSE	Continuous
PControlLowVel	REAL	>0.0	n.def.	0.0001	Continuous
MasterVelocity-Low	REAL	>0.0	<=MasterVelocity-High	0.0	Continuous
PControlHighVel	REAL	>0.0	n.def.	0.0	Continuous
MasterVelocity-High	REAL	>=MasterVelocity-Low	n.def.	1.0	Continuous
IControl	TIME	0 s	n.def.	20 ms	Continuous
WindowValue	REAL	0.0	100.0	0.0	Continuous
WindowMaxValue	REAL	>=WindowMinValue	n.def.	0.0	Continuous
WindowMinValue	REAL	0.0	<=WindowMaxValue	0.0	Continuous
StandstillEnable	BOOL			FALSE	Continuous
StandstillVel	REAL	0.0	n.def.	0.0	Continuous
StandstillDeadband	REAL	0.0	n.def.	1.0	Continuous
DisableCyclic-Writes	BOOL			FALSE	Rising edge at "Enable"
ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	See RIL_Loop-Control	See RIL_Loop-Control	See data structure	Rising edge at "Enable"
Config	MB_TENSION_CONTROL_CONFIG	See data structure	See data structure	See data structure	Rising edge at "Enable"
Adjust	MB_TENSION_CONTROL_ADJUST	See data structure	See data structure	See data structure	Continuous
Setpoint	REAL	n.def.	n.def.	0.0	Continuous

Tab. 7-9: Min./max. and default values of the MB_TensionControlLoadCell-Type02 function block

Functional description

As shown in the following, several draw rollers are controlled by the function block MB_TensionControlLoadCellType02. The measured values of the load cell are assigned to the "Feedback" input of the function block. The command tension is set via the "SetPoint" input. In normal mode ("InVelocity Control" branch is active), the tension is controlled by the gear ratio fine adjustment of the draw rollers. Depending on the axis type, it is written on the respective parameter (P-0-0694 at interpolation in the drive or A-0-2605 at interpolation

on the control). If the master axis is at standstill, the draw rollers are moved via the additive master axis position, since the gear ratio fine adjustment does not influence the tension. Depending on the axis type, it is written on the respective parameter (P-0-0692 at interpolation in the drive or A-0-2600 at interpolation on the control).

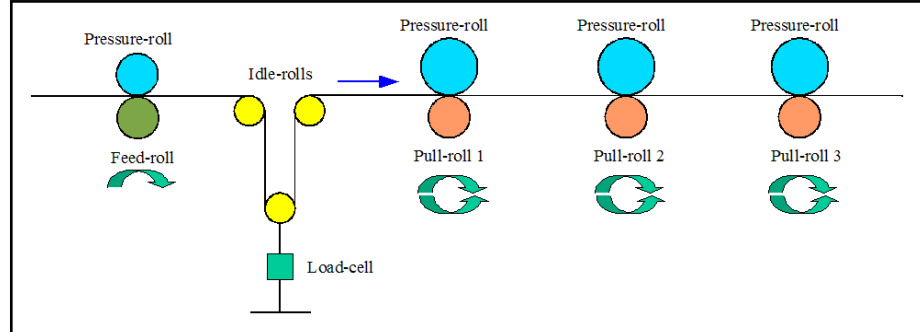


Fig. 7-6: Multi-axis tension controller

The following explains the individual functions of the function block inputs in detail. In general, all outputs are continuously analyzed if the "Enable" input is set. That means that changes made to the inputs are immediately effective without toggling the "Enable" input again.

An exception is the "PresetValue" input which is only applied at a rising edge at the "Preset" input. Likewise, the "DisableCyclicWrites" input is only applied at a rising edge at the "Enable" input.

The complete mode of operation of the tension controller is shown in the following flowcharts. Depending on the "StandstillEnable" input, either the "In-Velocity" path or "Standstill Control" is effective.

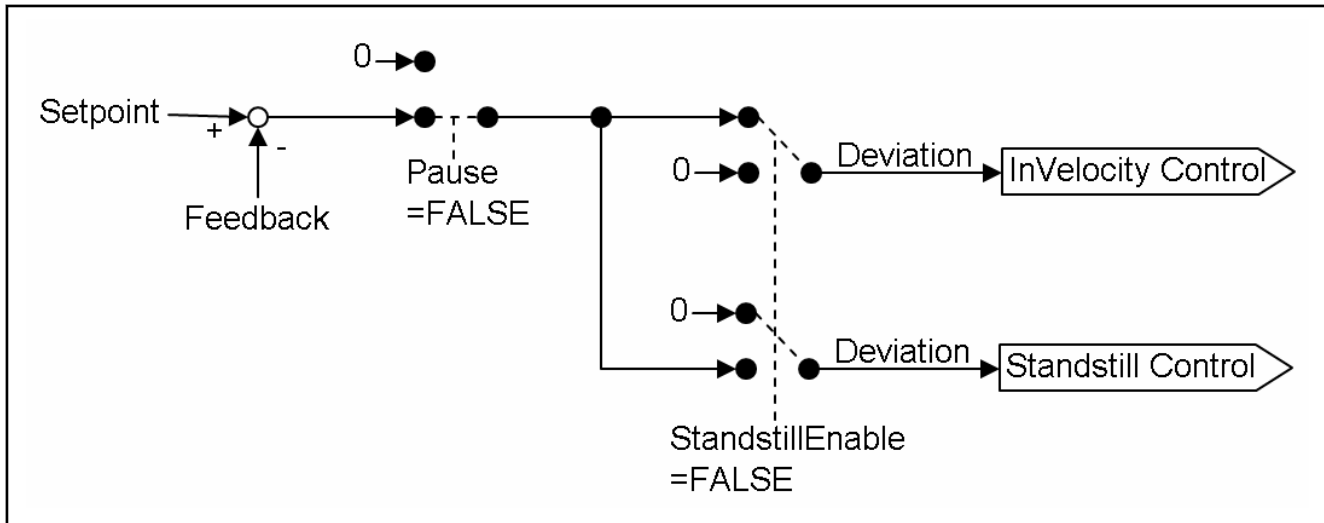


Fig. 7-7: Flowchart of the tension controller with branch for the control at standstill/in velocity

The "InVelocity Control" path is effective if the "StandstillEnable" input is not set. The tension is controlled via the parameter "Gear ratio fine adjustment" P-0-0694 (slave axis with interpolation in the drive/IPO Drive) or A-0-2605 (slave axis with interpolation on the control/IPO Control). It is directly written on the parameters if the "DisableCyclicWrites" input is not set. For real axes with interpolation in the drive, the configuration of the parameter P-0-0694 in the optional cyclic Sercos data (MDT) is thus required.

The control variable is additionally output via the "CorrectionValue" output.

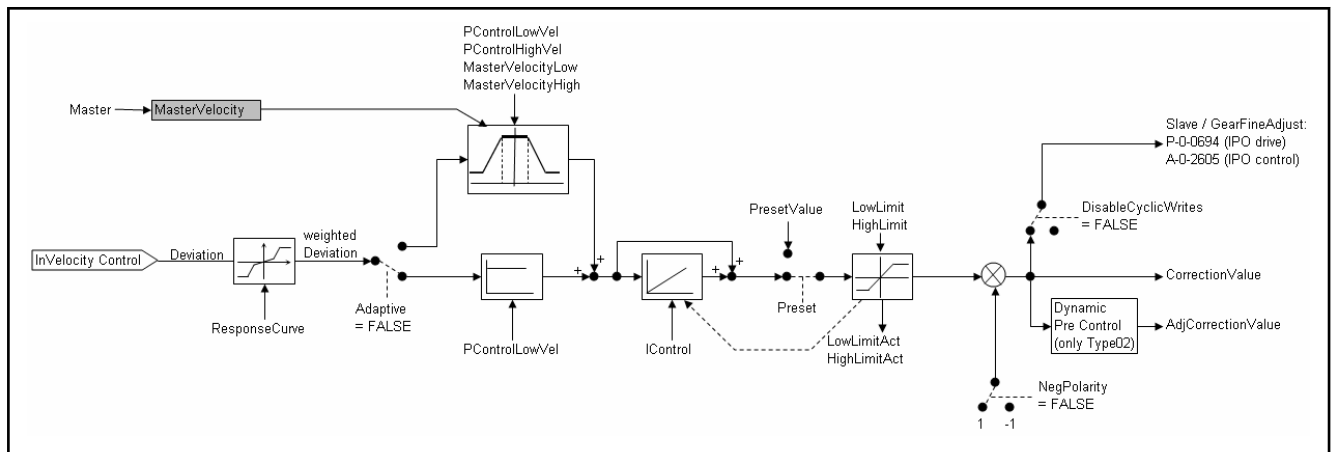


Fig. 7-8: Flowchart of the tension controller for the "InVelocity Control" branch

The "Standstill Control" path is effective if the "StandstillEnable" input is set. This path has an effect on the additive master axis position of the slave axis to specify a velocity. It is directly written on the parameters if the "DisableCyclicWrites" input is not set. For slave axes with interpolation in the drive, it is written on the target parameter P-0-0692 as additive master axis position. For slave axes with interpolation on the control, it is written on the target parameter A-0-2600 as additive master axis position. For real axes with interpolation in the drive, the configuration of the parameter P-0-0692 in the optional cyclic Sercos data (MDT) is thus required.

At the same time, the additive master axis position is available via the "AdditivePosInc" output (at interpolation in the drive) or "AdditivePosDeg" (at interpolation on the control).

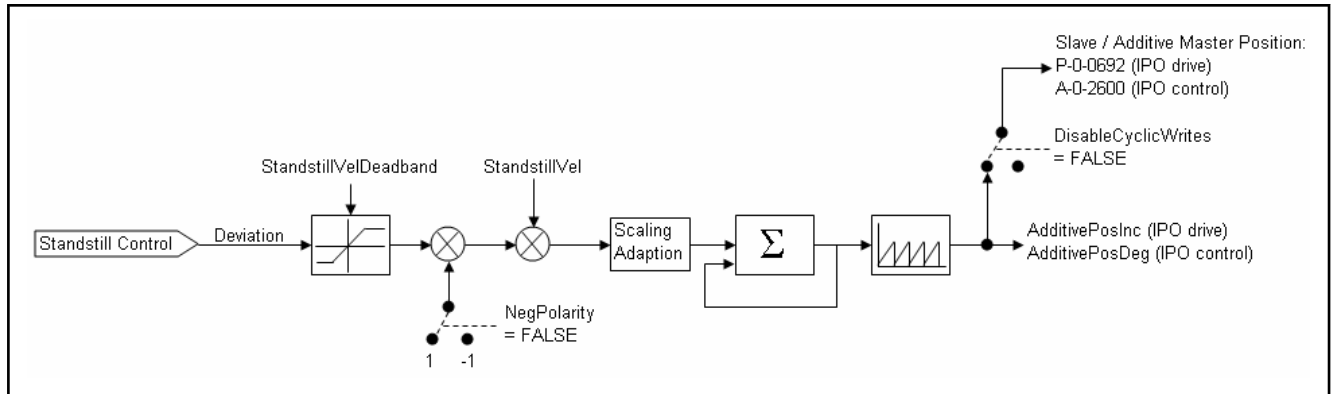


Fig. 7-9: Flowchart of the tension controller for the closed-loop control at standstill (Standstill Control)

To ensure user-friendliness, the flowchart does not contain the "WindowValue", "WindowMaxValue" or "WindowMinValue" limitations and their status displays.

Commissioning

If "DisableCyclicWrites" is inactive, P-0-0694 (Gear ratio fine adjustment) and P-0-0692 (Additive master axis position) parameters have to be applied to the cyclic command value channel of the slave axes if real slave axes (Config.Slave[1..8]) with interpolation in the drive are used. At a rising edge at "Enable", the cyclic data channels are checked. If one of the two parameters is not found, an error message is returned at the error outputs of the function block. If "DisableCyclicWrites" is enabled, the configuration of the cyclic command value data channels is not checked.

For slave axes with interpolation on the control (e.g. virtual axis), the cyclic data channels do not have to be configured.



- The slave axes have to be in the synchronous operation mode before the "Enable" input is enabled. This can be executed by the AxisInterface or by the PLCopen function blocks "GearIn" or "GearInPos". Exception: The synchronous operation mode is not required if the "Pause" input is set
- The master axis has to be scaled to modulo 360 degrees
- The slave axes can be configured as combined (e.g. slave axis 1 with interpolation on the control, slave axis 2 with interpolation in the drive). **In case of a mixed configuration, the default value of the parameter P-0-0750 of each real slave axis with interpolation on the drive has to be 1**

Config Static configuration parameters such as axis configuration and geometry are determined via the "MB_TENSION_CONTROL_CONFIG" structure. The "Master" entry contains the axis reference to the master axis. All draw rollers are specified in the "Slave" array input. An axis that is not present is represented by the "NoAxis" constant. For the dynamic precontrol, the circumference of the first draw roller is required in the "Circumference" structure entry as well as the web length between the feed roll and the first draw roller in the "DistanceRoll1to2" structure entry.

Adjust The "MB_TENSION_CONTROL_ADJUST" structure contains settings that can be changed at runtime. The type of draw roller control is specified in the "SlavePreCtrl" array. If the constant "NO_PRE_CTRL" is entered for an axis, the axis is not controlled. The gear ratio fine adjustment (P-0-0694 or A-0-2605) is set to "0" and cannot be changed.

The constant "PRE_CTRL_100" causes the value of the output "Correction-Value" to be written to the gear ratio fine adjustment.

In the "PRE_CTRL_DYNAMIC" setting, the "AdjCorrectionValue" output is written to the gear ratio fine adjustment of the axis. The dynamic draw roll control is used for all rolls following the first draw roller. To accomplish this, the constant "PRE_CTRL_DYNAMIC" has to be entered in "SlavePreCtrl" of the subsequent draw rollers. The first draw roller has to be on "PRE_CTRL_100". Due to this setting, the tension of the following draw rollers is not affected when the velocity is modified.

The settings in "SlavePreCtrl" do not affect the motion in Standstill mode.

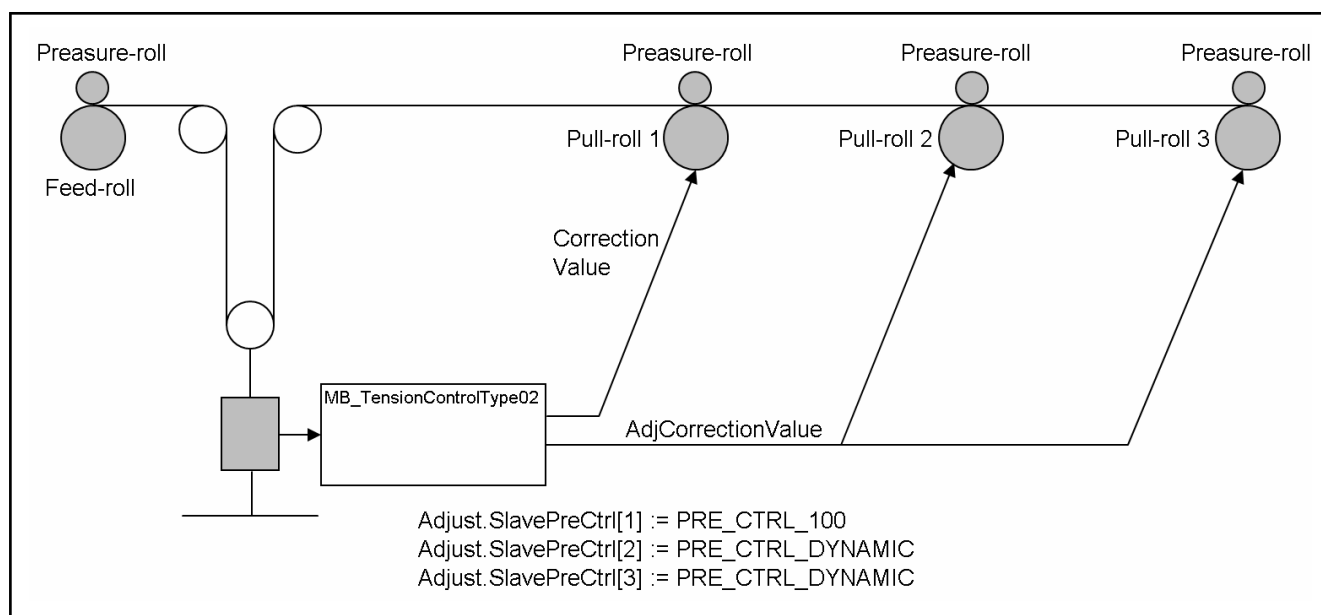


Fig. 7-10: MB_TensionControlType02: Example cascade with three draw rollers

Preset/PresetValid

A positive edge at the "Preset" input specifies an initial value for the control variable using the "PresetValue" variable. At the same time, the internal variables of the controller are set in a way that the controller is approached without any jerk. "Preset" acts only on the gear fine adjustment in the path "InVelocity Control".

The "Preset" input is only evaluated if "Enable" is TRUE. At a rising edge at "Preset", the new value is applied at "PresetValue" and "PresetAck" is set. A falling edge at "Preset" or "Enable" deletes "PresetAck". The "Preset" inputs dominates "Pause". That means that the value at "PresetValue" is applied at the output even though "Pause" is set. If "Preset" is set at the rising edge at "Enable", "PresetValue" is immediately applied as control variable.



The "Preset" function should be executed in standstill (StandstillEnable=TRUE). Otherwise, velocity jumps can occur due to the setting of the "ControlValue" to the "PresetValue". The value of the jump is proportional to the velocity.

The "Preset" input is only effective at the rising edge.

DisableCyclicWrites

The "DisableCyclicWrites" input determines whether the values calculated by the function block are to be written directly via the cyclic channel (FALSE) or whether they are only available at the function block output (TRUE).

Thus, it is possible to cascade several function blocks if "DisableCyclicWrites" is set to TRUE. Subsequently, only the outputs ("CorrectionValue" and "AdditivePosInc" or "AdditivePosDeg") can be used and modified later on. However, the user still has to write the values to the target parameter.

If "DisableCyclicWrites" is set to TRUE, it is not checked whether the optional cyclic parameters are also configured.

ResponseCurve

Using the "ResponseCurve" input, the control deviation can be scaled. Use this functionality to specify a smaller scaling of the control deviation for smaller control deviations. This results in a lower proportional gain and thus causes a smoother control behavior in a steady state. When parameterizing the input, the controller parameters should first be set without evaluation. Subsequently, a lower scaling should be specified. This allows to globally set the

controller parameters first using the system dynamics, and then, a local optimization is ensured (e.g. for the steady state).

The local optimization is parameterized with a characteristic curve with up to four data points. Negative deviations are processed by point reflecting the curve at the origin. This results in a total of nine linear sections. The section s5 runs through the origin. The sections s1 and s9 have a slope of 1 and each run into positive or negative infinity. If the input is not assigned, a 1-to-1 characteristic curve is used as default value. That means that it is not scaled.

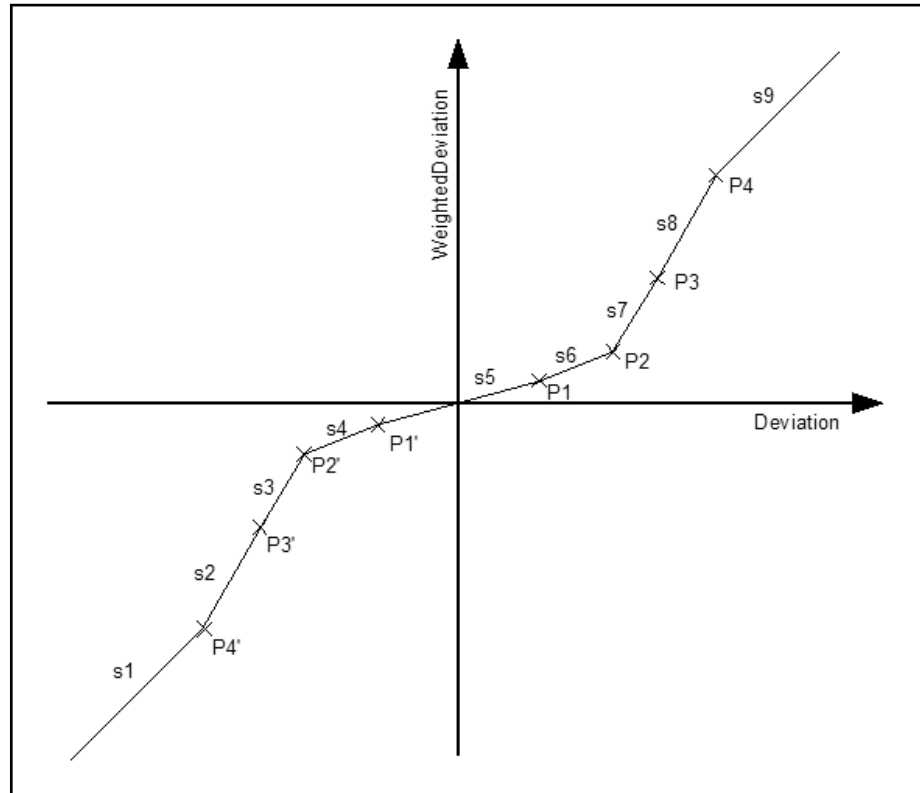


Fig. 7-11: Scaling the control deviation using the "ResponseCurve" input

Points P1 to P4 are arranged in an array made up of IL_COORD_TYPE01. If fewer than four points are required, the X-coordinates of the points not required (beginning with the last point) are set to 0. The X-values have to be positive and rise in a positive direction. It is permitted to use the same point twice.

Adaptive, PControlLowVel, PControlHighVel, MasterVelocityLow, MasterVelocityHigh

The P-gain can either be specified as constant or adaptive P-gain. Via the "Adaptive" input, either an adaptive or a constant P-gain is selected. Setting "Adaptive" to TRUE selects the adaptive P-gain.

If a constant P-gain is selected, only the value in "PControlLowVel" is effective for the P-gain.

If an adaptive P-gain is selected, the "PControlLowVel", "PControlHighVel", "MasterVelocityLow" and "MasterVelocityHigh" inputs have to be assigned. The characteristic curve looks as follows:

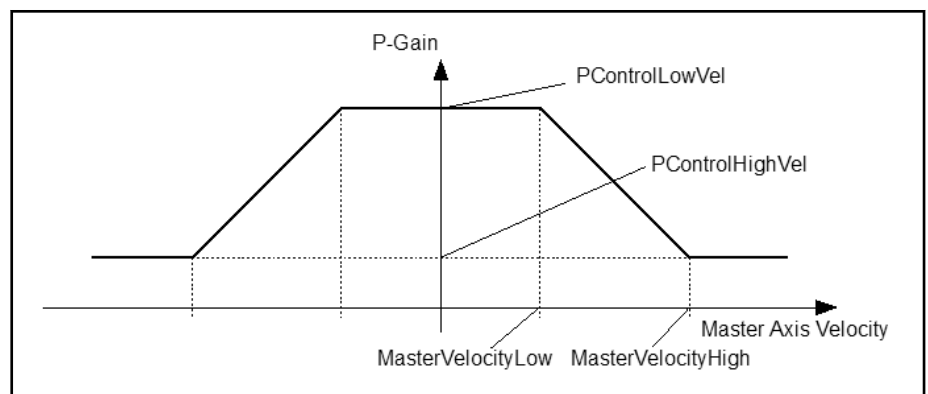


Fig. 7-12: Characteristic curve for the adaptive P-gain

The individual inputs describe the characteristic curve for positive master axis velocities. The characteristic curve for negative master axis velocities results from the mirroring at the axis of the P-gain. This means that only symmetric characteristics are possible. The P-gain between the master axis velocities "MasterVelocityLow" and "MasterVelocityHigh" is linearly interpolated. This results in some conditions for the value ranges of the inputs:

- The values for "MasterVelocityLow" and "MasterVelocityHigh" have to be positive
- The values for "MasterVelocityLow" and "MasterVelocityHigh" may not be identical
- The "MasterVelocityLow" value has to be lower than the "MasterVelocityHigh" value

If one of these conditions is not met, an error is generated at the output of the function block.

SetpointLock

Setting the "SetpointLock" input applies the current actual tension value as the new command value for the tension controller. For example, the function allows the command value of the tension to obtain a reasonable initial value that reflects the current state of the machine.

Pause

The "Pause" input allows the tension controller to pause. If this input is set to TRUE, the control deviation is set to "0". That, the controller output is frozen at the current I value. If "Pause" is active, the "InPause" output is set. At active pause, the synchronization mode is not required and the power (MC_Power) can be switched off.



If no tension control is used at standstill (StandstillEnable is always FALSE), the "Pause" input has to be set at machine standstill.

NegPolarity

The "NegPolarity" input affects the control direction. Depending on the position of the draw roller with regard to the direction the material moves and the position of the feed roll, the control direction has to be either positive or negative. If this input is FALSE, the control direction is positive and a positive control deviation results in a positive control motion. The input also affects the standstill control.

Feedback

This input represents the actual value of the tension. It can be read via an analog input at the drive or at the Inline I/O for example. The user is responsible for a reasonable scaling.

PresetValue

Setting the "Preset" input specifies an initial value for the control variable using the "PresetValue" variable. At the same time, the internal variables of the controller are set in a way that the controller is approached without any jerk.

HighLimit, HighLimitAct This input limits the control variable of the controller to a maximum value. If the limitation occurs, the I-value of the controller is no longer summed up ("Anti-reset-windup"). The maximum value is determined by the maximum value for the gear ratio fine adjustment defined in the drive. If a limitation occurs, the "HighLimitAct" output is set.

LowLimit, LowLimitAct This input limits the control variable of the controller to a minimum value. If the limitation occurs, the I-value of the controller is no longer summed up ("Anti-reset-windup"). The minimum value is -95 irrespective of possible lower values in the drive (also refer to the section "Gear ratio fine adjustment"). If a limitation occurs, the "LowLimitAct" output is set.

WindowValue, WindowIn With the "WindowValue" value, a symmetric window can be defined containing the current actual tension value. The window is defined as follows:

$$\left(SetPoint - |SetPoint| \cdot \frac{WindowValue}{100\%} \right) \leq Feedback \leq \left(SetPoint + |SetPoint| \cdot \frac{WindowValue}{100\%} \right)$$

Fig. 7-13: WindowValue, WindowIn

If this condition is met, the "WindowIn" output is set. If the condition is not met, the output is not set. A limitation is thus not carried out.

WindowMaxValue, WindowMax This value can be used to set the upper value for the actual tension value. The value is evaluated as follows:

$$Feedback > \left(SetPoint + |SetPoint| \cdot \frac{WindowMaxValue - 100}{100} \right)$$

Fig. 7-14: WindowMaxValue, WindowMax

If this condition is met, the "WindowMax" output is set. If the condition is not met, the output is not set. A limitation is thus not carried out.

WindowMinValue, WindowMin This value can be used to specify a lower value for the actual tension value. The value is evaluated as follows:

$$Feedback < \left(SetPoint + |SetPoint| \cdot \frac{WindowMinValue - 100}{100} \right)$$

Fig. 7-15: WindowMinValue, WindowMin

If this condition is met, the "WindowMin" output is set. If the condition is not met, the output is not set. A limitation is thus not carried out.

StandstillEnable By setting the "StandstillEnable" input, the velocity specified in "StandstillVel" is additively added to the axes to be controlled. This allows a closed-loop tension control at standstill as well.



At lower velocities, the influence of the gear ratio fine adjustment decreases continuously. There is also an open control loop at master axis standstill, since the gear ratio fine adjustment does no influence anymore. Thus, the "StandstillEnable" input has to be set when reaching the standstill window in order not to increase the I-value up to the limit in the "InVelocity Control" path by the open control loop.



For axes with interpolation on the control, the "AdditivePosInc" output is given in modulo format 2²⁰ (12VRS or higher).



If this input is used, ensure that the tension controller is only used in the Sercos-synchronous task. If this is not the case, the motor cannot run smoothly or oscillation around the standstill position can result.

StandstillVel

This value indicates the additive velocity sent to the drive when setting the "StandstillEnable" input. The units for the velocity correspond to the scaling set in the drive.



At standstill, the velocity is generated in jumps. Since the velocity has to be reached in one Sercos cycle, the values specified cannot be indefinitely high. If the value selected for the standstill velocity is too high, the drive can oscillate. The maximum value depends on the scaling of the velocity, the maximum acceleration set and the mass connected to the drive. To avoid big velocity jumps, a filter time constant can be defined in the drive parameter P-0-0693. This smoothes the jump.

Background information

Gear ratio fine adjustment

The tension controller generally operates with velocities greater than "0". The gear ratio fine adjustment is an appropriate tool to maintain the tension. However, there are also some boundary conditions that are explained in the following.

The gear ratio fine adjustment is internally limited to a minimum of -95%. If a lower value is entered, the function block outputs an error.

Reason for this limitation is the calculated output value defined as follows at a given input value:

$$\text{Output value} = \text{Input value} * \left(1 + \frac{\text{Gear fine adjust}}{100} \right)$$

Fig. 7-16: Output value

Based on the equation, it can be seen that when the gear ratio fine adjustment approaches 100%, the output value approaches "0". Thus, the control loop practically opens since a modification of the input value does not cause any response at the output. It is even worse at a gear ratio fine adjustment of less than -100%, since the sign of the output value reverses. In the control loop, positive feedback occurs that can result in an unstable system.

StandstillEnable, StandstillVel, StandstillVelDeadband

The standstill control can be executed using the "StandstillEnable" and "StandstillVel" inputs. This adds an additive velocity. To achieve better control quality, it is reduced by the command value. This can be executed with a modified signum function provided with the following transfer function:

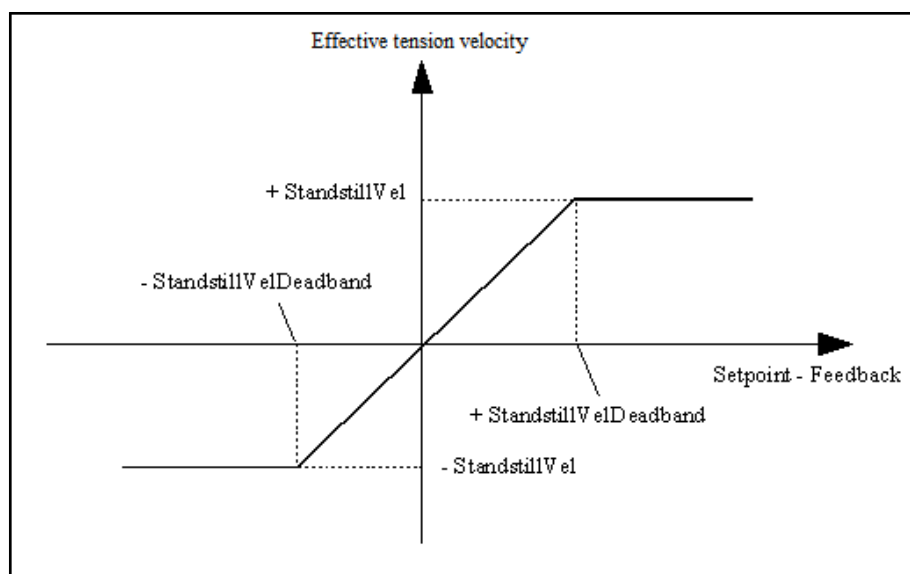


Fig. 7-17: Modified signum function to limit the added "StandstillVel"

The control deviation is relativized in the "Setpoint" range. If the "Feedback" approaches the "Setpoint", the closed-loop control becomes "more sensitive" than a simple two-position controller. The value responsible for the limitation can be set at the "StandstillVelDeadband" input. The values are provided with the same units as the "Setpoint" input or "Feedback" input. Linear interpolation occurs between the two extreme values "+StandstillVel" and "-StandstillVel" so that the transfer function above results for control deviation and tension velocity.

Subsequently, the factor determined is multiplied by the specified "Standstill-Vel". The velocity is processed as speed in the scaling set in the drive. If the web velocity should be entered instead of the speed, calculate it before.

To reverse the direction of motion for the standstill control, set the "NegPolarity" input.

Subsequently, the result is multiplied by a correction factor. This correction factor results from the gear ratio. However, the gear ratio fine adjustment is not considered. The gear ratio fine adjustment is generally very small and therefore not important. The gear ratio is only applied at a rising edge at the "Enable" input.

The flowchart shows that the velocity specified is converted to the additive master axis position in increments or degrees. A few rules are to be considered:

- The specified velocity cannot be unlimited, since the drive has to be able to reach the velocity in one Sercos cycle. Thus, only very small values are allowed. The value size depends on the scaling and the mass connected to the motor. The calculated increments/position are specified via the parameter P-0-0692 (slave axis with interpolation in the drive) or A-0-2600 (slave axis with interpolation on the control). It can also be smoothed via filters P-0-0693 (slave axis with interpolation in the drive) or A-0-2601 (slave axis with interpolation on the control) thus allowing higher jumps
- The calculated increments are added before the gear. The gear ratio (A-0-2720 and A-0-2721 or P-0-0156 and P-0-0157) is included in the calculation as well. The gear ratio fine adjustment (P-0-0083/A-0-2722 and P-0-0694/A-0-2605) is not considered, since it is generally very

small. This can result in the axis moving at a slightly different velocity than the velocity set

- The units are scaled. In addition, the velocity value entered is not checked against the maximum and minimum values. This can result in very large values for the number of increments/position per Sercos cycle. If the calculated intermediate value exceeds the maximum values by $\pm$ half of a modulo value, it is automatically limited. This results in a less additive velocity than specified at the "StandstillVel" input
- At very low velocities, it has to be considered as well that the current velocity can deviate from the specified velocity. This can be caused due to the increments granulated per Sercos cycle. If one increment corresponds to a velocity of 0.03 rpm for example, only velocities that are multiples of 0.03 rpm become effective. If this is not the case, it is rounded up to the next value.
- The calculated increments are processed with each Sercos cycle. Thus, when using the standstill control, the function block has to be processed in the Motion/Sercos-synchronous task. If this is not the case, the drive moves one Sercos cycle by the calculated number of increments and in the following Sercos cycles it comes to standstill again which results in a jerky motion. The filter in P-0-0693 can also be helpful.



For slave axes with interpolation in the drive, it is recommended to set the filter time constant of the P-0-0693 with a quadruple of the Sercos cycle time. If the filter time constant is = "0", the drive outputs the warning E2070 "Velocity limitation activated" when tensioning at standstill since the additive master axis position transferred during the Sercos cycle is not precisely interpolated in the position control cycle in the drive.

Error codes

The function block generates the following error messages in Additional1/Additional2 for the table **F_RELATED_TABLE**, 16#0170:

ErrorID	Additional1	Additional2	Description
ACCESS_ERROR (16#0004)	16#00000A00	16#00000001	The parameter P-0-0694 is not parameterized in the optional cyclic telegram of the slave axis
ACCESS_ERROR (16#0004)	16#00000A00	16#00000002	The parameter P-0-0692 is not parameterized in the optional cyclic telegram of the slave axis
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000001	The master axis velocity in the lower operating point ("MasterVelocity-Low") is < 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000002	The master axis velocity in the upper operating point ("MasterVelocityHigh") is < 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000003	The master axis velocity in the lower operating point ("MasterVelocity-Low") and upper operating point ("MasterVelocityHigh") are provided with the same values

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000004	The value of the master axis velocity in the lower operating point ("MasterVelocityLow") is greater than the value in the upper operating point ("MasterVelocityHigh")
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000005	The value of the "LowLimit" input is lower than the minimum allowed (-95.0)
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000006	The value of the inputs "PGainLowVel" or "PGainHighVel" is lower than or equal to 0
DEVICE_ERROR (16#0008)	16#00000A02	16#00000007	No power is added to the drive specified in "Slave". A closed-loop tension control is therefore not possible
DEVICE_ERROR (16#0008)	16#00000A02	16#00000008	The drive specified in the "Slave" is not in a synchronous mode
INPUT_INVALID_ERROR (16#0001)	16#00000A03	16#00000000	A "SlavePreCtrl" input has an invalid value
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: Error codes of the MB_TensionControlLoadCellType02 function block

7.5 MB_TensionControlLoadCellMatParType02

Brief description

Depending on the measured value of a load cell, the MB_TensionControlLoadCellMatParType02 multi-axis tension controller controls the web tension of a leading web section to the command value, decoupled from the following web tension values.

Compared to the function block [MB_TensionControlLoadCellType02](#), [page 433](#), this variant offers the following benefits.

- Controller parameterization by machine and material data
- Individual controller design (command value or disturbance response)
- Customized control loop dynamics (aperiodic or periodic)
- Automatic adaptation of the controller parameters

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	ML_TechTensionAdvanced

Tab. 7-11: MB_TensionControlLoadCellMatParType02 reference table



The use of the function block is subject to license (for details, refer to [Introduction and overview](#), [page 425](#)).



The "ResponseCurve" input is declared in the RIL_LoopControl library. Therefore, the RIL_LoopControl library has to additionally be integrated in the library manager if the "ResponseCurve" input is used.

Interface description

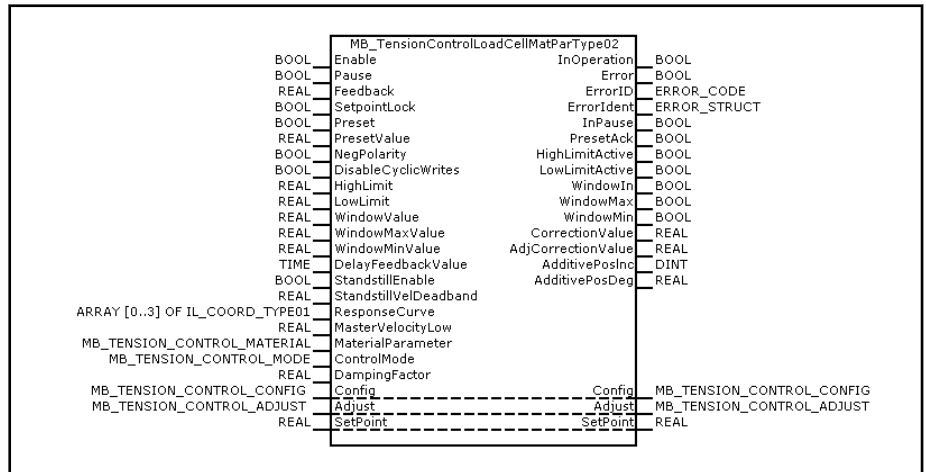


Fig. 7-18: MB_TensionControlLoadCellMatParType02 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	If this input is set, the function block operates
	Pause	BOOL	If this input is set, the tension controller is paused and the output value is frozen. The "InPause" output signalizes an active "Pause"
	Feedback	REAL	Current actual value of the tension [N]*
	SetpointLock	BOOL	With a rising edge at this input, the current "Feedback" actual tension value is applied as new "Setpoint" command value for the tension controller
	Preset	BOOL	With a rising edge at this input, the control output is set to the value parameterized in "PresetValue" and, if successful, the "PresetAck" output is set
	PresetValue	REAL	Value to which the "CorrectionValue" controller output is set if the "Preset" input is set [%]
	NegPolarity	BOOL	If the input is set, the control direction is inverted
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output
	HighLimit	REAL	Maximum value of the controller output [%]. If the control variable calculated by the controller is greater than this value, the control variable is limited and the I-value of the controller is not summed up anymore
	LowLimit	REAL	Minimum value of the controller output [%]. If the control variable calculated by the controller is lower than this value, the control variable is limited and the I-value of the controller is not summed up anymore
	WindowValue	REAL	Process parameter window (relative) [%]

I/O type	Name	Data type	Description
	WindowMaxValue	REAL	Maximum value of the process parameter (absolute) [%]
	WindowMinValue	REAL	Minimum value of the process parameter (absolute) [%]
	DelayFeedbackValue	TIME	Delay time of the web tension determination (excluding the Sercos cycle time)
	StandstillEnable	BOOL	StandstillEnable = FALSE: "InVelocity Control" branch is active. Control variable: Gear ratio fine adjustment StandstillEnable = TRUE: "Standstill control" branch is active. Control variable: Master axis position, additive
	StandstillVelDeadband	REAL	Maximum effective control deviation in the standstill control [N]*. The tensioning velocity is limited according to this control deviation
	ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	Characteristics to scale the control deviation in the "In-Velocity Control" branch IL_COORD_TYPE01, RIL_LoopControl.library
	MasterVelocityLow	REAL	Master axis velocity below which the controller adaptation is limited [1/min]
	MaterialParameter	MB_TENSION_CONTROL_MATERIAL	Structure with material-specific data (type, width, thickness, grammage and modulus of elasticity) MB_TENSION_CONTROL_MATERIAL, page 292
	ControlMode	MB_TENSION_CONTROL_MODE	Selection of the control loop response. Optimized for command value or disturbance response MB_TENSION_CONTROL_MODE, page 293
	DampingFactor	REAL	This input affects the control loop response [%]. It can be steplessly selected between a slow aperiodic and a fast periodic response
VAR_OUTPUT	InOperation	BOOL	The tension controller operates and the outputs are valid
	Error	BOOL	Indicates that an error occurred
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description according to error table (see Error codes, page 461)
	InPause	BOOL	"Pause" is active
	PresetAck	BOOL	If this output is active, the "Preset" input was analyzed and the "CorrectionValue" output applied the "Preset-Value" value
	HighLimitAct	BOOL	This output is set if the control variable calculated by the controller is greater than the value specified in "HighLimit"

I/O type	Name	Data type	Description
	LowLimitAct	BOOL	This output is set if the control variable calculated by the controller is lower than the value specified in "Low-Limit"
	WindowIn	BOOL	If TRUE is set, the process variable is in the parameterized window
	WindowMax	BOOL	If TRUE is set, the process variable exceeded the parameterized maximum value
	WindowMin	BOOL	If TRUE is set, the process variable is below the parameterized minimum value
	CorrectionValue	REAL	Control variable of the tension controller [%]
	AdjCorrectionValue	REAL	Control variable of the tension controller with dynamic precontrol [%]
	AdditivePosInc	DINT	Additive master axis position in increments. It is used for the real slave axis with interpolation in the drive (target parameter: P-0-0692) [-]
	AdditivePosDeg	REAL	Additive master axis position in degrees. It is used for the slave axis with interpolation on the control system (e.g. for virtual axes or real axes with interpolation on the control. Target parameter: A-0-2600) [°]
VAR_IN_OUT	Config	MB_TENSION_CONTROL_CONFIG	Configuration of the tension controller (axes, distribution of tension) MB_TENSION_CONTROL_CONFIG, page 462
	Adjust	MB_TENSION_CONTROL_ADJUST	Structure for the slave axis settings at runtime MB_TENSION_CONTROL_ADJUST, page 463
	Setpoint	REAL	Tension command value [N]*. This is generally set by the user but can also be changed by the function block (see SetpointLock, page 454)

*

The unit [N] is obligatory if the internal material database (E [N/mm²]) is used. If user-defined modulus of elasticity are used, you may use any unit such as [kg]. Ensure that the selected unit is always used.

Tab. 7-12: *Interface variables of the MB_TensionControlLoadCellMatParType02 function block*

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL	FALSE	TRUE	FALSE	Continuous
Pause	BOOL	FALSE	TRUE	FALSE	Continuous
Feedback	REAL	n.def.	n.def.	0.0	Continuous
SetpointLock	BOOL	FALSE	TRUE	FALSE	Continuous
Preset	BOOL	FALSE	TRUE	FALSE	Continuous
PresetValue	REAL	-95	n.def.	0.0	Rising edge at "Preset"

Name	Type	Min. value	Max. value	Default value	Effective
NegPolarity	BOOL	FALSE	TRUE	FALSE	Continuous
DisableCyclic-Writes	BOOL	FALSE	TRUE	FALSE	Rising edge at "Enable"
HighLimit	REAL	>LowLimit	n.def.	200	Continuous
LowLimit	REAL	-95.0	<HighLimit	-50	Continuous
WindowValue	REAL	n.def.	n.def.	0.0	Continuous
WindowMaxValue	REAL	>=WindowMinValue	n.def.	0.0	Continuous
WindowMinValue	REAL	n.def.	<=WindowMaxValue	0.0	Continuous
DelayFeedback-Value	TIME	n.def.	n.def.	T#100ms	Continuous
StandstillEnable	BOOL	FALSE	TRUE	FALSE	Continuous
StandstillVel-Deadband	REAL	>0.0	n.def.	1.0	Continuous
ResponseCurve	ARRAY [0..3] OF IL_COORD_TYPE01	See RIL_Loop-Control	See RIL_Loop-Control	See RIL_Loop-Control	Rising edge at "Enable"
MasterVelocity-Low	REAL	>0.0	n.def.	1.0	Continuous
MaterialParameter	MB_TENSION_CONTROL_MATERIAL	See data structure	See data structure	See data structure	Continuous
ControlMode	MB_TENSION_CONTROL_MODE	See data structure	See data structure	See data structure	Continuous
DampingFactor	REAL	20.0	200.0	100.0	Continuous
Config	MB_TENSION_CONTROL_CONFIG	See data structure	See data structure	See data structure	Rising edge at "Enable"
Adjust	MB_TENSION_CONTROL_ADJUST	See data structure	See data structure	See data structure	Continuous
Setpoint	REAL	n.def.	n.def.	0.0	Continuous

Tab. 7-13: MB_TensionControlLoadCellMatParType02 input behavior

Functional description

As shown in the following, several draw rollers are controlled by the MB_TensionControlLoadCellMatParType02 function block. The measured value of the load cell is assigned to the "Feedback" input of the function block.

The command tension is set via the "SetPoint" input. The resulting controller signal is applied to up to 8 draw rollers. This allows this multi-axis web tension controller to control the web tension of the leading web section decoupled from the following web tension values.

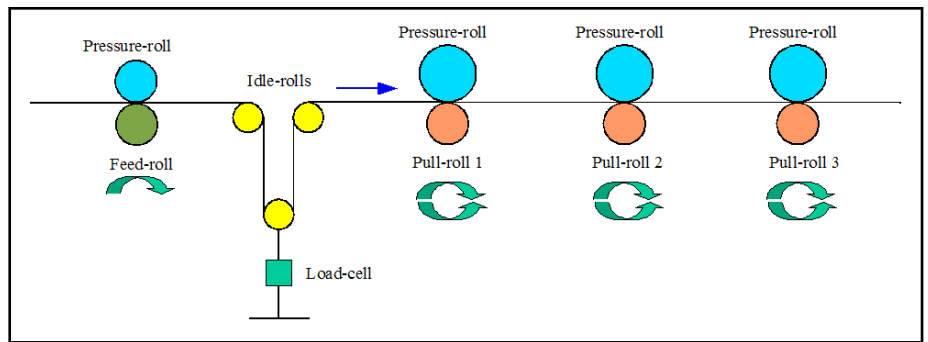


Fig. 7-19: Multi-axis tension controller

The mode of action of the tension controller is shown in the following flow-charts. Depending on the "StandstillEnable" input, either the "InVelocity Control" or "Standstill Control" path is effective.

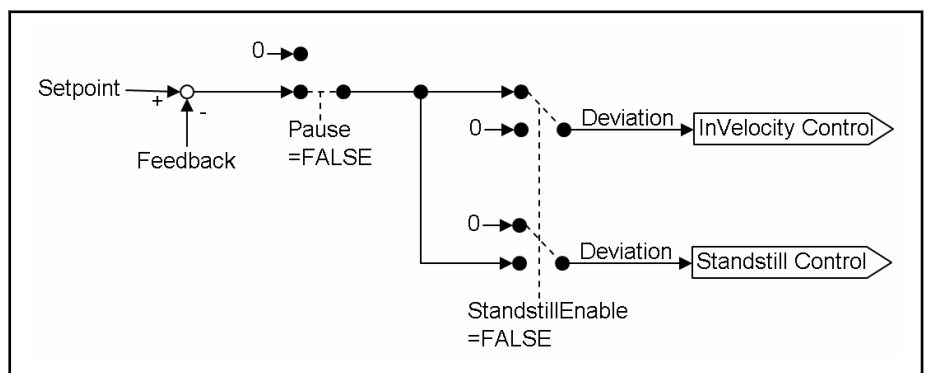


Fig. 7-20: Flowchart of the tension controller with branch for the control at standstill/in velocity

The "InVelocity Control" path is effective if the "StandstillEnable" input is not set. The tension is controlled via the parameter "Gear ratio fine adjustment" P-0-0694 (slave axis with interpolation in the drive/IPO Drive) or A-0-2605 (slave axis with interpolation on the control/IPO Control).

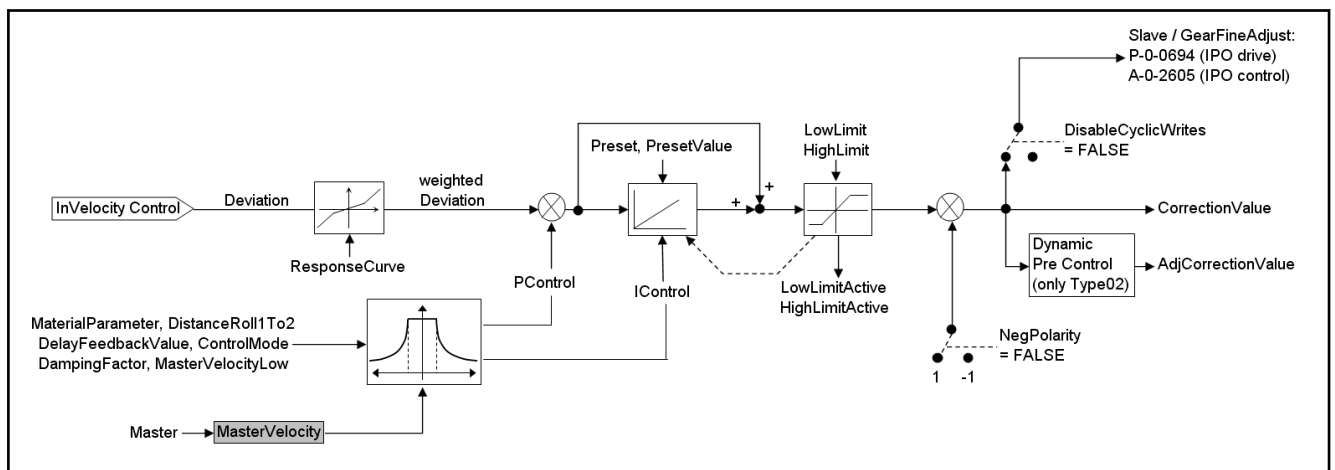


Fig. 7-21: Flowchart of the tension controller for the (InVelocity Control) branch

For the control during standstill, the "Standstill Control" path may be activated. This path is effective if the "StandstillEnable" input is set. The tension is controlled via the "additive master axis position" P-0-0692 (IPO Drive) or A-0-2600 (IPO Control) parameter.

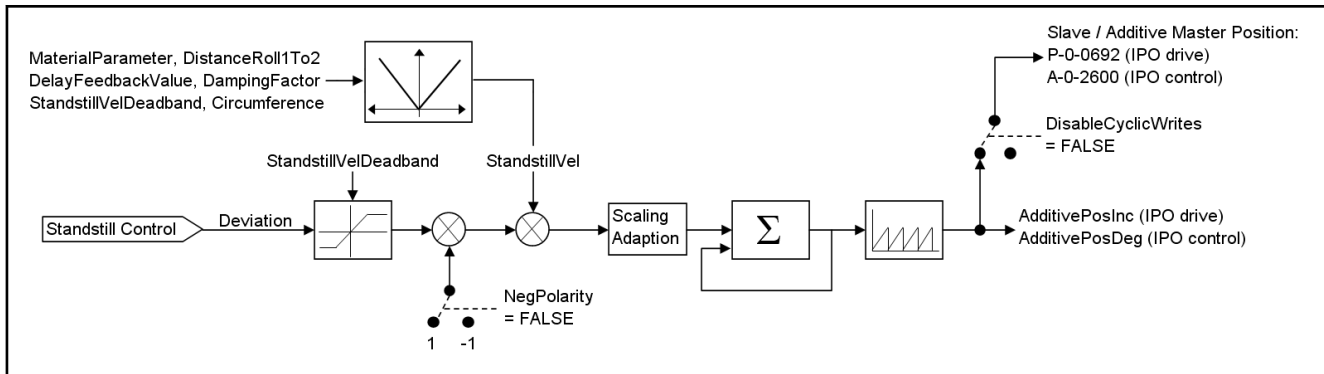


Fig. 7-22: Flowchart of the tension controller for the closed-loop control at standstill (Standstill Control)

To ensure user-friendliness, the flowcharts do not contain the "WindowValue", "WindowMaxValue" or "WindowMinValue" limitations and their status displays.

Commissioning

If "DisableCyclicWrites" is inactive and real slave axes (Config.Slave[1..8]) with interpolation are used in the drive, the parameters P-0-0694 ("InVelocity Control") and P-0-0692 ("Standstill Control") have to be applied to the cyclic command value channel of the slave axis. At a rising edge at "Enable", the cyclic data channels are checked. If one of the two parameters is not found, an error message is returned at the error outputs of the function block. If "DisableCyclicWrites" is enabled, the configurations of the cyclic command value data channels are not checked.



The slave axes can be configured as combined (e.g. slave axis 1 with interpolation on the control system and slave axis 2 with interpolation in the drive). **In case of a mixed configuration, the default value of the parameter P-0-0750 of each real slave axis with interpolation on the drive has to be 1.**

The following explains the individual function block inputs in detail.

In general, all inputs are continuously analyzed if the "Enable" input is set. That means that changes made to the inputs are immediately effective without toggling the "Enable" input again. One exception to this rule are the "PresetValue", "DisableCyclicWrites", "ResponseCurve", and "Config" inputs. In this connection, the "PresetValue" input will only be accepted in case of a rising edge at the "Preset" input and the "DisableCyclicWrites", "ResponseCurve", and "Config" inputs only in case of a rising edge at the "Enable" input.

Pause, InPause

The "Pause" input allows the tension controller to pause. If this input is set to TRUE, the control deviation is set to "0", i.e. the control output is frozen at the current I-value. If "Pause" is active, the "InPause" output is set.



If no tension control is used at standstill (StandstillEnable is always FALSE), the "Pause" input has to be set at machine standstill.

Feedback

This input represents the actual value of the tension. It can be read via an analog input at the drive or at the Inline I/O for example. The user is responsible for a reasonable scaling.

SetpointLock

Setting the "SetpointLock" input applies the current actual tension value as the new "Setpoint" command value for the tension controller. For example, the function allows the command value of the tension to obtain a reasonable initial value that reflects the current state of the machine.

Preset, PresetValue, PresetAck

A positive edge at the "Preset" input specifies an initial value for the "CorrectionValue" control variable using the "PresetValue" variable. At the same time, the internal variables of the controller are set in a way that the controller is approached without any jerk. "Preset" acts only on the gear fine adjustment in the path "InVelocity Control".

The "Preset" input is only evaluated if "Enable" is TRUE. At a rising edge at "Preset", the new value is applied at "PresetValue" and "PresetAck" is set. A falling edge at "Preset" or "Enable" deletes "PresetAck". The "Preset" inputs dominates "Pause". That means that the value at "PresetValue" is applied at the output even though "Pause" is set. If "Preset" is set at the rising edge at "Enable", "PresetValue" is immediately applied as control variable.



The "Preset" function is to be executed in standstill (StandstillEnable=TRUE). Otherwise, velocity jumps can occur due to the setting of the "CorrectionValue" to the "PresetValue". The value of the jump is proportional to the velocity.

The "Preset" input is only effective at the rising edge.

NegPolarity

The "NegPolarity" input affects the control direction. Depending on the position of the draw roller with regard to the direction the material moves and the position of the feed roll, the control direction has to be either positive or negative. If this input is FALSE (positive control direction), a positive control deviation leads to a positive control motion. The input also affects the standstill control.

DisableCyclicWrites

The "DisableCyclicWrites" input determines whether the values calculated by the function block are to be written directly via the cyclic channel (FALSE) or whether they are only available at the function block output (TRUE).

If "DisableCyclicWrites" is set to TRUE, it is possible to pick up and modify the "CorrectionValue", "AdjCorrectionValue", "AdditivePosInc" and "AdditivePosDeg" controller outputs whereas it is not possible to cyclically apply them to the command value channel of the slave axis. Apart from that, it will in this case not be checked whether the optional cyclic parameters have been configured. This setting is, for example, necessary if the "MB_TensionControlDecouplingNetworkType01" decoupling network is used as here, the controller values are modified first and then written.

HighLimit, HighLimitAct

This input limits the "CorrectionValue" control variable of the controller to a maximum value. If the limitation occurs, the I-value of the controller is no longer summed up ("Anti-reset-windup"). The maximum value is determined by the maximum value for the gear ratio fine adjustment defined in the drive. If a limitation occurs, the "HighLimitAct" output is set.

LowLimit, LowLimitAct

This input limits the "CorrectionValue" control variable of the controller to a minimum value. If the limitation occurs, the I-value of the controller is no longer summed up ("Anti-reset-windup"). The minimum value is -95 irrespective of possible lower values in the drive (also refer to [Gear ratio fine adjustment, page 461](#)). If a limitation occurs, the "LowLimitAct" output is set.

WindowValue, WindowIn

With the "WindowValue" value, a symmetric window can be defined containing the current actual tension value. The window is defined as follows:

$$\left(\text{SetPoint} - |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right) \leq \text{Feedback} \leq \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right)$$

Fig. 7-23: WindowValue, WindowIn

WindowMaxValue, WindowMax

If this condition is met, the "WindowIn" output is set. If the condition is not met, the output is not set. There is only a Boolean output, however no limitation at all.

This value can be used to set the upper value for the actual tension value. The value is evaluated as follows:

$$Feedback > \left(SetPoint + |SetPoint| \cdot \frac{WindowMaxValue - 100}{100} \right)$$

Fig. 7-24: WindowMaxValue, WindowMax

If this condition is met, the "WindowMax" output is set. If the condition is not met, the output is not set. There is only a Boolean output, however no limitation at all.

WindowMinValue, WindowMin

This value can be used to specify a lower value for the actual tension value. The value is evaluated as follows:

$$Feedback < \left(SetPoint + |SetPoint| \cdot \frac{WindowMinValue - 100}{100} \right)$$

Fig. 7-25: WindowMinValue, WindowMin

If this condition is met, the "WindowMin" output is set. If the condition is not met, the output is not set. There is only a Boolean output, however no limitation at all.

DelayFeedbackValue

The "DelayFeedbackValue" input is used for the internal calculation of the controller parameters. It describes the delay time T_u of the web tension control system. So this value corresponds to the total of peripheral time constants such as motor dynamics, filter times, sensors, motion cycle time.

StandstillEnable, StandstillVel-Deadband

Set the "StandstillEnable" input to add an additive velocity to the axis to be controlled in order to control the web tension. As long as the "StandstillVel-Deadband" limitation is not enabled, this speed is directly proportional to the control difference.



At lower velocities, the influence of the gear ratio fine adjustment decreases continuously. There is also an open control loop at master axis standstill, since the gear ratio fine adjustment does no influence anymore. Thus, the "StandstillEnable" input has to be set when reaching the standstill window in order not to increase the I-value up to the limit in the "InVelocity Control" path by the open control loop.



For axes with interpolation on the control system, the "AdditivePosInc" output is given in modulo format 2^{20} .

The "StandstillVelDeadband" control signal limitation corresponds to a limitation of the control difference and thus – due to the proportional controller – also a limitation of the control signal. The value is provided with the same units as the "Setpoint" input or the "Feedback" input.

If the difference between "Setpoint" and "Feedback" is larger than the value specified at the "StandstillVelDeadband" input, the tension velocity will be limited. This tension velocity limitation is e.g. relevant for a newly inserted, non-tensioned material web. Accordingly, this input is to be set so that control differences relevant with regard to the control technology can be corrected without limitation and control differences caused by the process are considered by the limited tension velocity.

A "StandstillVelDeadband" of 5% to 25% of the "Setpoint" should be sufficient for the most processes.



At standstill, the axis velocity is generated in jumps. Since the velocity has to be reached in one Sercos cycle, the values specified cannot be indefinitely high and the tension controller has to be used in the Sercos-synchronous task. The maximum value depends on the maximum acceleration of the drive and the connected mass.

In order to prevent large velocity jumps, you can on the one hand increase the filter time constant in the drive parameter P-0-0693 (4 times the Sercos cycle time is recommended) and/or on the other hand limit the maximum control signal by means of the "StandstillVelDeadband" standstill window.

To reverse the direction of motion for the standstill control, set the "NegPolarity" input.

ResponseCurve

Using the "ResponseCurve" input, the control deviation can be scaled. Use this functionality to specify a smaller scaling of the control deviation for smaller control deviations. This results in a lower proportional gain and thus causes a smoother control behavior in a steady state. When parameterizing the input, the controller parameters should first be set without evaluation. Subsequently, a lower scaling should be specified. This allows to globally set the controller parameters first using the system dynamics, and then, a local optimization is ensured (e.g. for the steady state).

The local optimization is parameterized with a characteristic curve with up to four data points. Negative deviations are processed by point reflecting the curve at the origin. This results in a total of nine linear sections. The section s5 runs through the origin. The sections s1 and s9 have a slope of 1 and each run into positive or negative infinity. If the input is not assigned, a 1-to-1 characteristic curve is used as default value. That means that it is not scaled.

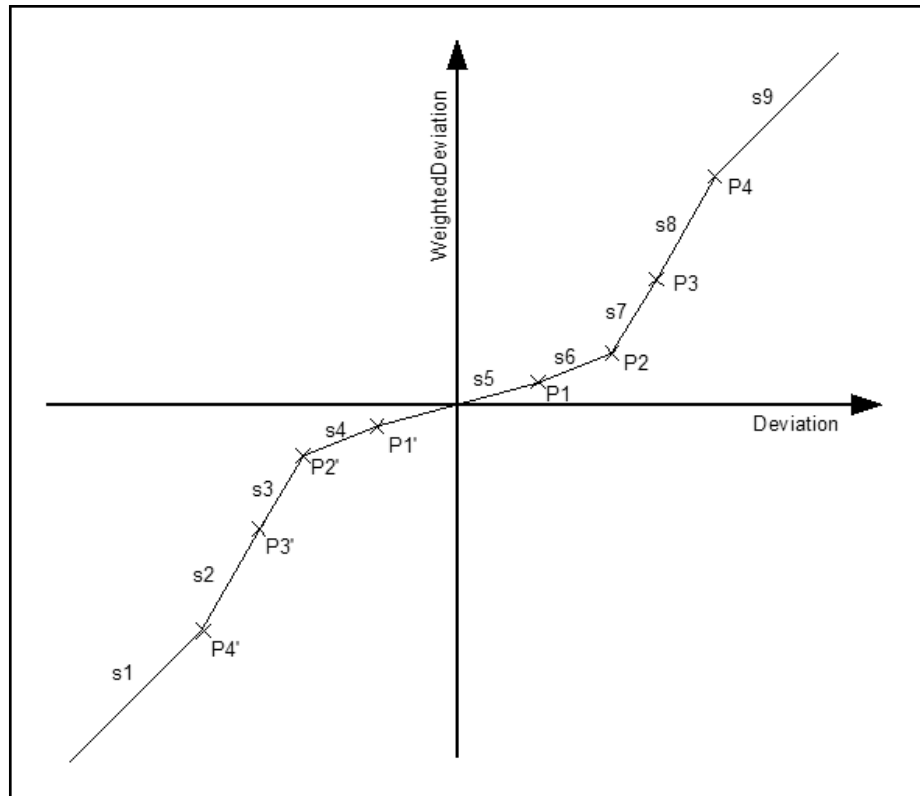


Fig. 7-26: Scaling the control deviation using the "ResponseCurve" input

Points P1 to P4 are arranged in an array made up of IL_COORD_TYPE01. If fewer than four points are required, the X-coordinates of the points not required (beginning with the last point) are set to 0. The X-values have to be positive and rise in a positive direction. It is permitted to use the same point twice.

MasterVelocityLow

The "MasterVelocityLow" input serves to limit the controller parameters. Due to the speed-dependent controlled system response, speed-dependent controller parameters result. Depending on the desired control loop response, either the controller gain and/or the controller reset time is speed-dependent. The following substitutionally represents the dependency of the controller gain from the web velocity in a graphical form.

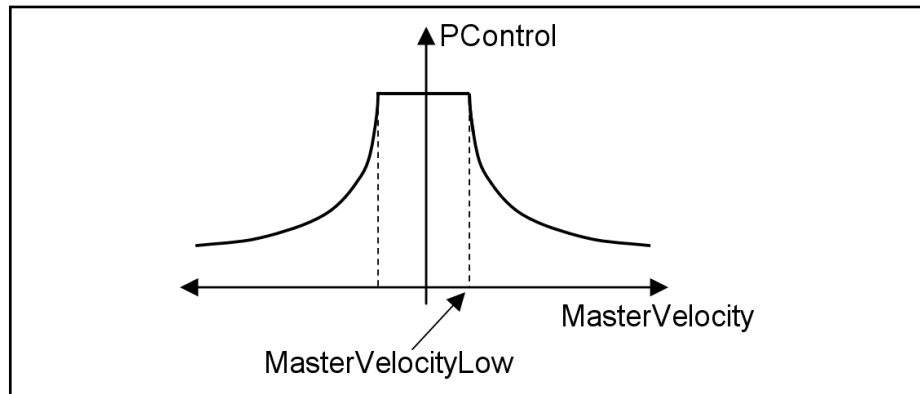


Fig. 7-27: Dependency of the controller gain from the web velocity

As it has already been explained in the section "StandstillEnable", the influence of the gear ratio fine adjustment decreases continuously at lower velocities. This behavior is directly reflected in the controller gain. In order to prevent unnaturally large controller parameters, the "MasterVelocityLow" input

serves to limit the controller parameters in case of "low" web velocities. In most cases, the value of this input should lie below the smallest production speed and above the switchover speed in the standstill path. This ensures unlimited control loop dynamics above the "MasterVelocityLow" velocity and at the same time, the controller parameters are limited in case of "low" web velocities.

MaterialParameter The "MaterialParameter" input structure is used for the internal calculation of the control parameters.

This structure contains information regarding the individual web material. The individual elements of this structure describe the type, the width, the thickness and/or the grammage and the modulus of elasticity of the web material.

For web materials consisting of paper, the grammage "WebGrammage" and for other web materials the material thickness "WebThickness" is analyzed. When selecting a given web material "MaterialParameter.MaterialType", the modulus of elasticity is taken from the internal database. Accordingly, the "MaterialParameter.MaterialEModule" input will only be analyzed if a user-defined material type "CUSTOM_PAPER" or "CUSTOM_NON_PAPER" is selected.

The following two exemplary circuits are to clarify this:

Example:

Paper from block-internal database

MaterialParameter.MaterialType := OFFSET_PAPER;

MaterialParameter.WebWidth := 500;

MaterialParameter.WebGrammage := 80;

Example:

User-defined web material (e.g. foil)

MaterialParameter.MaterialType := CUSTOM_NON_PAPER;

MaterialParameter.WebWidth := 500;

MaterialParameter.WebThickness := 200;

MaterialParameter.MaterialEModule := 650;

If the web material used is not listed in the internal database and if there is no information regarding the modulus of elasticity, it may be determined in a standstill measurement.

$$\Delta F \approx A \cdot E \cdot \frac{\Delta L}{L} \Rightarrow E \approx \frac{L \cdot \Delta F}{A \cdot \Delta L}$$

Fig. 7-28: Hooke's law

You can see from this Hooke's formula that by means of a measured force difference, due to a given elongation (change in length due to an additive master axis position), the individual modulus of elasticity [N/mm²] can be determined at standstill. Apart from that, you can also directly determine the product of modulus of elasticity [N/mm²] and cross-sectional area [mm²], and by standardization of the cross-sectional area ("CUSTOM_NON_PAPER", "WebWidth:=1[mm]", "WebThickness:=1000[μm]") to A=1[mm²], it may be specified as modified modulus of elasticity [N].

ControlMode You can use the "ControlMode" input to determine the primary control loop response. The web tension controller may either be optimized for command

value or for disturbance response. In case of fixed value control of the web tension, "CONTROL_MODE_OPT_DISTURBANCE" is to be assigned to this input. As a result, the internal controller parameters will be determined so that faults are perfectly corrected. The other way round, "CONTROL_MODE_OPT_COMMAND" is to always be assigned to this input in case of sequential control. As a result, the internal controller parameters will be determined so that the controlled variable perfectly follows the reference variable.

- DampingFactor** You can use the "DampingFactor" input to determine the control loop dynamics. It acts on both branches, standstill control and control during web operation. The control loop dynamics may either be set aperiodically (without overshoot) or periodically (with overshoot). For exactly controlled system parameters, a "DampingFactor" of 100% corresponds exactly to an aperiodic boundary case and a "DampingFactor" of 120% corresponds to a transient oscillation with approximately 20% overshoot. In addition, the control loop response can be decelerated accordingly with a "DampingFactor" of up to 20% or can be accelerated with a "DampingFactor" of 200% respectively. If there are controlled system parameters ("Circumference", "DistanceRoll1To2", "DelayFeedbackValue", "MaterialParameter") or a system response deviating from the real behavior (e.g. location- and time-dependent humidities and temperatures), the desired dynamic control loop response can be corrected via this input.
- Config** Static configuration parameters such as axis configuration and geometry are determined via the "MB_TENSION_CONTROL_CONFIG" structure. The "Master" entry contains the axis reference to the master axis. All draw rollers are specified in the "Slave" array input. An axis that is not present is represented by the "NoAxis" constant. For the dynamic precontrol and the internal controller design, the circumference of the feed roll ("Master") is required in the "Circumference" structure entry as well as the web length between the feed roll ("Master") and the first draw roller ("Slave[1]") in the "DistanceRoll1to2" structure entry.
- Adjust** The "MB_TENSION_CONTROL_ADJUST" structure with the "SlavePreCtrl" array determines the type of draw roll control.
- If the constant "NO_PRE_CTRL" is entered for an axis, the axis is not controlled. The gear ratio fine adjustment (P-0-0694 or A-0-2605) is set to "0" and cannot be changed.
- The constant "PRE_CTRL_100" causes the value of the output "Correction-Value" to be written to the gear ratio fine adjustment.
- In the "PRE_CTRL_DYNAMIC" setting, the "AdjCorrectionValue" output is written to the gear ratio fine adjustment of the axis. The dynamic draw roll control should be used for all rolls following the first draw roller. To accomplish this, the constant "PRE_CTRL_DYNAMIC" has to be entered in "SlavePreCtrl" of the subsequent draw rollers (2, 3). The first draw roller (1) should be on "PRE_CTRL_100". As a consequence of this setting, the web tension of the leading web section is controlled decoupled from the following web tension values.
- The settings in "SlavePreCtrl" do not affect the motion in Standstill mode.

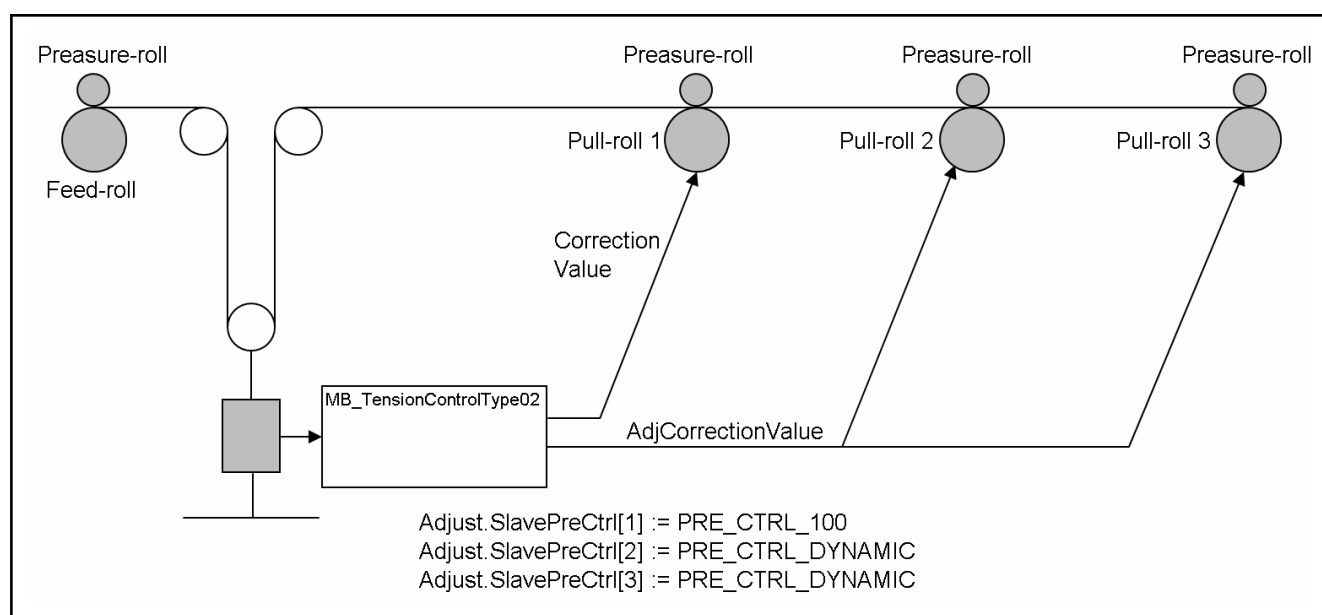


Fig. 7-29: *MB_TensionControlLoadCellMatParType02 example cascade with 3 draw rollers*

Background information

Gear ratio fine adjustment

The tension controller generally operates with velocities greater than "0". The gear ratio fine adjustment is an appropriate tool to control the tension.

The gear ratio fine adjustment is internally limited to a minimum of -95%. If a lower value is entered, the function block outputs an error.

Reason for this limitation is the calculated output value defined as follows at a given input value:

$$\text{Output value} = \text{Input value} * \left(1 + \frac{\text{Gear fine adjust}}{100} \right)$$

Fig. 7-30: *Output value*

It can be seen from this formal description that a gear ratio fine adjustment of less than "-100%" results in an inversion of the direction of rotation.

Error codes

The function block generates the following error messages in Additional1/ Additional2 for the table **F_RELATED_TABLE**, 16#0170:

ErrorID	Additional1	Additional2	Description
ACCESS_ERROR (16#0004)	16#00000A00	16#00000001	The parameter P-0-0694 is not parameterized in the optional cyclic telegram of the slave axis
ACCESS_ERROR (16#0004)	16#00000A00	16#00000002	The parameter P-0-0692 is not parameterized in the optional cyclic telegram of the slave axis
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000001	The master axis velocity in the lower operating point ("MasterVelocity-Low") is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A01	16#00000005	The value of the "LowLimit" input is lower than the minimum allowed (-95.0)

ErrorID	Additional1	Additional2	Description
DEVICE_ERROR (16#0008)	16#00000A02	16#00000005	The Sercos cycle time (C-0-0503) is less than or equal 0
DEVICE_ERROR (16#0008)	16#00000A02	16#00000006	The gear ratio of the controlled axis is inadmissible (A-0-2720, A-0-2721)
INPUT_INVALID_ERROR (16#0001)	16#00000A03	16#00000000	A "SlavePreCtrl" input has an invalid value
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000000	The Config.Circumference input is less than or equal to 0.0
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000001	The "Config.DistanceRoll1To2" input is less than or equal to 0.0
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000002	The value specified at the "Control-Mode" input is not contained in the "MB_TENSION_CONTROL_MODE" data type.
INPUT_INVALID_ERROR (16#0001)	16#00000A04	16#00000003	The value specified at the "MaterialParameter.MaterialType" input is not contained in the "MB_TENSION_CONTROL_MATERIAL_TYPE" data type.
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000007	The "StandstillVelDeadband" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000009	The "MaterialParameter.WebWidth" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000010	The "MaterialParameter.WebThickness" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000011	The "MaterialParameter.MaterialE-Modul" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000012	The "MaterialParameter.WebGrammage" input is less than or equal 0
INPUT_RANGE_ERROR (16#0006)	16#00000A05	16#00000013	The "DampingFactor" input is less than 20 or higher than 200
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	State machine of the function block is in an invalid status
RESOURCE_ERROR (16#0003)	16#00001503	16#00000000	AxisData is not active in the control. This error occurs only on the MLD

Tab. 7-14: Error codes of the MB_TensionControlLoadCellMatParType02 function block

7.6 MB_TENSION_CONTROL_CONFIG

The MB_TENSION_CONTROL_CONFIG structure is used to configure the tension controller (axes, tension distribution).

Name	Type	Min. value	Max. value	Default value	Description
Master	AXIS_REF			NoAxis	Master axis
Slave[1]	AXIS_REF			NoAxis	Draw roller 1
Slave[2]	AXIS_REF			NoAxis	Draw roller 2
Slave[3]	AXIS_REF			NoAxis	Draw roller 3
Slave[4]	AXIS_REF			NoAxis	Draw roller 4
Slave[5]	AXIS_REF			NoAxis	Draw roller 5
Slave[6]	AXIS_REF			NoAxis	Draw roller 6
Slave[7]	AXIS_REF			NoAxis	Draw roller 7
Slave[8]	AXIS_REF			NoAxis	Draw roller 8
Circumference	REAL	0.0	n.def.	0.0	Axis circumference of the feed roll (MASTER) [mm]*
DistanceRoll1To2	REAL	0.0	n.def.	0.0	Web length between feed roll (master) and first draw roller (slave[1]) [mm]*

*

The unit [mm] is obligatory if the internal material database (E [N/mm²]) is used. If user-defined modulus of elasticity are used, you may use any unit such as [ft]. Ensure that the selected unit is always used.

Tab. 7-15: *MB_TENSION_CONTROL_CONFIG structural description*

7.7 MB_TENSION_CONTROL_ADJUST

The MB_TENSION_CONTROL_ADJUST structure contains the slave axis settings for the individual axes during operation.

This structure uses the enumeration type [MB_TENSION_PRE_CTRL](#), page 463.

Structure element	Type	Default value	Description
SlavePreCtrl[1]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Precontrol for slave axis 1
SlavePreCtrl[2]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Precontrol for slave axis 2
SlavePreCtrl[3]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Precontrol for slave axis 3
SlavePreCtrl[4]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Precontrol for slave axis 4
SlavePreCtrl[5]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Precontrol for slave axis 5
SlavePreCtrl[6]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Precontrol for slave axis 6
SlavePreCtrl[7]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Precontrol for slave axis 7
SlavePreCtrl[8]	MB_TENSION_PRE_CTRL	NO_PRE_CTRL	Precontrol for slave axis 8

Tab. 7-16: *Structural description of MB_TENSION_CONTROL_ADJUST*

7.8 MB_TENSION_PRE_CTRL

The enumeration type MB_TENSION_PRE_CTRL describes possible variants in the precontrol of an axis.

The enumeration type is used by the structure [MB_TENSION_CONTROL_ADJUST](#), page 463.

Element	Description
NO_PRE_CTRL	No precontrol/closed-loop control of the axis
PRE_CTRL_100	Precontrol of the axis without time constant
PRE_CTRL_DYNAMIC	Dynamic precontrol

Tab. 7-17: Structure of enumeration type MB_TENSION_PRE_CTRL

7.9 Decoupling network

7.9.1 MB_TensionDecouplingNetworkType01

Brief description The tension controller controls the tension in a web section to the desired command value (MB_TensionControlLoadCellType01, ML_TechMotion / [MB_TensionControlLoadCellType02](#), page 433) depending on the measured value of a load cell.

The decoupling network MB_TensionDecouplingNetworkType01 can be used to forward a control signal change of any web tension controller of the continuous production facility to all axes required for decoupling.

These decoupled precontrols impede a temporary and stationary change of all web tensions in the machine except the ones to be controlled locally.

Target system	Libraries
MLC 14VRS TEC function package	ML_TechTensionAdvanced

Tab. 7-18: Reference table of the MB_TensionDecouplingNetworkType01



The use of the function block is subject to license (for details, refer to [Introduction and overview](#), page 425).

Interface description

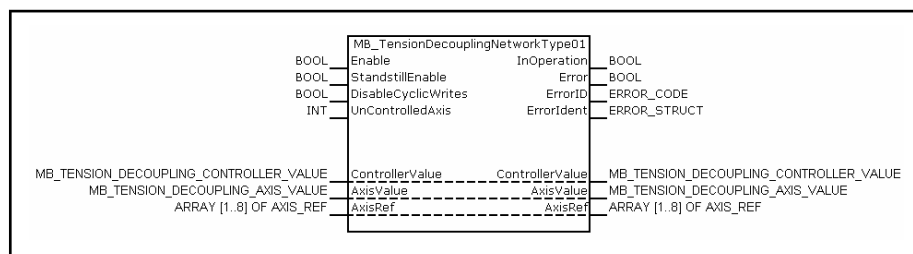


Fig. 7-31: MB_TensionDecouplingNetworkType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enabling the function block (once, edge-controlled)
	StandstillEnable	BOOL	Closed-loop standstill control enabled if the input is set
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output
	UnControlledAxis	INT	Axis in the system not used for the closed-loop control and decoupling of the web tensions
VAR_OUTPUT	InOperation	BOOL	The decoupling network operates and the outputs are valid
	Error	BOOL	Processing completed with error

I/O type	Name	Data type	Description
	ErrorID	ERROR_CODE	Diagnostic description in the event of an error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling, page 471
VAR_IN_OUT	ControllerValue	MB_TENSION_DECOUPLING_CONTROLLER_VALUE	Structure of the controller outputs (CorrectionValue, AdjCorrectionValue, AdditivePosInc and AdditivePosDeg) MB_TENSION_DECOUPLING_CONTROLLER_VALUE, page 471
VAR_IN_OUT	AxisValue	MB_TENSION_DECOUPLING_AXIS_VALUE	Structure of the axis inputs (CorrectionValue, AdditivePosInc and AdditivePosDeg) MB_TENSION_DECOUPLING_AXIS_VALUE, page 472
VAR_IN_OUT CONSTANT	AxisRef	ARRAY [1..8] OF AXIS_REF	Control axes of the system (without "UnControlledAxis")

Tab. 7-19: Interface description of the MB_TensionDecouplingNetworkType01

Min./max. and default values

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
ControllerValue	MB_TENSION_DECOUPLING_CONTROLLER_VALUE			0	Continuous
AxisValue	MB_TENSION_DECOUPLING_AXIS_VALUE			0	Continuous
AxisRef	ARRAY [1..8] OF AXIS_REF			0	Rising edge at "Enable"
Enable	BOOL			FALSE	Continuous
StandstillEnable	BOOL			FALSE	Continuous
DisableCyclicWrites	BOOL			FALSE	Rising edge at "Enable"
UnControlledAxis	INT	1	9	1	Rising edge at "Enable"

Tab. 7-20: Input behavior of the MB_TensionDecouplingNetworkType01

Functional description

The following explains the individual functions of the function block inputs in detail. In general, all outputs are continuously evaluated if the "Enable" input is set. That means that input changes are immediately applied without toggling that input again. An exception of this rule are the inputs "UnControlledAxis", "DisableCyclicWrites" and the input/output array "AxisRef" which can only be evaluated at a rising edge at the "Enable" input. "0" or FALSE are assigned internally to all input values whose inputs are not connected. An exception of this rule is the "UnControlledAxis" input. If it is not assigned, it is set to "1". As for the web tension controller "MB_Tension_ControlLoadCellType02", the control variable is written to the gear ratio fine adjustment in normal mode. Depending on the axis type, it is written directly on the respective parameter (P-0-0694 at interpolation in the drive or A-0-2605 at interpolation on the control), if "DisableCyclicWrites" is disabled. Additionally, the con-

control signals of all axes are always provided via the "MB_TENSION_DECOUPLING_AXIS_VALUE" structure at the output of the decoupling network. If there are uniform axis types, it is sufficient to add only the respective standstill signal ("AdditivePosInc" at interpolation in the drive or "AdditivePosDeg" at interpolation in the control) from the controllers to the decoupling network. If there are mixed axis types (interpolation in the drive and interpolation in the control), it is absolutely necessary to add both standstill signals (AdditivePosInc and AdditivePosDeg) to the decoupling network for all controllers. Since the outputs "CorrectionValue", "AdjCorrectionValue", "AdditivePosInc" and "AdditivePosDeg" of the path tension controller may not be written directly on the axes anymore when using the decoupling network, it is absolutely necessary to disable the cyclic writing of values in all participating web tension controllers (DisableCyclicWrites:=TRUE). The complete mode of operation of the decoupling network is shown in the following diagrams.

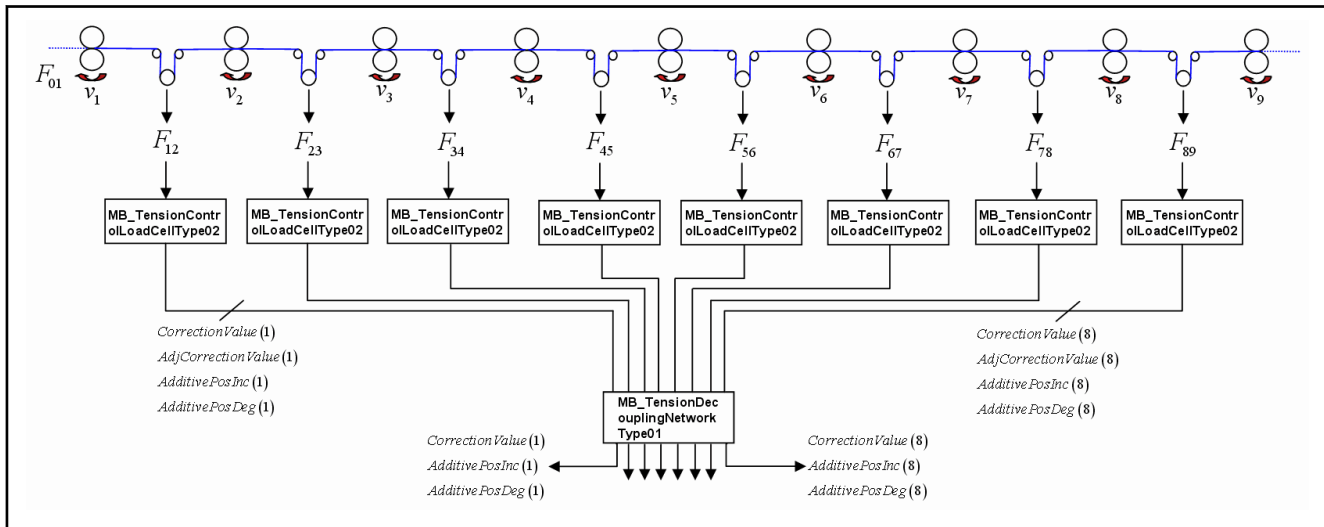


Fig. 7-32: General diagram of the MB_TensionDecouplingNetworkType01 function block

It can be seen that each web section to be controlled has to be equipped with a web tension controller MB_TensionControlLoadCellType02. It is to be considered that all web tension controllers have to operate in positive polarity (NegPolarity:=FALSE). The controller outputs "CorrectionValue", "AdjCorrectionValue", "AdditivePosInc" and "AdditivePosDeg" are then added to the decoupling network MB_TensionDecouplingNetworkType01. In the example shown, the web tension control should be controlled independently from each other in eight web sections. At least eight control elements are required. Since n web sections to be controlled are always limited by $n+1$ axes, any axis can be selected that is exclusively operated with master axis velocity and does not have to execute additional control movements ("CorrectionValue", "AdditivePosInc" or "AdditivePosDeg"). The following figure shows the first axis of the system to be excluded for the decoupled closed-loop control for the web tensions (UnControlledAxis:=1). This decoupling variant also corresponds to the basic function of the MB_TensionControlLoadCellType02 if only the first web tension controller is active.

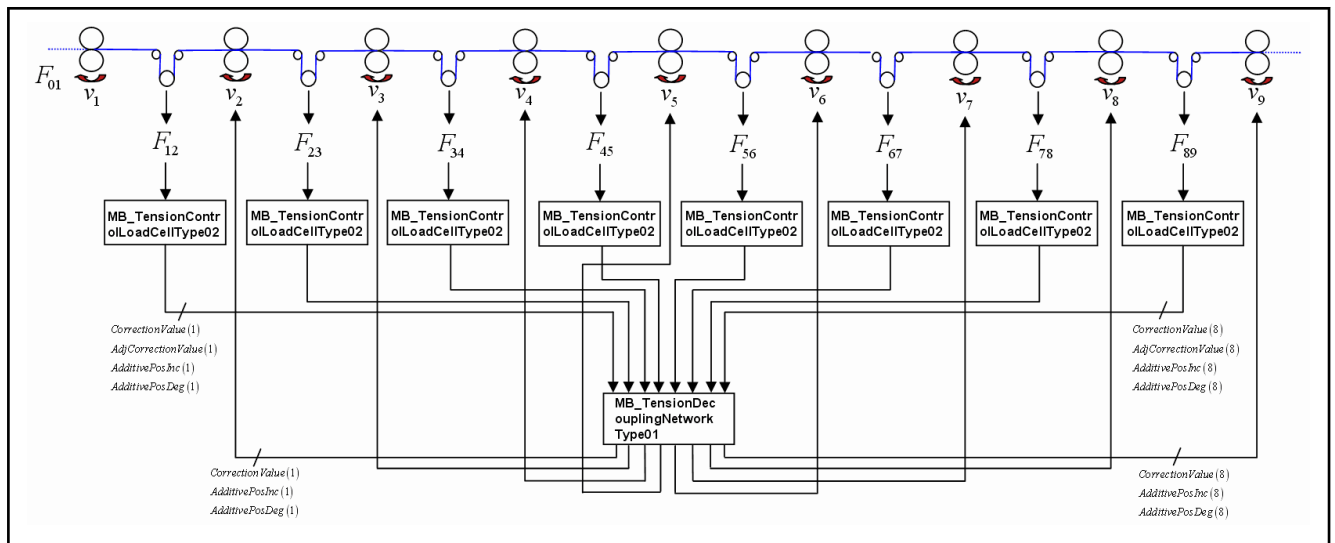


Fig. 7-33: Decoupling web tensions by precontrol exclusively in web direction (leading nips are not required for the closed-loop control and decoupling)

If this variant is selected for a system with less than eight web sections to be controlled, the following web tension controllers in web direction remain unconsidered.

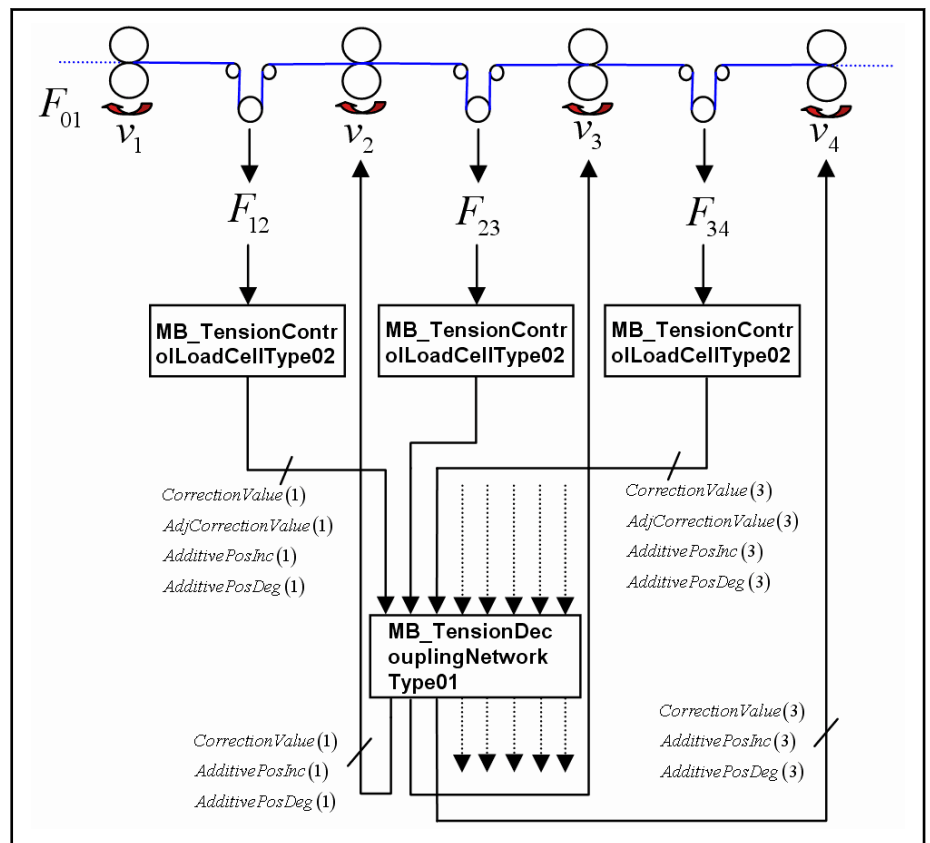


Fig. 7-34: Decoupling web tensions by precontrol exclusively in web direction (representation for three web sections to be controlled)

Another option is shown in the following. The last axis of the system is excluded for the closed-loop control of the web tensions (UnControlledAxis:=9).

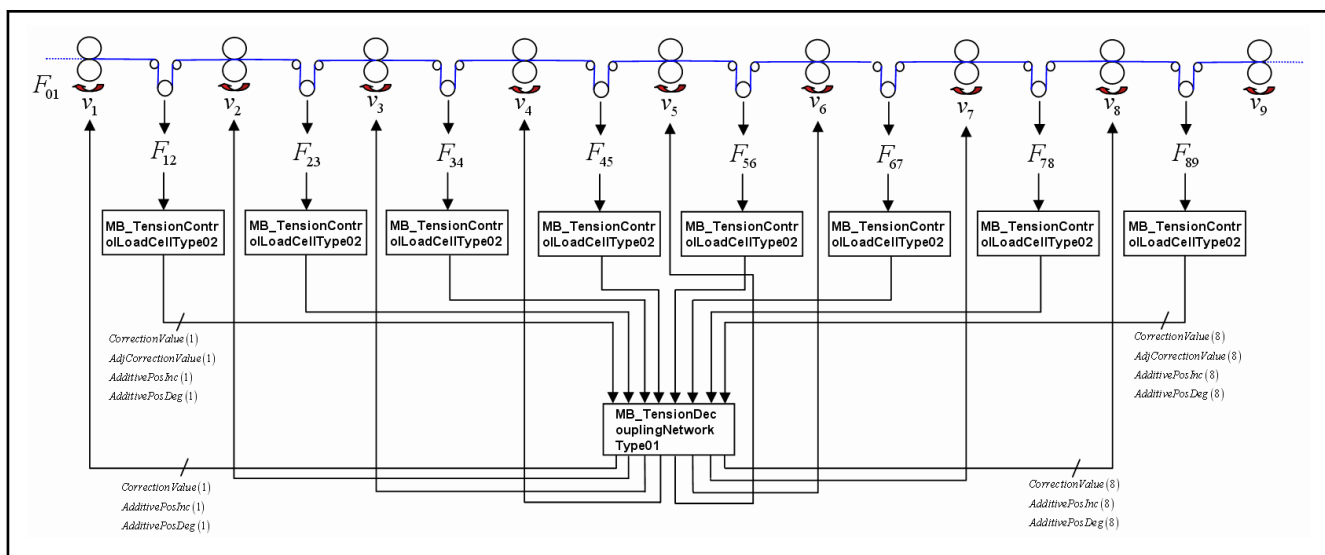


Fig. 7-35: Decoupling of web tensions by precontrol exclusively opposite web direction (following nips are not required for the closed-loop control and decoupling)

If this variant is selected for a system with less than eight web sections to be controlled, the leading web tension controllers opposite web direction remain unconsidered.

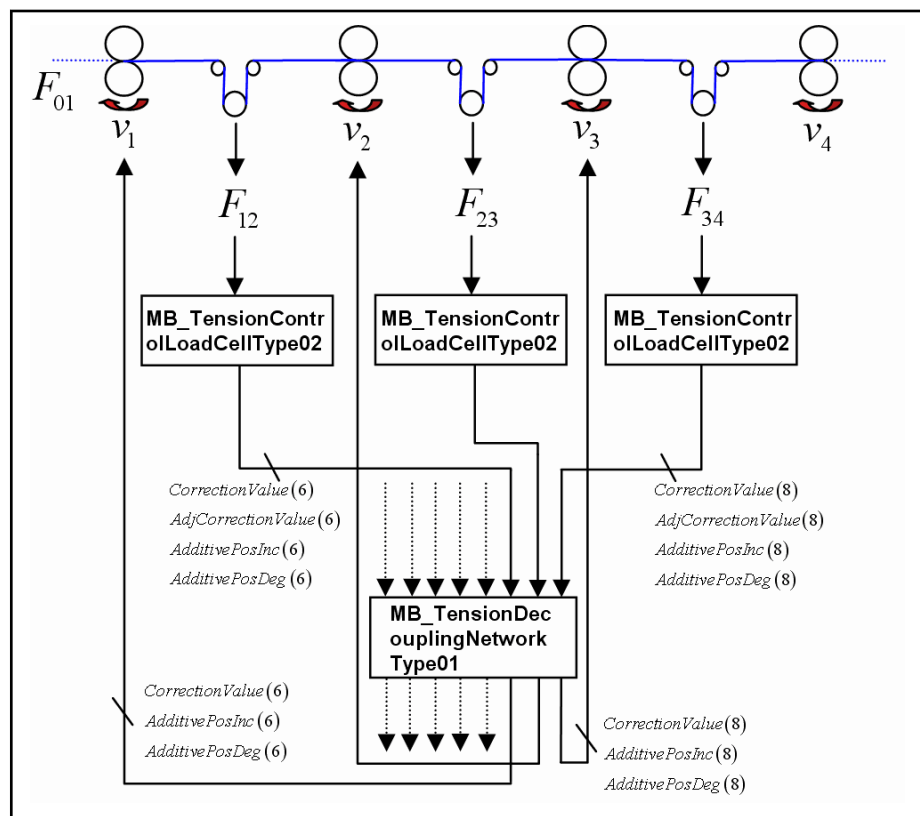


Fig. 7-36: Decoupling web tensions by precontrol exclusively opposite web direction (representation for three web sections to be controlled)

Since any axis of the system can be excluded for controlling the web tensions, another option is to exclude the centered axis. This corresponds to the behavior "UnControlledAxis":=2-8 and is exemplarily shown for the fifth axis (UnControlledAxis:=5).

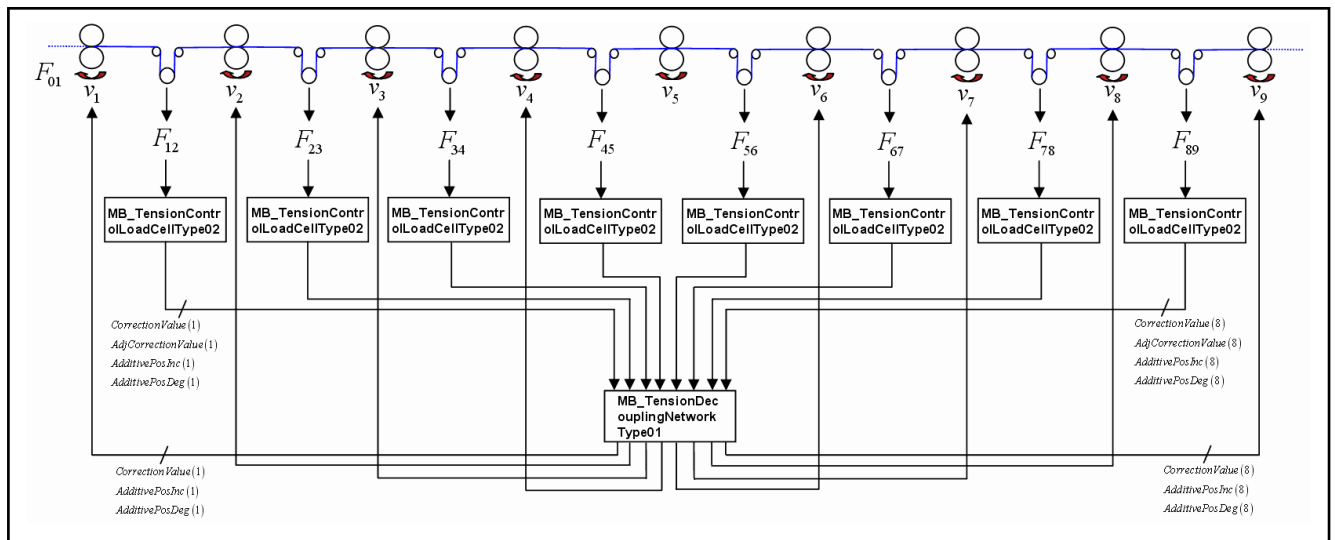


Fig. 7-37: Decoupling of web tensions by precontrol in and opposite web direction (centered nips in figure "v5" are not required for the closed-loop control and decoupling)

If this variant is selected for a system with less than eight web sections to be controlled, the leading and/or following web tension controllers in web direction remain unconsidered as in the two previously presented variants. If the axis not to be controlled is selected according to the real system in web direction, do not consider the following real non-existing web tension controllers.

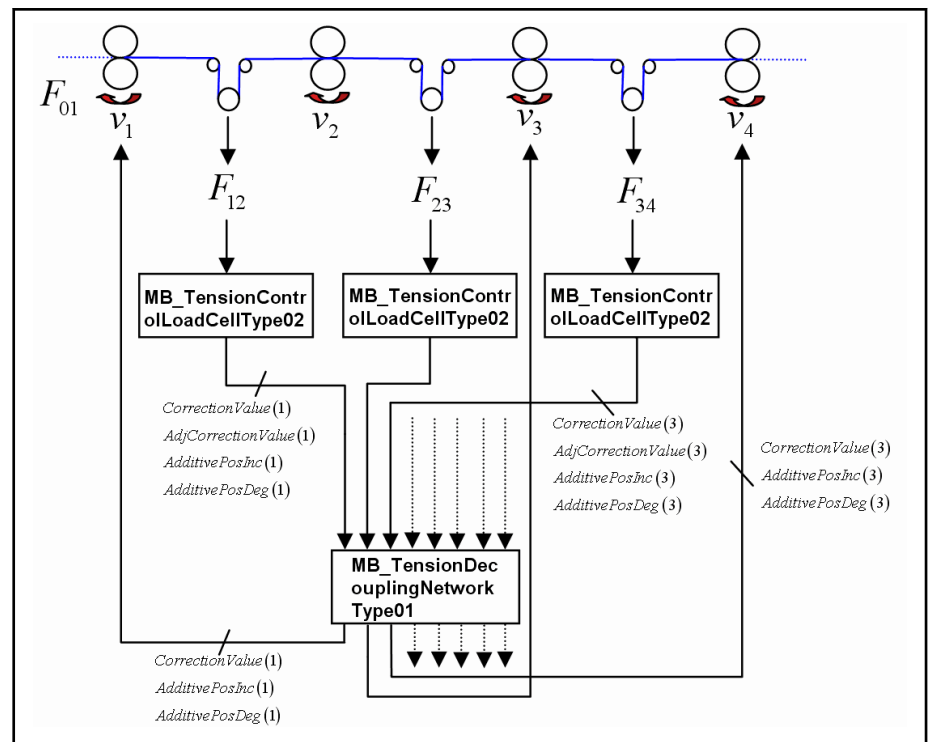


Fig. 7-38: Decoupling web tensions by precontrol in and opposite web direction (representation for three web sections to be controlled)

Apart from these three variants of decoupling at which always one axis is not required for closed-loop control and decoupling, the decoupling network can nearly be used as desired. Exemplarily, a system with up to 15 web sections to be controlled and the request for a minimum control effort is shown.

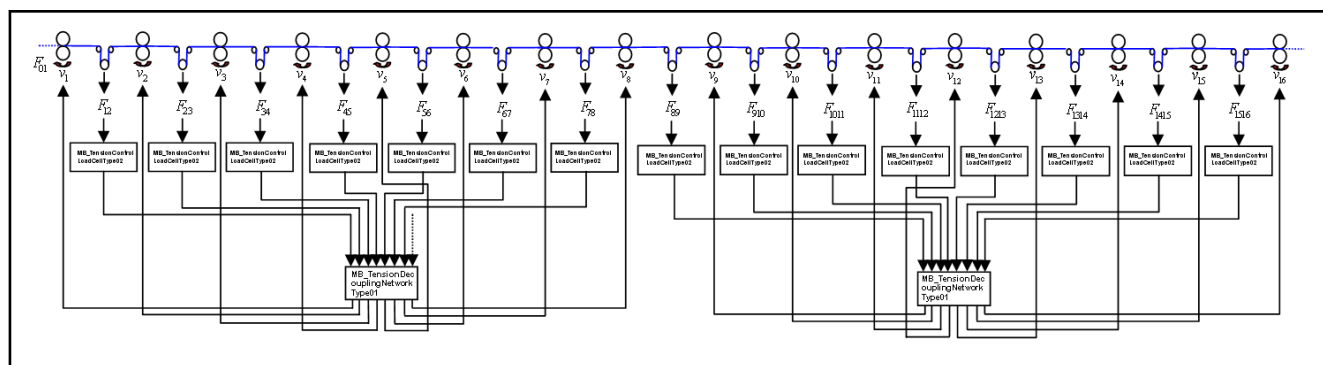


Fig. 7-39: Expansion of the decoupling network to 15 web sections to be controlled at an additional request to a minimum control effort

This figure shows that the two function blocks MB_TensionDecouplingNetworkType01 are used and that a control signal change of a web tension controller of a maximum of eight (controlled sections/2) causes control movements. Furthermore, it can be seen that an upstream decoupling is used in the leading decoupling network (UnControlledAxis:=9) and a downstream decoupling (UnControlledAxis:=1) in the following decoupling network. Only the last web tension controller of the leading decoupling network was not assigned. Since the last web section of the leading decoupling network is completely decoupled from all web sections in the front, this combination of the two decoupling networks can be used without cross communication or cross precontrol. If two or more decoupling networks are used, these can be combined in any number of variants. If another variant with multiple decoupling networks is reasonable for a special system, the inputs and outputs of the decoupling networks can be combined externally. Depending on the desired behavior, it might be additionally necessary to disable cyclic writing (DisableCyclicWrites:=TRUE). An example of these unlimited number of variants are systems with more than eight web sections and the requirements that the first, last or any axis may not be controlled and/or precontrolled. It is then reasonable to combine one or several outputs of a decoupling network with one or several inputs of another decoupling network. Detailed and advanced knowledge is required for these implementations.

Commissioning

If "DisableCyclicWrites" is inactive, P-0-0694 (Gear ratio fine adjustment) and P-0-0692 (Additive master axis position) parameters have to be applied in the cyclic command value channel of the slave axis. At a rising edge at "Enable", the cyclic data channels are checked. If one of the two parameters is not found, an error message is returned at the error outputs of the function block. If "DisableCyclicWrites" is enabled, the configuration of the cyclic command value data channels is not checked.

Enable

The "Enable" input activates the function block.



When the function block is activated, all controller control signals and axis positions are evaluated in order to prevent jumps in position on the axes.

As a consequence, this Enable input should not be activated until all web tension controllers have reached an active "InOperation" output.

StandstillEnable

The closed-loop tension control is enabled by setting the "StandstillEnable" input. Thus, the inputs "AdditivePosInc" and "AdditivePosDeg" are evaluated and used to calculate and output the complete control variables "AdditivePosInc" and "AdditivePosDeg" of the individual axes.

DisableCyclicWrites The "DisableCyclicWrites" input determines whether the values calculated by the function block are to be written directly via the cyclic channel (FALSE) or whether they are only available at the function block output (TRUE). With "DisableCyclicWrites":=TRUE, it is possible to combine multiple decoupling networks with each other by manually linking the outputs of multiple decoupling networks.

UnControlledAxis The "UnControlledAxis" input determines the decoupling variant as well as the position of the axis not to be controlled and precontrolled. By default, this input is set to 1. That means for the decoupling network that a precontrol should exclusively be executed in web direction (downstream decoupling). If this input is set to 9, the last axis of the system for the closed-loop control and decoupling is excluded (upstream decoupling). In this case, a precontrol is only executed in opposite web direction. Furthermore, any axis of the system can be excluded for the web tension control and decoupling (UnControlledAxis:=2-8).

Error handling The function block generates the following error messages in Additional1/Additional2 for the table **F_RELATED_TABLE**, 16#0170:

ErrorID	Additional1	Additional2	Description
ACCESS_ERROR, 16#0004	16#00000A00	16#00000001	The parameter P-0-0694 is not parameterized in the optional cyclic telegram of the precontrolled axis
ACCESS_ERROR, 16#0004	16#00000A00	16#00000002	The parameter P-0-0692 is not parameterized in the optional cyclic telegram of the precontrolled axis
INPUT_RANGE_ERROR, 16#0006	16#00000A01	16#00000007	The "UnControlledAxis" input is not in the valid range
DEVICE_ERROR, 16#0008	16#00000A02	16#00000007	No power is added to a drive. A closed-loop tension control is therefore not possible
DEVICE_ERROR, 16#0008	16#00000A02	16#00000008	A drive is not in a synchronous operation mode

Tab. 7-21: Error codes of the MB_TensionDecouplingNetworkType01

7.9.2 MB_TENSION_DECOUPLING_CONTROLLER_VALUE

The structure MB_TENSION_DECOUPLING_CONTROLLER_VALUE contains the controller outputs (CorrectionValue, AdjCorrectionValue, AdditivePosInc, and AdditivePosDeg) for the decoupling network.

This structure is used by the "MB_TensionDecouplingNetworkType01" function block.

Name	Type	Min. value	Max. value	Description
ControllerCorrectionValue [1]	REAL			CorrectionValue of the first web tension controller
ControllerCorrectionValue [2]	REAL			CorrectionValue of the second web tension controller
ControllerCorrectionValue [3]	REAL			CorrectionValue of the third web tension controller
ControllerCorrectionValue [4]	REAL			CorrectionValue of the fourth web tension controller
ControllerCorrectionValue [5]	REAL			CorrectionValue of the fifth web tension controller
ControllerCorrectionValue [6]	REAL			CorrectionValue of the sixth web tension controller
ControllerCorrectionValue [7]	REAL			CorrectionValue of the seventh web tension controller

Name	Type	Min. value	Max. value	Description
ControllerCorrectionValue [8]	REAL			CorrectionValue of the eighths web tension controller
ControllerAdjCorrectionValue [1]	REAL			AdjCorrectionValue of the first web tension controller
ControllerAdjCorrectionValue [2]	REAL			AdjCorrectionValue of the second web tension controller
ControllerAdjCorrectionValue [3]	REAL			AdjCorrectionValue of the third web tension controller
ControllerAdjCorrectionValue [4]	REAL			AdjCorrectionValue of the fourth web tension controller
ControllerAdjCorrectionValue [5]	REAL			AdjCorrectionValue of the fifth web tension controller
ControllerAdjCorrectionValue [6]	REAL			AdjCorrectionValue of the sixth web tension controller
ControllerAdjCorrectionValue [7]	REAL			AdjCorrectionValue of the seventh web tension controller
ControllerAdjCorrectionValue [8]	REAL			AdjCorrectionValue of the eighth web tension controller
ControllerAdditivePosInc [1]	DINT			AdditivePosInc of the first web tension controller
ControllerAdditivePosInc [2]	DINT			AdditivePosInc of the second web tension controller
ControllerAdditivePosInc [3]	DINT			AdditivePosInc of the third web tension controller
ControllerAdditivePosInc [4]	DINT			AdditivePosInc of the fourth web tension controller
ControllerAdditivePosInc [5]	DINT			AdditivePosInc of the fifth web tension controller
ControllerAdditivePosInc [6]	DINT			AdditivePosInc of the sixth web tension controller
ControllerAdditivePosInc [7]	DINT			AdditivePosInc of the seventh web tension controller
ControllerAdditivePosInc [8]	DINT			AdditivePosInc of the eighth web tension controller
ControllerAdditivePosDeg [1]	REAL			AdditivePosDeg of the first web tension controller
ControllerAdditivePosDeg [2]	REAL			AdditivePosDeg of the second web tension controller
ControllerAdditivePosDeg [3]	REAL			AdditivePosDeg of the third web tension controller
ControllerAdditivePosDeg [4]	REAL			AdditivePosDeg of the fourth web tension controller
ControllerAdditivePosDeg [5]	REAL			AdditivePosDeg of the fifth web tension controller
ControllerAdditivePosDeg [6]	REAL			AdditivePosDeg of the sixth web tension controller
ControllerAdditivePosDeg [7]	REAL			AdditivePosDeg of the seventh web tension controller
ControllerAdditivePosDeg [8]	REAL			AdditivePosDeg of the eighth web tension controller

Tab. 7-22: Structural description of the MB_TENSION_DECOUPLING_CONTROLLER_VALUE

7.9.3 MB_TENSION_DECOUPLING_AXIS_VALUE

The structure MB_TENSION_DECOUPLING_AXIS_VALUE contains the values for the axis inputs (CorrectionValue, AdditivePosInc, and AdditivePosDeg).

This structure is used by the "MB_TensionDecouplingNetworkType01" function block.

Name	Type	Min. value	Max. value	Description
AxisCorrectionValue [1]	REAL			CorrectionValue of the first axis
AxisCorrectionValue [2]	REAL			CorrectionValue of the second axis

Name	Type	Min. value	Max. value	Description
AxisCorrectionValue [3]	REAL			CorrectionValue of the third axis
AxisCorrectionValue [4]	REAL			CorrectionValue of the fourth axis
AxisCorrectionValue [5]	REAL			CorrectionValue of the fifth axis
AxisCorrectionValue [6]	REAL			CorrectionValue of the sixth axis
AxisCorrectionValue [7]	REAL			CorrectionValue of the seventh axis
AxisCorrectionValue [8]	REAL			CorrectionValue of the eighth axis
AxisAdditivePosInc [1]	DINT			AdditivePosInc of the first axis
AxisAdditivePosInc [2]	DINT			AdditivePosInc of the second axis
AxisAdditivePosInc [3]	DINT			AdditivePosInc of the third axis
AxisAdditivePosInc [4]	DINT			AdditivePosInc of the fourth axis
AxisAdditivePosInc [5]	DINT			AdditivePosInc of the fifth axis
AxisAdditivePosInc [6]	DINT			AdditivePosInc of the sixth axis
AxisAdditivePosInc [7]	DINT			AdditivePosInc of the seventh axis
AxisAdditivePosInc [8]	DINT			AdditivePosInc of the eighth axis
AxisAdditivePosDeg [1]	REAL			AdditivePosDeg of the first axis
AxisAdditivePosDeg [2]	REAL			AdditivePosDeg of the second axis
AxisAdditivePosDeg [3]	REAL			AdditivePosDeg of the third axis
AxisAdditivePosDeg [4]	REAL			AdditivePosDeg of the fourth axis
AxisAdditivePosDeg [5]	REAL			AdditivePosDeg of the fifth axis
AxisAdditivePosDeg [6]	REAL			AdditivePosDeg of the sixth axis
AxisAdditivePosDeg [7]	REAL			AdditivePosDeg of the seventh axis
AxisAdditivePosDeg [8]	REAL			AdditivePosDeg of the eighth axis

Tab. 7-23: Structural description of the MB_TENSION_DECOUPLING_AXIS_VALUE

8 RMB_TechWinder.library // MX_TechWinder

8.1 Introduction and overview

Different function blocks are combined in the libraries RMB_TechWinder (MLC) and MX_TechWinder.lib (MLD) summarizing different functionalities that provide functionalities for winding and unwinding fabric webs (winders).



The library RMB_TechWinder (MLC) requires the **TEC (technology) function package** of the control.

Library dependencies

MLC control - RMB_TechWinder

The required libraries are automatically included.

MLD control - MX_TechWinder

IndraLogic 1.x (drive firmware < MPx18):

The following libraries are required:

- MX_Technology0x (MPX06 drive firmware)
- RIL_LoopControl

IndraLogic 2G (drive firmware from MPX20):

The required libraries are automatically included.

Function packages for MLD

For the MLD, the drive function package MA "MLD Advanced" is required for the function blocks of the "MX_TechWinder library" .

This function package is to be ordered separately with the respective drive firmware.



For the MLC, the MX_TechWinder library differs from the RMB_TechWinder library with regard to the functional scope. The function blocks available for each target can be found out using the target system tables for the respective function blocks.

These function blocks are used to implement the controller functions of a

- Dancer position controller
- Diameter calculator with dancer
- Diameter calculator with closed-loop tension control
- Diameter calculator with open-loop tension control

The function blocks required to implement these functionalities are described in the following.

The following table provides an overview on all the documented function blocks and their respective application.

Already connected complete winder function blocks

Function block	Description
MB_WinderDancerCtrlType01, page 509	Function block for winders with dancer contains among others: MB_DancerControlType03, page 479 MB_DiameterCalculatorType03, page 494 MB_DiameterMeasurementType01, page 503 (The control variable of the dancer position controller is standardized with the format length of the reference axis and the current winding diameter.)
MB_WinderDancerCtrlType02, page 516	Extended function block for winders with dancer contains among others: MB_DancerControlType04, page 485 MB_DiameterCalculatorType03, page 494 MB_DiameterMeasurementType01, page 503 1. Standardizing the dancer control variable irrespective of the format length of the reference axis 2. Additional diagnostic outputs for the dancer control
MB_WinderTensionCtrlType02, page 526	The function block for winders with closed-loop or open-loop tension control (without sensor) contains among others: MB_DiameterCalculatorType03, page 494 MB_DiameterMeasurementType01, page 503
MB_WinderTensionCtrlSpeedType01, page 544	The function block for winders with closed-loop tension control and speed correction contains among others: MB_DiameterCalculatorType03, page 494 MB_DiameterMeasurementType01, page 503

Tab. 8-1: Already connected winder function blocks



The preconnected function blocks combine all functions of the individual function blocks and are responsible for the correct connection of the individual function blocks.

Detailed information for commissioning are contained in "MLC 14VRS Winder Function Applications DOK-MLC***-TF*WIND*V14-APRS-DE-P, R911341513" in the media directory.

Function blocks for individual functionalities

Function block	Description
MB_DancerControlType03, page 479	Function block to calculate a closed-loop dancer position control for a tension-controlled axis with dancer or winding axis (The control variable of the dancer position controller is standardized with the format length of the reference axis and the reference diameter.)
MB_DancerControlType04, page 485	Extended function block to calculate a closed-loop dancer position control for a tension-controlled axis with dancer or winding axis 1. Standardizing the dancer control variable irrespective of the format length of the reference axis 2. Additional diagnostic outputs for the dancer control
MB_DiameterCalculatorType03, page 494	Function block to calculate the current diameter for winding axes with velocity adaptation
MB_DiameterMeasurementType01, page 503	Function block to read a measured diameter for winding axes with velocity adaptation

Tab. 8-2: Single function blocks

Function blocks for additional winding functions

Function block	Description
MB_CalcInertiaLimitType02, page 559	Function block to calculate the maximum moment of inertia of the winding material on a winder
MB_UnwindMaterialType02, page 562	Function block to calculate the remaining length, the remaining time, etc. of the fabric web for an unwinder
MB_WindTaperProfileType01, page 567	Function block to calculate a winding taper profile for a winder with closed-loop or closed-loop tension control
MB_WindSpeedControlAdaptionType01, page 570	Function block for an automatic adaptation of the K_p gain of the speed controller
MB_TraverseControlType01, page 574	The function blocks controls a translatory traversing axis allowing to traverse different materials (e.g. wire) together with a winder (rotary axis).
MB_DancerCalcControlParamType01, page 590	The function block uses the physical data of the dancer system (diameter of the draw roller or the core diameter of the winder as well as the delay time of the system) to compute the controller parameters of the dancer position controller

Tab. 8-3: Function blocks for additional functionalities

8.2 Closed-loop dancer position control

8.2.1 Introduction and overview

The dancer position controller is used for example on tension-controlled axes or winders with dancers. The dancer position controller controls the position of a dancer. The tension is specified by a force which acts upon the movable part of the dancer. Apply this force by a weight, a spring or a pneumatic element. The actual value of the dancer position is recorded, for example, via an

analog channel, applied to the control and processed in the process controller.

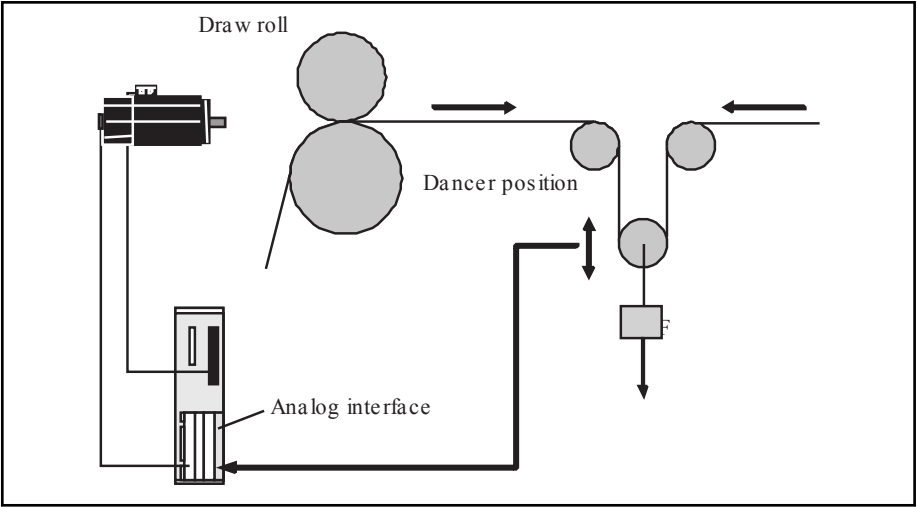


Fig. 8-1: Principle of a closed-loop dancer position control

The dancer position controller is implemented in the function block [MB_DancerControlType03](#), [page 479](#), and designed as PID controller. By setting the controller parameters, the individual controller functions (P value: Proportional value, I-value: Integral value, D-value: differential value) can be controlled. If a parameter is set to zero, the respective controller value is also not effective.

The following table shows the possible and permitted controller configurations.

No.	P-value	I-value	D-value	Permitted	Note
1	0	0	0	Yes	Controller ineffective, output = 0.0
2	0	0	1	No	D-controller not defined, error
3	0	1	0	Yes	Exclusive I-controller
4	0	1	1	No	DI-controller not defined, error
5	1	0	0	Yes	Exclusive P-controller
6	1	0	1	Yes	PD-controller
7	1	1	0	Yes	PI-controller
8	1	1	1	Yes	PID-controller

Tab. 8-4: Permitted and prohibited controller configurations (1 corresponds to the value set (value ≥ 0.0), 0 corresponds to the value not set (value = 0.0))

Functional diagram of the dancer position controller

The following functional diagram schematically illustrates the mode of operation of the dancer position controller.

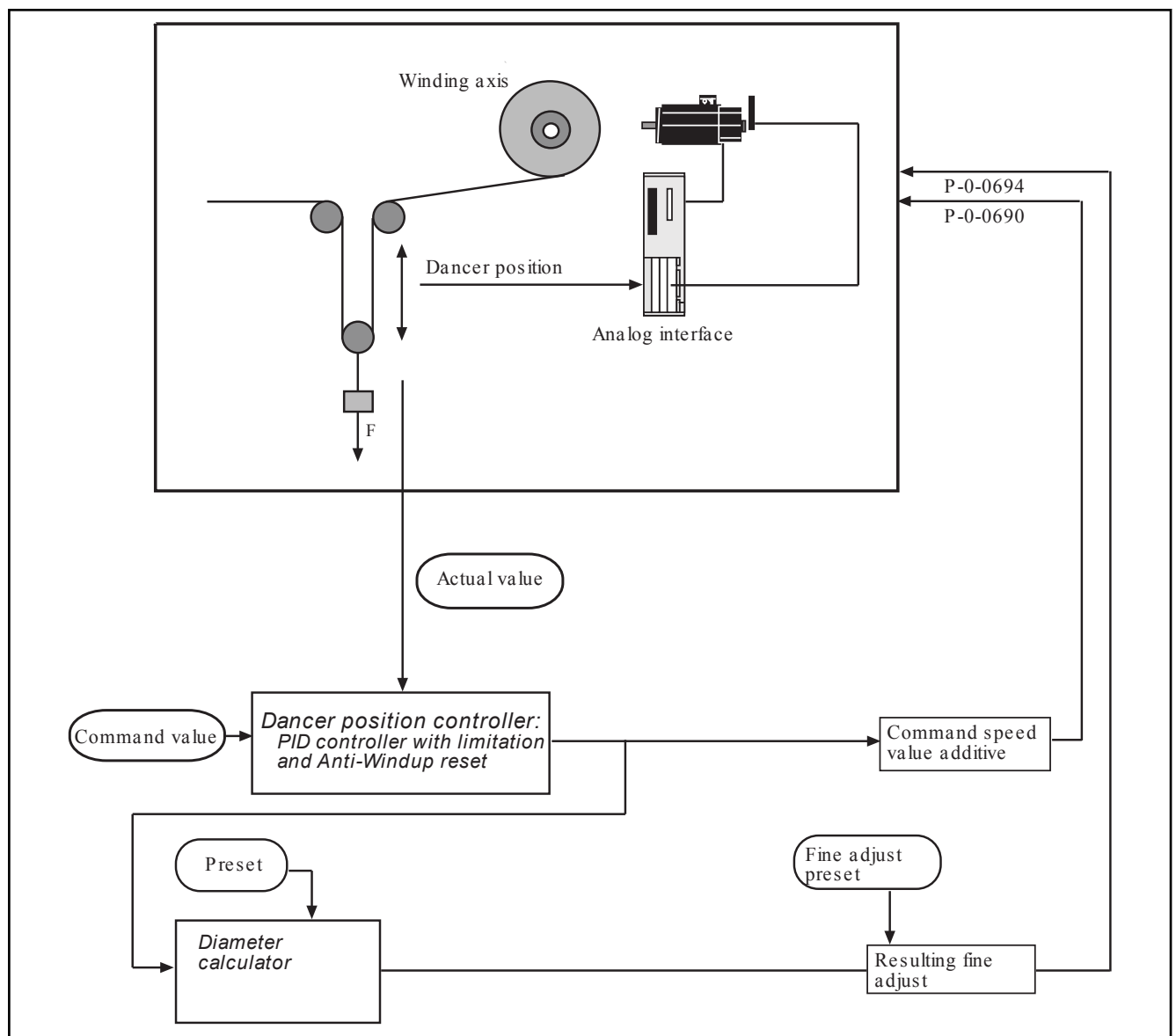


Fig. 8-2: Functional diagram of the dancer position controller

8.2.2 MB_DancerControlType03

Brief description The function block cyclically calculates a closed-loop dancer position control for a draw roller with dancer or for a winding axis with dancer. The cyclic output variable of the dancer position controller is the speed offset of the respective drive of a draw roller or winding axis. The current control deviation of the dancer position controller is output.

The "AddVelocityDancer" output of the function block affects either the parameter "P-0-0690 Additive velocity command value, process loop" (interpolation in the drive) or the parameter "A-0-2615 ELS, additive velocity command value, process controller" (interpolation in the control). If this is not desired, it can be prevented using the input "DisableCyclicWrites". It is only written to the "AddVelocityDancer" output.

Assignment: Target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder
MLD with MPx07/08/17VRS with MA function package	MX_TechWinder.lib

Tab. 8-5: Reference table of the MB_DancerControlType03

Interface description

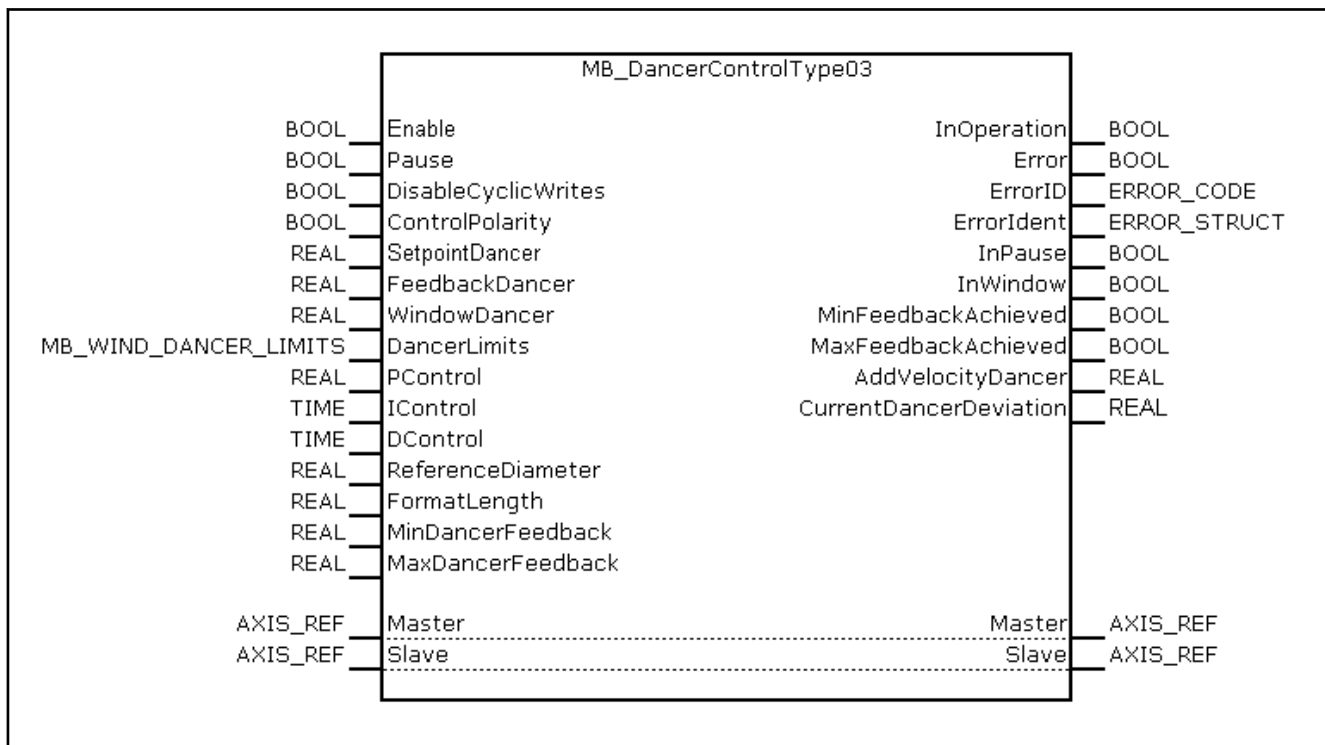


Fig. 8-3: Function block MB_DancerControlType03

I/O type	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Function block processing enabled (level-controlled)
	Pause	BOOL	Dancer controller paused, no velocity adaptation (level-controlled)
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output
	ControlPolarity	BOOL	Polarity of the closed-loop dancer control is inverted (applied at a positive at Enable)
	SetpointDancer	REAL	Command value (setpoint) for the closed-loop dancer control
	FeedbackDancer	REAL	Actual value (feedback) for the dancer position

I/O type	Name	Type	Comment
	WindowDancer	REAL	Windows to monitor the actual dancer position value [%] with regard to the command dancer position value. If the dancer position deviates from the command value by more than the given value, the function block reports this deviation by resetting the "InWindow" output
	DancerLimits	MB_WIND_DANCER_LIMITS	Structure for the automatic control variable limitation of the dancer MB_WIND_DANCER_LIMITS, page 597
	PControl	REAL	Value for the Kp-gain of the closed-loop dancer control
	IControl	TIME	Value for the integral action time of the closed-loop dancer control
	DControl	TIME	Value for the derivative time of the closed-loop dancer control
	ReferenceDiameter	REAL	Reference diameter (value to standardize PControl and the control variable limitation "DancerLimits". That means that the values are internally adapted) For an exclusive dancer position controller: Draw roller diameter [mm] For winders with dancer: Assign current winding diameter [mm] from the diameter calculator function block
	FormatLength	REAL	Format length [mm], (value to standardize PControl) (1 revolution of the reference axis unwound)
	MinDancerFeedback	REAL	Lower threshold for the actual dancer value (only for a display with binary output)
	MaxDancerFeedback	REAL	Upper threshold for the actual dancer value (only for a display with binary output)
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling, page 484
	InPause	BOOL	Dancer controller pausing
	InWindow	BOOL	Actual dancer value within dancer window
	MinFeedbackAchieved	BOOL	Actual value of the dancer position is below the input "MinDancerFeedback"
	MaxFeedbackAchieved	BOOL	Actual value of the dancer position is above the input "MaxDancerFeedback"
	AddVelocityDancer	REAL	Speed offset for the winding drive or draw roller drive [rpm]

I/O type	Name	Type	Comment
	CurrentDancerDeviation	REAL	Current control deviation of the dancer controller
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to the slave axis (tension-controlled axis)

Tab. 8-6: Interface variables of the MB_DancerControlType03 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
DisableCyclicWrites	BOOL			FALSE	Rising edge at "Enable"
ControlPolarity	BOOL			FALSE	Rising edge at "Enable"
SetpointDancer	REAL	0.0	n. def.	0.0	Continuous
FeedbackDancer	REAL	N. def.	N. def.	0.0	Continuous
WindowDancer	REAL	0.0	n. def.	0.0	Continuous
DancerLimits	STRUCT				
PControl	REAL	0.0	n. def.	0.0	Continuous
IControl	TIME	0 s	N. def.	0 s	Continuous
DControl	TIME	0 s	N. def.	0 s	Continuous
ReferenceDiameter	REAL	0.0	N. def.	100	Continuous
FormatLength	REAL	0.0	n. def.	0.0	Rising edge at "Enable"
MinDancerFeedback	REAL	0.0	MaxDancerFeedback	0.0	Continuous
MaxDancerFeedback	REAL	MinDancerFeedback	N. def.	0.0	Continuous

Tab. 8-7: Min./max. and default values of the MB_DancerControlType03 inputs

Control variable limitation

The control variable limitation of the dancer position controller is dynamically adjusted according to the reference axis velocity and the diameter ratio ($\text{format length} / \text{Pi} * \text{ReferenceDiameter}$). This means that, at a higher velocity, the limitation of the additive velocity of the dancer position controller is automatically higher than at a lower velocity. Consideration of the diameter ratio results in the limitation for winding applications automatically being adapted to a changing "ReferenceDiameter" (the limitation reduces automatically with an increasing diameter).

Signal-time diagram

The following signal-time diagram represents selected function block signals.

Relation of the signals

Signal input	Signal output	Description
Enable	InOperation	Commissioning the function block
	Error	Operational readiness was interrupted
Pause	InPause	Pausing the internal controller process

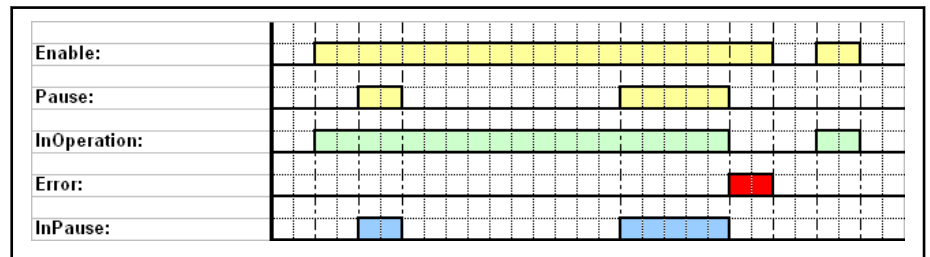


Fig. 8-4: Signal-time diagram of selected inputs and outputs of the function block MB_DancerControlType03

Functional Description

The function block cyclically calculates a closed-loop dancer position control for a draw roller with dancer or for a winding axis with dancer. The IL_PID-Type01 function block from the RIL_LoopControl library is used as controller. The controller is a PID-controller. The cyclic output variables of the dancer position controller are the speed offset of the respective drive for the draw roller or the winding axis, as well as the current control deviation of the dancer position controller. The "AddVelocityDancer" output of the function block should act on the parameter "P-0-0690 Additive velocity command value, process loop" (external connection) for a closed-loop dancer position control. The current control deviation "CurrentDancerDeviation" is available for analysis. The proportional gain (factor K_p) and the low and high values for control variable limitation "DancerLimits" are automatically adjusted with the "ReferenceDiameter" input based on the format length "FormatLength". The diameter of the draw roller "ReferenceDiameter" has to be specified for closed-loop dancer position control for a draw roller. For a winder with dancer position control, "ReferenceDiameter" is the current winding diameter (the proportional gain of the dancer position control automatically reduces with an increasing diameter). If the values for the integral part (integral action time T_n , "IControl") and the lead element (lead time T_D , "DControl") are zero, the internal memory parameters of the PID controller are deleted. There has to be a value proportional to the dancer roll adjustment (e.g. an analog tension value) at the "FeedBackDancer" input. The command value and the actual value must have the same unit.



The function block only processes positive command values and actual values.

Negative command values result in an error.

Negative actual values are limited to zero.

Closed-loop dancer position control

To determine the dancer position, a measuring device is required that converts the dancer position into a proportional signal. The actual position of the dancer is subtracted from a given command value and specified as a control deviation on an internally integrated PID-controller. The control variable of the controller has an effect on the speed command value by addition. Thus, the

winding axis rotates faster or slower than the master axis. The position of the dancer is thereby controlled.

Under normal conditions (e. g. no high-speed command value jumps), the circumferential velocity of a winding axis is approximately adjusted by the fine adjustment of the gear. The tension-controlled axis is approximately adjusted by the electronic gear. Thus, the dancer position controller should thus control only small deviations.

Direction of rotation of the tension or winding axis, control direction of the closed-loop dancer control

Set the direction of rotation of the tension or winding axis using the drive parameter "P-0-0108 Master drive polarity" or the axis parameter "A-0-2798 Polarity of master axis position". If the winding axis does not rotate in the desired direction, the parameter P-0-0108 or axis parameter A-0-2798 has to be reset accordingly. When inverting the drive direction of rotation, the control direction (ControlPolarity) is also to be inverted.

Error handling

The following table lists and describes the error codes of the function block MB_DancerControlType03.

The error codes from the F_RELATED_TABLE (16#0170) are used.

Additionally, the error codes of the IL_PIDType01, RIL_LoopControl and MB_GetCyclicParameterHandle, ML_TechMotionBase function blocks can be output.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1160	16#0020	Error in 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	Number of slave axis incorrect
		16#000F	Number of master axis incorrect
DEVICE_ERROR	16#1162	16#0001	Axis not in operating mode
RESSOURCE_ERROR	16#0004	16#0000	Incorrect drive firmware
RESSOURCE_ERROR	16#000F	16#0003	Drive function package "MA" not enabled

Tab. 8-8: Error codes of the FB MB_DancerControlType03

Required hardware

- CMLx, XMx, VPx
- IndraDrive

Required firmware	• MLC firmware FWA-CMLXX*-MLC-14VRS XX - Device variant, e.g. CML65 • IndraDrive drive firmware
Required software	• IndraWorks 14VRS for MLC or MLD
Special features for the MLC	The scaling type has to be identical for the master and the slave axis. Rotary and preference scaling are recommended. Interpolation in the drive The parameter "P-0-0690 Additive velocity command value, process loop" has to be entered into the cyclic Sercos channel of the corresponding drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks. Interpolation in the control No parameter has to be written to the cyclic Sercos channel of the respective drive. The target variable is the control parameter "A-0-2615 ELS, additive velocity command value, process controller".
Special features for the MLD	"Interpolation in the drive" has to be activated in the drive for the dancer controller axis when using the MLD-S control with a superordinate MLC control (Real axes ▶ Corresp. axis ▶ Properties). This setting does not exist when using the MLD-M control. The scaling type has to be identical for the master and the slave axis. Rotary and preference scaling are recommended. The parameter "P-0-0690 Additive velocity command value, process loop" has to be entered in the cyclic channel between the control and drive of the respective drive ("UserCmdDataX") The real axes are parameterized in the parameterization mode in IndraWorks. (The dialog is in the project tree below the MLD node AxisData). The "AxisData structure" of the drive has to be parameterized ("P-0-1367 PLC configuration, bit 6 is set" or dialog "PLC configuration")
Commissioning the closed-loop dancer position control	To commission the closed-loop dancer position control, refer to chapter 8.2.4 "Commissioning of a closed-loop dancer position control" on page 492.

8.2.3 MB_DancerControlType04

Brief description	The function block cyclically calculates a closed-loop dancer position control for a draw roller with dancer or for a winding axis with dancer. The cyclic output variable of the dancer position controller is the speed offset of the respective drive of a draw roller or winding axis. The current control deviation of the dancer position controller is output. The "AddVelocityDancer" output of the function block affects either the parameter "P-0-0690 Additive velocity command value, process loop" (interpolation in the drive) or the parameter "A-0-2615 ELS, additive velocity command value, process controller" (interpolation in the control). If this is not desired, it can be prevented using the input "DisableCyclicWrites". It is only written to the "AddVelocityDancer" output. The control variable limitation is standardized or the P-gain is adapted via a settable minimum reference diameter (e.g. minimum core diameter). It does not depend on the format length set at the diameter calculator. Diagnostic inputs for the control variable limitation are also available.

RMB_TechWinder.library // MX_TechWinder

Assignment: Target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder
MLD MPx20 and higher with MA function package	MX_TechWinder

Tab. 8-9: Reference table of the MB_DancerControlType04

Interface description

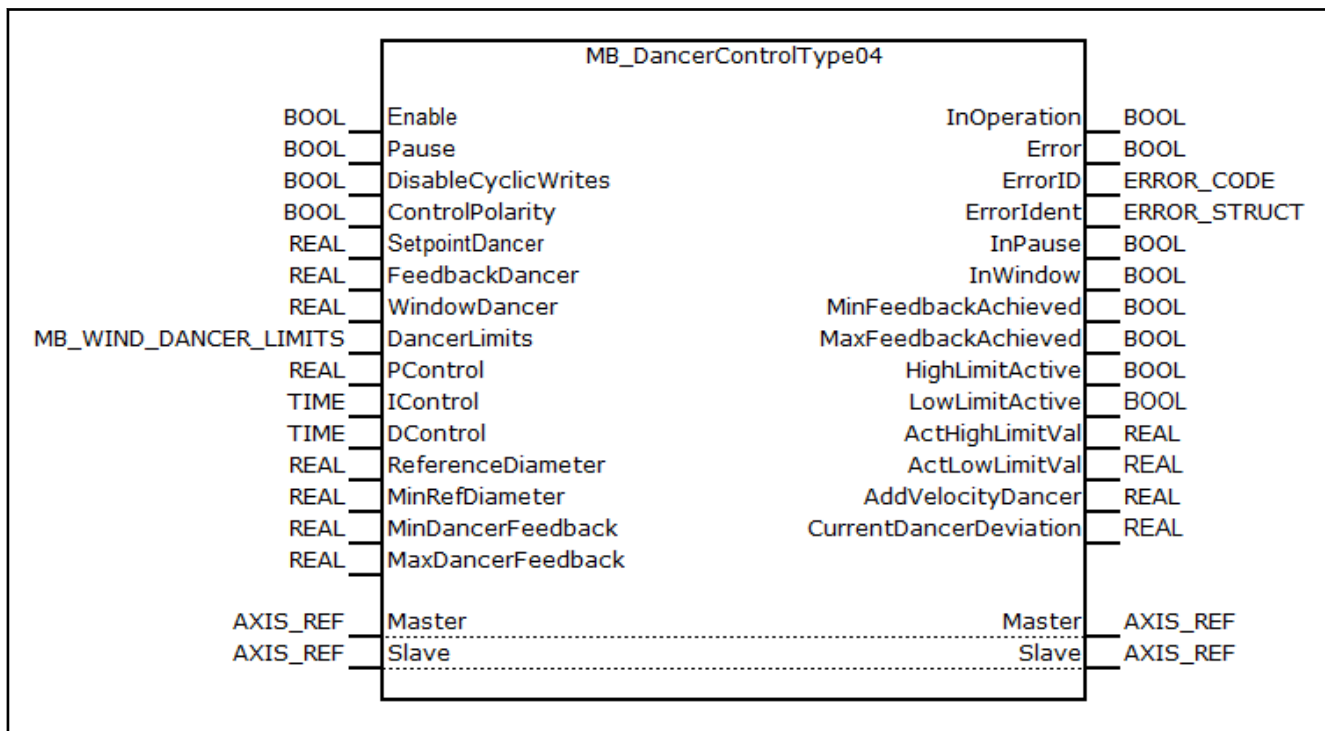


Fig. 8-5: Function block MB_DancerControlType04

I/O type	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Function block processing enabled (level-controlled)
	Pause	BOOL	Dancer controller paused, no velocity adaptation (level-controlled)
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output
	ControlPolarity	BOOL	Polarity of the closed-loop dancer control is inverted (applied at a positive at Enable)
	SetpointDancer	REAL	Command value (setpoint) for the closed-loop dancer control
	FeedbackDancer	REAL	Actual value (feedback) for the dancer position

I/O type	Name	Type	Comment
	WindowDancer	REAL	Windows to monitor the actual dancer position value [%] with regard to the command dancer position value. If the dancer position deviates from the command value by more than the given value, the function block reports this deviation by resetting the "InWindow" output
	DancerLimits	MB_WIND_DANCER_LIMITS	Structure for the automatic control variable limitation of the dancer MB_WIND_DANCER_LIMITS, page 597
	PControl	REAL	Value for the Kp-gain of the closed-loop dancer control
	IControl	TIME	Value for the integral action time of the closed-loop dancer control
	DControl	TIME	Value for the derivative time of the closed-loop dancer control
	ReferenceDiameter	REAL	Reference diameter (value to standardize PControl and the control variable limitation "DancerLimits". That means that the values are internally adapted) <i>For an exclusive dancer position controller:</i> - Draw roller diameter [mm] <i>For winders with dancer:</i> - Assign current winding diameter [mm] from the diameter calculator function block
	MinRefDiameter	REAL	Minimum diameter [mm], (value to standardize PControl) <i>For an exclusive dancer position controller:</i> - Identical to the ReferenceDiameter <i>For winders with dancer:</i> - Smallest core diameter possible
	MinDancerFeedback	REAL	Lower threshold for the actual dancer value (only for a display with binary output)
	MaxDancerFeedback	REAL	Upper threshold for the actual dancer value (only for a display with binary output)
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling, page 490
	InPause	BOOL	Dancer controller pausing
	InWindow	BOOL	Actual dancer value within dancer window
	MinFeedbackAchieved	BOOL	Actual value of the dancer position is below the input "MinDancerFeedback"

I/O type	Name	Type	Comment
	MaxFeedbackAchieved	BOOL	Actual value of the dancer position is above the input "MaxDancerFeedback"
	HighLimitActive	BOOL	Limit of speed offset "HighLimit " is active
	LowLimitActive	BOOL	Limit of speed offset "LowLimit" is active
	ActHighLimitVal	REAL	Current value for the upper limit of the speed offset
	ActLowLimitVal	REAL	Current value for the lower limit of the speed offset
	AddVelocityDancer	REAL	Speed offset for the winding drive or draw roller drive [rpm]
	CurrentDancerDeviation	REAL	Current control deviation of the dancer controller
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to the slave axis (tension-controlled axis)

Tab. 8-10: Interface variables of the MB_DancerControlType04 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
DisableCyclicWrites	BOOL			FALSE	Rising edge at "Enable"
ControlPolarity	BOOL			FALSE	Rising edge at "Enable"
SetpointDancer	REAL	0.0	n. def.	0.0	Continuous
FeedbackDancer	REAL	N. def.	N. def.	0.0	Continuous
WindowDancer	REAL	0.0	n. def.	0.0	Continuous
DancerLimits	STRUCT				
PControl	REAL	0.0	n. def.	0.0	Continuous
IControl	TIME	0 s	N. def.	0 s	Continuous
DControl	TIME	0 s	N. def.	0 s	Continuous
ReferenceDiameter	REAL	0.0	N. def.	100	Continuous
MinRefDiameter	REAL	0.0	n. def.	0.0	Rising edge at "Enable"
MinDancerFeedback	REAL	0.0	MaxDancerFeedback	0.0	Continuous
MaxDancerFeedback	REAL	MinDancerFeedback	N. def.	0.0	Continuous

Tab. 8-11: Min./max. and default values of the MB_DancerControlType04 inputs

Control variable limitation The control variable limitation of the dancer position controller is dynamically adjusted according to the reference axis velocity and the diameter ratio ("MinRefDiameter" / π * "ReferenceDiameter"). This means that, at a higher velocity, the limitation of the additive velocity of the dancer position controller is automatically higher than at a lower velocity. Consideration of the diameter ratio results in the limitation for winding applications automatically being adapted to a changing "ReferenceDiameter" (the limitation reduces automatically with an increasing diameter).

If the control variable of the dancer position controller is limited, this is displayed via the binary outputs "HighLimitActive" and "LowLimitActive". The binary outputs "ActHighLimitVal" and "ActLowLimitVal" show the current values for the limitation.

Signal-time diagram The following signal-time diagram represents selected function block signals.

Relation of the signals

Signal input	Signal output	Description
Enable	InOperation	Commissioning the function block
	Error	Operational readiness was interrupted
Pause	InPause	Pausing the internal controller process

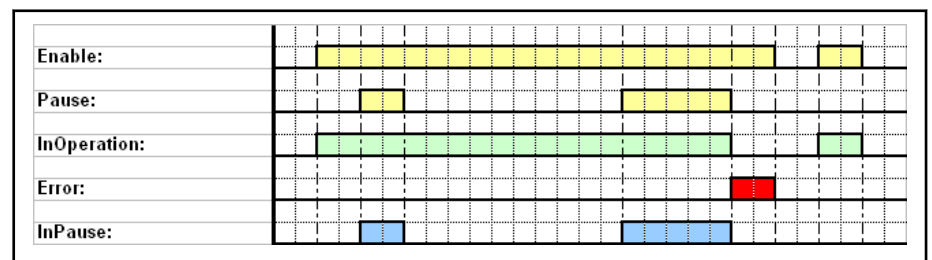


Fig. 8-6: Signal-time diagram of selected inputs and outputs of the function block MB_DancerControlType04

Functional Description The function block cyclically calculates a closed-loop dancer position control for a draw roller with dancer or for a winding axis with dancer.

The IL_PIDType01, RIL_LoopControl function block is used as controller. The controller is a PID-controller. The cyclic output variables of the dancer position controller are the speed offset of the respective drive for the draw roller or the winding axis, as well as the current control deviation of the dancer position controller. The "AddVelocityDancer" output of the function block has an effect on the parameter "P-0-0690 Additive velocity command value, process loop". The current control deviation "CurrentDancerDeviation" is available for analysis. The proportional gain (factor K_p) and the low and high values for control variable limitation "DancerLimits" are automatically adjusted with the "ReferenceDiameter" input based on the "MinRefDiameter" input. The diameter of the draw roller "ReferenceDiameter" and "MinRefDiameter" has to be specified for a closed-loop dancer position control for a draw roller. For a winder with dancer position control, "ReferenceDiameter" is the current winding diameter (the proportional gain of the dancer position control automatically reduces with an increasing diameter) and "MinRefDiameter" is the smallest core diameter possible. If the values for the integral part (integral action time T_n , "IControl") and the lead element (lead time T_D , "DControl") are zero, the internal memory parameters of the PID controller are deleted. There has to be a value proportional to the dancer roll adjustment (e.g. an analog tension value) at the "FeedBackDancer" input. The command value and the actual value must have the same unit.



The function block only processes positive command values and actual values.

Negative command values result in an error.

Negative actual values are limited to zero.

Closed-loop dancer position control

To determine the dancer position, a measuring device is required that converts the dancer position into a proportional signal. The actual position of the dancer is subtracted from a given command value and specified as a control deviation on an internally integrated PID-controller. The control variable of the controller has an effect on the speed command value by addition. Thus, the winding axis rotates faster or slower than the master axis. The position of the dancer is thereby controlled.

Under normal conditions (e. g. no high variations of the speed command values), the approximate adjustment of the circumferential velocity of a winding axis is performed via the fine adjustment of the gear. The electronic gear is used for the approximate adjustment of a tension-controlled axis. Thus, the dancer position controller should thus control only small deviations.

Direction of rotation of the tension or winding axis, control direction of the closed-loop dancer control

Set the direction of rotation of the tension or winding axis using the drive parameter "P-0-0108 Master drive polarity" or the axis parameter "A-0-2798 Polarity of master axis position". If the winding axis does not rotate in the desired direction, the parameter P-0-0108 or axis parameter A-0-2798 has to be reset accordingly. When inverting the drive direction of rotation, the control direction (ControlPolarity) is also to be inverted.

Error handling

The following table lists and describes the error codes of the function block MB_DancerControlType04.

The error codes from the F_RELATED_TABLE (16#0170) are used.

Additionally, the error codes of the IL_PIDType01, RIL_LoopControl and MB_GetCyclicParameterHandle, ML_TechMotionBase function blocks can be output.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1160	16#0020	Error in 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	Number of slave axis incorrect
		16#000F	Number of master axis incorrect
DEVICE_ERROR	16#1162	16#0001	Axis not in operating mode
RESSOURCE_ERROR	16#0004	16#0000	Incorrect drive firmware
RESSOURCE_ERROR	16#000F	16#0003	Drive function package "MA" not enabled

Tab. 8-12: Error codes of the FB MB_DancerControlType04

Required hardware	• CMLx, XMx, VPx • IndraDrive
Required firmware	• MLC firmware FWA-CMLXX*-MLC-14VRS XX - Device variant, e.g. CML65 • IndraDrive drive firmware
Required software	• IndraWorks 14VRS for MLC
Special features for the MLC	The scaling type has to be identical for the master and the slave axis. Rotary and preference scaling are recommended. Interpolation in the drive: The parameter "P-0-0690 Additive velocity command value, process loop" has to be entered into the cyclic Sercos channel of the corresponding drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks. Interpolation in the control: No parameter has to be written to the cyclic Sercos channel of the respective drive. The target variable is the control parameter "A-0-2615 ELS, additive velocity command value, process controller".
Special features for the MLD	"Interpolation in the drive" has to be activated in the drive for the dancer controller axis when using the MLD-S control with a superordinate MLC control (Real axes ▶ Corresp. axis ▶ Properties). This setting does not exist when using the MLD-M control. The scaling type has to be identical for the master and the slave axis. Rotary and preference scaling are recommended.

The parameter "P-0-0690 Additive velocity command value, process loop" has to be entered in the cyclic channel between the control and drive of the respective drive ("UserCmdDataX")

The real axes are parameterized in the parameterization mode in IndraWorks. (The dialog is in the project tree below the MLD node **AxisData**.)

The "AxisData structure" of the drive has to be parameterized ("P-0-1367 PLC configuration, bit 6 is set" or dialog "PLC configuration").

Axis reference for the master axis:

For technology function blocks, the following is recommended as axis reference for the master axis:

- Use axis reference "**VmAxis1**" for a virtual master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 20 (VMA_1).
- Use axis reference "**RmAxis1**" for a real master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 23 (RMA_1).
Additionally, the logic address of the measuring encoder has to be entered in parameter "P-0-1820.0.2 CCD: Source of measuring encoder position feedback value".

For detailed information, refer to Functional Description MLD.

Commissioning the closed-loop dancer position control

To commission the closed-loop dancer position control, refer to [chapter 8.2.4 "Commissioning of a closed-loop dancer position control" on page 492](#).

8.2.4 Commissioning of a closed-loop dancer position control

Drive control loop

It is assumed that the speed control loop of the draw roller drive was set. That means that the parameters "S-0-0100 Velocity loop proportional gain" and "S-0-0101 Velocity loop integral action time" have to be set according to the application.

Commanding the dancer controller axis

The correct operation mode of the drive and the drive enable of the drive have to be set for the dancer controller axis.

There are two ways to set the operation mode and controller enabled:

- Set the "ModeSyncVel" mode using the AxisInterface (recommended)
- Use the PLCopen "MC_Power", "MC_GearIn", "MC_Stop" function blocks to command the axis

Basic settings

First of all, the control direction of the dancer position controller is defined depending on the direction of rotation of the tension axis. The controller acts in opposite directions, depending on this setting. The control direction is defined by setting the "ControlPolarity" input.

Another important basic setting is the setting of the reference diameter ("ReferenceDiameter" input). The currently calculated winding diameter is applied for a winder with dancer (see [Diameter calculator with dancer 508](#)). The diameter of the tension-controlled axis is entered for the dancer position controller. Using the "MinDancerFeedback" and "MaxDancerFeedback" inputs, values for the adjustment range of the dancer can be entered, which are displayed by the "MinFeedbackAchieved" and "MaxFeedbackAchieved" outputs. These values should be within the end positions of the dancer.

The process variable window ("WindowDancer") is now defined. The monitoring of the dancer command position is parameterized in this window. If the actual dancer position value differs from the position command value ("Set-pointDancer"), related to the command value by more than the specified per-

centage, the "InWindow" binary output is reset ("InWindow": = FALSE). This allows the superordinate PLC to respond to the process accordingly.

Exiting the parameterized window does initially not affect the function of the dancer position controller. If, for example, the dancer command value "SetpointDancer" is 500.0 and the window size are set to "WindowDancer" = 10 %, the limits of the window are 450.0 and 550.0. Within this value range, the "InWindow" output = TRUE.

For the function block [MB_DancerControlType03](#), page 479:

Finally, the format length has to be specified. The format length corresponds to an unwound revolution of the reference axis ("FormatLength" input).

For the function block [MB_DancerControlType04](#), page 485:

For a dancer position controller, the identical value as for the "ReferenceDiameter" input is entered for the "MinRefDiameter" input. Enter the smallest core diameter possible for a winder with dancer.

Parameterization of the dancer position controller

The controller parameters are set in the next step:

- Proportional gain ("PControl")
- Integral action time ("IControl")
- Derivative time ("DControl")

The Ziegler and Nichols method can be used to set the parameters. The control loop is closed and the proportional gain ("PControl") is increased until the output of the control loop performs a constant oscillation with the period T_{crit} at the proportional gain $K_{p,crit}$ at a constant input. The setting rules for the gain K_p ("PControl"), the integral action time T_n ("IControl") and the derivative time T_v ("DControl") are as given in the following table.

Controllers	PControl	IControl	DControl
P-controller	$0.5 \cdot K_{p,crit}$		
PI-controller	$0.45 \cdot K_{p,crit}$	$0.85 \cdot T_{crit}$	
PD-controller	$0.55 \cdot K_{p,crit}$		$0.15 \cdot T_{crit}$
PID-controller	$0.6 \cdot K_{p,crit}$	$0.5 \cdot T_{crit}$	$0.12 \cdot T_{crit}$

Tab. 8-13: Controller parameterization according to Ziegler and Nichols

Finally, the inputs of the MB_WIND_DANCER_LIMITS structure are assigned. The operating area of the dancer position controller is determined by specifying the upper and lower limit ("LowLimitControl", "HighLimitControl") of the internal limiter.

Initial startup of the dancer position controller

First, a value is specified as command value at the "SetpointDancer" input. Generally, it is helpful if this value corresponds to the centered position of the dancer. In this case, the dancer has a comparatively long web distance in both directions.

After enabling the dancer position controller ("Enable" = TRUE), the dancer moves to the defined command value position. The web velocity fluctuations cause fluctuations in the web tension that act on the dancer positions as thus on the control deviation of the dancer position controller. The dancer tries to compensate this using the dancer position controller by adapting the drive velocity of the tension-controlled axis or of the winding axis. These variables are displayed at the "CurrentDancerDeviation" and "AddVelocityDancer" outputs and can thus be monitored.

8.3 Diameter calculation with diameter calculator

8.3.1 Introduction and overview

The MB_DiameterCalculatorType03 function block provides a versatile function block for diameter calculators. It is used in all diameter calculators described in this manual. Therefore, it is described first. The function block is described in detail in the following.

8.3.2 MB_DiameterCalculatorType03

Brief description The function block cyclically calculates the current diameter, the velocity adaptation for the winding axis and the current moment of inertia of the winding material on a winder. The function block is called for each winding axis. Cyclic output variables are the current winding diameter "CurrentDiameter", the resulting fine adjustment of the gear for the respective winding drive "GearRatioFineAdjust" and the current moment of inertia of the winding material "CurrentInertia".

The gear fine adjustment "GearRatioFineAdjust" affects either the parameter "P-0-0694 Gear ratio fine adjustment, process loop" (interpolation in the drive) or the parameter "A-0-2605 ELS, Gear ratio fine adjustment process controller" (interpolation on the control). This parameter is entered into the respective cyclic Sercos channel. It is parameterized in the parameterization mode.

Assignment: Target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder
MLD with MPx07/08/17VRS with MA function package	MX_TechWinder.lib (IndraLogic 1.x)
MLD MPx20 and higher with MA function package	MX_TechWinder

Tab. 8-14: Reference table of the MB_DiameterCalculatorType03

Interface description

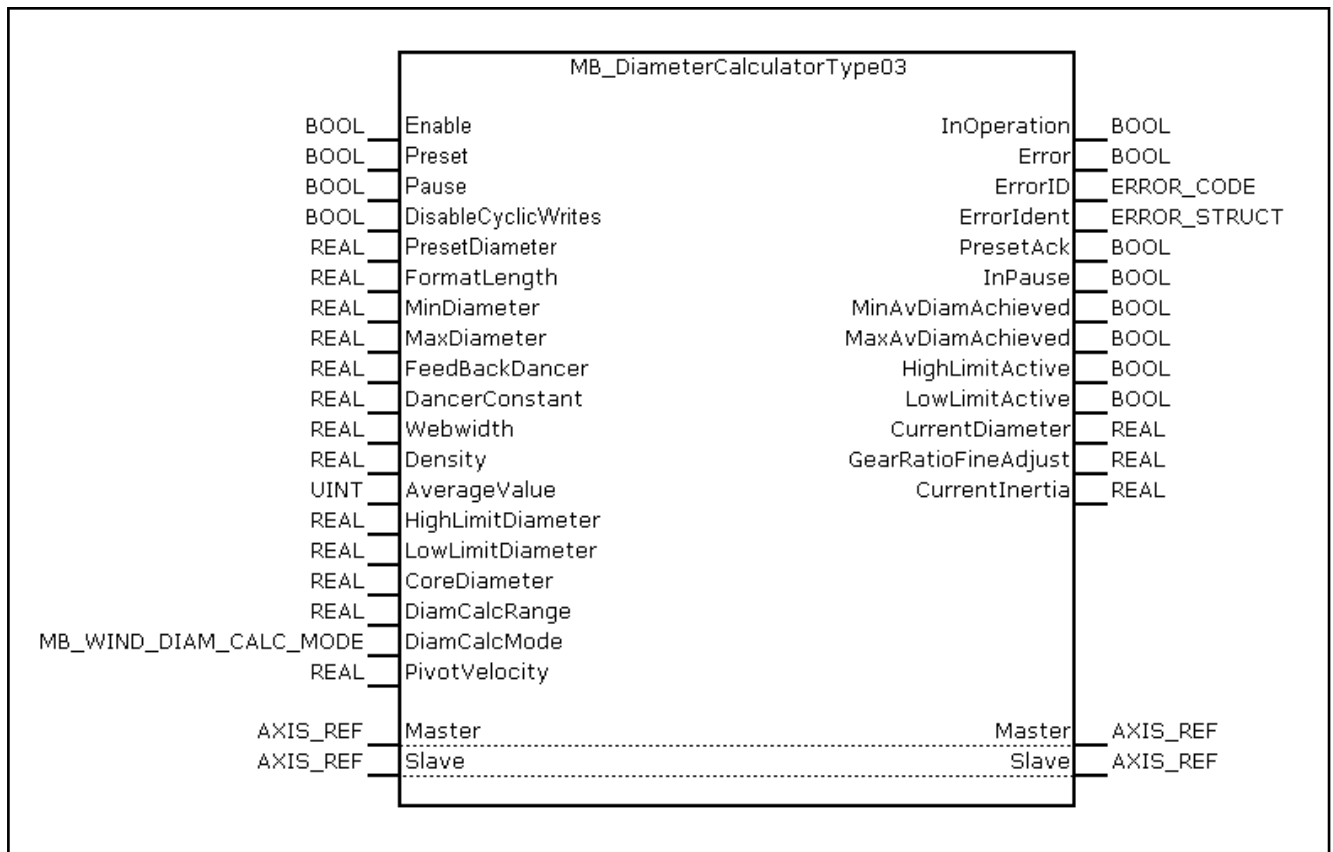


Fig. 8-7: Function block MB_DiameterCalculatorType03

I/O type	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Function block processing enabled (level-controlled)
	Preset	BOOL	Preset of the current winding diameter
	Pause	BOOL	Diameter calculator pauses (level-controlled)
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output
	PresetDiameter	REAL	CurrentDiameter is set to PresetVal, GearRatioFineAdjust is adjusted accordingly (edge-controlled)
	FormatLength	REAL	Format length [mm] (Corresponds to an unwound revolution of the reference axis)
	MinDiameter	REAL	Minimum diameter [mm] Is only used for display purposes by the "MinAvDiamAchieved" output
	MaxDiameter	REAL	Maximum diameter [mm] Is only used for display purposes by the "MaxAvDiamAchieved" output
	FeedbackDancer	REAL	Actual dancer position value (only for the "Winder with dancer" functionality)

I/O type	Name	Type	Comment
	DancerConstant	REAL	Web distance when adjusting the dancer roll related to the unit of the dancer adjustment [mm] Sign has to be set depending on the dancer design! (only for the "Winder with dancer" functionality)
	Webwidth	REAL	Web width [mm]
	Density	REAL	Material density [kg/dm ³]
	AverageValue	UINT	Number of diameter values to be averaged. The input is limited to 30
	HighLimitDiameter	REAL	Upper limit of the diameter [mm]. The diameter is limited to this value
	LowLimitDiameter	REAL	Lower limit of the diameter [mm]. The diameter does not fall below this value
	CoreDiameter	REAL	Core diameter of the winder [mm]
	DiamCalcRange	REAL	Range for the diameter calculation [degrees]. It specifies the number of degrees after which the diameter should be calculated on a regular basis. Relates to the winding axis (load-sided)
	DiamCalcMode	MB_WIND_DI-AM_CALC_MODE	Diameter calculation mode MB_WIND_DIAM_CALC_MODE , page 598
	PivotVelocity	REAL	Pivot velocity of the winder (relative velocity to the winder motion) [mm/s] This corrects the diameter calculation if there is a relative motion of the winding axis
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling , page 500
	PresetAck	BOOL	Preset diameter was applied (is set as long as "Preset" is pending)
	InPause	BOOL	Diameter calculator paused
	MinAvDiamAchieved	BOOL	Minimum diameter reached (Refers to the averaged diameter)
	MaxAvDiamAchieved	BOOL	Maximum diameter reached (Refers to the averaged diameter)
	HighLimitActive	BOOL	Currently calculated diameter has reached the upper limit (Diameter value is discarded)
	LowLimitActive	BOOL	Currently calculated diameter reached lower limit. (Diameter value is discarded)
	CurrentDiameter	REAL	Current winding diameter

I/O type	Name	Type	Comment
	GearRatioFineAdjust	REAL	Fine gear adjustment for speed lagging of the drive [%]
	CurrentInertia	REAL	Current total moment of inertia (related to the motor side [kg/m ²])
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to the slave axis (tension-controlled axis)

Tab. 8-15: Interface variables of the MB_DiameterCalculatorType03 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Preset	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
DisableCyclic-Writes	BOOL			FALSE	Rising edge at "Enable"
PresetDiameter	REAL	> 0.0	N. def.	100.0	Rising edge at Preset
FormatLength	REAL	> 0.0	N. def.	100.0	Rising edge at "Enable"
MinDiameter	REAL	> 0.0	N. def.	100.0	Continuous
MaxDiameter	REAL	> 0.0	N. def.	2000.0	Continuous
FeedbackDancer	REAL	0.0	n. def.	0.0	Continuous
DancerConstant	REAL	N. def.	N. def.	0.0	Continuous
Webwidth	REAL	0.0	n. def.	0.0	Rising edge at "Enable"
Density	REAL	0.0	N. def.	1.0	Rising edge at "Enable"
AverageValue	UINT	1	30	1	Continuous
HighLimitDiameter	REAL	0.0	N. def.	2000.0	Continuous
LowLimitDiameter	REAL	0.0	N. def.	100.0	Continuous
CoreDiameter	REAL	0.0	N. def.	100.0	Rising edge at "Enable"
DiamCalcRange	REAL	0.0	N. def.	360.0	Continuous
DiamCalcMode	ENUM			0	Rising edge at "Enable"
PivotVelocity	REAL	0.0	n. def.	0.0	Continuous

Tab. 8-16: Min./max. and default values of the MB_DiameterCalculatorType03

Signal-time diagram The following signal-time diagram describes the inputs and outputs of the function block.

Relation of the signals

Signal input	Signal output	Description
Enable	InOperation	Commissioning the function block
	Error	Operational readiness was interrupted
Pause	InPause	Internal calculation paused
Preset	PresetAck	The diameter calculator is started with a positive edge at the "Preset" input. The initial value is the value present at the "PresetDiameter" input
PresetDiameter		

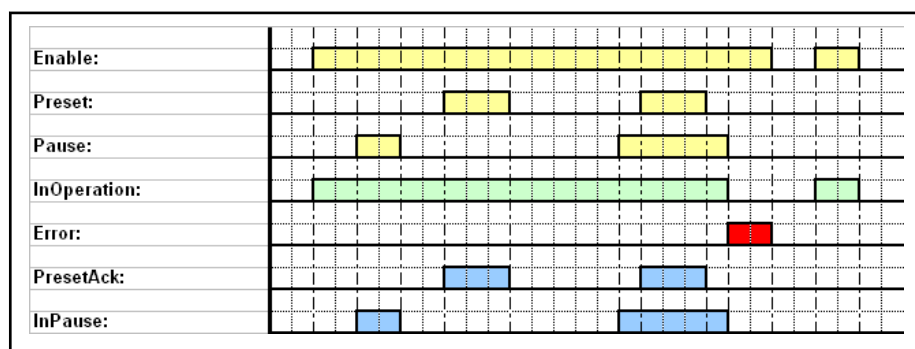


Fig. 8-8: Signal-time diagram of selected inputs and outputs of the MB_DiameterCalculatorType03 function block

Functional Description Preset of the diameter

Since the calculated diameter directly affects the winding axis velocity via the resulting fine adjustment, a reasonable value has to be preset for the diameter at the start of the calculation when activating the diameter calculator. Thus, a value for the diameter, which is close to the actual diameter, should be specified at the input "PresetDiameter" and a preset should be performed. For a rewinder, the value should be slightly lower and for an unwinder, it should be slightly higher. This allows an easier startup in each case.

With each new roll, preset the winding diameter once upon start. Use the "Preset" input. The diameter is set to the value of the "PresetDiameter" input. Setting the diameter using "Preset" is always possible without setting the "Pause" input.

After switching on the control, the winding diameter, the gear fine adjustment and the moment of inertia are all equal zero. This means that a preset should be performed each time the control is switched on to preset the respective variables with reasonable values. If this is not done, no reasonable value can be calculated for the current winding diameter. This can cause a malfunction of the diameter calculator and thus lead to a winder stop.

If it not possible to specify a correct preset value for the initial winding, we recommend the use of the "auto-calc mode" to calculate the diameter (FIRST_REV_AUTO_CALC). In this mode, the range of diameter calculation (DiamCalcRange) is automatically adjusted for the first winding axis revolution after the start to obtain a reasonable starting diameter more quickly.



If the current diameter is to be kept after the control has been switched off, the "CurrentDiameter" output has to be copied to a remanent variable. After switching on, use this value for a preset.

Diameter calculation

The calculation of the diameter is proportional to the quotient from the reference axis velocity n_{Ref} and the winding axis velocity n_{Winder} and to the format length F (corresponding to an unwound revolution of the reference axis).

$$d \sim \frac{n_{Ref} \cdot F}{n_{Winder}}$$

Fig. 8-9: Proportionality in diameter calculation



The ratio "Format length"/"CoreDiameter" may not be greater than 25.

Otherwise, the gear adjustment via the gear fine adjustment is insufficient!



The reference axis has to rotate to calculate the diameter.

The standstill window of the reference axis has to be set accordingly.

The calculated winding diameter directly influences the resulting fine adjustment of the gear. Thus, the velocity of the winder is adjusted according to the diameter.

Average determination

The calculated current diameter "CurrentDiameter" can be averaged by setting the "AverageValue" input.

The number of the diameter values to be averaged is applied at "AverageValue". Only the current, internally calculated, diameters within the limits of "HighLimitDiameter" and "LowLimitDiameter" contribute to the generation of the average value. Values outside these limits are discarded. Values between 1 and 10 are appropriate for "AverageValue". Therefore, "AverageValue" is limited to 30. If a higher value is entered, "AverageValue" is limited to 30. The higher the number selected, the more precise the calculated diameter. However, the accuracy of the diameter is reduced by increasing values for "AverageValue", as it is lagging due to the "filtering". Thus, select a reasonable value for "AverageValue". With smaller winding diameters, the variation in the diameter values calculated is higher. Therefore, the value of the filter may not be too high. Otherwise, the delay of the diameter calculated, caused by the averaging, would result in a distinct "lagging" of the diameter calculated. For a winder with dancer for example, this can result in an undesired displacement of the dancer.

Calculation of the moment of inertia

Using the inputs "Webwidth" and "Density", the moment of inertia of the winder is calculated internally in the diameter calculator and displayed at the output "CurrentInertia". Using the moment of inertia that depends on the diameter, an adaptation of the K_P -gain of the speed loop (S-0-0100) can be implemented in the PLC project in IndraLogic.

Preparation of a splice

Finally, the inputs "MinDiameter" and "MaxDiameter" can be set. The variables specified affect the outputs "MinAvDiamAchieved" and "MaxAvDiamAchieved" and display a message if the currently calculated winding diameter exceeds the specified limits. The respective values should be within the admissible diameter range defined by the inputs "HighLimitDiameter" and "LowLimitDiameter". Thus, the time for specific process steps, for example, for a splice or adjustment of the "DiamCalcRange" input, can be detected and displayed in the PLC, based on the specification of the minimum/maximum diameter.

Error handling The following table lists and describes the error codes of the function block MB_DiameterCalculatorType03.

The error codes from the F_RELATED_TABLE are used.

Additionally, the error codes of the MB_GetCyclicParameterHandle, ML_TechMotionBase function block can be output.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1100	16#0020	Error in state machine
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 invalid
		16#000D	Number of slave axis incorrect
		16#000E	Number of master axis incorrect
DEVICE_ERROR	16#1102	16#0001	No operating mode (no sercos phase 4)
RESSOURCE_ERROR	16#0004	16#0000	Incorrect drive firmware
RESSOURCE_ERROR	16#000F	16#0003	Drive function package "MA" not enabled

Tab. 8-17: Error codes of the FB MB_DiameterCalculatorType03

Required hardware	• CMLx, XMx, VPx • IndraDrive
Required firmware	• MLC firmware FWA-CMLXX*-MLC-14VRS XX - Device variant, e.g. CML65 • IndraDrive drive firmware
Required software	• IndraWorks 14VRS for MLC or MLD

Special features for the MLC

The scaling type has to be identical for the master and the slave axis. Rotary and preference scaling are recommended.

An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear".

The standstill window of the reference axis is to be set as low as possible (e.g. 0.1 rpm). When there is a standstill message of the reference axis, the diameter calculator does not calculate!

Interpolation in the drive: The parameter "P-0-0694 Gear ratio fine adjustment, process loop" has to be entered into the cyclic Sercos channel of the corresponding drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks.

Interpolation in the control: No parameter has to be written to the cyclic Sercos channel of the respective drive. The target variable is the control parameter "A-0-2605 ELS, Gear ratio fine adjustment process controller".

Special features for the MLD

Interpolation in the drive has to be activated in the drive for the winding axis when using the MLD-S control with a superordinate MLC control (**Real axes** ► **Corresp. axis** ► **Properties**). This setting does not exist when using the MLD-M control.

The scaling type has to be identical for master axis and slave axis. Rotary and preference scaling are recommended.

The parameter "P-0-0694 Gear ratio fine adjustment, process loop" has to be entered into the cyclic channel between the control and the drive of the corresponding drive ("UserCmdDataX")

The real axes are parameterized in the parameterization mode in IndraWorks (the dialog is located in the project tree below the MLD node → AxisData).

The "AxisData structure" of the drive has to be parameterized ("P-0-1367 PLC configuration, bit 6 is set" or dialog "PLC configuration").

An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear".

Standstill window of the reference axis:

The standstill window of the reference axis is to be set as low as possible (e.g. 0.1 rpm). When there is a standstill message of the reference axis, the diameter calculator does not calculate!

For the reference axis, the function block cannot access any value for the standstill window. 1 rpm is assumed internally. If this value is not suitable for the application, modify it using the following PLC code lines:

Declaration

```
prMaStandstillWdw: POINTER TO REAL; // for StandstillWindow master MLD
rTestMaStandstillWdw: REAL := 0.5; // for StandstillWindow master MLD
```

Implementation

```
prMaStandstillWdw := ADR(BausteinInstanznameDurchmesserrechner.rMaStandstillWindow);
prMaStandstillWdw^ := rTestMaStandstillWdw;
```

The value of this variable is not saved. Thus, it has to be written again when switching on the PLC.

Axis reference for the master axis:

For technology function blocks, the following is recommended as axis reference for the master axis:

- Use axis reference "**VmAxis1**" for a virtual master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 20 (VMA_1).
- Use axis reference "**RmAxis1**" for a real master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 23 (RMA_1).
Additionally, the logic address of the measuring encoder has to be entered in parameter "P-0-1820.0.2 CCD: Source of measuring encoder position feedback value".

For detailed information, refer to Functional Description MLD.

Commissioning the diameter calculator

To commission the diameter calculator, refer to [chapter 8.3.3 "Commissioning the diameter calculator" on page 502](#).

8.3.3 Commissioning the diameter calculator

Drive control loop

It is assumed that the speed control loop of the winding drive was set. That means that the parameters "S-0-0100 Velocity loop proportional gain" and "S-0-0101 Velocity loop integral action time" have to be set according to the application.

Proceed with the following to commission the diameter calculator:

- Specify the fabric web variables
- Specify the winding variables
- Specify the cyclically changeable inputs
- Specify the inputs for manipulation of the diameter calculation
- Further specifications

Specifying the fabric web variables

The fabric web variables are the inputs and values related to the dimensions of the fabric web. They can normally be considered as constant values during operation. Exceptions are webs of fabric with variable materials or variable dimensions.

The following inputs describe fabric web variables:

- FormatLength
- Webwidth
- Density

Since these variables are not read in cyclically, they have to be applied as input values at the inputs before activating the function block.

Specify the winding variables

Winding variables describe the spatial and general dimensions of the winder. These are the core diameter "CoreDiameter" and the maximum and minimum permissible winding diameters "LowLimitDiameter" and "HighLimitDiameter". Using the ratio of reference axis speed and winding axis speed, the current diameter can be determined with the core diameter. The current diameter "CurrentDiameter" is limited to these defined limits.

Specify the cyclically variable inputs

At the beginning and during the diameter calculation, it can be necessary to manually set the value of the diameter. Therefore, the "PresetDiameter" input is described. It is read in cyclically and thus becomes effective within a cycle.

Specify the inputs for manipulation of the diameter calculation

The method for calculation of the diameter is implemented in such a way that the result of the calculation can be influenced using two inputs of the function

block. "AverageValue" specifies the number of diameter values to be used for averaging. "DiamCalcRange" allows to affect the frequency of the calculation, which, in turn, affects the accuracy of the diameter value. Both variables can be manipulated and adjusted at runtime, as they are read and processed cyclically.

Further specifications

Depending on the respective application, the inputs "DancerConstant", "FeedBackDancer" (both only for a winder with dancer), "MinDiameter" and "MaxDiameter" are assigned. If these inputs are not required for the functions to be implemented in the PLC project, it is recommended to apply the same values to the inputs "MinDiameter" and "MaxDiameter" which have been applied to the inputs "LowLimitDiameter" and "HighLimitDiameter". That avoids a respective error message.

Commanding the winding axis

The correct operation mode of the drive and the drive enable of the drive have to be set for the winding axis.

There are two ways to set the operation mode and controller enabled:

- Set the "ModeSyncVel" mode using the AxisInterface (recommended)
- Use the PLCopen function blocks "MC_Power", "MC_GearIn", "MC_Stop" to command the axis

8.4 Detecting an externally measured diameter

8.4.1 Introduction and overview

The function block [MB_DiameterMeasurementType01](#), page 503, loads a measured diameter and adapts the velocity of the winding drive accordingly. In this case, it replaces the function block [MB_DiameterCalculatorType03](#), page 494.

It is used in the function blocks [MB_WinderDancerCtrlType01](#), page 509, and [MB_WinderTensionCtrlType02](#), page 526,.

8.4.2 MB_DiameterMeasurementType01

Brief description

The function block calculates and loads a measured diameter and calculates the velocity adaptation for the winding axis and the current moment of inertia of the winding material on a winder. The function block is called up for every winding axis, where the diameter is measured. Cyclic output variables are the resulting gear fine adjustment for the associated winding drive "GearRatioFineAdjust" and the current moment of inertia of the winding material "CurrentInertia".

The gear fine adjustment "GearRatioFineAdjust" affects either the parameter "P-0-0694 Gear ratio fine adjustment, process loop" (interpolation in the drive) or the parameter "A-0-2605 ELS, Gear ratio fine adjustment process controller" (interpolation on the control). This parameter is entered into the respective cyclic Sercos channel. It is parameterized in the parameterization mode.

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder

MLD with MPx07/08/17 with MA function package	MX_TechWinder.lib (IndraLogic 1.x)
MLD MPx20 and higher with MA function package	MX_TechWinder

Tab. 8-18: Reference table of the MB_DiameterMeasurementType01

Interface description

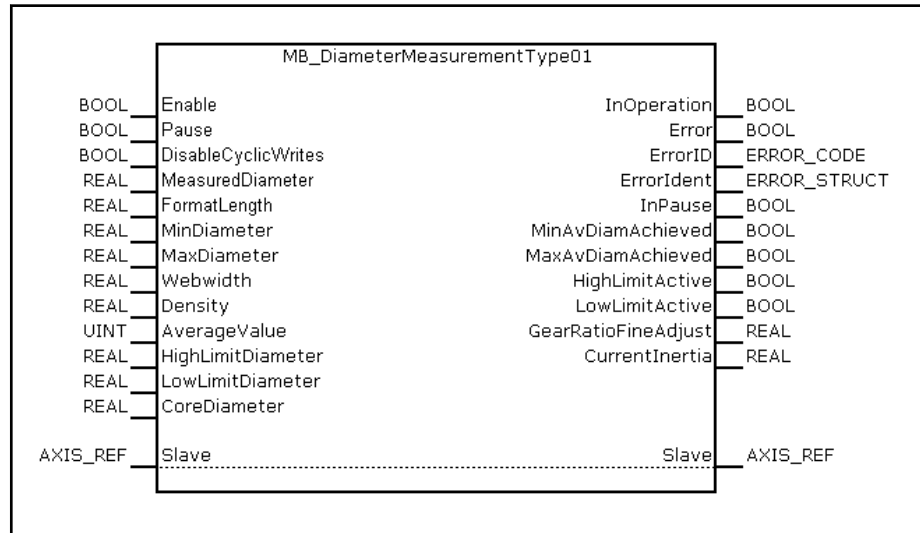


Fig. 8-10: MB_DiameterMeasurementType01 function block

I/O type	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Function block processing enabled (level-controlled)
	Pause	BOOL	Function block pauses, gear fine adjustment remains "frozen"
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output
	MeasuredDiameter	REAL	Measured diameter [mm]
	FormatLength	REAL	Format length [mm] (1 revolution of the master axis unwound)
	MinDiameter	REAL	Minimum diameter [mm], Is only used for display purposes by the "MinAvDiamAchieved" output
	MaxDiameter	REAL	Maximum diameter [mm], Is only used for display purposes by the "MaxAvDiamAchieved" output
	WebWidth	REAL	Web width [mm]
	Density	REAL	Material density [kg/dm ³]
	AverageValue	UINT	Number of diameter values to be averaged. The input is limited to 30
	HighLimitDiameter	REAL	Upper limit for the diameter [mm] The diameter is limited to this value

I/O type	Name	Type	Comment
	LowLimitDiameter	REAL	Lower limit for the diameter [mm] The diameter is limited to this value
	CoreDiameter	REAL	Core diameter of the winder [mm]
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics "Error handling" on page 506
	InPause	BOOL	Dancer controller pausing
	MinAvDiamAchieved	BOOL	Minimum diameter reached (Refers to the averaged diameter)
	MaxAvDiamAchieved	BOOL	Maximum diameter reached (Refers to the averaged diameter)
	HighLimitActive	BOOL	Currently calculated diameter has reached the upper limit (Diameter value is discarded)
	LowLimitActive	BOOL	Currently calculated diameter has reached the lower limit (Diameter value is discarded)
	GearRatioFineAdjust	REAL	Fine gear adjustment for speed lagging of the drive [%]
	CurrentInertia	REAL	Current total moment of inertia (Refers to the motor side) [kg/m ²]
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Reference to the slave axis

Tab. 8-19: Interface variables of the MB_DiameterMeasurementType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
DisableCyclic-Writes	BOOL			FALSE	Rising edge at Enable
MeasuredDiameter	REAL	> 0.0	n. def.	100.0	Continuous
FormatLength	REAL	> 0.0	n. def.	100.0	Rising edge at Enable
MinDiameter	REAL	> 0.0	n. def.	100.0	Continuous
MaxDiameter	REAL	> 0.0	n. def.	2000.0	Continuous

Name	Type	Min. value	Max. value	Default value	Effective
Webwidth	REAL	0.0	n. def.	0.0	Rising edge at Enable
Density	REAL	0.0	n. def.	1.0	Rising edge at Enable
AverageValue	UINT	0	30	1	Continuous
HighLimitDiameter	REAL	0.0	n. def.	2000.0	Continuous
LowLimitDiameter	REAL	0.0	n. def.	100.0	Continuous
CoreDiameter	REAL	0.0	n. def.	100.0	Rising edge at Enable

Tab. 8-20: Min./max. and default values of the MB_DiameterMeasurementType01 inputs

Signal-time diagram

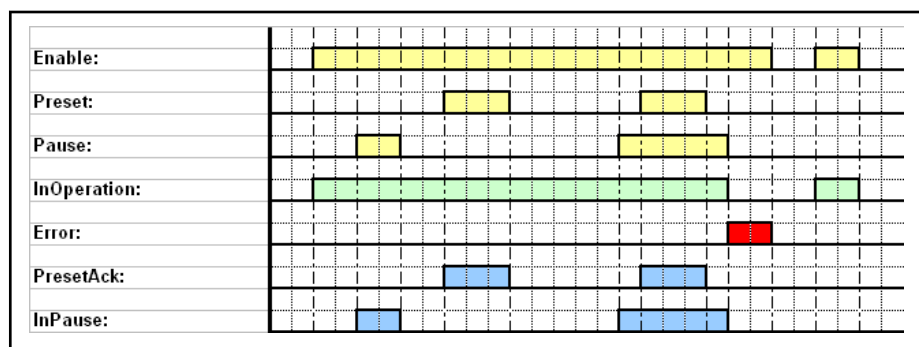


Fig. 8-11: Signal-time diagram of selected inputs and outputs of the function block MB_DiameterMeasurementType01

Functional description

The function block MB_DiameterMeasurementType01 reads a measured diameter value (e.g. with an ultra-sound sensor) and ensures the correct velocity adjustment of the respective winding axis based on the diameter measured.

The function block is used in the function blocks [MB_WinderDancerCtrlType01](#), page 509, and [MB_WinderTensionCtrlType02](#), page 526,.



The ratio "Format length"/"CoreDiameter" may not be greater than 25. Otherwise, the gear adjustment via the gear fine adjustment is insufficient.

Error handling

The following table lists and describes the error codes of the function block MB_DiameterMeasurementType01.

The error codes from the F_RELATED_TABLE (16#0170) are used.

Additionally, the error codes of the MB_GetCyclicParameterHandle, ML_TechMotionBase function block can be output.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1180	16#0020	Error in state machine
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
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

ErrorID	Additional1	Additional2	Text
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	Incorrect slave axis number
DEVICE_ERROR	16#1182	16#0001	No operating mode (sercos phase 4)
RESSOURCE_ERROR	16#0004	16#0000	Incorrect drive firmware
RESSOURCE_ERROR	16#000F	16#0003	Drive function package "MA" not enabled

Tab. 8-21: Error codes of the FB MB_DiameterMeasurementType01

Required hardware	• CMLx, XMx, VPx • IndraDrive
Required firmware	• MLC firmware FWA-CMLXX*-MLC-14VRS XX - Device variant, e.g. CML65 • IndraDrive drive firmware
Required software	• IndraWorks 14VRS for MLC or MLD
Special features for the MLC	The scaling type has to be identical for the master and the slave axis. Rotary and preference scaling are recommended. An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear". The standstill window of the reference axis is to be set as low as possible (e.g. 0.1 rpm). When there is a standstill message of the reference axis, the diameter calculator does not calculate! Interpolation in the drive: The parameter "P-0-0694 Gear ratio fine adjustment, process loop" has to be entered into the cyclic Sercos channel of the corresponding drive "UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks. Interpolation in the control: No parameter has to be written to the cyclic Sercos channel of the respective drive. The target variable is the control parameter "A-0-2605 ELS, Gear ratio fine adjustment process controller".
Special features for the MLD	Interpolation in the drive has to be activated in the drive for the winding axis when using the MLD-S control with a superordinate MLC control (Real axes → Corresp. axis → Properties). This setting does not exist when using the MLD-M control. The scaling type has to be identical for the master and the slave axis. Rotary and preference scaling are recommended. The parameter "P-0-0694 Gear ratio fine adjustment, process loop" has to be entered into the cyclic channel between the control and the drive of the corresponding drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks. (The dialog is in the project tree below the MLD node → AxisData.) The "AxisData structure" of the drive has to be parameterized ("P-0-1367 PLC configuration, bit 6 is set" or dialog "PLC configuration")

An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear".

Axis reference for the master axis:

For technology function blocks, the following is recommended as axis reference for the master axis:

- Use axis reference "**VmAxis1**" for a virtual master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 20 (VMA_1).
- Use axis reference "**RmAxis1**" for a real master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 23 (RMA_1).
Additionally, the logic address of the measuring encoder has to be entered in parameter "P-0-1820.0.2 CCD: Source of measuring encoder position feedback value".

For detailed information, refer to Functional Description MLD.

8.5 Diameter calculator with dancer

8.5.1 Introduction and overview

The diameter calculator with dancer is used for center-driven winders. The tension is specified by a force which acts upon the movable part of the dancer. Apply this force by a weight, a spring or a pneumatic element.

The closed-loop control of the dancer position is performed in the dancer position controller. The actual value of the dancer position is recorded for example via an analog channel and is transferred to the closed-loop control.

The current winding diameter is calculated by an internally integrated diameter calculator. The velocity of the winding axis is adjusted using the calculated winding diameter. Thus, the calculated winding diameter replaces an actual diameter recorded via a sensor system.

The diameter calculator with dancer thus represents a combination of a winder (represented by the winding axis and the diameter calculator) and a dancer position controller. The diameter calculator influences the gear fine adjustment and the dancer position controller influences the additive speed command value.

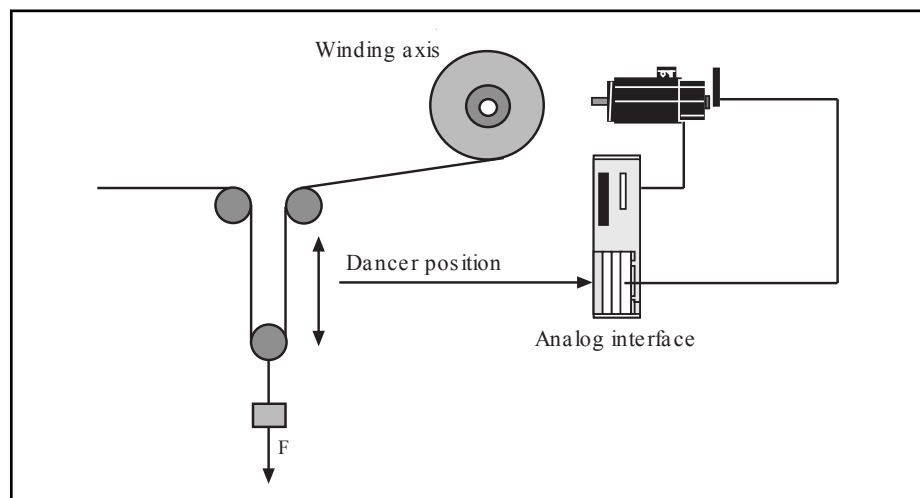


Fig. 8-12: Principle of the diameter calculator with dancer position controller

8.5.2 MB_WinderDancerCtrlType01

Assignment: Target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder
MLD with MPx07/08/17 with MA function package	MX_TechWinder.lib

Tab. 8-22: Reference table of the MB_WinderDancerCtrlType01

Interface description

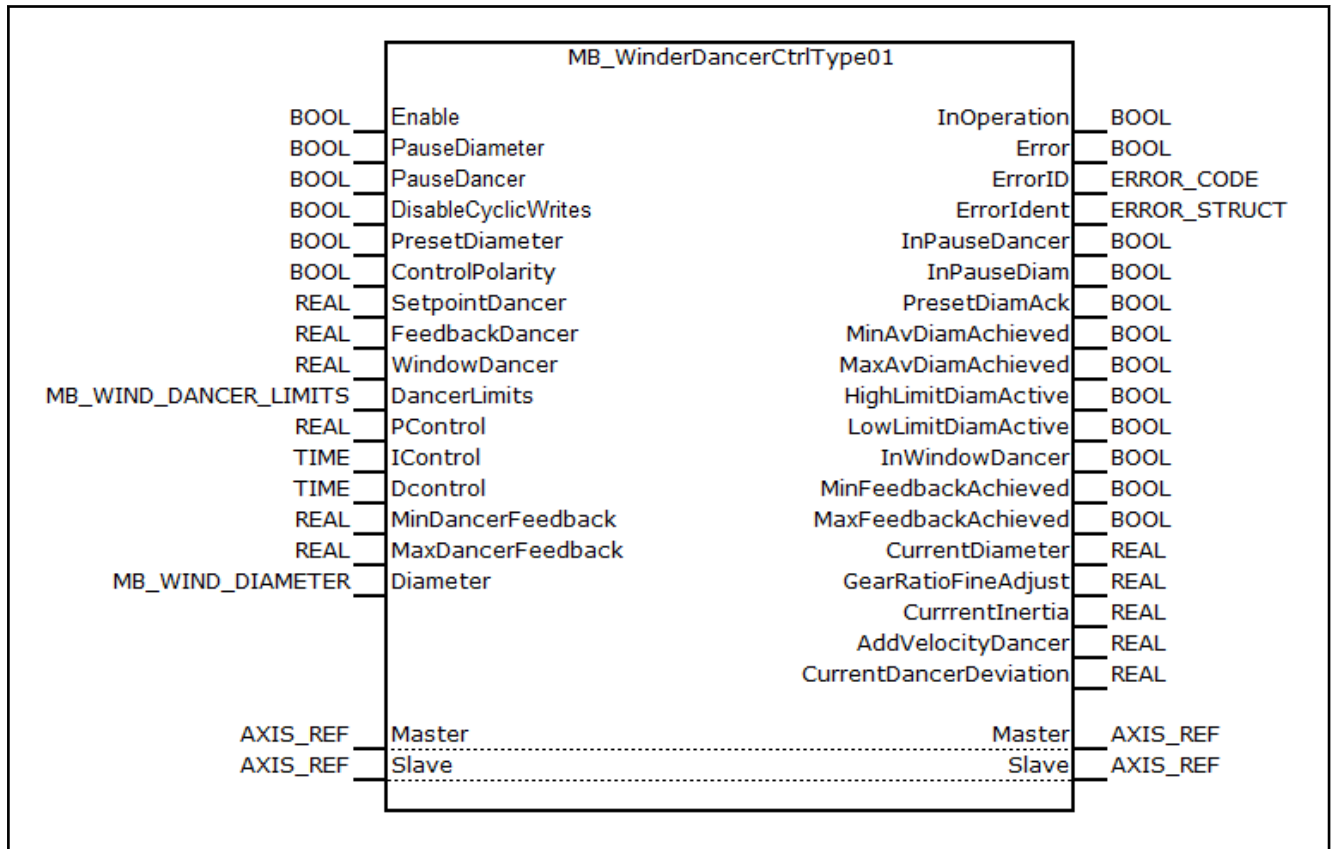


Fig. 8-13: MB_WinderDancerCtrlType01 function block

I/O type	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Function block processing enabled (level-controlled)
	PauseDiameter	BOOL	Diameter calculator paused; no further diameter calculation and velocity adjustment (level-controlled)
	PauseDancer	BOOL	Dancer controller paused (level-controlled)
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output
	PresetDiameter	BOOL	CurrentDiameter is set to PresetVal. GearRatioFineAdj is adjusted accordingly (edge-controlled)
	ControlPolarity	BOOL	Polarity of the closed-loop dancer control is inverted (applied at a positive at Enable)

I/O type	Name	Type	Comment
	SetpointDancer	REAL	Command value (setpoint) for the closed-loop dancer control
	FeedbackDancer	REAL	Actual value (feedback) for the dancer position
	WindowDancer	REAL	Window to monitor the actual dancer position value [%] with regard to the command dancer position value. If the dancer position deviates from the command value by more than the given value, the function block reports this deviation by resetting the "InWindow" output
	DancerLimits	MB_WIND_DANCER_LIMITS	Structure for the automatic control variable limitation of the dancer MB_WIND_DANCER_LIMITS, page 597
	PControl	REAL	Value for the Kp-gain of the closed-loop dancer control
	IControl	TIME	Value for the integral action time of the closed-loop dancer control
	DControl	TIME	Value for the derivative time of the closed-loop dancer control
	MinDancerFeedback	REAL	Lower threshold for the actual dancer value (only for a display with binary output)
	MaxDancerFeedback	REAL	Upper threshold for the actual dancer value (only for a display with binary output)
	Diameter	MB_WIND_DIAMETER	Structure for diameter calculator MB_WIND_DIAMETER , page 598
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling, page 514
	InPauseDancer	BOOL	Dancer controller pausing
	InPauseDiam	BOOL	Diameter calculator paused
	PresetDiamAck	BOOL	Preset diameter was applied (Set if "Preset" is present)
	MinAvDiamAchieved	BOOL	Minimum diameter reached (Refers to the averaged diameter)
	MaxAvDiamAchieved	BOOL	Maximum diameter reached (Refers to the averaged diameter)
	HighLimitDiamActive	BOOL	Currently calculated diameter has reached the upper limit (Diameter value is discarded)

I/O type	Name	Type	Comment
	LowLimitDiamActive	BOOL	Currently calculated diameter has reached the lower limit (diameter value is discarded)
	InWindowDancer	BOOL	Actual dancer value within dancer window
	MinFeedbackAchieved	BOOL	Actual value of the dancer position is below the input "MinDancerFeedback"
	MaxFeedbackAchieved	BOOL	Actual value of the dancer position is above the input "MaxDancerFeedback"
	CurrentDiameter	REAL	Current winding diameter
	GearRatioFineAdjust	REAL	Fine gear adjustment for speed lagging of the drive [%]
	CurrentInertia	REAL	Current moment of inertia of the winding material + mechanics (Refers to the motor side) [kg m ²]
	AddVelocityDancer	REAL	Speed offset for the winding drive [rpm]
	CurrentDancerDeviation	REAL	Current control deviation of the dancer controller
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to the slave axis (winding axis)

Tab. 8-23: Interface variables of the MB_WinderDancerCtrlType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
PauseDiameter	BOOL			FALSE	Continuous
PauseDancer	BOOL			FALSE	Continuous
DisableCyclicWrites	BOOL			FALSE	Rising edge at "Enable"
PresetDiameter	BOOL			FALSE	Continuous
ControlPolarity	BOOL			FALSE	Rising edge at "Enable"
SetpointDancer	REAL	0.0	n. def.	0.0	Continuous
FeedbackDancer	REAL	N. def.	N. def.	0.0	Continuous
WindowDancer	REAL	0.0	n. def.	0.0	Continuous
DancerLimits	STRUCT				
PControl	REAL	0.0	n. def.	0.0	Continuous
IControl	TIME	0 s	N. def.	0 s	Continuous
DControl	TIME	0 s	N. def.	0 s	Continuous

Name	Type	Min. value	Max. value	Default value	Effective
MinDancerFeed-back	REAL	0.0	MaxDancerFeed-back	0.0	Continuous
MaxDancerFeed-back	REAL	MinDancerFeed-back	N. def.	0.0	Continuous
Diameter	STRUCT				Depending on the structure element

Tab. 8-24: Min./max. and default values of the MB_WinderDancerCtrlType01 inputs

Functional Description

Diameter calculation

The diameter calculation provides the currently calculated diameter via the cyclically called function block [MB_DiameterCalculatorType03](#), page 494. The additive fine adjustment of the winding axis is set with regard to the diameter. The axis is moved faster or slower in relation to the reference axis to achieve a constant tension of the fabric web.

A relative motion of the dancer roll is taken into account in the diameter calculation using the "DancerConstant" input. For this purpose, determine this dancer constant once.

The following procedure is recommended:

1. Where there is deflection of the dancer from one end position to the other end position, determine the web distance (in mm).
2. Determine the difference between the two actual dancer values of the respective end position.
3. Calculate the quotient of both values.
4. Determine the sign of the dancer constant by conducting a practical test. For this purpose, the diameter calculation is observed at a slowly rotating winding axis and reference axis. Command value jumps to the dancer command value should not significantly affect the calculation of the diameter if the sign is correct.
5. "Subsequently optimize" the calculated value for the dancer constant as described in step 4 if required.



If the current diameter is to be kept after the control has been switched off, the "CurrentDiameter" output has to be copied to a remanent variable. After switching on, use this value for a preset.

Splice

After a splice, the winding diameter generally changes abruptly. The last calculated internal value for the winding diameter does no longer correspond to the current diameter. For a correct diameter calculation, a winding diameter preset is reasonable. The input value for the preset should correspond approximately to the actual value of the winding diameter to achieve the most accurate resulting fine adjustment possible.



If the preset is not performed, malfunctions can occur. Thus, perform the preset before activating the diameter calculator.

With a flying splice, it has to be ensured that the circumferential velocity of the new roll corresponds to the current one. This has to be ensured with an appropriate PLC project in the IndraLogic. The "Pause" input can be used when activating the diameter calculator for the new roll, for example "Pause" can be disabled after removing the old roll.

Closed-loop dancer position control

To determine the dancer position, a measuring device is required that converts the dancer position into a proportional signal. The actual position of the dancer is subtracted from a given command value and added as a control deviation to an internal PID-controller. The control variable of the controller has an effect on the speed command value by addition. Thus, the winding axis rotates faster or slower than the master axis. The tension of the dancer is thereby controlled ([Diameter calculator with open-loop and closed-loop tension control](#) , page 524).

Under normal conditions (e. g. no high variations of the speed command value or abrupt changes in the diameter of the winder), the approximate adjustment of the circumferential velocity of the winding axis is performed by the fine adjustment of the gear. Therefore, the diameter calculator described previously is used. Thus, the dancer position controller has to control only small deviations.

Direction of rotation of the winding axis, control direction of the closed-loop dancer control

Set the direction of rotation of the tension or winding axis using the drive parameter "P-0-0108 Master drive polarity" or the axis parameter "A-0-2798 Polarity of master axis position". If the winding axis does not rotate in the desired direction, the parameter P-0-0108 or axis parameter A-0-2798 has to be reset accordingly. When inverting the drive direction of rotation, the control direction (ControlPolarity) is also to be inverted.

Functional diagram of the dancer position controller

The following functional diagram schematically explains the functions of the diameter calculator with dancer.

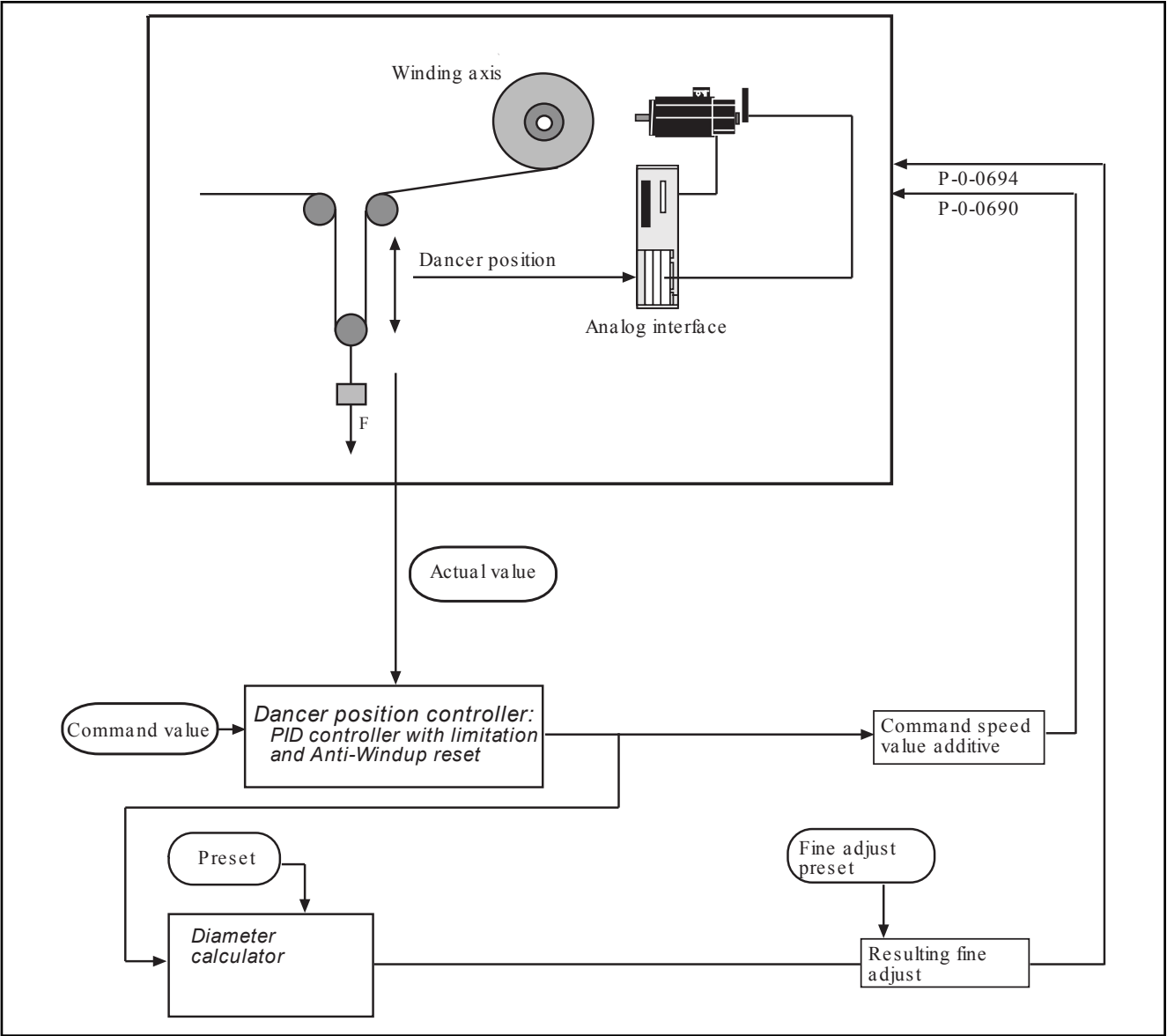


Fig. 8-14: Functional diagram of the diameter calculator with dancer

Error handling The following table lists and describes the error codes of the function block MB_WinderDancerCtrlType01.

The error codes from the F_RELATED_TABLE (16#0170) are used.

Additionally, the error codes of the function blocks MB_DancerControlType03, MB_DiameterCalculatorType03, MB_DiameterMeasurementType01, IL_PIDType01, RIL_LoopControl, as well as MB_GetCyclicParameterHandle, ML_TechMotionBase, can be output.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1190	16#0020	Error in state machine

Tab. 8-25: Error codes of the FB MB_WinderDancerCtrlType01

- Required hardware**
- CMLx, XMx, VPx
 - IndraDrive
- Required firmware**
- MLC firmware FWA-CMLXX*-MLC-14VRS

	XX - Device variant, e.g. CML65
Required software	• IndraDrive drive firmware • IndraWorks 14VRS for MLC or MLD
Special features for the MLC	The scaling type has to be identical for the master and the slave axis. Rotary and preference scaling are recommended. An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear". The standstill window of the reference axis is to be set as low as possible (e.g. 0.1 rpm). When there is a standstill message of the reference axis, the diameter calculator does not calculate! Interpolation in the drive: The parameters "P-0-0694 Gear ratio fine adjustment, process loop" and Parameter "P-0-0690 Additive velocity command value, process loop" have to be entered into the cyclic Sercos channel of the corresponding drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks. Interpolation in the control: No parameter has to be written to the cyclic Sercos channel of the respective drive. The target variables are the control parameters "A-0-2605 ELS, Gear ratio fine adjustment process controller" and "A-0-2615 ELS, additive velocity command value, process controller".
Special features for the MLD	Interpolation in the drive has to be activated in the drive for the winding axis when using the MLD-S control with a superordinate MLC control (Real axes ► Corresp. axis ► Properties). This setting does not exist when using the MLD-M control. The scaling type has to be identical for master axis and slave axis. Rotary and preference scaling are recommended. The parameters "P-0-0694 Gear ratio fine adjustment, process loop" and "P-0-0690 Additive velocity command value, process loop" have to be entered into the cyclic Sercos channel of the corresponding drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks. The real axes are parameterized in the parameterization mode in IndraWorks (the dialog is located in the project tree below the MLD node AxisData). The "AxisData structure" of the drive has to be parameterized ("P-0-1367 PLC configuration, bit 6 is set" or dialog "PLC configuration"). An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear". The standstill window of the reference axis is to be set as low as possible (e.g. 0.1 rpm). When there is a standstill message of the reference axis, the diameter calculator does not calculate!
Commissioning the diameter calculator with dancer	To commission the diameter calculator with dancer, refer to chapter 8.5.4 "Commissioning the diameter calculator with dancer" on page 523.

8.5.3 MB_WinderDancerCtrlType02

Assignment: Target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder
MLD MPx20 and higher with MA function package	MX_TechWinder

Tab. 8-26: Reference table of the MB_WinderDancerCtrlType02

Interface description

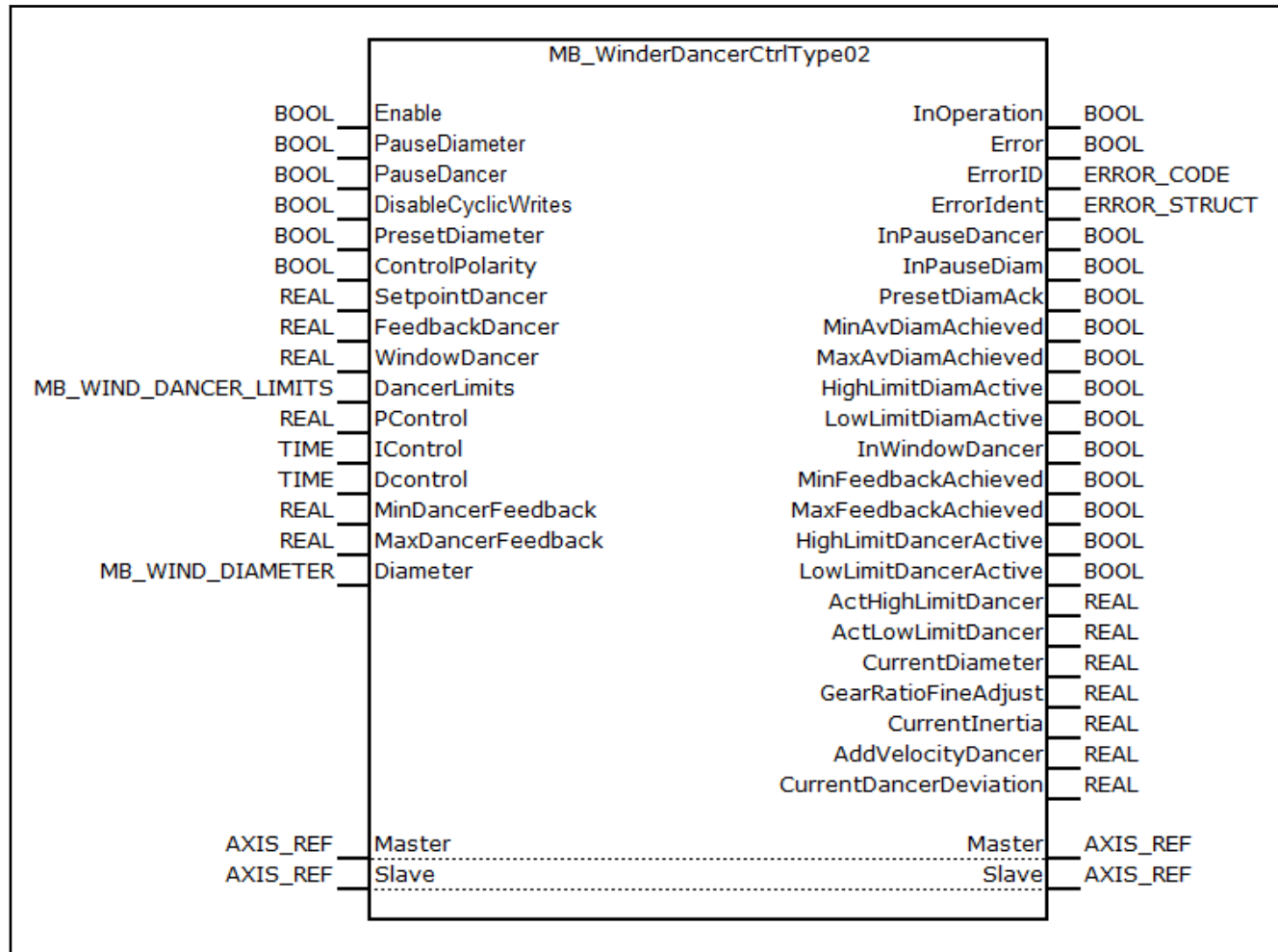


Fig. 8-15: MB_WinderDancerCtrlType02 function block

I/O type	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Function block processing enabled (level-controlled)
	PauseDiameter	BOOL	Diameter calculator paused; no further diameter calculation and velocity adjustment (level-controlled)
	PauseDancer	BOOL	Dancer controller paused (level-controlled)
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output

I/O type	Name	Type	Comment
	PresetDiameter	BOOL	CurrentDiameter is set to PresetVal. GearRatioFineAdj is adjusted accordingly (edge-controlled)
	ControlPolarity	BOOL	Polarity of the closed-loop dancer control is inverted (applied at a positive at Enable)
	SetpointDancer	REAL	Command value (setpoint) for the closed-loop dancer control
	FeedbackDancer	REAL	Actual value (feedback) for the dancer position
	WindowDancer	REAL	Window to monitor the actual dancer position value [%] with regard to the command dancer position value. If the dancer position deviates from the command value by more than the given value, the function block reports this deviation by resetting the "InWindow" output
	DancerLimits	MB_WIND_DANCER_LIMITS	Structure for the automatic control variable limitation of the dancer MB_WIND_DANCER_LIMITS, page 597
	PControl	REAL	Value for the Kp-gain of the closed-loop dancer control
	IControl	TIME	Value for the integral action time of the closed-loop dancer control
	DControl	TIME	Value for the derivative time of the closed-loop dancer control
	MinDancerFeedback	REAL	Lower threshold for the actual dancer value (only for a display with binary output)
	MaxDancerFeedback	REAL	Upper threshold for the actual dancer value (only for a display with binary output)
	Diameter	MB_WIND_DIAMETER	Structure for diameter calculator MB_WIND_DIAMETER, page 598
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling, page 521
	InPauseDancer	BOOL	Dancer controller pausing
	InPauseDiam	BOOL	Diameter calculator paused
	PresetDiamAck	BOOL	Preset diameter was applied (Set if "Preset" is present)
	MinAvDiamAchieved	BOOL	Minimum diameter reached (Refers to the averaged diameter)
	MaxAvDiamAchieved	BOOL	Maximum diameter reached (Refers to the averaged diameter)

I/O type	Name	Type	Comment
	HighLimitDiamActive	BOOL	Currently calculated diameter has reached the upper limit (Diameter value is discarded)
	LowLimitDiamActive	BOOL	Currently calculated diameter has reached the lower limit (diameter value is discarded)
	InWindowDancer	BOOL	Actual dancer value within dancer window
	MinFeedbackAchieved	BOOL	Actual value of the dancer position is below the input "MinDancerFeedback"
	MaxFeedbackAchieved	BOOL	Actual value of the dancer position is above the input "MaxDancerFeedback"
	HighLimitDancerActive	BOOL	Limit of speed offset "HighLimit " is active
	LowLimitDancerActive	BOOL	Limit of speed offset "LowLimit" is active
	ActHighLimitDancer	REAL	Current value for the upper limit of the speed offset
	ActLowLimitDancer	REAL	Current value for the lower limit of the speed offset
	CurrentDiameter	REAL	Current winding diameter
	GearRatioFineAdjust	REAL	Fine gear adjustment for speed lagging of the drive [%]
	CurrentInertia	REAL	Current moment of inertia of the winding material + mechanics (Refers to the motor side) [kg m ²]
	AddVelocityDancer	REAL	Speed offset for the winding drive [rpm]
	CurrentDancerDeviation	REAL	Current control deviation of the dancer controller
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to the slave axis (winding axis)

Tab. 8-27: Interface variables of the MB_WinderDancerCtrlType02 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
PauseDiameter	BOOL			FALSE	Continuous
PauseDancer	BOOL			FALSE	Continuous
DisableCyclicWrites	BOOL			FALSE	Rising edge at "Enable"
PresetDiameter	BOOL			FALSE	Continuous
ControlPolarity	BOOL			FALSE	Rising edge at "Enable"
SetpointDancer	REAL	0.0	n. def.	0.0	Continuous

Name	Type	Min. value	Max. value	Default value	Effective
FeedbackDancer	REAL	N. def.	N. def.	0.0	Continuous
WindowDancer	REAL	0.0	n. def.	0.0	Continuous
DancerLimits	STRUCT				
PControl	REAL	0.0	n. def.	0.0	Continuous
IControl	TIME	0 s	N. def.	0 s	Continuous
DControl	TIME	0 s	N. def.	0 s	Continuous
MinDancerFeed-back	REAL	0.0	MaxDancerFeed-back	0.0	Continuous
MaxDancerFeed-back	REAL	MinDancerFeed-back	N. def.	0.0	Continuous
Diameter	STRUCT				Depending on the structure element

Tab. 8-28: Min./max. and default values of the MB_WinderDancerCtrlType02 inputs

Functional Description Diameter calculation

The diameter calculation provides the currently calculated diameter via the cyclically called function block MB_DiameterCalculatorType03. The additive fine adjustment of the winding axis is set with regard to the diameter. The axis is moved faster or slower in relation to the reference axis to achieve a constant tension of the fabric web.

A relative motion of the dancer roll is taken into account in the diameter calculation using the "DancerConstant" input. For this purpose, determine this dancer constant once.

The following procedure is recommended:

1. Where there is deflection of the dancer from one end position to the other end position, determine the web distance (in mm).
2. Determine the difference between the two actual dancer values of the respective end position.
3. Calculate the quotient of both values.
4. Determine the sign of the dancer constant by conducting a practical test. For this purpose, the diameter calculation is observed at a slowly rotating winding axis and reference axis. Command value jumps to the dancer command value should not significantly affect the calculation of the diameter if the sign is correct.
5. "Subsequently optimize" the calculated value for the dancer constant as described in step 4 if required.



If the current diameter is to be kept after the control has been switched off, the "CurrentDiameter" output has to be copied to a remanent variable. After switching on, use this value for a preset.

Splice

After a splice, the winding diameter generally changes abruptly. The last calculated internal value for the winding diameter does no longer correspond to the current diameter. For a correct diameter calculation, a winding diameter preset is reasonable. The input value for the preset should correspond ap-

proximately to the actual value of the winding diameter to achieve the most accurate resulting fine adjustment possible.



If the preset is not performed, malfunctions can occur. Thus, perform the preset before activating the diameter calculator.

With a flying splice, it has to be ensured that the circumferential velocity of the new roll corresponds to the current one. This has to be ensured with an appropriate PLC project in the IndraLogic. When adding the diameter calculator for the new roll, the "Pause" input can be used and disabled again after removing the old roll "Pause".

Closed-loop dancer position control

To determine the dancer position, a measuring device is required that converts the dancer position into a proportional signal. The actual position of the dancer is subtracted from a given command value and added as a control deviation to an internal PID-controller. The control variable of the controller has an effect on the speed command value by addition. Thus, the winding axis rotates faster or slower than the master axis. This controls the tension of the dancer (cf. [Diameter calculator with open-loop and closed-loop tension control](#), page 524).

Under normal conditions (e. g. no high variations of the speed command value or abrupt changes in the diameter of the winder), the approximate adjustment of the circumferential velocity of the winding axis is performed by the fine adjustment of the gear. Therefore, the diameter calculator described previously is used. Thus, the dancer position controller has to control only small deviations.

Direction of rotation of the winding axis, control direction of the closed-loop dancer control

Set the direction of rotation of the tension or winding axis using the drive parameter "P-0-0108 Master drive polarity" or the axis parameter "A-0-2798 Polarity of master axis position". If the winding axis does not rotate in the desired direction, the parameter P-0-0108 or axis parameter A-0-2798 has to be reset accordingly. When inverting the drive direction of rotation, the control direction (ControlPolarity) is also to be inverted.

Functional diagram of the dancer position controller

The following functional diagram schematically explains the functions of the diameter calculator with dancer.

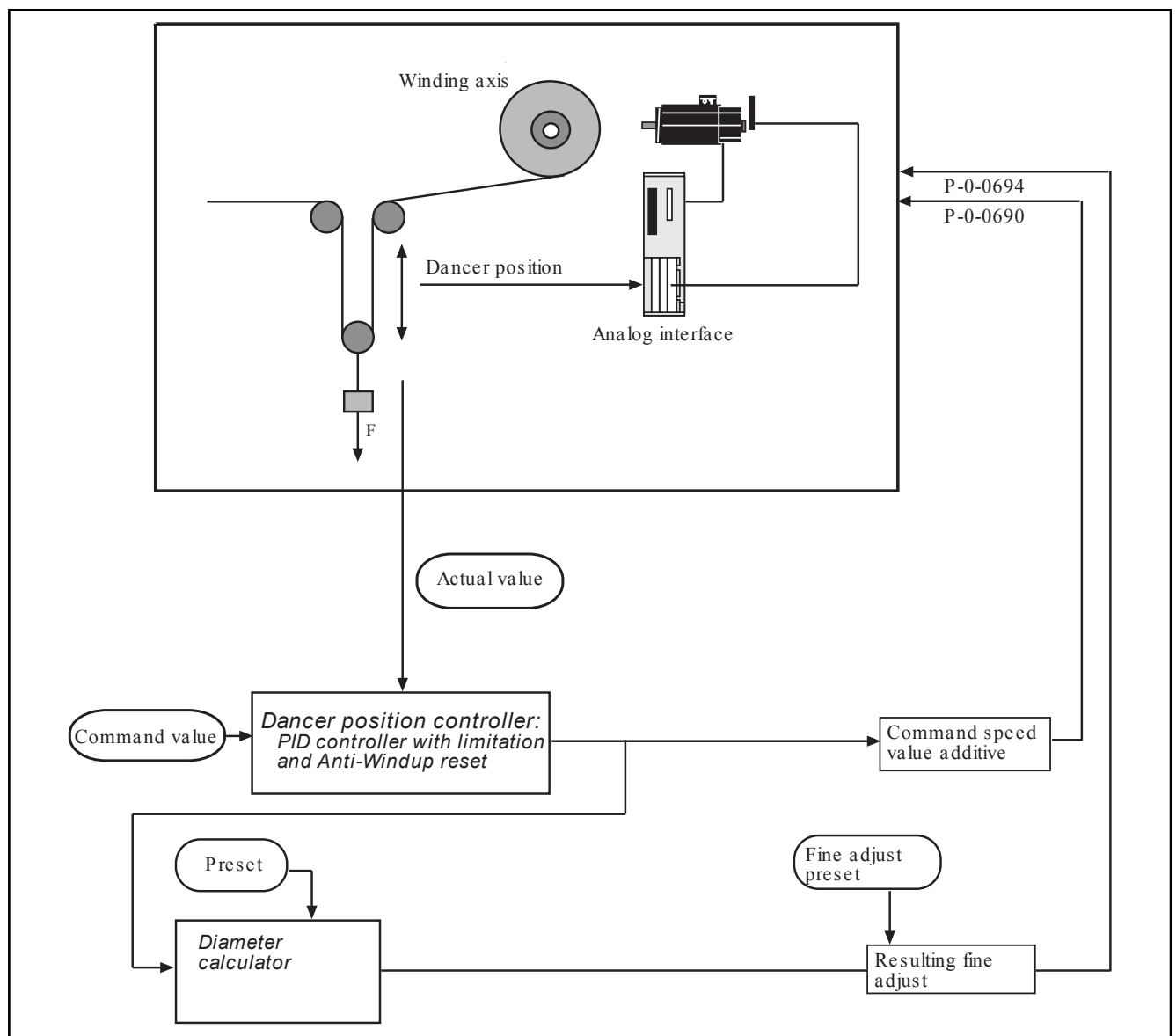


Fig. 8-16: Functional diagram of the diameter calculator with dancer

Error handling

The following table lists and describes the error codes of the function block MB_WinderDancerCtrlType02.

The error codes from the F_RELATED_TABLE (16#0170) are used.

Additionally, the error codes of the function blocks MB_DancerControlType03, MB_DiameterCalculatorType03, MB_DiameterMeasurementType01, IL_PIDType01, RIL_LoopControl, as well as MB_GetCyclicParameterHandle, ML_TechMotionBase, can be output.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1190	16#0020	Error in state machine

Tab. 8-29: Error codes of the FB MB_WinderDancerCtrlType02

Required hardware

- CMLx, XMx, VPx
- IndraDrive

Required firmware

- MLC firmware FWA-CMLXX*-MLC-14VRS

	XX - Device variant, e.g. CML65
Required software	• IndraDrive drive firmware • IndraWorks 14VRS for MLC
Special features for the MLC	The scaling type has to be identical for the master and the slave axis. Rotary and preference scaling are recommended. An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear". The standstill window of the reference axis is to be set as low as possible (e.g. 0.1 rpm). When there is a standstill message of the reference axis, the diameter calculator does not calculate! Interpolation in the drive: The parameters "P-0-0694 Gear ratio fine adjustment, process loop" and Parameter "P-0-0690 Additive velocity command value, process loop" have to be entered into the cyclic Sercos channel of the corresponding drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks. Interpolation in the control: No parameter has to be written to the cyclic Sercos channel of the respective drive. The target variables are the control parameters "A-0-2605 ELS, Gear ratio fine adjustment process controller" and "A-0-2615 ELS, additive velocity command value, process controller".
Special features for the MLD	Interpolation in the drive has to be activated in the drive for the winding axis when using the MLD-S control with a superordinate MLC control (Real axes ► Corresp. axis ► Properties). This setting does not exist when using the MLD-M control. The scaling type has to be identical for master axis and slave axis. Rotary and preference scaling are recommended. The parameters "P-0-0694 Gear ratio fine adjustment, process loop" and "P-0-0690 Additive velocity command value, process loop" have to be entered into the cyclic Sercos channel of the corresponding drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks. The real axes are parameterized in the parameterization mode in IndraWorks (the dialog is located in the project tree below the MLD node → AxisData). The "AxisData structure" of the drive has to be parameterized ("P-0-1367 PLC configuration, bit 6 is set" or dialog "PLC configuration"). An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear". Standstill window of the reference axis: The standstill window of the reference axis is to be set as low as possible (e.g. 0.1 rpm). When there is a standstill message of the reference axis, the diameter calculator does not calculate! For the reference axis, the function block cannot access any value for the standstill window. 1 rpm is assumed internally. If this value is not suitable for the application, modify it using the following PLC code lines:

Declaration

```
prMaStandstillWdw:    POINTER TO REAL; // for StandstillWindow master MLD
rTestMaStandstillWdw: REAL := 0.5;    // for StandstillWindow master MLD
```

Implementation

```
prMaStandstillWdw := ADR(BausteinInstanznameWickler.fbDiameterCalculator.rMaStandstillWindow);
prMaStandstillWdw^ := rTestMaStandstillWdw;
```

The value of this variable is not saved. Thus, it has to be written again when switching on the PLC.

Axis reference for the master axis:

For technology function blocks, the following is recommended as axis reference for the master axis:

- Use axis reference "**VmAxis1**" for a virtual master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 20 (VMA_1).
- Use axis reference "**RmAxis1**" for a real master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 23 (RMA_1).
Additionally, the logic address of the measuring encoder has to be entered in parameter "P-0-1820.0.2 CCD: Source of measuring encoder position feedback value".

For detailed information, refer to Functional Description MLD.

Commissioning the diameter calculator with dancer

To commission the diameter calculator with dancer, refer to [chapter 8.5.4 "Commissioning the diameter calculator with dancer" on page 523](#).

8.5.4 Commissioning the diameter calculator with dancer

Drive control loop

It is assumed that the speed control loop of the winding drive was set. That means that the parameters "S-0-0100 Velocity loop proportional gain" and "S-0-0101 Velocity loop integral action time" have to be set according to the application.

This section describes the commissioning of a diameter calculator with dancer. For this purpose, the following steps are reasonable:

- Parameterization of the dancer position controller
- Parameterization of the diameter calculator
- Initial startup of the winding axis

Commanding the winding axis

The correct operation mode of the drive and the drive enable of the drive have to be set for the winding axis.

There are two ways to set the operation mode and controller enabled:

- Set the "ModeSyncVel" mode using the AxisInterface (recommended)
- Use the PLCopen "MC_Power", "MC_GearIn", "MC_Stop" function blocks to command the axis

Parameterization of the dancer position controller

Since the function of the diameter calculator with dancer is built up on the function of the dancer position controller, the parameterization of the dancer position controller can be performed as described under [Closed-loop dancer position control, page 477](#),,.

Parameterization of the diameter calculator

The individual inputs of the diameter calculator MB_DiameterCalculator-Type03 are assigned as described before. Multiple special aspects have to be taken into account when commissioning the diameter calculator with dancer.

Due to the parameterization of the dancer position controller, the inputs "FormatLength" and "FeedBackDancer" of the diameter calculator are already

preset. Another variable, which affects the dancer, is the dancer constant "Dancer constant". This specifies the web distance with the full displacement of the dancer takes from negative to positive end, based on the unit of dancer adjustment. The "ReferenceDiameter" input should be linked to the currently calculated diameter "CurrentDiameter".

Since the calculated diameter directly affects the velocity of the winding axis via the resulting fine adjustment, a meaningful value has to be preset for the diameter prior to the start of the calculation when activating the diameter calculator. Thus, a diameter value, which is close to the actual diameter, has to be specified at the "PresetDiameter" input. With a rewinder, the value can be slightly lower and slightly higher with an unwinder.

Finally, the inputs "MinDiameter" and "MaxDiameter" can be set. The variables specified act on the outputs "MinAvDiamAchieved" and "MaxAvDiamAchieved". The respective values should be within the admissible diameter range defined by the inputs "HighLimitDiameter" and "LowLimitDiameter". Thus, the time for specific process steps, for example, for a splice or adjustment of the "DiamCalcRange", can be detected and displayed in the PLC, based on the specification of the minimum/maximum diameter.



The ratio "Format length"/"CoreDiameter" may not be greater than 25.

Otherwise, the gear adjustment via the gear fine adjustment is insufficient!

Initial startup of the winding axis

The dancer position controller and the diameter calculator are subsequently activated. After activating the closed-loop dancer position control, the dancer moves to its command position. As described before, this position should correspond to the central position of the dancer.

The phase positions of the winding axis and the reference axis are summed up and, depending on the value of the "DiamCalcRange" input, the diameter is regularly calculated and displayed as current diameter value at the "CurrentDiameter" output. After having activated the diameter calculator, "CurrentDiameter" initially displays the zero value until a diameter has been calculated for the first time. A preset has to be executed if another value, apart from zero, is used. The value "PresetDiameter" is applied as the current diameter value at a positive edge at "Preset".

The period between two calculations depends on the winding axis speed and the value at the "DiamCalcRange" input.

The diameter calculator affects the fine adjustment of the gear. The respective value is displayed at the "GearRatioFineAdjust" output.

8.6 Diameter calculator with open-loop and closed-loop tension control

8.6.1 Introduction and overview

The function block provides two different diameter calculators:

- Diameter calculator with open-loop tension control
 - corresponds to a diameter calculator without sensor
- Diameter calculator with closed-loop tension control
 - corresponds to a diameter calculator with load cell

The diameter calculators are implemented in the MB_WinderTensionCtrl-Type02 function block.

The function block cyclically calculates the current winding diameter, the current moment of inertia of the winder, the corresponding velocity adaptation via the fine adjustment of the gear, an additive speed offset for tensioning the web of fabric and the required torque limit.



When using this winding concept, the mechanical friction (motor coupling, gear...) should be kept as low as possible. The mechanical friction should always be lower than the smallest tension torque that results at the minimum tension and the minimum winding diameter.

Diameter calculator with open-loop tension control

The diameter calculator with tension control is used on center-driven winders. The drive of the winding axis generally operates at its torque limit. The torque limit depends on the current winding diameter and the tension set.

The tension is generated by adjusting the torque of the winding axis. Additional measuring elements are not required.

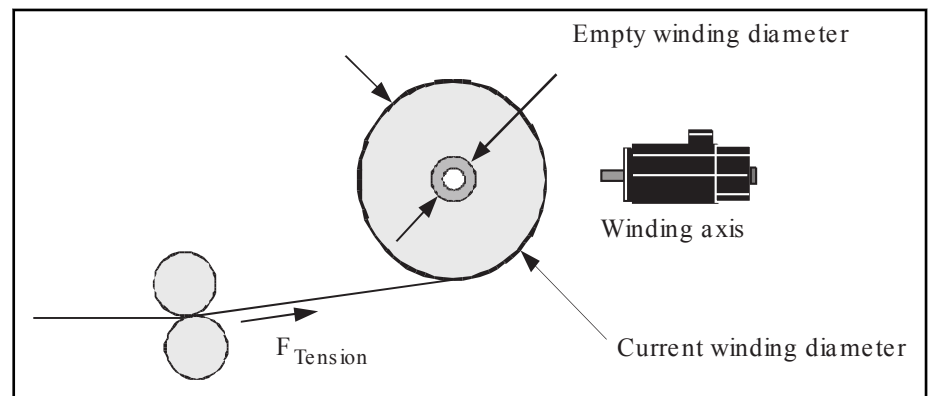


Fig. 8-17: Diameter calculator with open-loop tension control

Diameter calculator with closed-loop tension control

The diameter calculator with closed-loop tension control is used on center-driven winders. The drive of the winding axis generally operates at its torque limit. The torque limit depends on the current winding diameter and the tension set. The torque limit corresponds to the value of the motor torque required to rotate the winder at the defined speed. It consists of load torque, friction torque and acceleration torque.

The tension controller is provided as PI-controller. It contains the function block "IL_PIDType01" from the "RIL_LoopControl" library. The control variable of the controller and the current winding diameter determine the torque limit value of the drive. The command value of the tension should also be scaled and transferred to the controller output ("PreControlWeighting" not equal to zero). When the scaling is not zero, the tension for the torque creation consists of the controller output variable added to the scaled command value. Scaling to zero means that only the tension controller is active. Thus, the tension controller should only be operated additively (e.g. with a scaling of 1.0).

The actual tension value is measured directly using a load cell and transferred to the tension controller as actual value via an analog channel or field

bus. With the web tensioned, the diameter calculator can calculate the current winding diameter.

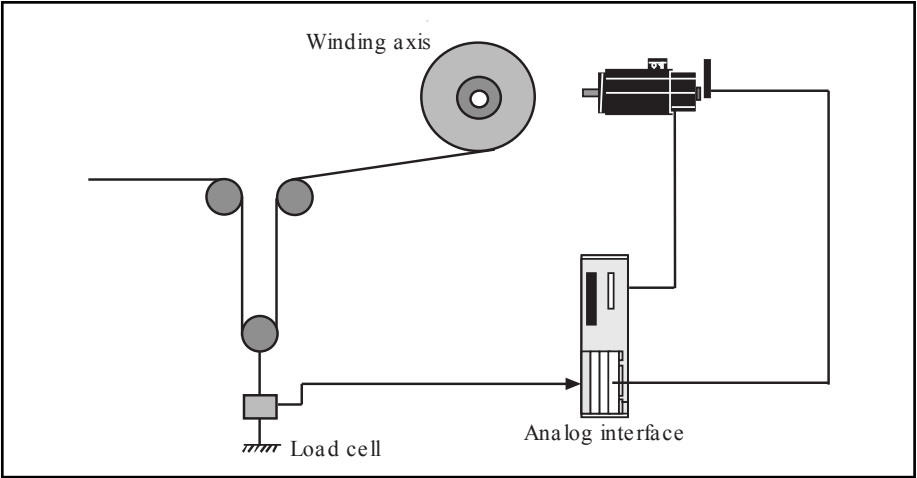


Fig. 8-18: Diameter calculator with closed-loop tension control

8.6.2

MB_WinderTensionCtrlType02

Brief description

The function block cyclically calculates the current winding diameter, the current winder moment of inertia, the corresponding velocity adaptation via the fine adjustment of the gear, an additive speed offset for tensioning the web of fabric and the required torque limit.

Target variables of the function block at the interpolation of the drive are the parameters "P-0-0694 Gear ratio fine adjustment, process loop", "P-0-0690 Additive velocity command value, process loop" and "S-0-0092 Bipolar torque/force limit value". These parameters have to be stored in the cyclic data container.

When interpolating in the control, the target variables are the parameters "A-0-2605 ELS, Gear ratio fine adjustment process controller", "A-0-2615 ELS, additive velocity command value, process controller" and "S-0-0092 Bipolar torque/force limit value".

Only the parameter S-0-0092 has to be stored in the cyclic data container.

Assignment: Target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder
MLD with MPx07/08/17 with MA function package	MX_TechWinder.lib
MLD MPx20 and higher with MA function package	MX_TechWinder

Tab. 8-30: Reference table of the MB_WinderTensionCtrlType02

Interface description

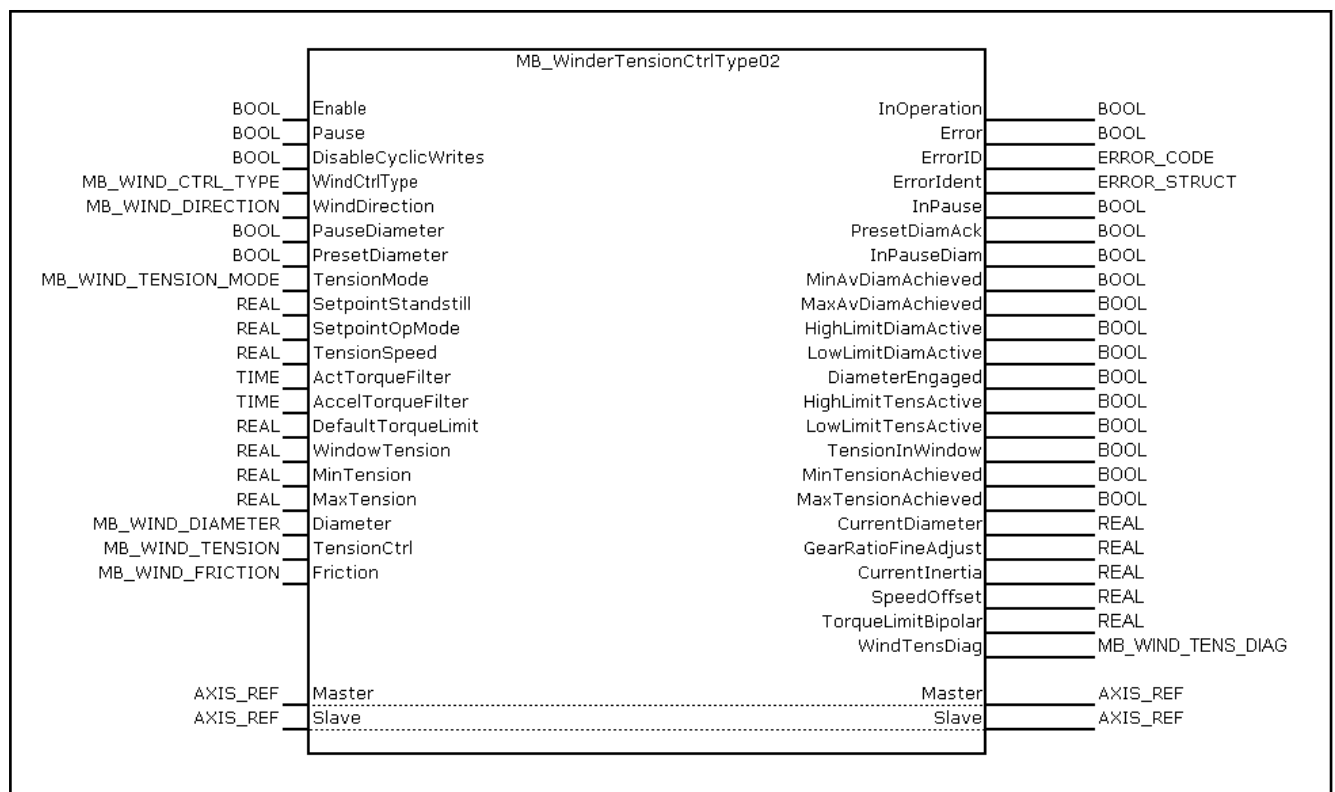


Fig. 8-19: MB_WinderTensionCtrlType02 function block

I/O type	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Function block processing enabled (level-controlled)
	Pause	BOOL	Pausing the tension controller and the diameter calculator (values are frozen)
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output
	WindCtrlType	MB_WIND_CTRL_TYPE	<i>Winding structure</i> - Open-loop tension control (without sensor): OPEN_LOOP (16#0000) - Open-loop tension control (load cell): CLOSED_LOOP (16#0001) MB_WIND_CTRL_TYPE , page 597
	WindDirection	MB_WIND_DIRECTION	<i>Winding direction</i> - Rewinder REWIND (16#0000) - Unwinder UNWIND (16#0001) MB_WIND_DIRECTION , page 600
	PauseDiameter	BOOL	Pausing the diameter calculation Closed-loop/open-loop control of the torque limit remains active
	PresetDiameter	BOOL	CurrentDiameter is set to PresetVal. GearRatioFineAdj is adjusted accordingly (edge-controlled)

I/O type	Name	Type	Comment
	TensionMode	MB_WIND_TENSION_MODE	<i>Mode to tension the material web</i> - Tensioning at constant speed CONST_SPEED - Tensioning at constant web velocity CONST_WEB_VEL MB_WIND_TENSION_MODE , page 601
	SetpointStandstill	REAL	Specification of the standstill tension [N] (Switching between standstill tension and operating tension using the standstill message of the drive)
	SetpointOpMode	REAL	Specification of the operating tension [N] (Switching between standstill tension and operating tension using the standstill message of the drive)
	TensionSpeed	REAL	Tensioning speed at an empty winder [min ⁻¹]
	ActTorqueFilter	TIME	Filter time constant for the actual torque S-0-0084 "Actual torque/force value"
	AccelTorqueFilter	TIME	Filter time constant for acceleration torque
	DefaultTorqueLimit	REAL	Value for the bipolar torque limit after deactivation of "Enable" [%]
	WindowTension	REAL	Window for the actual tension value [%] related to the tension command value (output "TensionInWindow")
	MinTension	REAL	Lower threshold for the actual tension value (only for a display with the binary output "MinTensionAck")
	MaxTension	REAL	Upper threshold for the actual tension value (only for the display with binary output "MaxTensionAck")
	Diameter	MB_WIND_DIAMETER	Structure for diameter calculator MB_WIND_DIAMETER , page 598
	TensionCtrl	MB_WIND_TENSION	Structure for tension controller MB_WIND_TENSION , page 601
	Friction	MB_WIND_FRICTION	Structure for friction torque MB_WIND_FRICTION , page 600
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling, page 537
	InPause	BOOL	Tension controller and diameter calculator pause
	PresetDiamAck	BOOL	Preset diameter was applied (is set as long as "Preset" is pending)
	InPauseDiam	BOOL	Diameter calculator pauses
	MinAvDiamAchieved	BOOL	Minimum diameter reached (refers to the averaged diameter)

I/O type	Name	Type	Comment
	MaxAvDiamAchieved	BOOL	Maximum diameter reached (refers to the averaged diameter)
	HighLimitDiamActive	BOOL	Currently calculated diameter has reached the upper limit (diameter value is discarded)
	LowLimitDiamActive	BOOL	Currently calculated diameter has reached the lower limit (diameter value is discarded)
	DiameterEngaged	BOOL	Diameter "engaged" TRUE: engaged FALSE: Disengaged
	HighLimitTensActive	BOOL	Output of the tension controller reached upper limit
	LowLimitTensActive	BOOL	Output of the tension controller reached lower limit
	TensionInWindow	BOOL	Actual tension value is within the tension window
	MinTensionAchieved	BOOL	Tension below the "MinTension" input
	MaxTensionAchieved	BOOL	Tension is above the input "MaxTension"
	CurrentDiameter	REAL	Currently calculated diameter [mm]
	GearRatioFineAdjust	REAL	Resulting fine adjustment of the gear for speed tracing of the drive [%]
	CurrentInertia	REAL	Current moment of inertia of the winding material + mechanics (related to the motor side [kg m ²])
	SpeedOffset	REAL	Tensioning speed [rpm]
	TorqueLimitBipolar	REAL	Bipolar torque limit [%]
	WindTensDiag	MB_WIND_TENS_DIAG	Current control deviation of the dancer controller MB_WIND_TENS_DIAG , page 601
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to the slave axis (winding axis)

Tab. 8-31: Interface variables of the MB_WinderTensionCtrlType02 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
DisableCyclic-Writes	BOOL			FALSE	Rising edge at "Enable"
WindCtrlType	MB_WIND_CTRL_TYPE				Rising edge at "Enable"
WindDirection	MB_WIND_DIRECTION				Rising edge at "Enable"
PauseDiameter	BOOL			FALSE	Continuous

Name	Type	Min. value	Max. value	Default value	Effective
PresetDiameter	REAL			FALSE	Rising edge at "Preset"
TensionMode	MB_WIND_TENSION_MODE				Rising edge at "Enable"
SetpointStandstill	REAL	0.0	n. def.	0.0	Continuous
SetpointOpMode	REAL	N. def.	N. def.	0.0	Continuous
TensionSpeed	REAL	0.0	n. def.	0.0	Continuous
ActTorqueFilter	TIME	0 s	N. def.	0 s	Continuous
AccelTorqueFilter	TIME	0 s	N. def.	0 s	Continuous
DefaultTorqueLimit	REAL	0.0	N. def.	100.0	Continuous
WindowTension	REAL	0.0	n. def.	0.0	Continuous
MinTension	REAL	0.0	n. def.	0.0	Continuous
MaxTension	REAL	0.0	n. def.	0.0	Continuous
Diameter	MB_WIND_DIAMETER				Depending on the structure element
TensionCtrl	MB_WIND_TENSION				Continuous
Friction	MB_WIND_FRIC-TION				Continuous

Tab. 8-32: Min./max. and default values of the MB_WinderTensionCtrlType02 function block

Signal-time diagram

The following signal-time diagram represents selected function block signals. With a positive edge at the "Enable" input, which is signaled at the "InOperation" output, the function block is operated. Operational readiness can be interrupted by an error acknowledged at the "Error" output. The internal controller process can be paused by the "Pause" input. "Pause" is signaled at the "InPause" output. The "CurrentDiameter" output is configured with a preset value ("PresetVal" input in the structure MB_WIND_DIAMETER) via a positive edge at the "PresetDiameter" input, irrespective of the "Pause" input. However, it is recommended to switch the function block to "Pause". By doing so, the diameter preset can be performed in a more controlled way. The setting of the diameter is displayed at the output "PresetDiamAck".

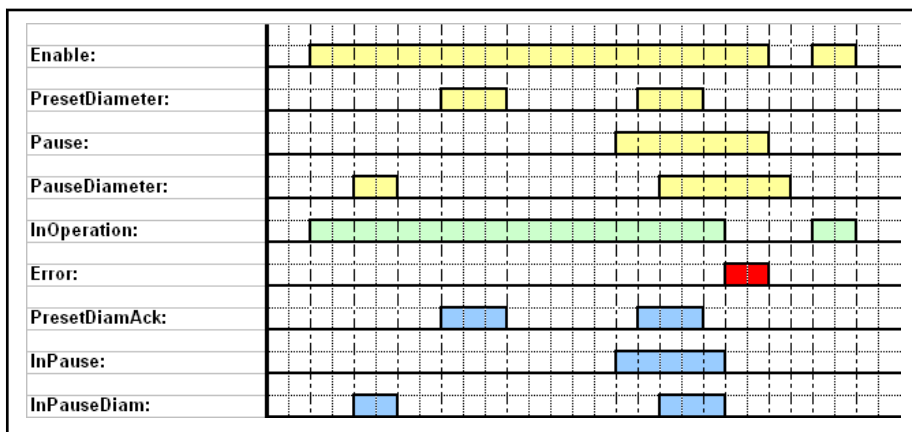


Fig. 8-20: Signal-time diagram of selected inputs and outputs of the function block MB_WinderTensionCtrlType02

Functional Description Diameter calculator with open-loop tension control

The functional scope of the diameter calculator with open-loop tension control includes:

- Tensioning of the untensioned winding material
- Friction compensation depending on the speed
- Compensation of acceleration torque
- Separate parameterization of the operating tension and the standstill tension
- Calculation of diameter-dependent variables

(With the MB_DiameterCalculatorType03 function block)

Functional diagram of the winder calculator with open-loop tension control

The following functional diagram schematically explains the functions of the diameter calculator with open-loop tension control.

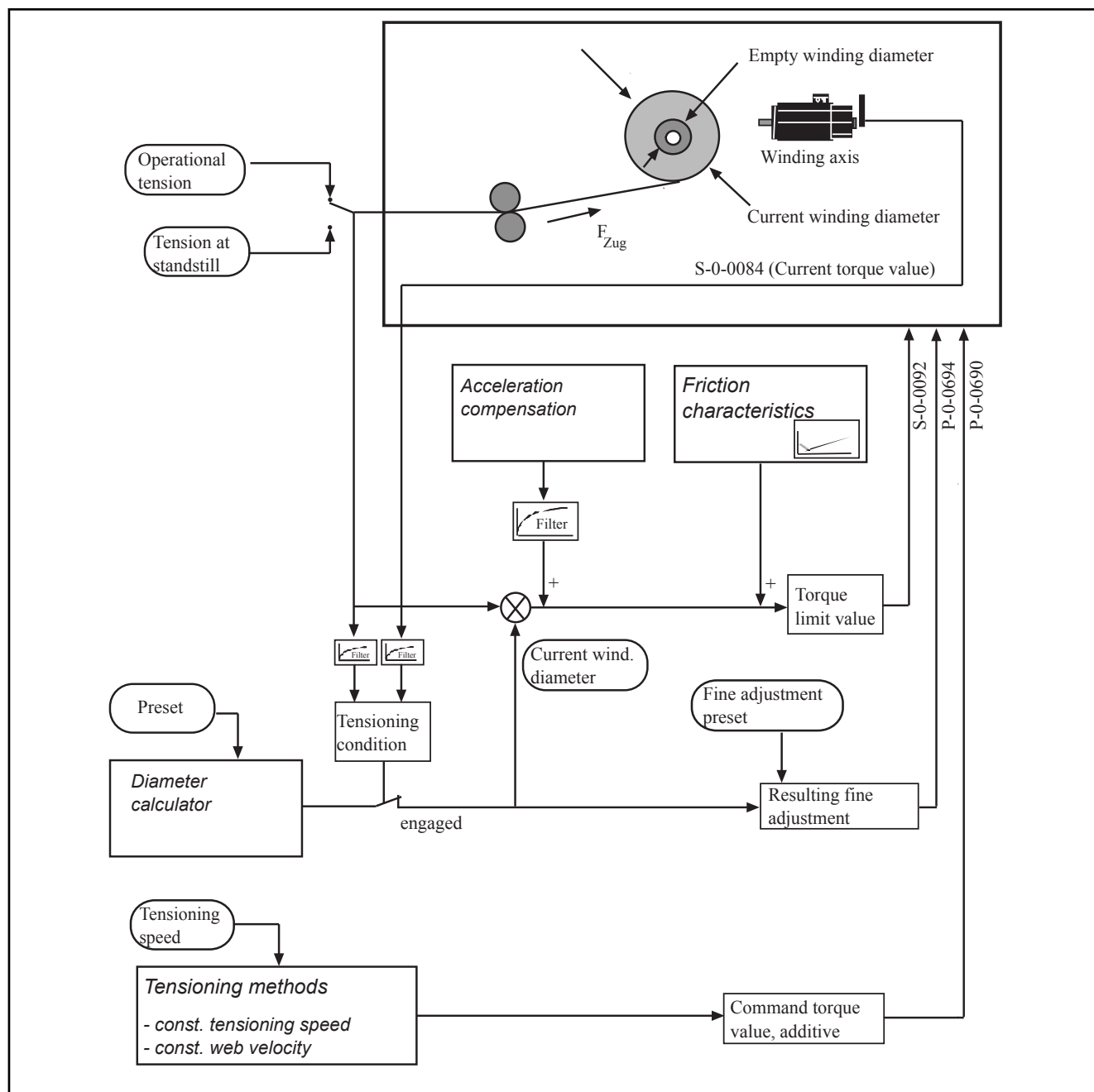


Fig. 8-21: Functional diagram of the winder with open-loop tension control

Diameter calculator with closed-loop tension control

The functional scope of the diameter calculator with closed-loop tension control includes:

- Tensioning of the untensioned winding material
- Friction compensation depending on the speed
- Compensation of the acceleration torque
- Separate parameterization of the operating tension and the standstill tension
- PI controller for tension (via IL_PIDType01) and

- Calculating diameter-dependent variables (via MB_DiameterCalculator-Type03)

Functional diagram of the winder calculator with closed-loop tension control

The following functional diagram schematically explains the functions of the diameter calculator with closed-loop tension control.

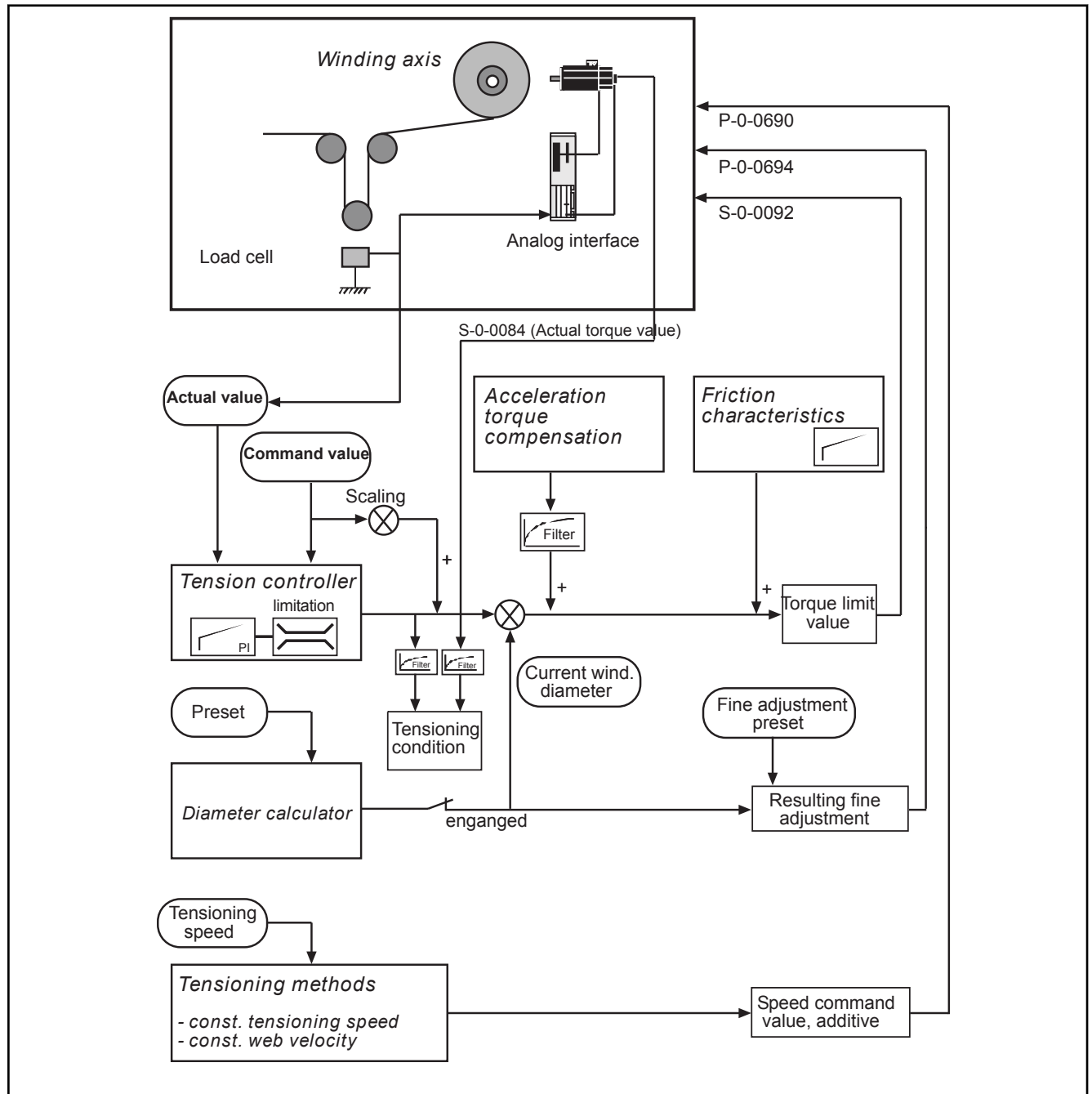


Fig. 8-22: Functional diagram of the winder with closed-loop tension control

Direction of rotation of the winding drive, rewinder/unwinder

With the input of the function block, the winder type is set to a rewinder or unwinder type. The direction of rotation of the drive has to be set either by the scaling parameters or by the polarity of the master drive.



The direction of rotation of the master axis of the winding drive (that is the master axis of the function block MC_GearIn or the master axis of the AxisInterface) must be positive which means a positive speed command value. The winding drive must be inverted using the polarity parameters.

Diameter calculation

The torque for the winding drive results from the product of the tension and the winding radius. Thus, the radius or the diameter of the winder has to be continuously determined. Thus, an internal diameter calculator is integrated in the function block (function block [MB_DiameterCalculatorType03](#) , page 494).

The diameter calculation will only become active when the winding material is tensioned and no standstill message of the reference axis is active. The calculation will be performed cyclically depending on the speed or the position of the winding axis. For diagnostic purposes, the current diameter can be retrieved at the function block.

The calculated winding diameter directly influences the resulting fine adjustment of the gear. If the web tension is too low or negative (formation of loops), the diameter calculator is interrupted since no useful diameter calculation can be performed any longer under these circumstances. If the actual torque of the drive is more than 25% below the command value, the diameter calculator stops calculating (engaging condition).



If the current diameter is to be kept after the control has been switched off, the "CurrentDiameter" output has to be copied to a remanent variable. After switching on, use this value for a preset.



The ratio "Format length"/"CoreDiameter" may not be greater than 25.

Otherwise, the gear adjustment via the gear fine adjustment is insufficient!

Tensioning the winding material

When enabling the winding calculator ("Enable" = TRUE), the winding material is initially tensioned with the tensioning speed. To avoid the formation of loops in the winding material, the master axis should not rotate. The winding material is tensioned if the torque command value of the drive exceeds a pre-defined limit and if the relative deviation between the torque command value and the actual torque value is lower than a defined limit.

Two different procedures can be used for tensioning the winding material:

- Tensioning with constant speed and
- Tensioning with constant web velocity.

When tensioning the winding material with constant speed, the tensioning speed is maintained on a constant level. Thus, the web velocity increases with an increasing winding diameter.

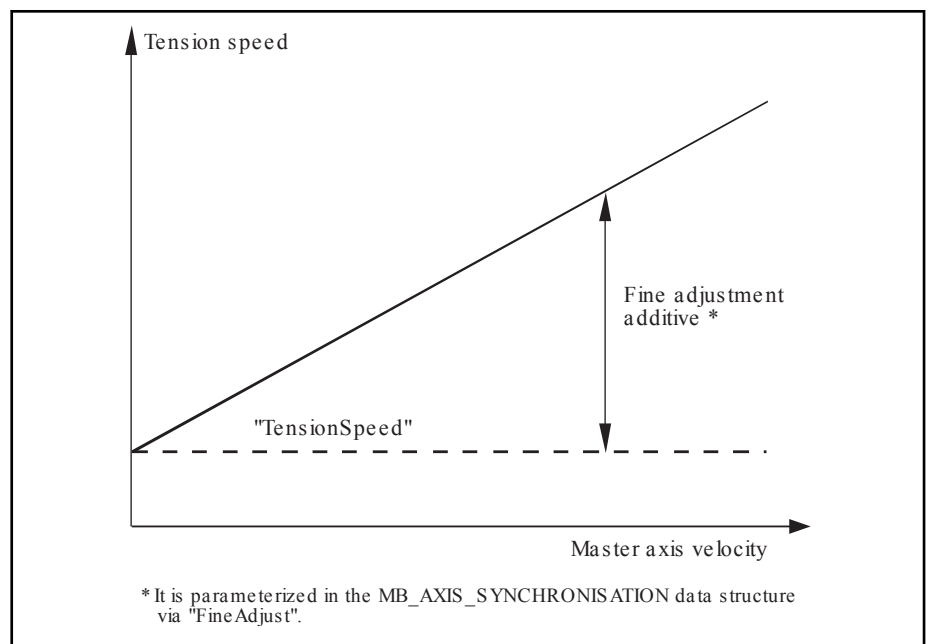


Fig. 8-23: Tensioning at constant tensioning speed

When tensioning the winding material at a constant web velocity, the tensioning speed is reduced reciprocally to the winding diameter. The calculation of the winding axis speed is based on the tensioning speed of an empty winder. This calculation has to be transferred to the function block.

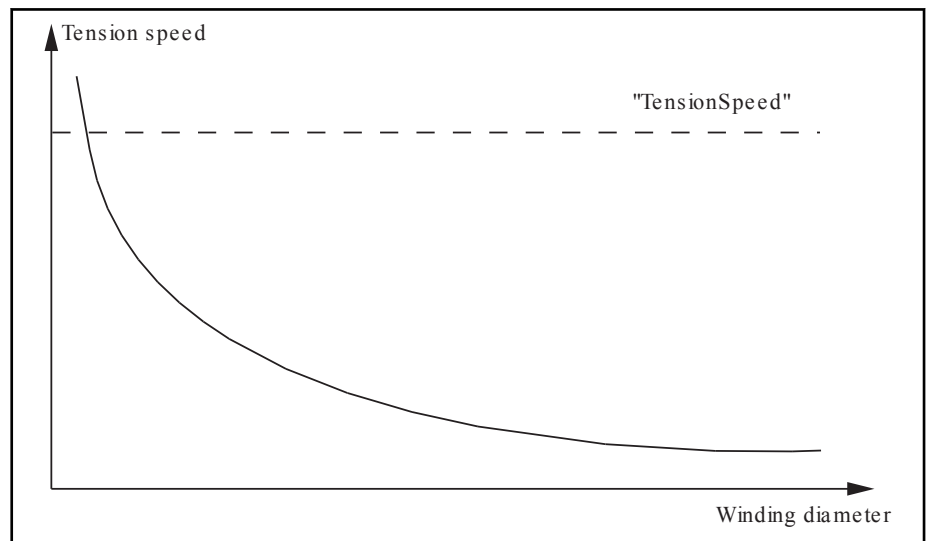


Fig. 8-24: Tensioning with constant web velocity

Friction compensation

Different inputs are used for a friction torque compensation. Approaching the friction behavior is defined as follows:

The friction is initially reduced from the standstill friction torque up to a specific speed. The friction torque increases linearly with the winding axis speed up to the maximum speed ("S-0-0091 Bipolar velocity limit value" or "A-0-0032 Velocity limit value, positive", "A-0-0033 Velocity limit value, negative"). This behavior also applies to negative speeds. The curve is mirrored at the y-axis (friction torque). With this friction torque behavior, the effect of the friction can be counteracted by a compensation.

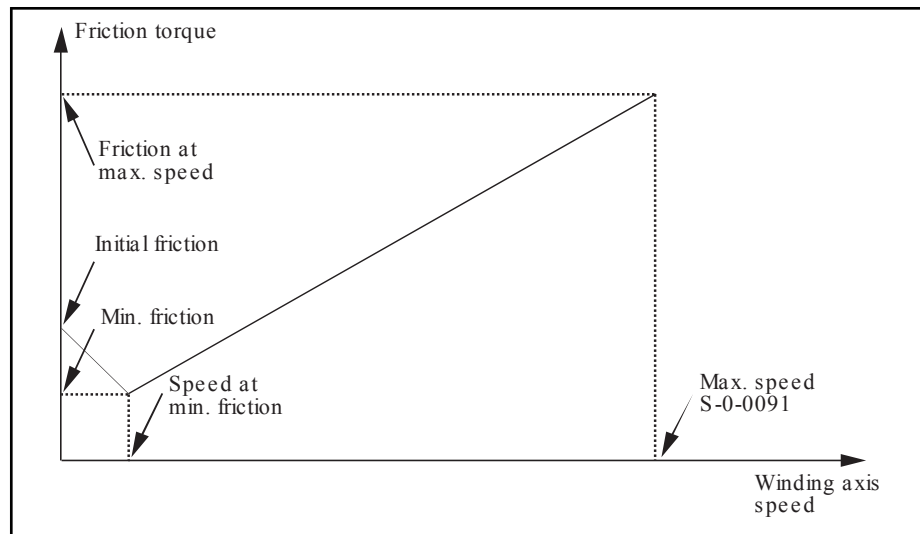


Fig. 8-25: Friction torque compensation

Compensation of acceleration torque

An additional acceleration torque is required for the acceleration of the winding axis. This torque depends on the winding diameter. By specifying certain material and fabric web data, the inertia of the winder is calculated according to the diameter. The calculated acceleration torque can also be filtered.

Splice

After a splice, the winding diameter generally changes abruptly. The last calculated internal value for the winding diameter does no longer correspond to the current diameter. A preset of the winding diameter is required to calculate the diameter correctly. The input value for the preset should correspond approximately to the actual value of the winding diameter.



If the preset is not performed, malfunctions can occur. The preset should therefore be completed before the diameter calculator becomes active again.

The value for the preset should be selected as follows:

- In the case of a rewinder, it has to be slightly smaller or equal to the actual winding diameter and
- in the case of an unwinder, it has to be slightly higher or equal to the actual winding diameter

With a flying splice, it has to be ensured that the circumferential velocity of the new roll corresponds to the current one. This should be implemented respectively in the PLC project of the IndraLogic.

Filtering the actual torque value

At low tensions, an unfiltered actual torque value can result in a disengaging diameter calculator. A filtering process should therefore be performed and the corresponding input should be activated. Therefore, the actual torque and the command torque of the drive are compared (e.g. 20 ms).

Additional leading and lagging of the winding axis

Set an additional leading or lagging via the gear fine adjustment (presetting gear fine adjustment "FineAdjust") using the structure variable "qrAxisCtrl[AxisNumber].Admin.SyncMode.FineAdjust" of the MB_AXIS_SYNCHRONISATION data structure of the "ML_TechInterface" library. A low positive value (e.g. 5%) should be set for a rewinder and a low negative value for an un-

Enabling the correction of the torque constant at synchronous motors

winder. This results in an improved startup reaction for the respective winding direction.

The torque resulting from a motor is determined at the drive using the motor current and a so-called torque constant P-0-0051 "Torque/force constant". However, this so-called "torque or force constant" is a physical but not a static value.

It depends on:

1. the level of the actual motor current flow
(the constant reduces at currents higher than I_{Nom})
2. the temperature of the motor winding and rotor
(the constant reduces with rising temperature)

A corrected torque constant can be used for the **synchronous motor**. This constant can be enabled in the parameter **P-0-0556 "Axis controller configuration" bit 11 (set)**.

Thus, the following influences on the torque/force constant are considered:

1. Magnetic field weakening by magnetic saturation due to increased motor current compared to nominal current.
2. Magnetic field weakening due to increased motor winding temperature.
3. Magnetic field weakening due to increased rotor temperature depending on the average speed.

The torque constant effective in this case can be read in the parameter P-0-0450 "Torque/force constant, current".

A significant deviation from the torque constant P-0-0051 is only caused if the current load increases to 100% (or higher) of the nominal current S-0-0111 "Standstill current motor". That means the higher the motor load, the better the impact of correction. For a motor loaded at low level, only a minimum or no correction results.



A correction of the torque constant is automatically enabled for an asynchronous motor and does thus not need to be set separately in the parameter P-0-0556.

Also refer to the functional description of the respective IndraDrive drive.

Error handling

The following table lists and describes the error codes of the function block MB_WinderTensionCtrlType02.

The error codes from the F_RELATED_TABLE (16#0170) are used.

Additionally, the error codes of the function blocks MB_DiameterCalculatorType03, and IL_PIDType01, RIL_LoopControl as well as MB_GetCyclicParameterHandle, ML_TechMotionBase, can be output.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1120	16#20	Error in state machine
INPUT_RANGE_ERROR	16#1121	16#01	Diameter.MechanicInertia < 0.0
INPUT_RANGE_ERROR	16#1121	16#02	TensionCtrl.PreControlWeighting < 0.0
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

ErrorID	Additional1	Additional2	Text
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	Winding axis not in "BB"
INPUT_RANGE_ERROR	16#1121	16#0D	Incorrect torque scaling (A-0-0050) Scaling for the motor torque has to be based on the motor side
INPUT_RANGE_ERROR	16#1121	16#0E	Axis number of the slave axis incorrect
INPUT_RANGE_ERROR	16#1121	16#0F	Axis number of the master axis incorrect
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	Incorrect drive firmware
RESOURCE_ERROR	16#000F	16#0003	Drive function package "MA" not enabled
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: Error codes of the FB MB_WinderTensionCtrlType02

Required hardware

- CMLx, XMx, VPx
- IndraDrive

Required firmware

- MLC firmware FWA-CMLXX*-MLC-14VRS
XX - Device variant, e.g. CML65
- IndraDrive drive firmware

Required software

- IndraWorks 14VRS for MLC or MLD

Special features for the MLC

The scaling type has to be identical for the master and the slave axis. Rotary and preference scaling are recommended.

An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear".

The standstill window of the reference axis is to be set as low as possible (e.g. 0.1 rpm). When there is a standstill message of the reference axis, the diameter calculator does not calculate!

Interpolation in the drive

The parameters "P-0-0694 Gear ratio fine adjustment, process loop" and Parameter "P-0-0690 Additive velocity command value, process loop" have to be entered into the cyclic Sercos channel of the corresponding drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks.

Interpolation in the control

Special features for the MLD

The target variables are the control parameters "A-0-2605 ELS, Gear ratio fine adjustment process controller" and "A-0-2615 ELS, additive velocity command value, process controller" and "S-0-0092 Bipolar torque/force limit value".

The parameter "S-0-0092 Bipolar torque/force limit value" has to be entered into the cyclic Sercos channel of the corresponding drive.

Interpolation in the drive has to be activated in the drive for the winding axis when using the MLD-S control with a superordinate MLC control (Real axes **Corresp. axis ► Properties**). This setting does not exist when using the MLD-M control.

The scaling type has to be identical for master axis and slave axis. Rotary and preference scaling are recommended.

The parameters "P-0-0694 Gear ratio fine adjustment, process loop" and "P-0-0690 Additive velocity command value, process loop" and "S-0-0092 Bipolar torque/force limit value" have to be entered into the cyclic Sercos channel of the corresponding drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks.

The real axes are parameterized in the parameterization mode in IndraWorks (the dialog is located in the project tree below the MLD node **AxisData**).

The "AxisData structure" of the drive has to be parameterized ("P-0-1367 PLC configuration, bit 6 is set" or dialog "PLC configuration").

An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear". The parameter "S-0-0091 Bipolar velocity limit value" should be set in accordance with the maximum speed of the winding drive.

The standstill window of the reference axis is to be set as low as possible (e.g. 0.1 rpm). When there is a standstill message of the reference axis, the diameter calculator does not calculate!

Standstill window of the reference axis:

For the reference axis, the function block cannot access any value for the standstill window. 1 rpm is assumed internally. If this value is not suitable for the application, modify it using the following PLC code lines:

Declaration

```
prMaStandstillWdw:    POINTER TO REAL; // for StandstillWindow master MLD
rTestMaStandstillWdw: REAL := 0.5;    // for StandstillWindow master MLD
```

Implementation

```
prMaStandstillWdw := ADR(BausteinInstanznameWickler.fbDiameterCalculator.rMaStandstillWindow);
prMaStandstillWdw^ := rTestMaStandstillWdw;
```

The value of this variable is not saved. Thus, it has to be written again when switching on the PLC.

Axis reference for the master axis:

For technology function blocks, the following is recommended as axis reference for the master axis:

- Use axis reference "**VmAxis1**" for a virtual master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 20 (VMA_1).
- Use axis reference "**RmAxis1**" for a real master axis.

The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 23 (RMA_1).

Additionally, the logic address of the measuring encoder has to be entered in parameter "P-0-1820.0.2 CCD: Source of measuring encoder position feedback value".

For detailed information, refer to Functional Description MLD.

Commissioning the winder calculator with open-loop or closed-loop tension control

To commission the winder calculator with open-loop or closed-loop tension control, refer to [chapter 8.6.3 "Commissioning the diameter calculator with open-loop and closed-loop tension control" on page 540](#).

8.6.3 Commissioning the diameter calculator with open-loop and closed-loop tension control

Drive control loop

It is assumed that the speed control loop of the winding drive was set. That means that the parameters "S-0-0100 Velocity loop proportional gain" and "S-0-0101 Velocity loop integral action time" have to be set according to the application.

The commissioning of the diameter calculator is described in seven steps in the following:

1. Determining the winder type
2. Determining the direction of rotation of the winding axis
3. Determining the friction torque at standstill
4. Determining the friction torque at maximum speed
5. Determining the minimum friction torque
6. Final settings
7. Initial startup of the winding axis

Commanding the winding axis

The correct operation mode of the drive and the drive enable of the drive have to be set for the winding axis.

There are two ways to set the operation mode and controller enabled:

- Set the "ModeSyncVel" mode using the AxisInterface (recommended)
- Use the PLCopen "MC_Power", "MC_GearIn", "MC_Stop" function blocks to command the axis

Determining the winder type

First, it has to be determined which winder type is available or is to be used. Subsequently, the winding direction is defined at the "WindDirection" input (TRUE: Unwinding, FALSE: Rewind). Furthermore, the diameter calculator is selected with the "WindCtrlType" input (TRUE: Diameter calculator with closed-loop tension control, FALSE: Diameter calculator with open-loop tension control).

Determining the direction of rotation of the winding axis

Set the direction of rotation of the tension or winding axis using the drive parameter "P-0-0108 Master drive polarity" or the axis parameter "A-0-2798 Polarity of master axis position". If the winding axis does not rotate in the desired direction, the parameter P-0-0108 or axis parameter A-0-2798 has to be reset accordingly. The polarity of the master axis position is only checked when enabling the function block ("Enable" = TRUE). Thus, modifying the mentioned parameters during the operation of the function block does not have any effect.

Alternatively, the direction of rotation can be changed via the scaling type of the winding drive. This can be done by switching the sign of the position/

acceleration data and the torque/force data (parameters S-0-0055, A-0-0029, S-0-0043, S-0-0085).

The following figure shows the principal relations between rewinders and unwinders.

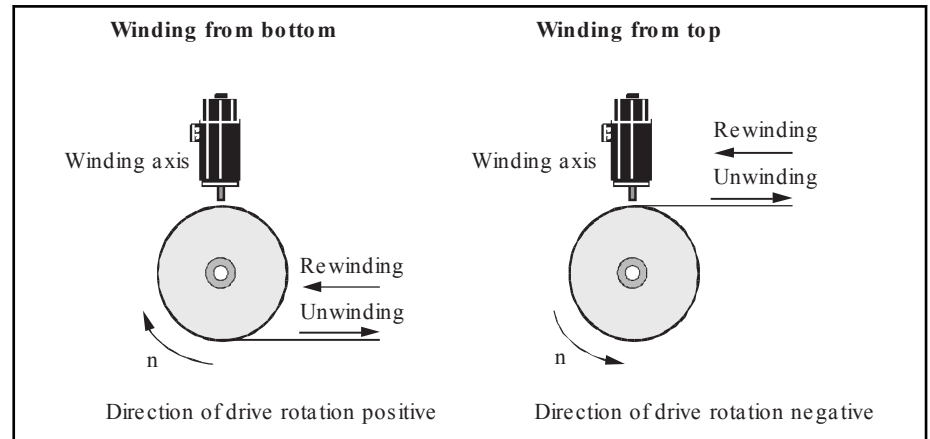


Fig. 8-26: Principles of rewinders and unwinders

Determining the friction torque at standstill

1. The "friction torque at standstill" ("StandstillTorque" variable in the data structure MB_WIND_TENSION) is determined when the master axis is at standstill and the winder is empty (core only). The winding function block is disabled. For the winding axis (slave axis), select the operation mode "Velocity control" (in the AxisInterface of the ML_TechInterface "MODE_VEL" library). The axis can also be controlled using the PLCopen function block MC_MoveVelocity. The command value can for example be set to 10 rpm.
2. The value of parameter "P-0-0109 Torque/force peak limit" is set to zero. The axis has to stop now. The value for this parameter is incremented until the drive starts to rotate. The last value corresponds to the friction torque at standstill.
3. Axis enabled "MODE_VEL" is canceled (alternatively use the PLCopen function block "MC_Stop").

The parameter "P-0-0109 Torque/force peak limit" is reset to its previous value (e.g. 400 %).

The friction torque setting at standstill is thus completed.

Determining the friction torque at maximum speed

1. The "friction torque at maximum speed" ("MaxTorque" variable in the data structure MB_WIND_TENSION) is determined when the master axis is at standstill and with an empty winder (core only). The function block is disabled.
2. For the winding axis (slave axis), select the operation mode "Velocity control" (in the AxisInterface of the ML_TechInterface "MODE_VEL", optionally PLCopen function block).
Increase the command value for the selected operation mode incrementally up to the value of the parameter "S-0-0091 Bipolar velocity limit value". The value for the parameter "S-0-0091 Bipolar velocity limit value" has to be entered according to the application and motor data.
3. In the next step, the value of the parameter "P-0-0109 Torque/force peak limit" is decreased incrementally (e.g. from 100 % towards 0 %).

until the drive speed also starts to decrease. The speed of the drive can be traced using the drive parameter "S-0-0040 Velocity feedback value of encoder 1" or the axis parameter "A-0-0102 Actual velocity value". The last value of "P-0-0109 Torque/force peak limit" corresponds to the maximum friction torque during maximum speed.

4. Axis enabled "MODE_VEL" is canceled (alternatively use the PLCopen function block "MC_Stop").

The parameter "P-0-0109 Torque/force peak limit" is reset to its previous value (e.g. 400 %).

Setting the friction torque at maximum speed "MaxTorque" is thus completed.

Determining the minimum friction torque

The minimum friction torque is determined using the deceleration test with an empty winder. The winding drive is decelerated from a constant velocity (e.g. 50 min⁻¹) to standstill. At the same time, the drive parameters "S-0-0080 Torque/force command value" (axis parameter "A-0-0038 Torque/force limit value, bipolar") and "S-0-0040 Velocity feedback value of encoder 1" (axis parameter "A-0-0102 Actual velocity value") are recorded with an oscilloscope or with the trace function in IndraWorks.

The minimum value is read from the torque curve and can be transferred to the structure MB_WIND_FRICTION as "MinTorque". The corresponding speed is entered as "MinTorqueVelocity" in MB_WIND_FRICTION.

Final settings

Finally, the remaining input variables are specified in the following table.

Input variable	Note/setting
SetpointOpMode	Setting the desired operating tension
SetpointStandstill	Setting the standstill tension
TensionMode	Selecting the tensioning speed characteristics: • TRUE: Tensioning with constant web velocity • FALSE: Tensioning at constant tensioning speed
Offset in fine adjustment	A fine adjustment presetting can be carried out in the data structure MB_AXIS_SYNCHRONISATION from the library ML_TechInterface. Therefore, the "FineAdjust" variable is used. Its value depends on the winding direction. This variable is cyclically processed if "EnableCyclicScanning" is set to TRUE. As described before, the winding direction is determined with "WindDirection". For an unwinder, a positive value is used for "FineAdjust". For a rewinder, a negative value is used. These values should be ≤ 5 %

Input variable	Note/setting
ActTorqueFilter	If the actual torque is to be filtered (as measured value), a value not equal to zero has to be applied at this input of the function block as a time constant. This value acts on the default parameter "S-0-0084 Torque/force feedback value"
AccelTorqueFilter	If the acceleration torque is to be filtered, a value unequal to zero has to be applied at this input of the function block as time constant

Tab. 8-34: Final determination of the remaining input variables

Initial startup of the winding axis

The winding material is placed in an untensioned state (slightly sagging). The winding axis has to be commanded either via the AxisInterface or via the PLCOpen function blocks. The diameter calculator is activated by setting the "Enable" input. The winding material should now be tensioned with the predefined tensioning speed ("TensionSpeed" input). The "SpeedOffset" output provides the tensioning speed.

The process variable window "WindowTension" can be set to a value from 10 - 20 %.

The master axis is enabled with a comparatively low speed (e. g. 10 min⁻¹). After some revolutions, the "DiameterEngaged" output has to be set. It is thus displayed that the diameter calculation is active ("engaged"). This depends on the web tension or the actual torque at the winding axis and the torque limit (drive parameter "S-0-0092 Bipolar torque/force limit value" or axis parameter "A-0-0038 Torque/force limit value, bipolar"). The torque/force limit is displayed at the "TorqueLimitBipolar" output.

When the diameter calculator is engaged, the standstill tension and the operating tension can be checked. Therefore, the value "CurrentTension" from the data structure ML_WIND_TENS_DIAG is monitored. In this data structure, the current torque limit value "TorqueLimit", the current acceleration torque "Acceleration torque" and the currently applied friction torque "Friction-Torque" can also be monitored.

The function block provides different process variables at its outputs. The calculated current diameter of the winder ("CurrentDiameter" output) should correspond to the actual diameter of the winder. Slight deviations can be disregarded. The gear fine adjustment for the speed tracing of the drive axis of the winder (output "GearRatioFineAdjust"), the current moment of inertia of the winder ("CurrentInertia" output) and the tensioning speed ("SpeedOffset" output) are output.

Monitoring torque limit values

At the initial control startup, a drive warning "E2056 Torque limit value = 0" can be output if the winder function block is not yet enabled.

In order to prevent this, write a value unequal to zero (e.g. 100 %) to the respective cyclic data container during the startup phase of the PLC program (e.g. warm start). The value of the parameter S-0-0092 "Bipolar torque/force" is written to the mentioned data container.

If the warning "E8260 Torque - force command value limit active" is output when the function block is enabled, clear this warning by setting bit 4 of the parameter P-0-0556, "Axis controller configuration".

In rare cases (e.g. multiple F8078 "Error in the speed control loop") it might be necessary to disable the velocity controller monitoring (set bit 1 of the parameter P-0-0556 "Axis controller configuration").

8.7 Diameter calculator with closed-loop tension control and speed correction

8.7.1 Introduction and overview

The diameter calculator with closed-loop tension control and speed correction is used on center-driven winders. The drive of the winding axis generally DOES NOT operates at its torque limit. Due to safety reasons, a torque limit is calculated that can additionally be set (the limit should be above the required motor torque and should not limit the drive in normal operation).

This winder is mainly used for elastic, expandable material.

The actual tension value is measured directly using a load cell and transferred to the tension controller as actual value via an analog channel or field bus. With the web tensioned, the diameter calculator can calculate the current winding diameter.

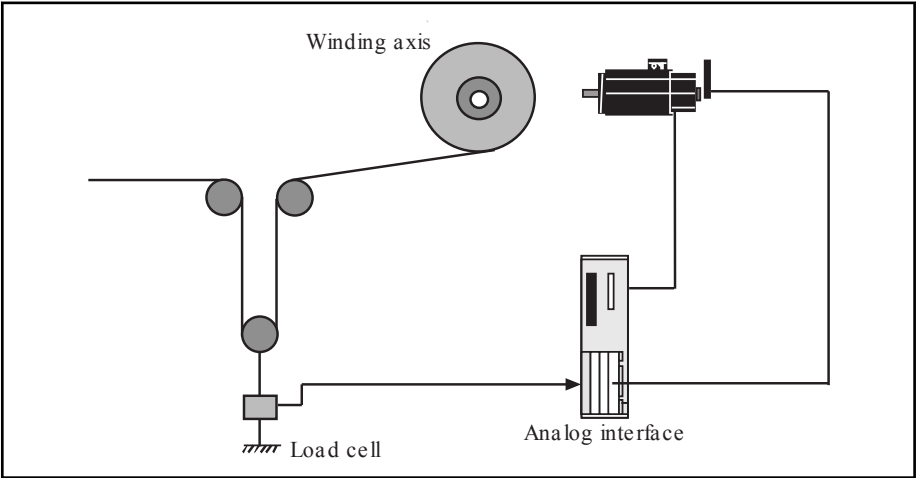


Fig. 8-27: Diameter calculator with closed-loop tension control

8.7.2 MB_WinderTensionCtrlSpeedType01

Brief description The function block cyclically calculates the current winding diameter, the current winder moment of inertia, the corresponding velocity adaptation via fine adjustment of the gear, an additive speed offset for tensioning the web and a parameterizable torque limit.

Target variables of the function block are the parameters "P-0-0694 Gear ratio fine adjustment, process loop", "P-0-0690 Additive velocity command value, process loop", S-0-0081 Additive torque/force command value and "S-0-0092 Bipolar torque/force limit value".

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder
MLD MPx20 and higher with MA function package	MX_TechWinder

Tab. 8-35: Reference table of the MB_WinderTensionCtrlSpeedType01

Interface description

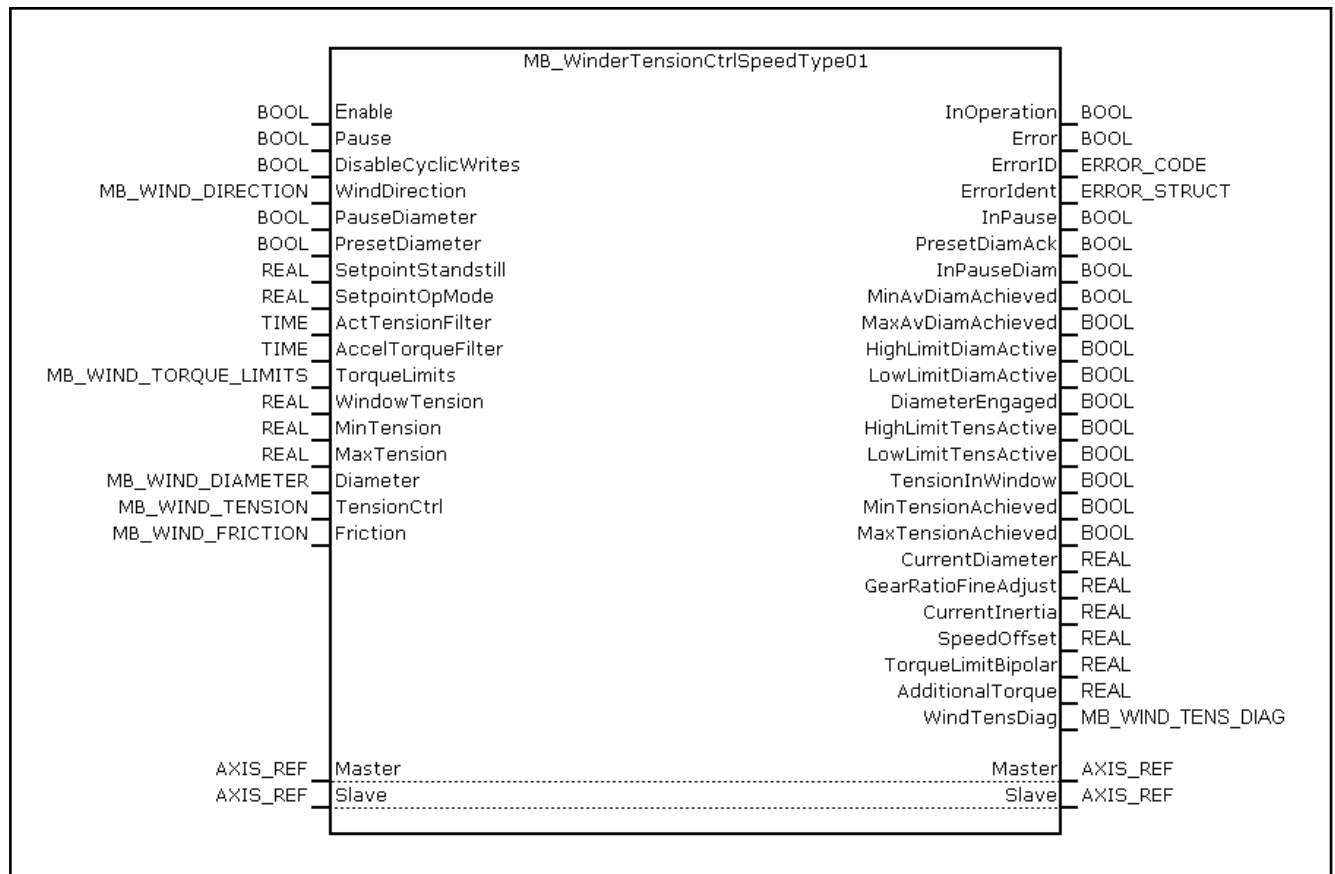


Fig. 8-28: MB_WinderTensionCtrlSpeedType01 function block

I/O type	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Function block processing enabled (level-controlled)
	Pause	BOOL	Pausing the tension controller and the diameter calculator (values are frozen)
	DisableCyclicWrites	BOOL	If this input is set, the cyclic data is not written directly to the optional cyclic data container, but only displayed at the output
	WindDirection	MB_WIND_DIRECTION	<i>Winding direction</i> - Rewinder REWIND (16#0000) - Unwinder UNWIND (16#0001) MB_WIND_DIRECTION , page 600
	PauseDiameter	BOOL	Pausing the diameter calculation.
	PresetDiameter	BOOL	CurrentDiameter is set to PresetVal, GearRatioFineAdjust is adjusted accordingly (edge-controlled)
	SetpointStandstill	REAL	Specification of the standstill tension [N] (Switching between standstill tension and operating tension using the standstill message of the drive)

I/O type	Name	Type	Comment
	SetpointOpMode	REAL	Specification of the operating tension [N] (Switching between standstill tension and operating tension using the standstill message of the drive)
	ActTensionFilter	TIME	Filter time constant for actual tension control
	AccelTorqueFilter	TIME	Filter time constant for acceleration torque
	TorqueLimits	MB_WIND_TORQUE_LIMITS	Structure for bipolar torque limit (S-0-0092) MB_WIND_TORQUE_LIMITS, page 602
	WindowTension	REAL	Window for the actual tension control value [%] in relation to the tension command value ("TensionInWindow")
	MinTension	REAL	Lower threshold for the actual tension value (only for a display with the binary output "MinTensionAck")
	MaxTension	REAL	Upper threshold for the actual tension control value (only for a display with binary output "MaxTensionAck")
	Diameter	MB_WIND_DIAMETER	Structure for diameter calculator MB_WIND_DIAMETER, page 598
	TensionCtrl	MB_WIND_TENSION	Structure for tension controller MB_WIND_TENSION, page 601
	Friction	MB_WIND_FRICTION	Structure for friction torque MB_WIND_FRICTION, page 600
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling, page 553
	InPause	BOOL	Tension controller and diameter calculator pause
	PresetDiamAck	BOOL	Preset diameter was applied (is set as long as "Preset" is pending)
	InPauseDiam	BOOL	Diameter calculator pauses
	MinAvDiamAck	BOOL	Minimum diameter reached (refers to the averaged diameter)
	MaxAvDiamAck	BOOL	Maximum diameter reached (refers to the averaged diameter)
	HighLimitDiamAck	BOOL	Currently calculated diameter has reached the upper limit (diameter value is discarded)
	LowLimitDiamAck	BOOL	Currently calculated diameter has reached the lower limit (diameter value is discarded)
	DiameterEngaged	BOOL	Diameter "engaged" TRUE: engaged FALSE: Disengaged

I/O type	Name	Type	Comment
	HighLimitTensActive	BOOL	Output of the tension controller reached upper limit
	LowLimitTensActive	BOOL	Output of the tension controller reached lower limit
	TensionInWindow	BOOL	Actual tension value is within the tension window
	MinTensionAchieved	BOOL	Tension below the "MinTension" input
	MaxTensionAchieved	BOOL	Tension is above the input "MaxTension"
	CurrentDiameter	REAL	Currently calculated diameter [mm]
	GearRatioFineAdjust	REAL	Resulting fine adjustment of the gear for speed tracing of the drive [%]
	CurrentInertia	REAL	Current total moment of inertia (related to the motor side [kg/m ²])
	SpeedOffset	REAL	Tensioning speed [rpm]
	TorqueLimitBipolar	REAL	Bipolar torque limit [%]
	AdditionalTorque	REAL	Additive torque [%]
	WindTensDiag	MB_WIND_TENS_DIAG	Current control deviation of the dancer controller MB_WIND_TENS_DIAG , page 601
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to the slave axis (winding axis)

Tab. 8-36: Interface variables of the MB_WinderTensionCtrlSpeedType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
DisableCyclic-Writes	BOOL			FALSE	Rising edge at "Enable"
WindDirection	MB_WIND_DIRECTION			FALSE	Rising edge at "Enable"
PauseDiameter	BOOL			FALSE	Continuous
PresetDiameter	BOOL			FALSE	Continuous
SetpointStandstill	REAL	0.0	n. def.	0.0	Continuous
SetpointOpMode	REAL	0.0	n. def.	0.0	Continuous
ActTensionFilter	TIME	0 s	n. def.	0 s	Continuous
AccelTorqueFilter	TIME	0 s	n. def.	0 s	Continuous
TorqueLimits	MB_WIND_TORQUE_LIMITS				Continuous
WindowTension	REAL	0.0	n. def.	0.0	Continuous

Name	Type	Min. value	Max. value	Default value	Effective
MinTension	REAL	0.0	n. def.	0.0	Continuous
MaxTension	REAL	0.0	n. def.	0.0	Continuous
Diameter	MB_WIND_DIAMETER				Depending on the structure element
TensionCtrl	MB_WIND_TENSION				Continuous
Friction	MB_WIND_FRIC-TION				Continuous

Tab. 8-37: Min./max. and default values of the MB_WinderTensionCtrlSpeedType01 inputs

Signal-time diagram

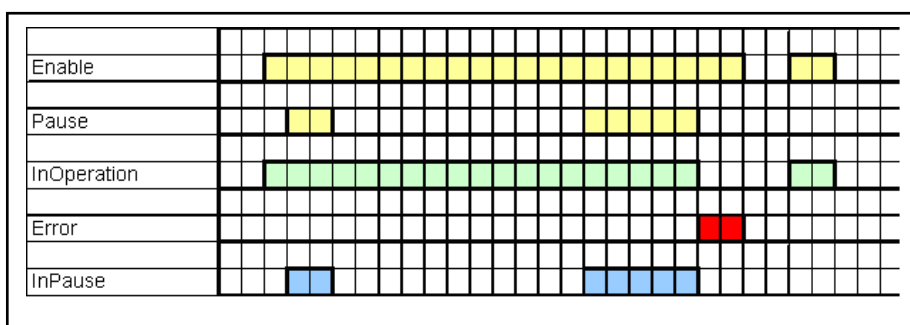


Fig. 8-29: Signal-time diagram of MB_WinderTensionCtrlSpeedType01

Functional description

The function block is a diameter calculator with open-loop tension control that usually does not operate at the torque limit.

The function block cyclically calculates the current winding diameter, the current moment of inertia of the winder, the corresponding velocity adaptation via the fine adjustment of the gear, an additive speed offset for tensioning the web of fabric and the command torque limit value for precontrol.

The MB_DiameterCalculatorType03 and MB_DiameterMeasurementType01 function blocks are used internally.

Target variables of the function block are the parameters "P-0-0694 Gear ratio fine adjustment, process loop", "P-0-0690 Additive velocity command value, process loop", "S-0-0081 Additive torque/force command value" and "S-0-0092 Bipolar torque/force limit value".

The velocity is adapted via the gear fine adjustment due to the changing winding diameter. The tension controller affects the velocity command value in an additive manner. The calculated torque (tension torque, friction torque, acceleration torque) is precontrolled via the additive torque command value.



An external activation of the input "TensionCtrl.PControl" can adjust the Kp-gain of the tension controller.

e.g. diameter-dependent: If the diameter increases, the Kp-gain decreases (multiplication with the ratio "MinRefDiameter"/"CurrentDiameter")

e.g. material-dependent: The more rigid the winding material, the lower the Kp

Due to safety reasons, the torque limit value on the currently calculated torque command value is multiplied with a scaling factor and is set with an additional additive value. In the normal winder mode, the torque limitation should

not be effective and the winding drive should be operated in a speed-controlled way.

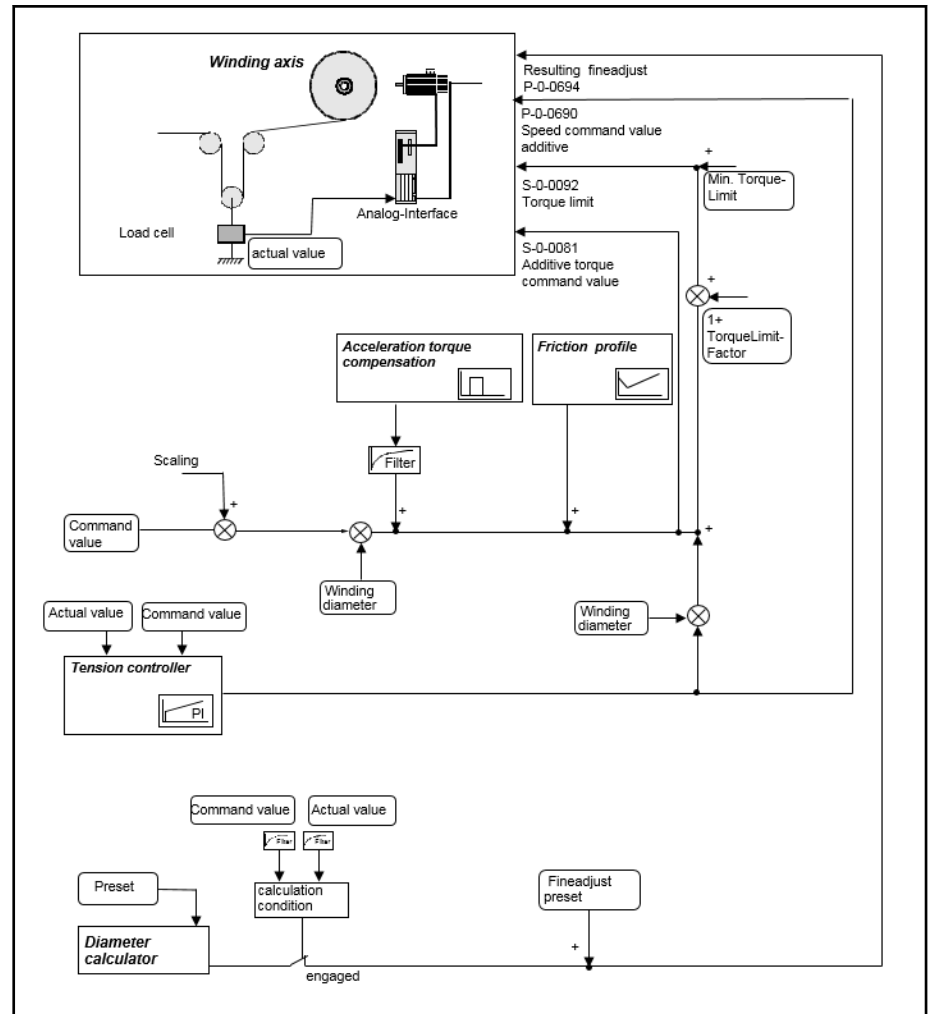


Fig. 8-30: Schematic representation of the control for a winding axis

Direction of rotation of the winding drive, rewriter/unwinder

With the enumeration "MB_WindDirection" of the function block, the winder type is set to a rewriter or unwinder type. The direction of rotation of the drive has to be set either by the scaling parameters or by the polarity of the master drive.



The direction of rotation of the master axis of the winding drive (that is the master axis of the function block MC_GearIn or the master axis of the AxisInterface) must be positive which means a positive speed command value. The winding drive must be inverted using the polarity parameters.

Diameter calculation

The torque for the winding drive results from the product of the tension and the winding radius. Thus, the radius or the diameter of the winder has to be continuously determined. Thus, an internal diameter calculator is integrated in the function block ([MB_DiameterCalculatorType03](#) , page 494).

The diameter calculation will only become active when the winding material is tensioned and no standstill message of the reference axis is active. The cal-

ulation will be performed cyclically depending on the speed or the position of the winding axis. For diagnostic purposes, the current diameter can be retrieved at the function block.

The calculated winding diameter directly influences the resulting fine adjustment of the gear. If the web tension is too low or negative (formation of loops), the diameter calculator is interrupted since no useful diameter calculation can be performed any longer under these circumstances.

If the actual tension control of the drive is more than 25 % below the command value, the diameter calculator stops calculating (**engaging condition**).



If the current diameter is to be kept after the control has been switched off, the "CurrentDiameter" output has to be copied to a remanent variable. After switching on, use this value for a preset.



The ratio "Format length"/"CoreDiameter" may not be greater than 25.

Otherwise, the gear adjustment via the gear fine adjustment is insufficient!

Affecting the engaging condition

In some cases (e.g. strong fluctuations in the actual tension control), the engaging condition (actual value force is 25 % below the command tension) can be affected. Thus, the value for a respective internal variable is changed using a pointer access.

Declaration

```
prTensionPercent:    POINTER TO REAL; // for tension condition
rTestTensionPercent: REAL := -0.5;    // example for tension condition 50%
```

Implementation

```
prTensionPercent := ADR(BausteinInstanznameWickler.fbDiameterCalculator.rCheckTensionPercent);
prTensionPercent^ := rTestTensionPercent;
```

Tensioning the winding material

When enabling the winding calculator ("Enable" = TRUE), the winding material is initially tensioned with the additive speed offset. To avoid the formation of loops in the winding material, the master axis should not rotate. The winding material is tensioned if the tension command value of the tension controller exceeds a defined minimum value and if the relative deviation between the tension value and the actual torque value is lower than a defined limit.

Friction compensation

Different inputs are used for a friction torque compensation. Approaching the friction behavior is defined as follows:

The friction is initially reduced from the standstill friction torque up to a specific speed. The friction torque increases linearly with the winding axis speed up to the maximum speed ("S-0-0091 Bipolar velocity limit value" or "A-0-0032 Velocity limit value, positive", "A-0-0033 Velocity limit value, negative").

This behavior also applies to negative speeds. The curve is mirrored at the y-axis (friction torque). With this friction torque behavior, the effect of the friction can be counteracted by a compensation.

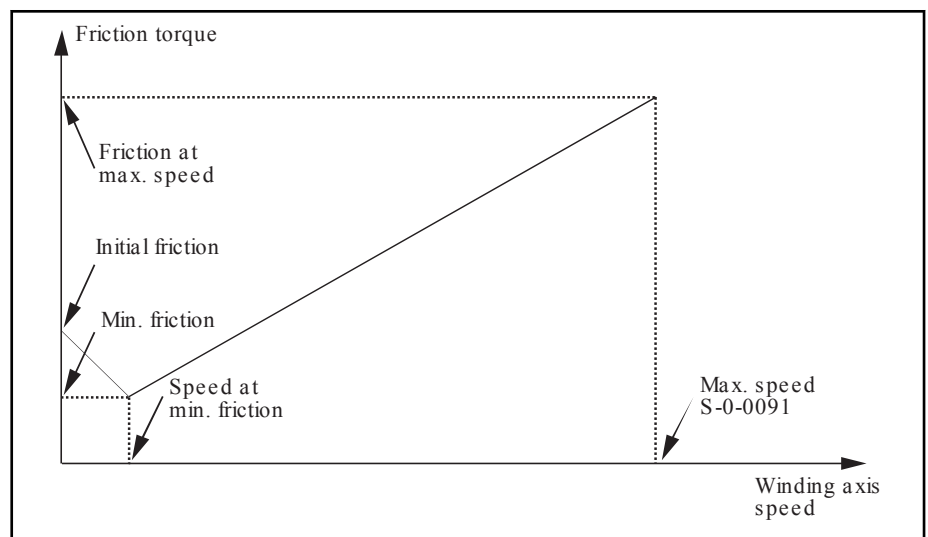


Fig. 8-31: Friction torque compensation

Compensation of acceleration torque

An additional acceleration torque is required for the acceleration of the winding axis. This torque depends on the winding diameter. By specifying certain material and fabric web data, the inertia of the winder is calculated according to the diameter. The calculated acceleration torque can also be filtered.

Torque limit and precontrol

The drive of the winding axis generally DOES NOT operate at its torque limit. Due to safety reasons, the calculated torque is set as value for the torque limit, in addition to a variable additive value (this value should be above the required motor torque and should not limit the drive in normal operation).

A torque limit proportional to the tension and the respective diameter of the winder r is calculated cyclically.

$$M_{Limit} \sim F_{Tension} * r_{Winder} + M_{Accel.} + M_{Friction}$$

Fig. 8-32: Torque limit

This cyclic value for the torque limit is precontrolled as additive torque command value.

Furthermore, this value can be increased by a proportional ratio (TorqueLimitFactor) and an additive ratio (MinTorqueLimit) so that the torque limit does not become active during normal winder operation.

$$M_{ResLimit} \sim M_{Limit} * (1 + TorqueLimitFactor) + MinTorqueLimit$$

Fig. 8-33: Resulting torque limit

The resulting torque limit corresponds to the TorqueLimitbipolar output. By default, the calculated torque limit is increased by a TorqueLimitFactor of 50 % and a minimum TorqueLimit of 10 %.

During tensioning it can be practical to set the torque limit lower (e.g. TorqueLimitFactor 10 %), if tensioning takes place with too much torque. The TorqueLimitFactor should be increased again after tensioning is completed.

Splice

After a splice, the winding diameter generally changes abruptly. The last calculated internal value for the winding diameter does no longer correspond to the current diameter. A preset of the winding diameter is required to calculate the diameter correctly. The input value for the preset should correspond approximately to the actual value of the winding diameter.



If the preset is not performed, malfunctions can occur. The preset should therefore be completed before the diameter calculator becomes active again.

The value for the preset should be selected as follows:

- In the case of a rewinder, it has to be slightly smaller or equal to the actual winding diameter
- in the case of an unwinder, it has to be slightly higher or equal to the actual winding diameter

With a flying splice, it has to be ensured that the circumferential velocity of the new roll corresponds to the current one. This should be implemented respectively in the PLC project of the IndraLogic.

Filtering the actual tension control value

At low tensions, an unfiltered actual tension value can result in a disengaging diameter calculator. A filtering process should therefore be performed and the corresponding input should be activated (e.g. 20 ms).

Enabling the correction of the torque constant at synchronous motors

The torque resulting from a motor is determined at the drive using the motor current and a so-called torque constant P-0-0051 "Torque/force constant". However, this so-called "torque or force constant" is a physical but not a static value.

It depends on:

1. the level of the actual motor current flow
(the constant reduces at currents higher than I_{Nom})
2. the temperature of the motor winding and rotor
(the constant reduces with rising temperature)

A corrected torque constant can be used for the **synchronous motor**. This constant can be enabled in the parameter **P-0-0556 "Axis controller configuration" bit 11 (set)**.

Thus, the following influences on the torque/force constant are considered:

1. Magnetic field weakening by magnetic saturation due to increased motor current compared to nominal current.
2. Magnetic field weakening due to increased motor winding temperature.
3. Magnetic field weakening due to increased rotor temperature depending on the average speed.

The torque constant effective in this case can be read in the parameter P-0-0450 "Torque/force constant, current".

A significant deviation from the torque constant P-0-0051 is only caused if the current load increases to 100% (or higher) of the nominal current S-0-0111 "Standstill current motor". That means the higher the motor load, the better the impact of correction. For a motor loaded at low level, only a minimum or no correction results.



A correction of the torque constant is automatically enabled for an asynchronous motor and does thus not need to be set separately in the parameter P-0-0556.

Also refer to the functional description of the respective IndraDrive drive.

Error handling

The following table lists and describes the error codes of the function block MB_WinderTensionCtrlSpeedType01.

The error codes from the F_RELATED_TABLE (16#0170) are used.

Additionally, the error codes of the blocks MB_DancerControlType03, MB_DiameterCalculatorType03, MB_DiameterMeasurementType01, IL_PIDType01, RIL_LoopControl, as well as MB_GetCyclicParameterHandle, ML_TechMotionBase, can be output.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#11B0	16#0020	Error in the 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	Incorrect torque scaling (A-0-0050 or S-0-0086), scaling for motor torque has to be on the motor side
INPUT_RANGE_ERROR	16#11B1	16#000E	Axis number of the slave axis incorrect
INPUT_RANGE_ERROR	16#11B1	16#000F	Axis number of the master axis incorrect
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	DefaultTorqueLimit maximum
INPUT_RANGE_ERROR	16#11B1	16#0016	MinTorqueLimit maximum
INPUT_RANGE_ERROR	16#11B1	16#0017	TorqueLimitFactor maximum
INPUT_RANGE_ERROR	16#11B2	16#0001	Winding axis not in "BB"
RESOURCE_ERROR	16#0004	16#0000	Incorrect drive firmware
RESOURCE_ERROR	16#000F	16#0003	Drive function package "MA" not enabled

ErrorID	Additional1	Additional2	Text
RESOURCE_ERROR	16#0012	16#00xx	AxisRef.CntrlNo incorrect
RESOURCE_ERROR	16#001C	16#00xx	Incorrect axis type for axis xx

Tab. 8-38: Error codes of the FB MB_WinderTensionCtrlSpeedType01

Special features for the MLC

The scaling type has to be identical for the master and the slave axis. Rotary and preference scaling are recommended.

An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear".

The scaling type has to be identical for master axis and slave axis. Rotary and preference scaling are recommended.

Set the parameter "A-0-0032 Velocity limit value, positive" or "A-0-0033 Velocity limit value, negative" according to the maximum speed of the winding drive (this value can be retrieved from the parameter "S-0-0091 Bipolar velocity limit value").

The standstill window of the reference axis is to be set as low as possible (e.g. 0.1 rpm). When there is a standstill message of the reference axis, the diameter calculator does not calculate!

Interpolation in the drive:

The parameters "P-0-0694 Gear ratio fine adjustment, process loop", "P-0-0690 Additive velocity command value, process loop", "S-0-0092 Bipolar torque/force limit value" and "S-0-0081 Additive torque/force command value" have to be entered into the cyclic Sercos channel of the corresponding drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks.

Interpolation in the control:

The target variables are the control parameters "A-0-2605 ELS, Gear ratio fine adjustment process controller" and "A-0-2615 ELS, additive velocity command value, process controller", "S-0-0092 Bipolar torque/force limit value" and "S-0-0081 Additive torque/force command value".

Write the parameters "S-0-0092 Bipolar torque/force limit value" and "S-0-0081 Additive torque/force command value" to the cyclic Sercos channel of the respective drive.

Special features for the MLD

Interpolation has to be activated in the drive for the winding axis when using the MLD-S control with a higher-order MLC control (Real axes **Corresp. axis ► Properties**). This setting does not exist when using the MLD-M control.

The scaling type has to be identical for master axis and slave axis. Rotary and preference scaling are recommended.

The parameters "P-0-0694 Gear ratio fine adjustment, process loop" and "P-0-0690 Additive velocity command value, process loop" and "S-0-0092 Bipolar torque/force limit value" have to be entered into the cyclic Sercos channel of the corresponding drive ("UserCmdDataX").

The real axes are parameterized in the parameterization mode in IndraWorks.

The real axes are parameterized in the parameterization mode in IndraWorks (the dialog is located in the project tree below the MLD node **AxisData**).

The "AxisData structure" of the drive has to be parameterized ("P-0-1367 PLC configuration, bit 6 is set" or dialog "PLC configuration").

An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear". The parameter "S-0-0091 Bipolar velocity limit value" should be set in accordance with the maximum speed of the winding drive.

Standstill window of the reference axis:

The standstill window of the reference axis is to be set as low as possible (e.g. 0.1 rpm). When there is a standstill message of the reference axis, the diameter calculator does not calculate!

For the reference axis, the function block cannot access any value for the standstill window. 1 rpm is assumed internally. If this value is not suitable for the application, modify it using the following PLC code lines:

Declaration

```
prMaStandstillWdw:    POINTER TO REAL; // for StandstillWindow master MLD
rTestMaStandstillWdw: REAL := 0.5;    // for StandstillWindow master MLD
```

Implementation

```
prMaStandstillWdw := ADR(BausteinInstanznameWickler.fbDiameterCalculator,rMaStandstillWindow);
prMaStandstillWdw^ := rTestMaStandstillWdw;
```

The value of this variable is not saved. Thus, it has to be written again when switching on the PLC.

Axis reference for the master axis:

For technology function blocks, the following is recommended as axis reference for the master axis:

- Use axis reference "**VmAxis1**" for a virtual master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 20 (VMA_1).
- Use axis reference "**RmAxis1**" for a real master axis.
The parameter "P-0-1820.0.1 CCD: Master axes configuration list" has to be set to value 23 (RMA_1).
Additionally, the logic address of the measuring encoder has to be entered in parameter "P-0-1820.0.2 CCD: Source of measuring encoder position feedback value".

For detailed information, refer to Functional Description MLD.

Commissioning the winder with tension control and speed correction

To commission the winder with closed-loop tension control and speed correction, refer to [chapter 8.7.3 "Commissioning the winder with tension control and speed correction" on page 555](#).

8.7.3 Commissioning the winder with tension control and speed correction

Drive control loop

It is assumed that the speed control loop of the winding drive was set. That means that the parameters "S-0-0100 Velocity loop proportional gain" and "S-0-0101 Velocity loop integral action time" have to be set according to the application.

The commissioning of the diameter calculator is described in seven steps in the following:

1. Determining the winder type
2. Determining the direction of rotation of the winding axis
3. Determining the friction torque at standstill
4. Determining the friction torque at maximum speed

5. Determining the minimum friction torque
6. Final settings
7. initial startup of the winding axis

Commanding the winding axis

There are two ways to set the operation mode and controller enabled:

- Set the "ModeSyncVel" mode using the AxisInterface (recommended)
- Use the PLCopen "MC_Power", "MC_GearIn", "MC_Stop" function blocks to command the axis

Determining the winder type

First, it has to be determined which winder type is available or is to be used. Subsequently, the winding direction is defined at the "WindDirection" input (TRUE: Unwinding, FALSE: Rewind).

Determining the direction of rotation of the winding axis

Set the direction of rotation of the tension or winding axis using the drive parameter "P-0-0108 Master drive polarity" or the axis parameter "A-0-2798 Polarity of master axis position". If the winding axis does not rotate in the desired direction, the parameter P-0-0108 or axis parameter A-0-2798 has to be reset accordingly. The polarity of the master axis position is only checked when enabling the function block ("Enable" = TRUE). Thus, modifying the mentioned parameters during the operation of the function block does not have any effect.

Alternatively, the direction of rotation can be changed via the scaling type of the winding drive. This can be done by switching the sign of the position/acceleration data and the torque/force data (parameters S-0-0055, A-0-0029, S-0-0043, S-0-0085).

The following figure shows the principal relations between rewinders and unwinders.

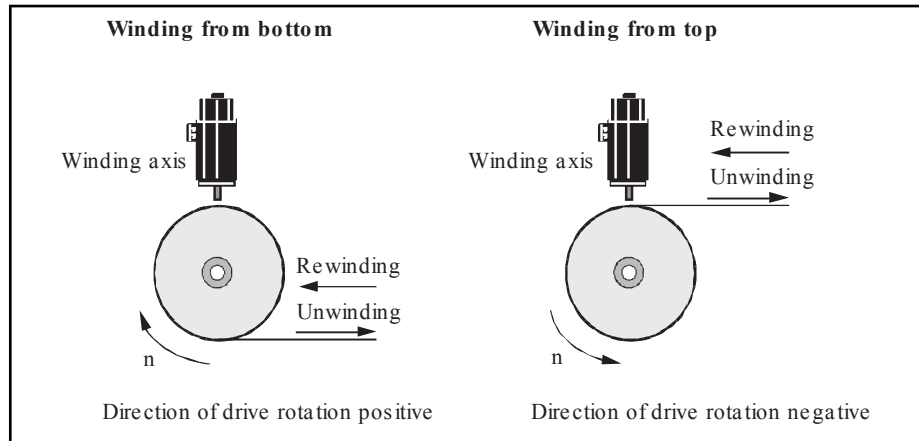


Fig. 8-34: Principles of rewinders and unwinders

Determining the friction torque at standstill

1. The "friction torque at standstill" ("StandstillTorque" variable in the data structure MB_WIND_TENSION) is determined when the master axis is at standstill and the winder is empty (core only). The winding function block is disabled. For the winding axis (slave axis), select the operation mode "Velocity control" (in the AxisInterface of the ML_TechInterface "MODE_VEL" library). The axis can also be controlled using the PLCopen function block MC_MoveVelocity. The command value can for example be set to 10 rpm.
2. The value of parameter "P-0-0109 Torque/force peak limit" is set to zero. The axis has to stop now. The value for this parameter is incre-

mented until the drive starts to rotate. The last value corresponds to the friction torque at standstill.

3. Axis enabled "MODE_VEL" is canceled (alternatively use the PLCopen function block "MC_Stop").

The parameter "P-0-0109 Torque/force peak limit" is reset to its previous value (e.g. 400 %).

The friction torque setting at standstill is thus completed.

Determining the friction torque at maximum speed

1. The "friction torque at maximum speed" ("MaxTorque" variable in the data structure MB_WIND_TENSION) is determined when the master axis is at standstill and with an empty winder (core only). The function block is disabled.
2. For the winding axis (slave axis), select the operation mode "Velocity control" (in the AxisInterface of the ML_TechInterface "MODE_VEL", optionally PLCopen function block).

Increase the command value for the selected operation mode incrementally up to the value of the parameter "S-0-0091 Bipolar velocity limit value". The value for the parameter "S-0-0091 Bipolar velocity limit value" has to be entered according to the application and motor data.

3. In the next step, the value of the parameter "P-0-0109 Torque/force peak limit" is decreased incrementally (e.g. from 100 % towards 0 %) until the drive speed also starts to decrease. The speed of the drive can be traced using the drive parameter "S-0-0040 Velocity feedback value of encoder 1" or the axis parameter "A-0-0102 Actual velocity value". The last value of "P-0-0109 Torque/force peak limit" corresponds to the maximum friction torque during maximum speed.
4. Axis enabled "MODE_VEL" is canceled (alternatively use the PLCopen function block "MC_Stop").

The parameter "P-0-0109 Torque/force peak limit" is reset to its previous value (e.g. 400 %).

Setting the friction torque at maximum speed "MaxTorque" is thus completed.

Determining the minimum friction torque

The minimum friction torque is determined using the deceleration test with an empty winder. The winding drive is decelerated from a constant velocity (e.g. 50 min⁻¹) to standstill. At the same time, the drive parameters "S-0-0080 Torque/force command value" (axis parameter A-0-0038 "torque/force value bipolar") and "S-0-0040 Velocity feedback value of encoder 1" (axis parameter A-0-0102 "actual velocity value") are recorded with an oscilloscope or with the trace function in IndraWorks.

The minimum value is read from the torque curve and can be transferred to the structure MB_WIND_FRICTION as "MinTorque". The corresponding speed is entered as "MinTorqueVelocity" in MB_WIND_FRICTION.

Final settings

Finally, the remaining input variables are specified as shown in the following table:

Input variable	Note/setting
SetpointOpMode	Setting the desired operating tension
SetpointStandstill	Setting the standstill tension
ActTensionFilter	If the actual voltage is to be filtered (as measured value), a value not equal to zero has to be applied at this input of the function block as a time constant
AccelTorqueFilter	If the acceleration torque is to be filtered, a value unequal to zero has to be applied at this input of the function block as time constant

Tab. 8-39: Final determination of the remaining input variables

Initial startup of the winding axis

The winding material is placed in an untensioned state (slightly sagging). The winding axis has to be commanded either via the AxisInterface or via the PLCopen function blocks. The diameter calculator is activated by setting the "Enable" input. Now, the winding material has to be tensioned using the web tensioning control. The "SpeedOffset" output represents the velocity used for tensioning.

The process variable window "WindowTension" can be set to a value from 10 - 20 %.

The master axis is enabled with a comparatively low speed (e. g. 10 min⁻¹). After some revolutions, the "DiameterEngaged" output has to be set. It is thus displayed that the diameter calculation is active ("engaged"). The calculation depends on the web tension or the actual torque at the winding axis.

When the diameter calculator is engaged, the standstill tension and the operating tension can be checked. Therefore, the value "CurrentTension" is monitored from the data structure [MB_WIND_TENS_DIAG](#), page 601,. In this data structure, the current torque limit value "TorqueLimit", the current acceleration torque "Acceleration torque" and the currently applied friction torque "FrictionTorque" can also be monitored.

The function block provides different process variables at its outputs. The calculated current diameter of the winder ("CurrentDiameter" output) should correspond to the actual diameter of the winder. Slight deviations can be disregarded. Furthermore, the fine adjustment of the gear for the speed tracing of the drive axis of the winder ("GearRatioFineAdjust" output), the current moment of inertia of the winder ("CurrentInertia" output) and the speed offset ("SpeedOffset" output), the additive torque "AddTorque" and the torque limit value ("TorqueLimitBipolar" output) are displayed.

Monitoring torque limit values

At the initial control startup, a drive warning "E2056 Torque limit value = 0" can be output if the winder function block is not yet enabled.

In order to prevent this, write a value unequal to zero (e.g. 100 %) to the respective cyclic data container during the startup phase of the PLC program (e.g. warm start). The value of the parameter "S-0-0092 Bipolar torque/force limit value" is written to the mentioned data container.

If the warning "E8260 Torque - force command value limit active" is output when the function block is enabled, clear this warning by setting bit 4 of the parameter "P-0-0556 Config word of axis controller".

In rare cases (e.g. multiple F8078 "Error in the speed control loop") it might be necessary to disable the velocity controller monitoring (set bit 1 of the parameter P-0-0556 " Config word of axis controller").

8.8 Supplementary function blocks

8.8.1 Introduction and overview

This section summarizes function blocks, which can supplement functionalities of the different diameter calculators and the dancer position controller. They can also be implemented into the PLC project of the IndraLogic.

The following function blocks are available for additional functionalities (e. g. remaining runtime for a splice):

Name	Type	Description
MB_CalcInertiaLimitType02 , page 559	FB	Calculates the maximum moment of inertia of the winding material on the winder
MB_UnwindMaterialType02 , page 562	FB	Calculates process variables for an unwinder like the unrolled and the remaining length of the web of fabric, the material thickness and the remaining runtime for a splice.
MB_WindTaperProfileType01, page 567	FB	It calculates a winding taper characteristic curve for a winder
MB_WindSpeedControlAdaptionType01, page 570	FB	It calculates for a winder a Kp-adaptation for the speed controller
MB_TraverseControlType01, page 574	FB	Controls a translatory traversing axis to traverse winding material smaller than the width of the winder (e.g. wire)
MB_DancerCalcControlParamType01, page 590	FB	The function block uses the physical data of the dancer system (diameter of the draw roller or the core diameter of the winder as well as the delay time of the system) to compute the controller parameters of the dancer position controller

Tab. 8-40: Brief description of additional function blocks for different winder types from the RMB_TechWinder library

8.8.2 MB_CalcInertiaLimitType02

Brief description

The function block represents a non-cyclic function block in contrast to the other function blocks described in this documentation. It is systematically, for example, called for the initialization of another function block.

In this function block, the material properties and dimensions of the web of fabric are analyzed and the maximum moment of inertia at the maximum diameter "MaxDiameter" is calculated for the winding material on the winder. The calculated maximum moment of inertia "MaxInertia" may be used, for example, for an adaptation of the speed controller (adaptation of the parameter "S-0-0100 Velocity loop proportional gain"). The proportional gain between a minimum value and a maximum value can be adapted. For example, a linear increase of the proportional gain factor can be set.

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder
MLD with MPx07/08/17 with MA function package	MX_TechWinder.lib

Tab. 8-41: Reference table of the MB_CalcInertiaLimitType02

Interface description

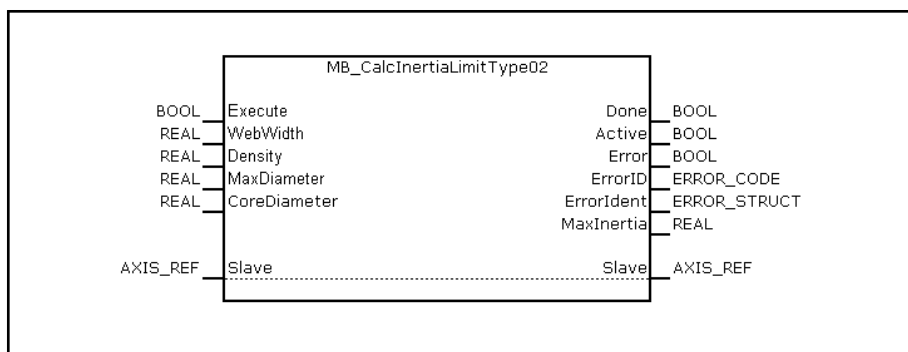


Fig. 8-35: MB_CalcInertiaLimitType02 function block

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Activating the function block (edge-controlled)
	WebWidth	REAL	Web width [mm]
	Density	REAL	Material density of the web [kg/dm ³]
	MaxDiameter	REAL	Maximum diameter of the winder [mm]
	CoreDiameter	REAL	Core diameter of the winder [mm]
VAR_OUTPUT	Done	BOOL	Maximum moment of inertia "MaxInertia" calculated
	Active	BOOL	Processing, function block enabled
	Error	BOOL	Error occurred
	ErrorID	ERROR_CODE	Error code
	ErrorIdent	ERROR_STRUCT	Advanced error display Error handling, page 561
VAR_IN_OUT CONSTANT	MaxInertia	REAL	Maximum moment of inertia [kg/m ²] referring to the motor side
	Slave	AXIS_REF	Axis structure of the winding axis

Tab. 8-42: Interface variables of the MB_CalcInertiaLimitType02 function block

Min./max. and default values

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
Webwidth	REAL	0.0	n. def.	0.0	Rising edge at "Execute"
Density	REAL	0.0	n. def.	1.0	Rising edge at Execute
MaxDiameter	REAL	> 0.0	n. def.	2000.0	Rising edge at "Execute"
CoreDiameter	REAL	> 0.0	n. def.	100.0	Rising edge at "Execute"

Tab. 8-43: Min./max. and default values of the MB_CalcInertiaLimitType02

Signal-time diagram

The following signal-time diagram for the operating inputs and outputs displays the edge-triggered behavior. If there is a positive edge at the "Execute"

input, the function block becomes active. The processing which is, in this example, specified for a duration of two cycles (two boxes in the figure) is displayed at the "Active" output. After the processing, the output is set to "Done". This state remains if "Execute" remains set and if no error ("Error") occurs. If "Execute" is not set, "Done" is deleted after one cycle. The "Error" output is set in case of error. "Error" remains set until "Execute" is reset. If "Execute" is not set, the error message at the "Error" output is deleted after one cycle.

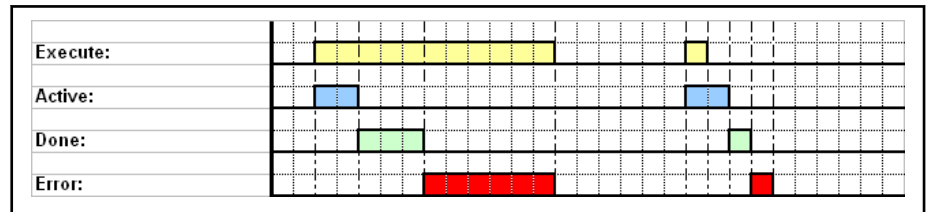


Fig. 8-36: Signal-time diagram of selected inputs and outputs of the MB_CalcInertiaLimitType02 function block

Functional description

With the core diameter "CoreDiameter" of the winder, the web width "WebWidth" and the density of the web of fabric "Density", the moment of inertia "MaxInertia" is calculated for the specified maximum diameter "MaxDiameter".

Error handling

The following table lists and describes the error codes of the function block MB_CalcInertiaLimitType02.

The error codes from the F_RELATED_TABLE (16#0170) are used.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1140	16#0020	Error in state machine
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	Number of slave axis incorrect
RESSOURCE_ERROR	16#0004	16#0000	Incorrect drive firmware
RESSOURCE_ERROR	16#000F	16#0003	Drive function package "MA" not enabled

Tab. 8-44: Error codes of the FB MB_CalcInertiaLimitType02

Required hardware

- CMLx, XMx, VPx
- IndraDrive

Required firmware

- MLC firmware FWA-CMLXX*-MLC-14VRS
XX - Device variant, e.g. CML65
- IndraDrive drive firmware

Required software

- IndraWorks 14VRS for MLC or MLD

Required parameterization

An actual existing mechanical gear ratio from the drive axis to the winding axis has to be written to the parameters "S-0-0121 Input revolutions of load gear" and "S-0-0122 Output revolutions of load gear".

8.8.3 MB_UnwindMaterialType02

Brief description The function block represents a material counter. For an unwinder, it calculates the following cyclic process variables: unrolled web length "UnrolledLength", remaining web length "RestLength", material thickness "MaterialThickness" and the remaining runtime "RemainRunTime".

This data can prepare a splice.

The current winding diameter "CurrentDiameter" and the remaining roll diameter "RestRollDiameter" are input variables.

Assignment target system/library

Target system	Library
MLC 14VRS	RMB_TechWinder
Function package TEC	

Tab. 8-45: Reference table of the MB_UnwindMaterialType02

Interface description

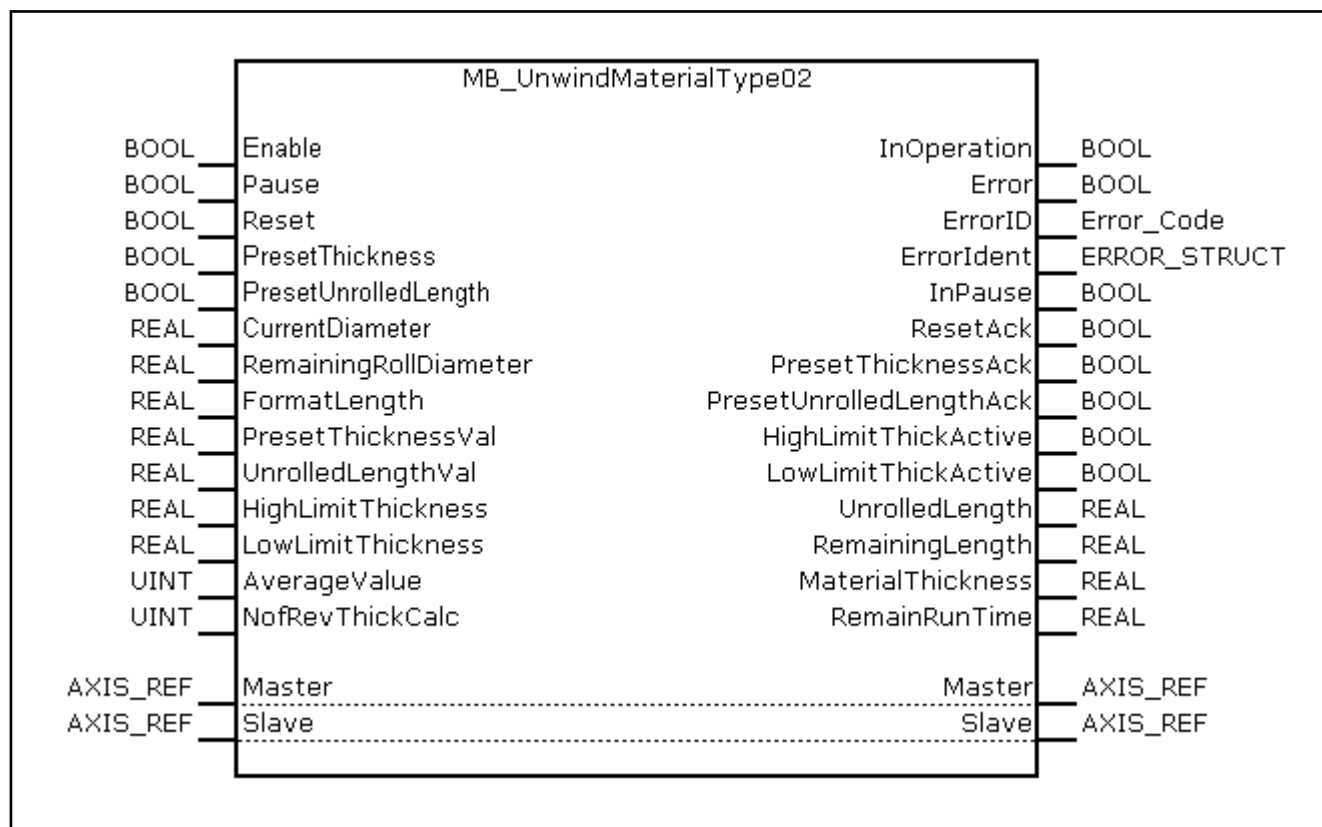


Fig. 8-37: MB_UnwindMaterialType02 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enabling the function block (level-controlled)
	Pause	BOOL	Material count paused
	Reset	BOOL	Resetting the values for the unrolled web length, remaining web length
	PresetThickness	BOOL	Preset for the material thickness edge-controlled
	PresetUnrolledLength	BOOL	Preset for the unrolled web length is edge-controlled

I/O type	Name	Data type	Description
	CurrentDiameter	REAL	Current winding diameter [mm] from the diameter calculator function block
	RemainingRollDiameter	REAL	Remaining roll diameter [mm]
	FormatLength	REAL	Format length [mm] (corresponds to an unwound revolution of the reference axis)
	PresetThicknessVal	REAL	Preset value for the material thickness [μm]
	UnrolledLengthVal	REAL	Preset value for the unwound web length [mm]
	HighLimitThickness	REAL	Upper limit of the material thickness [μm]
	LowLimitThickness	REAL	Lower limit of the material thickness [μm]
	AverageValue	UINT	Number of the material thickness values to be averaged. The input is limited to 30
	NofRevThickCalc	UINT	Number of revolutions after which the material thickness is calculated
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Error occurred
	ErrorID	ERROR_CODE	Error code
	ErrorIdent	ERROR_STRUCT	Advanced error display Error handling, page 566
	InPause	BOOL	Material calculation paused
	ResetAck	BOOL	Reset completed
	PresetThicknessAck	BOOL	Material thickness preset
	PresetUnrolledLengthAck	BOOL	Unrolled length preset
	HighLimitThickAchieved	BOOL	The currently calculated material thickness achieved the upper limit (value is discarded)
	LowLimitThickAchieved	BOOL	The currently calculated material thickness reached the lower limit (value is discarded)
	RemainingLength	REAL	Remaining web length on the winder [mm]
	MaterialThickness	REAL	Material thickness of the web [μm]
	UnrolledLength	REAL	Unrolled web length [mm]
	RemainRunTime	REAL	Remaining runtime until splice (refers to the remaining roll diameter) [s]
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Axis structure of the reference axis
	Slave	AXIS_REF	Axis structure of the winding axis

Tab. 8-46: Interface variables of the MB_UnwindMaterialType02 function block

Min./max. and default values

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Reset	BOOL			FALSE	Continuous
PresetThickness	BOOL			FALSE	Continuous
PresetUnrolled- Length	BOOL			FALSE	Continuous
CurrentDiameter	REAL	0.0	n. def.	0.0	Continuous
RestRollDiameter	REAL	0.0	n. def.	0.0	Rising edge at "Enable"
FormatLength	REAL	0.0	n. def.	0.0	Rising edge at "Enable"
PresetThick- nessVal	REAL	> 0.0	n. def.	0.0	Continuous
UnrolledLengthV- al	REAL	0.0	n. def.	0.0	Continuous
HighLimitThick- ness	REAL	0.0	n. def.	500.0	Continuous
LowLimitThick- ness	REAL	0.0	n. def.	10.0	Continuous
AverageValue	UINT	0	30	1.0	Continuous
NofRevThickCalc	UINT	0	n. def.	50	Continuous

Tab. 8-47: Min./max. and default values of the MB_UnwindMaterialType02 function block

Signal-time diagram

The following explains the signal-time diagram.

The function block is activated with the "Enable" input. This is displayed at the "InOperation" output. If an error occurs, the "Error" output is set and operational availability is interrupted. The execution of the function block can be interrupted with the "Pause" input. This is signaled at the "InPause" output. Resetting the function block ("Reset" input) or setting material parameters ("PresetThickness" and "PresetUnrolledLength" inputs) is performed irrespective of the "Pause" input. However, it is recommended that the function block is activated in "Pause". Thus, the preset and the reset can be performed in a more controlled way. Setting and resetting is displayed at the "ResetAck", "PresetThicknessAck" and "PresetUnrolledLengthAck" outputs.

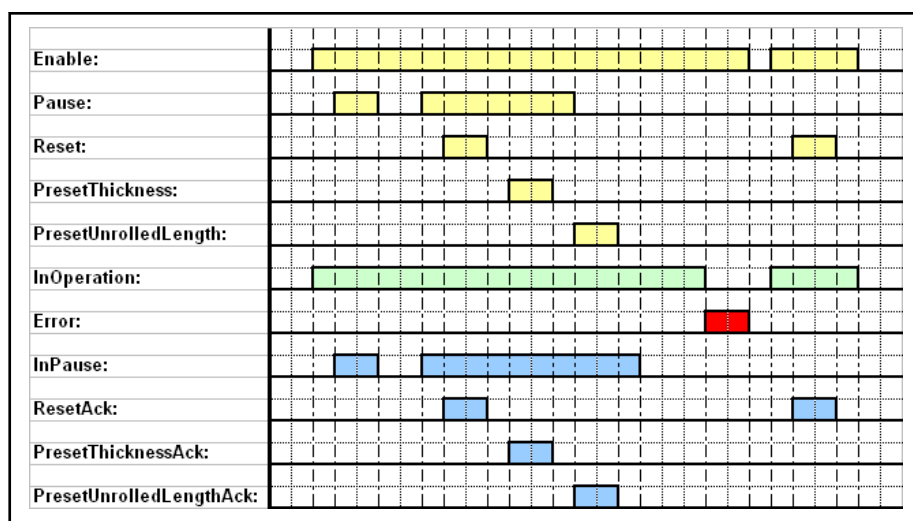


Fig. 8-38: Signal-time diagram of selected inputs and outputs of the MB_UnwindMaterialType02 function block

Functional description

Calculation of the material variables

The remaining length of the "RestLength" web is calculated using the current diameter. Thus, variation in the current diameter can also be seen in the calculation of the remaining length ("RestLength"). The same applies to the calculation of the material thickness ("MaterialThickness") which, in turn, works with the value of the remaining length as well as for the calculation of the remaining runtime ("RemainRunTime") which uses the material thickness and velocity of the reference axis. The more precisely the current diameter is determined, the more precise are the outputs: "UnrolledLength", "RestLength", "MaterialThickness" and "RemainRunTime" of the function block to be calculated.

Further input variables are used. In order to obtain physically useful values for the material thickness, this is limited by "HighLimitThickness" and "LowLimitThickness". This limitation is effective when averaging the material thickness. A fixed number of values "AverageValue" is averaged. A calculation of the material thickness calculates all "NofRevThickCalc" revolutions of the winding axis.

Preset of the material variables

At the beginning of the material counter, the outputs are initialized. This is performed with the values "PresetThicknessVal" and "UnrolledLengthVal". These values can also be used as preset values at runtime. These are set by the inputs "PresetThickness" and "PresetUnrolledLength".

Reset of the material calculator

Moreover, the material counter can be reset by the "Reset" input. Resetting ("Reset") and presetting (Preset) the outputs can be performed at any time if the function block is active.

Average determination of the material thickness

The calculated material thickness can be averaged by setting the input "AverageValue". The number of material thickness values to be averaged is entered at "AverageValue".

Only the current diameters internally calculated and within the limits of "HighLimitThickness" and "LowLimitThickness" contribute to the average value. Values outside these limits are discarded. Values between 1 and 10 are reasonable for "AverageValue". Therefore, "AverageValue" is limited to 30. If a higher value is entered, "AverageValue" is set to 30. However, no error mes-

sage is displayed as this does not represent a critical process. The higher the selected number, the more precise the calculated value of the material thickness. However, the accuracy is reduced with increasing values for "AverageValue", since the calculated value is lagging due to "filtering". The value for "AverageValue" should therefore be selected appropriately.



After the control has been switched off, the memories are deleted. The values for "UnrolledLength", "RestLength", "MaterialThickness" and "RemainRunTime" are therefore 0 when the control is switched on again.

Error handling

The following table lists and describes the error codes of the function block MB_UnwindMaterialType02.

The error codes from the F_RELATED_TABLE (16#0170) are used.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#11A0	16#20	Error in state machine
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	Number of slave axis incorrect
INPUT_RANGE_ERROR	16#11A1	16#0C	Number of master axis incorrect
RESSOURCE_ERROR	16#0004	16#0000	Incorrect drive firmware
RESSOURCE_ERROR	16#000F	16#0003	Drive function package "MA" not enabled

Tab. 8-48: Error codes of the FB MB_UnwindMaterialType02

- Required hardware**
- CMLx, XMx, VPx
 - IndraDrive

- Required firmware**
- MLC firmware FWA-CMLXX*-MLC-14VRS
XX - Device variant, e.g. CML65
 - IndraDrive drive firmware

- Required software**
- IndraWorks 14VRS for MLC or MLD

Required parameterization The interpolation in the drive has to be enabled for the winding axis.

(Real axes ► Corresp. axis ► Properties)

The scaling type has to be identical for the master and the slave axis. Rotary and preference scaling are recommended.

8.8.4 MB_WindTaperProfileType01

Brief description The function block realizes a diameter-dependent adaptation of the winding taper for a winder.

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder
MLD MPx20 and higher with MA function package	MX_TechWinder

Tab. 8-49: Reference table of the MB_WindTaperProfileType01

Interface description

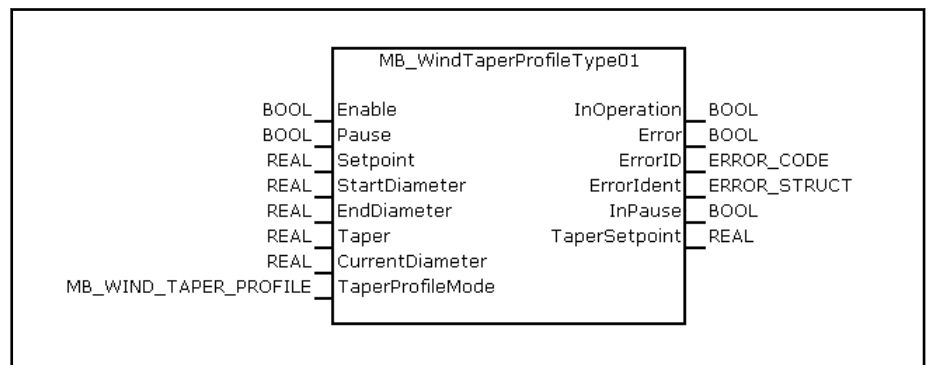


Fig. 8-39: MB_WindTaperProfileType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Function block processing enabled (level-controlled)
	Pause	BOOL	Function block pauses, TaperTensionSetpoint remains "frozen"
	Setpoint	REAL	Command value [-] tension command value for a winder with torque limitation. Command value for pneumatic force for a winder with dancer
	StartDiameter	REAL	Starting diameter [mm] from which the tension should be adapted
	EndDiameter	REAL	Diameter [mm] from which the tension is not adapted anymore (remains constant). Applies only to the linear characteristics
	Taper	REAL	Winding taper [%] specifies the winding taper characteristic curve. If the value is between 0 and 100%, the tension decreases with an increasing diameter. If the value is between 0 and -100%, the tension increases with an increasing diameter
	CurrentDiameter	REAL	Current diameter [mm] of the rewinder (e.g. of the MB_WinderTensionCtrlType02 function block)
	TaperProfileMode	MB_WIND_TAPER_PROFILE	Selection of winding taper profile MB_WIND_TAPER_PROFILE , page 601
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error

I/O type	Name	Data type	Description
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling, page 570
	InPause	BOOL	Winding taper pauses (tension command value is "frozen")
	TaperSetpoint	REAL	Command value considering the winding taper characteristic curve

Tab. 8-50: Interface variables of the MB_WindTaperProfileType01 function block

Min./max. and default values The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
Setpoint	REAL	0.0	n.def.	0.0	Continuous
StartDiameter	REAL	> 0.0	n.def.	100.0	Continuous
EndDiameter	REAL	StartDiameter	n.def.	1000.0	Continuous
Taper	REAL	-100.0	100.0	0.0	Continuous
CurrentDiameter	REAL	0.0	n.def.	0.0	Continuous
TaperProfileMode	MB_WIND_TAPER_PROFILE			NO_PROFILE	Rising edge at "Enable"

Tab. 8-51: Min./max. and default values of the MB_WindTaperProfileType01 function block

Signal-time diagram The following signal-time diagram represents selected function block signals.

Relation of the signals

Signal input	Signal output	Description
Enable	InOperation	Commissioning the function block
	Error	Operational readiness was interrupted
Pause	InPause	Pausing the internal controller process

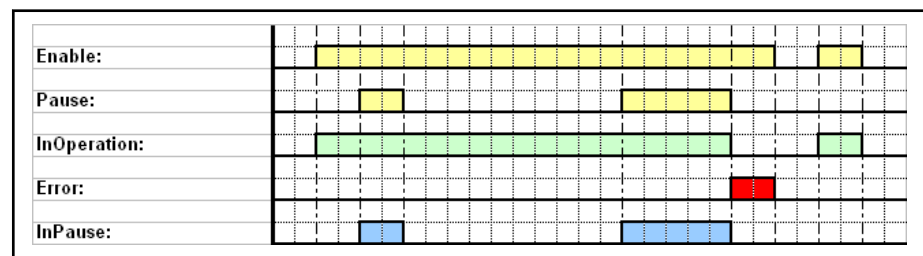


Fig. 8-40: Signal-time diagram of selected inputs and outputs of the MB_WindTaperProfileType01 function block

Functional description When winding material, it is often desired that the tension used for winding is influenced depending on the winding diameter. While winding, the tension

should be higher at the beginning and should be reduced with the winding diameter increasing.

The MB_WindTaperProfileType01 function block influences a determinable command value according to a selectable characteristic curve and a determinable winding diameter. The winding taper characteristic curve can be influenced via the "Taper" input (winding taper). If the value for the winding taper is between 0% and 100%, the tension decreases with an increasing diameter. If the value is between 0% and -100%, the tension increases with an increasing diameter. The higher the absolute value for the winding taper, the higher the increase or decrease of the tension.

The function block can be used for a winder with torque limitation. It is connected in front of the function block MB_WinderTensionCtrlType02, page 526. That means that the "SetpointOpMode" input of the winder function block MB_WinderTensionCtrlType02 is interconnected with the "TaperSetpoint" output.

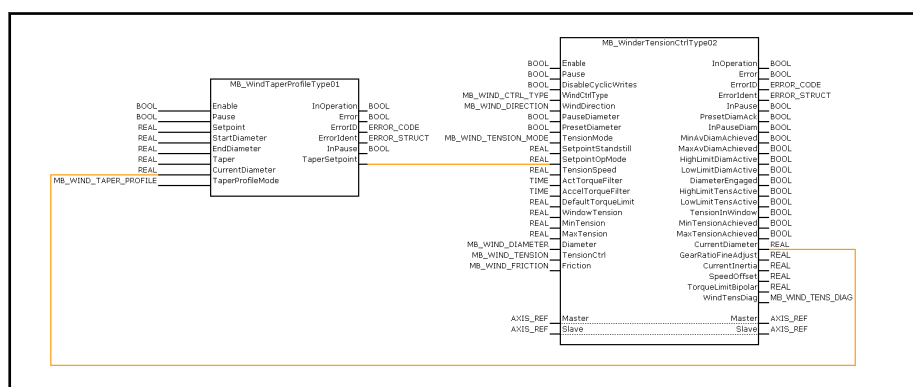


Fig. 8-41: MB_WindTaperProfileType01, interconnection

If the function block for a winder with dancer is used, the "TaperSetpoint" output affects (e.g. analog tension value) the pneumatic force the added to the dancer). The "Setpoint" input then also corresponds to a value for the pneumatic force.

Linear winding taper characteristic curve:

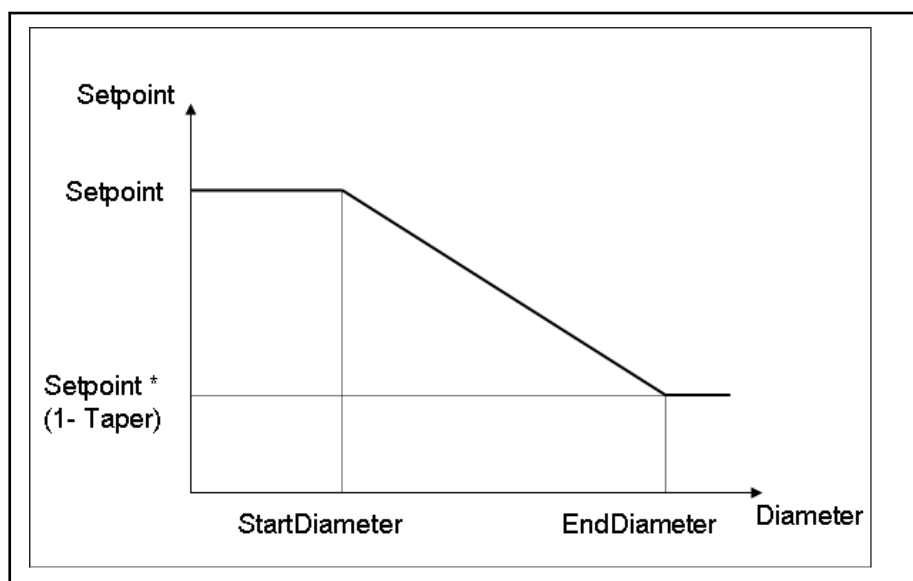


Fig. 8-42: MB_WindTaperProfileType01, linear winding taper characteristic curve

Hyperbolic winding taper characteristic curve:

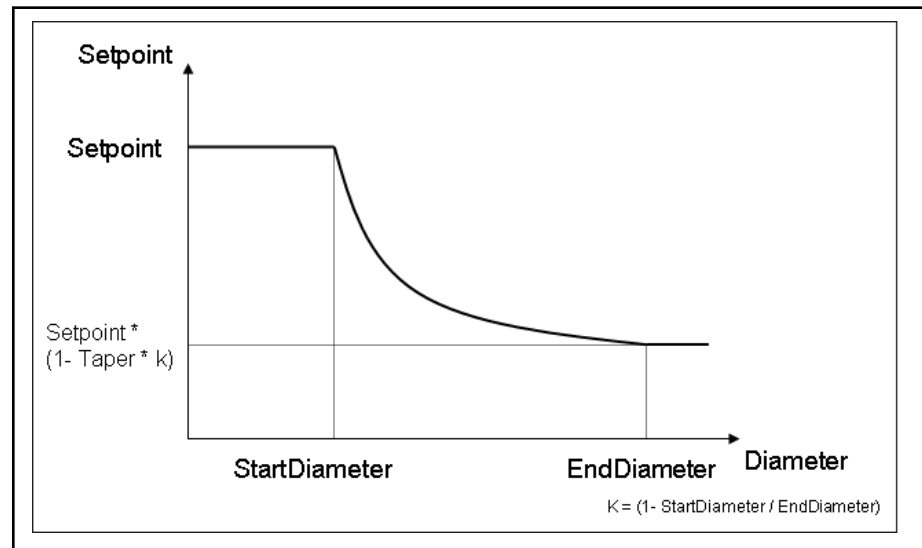


Fig. 8-43: MB_WindTaperProfileType01, hyperbolic winding taper characteristic curve

Error handling The following table lists and describes the error codes of the function block MB_WindTaperProfileType01.

The error codes from the F_RELATED_TABLE (16#0170) are used.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1150	16#0020	Error in the state machine.
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 or Taper > 100.0
INPUT_RANGE_ERROR	16#1151	16#0006	CurrentDiameter <= 0.0
INPUT_RANGE_ERROR	16#1151	16#0007	Incorrect TaperProfile mode

Tab. 8-52: Error codes of the FB MB_WindTaperProfileType01

Required hardware

- CMLx, XMx, VPx
- IndraDrive

Required firmware

- MLC firmware FWA-CMLXX*-MLC-14VRS
XX - Device variant, e.g. CML65
- IndraDrive drive firmware

Required software

- IndraWorks 14VRS for MLC

Special features for the MLC No special parameterization required.

Special features for the MLD No special parameterization required.

8.8.5 MB_WindSpeedControlAdaptionType01

Brief description The function block executes an adaptation of the Kp-gain of the speed controller of the respective winding drive that depends on the current moment of inertia of the winder and a minimum and maximum Kp-value.

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder

Tab. 8-53: Reference table of the MB_WindSpeedControlAdaptionType01

Interface description

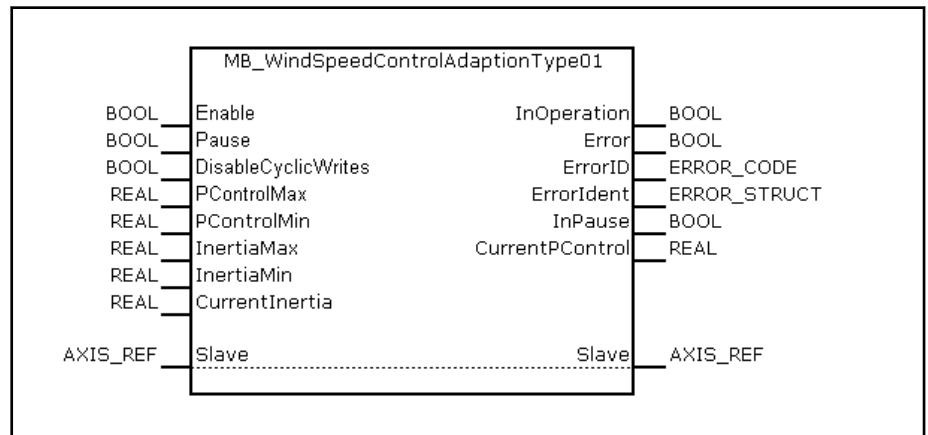


Fig. 8-44: MB_WindSpeedControlAdaptionType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Function block processing enabled (once, edge-controlled)
	Pause	BOOL	Function block pauses, CurrentPControl remains "frozen"
	DisableCyclicWrites	BOOL	Writing on drive parameter S-0-0100 can be switched off
	PControlMax	REAL	Kp-gain for maximum diameter [Nm/(rad/s)]
	PControlMin	REAL	Kp-gain for minimum diameter [Nm/(rad/s)]
	InertiaMax	REAL	Moment of inertia at maximum diameter or moment of inertia [kg*m ²] (related to the motor shaft)
	InertiaMin	REAL	Moment of inertia at minimum diameter (core) or moment of inertia [kg*m ²] (related to the motor shaft)
	CurrentInertia	REAL	Current moment of inertia of the winder [kg/m ²] (related to the motor shaft)
VAR_OUTPUT	InOperation	BOOL	Processing completed without errors, output data valid
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling, page 573
	InPause	BOOL	Function block pauses (CurrentPControl is "frozen")

I/O type	Name	Data type	Description
	CurrentPControl	REAL	Current Kp-value (speed controller) [Nm/(rad/s)]
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Axis structure of the winding axis

Tab. 8-54: Interface variables of the MB_WindSpeedControlAdaptionType01 function block

Min./max. and default values The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Pause	BOOL			FALSE	Continuous
DisableCyclic-Writes	BOOL			FALSE	Rising edge at "Enable"
PControlMax	REAL	PControlMin	n.def.	0.0	Rising edge at "Enable"
PControlMin	REAL	0.0	n.def.	0.0	Rising edge at "Enable"
InertiaMax	REAL	InertiaMin	n.def.	0.0	Rising edge at "Enable"
InertiaMin	REAL	0.0	n.def.	0.0	Continuous
CurrentInertia	REAL	0.0	n.def.	0.0	Continuous

Tab. 8-55: Min./max. and default values of the MB_WindSpeedControlAdaptionType01 function block

Signal-time diagram The following signal-time diagram represents selected function block signals.

Relation of the signals

Signal input	Signal output	Description
Enable	InOperation	Commissioning the function block
	Error	Operational readiness was interrupted
Pause	InPause	Pausing the internal controller process

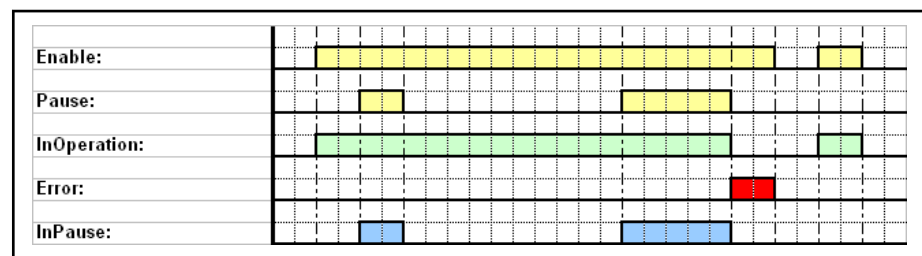


Fig. 8-45: Signal-time diagram of selected inputs and outputs of the MB_WindSpeedControlAdaptionType01 function block

Functional description The function block linearly calculates the value for the Kp of the speed controller depending on the current moment of inertia. This value is calculated between a minimum and maximum value. The minimum and maximum Kp have to be determined at empty core and at full winding diameter. Furthermore, the values for the minimum and maximum moment of inertia (at the

core or at full winding diameter) are required. These can be determined using the function block "MB_CalcInertiaLimitType02".

The current Kp-value of the speed controller is calculated from these values with regard to the current moment of inertia of the winder.

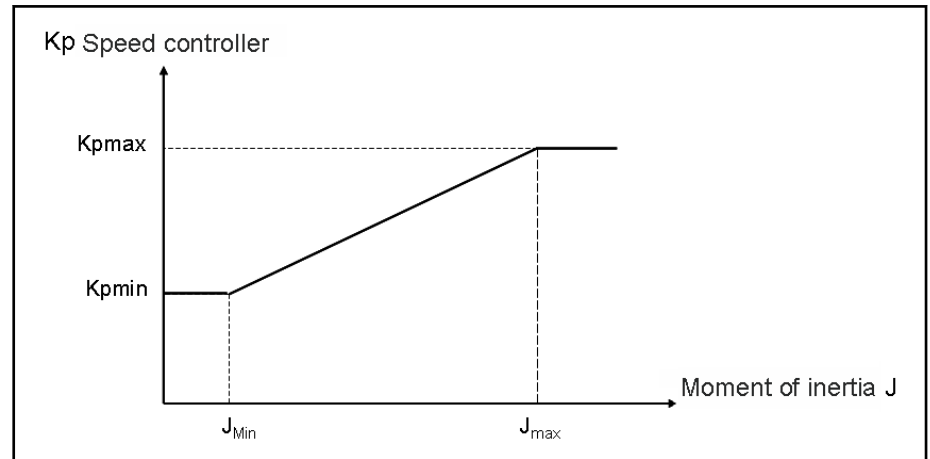


Fig. 8-46: MB_WindSpeedControlAdaptionType01

The current Kp-value is cyclically written to the drive parameter "S-0-0100 Velocity loop proportional gain". Therefore, the parameter S-0-0100 has to be configured in the cyclic data channels between control and drive. If the function block is switched off, the last value of "CurrentPControl" remains present.

This writing can be disabled via the "DisableCyclicWrites" input. The "CurrentPControl" output is then not written internally by the function block, but has to be transferred from the PLC program to the drive parameter.



When switching on the control, the cyclic data channels are always initialized with zero between control and drive. If the parameter "S-0-0100 Velocity loop proportional gain" is configured cyclically, ensure that the respective cyclic channel is pre-initialized with a reasonable value after control start. This applies until the function block was enabled.

Error handling

The following table lists and describes the error codes of the function block MB_WindSpeedControlAdaptionType01.

The error codes from the F_RELATED_TABLE (16#0170) are used.

ErrorID	Additional1	Additional2	Text
INPUT_RANGE_ERROR	16#11E0	16#0020	Error in state machine
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
INPUT_RANGE_ERROR	16#11E1	16#07	InertiaMax <= InertiaMin
DEVICE_ERROR	16#11E2	16#01	Axis not in BB

Tab. 8-56: Error codes of the FB MB_WindSpeedControlAdaptionType01

Required hardware	• CMLx, XMx, VPx • IndraDrive
Required firmware	• MLC firmware FWA-CMLXX*-MLC-14VRS XX - Device variant, e.g. CML65 • IndraDrive drive firmware
Required software	• IndraWorks 14VRS for MLC
Special features for the MLC	The parameter "S-0-0100 Velocity loop proportional gain" has to be entered into the cyclic Sercos channel of the corresponding drive "UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks.



The initial values are always 0 for the parameter in the cyclic Sercos channel. That means that when switching on control, the value for the parameter "S-0-0100 Velocity loop proportional gain" is 0 if it is entered in the cyclic Sercos channel and if the function block is not enabled. To avoid this, the respective data container can be once initialized with a value when enabling the control, e.g.

```
AxisData[DriveName.AxisNo].dwUserCmdDataX_q.RE-
AL_ := 10.0;
```

8.8.6 MB_TraverseControlType01

Brief description The function blocks controls a translatory traversing axis allowing to traverse different materials together with a winder (rotary axis). The diameter of the material is much smaller than the width of the winder. Thus, multiple material windings are placed next to each other. It is distinguished between rewinding and unwinding.

While rewinding, the velocity of the traversing axis is coupled to the velocity of the winder and ensures a constantly tight and parallel traversing of the material.

While unwinding, multiple procedures are available, which can be selected depending on the prerequisites (e.g. existing hardware/load cells).

Output variables of the function block are a target position and a positioning velocity for an MC_MoveAbsolute positioning function block or "Position" and "Velocity" of the operation mode "ModePosAbs" of the AxisInterface for a translatory axis.

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder

Tab. 8-57: Reference table of the MB_TraverseControl01 function block

Interface description

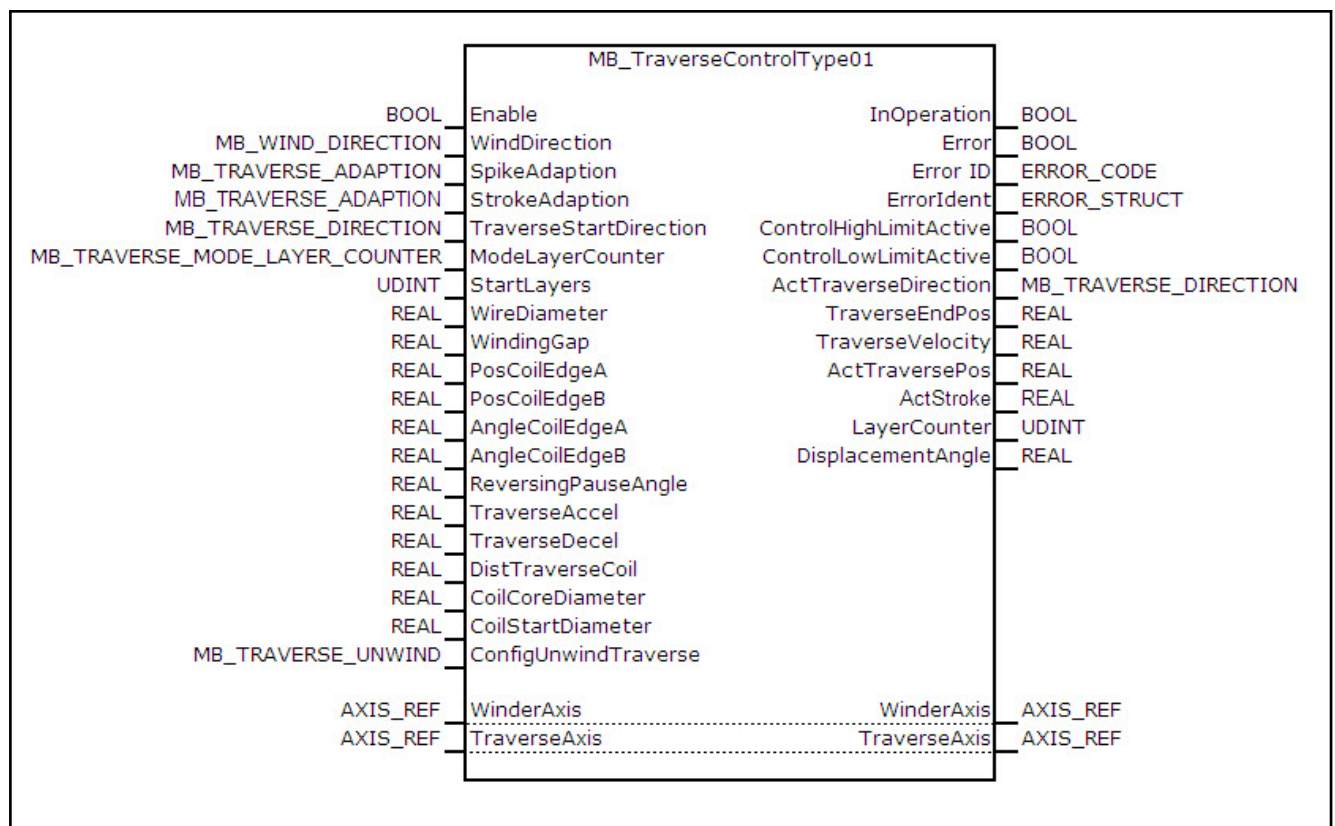


Fig. 8-47: MB_TraverseControlType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Function block processing enabled
	WindDirection	MB_WIND_DIRECTION	<i>Winding direction</i> - Rewinder REWIND (16#0000) - Unwinder UNWIND (16#0001) MB_WIND_DIRECTION , page 600
	SpikeAdaption	MB_TRAVERSE_ADAPTION	Switching on and off the velocity adaption to compensate the missing path during the acceleration and deceleration phase. MB_TRAVERSE_ADAPTION , page 594
	StrokeAdaption	MB_TRAVERSE_ADAPTION	Switching on and off the stroke adaption to the coil edge. In the switch-on state, the traverser moves beyond the coil edge by the value "ActStroke" to ensure that the wire or the material reaches the end of the coil. If switched off, the traversing motion ends at the coil edge. MB_TRAVERSE_ADAPTION , page 594
	TraverseStart-Direction	MB_TRAVERSE_DIRECTION	Starting direction of traversing. From the edge of coil A to the edge of coil B (A_TO_B) or vice versa (B_TO_A). MB_TRAVERSE_DIRECTION , page 595

I/O type	Name	Data type	Description
	ModeLayer-Counter	MB_TRAVERSE_MODE_LAYER_COUNTER	<i>Selection how to determine the starting angle for the layer counter at a positive edge at Enable:</i> - Via specification at "StartLayers". The starting diameter is then calculated and the "CoilStartDiameter" input is not considered. - Calculating via the starting diameter ("CoilStartDiameter"). The "StartLayers" input is then not considered. MB_TRAVERSE_MODE_LAYER_COUNTER, page 595
	StartLayers	UDINT	Number of already wound layers upon start. Is only evaluated at "INPUT_START_LAYERS" at "ModeLayerCounter". [-]
	WireDiameter	REAL	Diameter of the winding material (wire, etc.). [mm]
	WindingGap	REAL	Gap between the individually windings (wires, etc.) of a layer. [% of WireDiameter]
	PosCoilEdgeA	REAL	Absolute position of coil edge A. [mm]
	PosCoilEdgeB	REAL	Absolute position of coil edge B. [mm]
	AngleCoilEdgeA	REAL	Angle of the coil edge against the vertical. (0: Coil edge straight, positive value: Coil edge inclined to the outside, i.e. the coil becomes wider) [degrees]
	AngleCoilEdgeB	REAL	Angle of the coil edge against the vertical. (0: Coil edge straight, positive value: Coil edge inclined to the outside, i.e. the coil becomes wider) [degrees]
	ReversingPauseAngle	REAL	Angle by which the winder rotates further while the traverser remains at the edge of the coil. [degrees]
	TraverseAccel	REAL	Acceleration of the traversing axis. Corresponds to the command value for MC_MoveAbsolute (or the operation mode "ModePosAbs" TechInterface) of the traversing axis. [mm/s ²]
	TraverseDecel	REAL	Deceleration of the traversing axis. Corresponds to the command value for MC_MoveAbsolute of the traversing axis. [mm/s ²]
	DistTraverseCoil	REAL	The distance between traverser and coil corresponds to the wire length passing freely and vertically through air from the traverser to the coil [mm]
	CoilCoreDiameter	REAL	Diameter of the empty coil. [mm]
	CoilStartDiameter	REAL	Diameter of the coil at the beginning (incl. wire). It is only analyzed in case of "CalculateStartLayers" at the "ModeLayerCounter" input. [mm]
	ConfigUnwind-Traverse	MB_TRAVERSE_UNWIND	Structure to configure the unwinding MB_TRAVERSE_UNWIND, page 595

I/O type	Name	Data type	Description
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling, page 589
	ControlHighLimitActive	BOOL	Output of the PID-controller reached upper limit value while unwinding
	ControlLowLimitActive	BOOL	Output of the PID-controller reached lower limit value while unwinding
	ActTraverseDirection	MB_TRAVERSE_DIRECTION	Current traversing direction, from the edge of coil A to the edge of coil B (A_TO_B) or vice versa (B_TO_A) MB_TRAVERSE_DIRECTION, page 595
	TraverseEndPos	REAL	Target position (point of reversal) of the traverser. Corresponds to the position command value of MC_MoveAbsolute of the traversing axis (incl. stroke and offset by inclined coil edges). [mm]
	TraverseVelocity	REAL	Command velocity of the traverser. Corresponds to the velocity command value of the _MoveAbsolute of the traversing axis. [mm/min]
	ActTraversePos	REAL	Current position of the traverser. [mm]
	ActStroke	REAL	Current stroke value (path exceeded when traveling beyond the coil edge). [mm]
	LayerCounter	UDINT	Number of layers on the coil. [-]
	DisplacementAngle	REAL	Angle difference at the coil end to the starting angle after a traversed layer. [degree]
VAR_IN_OUT CONSTANT	WinderAxis	AXIS_REF	Axis reference to the winding axis
	TraverseAxis	AXIS_REF	Axis reference to the traversing axis

Tab. 8-58: Interface variables of the MB_TraverseControlType01 function block

Min./max. and default values

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
WindDirection	MB_WIND_DIRECTION			REWIND	Continuous
SpikeAdaption	MB_TRAVERSE_ADAPTION			ADAPTION_OFF	Rising edge at Enable
StrokeAdaption	MB_TRAVERSE_ADAPTION			ADAPTION_OFF	Rising edge at Enable
TraverseStartDirection	MB_TRAVERSE_DIRECTION			A_TO_B	Rising edge at Enable

Name	Type	Min. value	Max. value	Default value	Effective
ModeLayerCounter	MB_TRAVERSE_MODE_LAYER_COUNTER			INPUT_START_LAYERS	Rising edge at Enable
StartLayers	UDINT	0	n. def.	0	Rising edge at Enable
WireDiameter	REAL	>0.0	n.def.	1.0	Continuous
WindingGap	REAL	0.0	n.def.	0.0	Continuous
PosCoilEdgeA	REAL	0.0	n.def.	0.0	Continuous
PosCoilEdgeB	REAL	> PosCoilEdgeA	n.def.	100	Continuous
AngleCoilEdgeA	REAL	>0.0	< 90.0	0.0	Continuous
AngleCoilEdgeB	REAL	>0.0	< 90.0	0.0	Continuous
ReversingPauseAngle	REAL	0.0	n.def.	60	Continuous
TraverseAccel	REAL	>0.0	n.def.	1000	Rising edge at Enable
TraverseDecel	REAL	>0.0	n.def.	1000	Rising edge at Enable
DistTraverseCoil	REAL	>0.0	n.def.	100	Rising edge at Enable
CoilCoreDiameter	REAL	>0.0	n.def.	100	Rising edge at Enable
CoilStartDiameter	REAL	>= CoilCoreDiameter	n.def.	100	Rising edge at Enable
ConfigUnwindTraverse	STRUCT				Continuous

Tab. 8-59: Min./max. and default values of the MB_TraverseControlType01 inputs

Signal-time diagram

Signal-time diagram

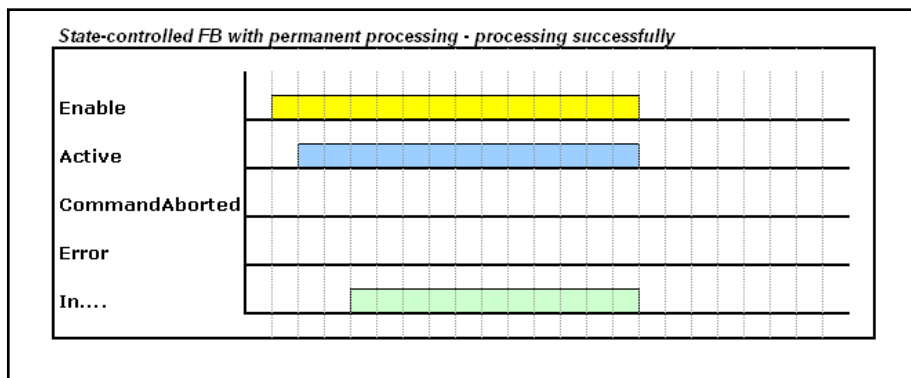


Fig. 8-48: Signal-time diagram of MB_TraverseControlType0

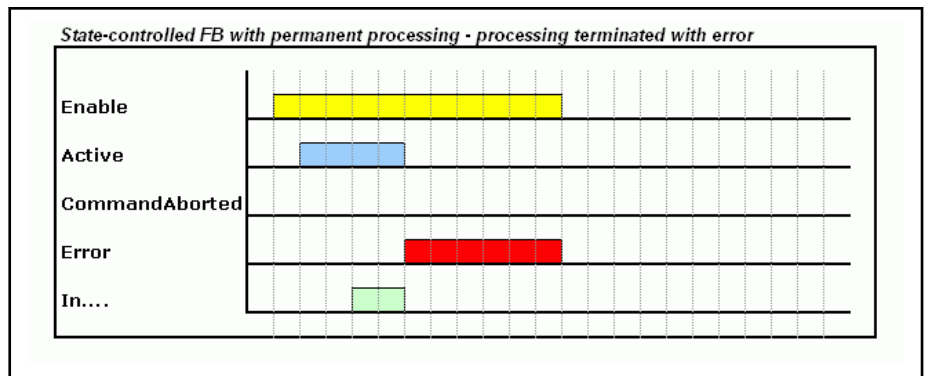


Fig. 8-49: Signal-time diagram of MB_TraverseControlType01 in case of error

Orientation helps:

[Functional description, page 579](#)

[Rewinding functionality, page 580](#)

[Unwinding functionality, page 583](#)

[Alternation between rewinding and unwinding, page 588](#)

[Basic use cases, page 589](#)

Functional description

The function block MB_TraverseControl allows together with a winder function block to rewind and unwind wires, threads, etc. to coils. It is always used with a translatory traversing axis.

Output variables of the function block are a target position and a positioning velocity for an MC_MoveAbsolute positioning function block or "Position" and "Velocity" of the operation mode "ModePosAbs" of the AxisInterface for a translatory axis.

Commanding the traversing axis

The correct operation mode of the drive and drive enable of the drive have to be set for the traversing axis.

There are two methods:

- Set the "ModePosAbs" mode using the AxisInterface (recommended)
- Use the PLCopen function blocks "MC_Power", "MC_MoveAbsolute", "MC_Stop" to command the axis

The corresponding winding axis has to be commanded as well.

The following switch-on switch-off sequence for a traversing axis and a winding axis is recommended:

Switch-on sequence:

1. Switch on the winding axis mode
2. Switch on the traversing axis mode
3. Possibly positioning of the traverser
4. Switch on the traverser function block
5. Switch on the winder function block
6. Switch on the master axis mode

Switch-off sequence:

1. Switch off the master axis mode
2. Switch off the traverser function block

3. Switch off the traverser mode
4. Switch off the winder function block
5. Switch off the winder mode

Interconnecting with the commanding function block MC_MoveAbsolute or AxisInterface

The values for the acceleration and deceleration of the commanding function block or the operation mode of the AxisInterface have to be connected to the inputs "TraverseAccel" and "TraverseDecel" of the traverser function block as well.

The outputs "TraverseEndpos" and "TraverseVelocity" of the traverser function block have to be connected to the inputs "Position" and "Velocity" of the commanding function block or the AxisInterface respectively.

Differences between rewinding an unwinding

It is generally differentiated between rewinding and unwinding for the traversing axis. The traversing axis is translatorily scaled.

- While rewinding, the velocity of the traversing axis is coupled to the velocity of the winder (of a winding axis) and ensures a constantly tight and parallel traversing of the material. It is alternated between coil edge A and coil edge B.
- While unwinding, multiple procedures are available, which can be selected depending on the prerequisites (e.g. existing hardware/load cells).

Rewinding and unwinding is distinguished in the "WindDirection" input.

Rewinding functionality

While rewinding, the traverser alternately moves from coil edge A to coil edge B. With each winding, it is automatically traveled by the material thickness and the gap.

Velocity ("TraverseVelocity"):

The traversing velocity is coupled to the winder speed. With each winder revolution, it is traveled by one winder step resulting from the material thickness and the gap between the windings.

Starting direction ("TraverseStartDirection"):

When the winding starts, specify the direction upon the start of the traverser, i.e. enter the direction (e.g. A_TO_B, that is from coil edge A to coil edge B) at "TraverseStartDirection".

The current direction (e.g. A_TO_B) and the position are output.

Coil edges:

Specify the coil edges as absolute positions. The value of "PosCoilEdgeA" has to be lower than the value of "PosCoilEdgeB". For coils with inclined edges, specify the end points at the coil body.

If the coil edges are not straight, but inclined by an angle against the vertical, specify this angle in "AngleCoilEdgeA" or "AngleCoilEdgeB". A positive angle corresponds to a widening coil. The edge position and thus the target position of the traverser (corresponds to the target position of the corresponding commanding function block "MC_MoveAbsolute" or the operation mode "ModePosAbs" of the TechInterface) is automatically adjusted with regard to the respective winding position.

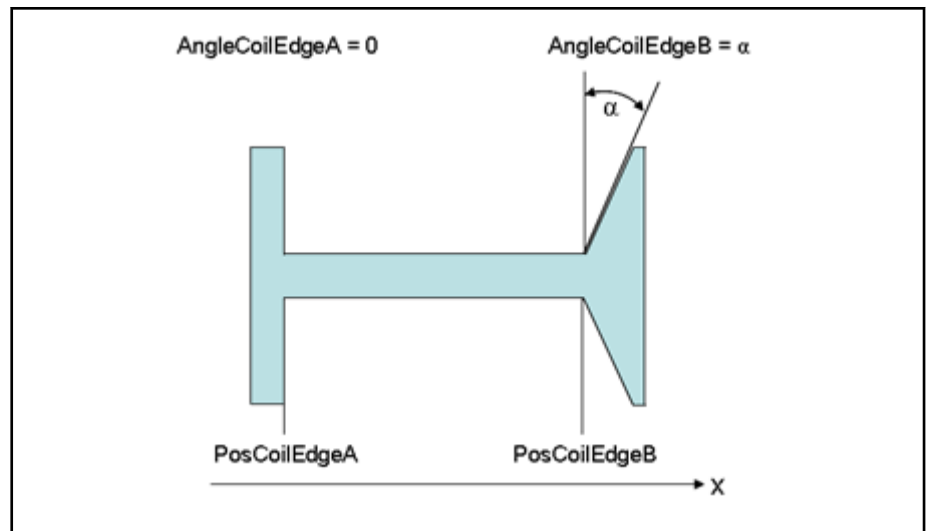


Fig. 8-50: Coil edges

Stroke ("StrokeAdaption"):

Due to the distance between traverser and coil, the traverser has to run a bit faster compared to the position, at which the wire is wound to the coil. This stroke increases with the winding step (material thickness plus gap) and the distance between traverser and coil. In addition, the current (changeable) coil diameter is considered.

To ensure that the wire reaches the coil end, it can be required to move beyond the coil edge. It can be switched on and off with "StrokeAdaption".

If "StrokeAdaption" is enabled, the traverser moves by the current stroke ("ActStroke") and thus beyond the actual coil edge. Thus, the end position is calculated with coil edge A - stroke or coil edge B + stroke. Observe, that the wire can rub at the upper edge especially in case of straight coil edges.

If "StrokeAdaption" is switched off, the traverser returns at the coil edge.

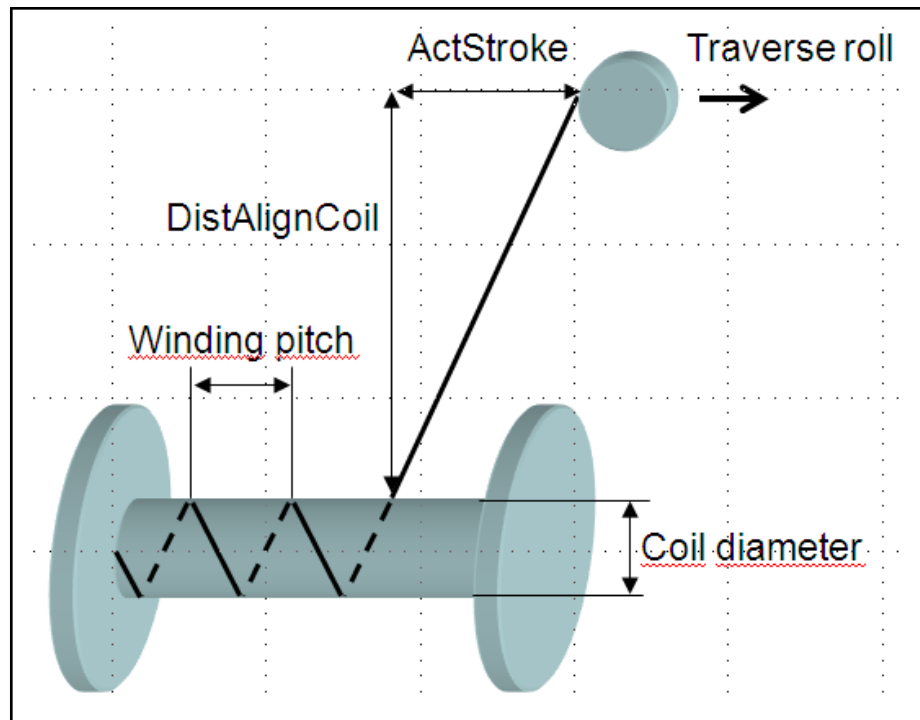


Fig. 8-51: Winding procedure

Layer counter ("LayerCounter"):

The number of wired layers are counted and output.

To keep on winding an already wound coil, determine the initial value for the layer counter. There are two options available that can be selected using the layer counter mode ("ModeLayerCounter").

In the first case, the user specifies the number of layers ("StartLayers"). The "CoilStartDiameter" input is not evaluated, but calculated from the number of layers.

In the second case, the number of layers is calculated from the starting diameter of the coil ("CoilStartDiameter"). The "StartLayers" input is not evaluated.

Waiting angle ("ReversingPauseAngle"):

When the traverser reaches its target position (coil edge), it remains there until the winder rotated by the waiting angle. Waiting allows that the wire can change to the next (upper) layer.

Velocity adaptation ("SpikeAdaption"):

When accelerating and decelerating at the point of reversal (coil edge), that is the beginning and the end of each layer (reversing), the desired traveling distance (winding step) is not fully reached for each winder revolution. If this deviation is too high, it can be compensated by velocity peaks, which can be added via "SpikeAdaption". If it is switched on, the velocity of the traverser is increased to such an extent that the loss is compensated during the first and the last winder revolution of a layer. This can for example be required in case of thick wires to be wound to coils with a relatively small diameter.

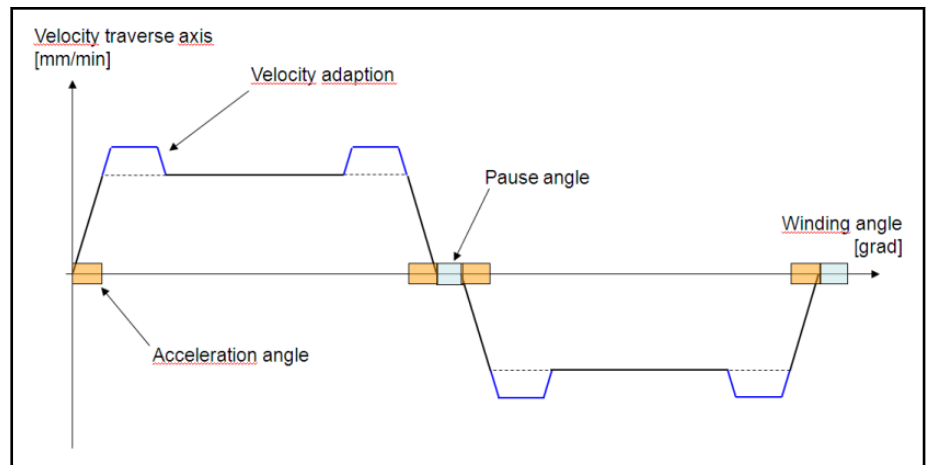


Fig. 8-52: Velocity characteristics with SpikeAdaption

Displacement angle ("DisplacementAngle"):

The wires of one layer are located on the greatest part of the coil circumference, in the grooves, between two wires of the lower layer. However, wire crossings occur in each layer wherever the height of the winding increases. To avoid an unbalance of the coil, these wire crossings have to be equally distributed across the circumference of the coil. For checking purposes, the displacement angle is output. It specifies the angle difference (rotary angle of the winding axis) from the beginning of a layer of one coil end to the beginning of the next layer of the other coil end. If possible, it should be no divisor of 180° . The gap between the windings and the waiting angle can affect the displacement angle.

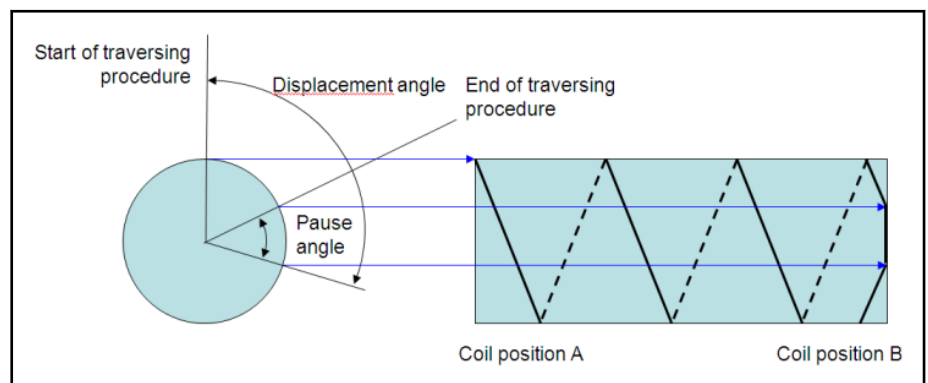


Fig. 8-53: Displacement angle

Unwinding functionality

There are four different methods for unwinding. They are selected via the "ConfigUnwindTraverse.Mode" input.

If the traversing of the wire on the coil is unknown, jumps or changes in direction can be included. Therefore, the methods "WIRE_TENSION_TRIM", "WIRE_TENSION_TRIM_DELTA" and "WINDING_STEP_TENSION_TRIM" are recommended, as the velocity control compensates such effects.

1. "MIDDLE_FIXED" traversing axis is always fix in center position
2. "WINDING_STEP" traversing axis is always traveled by the material thickness (as for rewinding)
3. "WIRE_TENSION_TRIM" traversing axis is scaled with regard to the difference of the actual tension value of the traverser and the actual ten-

sion value outside the traverser and traveled with a PID controller (two load cells are required)

4. "WINDING_STEP_TENSION_TRIM" combination of the methods 2 and 3. It is traveled by the material thickness and the tension difference
(Two load cells are required)

If no measuring devices are available, the methods "MIDDLE_FIXED" and "WINDING_STEP" can be used. "MIDDLE_FIXED" is the easiest method; "WINDING_STEP" if the coil to be unwound was regularly rewound.

If an automatic adaptation of the position of the traverser is intended and the regularly rewound coil is not ensured, use either the method "WIRE_TENSION_TRIM" or "WINDING_STEP_TENSION_TRIM". The tension in the range of the traverser is compared to the web tension outside the traverser. This web tension difference is scaled via a settable PID controller and added to the output velocity of the traverser "TraverseVelocity" with the **value** and the **sign**.

Calibrating load cells:

When calibrating the load cells, ensure that both display the same value if the wire is vertical from the traverser to the coil. Even for the load cell outside the traverser, the wire has to be fed around the roll as in case of operation.

Force measurement

The wire or the material is fed across a roll used for measuring the tensile force. The measured force also depends on how much of the roll is covered by wire or material, i.e. the size of the wrap angle.

The values of the forces F_D and F_{D1} or F_{D2} are equal and represent the tensile force of the wire.

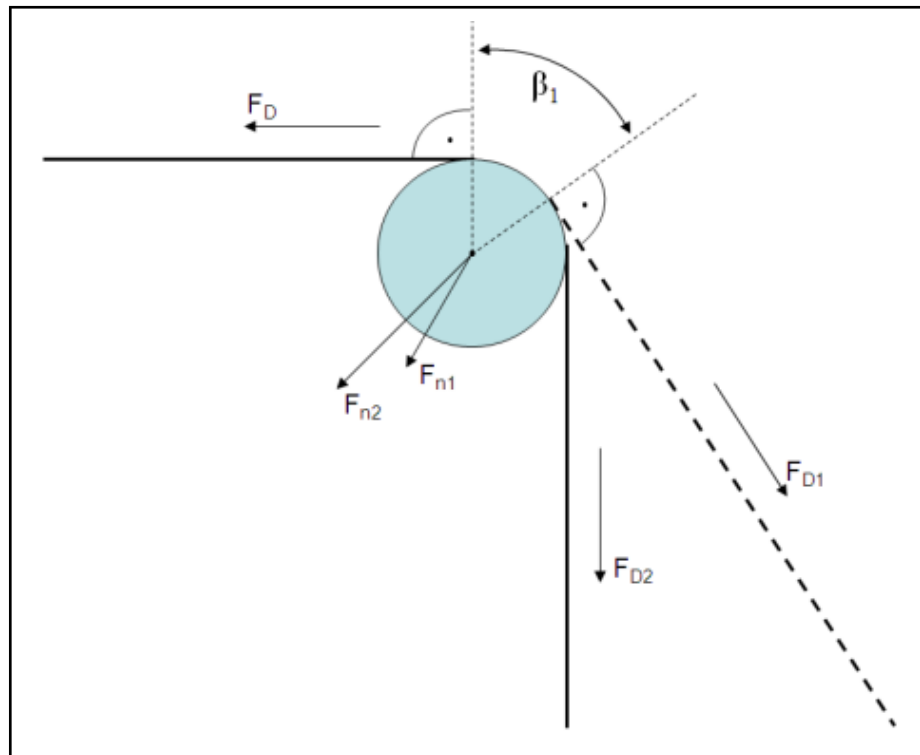


Fig. 8-54: Forces and wrap angle

The force F_{n1} (F_{n2} for β_2) measured by the load cell depends on the wrap angle β_1 . The wrap angle β identifies the area of the roll, in which the wire or the

material is located on the roll. In other words, the angle, which the two verticals to the wire or material enclose together through the roll center point.

The following applies to the measured actual tensile force:

$$F_x = 2 * \sin(\beta/2) * F_{Mx}$$

Fig. 8-55: Correlation between actual tensile force and wrap angle

Wire direction ("ConfigUnwindTraverse.WireFeedDirection"):

To ensure a correct scaling of the velocity, specify the "ConfigUnwindTraverse.WireFeedDirection" input. This input determines the direction in which the wire is fed around the traverser. If the wire direction is from coil edge A to the traverser and then below to the coil, enter "A_TO_B".

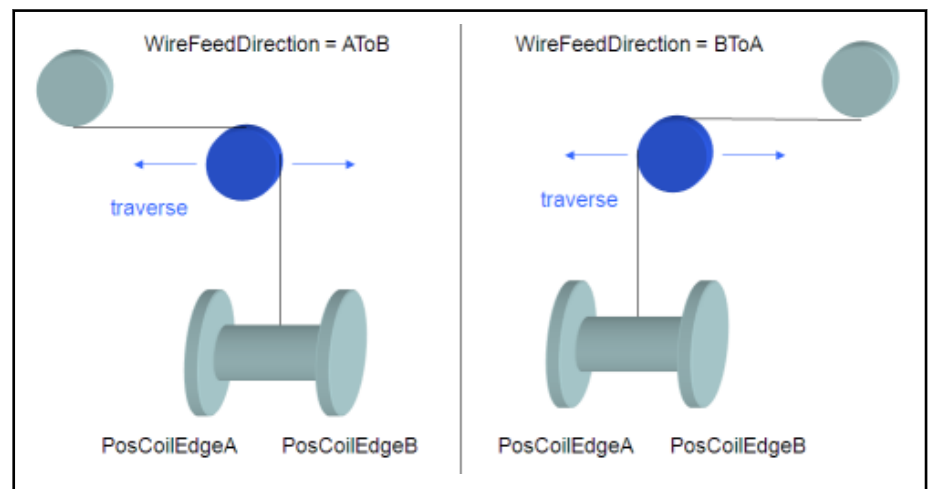


Fig. 8-56: WireFeedDirection

Layer counter ("LayerCounter"):

The layer counter continues. However, an end of the layer cannot exactly be determined. The virtually possible windings of a layer are calculated instead and the winder revolutions are counted. If the calculated windings are reached, a layer is assumed as ended and the layer counter decrements.

The initial value for the layer counter has to be determined. There are two options available that can be selected using the layer counter mode ("Mode-LayerCounter").

In the first case, the user specifies the number of layers ("StartLayers"). The "CoilStartDiameter" input is not evaluated, but calculated from the number of layers.

In the second case, the number of layers is calculated from the starting diameter of the coil ("CoilStartDiameter"). The "StartLayers" input is not evaluated.

Mode "MIDDLE_FIXED":

With this unwinding method, the traverser is fixedly positioned in the coil center. The position is automatically calculated from the coil edge positions and approached with the velocity specified in "ConfigUnwindTraverse.TraverseVelMiddleFixed". If the center position is reached, the traverser stops.

Application:

This mode is only recommended in case of a relatively high distance between the traverser and the coil, relatively narrow coils and insensitive wires, as the wire starts oscillating due to the friction on the lower layers caused by its inclined position at the coil ends.

It cannot be directly switched from this mode to the rewinding mode, as the position of the traverser is lost.

Mode "WINDING_STEP":

With this unwinding method, the winder moves as in rewinding mode (with all its properties). Thus, the traversing axis is moved further by the material thickness and the gap after each winding.

Application:

This method is recommended if the wire was wound before, thus no jumps or changes in direction are expected and a proper regular winding of the coil to be unwound is assumed.

This method requires the least effort.

Mode "WIRE_TENSION_TRIM":

In the "WIRE_TENSION_TRIM" mode, the velocity of the traverser is controlled to ensure that the wire always goes vertically from the traverser to the coil.

Force measurement:

Two forces are required: ("ConfigUnwindTraverse.ActWireTension" and "ConfigUnwindTraverse.ActWireTensionTraverse"). These are each filtered via a PT1 element, whose filter time constants can be set via "ConfigUnwindTraverse.FilterWireTension" or "ConfigUnwindTraverse.FilterWireTensionTraverse" (e.g. 20 ms).

Velocity Control:

Apply the two forces on the controller. The force outside the traverser is the command value and the force in the traverser is the actual value. As both forces are equal in case of a vertical position of the traverser over the coil (cf. "Calibrating the load cells"), the velocity of the traverser tries to ensure with controlling that it is always vertical.

In both cases, the controller output is the command velocity of the traverser (this corresponds to the command velocity of the respective commanding function block MC_MoveAbsolute or the operation mode "ModePosAbs" of the TechInterface).

The control parameters can be set via "ConfigUnwindTraverse.Control". The P-value increases the present control difference. When selecting it, ensure that the traverser can quickly react on jumps in the wire position. The I-value completely controls the control difference. The D-value reacts on a change in the control difference increase.

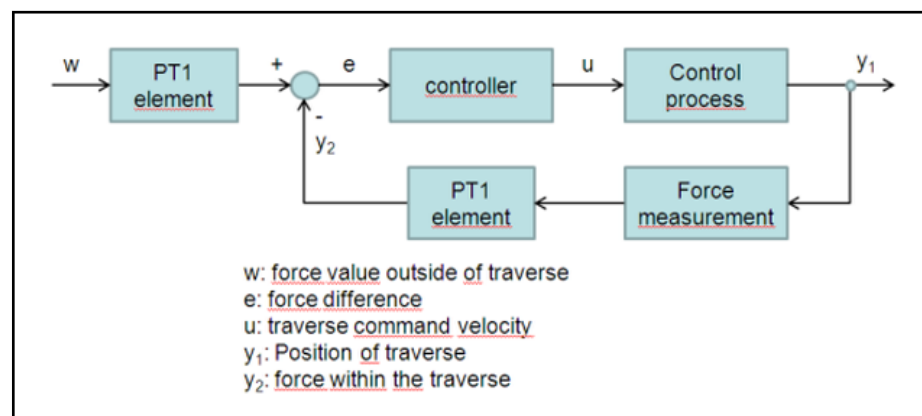


Fig. 8-57: Velocity control with force difference

Only a P-value can be used in the first step.

Set it that high that the traverser does not oscillate at standstill. As the control difference is comparatively low, select a high value (e.g. 1000) for a quick reaction of the traverser.

$$(K_p = 0.5 \cdot k_{p\text{critical}})$$

Target position ("TraverseEndPos"):

The target position of the traverser (corresponds to the target position of the corresponding commanding function block MC_MoveAbsolute or the operation mode "ModePosAbs" of the TechInterface) is the respective coil edge (not mandatorily to be reached). Depending on the determined command velocity, the direction and thus the target position (coil edge) are changed. That means that "TraverseEndPos" is changed when the sign is changed.

Mode "WINDING_STEP_TENSION_TRIM":

This unwinding mode is a combination of "WINDING_STEP" and "WIRE_TENSION_TRIM". The traverser is moved further by one winding step per winder revolution. A closed-loop velocity control is superimposed to compensate irregularities in the specified winding.

Velocity Control:

The "basic velocity" of the traverser is calculated from the winder speed and the winding step (material width plus gap). No velocity compensation by velocity peaks is intended.

A velocity control is superimposed as for the "WireTensionTrim" setting. The control parameters can be set via "ConfigUnwindTraverse.Control". When selecting the P-value, ensure that the superimposed velocity does not overwrite the "basic velocity" in case of small deviations from the vertical position, but that it is significant in case of greater deviations. Additionally, the controller parameters, especially the P-value, have to be adjusted to the winder velocity at runtime by the user. As the "basic velocity" of the traverser increases at an increasing winder speed, the ratio between "basic velocity" and superimposed controlled velocity changes.

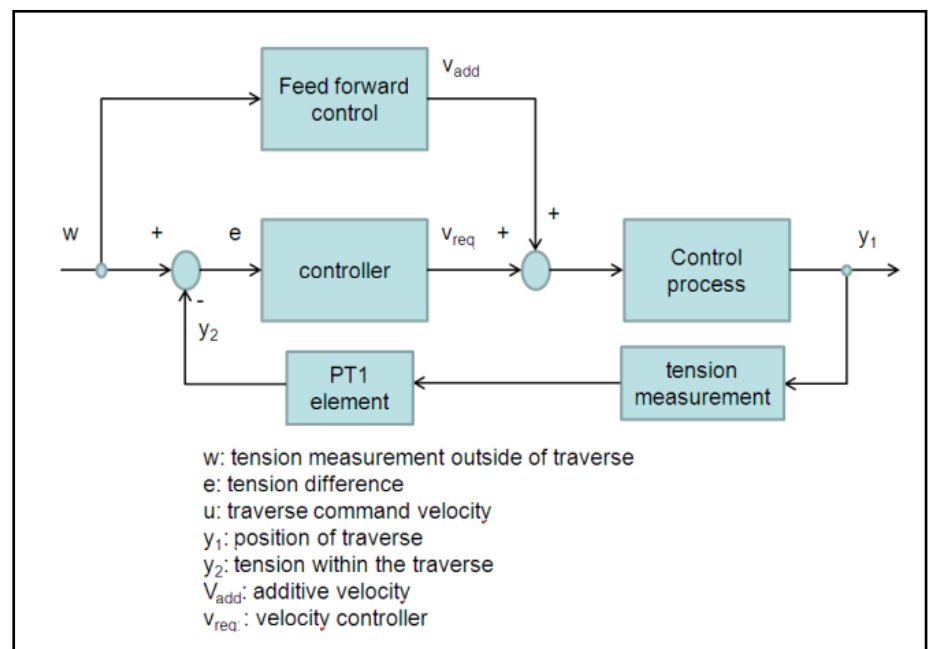


Fig. 8-58: Velocity control with force difference and precontrol

The unwinding method "WindingStepTensionTrim" is a combination of an open-loop control and a superimposed closed-loop control of the traversing

velocity. The two determined velocities are added and the sum is output at the traverser function block. The controller design has to ensure that it does not overwrite the precontrol, but that it can still be effective. A P-controller can be used again in the first step. One possible procedure to determine the P-value:

The desired material velocity " v_{Mat} " is specified. Use the following formula to determine the traverser velocity " $v_{traverse}$ " from the material velocity if the material was equally traversed:

$$n_{Wind} = \frac{v_{Mat}}{d_{coil} * \pi} \text{ and } v_{traverse} = WS * v_{Wind}$$

For example, for a material velocity of " v_{Mat} " = 1 m/s, a coil diameter " d_{coil} " = $80 * 10^{-3}$ m and a winding step (material width + gap) $ws = 0.3$ mm, a traversing velocity of " $v_{traverse}$ " = 31 mm/min results.

Use the formula [fig. 8-55 "Correlation between actual tensile force and wrap angle" on page 585](#) to calculate a force difference with regard to the wrap angle. If for example the wrap angle is changed by five degrees (against 90°), the force and thus the closed-loop control difference of approximately 0.6 results (force at 90° minus force at 85°).

For a maximum deflection angle of 5 degrees, set the proportional controller gain to a value that the traversing velocity becomes 0. The control variable and the velocity determined by the precontrol have to be set to an equal value. Thus, the traversing velocity v_{Trav} has to be divided by the control difference resulting in a P-value of approximately 51.7.

However, this calculation depends on the changeable winder speed. Five degrees as limit value for the wrap angle change depends on the application and was only exemplarily used.

Due to the precontrol, the velocity characteristics are smoother than for "WireTensionTrim". However, the significantly smaller P-value and the resulting lower traversing velocity cause a slower reaction on jumps, etc. in the wire position.

Target position ("TraverseEndPos"):

The target position of the traverser (corresponds to the target position of the corresponding commanding function block MC_MoveAbsolute or the operation mode "ModePosAbs" of the TechInterface) is the respective coil edge.

The direction and thus the target position of the traverser are changed if either the coil edge is reached and it was waited for the waiting angle or if the superimposed velocity exceeds the "basic velocity" and the sum becomes negative.

Application:

This method is especially suitable to unwind coils with a regularly traversed wire/material due to the implicit precontrol. Compared to the controlled unwinding with a winding step, it has the advantage that the settings used (e.g. gap between the windings) while rewinding do not have to be exactly known. Additionally, it can be reacted on irregularities while traversing.

Alternation between rewinding and unwinding

It can be alternated between the two different modes while the function block is operated.

When changing from rewinding to unwinding mode or vice versa, the winding direction is reversed (e.g. "A_TO_B" becomes "B_TO_A").

Do not change from the unwinding mode "MiddleFixed" to the rewinding mode, as no starting position is specified for the traverser. Disable the function block positioning the traverser and enable the function block again.

Basic use cases *Basically, there are two use cases:*

- In the first case, the winding material (e.g. wire) is unwound from or rewound on a coil. The coil is completely unwound and rewound. Subsequently, a new coil is used. In this case, the traverser function block and the winder function block are stopped to change the coils. Subsequently, they are re-initialized and started again. (Application example: coil manufacturing)
- In the second use case, the coil is not necessarily completely unwound or rewound, but a part is unwound and immediately rewound (or vice versa). In this case, there is a direct change from the unwinding to the rewinding mode (or vice versa). The function block is not stopped.

(Exemplary use case for the second use case: ingot saw)

Error handling The following table lists and describes the error codes of the function block MB_TraverseControlType01.

The error codes from the F_RELATED_TABLE (16#0170) are used.

Additionally, the error codes of function blocks MB_DancerControlType03, MB_DiameterCalculatorType03, IL_PIDType01, RIL_LoopControl, and IL_PT1Type01, RIL_LoopControl, can be output.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#0006	16#FFFF	Invalid state of the state machine
INPUT_RANGE_ERROR	16#1111	16#0001	WindDirection is stated incorrectly
INPUT_RANGE_ERROR	16#1111	16#0002	SpikeAdaption is stated incorrectly
INPUT_RANGE_ERROR	16#1111	16#0003	StartDirection is stated incorrectly
INPUT_RANGE_ERROR	16#1111	16#0004	ModeLayerCounter is stated incorrectly
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
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 is stated incorrectly

ErrorID	Additional1	Additional2	Text
INPUT_RANGE_ERROR	16#1111	16#0013	ConfigUnwindTraverse.TraverseVelMiddleFixed < 0.0
INPUT_RANGE_ERROR	16#1111	16#0014	StrokeAdaption is stated incorrectly
INPUT_RANGE_ERROR	16#1111	16#0015	PControl < 0.0
INPUT_RANGE_ERROR	16#1111	16#0016	ConfigUnwindTraverse.WireFeedDirection is stated incorrectly
DEVICE_ERROR	16#1112	16#0001	Traverser axis not in "BB"
DEVICE_ERROR	16#1112	16#0002	Wrong traverser axis number
DEVICE_ERROR	16#1112	16#0003	Wrong winder axis number

Tab. 8-60: Error codes FB MB_TraverseControlType01

- Required hardware**
- CMLx, XMx, VPx
 - IndraDrive
- Required firmware**
- MLC firmware FWA-CMLXX*-MLC-14VRS
XX - Device variant, e.g. CML65
 - IndraDrive drive firmware
- Required software**
- IndraWorks 14VRS for MLC
- Special features for the MLC** Set a translatory scaling for the traversing axis.

8.8.7 MB_DancerCalcControlParamType01

Brief description The function block uses the physical data of the dancer system (diameter of the draw roller or the core diameter of the winder as well as the delay time of the system) to compute the controller parameters of the dancer position controller. The computed controller parameters for "PControl", "IControl" and "DControl" can be used for the function blocks [MB_DancerControlType04](#), [page 485](#), and [MB_WinderDancerCtrlType02](#), [page 516](#).

This controller parameterization using specific distance parameters is useful if the commissioner of the system does not have knowledge in control engineering.

Assignment: Target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechWinder

Tab. 8-61: Reference table of the MB_DancerCalcControlParamType01

Interface description

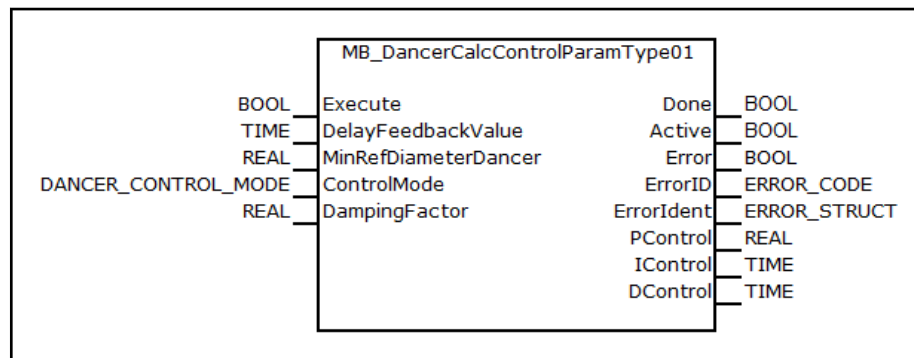


Fig. 8-59: MB_DancerCalcControlParamType01 function block

I/O type	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Starts the function block (once, edge-controlled)
	DelayFeedbackValue	TIME	Control loop delay (i.e. the sum of all filter and dead time added by the user, including the cycle time when calling the respective winder function block, e.g. MB_WinderDancerCtrlType02)
	MinRefDiameterDancer	REAL	Minimum reference diameter (in mm) for winders with dancer
	ControlMode	MB_DANCER_CONTROL_MODE	Controller type selection (P-, PI-, PID-controller) MB_DANCER_CONTROL_MODE, page 602
	DampingFactor	REAL	This input affects the control response (in %). The following can be selected for the control response: From slow, aperiodic/creeping to fast overshooting/oscillating.
VAR_OUTPUT	Done	BOOL	Computation completed without error and the output values are valid
	Active	BOOL	Computation running and the output values are invalid
	Error	BOOL	Computation completed without error (see ErrorID and ErrorIdent)
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling, page 490
	PControl	REAL	Proportional gain
	IControl	TIME	Integral action time
	DControl	TIME	Lead time

Tab. 8-62: Interfaces of the MB_DancerCalcControlParamType01 function block

Min./max. and default values and effective of the inputs

- Describes the min./max. and default values and effective of the inputs
- Boolean and enumeration values do neither have a minimum nor a maximum value (only a default value)

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Rising edge
DelayFeedback-Value	TIME	T#10ms	T#1s	T#100ms	Rising edge at "Enable"
ControlMode	MB_DANCER_CONTROL_MODE			PI_CONTROL	Rising edge at "Enable"
DampingFactor	REAL	20.0	200.0	100.0	Rising edge at "Enable"
MinRefDiameter-Dancer	REAL	> 0.0	N. def.	100.0	Rising edge at "Enable"

Tab. 8-63: Min./max. and default values of the MB_DancerCalcControlParamType01 inputs

Functional description This function block computes the controller parameters for a dancer position controller using the path geometry and some additional inputs such as the process delays.

The user can select from three different controller types:

- P-controller
- PI-controller
- PID-controller

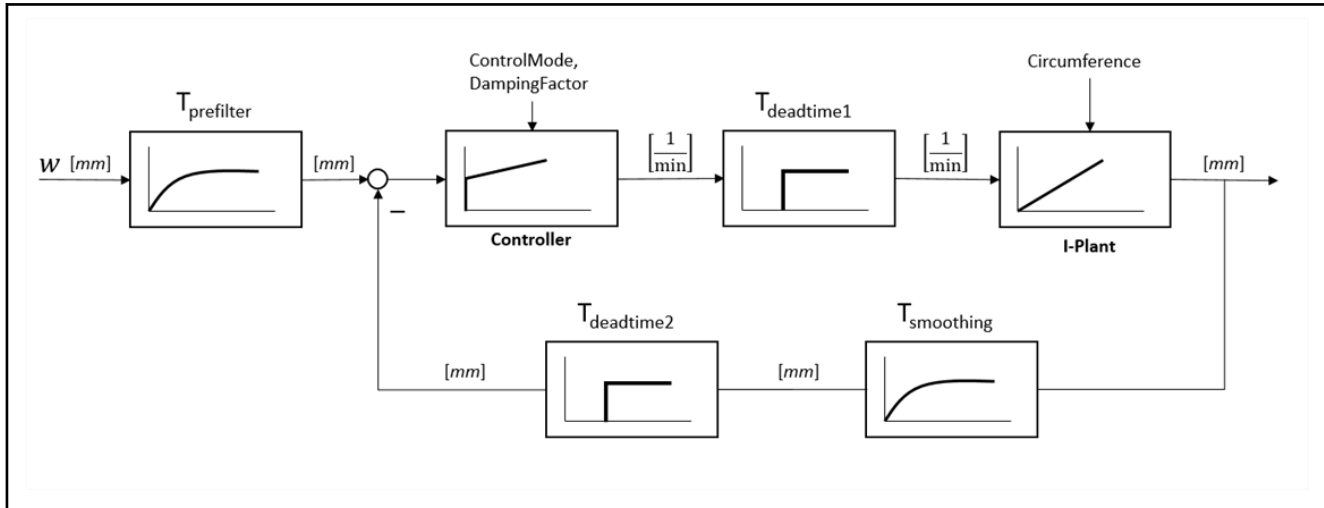


Fig. 8-60: Diagram of the MB_DancerCalcControlParamType01



The command value and the actual value of the dancer position controller have to be scaled in **mm** when using this function block!

DelayFeedbackValue:

All feedback delays and smoothing times are included by the DelayFeedbackValue input.

Thus, the block diagram shows a total delay time of $\text{DelayFeedbackValue} = T_{\text{deadtime1}} + T_{\text{smoothing}} + T_{\text{deadtime2}}$.

The value DelayFeedbackValue can also be estimated with a jump response. Make a jump response and record it (addition of an additive speed command value in jumps to a winder axis and measurement of a dancer position $h(t)$ as shown in the following figure). Measure T_1 using the cursor function to estimate the DelayFeedbackValue value.

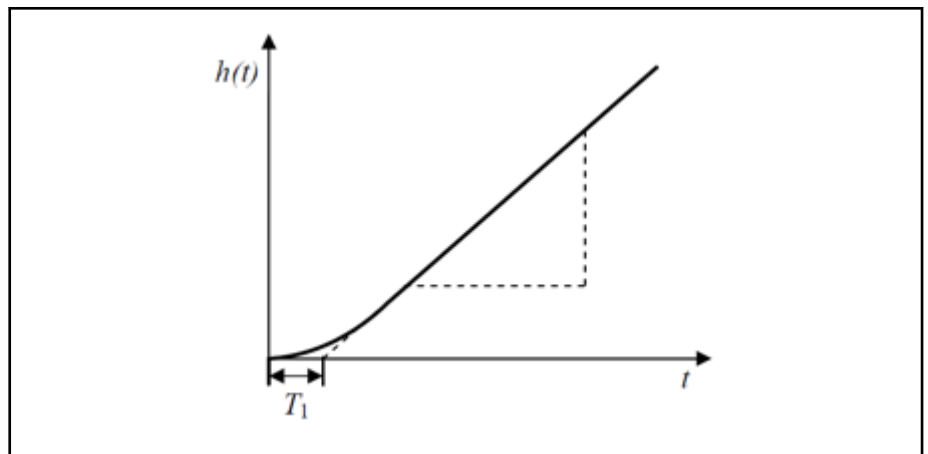


Fig. 8-61: Jump response of the dancer position

MinRefDiameterDancer:

The MinRefDiameterDancer input refers to the input with the same name in the data structure [MB_WIND_DIAMETER](#), page 598 of the function block MB_WinderDancerCtrlType02.

The same value as for the function blocks has to be set (diameter of the winding core).

ControlMode:

Different controller types (P-, PI- and PID-controller) can be selected using the "ControlMode" input as described in [MB_DANCER_CONTROL_MODE](#), page 602.

DampingFactor:

This input affects the control response (in %). The following can be selected for the control response: From slow, aperiodic/creeping to fast overshooting/oscillating. The correct value depends on the mechanical system dynamics. Use "trial and error". Start with a low value (e.g. 30 %).

Prefilter:

Normally, dancer position controllers keep a specified command position and compensate interferences the best possible. The computed parameterization generally causes an overshooting of the dancer position controller if the dancer position command value of the dancer position controller changes, as the controller design was optimized for disturbance responses. To avoid this response, the user can add an additional command value filter (IL_PT1Type01) in front of the "SetPointDancer" of the dancer position controller to the PLC program. The output value "IControl" provides a functional filter time constant for this command value filter.

Error handling

The following table lists and describes all error codes of the MB_DancerCalc-ControlParamType01 function block.

The error codes from F_RELATED_TABLE are used.

ErrorID	Additional1	Additional2	Text
INPUT_RANGE_ERROR	16#00000002	16#00000001	DampingFactor exceeds the limits
INPUT_RANGE_ERROR	16#00000002	16#00000002	DelayFeedbackValue exceeds the limits
INPUT_RANGE_ERROR	16#00000002	16#00000003	MinRefDiameterDancer ≤ 0.0

Tab. 8-64: Error codes of the MB_DancerCalcControlParamType01 function block

8.9 Data types

8.9.1 Introduction and overview

This section summarizes the data types used by the function block.

Name	Type	Description
MB_TRAVERSE_ADAPTION , page 594	ENUM	Adaptation on/off
MB_TRAVERSE_MODE_LAYER_COUNTER , page 595	ENUM	Initial value for layer counter
MB_TRAVERSE_UNWIND , page 595	STRUCT	Settings at the traverser for unwinding
MB_TRAVERSE_UNWIND_MODE , page 597	ENUM	Selection of the unwinding method
MB_TRAVERSE_DIRECTION , page 595	ENUM	Traveling direction of the traverser
MB_TRAVERSE_UNWIND_CONTROL , page 596	STRUCT	Assigning the PID controller for the unwinding
MB_WIND_CTRL_TYPE , page 597	ENUM	Open-loop/closed-loop tension control
MB_WIND_DANCER_LIMITS , page 597	STRUCT	Delimiters of the controller control variables
MB_WIND_DIAM_CALC_MODE , page 598	ENUM	Selection of the diameter calculation method
MB_WIND_DIAMETER , page 598	STRUCT	Diameter calculator data
MB_WIND_DIRECTION , page 600	ENUM	Winding unit type
MB_WIND_FRICTION , page 600	STRUCT	Friction torque data
MB_WIND_TAPER_PROFILE , page 601	ENUM	Cam profiles
MB_WIND_TENS_DIAG , page 601	STRUCT	Controller deviation of the dancer profile
MB_WIND_TENSION , page 601	STRUCT	Tension controller data
MB_WIND_TENSION_MODE , page 601	ENUM	Selection of the mode to tension the material web
MB_WIND_TORQUE_LIMITS , page 602	STRUCT	Bipolar torque limitation
MB_DANCER_CONTROL_MODE , page 602	ENUM	Possible variants of a closed-loop dancer control

Tab. 8-65: Data types of different winder types from the RMB_TechWinder library

8.9.2 MB_TRAVERSE_ADAPTION

This enumeration is used to activate or deactivate an adaptive variable or a mode (for example, stroke adaption "StrokeAdaption", velocity adaption "SpikeAdaption")

Element	Value	Description
ADAPTION_OFF	0	Adaption is switched off
ADAPTION_ON	1	Adaption is switched on

Tab. 8-66: Elements of the enumeration type MB_TRAVERSE_ADAPTION

8.9.3 MB_TRAVERSE_DIRECTION

This enumeration is used to differentiate between the direction of travel of the traverser. It is possible to travel from coil edge A (PosCoilEdgeA) towards coil edge B (PosCoilEdgeB) or in the opposite direction:

Element	Value	Description
A_TO_B	0	From edge of coil A to edge of coil B.
B_TO_A	1	From edge of coil B to edge of coil A.

Tab. 8-67: Elements of the enumeration type MB_TRAVERSE_DIRECTION

8.9.4 MB_TRAVERSE_MODE_LAYER_COUNTER

This enumeration type defines how the initial value for the layer counter is to be determined with a positive edge at Enable.

Element	Value	Description
INPUT_START_LAYERS	0	Imports the initial value for the layer counter from the input
CALCULATE_START_LAYERS	1	Calculates the initial value for the layer counter from the coil diameter

Tab. 8-68: Elements of the enumeration type MB_TRAVERSE_MODE_LAYER_COUNTER

8.9.5 MB_TRAVERSE_UNWIND

This structure is used to make the required settings on the traverser for the unwinding process.

The selection of the unwinding method is made using the "Mode" element.

The "WireFeedDirection" input describes the direction from which the wire or material is wrapped around the traverser.

Use the "Control" structure to configure a PID controller that controls the traverser velocity (depending on the unwinding method).

The "TraverseVelMiddleFixed" input defines the velocity at which the traverser travels to the center position in "MIDDLE_FIXED" mode.

Use the two inputs "ActWireTension" and "ActWireTensionTraverse" to import the values of the load cells (depending on the unwinding method). Both load cell values can each be filtered using a PT1 element whose time constants are specified using the two "FilterWireTensionTraverse" and "FilterWireTension" inputs.

Name	Type	Min. value	Max. value	Default value	Effective	Description
Mode	MB_TRAVERSE_UNWIND_MODE , page 597			MIDDLE_FIXED	Rising edge at "Enable"	Selection of the various unwinding methods.
WireFeed-Direction	MB_TRAVERSE_DIRECTION , page 595			A_TO_B	Continuous	Direction from which the wire or material is guided around the traverser.
Control	MB_TRAVERSE_UNWIND_CONTROL , page 596				Continuous	Configuration of the controller for velocity control during unwinding
VelMiddle-Fixed	REAL	0.0	n.def.	50	Continuous	Velocity at which the traverser travels to the center position in "MIDDLE_FIXED" mode. [mm/min]
ActWireTensionTraverse	REAL	0.0	n.def.	0.0	Continuous	Current measured value of the wire or material tension on the traverser, measured using a load cell in the traverser unit. [N]
ActWireTension	REAL	0.0	n.def.	0.0	Continuous	Current measured value of the wire or material tension in the "System", measured using a load cell outside the traverser unit. [N]
FilterWire-TensionTraverse	TIME	0 s	n.def.	0 s	Continuous	Time constant of the PT1 element for filtering "ActWireTensionTraverse"
FilterWire-Tension	TIME	0 s	n.def.	0 s	Continuous	Time constant of the PT1 element for filtering "ActWireTension"

Tab. 8-69: Min./max. and default values of the MB_TRAVERSE_UNWIND structure

8.9.6 MB_TRAVERSE_UNWIND_CONTROL

This structure is used to assign the inputs of the PID controller for the unwinding process. The controller output additively affects the velocity of the traversing axis.

Name	Type	Min. value	Max. value	Default value	Effective	Description
PControl	REAL	0.0	n.def.	0.0	Continuous	Kp-gain of the PID velocity controller
IControl	TIME	0 s	n.def.	0 s	Continuous	Integral action time of the PID velocity controller
DControl	TIME	0 s	n.def.	0 s	Continuous	Lead time of the PID velocity controller

Name	Type	Min. value	Max. value	Default value	Effective	Description
HighLimit	REAL	LowLimit	n.def.	0.0	Continuous	Upper limit of the PID velocity controller [mm/min]
LowLimit	REAL	n.def.	HighLimit	0.0	Continuous	Lower limit of the PID velocity controller [mm/min]

Tab. 8-70: Min./max. and default values of the MB_TRAVERSE_UNWIND_CONTROL structure

8.9.7 MB_TRAVERSE_UNWIND_MODE

This enumeration type allows the selection of the unwinding method.

Element	Value	Description
MIDDLE_FIXED	0	Traverser is fixed in the middle of the coil
WINDING_STEP	1	Traverser travels in a controlled way by the wire width or material width (plus gap)
WIRE_TENSION_TRIM	2	Automatic adjustment of the traverser velocity depending on the force deviation in the traverser and outside the traverser. The end position is the edge of the coil.
WINDING_STEP_TENSION_TRIM	3	Combination of WINDING_STEP and WIRE_TENSION_TRIM, i.e. traveling by wire width or material width (plus gap) and additive correction using force deviation

Tab. 8-71: Elements of the enumeration type MB_TRAVERSE_UNWIND_MODE

8.9.8 MB_WIND_CTRL_TYPE

Short description The enumeration type MB_WIND_CTRL_TYPE allows the selection of the open loop tension control (sensorless) and the closed loop tension control (load cell)..

Name	Value	Transfer	Description
OPEN_LOOP	16#0000	at a rising edge of Enable	Winding type "open loop" without sensor
CLOSED_LOOP	16#0001	at a rising edge of Enable	Winding type "closed loop" with load cell

Tab. 8-72: Interface description of the MB_WIND_CTRL_TYPE

8.9.9 MB_WIND_DANCER_LIMITS

Brief description The data structure MB_WIND_DANCER_LIMITS describes the limiter of the controller control variable in the diagram.

Name	Type	Min. value	Max. value	Default value	Effective	Description
LowLimitControl	REAL	0.0	n. def.	0.0	Continuous	Lower limitation of the controller output (at a speed of 0) bipolar [rpm] (the value is internally standardized with the diameter ratio of "ReferenceDiameter" and "MinRefDiameter")
HighLimitControl	REAL	0.0	n. def.	0.0	Continuous	Upper limitation of the controller output (at a speed of VelocityHighLimit) bipolar [rpm] (the value is internally standardized with the diameter ratio of "ReferenceDiameter" and "MinRefDiameter")
VelocityHighLimit	REAL	0.0	n. def.	0.0	Continuous	Speed of the master for HighLimitControl [rpm]

Tab. 8-73: Interface description of the data structure of the MB_WIND_DANCER_LIMITS

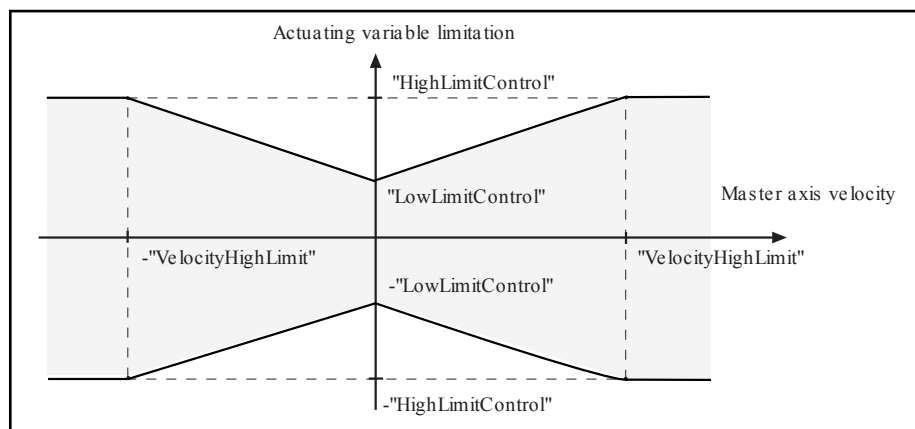


Fig. 8-62: Control variable limitation of the dancer position controller

8.9.10 MB_WIND_DIAM_CALC_MODE

Short description The enumeration type MB_WIND_DIAM_CALC_MODE allows the selection of the diameter calculation type.

Name	Value	Transfer	Description
CALC_DIAM_MANUAL	16#0000	at a rising edge of Enable	Diameter calculation dependent on CalcRange
FIRST_REV_AUTO_CALC	16#0001	at a rising edge of Enable	Diameter calculation is done every 3 degrees for the first revolution and thereafter depending on the CalcRange

Tab. 8-74: MB_WIND_DIAM_CALC_MODE interface description

8.9.11 MB_WIND_DIAMETER

Brief description The MB_WIND_DIAMETER data structures describes the data required for the diameter calculator.

Name	Type	Min. value	Max. value	Default value	Effective	Description
PresetVal	REAL	> 0.0	n.def	100.0	Continuous	Default value for the preset diameter [mm]
CoreVal	REAL	> 0.0	n.def	100.0	Rising edge at "Enable"	Diameter of the core [mm]
CalcRange	REAL	0.0	n.def	360.0	Continuous	Range for diameter calculation [degrees] (load-sided)
RepeatLength	REAL	> 0.0	n.def	100.0	Rising edge at "Enable"	Format length [mm] (1 revolution of the reference axis unwound)
Webwidth	REAL	0.0	n.def	0.0	Rising edge at "Enable"	Web width [mm] (is required for the moment of inertia)
Density	REAL	0.0	n.def	1.0	Rising edge at "Enable"	Material density [kg/dm ³] (is required for the moment of inertia)
AverageValue	UINT	0	20	1	Continuous	Number of diameter values to be averaged [-]
HighLimit	REAL	LowLimit	n.def	2000.0	Continuous	Upper limit for the diameter [mm] The diameter is limited to this value
LowLimit	REAL	> 0.0	HighLimit	100.0	Continuous	Lower limit for the diameter [mm] The diameter does not fall below this value
MinDiameter	REAL	> 0.0	n.def	100.0	Continuous	Minimum diameter [mm]
MaxDiameter	REAL	> 0.0	n.def	2000.0	Continuous	Maximum diameter [mm]
MechanicInertia	REAL	0.0	n.def	0.0	Continuous	Moment of inertia of the mechanics [kg m ²] based on the motor side (is required for the resulting moment of inertia)
DancerConstant (is only required for a winder with dancer)	REAL	0.0	n.def	0.0	Continuous	Displacement of the dancer, based on the unit of dancer adjustment [mm/-] (is only required for a winder with dancer)
MeasuredVal	REAL	0.0	n. def	0.0	Continuous	Measured diameter [mm]
PivotVelocity	REAL	0.0	n.def	0.0	Continuous	Pivot velocity (for relative motion of the winding axis) [mm/s]

Name	Type	Min. value	Max. value	Default value	Effective	Description
DiamCalcMode	MB_WIND_DIAM_CALC_MODE , page 598	CALC_DIAM_MANUAL	FIRST_REV_AUTO_CALC	CALC_DIAM_MANUAL	Rising edge at "Enable"	Diameter calculation mode CALC_DIAM_MANUAL Diameter calculation depending on CalcRange FIRST_REV_AUTO_CALC Diameter calculation is executed after every 5 degrees for the first revolution. Thereafter, it depends on the CalcRange
DiamSource	BOOL				Rising edge at "Enable"	Diameter determination: FALSE: Diameter is calculated TRUE: Diameter is measured (MeasuredVal)
MinRefDiameter-Dancer	REAL	0.0	n.def	100.0	Rising edge at "Enable"	Minimum reference diameter for winders with a dancer

Tab. 8-75: Interface description of the MB_WIND_DIAMETER data structure

8.9.12 MB_WIND_DIRECTION

Short description The enumeration type MB_WIND_DIRECTION allows the selection of the diameter direction type.

Name	Value	Transfer	Description
REWIND	16#0000	at a rising edge of Enable	Rewinder
UNWIND	16#0001	at a rising edge of Enable	Unwinder

Tab. 8-76: Interface description of the MB_WIND_DIRECTION

8.9.13 MB_WIND_FRICTION

Short description The data structure of the MB_WIND_FRICTION characterizes the friction torque.

Name	Type	Min. value	Max. value	Default value	Transfer	Description
StandstillTorque	REAL	0.0	Not defined	0.0	Continuous	Friction torque at standstill
MaxTorque	REAL	0.0	Not defined	0.0	Continuous	Friction torque at S-0-0091, Bipolar velocity limit value
MinTorque	REAL	0.0	Not defined	0.0	Continuous	Minimum friction torque
MinTorqueVelocity	REAL	0.0	Not defined	0.0	Continuous	Velocity at minimum friction torque

Tab. 8-77: Interface description of the MB_WIND_FRICTION

8.9.14 MB_WIND_TAPER_PROFILE

Element	Value	Description
NO_PROFILE	0	no taper profile
PROFILE_LINEAR	1	linear taper profile
PROFILE_HYPERBOLIC	2	hyperbolic taper profile

Tab. 8-78: ENUM MB_WIND_TAPER_PROFILE

8.9.15 MB_WIND_TENS_DIAG

Brief description The data structure MB_WIND_TENS_DIAG describes individual components of the torque limit value as well as the current actual tension control value for diagnostic purposes.

Structural element	Type	Min. value	Max. value	Default value	Effective	Description
TensionCtrlOutput	REAL	N. def.	n. def.	0.0	Continuous	Output of the tension controller [N]
TorqueLimit	REAL	n. def.	n. def.	0.0	Continuous	Signed torque limit [%]
AccelerationTorque	REAL	N. def.	n. def.	0.0	Continuous	Current acceleration torque [%]
FrictionTorque	REAL	n. def.	n. def.	0.0	Continuous	Current friction torque [%]
CurrentTension	REAL	0.0	n. def.	0.0	Continuous	Current actual tension voltage value [N]

Tab. 8-79: Interface description of the MB_WIND_TENS_DIAG data structure

8.9.16 MB_WIND_TENSION

Brief description The data structure MB_WIND_TENSION describes the data of the tension controller.

Name	Type	Min. value	Max. value	Default value	Effective	Description
PControl	REAL	0.0	n.def	0.0	Continuous	Kp-gain of the tension controller
IControl	TIME	0.0	n.def	0.0	Continuous	Tn integral action time of the tension controller
FeedbackVal	REAL	0.0	n.def	0.0	Continuous	Actual value of the tension
PreControlWeighting	REAL	0.0	n.def	1.0	Continuous	Scaling of the command value addition
HighLimit	REAL	0.0	n.def	0.0	Continuous	Upper limit of the tension controller
LowLimit	REAL	0.0	n.def	0.0	Continuous	Lower limit of the tension controller

Tab. 8-80: Interface description of the MB_WIND_TENSION

8.9.17 MB_WIND_TENSION_MODE

Short description The enumeration type MB_WIND_TENSION_MODE allows the selection of the mode to tension the material web.

Name	Value	Transfer	Description
CONST_SPEED	16#0000	at a rising edge of Enable	Tensioning method: constant tensioning speed
CONST_WEB_VEL	16#0001	at a rising edge of Enable	Tensioning method: constant web velocity

Tab. 8-81: Interface description of the MB_WIND_TENSION_MODE

8.9.18 MB_WIND_TORQUE_LIMITS

Structure for bipolar torque limit (S-0-0092).

Name	Type	Min. value	Max. value	Default value	Effective	Description
DefaultTorqueLimit	REAL	0.0	400.0	100.0	Continuous	Value for the bipolar torque limit after deactivation of "Enable" [%]
TorqueLimitFactor	REAL	0.0	400.0	50.0	Continuous	Factor by which the calculated TorqueLimit is increased [%]
MinTorqueLimit	REAL	0.0	400.0	10.0	Continuous	Minimum TorqueLimit by which the calculated TorqueLimit is increased [%]

Tab. 8-82: Interface description of the data structure of the MB_WIND_TORQUE_LIMITS

8.9.19 MB_DANCER_CONTROL_MODE

The enumeration type MB_DANCER_CONTROL_MODE describes the possible variants of a closed-loop dancer control.

Element	Value	Description
P_CONTROL	1	Parameter computation of the P-controller
PI_CONTROL	2	Parameter computation of the PI-controller
PID_CONTROL	3	Parameter computation of the PID-controller

Tab. 8-83: Elements of the enumeration type MB_DANCER_CONTROL_MODE

8.10 Parameter overview

This section lists the parameters used in the documentation.

There are "axis parameters" (A-parameters) for the MLC control, "product-specific parameters" (P-parameters) and "standard parameters" (S-parameters) for the respective drive.

For the MLC control, there are only "product-specific parameters" (P-parameters) and "standard parameters" (S-parameters).

Parameters	Name
A-0-0029	Position polarities
A-0-0038	Torque/force limit value, bipolar
A-0-0050	Scaling type for torque/force data
A-0-0102	Actual velocity value
A-0-0500	Configuration of user-defined command value A
A-0-0502	Configuration of user-defined command value B

Parameters	Name
A-0-0504	Configuration of user-defined command value C
A-0-0506	Configuration of user-defined command value D
A-0-0520	Configuration of actual user-defined value A
A-0-0522	Configuration of actual user-defined value B
A-0-0524	Configuration of actual user-defined value C
A-0-0526	Configuration of actual user-defined value D
A-0-2605	ELS, Gear ratio fine adjustment process controller
A-0-2615	ELS, additive velocity command value, process controller
A-0-2798	Polarity of master axis position

Tab. 8-84: MLC axis parameters (A-parameters)

Parameters	Name
S-0-0040	Velocity feedback value of encoder 1
S-0-0043	Velocity polarity parameter
S-0-0055	Position polarities
S-0-0080	Torque/force command value
S-0-0081	Additive torque/force command value
S-0-0084	Torque/force feedback value
S-0-0085	Torque/force polarity parameter
S-0-0091	Bipolar velocity limit value
S-0-0092	Bipolar torque/force limit value
S-0-0100	Velocity loop proportional gain
S-0-0121	Input revolutions of load gear
S-0-0122	Output revolutions of load gear
S-0-0124	Standstill window

Tab. 8-85: Default parameters (S-parameters)

Parameters	Name
P-0-0108	Master drive polarity
P-0-0556	Config word of axis controller
P-0-0690	Additive velocity command value, process loop
P-0-0694	Gear ratio fine adjustment, process loop
P-0-0912	Virtual master axis, standstill window
P-0-1367	PLC configuration

Tab. 8-86: Product-specific parameters (P-parameters)

Parameters	Name
P-0-1623	CCD: Configuration list master cmd values
P-0-1624	CCD: Configuration list actual master values
P-0-1625	CCD: Configuration list slave cmd values
P-0-1626	CCD: Configuration list actual slave values
P-0-1730	CCD: Command value data container 1, Master 4Byte
P-0-1731-39	CCD: Command value data container 1, Slave 1 4Byte, slave n (n = 1-9)
P-0-1740	CCD: Command value data container 2, Master 4Byte
P-0-1741-49	CCD: Command value data container 2, Slave 1 4Byte , slave n (n = 1-9)
P-0-1750	CCD: Command value data container 3, Master 4Byte
P-0-1751-59	CCD: Command value data container 3, Slave 1 4Byte , slave n (n = 1-9)
P-0-1760	CCD: Command value data container 4, Master 4Byte
P-0-1761-69	CCD: Command value data container 4, Slave 1 4Byte , slave n (n = 1-9)
P-0-1770	CCD: Actual value data container 1, Master 4Byte
P-0-1771-79	CCD: Actual value data container 1, Slave 1 4Byte , slave n (n = 1-9)
P-0-1780	CCD: Actual value data container 2, Master 4Byte
P-0-1781-89	CCD: Actual value data container 2, Slave 1 4Byte , slave n (n = 1-9)
P-0-1790	CCD: Actual value data container 3, Master 4Byte
P-0-1791-99	CCD: Actual value data container 3, Slave 1 4Byte , slave n (n = 1-9)
P-0-1800	CCD: Actual value data container 4, Master 4Byte
P-0-1801-09	CCD: Actual value data container 4, Slave 1 4Byte , slave n (n = 1-9)
P-0-1820.0.1	CCD: Master axes configuration list
P-0-1820.0.2	CCD: Source of measuring encoder position feedback value
P-0-1820.0.3	CCD: Actual position value of measuring encoder

Tab. 8-87: *Product-specific parameters for the cyclic channel of the MLD control*

9 RMB_TechCrossCutCrossSeal.library // MX_TechCrossCutCrossSeal

9.1 Introduction and overview

The "RMB_TechCrossCutCrossSeal" library contains function blocks for cross sealer and cross cutter applications.



The library requires the **TEC (technology) function package** for the control system.

RMB_TechCrosscutCrossseal function blocks

Area of application (folder in library)	
Function block	Description
Cross sealer and cross cutter applications (CrossCutSeal)	
MB_CrossCutSealType01, page 605	calculates, loads and activates a movement profile for cross sealer or cross cutter applications
Cross sealer and cross cutter calculations (CrossCutter/Calculation)	
MB_CrossCutterCalcType04, page 619	provides all required data that has to be loaded into the movement profile of an axis to operate a cross cutter or cross sealer with the desired parameters at its outputs
MB_CrossCutterCalcType05, page 627	calculates a cam for the application as cross cutter for use in an axis with interpolation in the drive. The cross cutter functionality can be configured.

Tab. 9-1: Overview on function block support by the library

9.2 MB_CrossCutSealType01

Brief description The function block MB_CrossCutSealType01 calculates, loads and activates a MotionProfile for CrossSealer or CrossCutter applications. The function block internally uses the function block [MB_CrossCutterCalcType04, page 619](#).

With the calculated MotionProfile, the slave axis moves with product velocity in the sealing or cutting area. Other functions such as OverSpeed, PushOut or cosine correction can be optionally added in this area.

Outside the sealing or cutting area, an almost optimum compensating motion (with regard to acceleration and energy loss) is performed if the specified format length does not correspond to the distance of the knife or the sealing jaw. Furthermore, a pendular motion of the CrossSealer or CrossCutter axis in freely selectable limits can be predefined to optimally use the acceleration capacity of the drive in case of large format lengths.

At runtime, input variables (such as the format length) can be changed to automatically calculate a new MotionProfile in the background. The newly calculated MotionProfile is enabled and becomes effective at the switching phase.

At a negative edge at "Enable", either the CrossSealer or the CrossCutter are disabled. The axis is immediately stopped via the predefined deceleration ramp.

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechCrosscutCrossseal
MLD with MPx07/08/17 with MA function package	MX_Technology07.lib

Tab. 9-2: Reference table of the MB_CrossCutSealType01

Interface description

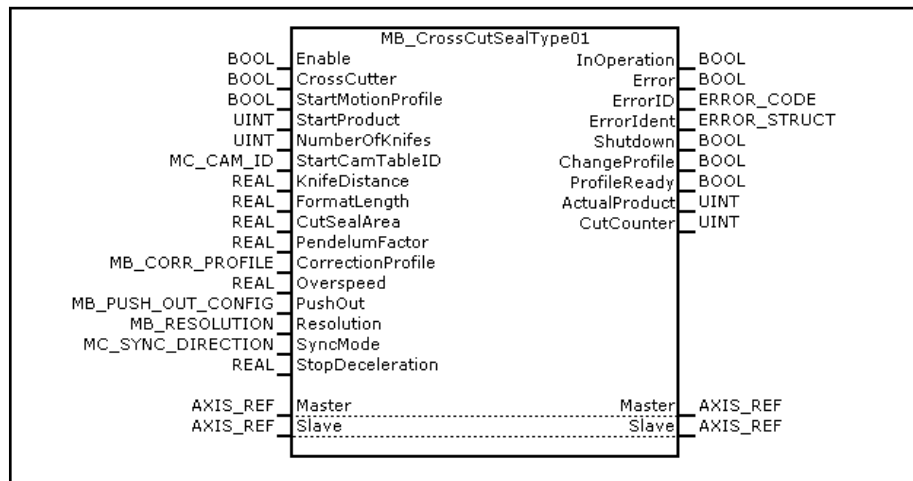


Fig. 9-1: MB_CrossSealType01 function block

I/O type	Name	Type	Comment
VAR_INPUT	Enable	BOOL	Processing of function block enabled
	CrossCutter	BOOL	If "CrossCutter" = TRUE, the CrossCutter use case (see the following detailed description) is selected. In this use case, the master axis has to be standardized so that the material to be cut covers the distance that corresponds to the knife distance in one revolution. The electronic gear of the MotionProfile is set to the gear ratio "FormatLength/KnifeDistance". Furthermore, the input "CutSealArea" functions as the cut angle of the CrossCutter whereas the input variables "ProductsPerCut" and "StartProduct" are not applied. If the "CrossCutter" = FALSE, the CrossSealer use case (see the following detailed description) is selected. In this use case, the master axis has to be standardized so that one revolution corresponds to a foil feed of one product length. The electronic gear of the MotionProfile sets the gear ratio "1/ProductsPerCut". Furthermore, the input "CutSealArea" corresponds to the sealing length and is thus defined in translatory units (e.g. mm or inch). The inputs "ProductsPerCut" and "StartProduct" are evaluated in this case

I/O type	Name	Type	Comment
	StartMotionProfile	BOOL	The activation of a new MotionProfile is blocked as long as this input is set to FALSE. "StartMotionProfile" = TRUE enables the new Motion-Profile if the output "ProfileReady" is TRUE
	ProductsPerCut	UINT	Input is only effective if "CrossCutter" = FALSE. Number of products per sealing process. Corresponds to the number of master revolutions per sealing or cutting process
	StartProduct	UINT	Input is only effective if "CrossCutter" = FALSE. If there is more than one product per sealing or cutting process ("ProductsPerCut" > 1), the start product can be defined via this input when enabling the function block
	NumberOfKnives	UINT	Number of sealing jaws or knives available on the circumference of the sealing or cutting cylinder. The sealing jaws or knives are positioned with same distance from each other
	StartCamTableID	MC_CAM_ID	Starting number of the first cam point table. A block of up to 6 cam tables is required
	KnifeDistance	REAL	Distance of the knives or sealing jaws. Corresponds to the circumference divided by the number of knives or sealing jaws [mm or in].
	FormatLength	REAL	Predefined cutting length or sealing distance/package length [mm or in]
	CutSealArea	REAL	In this area, the knife or the sealing jaw moves with product velocity. The velocity in this window can be affected by additional inputs such as "OverSpeed", "PushOut" or cosine correction. If the input "CrossCutter" = FALSE, this input corresponds to the sealing length and is thus defined in translatory units (e.g. in mm or inch). If the input "CrossCutter" = TRUE, this input acts rotatorily as angle in [°]
	PendulumFactor	REAL	Maximum pendulum radius in [%] 0%: No pendular motion (standstill at large format length) 100%: Maximum pendular motion (pendular motion possible up to the edges of the sealing area or cutting area)
	CorrectionProfile	MB_CORR_PROFILE	Special profile selection. The inputs "OverSpeed" or "PushOut" have to be used for the parameterization of the special profile MB_CORR_PROFILE, page 635

I/O type	Name	Type	Comment
	Overspeed	REAL	Constant overspeed or reduction of the master axis velocity in the cutting area in [%] 0%: No overspeed 100%: Double master axis velocity -50% : Half the master axis velocity This input is only effective if the option CORR_PROFILE_OVERSPEED is selected at the input "CorrectionProfile"
	PushOut	MB_PUSH_OUT_CONFIG	Structure to define a "PushOut" motion during the cut or the sealing process. MB_PUSH_OUT_CONFIG, page 636.. In the CorrectionProfile, either the option CORR_PROFILE_PUSH_OUT or CORR_PROFILE_COS_COMP_AND_PUSH_OUT has to be selected. Otherwise, "PushOut" is deactivated and thus ineffective
	Resolution	MB_RESOLUTION	The resolution of the cam calculation is specified by determining the number of data points in three stages. MB_RESOLUTION, page 637
	SyncMode	MC_SYNC_DIRECTION	Synchronizes the direction (Shortest_Way, Catch_Up, Slow_Down)
	StopDeceleration	REAL	At a negative edge at "Enable", the slave axis is stopped with the deceleration defined. Units according to axis scaling in rotary units
VAR_OUTPUT	InOperation	BOOL	CrossCutter runs without any faults
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in case of error
	ErrorIdent	ERROR_STRUCT	See Troubleshooting, page 618
	Shutdown	BOOL	The CrossSealer is stopped with the deceleration ramp "StopDeceleration" after a negative edge at "Enable". During this time, the output "ShutDown" is enabled to display the defined stop
	ChangeProfile	BOOL	If input variables (such as the "FormatLength") are changed at runtime, a new MotionProfile is calculated and loaded in the background. Subsequently, the new MotionProfile is used. Since this process requires several cycles, this output is enabled for this period. If any other input variables are changed during this process (e.g. "FormatLength"), the current process is interrupted and performed again with new data. The output is reset as soon as the newly calculated MotionProfile becomes effective and thus the switching phase was passed

I/O type	Name	Type	Comment
	ProfileReady	BOOL	This output is set as soon as the MotionProfile was calculated, loaded and enabled. This output is deleted as soon as the MotionProfile becomes effective (after having passed the switching phase).
	ActualProduct	UINT	If there is more than one product per sealing process ("ProductsPerCut" > 1), the currently processed product is displayed via this output. At a rising edge at "Enable", this output is reset to the input variable "StartProduct".
	CutCounter	UINT	The "CutCounter" increases with each modulo overflow ($360^\circ \rightarrow 0^\circ$) of the slave axis. At a rising edge at "Enable", the "CutCounter" is reset to zero
VAR_IN_OUT CONSTANT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to slave axis (= CrossCutter/CrossSealer) The modulo value is 360° electrical and is mechanically defined as the distance between two knives/sealing jaws

Tab. 9-3: Interface variables of the MB_CrossSealType01 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
CrossCutter	BOOL			FALSE	Rising edge at "Enable"
StartMotionProfile	BOOL			TRUE	Continuous
ProductsPerCut	UINT	1	32	1	Rising edge at "Enable"
StartProduct	UINT	1	32	1	Rising edge at "Enable"
NumberOfKnives	UINT	1	8	1	Rising edge at "Enable"
StartCamTableID	MC_CAM_ID	1	94 (for MLC) 3 (for MLD)	1	Rising edge at "Enable"
KnifeDistance	REAL	> 0.0 mm/in	n. def	100.0mm/in	Rising edge at "Enable"
FormatLength	REAL	> 0.2* KnifeDistance	30* KnifeDistance	100.0mm/in	Continuous*
CutSealArea	REAL	> 0° > 0mm	< 360° < KnifeDistance	10° 10mm/in	Continuous*
PendelumFactor	REAL	0.0%	100%	0.0%	Continuous*

Name	Type	Min. value	Max. value	Default value	Effective
CorrectionProfile	MB_CORR_PROFILE	0	4	0	Continuous*
Overspeed	REAL	- 50.0%	+100.0%	0.0%	Continuous*
PushOut	MB_PUSH_OUT_CONFIG				Continuous*
Resolution	MB_RESOLUTION	RESOLUTION_LOW	RESOLUTION_HIGH	RESOLUTION_MEDIUM	Rising edge at "Enable"
SyncMode	MC_SYNC_DIRECTION	0	2	Catch_Up	Rising edge at "Enable"
StopDeceleration	REAL	> 0.0	n. def	1000.0	Continuous

* Applying the inputs takes several cycles and is displayed at the "ChangeProfile" output

Tab. 9-4: Min./max. and default values of the MB_CrossSealType01 function block

Functional description

After the processing was enabled via **Enable**, the function block calculates a MotionProfile as well as cam point tables for the CrossSealer/CrossCutter application according to the specified input data. The function block also supports multiple CrossSealers/CrossCutters with equally distributed knives or sealing jaws.



For multiple CrossCutters or CrossSealers:

For multiple CrossCutter or CrossSealer applications, the parameterization of the mechanical gear has to be adjusted (see also ["Necessary boundary conditions and requirements" on page 617](#)).

Therefore, the calculated motion sequence is divided into the following two sections:

- The **cutting or sealing area** is defined by the input variable **CutSealArea**. In this area, the knife or the sealing jaw moves with the product velocity. In addition, other functions such as **OverSpeed** for overspeed or a reduction of the velocity, **PushOut** for a push out motion or the **cosine correction** can be optionally added.

In this area, the following input variables become effective:

- **CorrectionProfile**
Provides special profiles in the cutting and sealing area (OverSpeed, PushOut, cosine correction, PushOut + cosine correction).
- **OverSpeed**
Specifies the increase or reduction of velocity in **CutSealArea** [%]
- **PushOut**
Defines a special PushOut motion during the cutting or sealing process

- **Outside** the cutting or sealing area, an optimum compensating motion with regard to jerk and energy is performed if the specified format length **FormatLength** does not correspond to the distance of the knives or the sealing jaws **KnifeDistance**. Furthermore, a **pendular motion** of the CrossSealer axis or CrossCutter axis can be specified in adjustable lim-

its to optimally use the acceleration capacity of the drive for large format lengths.

The following input variable becomes effective in this area:

- **PendelumFactor** specifies the maximum admissible pendulum radius [%]

The MotionProfile and the cam point tables are automatically loaded and enabled after the calculation. During this period, the output **ChangeProfile** is set. If the MotionProfile is effective and if the slave axis is synchronized, the output **InOperation** is activated. In this state, the slave axis moves synchronously to the master axis using the previously calculated MotionProfile. In this state, the cut or seal counter **CutCounter** is increased with each revolution of the CrossCutter axis or CrossSealer axis.

If input variables such as **FormatLength** are changed at runtime, a new MotionProfile is calculated in the background. Since this process requires several cycles, the output **ChangeProfile** is set during this period. The output **ProfileReady** is set if the new MotionProfile has been enabled, but the switching phase has not yet been passed. If the switching phase is reached subsequently, the calculated MotionProfile becomes effective and the outputs **ChangeProfile** and **ProfileReady** are reset. If any input data is changed again during the current change of the MotionProfile, the changes are discarded and the calculation with the new input data is started.

The data points of the cam point tables used in the MotionProfile can be influenced via the **Resolution** input. A high resolution results in a maximum accuracy but requires most of the time during the calculation. The time effort can be reduced by the factor 3 or 9 using the settings "Medium" and "Low". The reduction of the resolution is useful if different product lengths are continuously processed with high cycle rates (> 500 cycles/min).

A negative edge at **Enable** results in a defined shutdown of the CrossCutter/ CrossSealer. The slave axis is decelerated until standstill with the adjustable ramp **StopDeceleration**. During the defined shutdown, the output **Shutdown** is set. The output **Shutdown** is reset if the slave axis has reached its standstill position or if an error occurred.

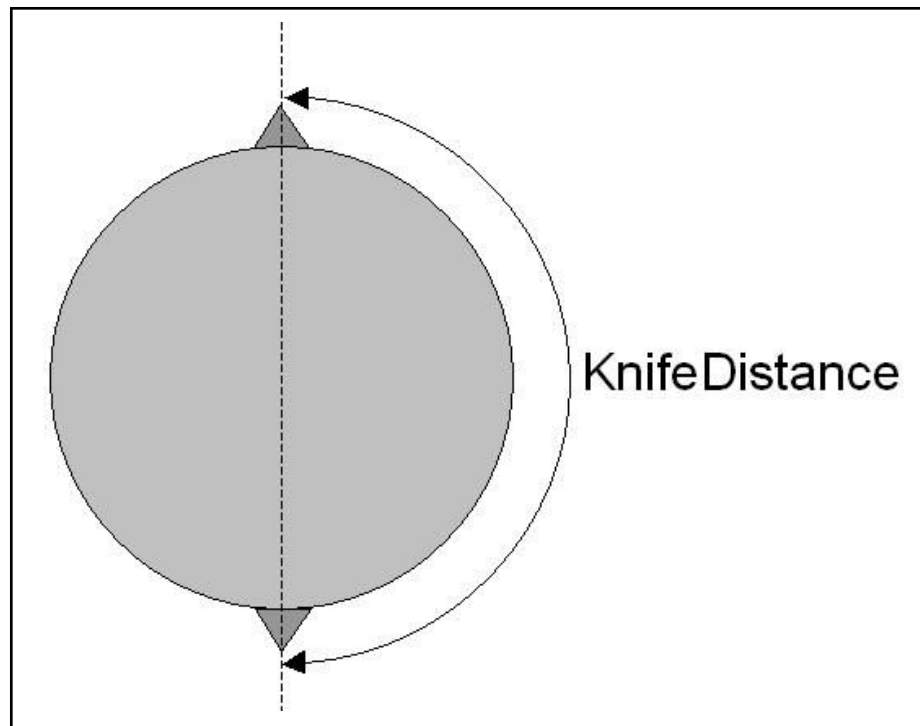


Fig. 9-2: Definition of the input "KnifeDistance"



For scaling:

The translatable input variables of the function block (e. g. FormatLength) can be specified either in [mm] or in inch [in]. However, it has to be considered that all translatable input variables are specified in the same unit (mm or inch).

CrossSealer use case

Via the **CrossCutter** input, the function block generally differentiates between the CrossCutter application and the CrossSealer use case:

If the input **CrossCutter** is FALSE, a cross sealer application is assumed. Triggering a cross sealer cylinder of flow wrappers is a typical example of this use case. The sealing jaw should move with the product velocity and with an optimum correction profile during the sealing process. Outside the sealing range, a compensating motion is to be performed if the format length **FormatLength** does not correspond to the distance of the sealing jaws **KnifeDistance**. Furthermore, the CrossSealer can be optionally controlled by an upstream, cycle-synchronous lock on and lock off motion to realize the functions "NoGap/NoSeal" and "NoProduct/NoBag".

The CrossSealer use case provides the following characteristics:

- The **CutSealArea** input corresponds to the length of the sealing seam and is thus specified in translatable units (mm or inch)
- One master axis revolution (360°) corresponds to the foil feed of one product length
- One sealing process can include several products. The number of products per sealing process is specified via the **ProductsPerCut** input. This input directly influences the electronic gear of the MotionProfile
- The first product is set via the **StartProduct** input after a positive edge at "Enable". This variable internally affects a position offset of the MotionProfile used. After a positive edge at "Enable", the output **ActualProduct** is set to the input **StartProduct**

- Optionally, the cycle-synchronous lock on and lock off can be connected upstream via another the MB_CamLock function block



To start the function block:

If the master axis and the slave axis are at any position at the function block start, the slave axis performs a dynamic synchronization to synchronize with the master axis. If this is not desired, the master axis and the slave axis have to be set either to the position 0° or 180° prior to the start of the function block.

The following figure shows the master axis structure and the effects of the function block for the CrossSealer use case with optional, cycle-synchronous lock on and lock off:

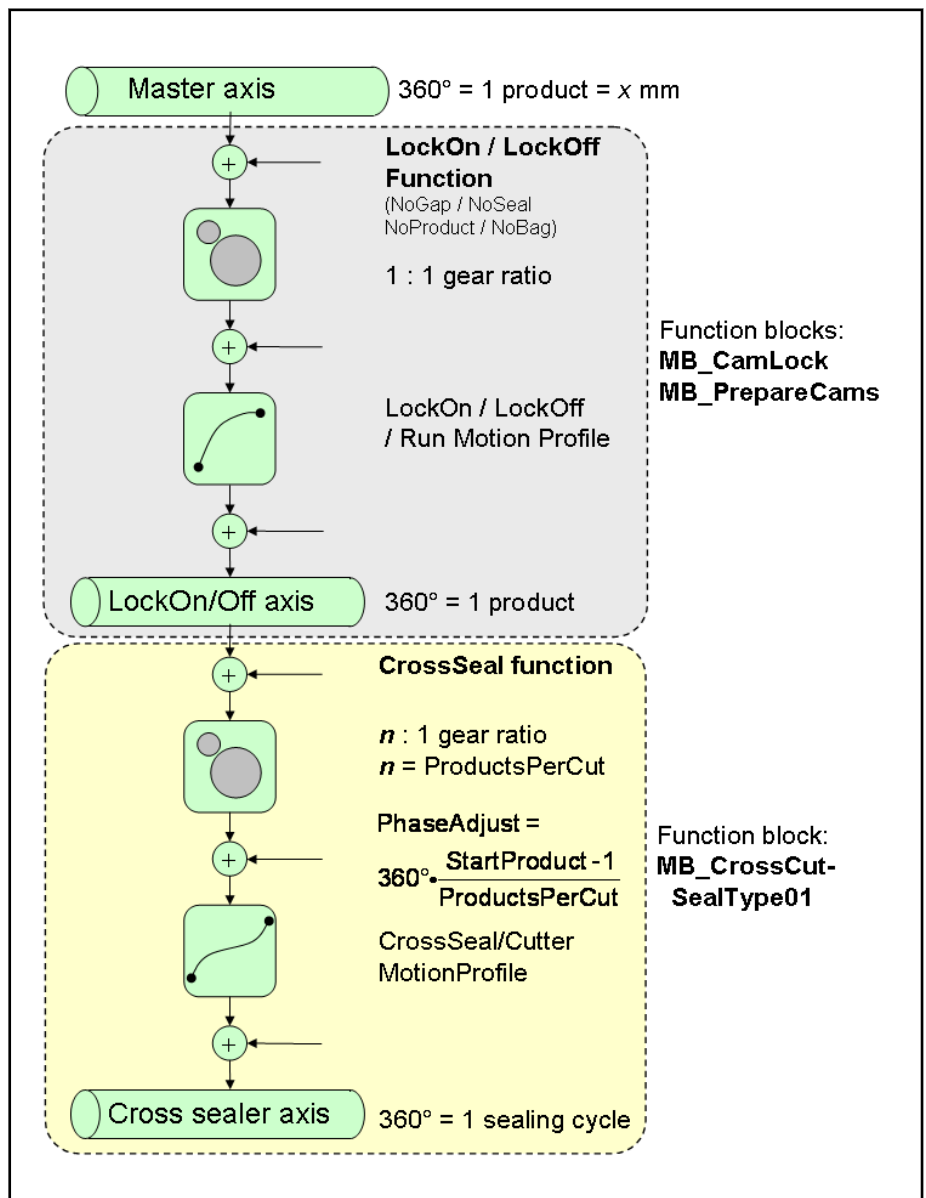


Fig. 9-3: Master axis structure for CrossSealer applications with lock on/lock off

The following figure shows an example of a signal-time diagram of the master axis and the CrossSealer axis for one product per package

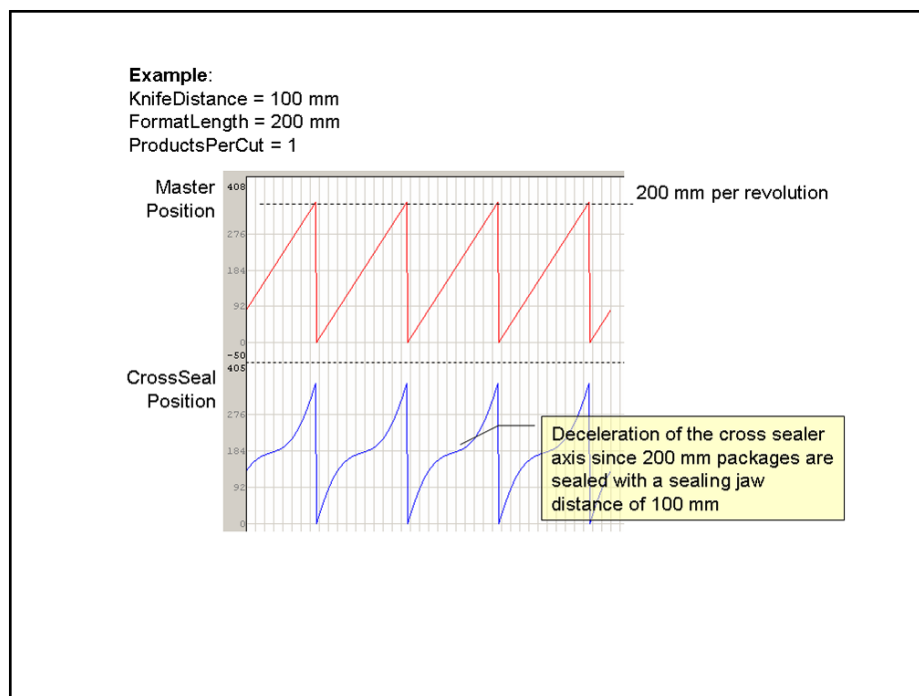


Fig. 9-4: Example of a signal-time diagram with one product per sealing process

The following figure shows an example of a signal-time diagram of the master axis and the CrossSealer axis for 3 products per package:

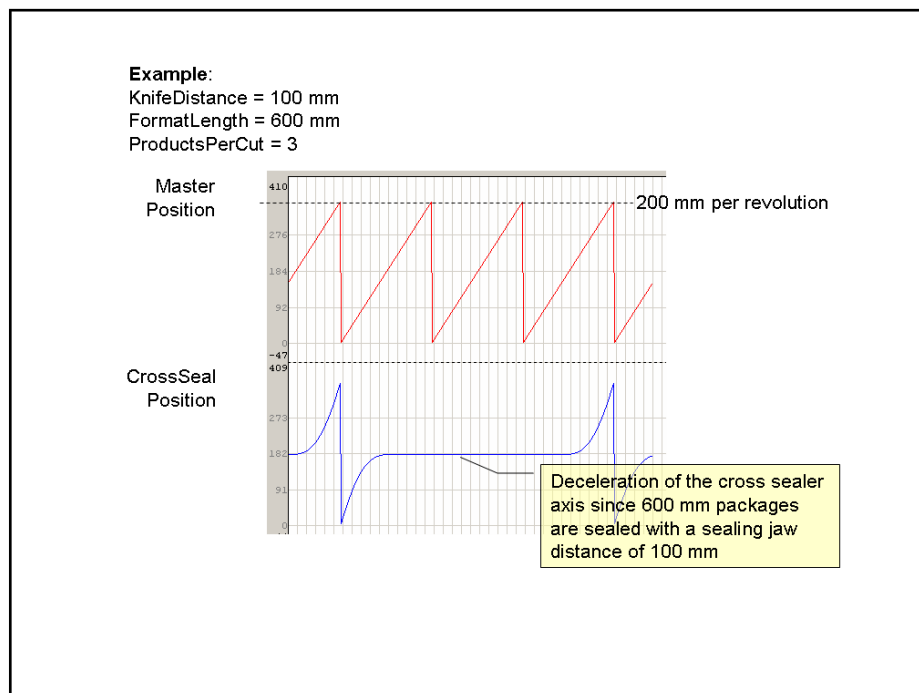


Fig. 9-5: Example of a signal-time diagram with 3 products per sealing process

CrossCutter use case

Via the **CrossCutter** input, the function block generally differentiates between the CrossCutter application and the CrossSealer use case:

If the input **CrossCutter** is TRUE, a CrossCutter application is assumed. During the cutting process, the knife should move with the velocity of the product and with an optional correction profile. Outside the cutting area, a compen-

sating motion is to be performed if the format length **FormatLength** does not correspond to the distance of the sealing jaws **KnifeDistance**.

The CrossCutter use case provides the following characteristics:

- The input **CutSealArea** corresponds to the angular range of the Cross-Cutter in which it is traveling with the velocity of the product and the optional correction profile
- One revolution of the master axis corresponds to the material feed length which corresponds to the knife distance **KnifeDistance**
- One cut per product is assumed. The inputs **ProductsPerCut** and **Start-Product** are not effective (these inputs is internally set to 1)
- The output **ActualProduct** is always set to "1"
- The electronic gear of the MotionProfile is set to the gear ratio **FormatLength/KnifeDistance**.

This gear only allows integer values from 1 to 65535 as numerator and denominator of the gear ratio. Thus, there is a maximum cutting length deviation due to quantization.

If a deviation of the cutting lengths occurs due to quantization, a respective gear ratio fine adjustment is calculated and specified. In this case, a drift between the master and the slave axis results due to the system conditions.

The following figure shows the master axis structure and the effects of the function block for the CrossCutter application:

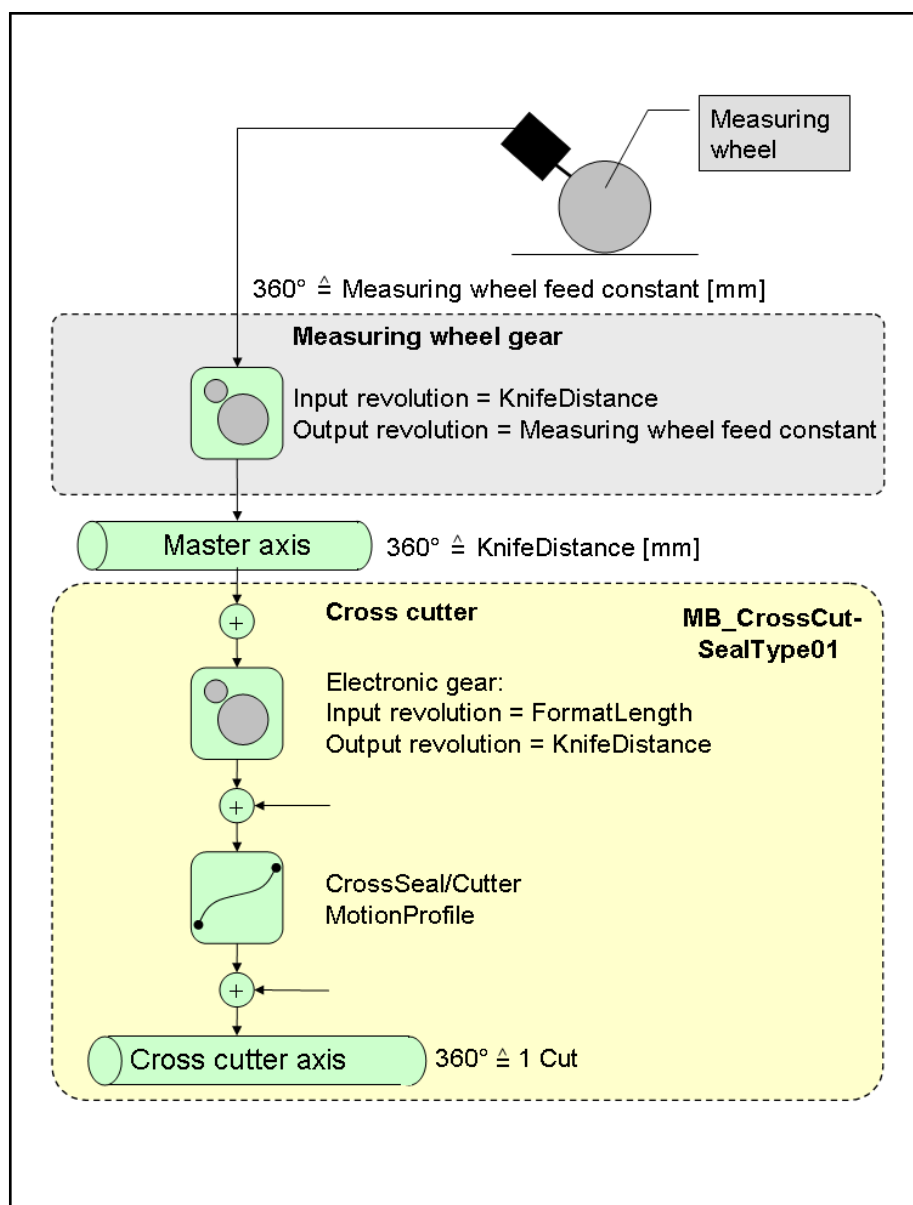


Fig. 9-6: Master axis structure for the CrossCutter application

The following figure shows an example of a signal-time diagram of the measuring wheel axis, the master axis and the CrossCutter axis for a short format (format length = 0.5* knife distance):

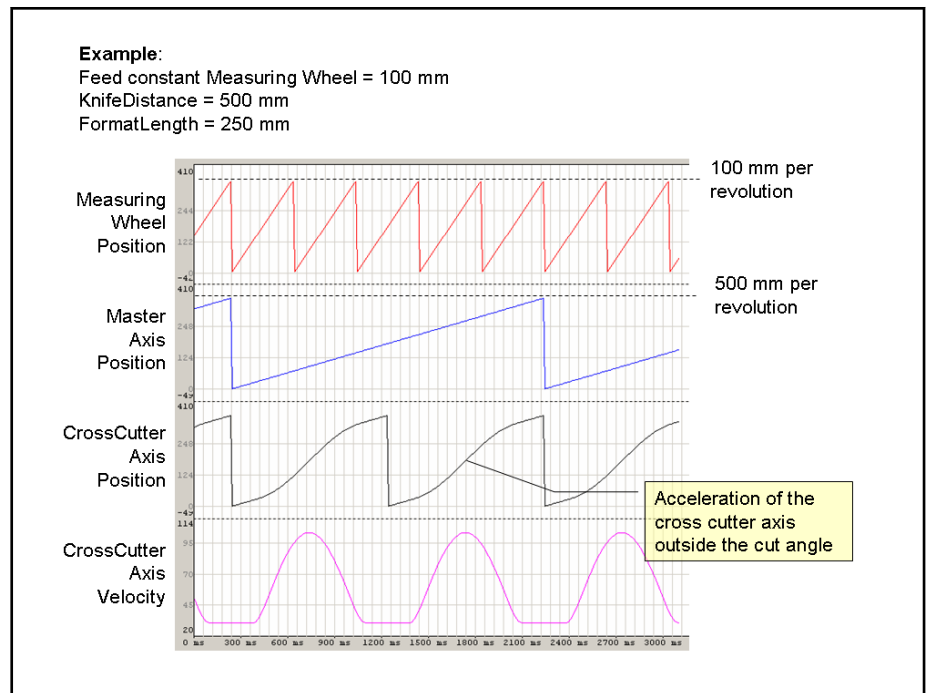


Fig. 9-7: Example of a signal-time diagram for a format length = $0.5 \cdot$ knife distance

The following figure shows an example of a signal-time diagram of the measuring wheel axis, the master axis and the CrossCutter axis for a long format (format length = $2 \cdot$ knife distance):

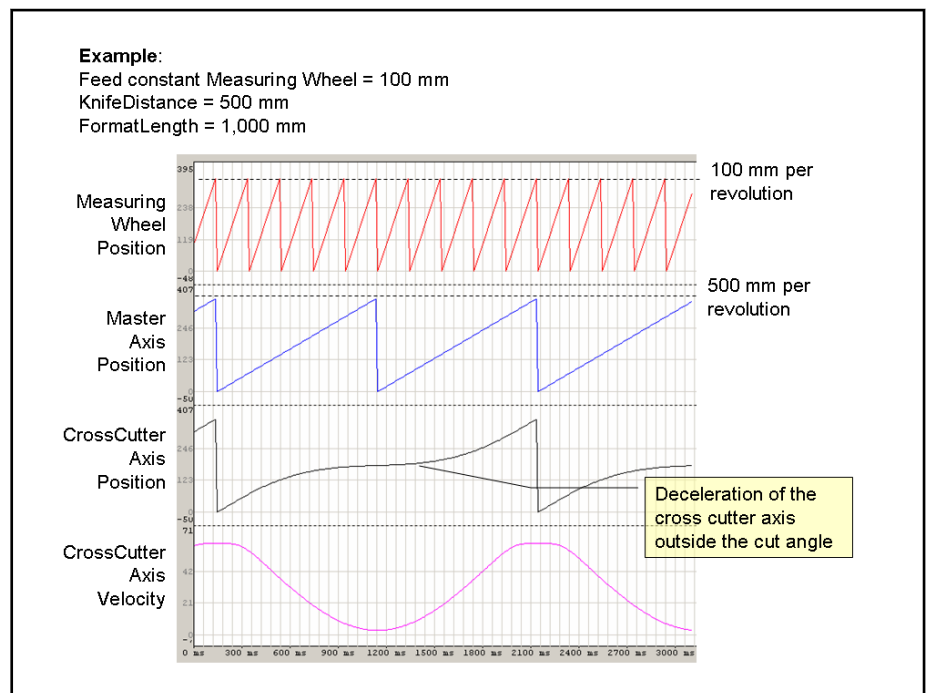


Fig. 9-8: Example of a signal-time diagram for a format length = $2 \cdot$ knife distance

Necessary boundary conditions and requirements

- The sealing jaws/knives are symmetrically distributed over the circumference
- The 0° position of the slave axis is always exactly in the middle of the CutSealArea

**For drive parameterization:**

If there is more than one sealing jaw or knife per circumference, the settings for the mechanical gear (drive parameters: S-0-0121, S-0-0122) have to be adapted. These parameters have to be set in a way so that a 360° electric (modulo value of the slave axis) corresponds to the distance between two knives or sealing jaws.

Example of a gear configuration

The configuration of the mechanical gear when using a CrossCutter with four equidistantly distributed knives on the CrossCutter axis

Input variables:

Mechanical gear at the motor, input revolutions $n_{\text{input}} = 10$

Mechanical gear at the motor, output revolutions $n_{\text{output}} = 1$

Number of knives per circumference $n_{\text{knife}} = 4$

Calculation of the required gear setting:

$$\text{Mechanical gear} = \frac{\text{Input revolutions}}{\text{Output revolutions}} =$$

$$= \frac{n_{\text{input}}}{n_{\text{output}} \cdot n_{\text{knife}}} = \frac{10}{1 \cdot 4} = \frac{10}{4}$$

- S-0-0121 = 10
- S-0-0122 = 4

If, with this parameterization, the cutting cylinder moves mechanically by 90°, this corresponds to an electrical movement of 360° (= modulo value of the slave axis).

Troubleshooting

The function block does not perform an independent, motion-side error reaction. If an error is detected, no further Motion command is transmitted to the axis and the function block is exited via the **Error** output. A negative edge at "Enable" deletes the error output and the error state in the function block.

The function block uses the error table F_RELATED-TABLE (16#0170).

ErrorID	Additional1	Additional2	Description
ACCESS_ERROR (16#0004)	16#0003	16#0000	The function block was aborted by another function block
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0001	The "ProductsPerCut" input is not in the permitted range (1...32)
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0002	The "StartProduct" input is not in the permitted range (1...32)
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0003	The "NumberOfKnives" input is not in the permitted range (1...8)
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0004	The "StartCamTableID" input is not in the permitted range (1...94)
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0005	The "KnifeDistance" input is not in the permitted range

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0006	The "FormatLength" input is not in the permitted range
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0007	The "CutSealArea" input is not in the permitted range
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0008	The "PendelumFactor" input is not in the permitted range
INPUT_RANGE_ERROR (16#0006)	16#0620	16#0009	The "Overspeed" input is not in the permitted range
INPUT_RANGE_ERROR (16#0006)	16#0620	16#000A	The "Resolution" input is not in the permitted range
INPUT_RANGE_ERROR (16#0006)	16#0621	16#0000	"CutSealArea" is too small with regard to the master axis velocity and the Sercos cycle time

Tab. 9-5: Error codes of the MB_CrossCutSealType01 function block

9.3 MB_CrossCutterCalcType04

Brief description The MB_CrossCutterCalcType04 function block provides all required data at its outputs which have to be loaded into the MotionProfile of an axis to operate a CrossCutter or a CrossSealer with the functions selected in the parameters of the function block. All inputs of the function block are transferred once at a rising edge at "Execute". In addition, the function block writes directly into the arrays specified by the pointers during its runtime. The written data is only valid if the function block was processed without errors and if it returns "Done".

Assignment: Target system/library

Target system	Libraries
MLC 14VRS TEC function package	RMB_TechCrosscutCrossseal
MLD with MPx07/08/17 with MA function package	MX_Technology07.lib

Tab. 9-6: Reference table of the MB_CrossCutterCalcType04

Interface description

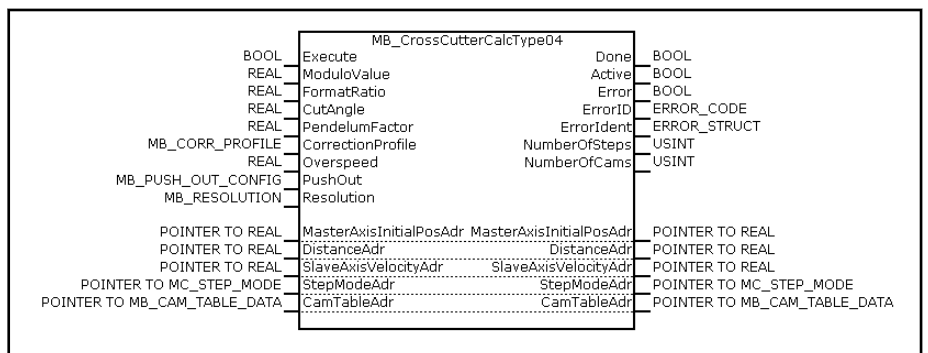


Fig. 9-9: MB_CrossCutterCalcType04 function block

I/O type	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Enabling the function block (once, edge-controlled)
	ModuloValue	REAL	Modulo value of the CrossCutter axis in [°]
	FormatRatio	REAL	Format ratio of the product to be cut (without unit)
	CutAngle	REAL	Length of the cutting area in [°]

I/O type	Name	Type	Comment
	PendelumFactor	REAL	Maximum pendulum radius in [%] (0%: No pendular motion, 100%: Maximum pendular motion (pendular motion up to the cutting area limits)
	CorrectionProfile	MB_CORR_PROFILE	Special profile selection. The inputs "Overspeed" or "PushOut" have to be used for the parameterization of the special profile. MB_CORR_PROFILE, page 635
	Overspeed	REAL	Constant overspeed or velocity reduction in the cutting area [%] of the master axis velocity. (0%: No overspeed 100%: Double master axis velocity -50%: Half the master axis velocity) In the "CorrectionProfile", the CORR_PRO- FILE_OVERSPEED option has to be selected. Other- wise, "Overspeed" is deactivated and thus ineffective
	PushOut	MB_PUSH_OUT_CON- FIG	Structure for a PushOut motion definition during the cut. In the "CorrectionProfile", either the option CORR_PROFILE_PUSH_OUT or CORR_PRO- FILE_COS_COMP_AND_PUSH_OUT has to be selec- ted. Otherwise, "PushOut" is deactivated and thus in- effective. MB_PUSH_OUT_CONFIG, page 636
	Resolution	MB_RESOLUTION	The resolution of the cam calculation is specified by determining the number of data points in three stages. MB_RESOLUTION, page 637
VAR_OUTPUT	Done	BOOL	Processing completed without errors, output data is valid
	Active	BOOL	Processing not yet completed, output data invalid
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, the output contains detailed error information Error handling, page 625
	NumberOfSteps	USINT	Number of generated and to be downloaded Motion- Profile segments
	NumberOfCams	USINT	Number of generated and to be downloaded cams
VAR_IN_OUT	MasterAxisInitialPo- sAdr	POINTER TO REAL	Pointer to an array with three elements to be created by the user in which the function block stores the starting positions of the master axis of the generated MotionProfile segments

I/O type	Name	Type	Comment
	DistanceAdr	POINTER TO REAL	Pointer to an array with three elements to be created by the user in which the function block stores the distances of the generated MotionProfile segments
	SlaveAxisVelocityAdr	POINTER TO REAL	Pointer to an array with three elements to be created by the user in which the function block stores the velocities of the slave axis at the segment limits of the MotionProfile segments
	StepModeAdr	POINTER TO MC_STEP_MODE	Pointer to an array with three elements to be created by the user in which the function block stores the motion step modes of the generated MotionProfile segments.
	CamTableAdr	POINTER TO MB_CAM_TABLE_DATA	Pointer to an array with three elements to be created by the user in which the function block stores cam tables as well as meta data of the cams which are inserted as segments in the MotionProfile MB_CAM_TABLE_DATA, page 633

Tab. 9-7: MB_CrossCutterCalcType04 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
ModuloValue	REAL	>0.0°	n.def.	360.0°	At a rising edge at "Execute"
FormatRatio	REAL	0.2	30.0	1.0	At a rising edge at "Execute"
CutAngle	REAL	0.0°	< "ModuloValue"	12.0°	At a rising edge at "Execute"
PendelumFactor	REAL	0.0%	100.0%	0.0%	At a rising edge at "Execute"
CorrectionProfile	MB_CORR_PROFILE			CORR_PROFILE_NONE	At a rising edge at "Execute"
Overspeed	REAL	-50.0%	100.0%	0.0%	At a rising edge at "Execute"
PushOut	MB_PUSH_OUT_CONFIG				At a rising edge at "Execute"
Resolution	MB_RESOLUTION			RESOLUTION_HIGH	At a rising edge at "Execute"

Tab. 9-8: Min./max. and default values of the MB_CrossCutterCalcType04 function block

Signal-time diagram

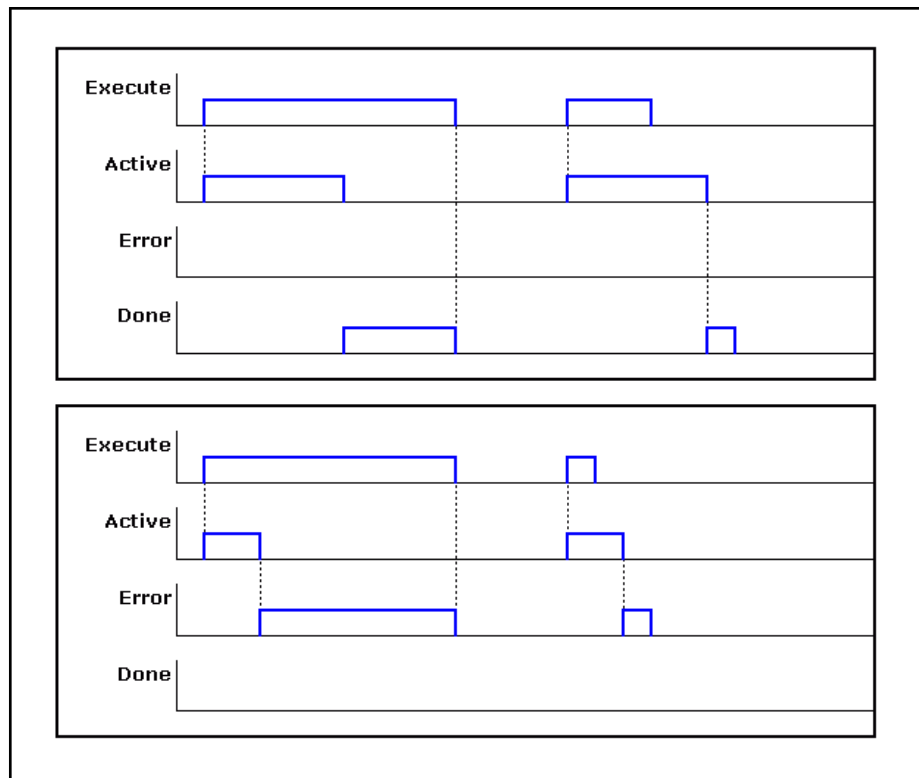


Fig. 9-10: Signal-time diagram of the MB_CrossCutterCalcType04 function block

Functional description

Based on the operation mode "Electronic MotionProfile", the function block calculates profiles for the "CrossCutter" application. The individual segments consist either of analytical segments or cam tables. After having enabled the processing with "Execute", an "Electronic MotionProfile" consisting of three segments is calculated once based on the input variables. The parameters describing the profile are saved in the respective arrays which have to be provided by pointers at the inputs of the "MB_CrossCutterCalcType04" function block. The arrays have to be large enough to be able to save the number of calculated motion steps (currently three steps). The pointer handling is similar to the "MB_ChangeProfileStep"/"MB_ChangeProfileSet" function blocks since these function blocks have to be called by the user after the calculation to write the calculated profile to the respectively free set of the "Electronic MotionProfile".

The following list describes the meaning of the individual inputs of the "MB_CrossCutterCalcType04" function block in detail:

- The "ModuloValue" input specifies the modulo value of the CrossCutter axis. 360.0° are specified by default
- The "FormatRatio" input specifies the product length to be cut. It describes the ratio of the desired product length and the circumference of the CrossCutter cylinder. This means that if "FormatRatio==1", the length of the cut product corresponds to the circumference of the CrossCutter cylinder (synchronous format). If "FormatRatio==5", the length of the product cut is five times the circumference of the CrossCutter cylinder
- The "CutAngle" input specifies the cutting area of the CrossCutter. In the cutting area, the CrossCutter either rotates with the material feed velocity or with a (selectable) correction motion

- The "PendelumFactor" input specifies the maximum pendular motion of the CrossCutter axis in percent. This means that if "PendelumFactor==0%", the CrossCutter axis never rotates with a negative velocity (backwards). If "PendelumFactor==100%", the CrossCutter axis rotates to the border of the cutting area to minimize the energy loss and the required acceleration. This provides special benefits in case of higher values for "FormatRatio". If "PendelumFactor==100%", the CrossCutter does not generally rotate up to the border of the cutting area since this is not always advantageous. The respective optimum utilization of the available area is always automatically calculated by the function block
- The "CorrectionProfile" input allows different correction motions in the cutting zone required for the respective use case. The following can be selected:
 - "CorrectionProfile==CORR_PROFILE_NONE"
(no correction: CrossCutter axis rotates with the master axis velocity)
 - "CorrectionProfile==CORR_PROFILE_OVERSPEED"
(constant overspeed of the CrossCutter axis in the cutting area)
 - "CorrectionProfile==CORR_PROFILE_COS_COMP"
(constant velocity vector in material direction, e. g. for guillotine cutters)
 - "CorrectionProfile==CORR_PROFILE_PUSH_OUT"
(PushOut of the material) as well as
 - "CorrectionProfile==CORR_PROFILE_COS_COMP_AND_PUSH_OUT"
(constant velocity vector in material direction as well as simultaneous PushOut, e.g. for guillotine cutters)
- The "Overspeed" input allows the exact setting of the overspeed in [%] in the cutting area if
"CorrectionProfile==CORR_PROFILE_OVERSPEED"
Thus, entering a value of 10 [%] increases the velocity of the CrossCutter axis in the cutting area by 10% compared to the material velocity
- The input "PushOut" parameterizes the PushOut correction
The input is only processed if
"CorrectionProfile==CORR_PROFILE_PUSH_OUT"
or
"CorrectionProfile==CORR_PROFILE_COS_COMP_AND_PUSH_OUT"
- The "Resolution" input allows the selection of the resolution of the cam tables used in 3 steps. Selecting a lower resolution saves calculation time, but reduces accuracy. High resolution increases the calculation time and the accuracy respectively

The successful calculation of a profile is signaled via the "Done" output. The parameters are written to the data arrays addressed by the provided pointers. Subsequently, these have to be written on a free MotionProfile using "MB_ChangeProfileSet"/"MB_ChangeProfileStep". The same procedure applies for the cams. Ensure that the table into which the cams have to be loaded is defined in the "MB_CAM_TABLE_DATA" structure. Subsequently, the loaded MotionProfile is to be activated by using the "MB_MotionProfile" function block. The exact cutting length is defined with the electronic gear at the "MB_MotionProfile" function block. The gear ratio corresponds to the "Forma-

tRatio" value. That means that a new profile is calculated for each product length possible. Each profile considers one input parameter only (especially "CutAngle" and "PendelumFactor"). The following figure shows a sequential chart describing the use of the MB_CrossCutterCalcType04 function block.

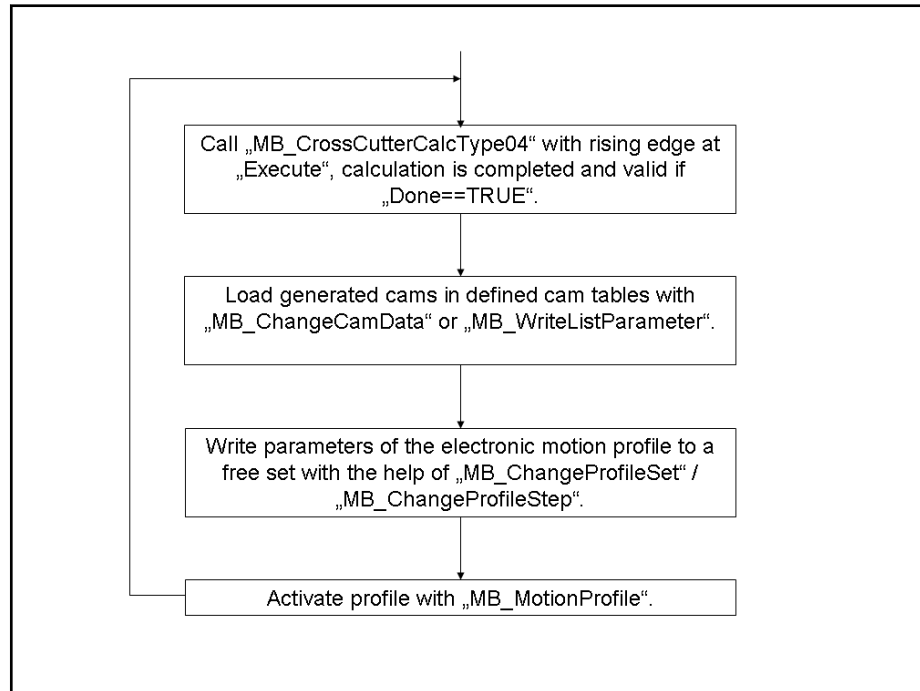


Fig. 9-11: Sequence diagram for using MB_CrossCutterCalcType04

To use several equidistantly distributed knives, a mechanical gear is to be configured in IndraWorks with reference to the load side. The number of input revolutions is set to 1 and the number of output revolutions is set to the number of the knives n used. Since no real mechanical gear exists, the drive only performs one n -fold revolution. Thus, the velocity of the slave axis in one revolution is n times identical to the velocity of the master axis.

Example:

The configuration of the mechanical gear when using a CrossCutter with four equidistantly distributed knives on the CrossCutter axis

Input variables:

Mechanical gear at the motor, input revolutions $n_{\text{input}} = 10$

Mechanical gear at the motor, output revolutions $n_{\text{output}} = 1$

Number of knives per circumference $n_{\text{knife}} = 4$

Calculation of the required gear setting:

$$\text{Mechanical gear} = \frac{\text{Input revolutions}}{\text{Output revolutions}} =$$

$$= \frac{n_{\text{input}}}{n_{\text{output}} \cdot n_{\text{knife}}} = \frac{10}{1 \cdot 4} = \frac{10}{4}$$

- S-0-0121 = 10
- S-0-0122 = 4

If, with this parameterization, the cutting cylinder moves mechanically by 90° , this corresponds to an electrical movement of 360° (= modulo value of the slave axis).

To adjust the master axis velocity and the slave axis velocity in the cutting area to the same value, the feed constant between the measuring wheel and the internal virtual master axis has to be adjusted. Therefore, the circumference of the internal virtual master axis is specified with the length between two knives of the real slave axis. This parameterization permits the cutting of each product with a different length at a corresponding preparation time.

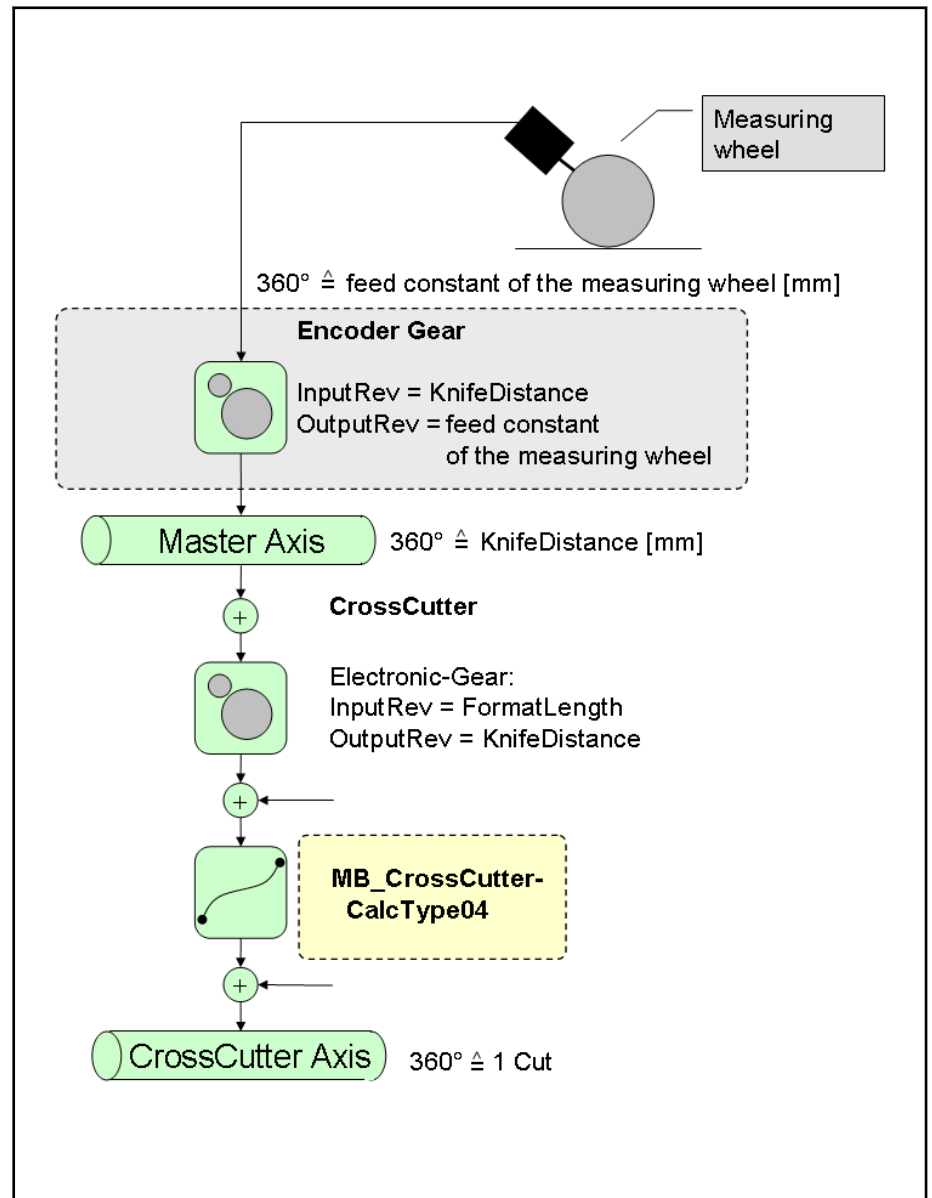


Fig. 9-12: Master axis structure when operating a CrossCutter

Error handling

This function block uses the error table **F_RELATED_TABLE**, 16#170. The function block can generate the following error messages in Additional1/Additional2.

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0001	Invalid value of the input "ModuloValue"
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0002	Invalid value of the input "FormatRatio"
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0003	Invalid value of the input "CutAngle"
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0004	Invalid value of the "CutAngle" input combined with the cosine compensation
INPUT_RANGE_ERROR (16#0006)	16#0610	16#0005	Invalid value of the input "PendulumFactor"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#0006	Invalid value of the input "CorrectionProfile"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#0007	Invalid value of the input "Overspeed"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#0008	Invalid starting point for the PushOut function
INPUT_INVALID_ERROR (16#0001)	16#0610	16#0009	Invalid value for "Overspeed_At_Cut" in the structure "MB_PUSH_OUT_CONFIG"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000A	Invalid value for "Overspeed_Max" in the structure "MB_PUSH_OUT_CONFIG"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000B	Invalid combination of "Overspeed_At_Cut" and "Overspeed_Max"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000C	Value of the "CutAngle" input too small combined with PushOut
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000D	Invalid value of the input "Resolution"
INPUT_INVALID_ERROR (16#0001)	16#0610	16#000E	Invalid values of the "POINTER" inputs
STATE_MACHINE_ERROR (16#0005)	16#0006	xyyy	Invalid state of the state machine xx specifies the invalid step yy specifies the invalid state

Tab. 9-9: Error codes of the MB_CrossCutterCalcType04 function block

Special features of MLC

With regard to the MLC, the following commissioning notes are to be considered when operating the technology function block MB_CrossCutterCalcType04, downloading the generated MotionProfiles and cams and when linking the CrossCutter axis with the master axis:

- The modulo scaling with the modulo value 360° has to be active for the CrossCutter axis
- The MB_MotionProfile function block always operates with the setting "0: Absolute synchronization without resynchronization" for the "Start-Mode" input to ensure the exact exchange of the profiles
- For the electronic gear, the MB_MotionProfile function block only allows integer values from 1 to 65535 as numerator and denominator of the gear ratio. Thus, there is a maximum cutting length deviation due to quantization. This can be minimized by multiplication with a big constant to be able to almost completely use the value range
- The generated MotionProfile can be downloaded on the MLC using the MB_ChangeProfileSet function block and the generated cams can be downloaded using either the MB_ChangeCamData function block or the MB_WriteListParameter function block. In contrast to the MB_WriteListParameter function block, the MB_ChangeCamData function block is faster. However, the cam is not saved persistently and has to be reloaded after a control reboot

When downloading the cams, please make sure that the respective cam is loaded at its assigned position (see "CamTableID" element in the table for the "interface variables of structure [MB_CAM_TABLE_DATA](#), page 633")

- A small format ratio (< 1) should be selected carefully since the Cross-Cutter axis reaches very high velocities in the compensation range (up to eight times the master axis velocity)
- Bit 1 is to be set in the parameter A-0-2930 of the CrossCutter axis so that the gear is exchanged at the same time as the MotionProfile at a switching phase of 0° and not when calling the MB_MotionProfile function block

Format ratio	Ratio of the slave axis velocity to the master axis velocity
1	1
0.8	1.5
0.6	2.2
0.4	3.7
0.2	8

Tab. 9-10: Effects of the format ratio on the maximum velocity of the CrossCutter axis in the compensation range

9.4 MB_CrossCutterCalcType05

Brief description This function block calculates a cam for the application as cross cutter. The cross cutter functionality can be configured.



If the block is to use several knives, only a limited function is available.

For details, refer to [Several knives at one cross cutter on page 631](#).

Assignment target system/library

Target system	Library
MLC 14VRS Function package TEC	RMB_TechCrosscutCrossseal
MLD with MPx20VRS and higher with MA function package	MX_TechCrosscutCrossseal

Tab. 9-11: Reference table MB_CrossCutterCalcType05

Interface description

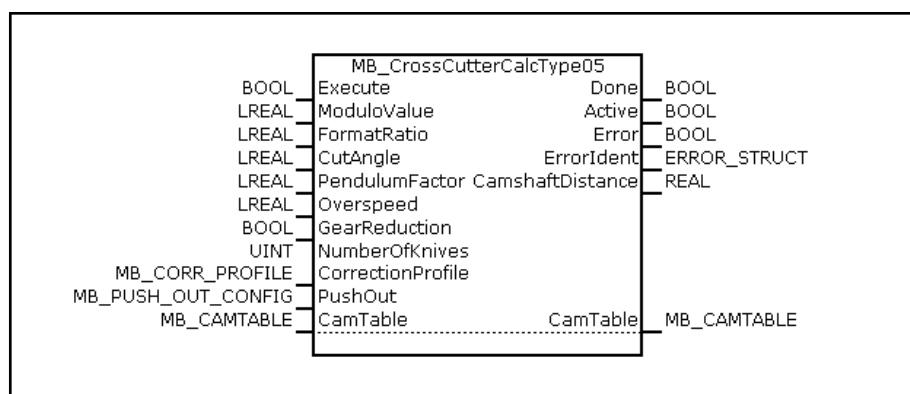


Fig. 9-13: FB MB_CrossCutterCalcType05

	Name	Type	Description
VAR_IN_OUT	CamTable	MB_CAMTABLE	Cam in the MB_CAMTABLE format MB_CAMTABLE on page 97
VAR_INPUT	Execute	BOOL	Function block processing enabled (once, edge-controlled)
	ModuloValue	LREAL	Modulo value of the CrossCutter axis in [°]
	FormatRatio	LREAL	Format ratio of the product to be cut (without unit)
	CutAngle	LREAL	Length of the cutting area in [°]
	PendulumFactor	LREAL	Maximum pendulum radius in [%] 0%: no pendular motion, 100%: maximum pendular motion (pendular motion up to the cutting area limits)
	Overspeed	LREAL	Constant overspeed or velocity reduction in the cutting area [%] of the master axis velocity. (0%: no overspeed, 100%: double master axis velocity, -50%: half the master axis velocity)
	GearReduction	BOOL	Operation with gear reduction: FALSE: no gear reduction, cam starts at 0% and ends at 100% TRUE: with gear reduction, cam starts and ends at 0%
	NumberOfKnives	UINT	This input is only effected if the "GearReduction" input is TRUE. It specifies the number of knives located on the cross cutter.
	CorrectionProfile	MB_CORR_PROFILE	Profile selection. The profiles are used for the different applications, e.g. overspeed during cutting, acceleration during cutting or cosine compensation. For details, refer to MB_CORR_PROFILE on page 635

	Name	Type	Description
	PushOut	MB_PUSH_OUT_CONFIG	Structure for a PushOut motion definition during the cut. In the "CorrectionProfile", either the option CORR_PROFILE_PUSH_OUT or CORR_PROFILE_COS_COMP_AND_PUSH_OUT has to be selected. Otherwise, "PushOut" is deactivated and thus ineffective. For details, refer to MB_PUSH_OUT_CONFIG on page 636
VAR_OUTPUT	Done	BOOL	Processing completed without errors, output data valid.
	Active	BOOL	Processing active, output data invalid.
	Error	BOOL	Processing completed with errors.
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification.
	ErrorIdent	ERROR_STRUCT	Detailed error information (see Troubleshooting, page 632)
	CamshaftDistance	REAL	If TRUE is assigned to the "GearReduction" input, this output specifies a value that is to be entered for several knives for the cam stroke in the drive.

Tab. 9-12: Function block interface description: MB_CrossCutterCalcType05

Min./max. and default values of the inputs

Name	Type	Min.-Wert	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
ModuloValue	LREAL	>0.0°	n.def.	360.0°	At a rising edge at "Execute"
FormatRatio	LREAL	0.02	30.0	1.0	At a rising edge at "Execute"
CutAngle	LREAL	0.0°	"ModuloValue"	12.0°	At a rising edge at "Execute"
PendulumFactor	LREAL	0.0%	100.0%	0.0%	At a rising edge at "Execute"
CorrectionProfile	MB_CORR_PROFILE			CORR_PROFILE_NONE	At a rising edge at "Execute"
GearReduction	BOOL	FALSE	TRUE	FALSE	At a rising edge at "Execute"
NumberOfKnives	UINT	1	20	1	At a rising edge at "Execute"
Overspeed	LREAL	-50.0%	100.0%	0.0%	At a rising edge at "Execute"
PushOut	MB_PUSH_OUT_CONFIG				At a rising edge at "Execute"

Tab. 9-13: Min./max. and default values of the function block: MB_CrossCutterCalcType05

Functional description

The function block calculates profiles for the "cross cutter" application based on the "cam" operation mode. After having enabled the processing with "Execute", a "cam" is calculated once based on the input variables. The cam

is available in the MB_CAMTABLE format and has 1024 data points. The cams are calculated in the "new format", i.e. the last point is at 0% or 100%.

([MB_CAMTABLE auf Seite 97](#)).

The cam can only be used with an axis with interpolation in the drive.

The following list describes the meaning of the individual inputs of the "MB_CrossCutterCalcType05" function block in detail:

- The "ModuloValue" input specifies the modulo value of the CrossCutter axis. 360.0° are specified by default.
- The "FormatRatio" input specifies the product length to be cut. It describes the ratio of the desired product length and the circumference of the CrossCutter cylinder. This means that if "FormatRatio=1", the length of the cut product corresponds to the circumference of the CrossCutter cylinder (synchronous format). If `FormatRatio=5`, the length of the product cut is five times the circumference of the cross cutter cylinder.
- The "CutAngle" input specifies the cutting area of the CrossCutter. In the cutting area, the CrossCutter is either traversed with the material feed velocity or with a (selectable) correction motion. The value relates to the cross cutter axis.
- The "PendulumFactor" input specifies the maximum pendular motion of the cross cutter axis in percent. This means that if "PendulumFactor=0%", the cross cutter axis never rotates with a negative velocity (backwards). If "PendulumFactor=100%", the cross cutter axis rotates to the border of the cutting area to minimize the energy loss and the required acceleration. This provides special benefits in case of higher values for "FormatRatio". If "PendulumFactor=100%", the cross cutter does not generally rotate up to the border of the cutting area since this is not always advantageous. The respective optimum utilization of the available area is automatically calculated by the function block



If the block is to use several knives, this input is NOT active.

For details, refer to [Several knives at one cross cutter on page 631](#).

- The "CorrectionProfile" input allows different correction motions in the cutting zone required for the respective use case ([MB_CORR_PROFILE on page 635](#)). The following can be selected:
 - `CorrectionProfile=CORR_PROFILE_NONE`
no correction: Cross cutter axis rotates with the master axis velocity
 - `CorrectionProfile=CORR_PROFILE_OVERSPEED`
constant overspeed of the cross cutter axis in the cutting area. This profile is only available for reasons of compatibility with the MB_CrossCutterCalcType04 block. As soon as a value is specified for the "Overspeed" input (value unequal 0), the `CorrectionProfile=CORR_PROFILE_OVERSPEED` profile will be automatically used even if the `CorrectionProfile=CORR_PROFILE_NONE` profile has been selected.
 - `CorrectionProfile=CORR_PROFILE_COS_COMP`
constant velocity vector in material direction, e.g. for guillotine cutters

- `CorrectionProfile=CORR_PROFILE_PUSH_OUT`
PushOut of the material
 - `CorrectionProfile=CORR_PROFILE_COS_COMP_AND_PUSH_OUT`
constant velocity vector in material direction as well as simultaneous PushOut, e.g. for guillotine cutters
 - The "GearReduction" input allows for operation with or without gear reduction. If the input is TRUE, the gear reduction will be applied in the drive. The cam will be calculated so that beginning and end are in each case at 0%.
 - The "NumberOfKnives" input will only be analyzed if the "GearReduction" input is also TRUE. In this case, a cam will be generated in the use of which the gear reduction has to be used in the drive. This allows for the use of a cross cutter with several knives. In this connection, the knives must be evenly distributed to the circumference of the cross cutter axis ([Several knives at one cross cutter on page 631](#)).
 - The "Overspeed" input allows the exact setting of the overspeed in [%] in the cutting area if
`CorrectionProfile=CORR_PROFILE_OVERSPEED` or `CorrectionProfile=CORR_PROFILE_NONE`.
Thus, entering a value of 10 [%] increases the velocity of the cross cutter axis in the cutting area by 10% compared to the material velocity.
 - The input "PushOut" parameterizes the PushOut correction (`MB_PUSH_OUT_CONFIG`). Traversing takes place at material velocity. It is accelerated in the cutting area to remove a product.
[MB_PUSH_OUT_CONFIG on page 636](#)
The input is only processed if
`CorrectionProfile=CORR_PROFILE_PUSH_OUT`
or
`CorrectionProfile=CORR_PROFILE_COS_COMP_AND_PUSH_OUT`.
- The successful calculation of a profile is signaled via the "Done" output. The parameters are written to the created "MB_CAMTABLE" data structure. Subsequently, the cam has to be written into the drive and activated by means of "MB_LoadCamData".

[MB_LoadCamData on page 94](#)

Several knives at one cross cutter

DANGER

If several knives are used, not all functions of the block are available.

The block CANNOT be used if one or several of the following conditions are satisfied:

- Correction profile ("CorrectionProfile" input): `CORR_PROFILE_COS_COMP` and `CORR_PROFILE_COS_COMP_AND_PUSH_OUT`
- Format ratio ("FormatRatio" input): with values 1 and less with `CorrectionProfile = CORR_PROFILE_PUSH_OUT`
- Modulo values ("ModuloValue" input): Values not corresponding to 360°

⚠ DANGER

If several knives are used, the verification of the "PendulumFactor" is no longer effective. That means that in a pendular motion, the knives may move into the cutting area.

It has to be determined in advance by means of tests whether with the given plant conditions, pendular motions reach the cutting area.

The block also offers the necessary functionality for the use of several, evenly (equidistantly) distributed knives. The number of knives is specified at the "NumberOfKnives" input. The "GearReduction" input must be preassigned with TRUE. The block will then calculate a cam the start and end value of which is in each case 0.

For the use of several knives, you have to switch on the gear reduction in the drive (set parameter P-0-0086, bit 4 in the drive). In addition, you must specify the number of knives in parameter P-0-0755, as well. This ensures that the knife positions are suitable for the master axis. The velocity in the intersection point does, however, not yet match. In order to achieve this, the block will also calculate a cam stroke ("CamshaftDistance"). Thus, the velocity in the cutting area is adjusted so that it corresponds to the master axis velocity. The cam stroke must be entered in the drive in parameter P-0-0093.

Troubleshooting

The function block uses the error table F_RELATED_TABLE, 16#170. It can generate the following error messages in Additional1 and Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0001	Invalid value of the "ModuloValue" input.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0002	Invalid value of the "FormatRatio" input.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0003	Invalid value of the "CutAngle" input.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0004	Invalid value of the "CutAngle" input combined with the cosine compensation.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0005	Invalid value of the "PendulumFactor" input.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0006	Invalid value of the "CorrectionProfile" input.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0007	Invalid value of the "Overspeed" input.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0008	Invalid starting point for the PushOut function.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#0009	Invalid value for "Overspeed_At_Cut" in the structure "MB_PUSH_OUT_CONFIG"
INPUT_RANGE_ERROR, 16#0006	16#0610	16#000A	Invalid value for "Overspeed_Max" in the structure "MB_PUSH_OUT_CONFIG"
INPUT_RANGE_ERROR, 16#0006	16#0610	16#000B	Invalid combination of "Overspeed_At_Cut" and "Overspeed_Max"
INPUT_RANGE_ERROR, 16#0006	16#0610	16#000C	Value of the "CutAngle" input too small combined with PushOut.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#000D	"NumberOfKnives" input has invalid values.
INPUT_RANGE_ERROR, 16#0006	16#0610	16#000E	In case of a value greater 1 for the "NumberOfKnives" input, the selected "CorrectionProfile", the "ModuloValue" or the combination of "CorrectionProfile" and "FormatRatio" is invalid.

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR, 16#0006	16#0610	16#000F	The "CamTable" input does not have any valid address. This can be caused by an online change.
STATE_MACHINE_ERROR, 16#0005	16#0006	xyxy	Invalid state of the state machine. xx specifies the invalid step. yy specifies the invalid state.

Tab. 9-14: Error numbers caused by MB_CrossCutterCalcType05

Special features for the MLC

To be able to use the function block with axes at an MLC, use the axes with **interpolation in drive**. Other axis types are not supported.

The "TECH" function package is required in the control.

The function block only generated a cam for use in the drive. More actions are required for further operation:

- Writing P-0-0086 (Configuration word synchronous operation modes): One of the bits 8-11 has to be set in this parameter in order for the cam to be interpreted in the "new format".
- Writing the cam to the drive using the "MB_LoadCamData" function block. Please note that the generated cam also has 1024 points and thus, only the first four cams can be used in the drive (the other cams only support up to 128 points).
- Writing A-0-2745 (ELS, gear reduction): Contains the reduction with which the cam is used in case of operation with several drives. The written value has to correspond to the number of used knives. Only required if several knives are used.
- Writing P-0-0086 (Configuration word synchronous operation modes): Bit 4 has to be set in this parameter so that the reduction path is enabled. Only required if several knives are used.
- The used axis has to be operated in the "cam" synchronous operation mode. The "CamShaftDistance" to be specified corresponds to the value output by the function block.

Special features for MLD

The "MA" function package is required in the MLD.

9.5 MB_CAM_TABLE_DATA

Brief description

The structure MB_CAM_TABLE_DATA is used by the function blocks [MB_CrossCutterCalcType04](#), [page 619](#), and [MB_CrossCutterCalcType05](#), [page 627](#),.

The structure contains an array with a cam table as well as meta data describing the number of array points as well as the position at which the table has to be loaded into the control/drive.

Structural element	Type	Description
CamTableID	DINT	Number of the cam table into which the described cam has to be loaded
NumberOfElements	UINT	Number of points of the described cam
CamTable	ARRAY[0..1023] OF DINT	Array with the cam data points

Tab. 9-15: Interface variables of the structure MB_CAM_TABLE_DATA

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value
CamTableID	DINT	0		0
NumberOfElements	UINT	0	1024	0

Tab. 9-16: Min./max. and default values of the MB_CAM_TABLE_DATA structure

Functional description The number of the cam table is saved in the structure element "CamTableID" in the format required by the function blocks ML_PLCOpen MB_ChangeCamData, MB_ChangeProfileStep and MB_ChangeProfileSet.

The table below shows the mapping between the "CamTableID" and the respective parameter number in the systems MLC and MLD.

CamTableID	Parameter number MLC	Parameter number MLD
1	C-0-2001	P-0-0072
2	C-0-2002	P-0-0092
3	C-0-2003	P-0-0780
4	C-0-2004	P-0-0781
5	C-0-2005	P-0-0784
6	C-0-2006	P-0-0785
7-99	C-0-2007 - C-0-2099	Not available

Tab. 9-17: Allocation of CamTableID to parameter numbers

The following diagram shows how to use the data of the MB_CAM_TABLE_DATA structure to load the cam either into the control or into the drive.

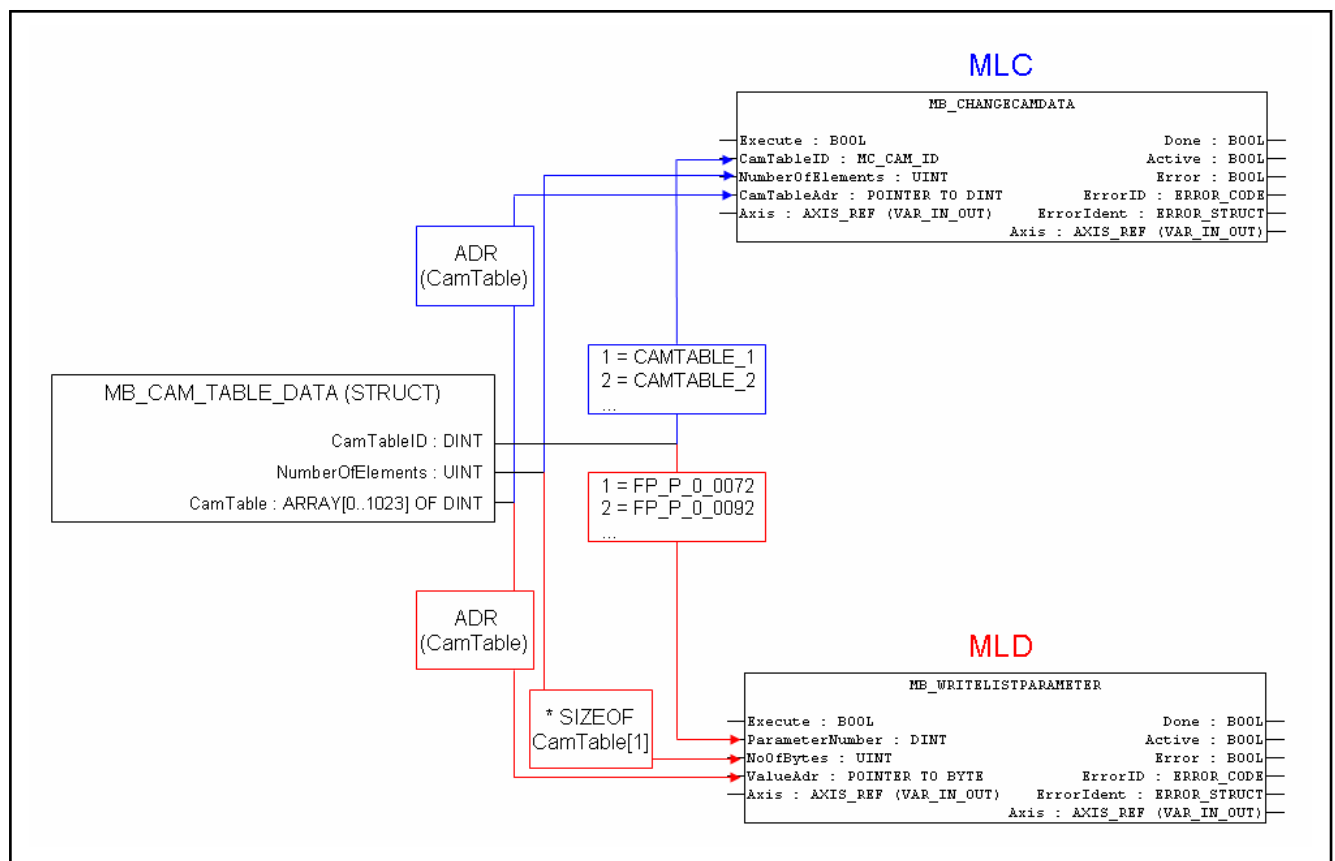


Fig. 9-14: Connection plan structure MB_CAM_TABLE_DATA

9.6 MB_CORR_PROFILE

Brief description The enumeration **MB_CORR_PROFILE** is used by the function block **MB_CrossCutterCalcType04**, page 619, and **MB_CrossCutterCalcType05**, page 627, to define a correction motion in the cutting area.

Enumeration type elements The following table lists the elements of the enumeration type.

Element	Value	Description
CORR_PROFILE_NONE	0	No correction in the cutting zone. The CrossCutter axis velocity corresponds to the master axis velocity
CORR_PROFILE_OVERSPEED	1	Constant overspeed or reduction of the CrossCutter axis velocity in the cutting area. The CrossCutter axis velocity is faster or slower than the master axis velocity by the value parameterized at the "Overspeed" input of the MB_CrossCutterCalcType04 function block
CORR_PROFILE_COS_COMP	2	Correction of the CrossCutter axis velocity The velocity is changed in a way that the CrossCutter axis rotates in material direction (the x component of the velocity vector) at a constant velocity. The speed of the CrossCutter axis has to be adapted depending on the current position in the cut

Element	Value	Description
CORR_PROFILE_PUSH_OUT	3	Executing a PushOut motion at the end of the cut. Parameterization via the "PushOut" input of the MB_CrossCutterCalcType04 function block. The speed of the CrossCutter axis has to be adapted depending on the current position in the cut. MB_PUSH_OUT_CONFIG, page 636
CORR_PROFILE_COS_COMP_AND_PUSH_OUT	4	Combination of the corrections "CORR_PROFILE_COS_COMP" and "CORR_PROFILE_PUSH_OUT". Parameterization via the "Push-Out" input of the MB_CrossCutterCalcType04 function block. MB_PUSH_OUT_CONFIG, page 636

Tab. 9-18: Elements of the enumeration type MB_CORR_PROFILE

9.7 MB_PUSH_OUT_CONFIG

Brief description The structure MB_PUSH_OUT_CONFIG is used by the function blocks [MB_CrossCutterCalcType04, page 619](#), and [MB_CrossCutterCalcType05, page 627](#),.

This structure can parameterize a defined PushOut at the end of the cut.

Interface description The following table lists the interface variables of the function block.

Structural element	Type	Description
PushOutBegin	REAL	Beginning of the PushOut area (parameter A). The value must not be equal to 0, unit in [°]
Overspeed_At_Cut	REAL	Overspeed during a cut (parameter B). The value must not be equal to 0, unit in [%]
Overspeed_Max	REAL	Maximum overspeed (parameter C), unit in [%]

Tab. 9-19: Interface variables of the structure MB_PUSH_OUT_CONFIG

Min./max. and default values of the inputs The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value
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. and default values of the MB_PUSH_OUT_CONFIG structure

Functional description The following graphics display the velocity characteristics of the CrossCutter roll in the cutting area. The individual parameters can be adjusted using the elements of the MB_PUSH_OUT_CONFIG structure.

A curve profile with a cut angle of 20° has been assumed for the graphics. The degree of the velocity change is determined with the help of the two variables "Overspeed_At_Cut" and "Overspeed_Max". The beginning of the ve-

locity increase is determined with the help of "PushOutBegin". For a unique parameterization, it is required that "PushOutBegin" and "Overspeed_At_Cut" are not equal to zero. Both figures show an example of the parameterization. The green line characterizes the degree of the increase and can be determined by connecting the points A, B and C.

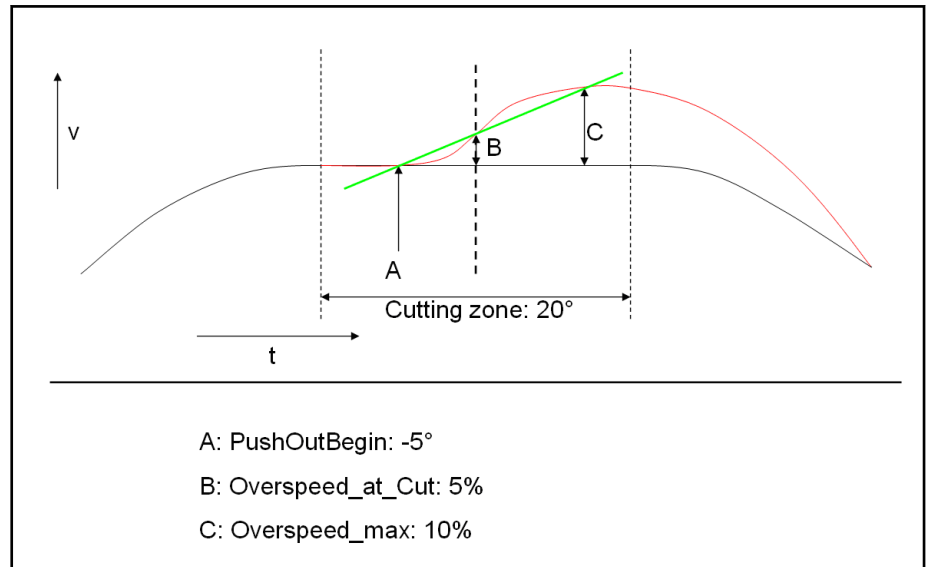


Fig. 9-15: Definition of PushOut.PushOutBegin < 0

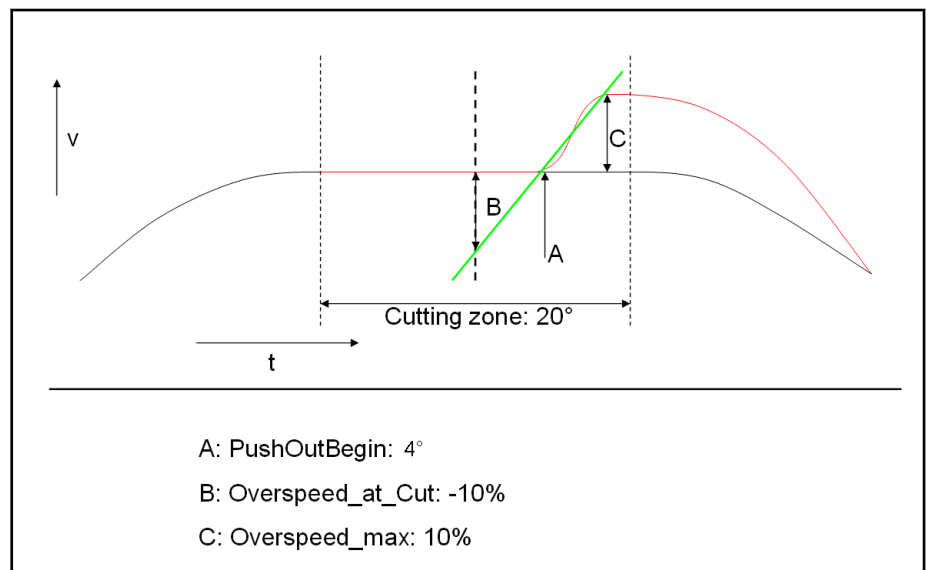


Fig. 9-16: Definition of PushOut.PushOutBegin > 0

9.8 MB_RESOLUTION

Brief description The enumeration MB_RESOLUTION is used by the [MB_CrossCutterCalc-Type04, page 619](#), function block.

With this enumeration, the number of cam table data points and thus the required calculation time can be modified.

Enumeration type elements The following table lists the elements of the enumeration type.

Element	Value	Description	Number of data points (cutting segment + compensating segment + cutting segment)
RESOLUTION_LOW	1	Low resolution, short calculation time required (faster by approx. factor 3 in contrast to medium resolution)	10 + 100 + 10
RESOLUTION_MIDDLE	2	Medium resolution, average calculation time required (faster by approx. factor 3 in contrast to high resolution)	25 + 300 + 25
RESOLUTION_HIGH	3	High resolution, much calculation time required	50 + 1000 + 50

Tab. 9-21: Elements of the enumeration type MB_RESOLUTION

Functional description

With the enumeration MB_RESOLUTION, the number of cam table data points and thus the required calculation time can be modified.

For short formats and little preparation time for a format change, the calculation time of the function block can be reduced selecting the "RESOLUTION_LOW" option. However, it is to be taken into account that the resolution of the generated cams is reduced and that the machine might not be running smoothly. For long formats and high preparation times for format changes, the "RESOLUTION_HIGH" option can be selected to generate the created cams with the highest resolution possible. Basically, this setting ("RESOLUTION_HIGH") is to be preferred.

The option to be selected depends on the application used and is to be specified by means of tests during the commissioning.

10 RMB_TechCam.library

10.1 Introduction and Overview

The library described includes function blocks to create cam tables.

These function blocks are available for different applications such as feed motion, cross cutting, slotters etc.

The function blocks MB_FeedMovementCalcType01, MB_FeedMovementCalcType02, and MB_CoordinatePairCamType01 operate with the enumeration [MB_CAMPROF](#), [page 97](#), and the structure [MB_CAMTABLE](#), [page 97](#), which can be found in the "ML_TechMotionBase" library.

The function blocks MB_SlotterCalcType01 and MB_SlotterCalcType02 operate with the enumeration [MB_CAMPROF_SLOTTER](#), [page 686](#), and the same "MB_CAMTABLE" structure.

The cam tables for the named function blocks can be loaded into the control or into a drive using function block [MB_LoadCamData](#), [page 94](#), from the "ML_TechMotionBase" library.



The library requires the **TEC (technology) function package** for the control.

RMB_TechCam function blocks

Function block	Description
MB_CoordinatePairCamType01 , page 639	Calculates a cam table with up to 1,024 data point elements that are determined depending on the selected coordinate pair type
MB_FeedMovementCalcType01 , page 647	Calculates a cam table with up to 1,024 data point elements for a feed motion
MB_FeedMovementCalcType02 , page 652	Calculates a cam table with up to 1,024 data points with a polynomial 7th order in which the inputs of the jerk, acceleration and velocity can be limited in order to ensure optimum processing
MB_SlotterCalcType01 , page 664	Calculates a cam table with 1,024 data point elements for a slotter with one knife on the slotter roller
MB_SlotterCalcType02 , page 676	Calculates a cam table with 1,024 data point elements for a slotter with two knives on the slotter roller

Tab. 10-1: RMB_TechCam

10.2 MB_CoordinatePairCamType01

Brief description The function block calculates a cam table with up to 1,024 data point elements.

The cam table is determined with regard to the selected coordinate pair type.

Assignment: Target system/library

MLC 14VRS	RMB_TechCam
Function package TEC	

Tab. 10-2: Reference table of the MB_CoordinatePairCamType01

Interface description

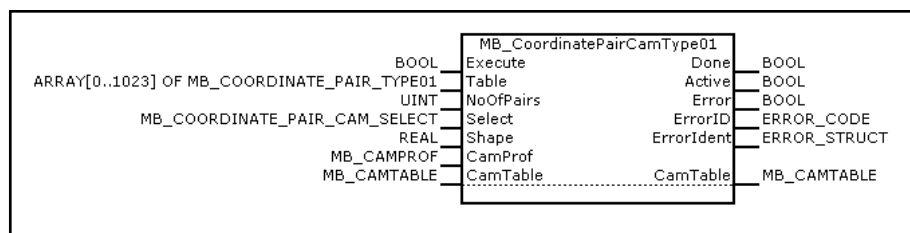


Fig. 10-1: MB_CoordinatePairCamType01 function block

I/O type	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Function block processing enabled (once, edge-controlled)
	Table	ARRAY[0..1023] OF MB_COORDINATE_PAIR_TYPE01	List of coordinate pairs MB_COORDINATE_PAIR_TYPE01, page 646
	NoOfPairs	UINT	Number of valid coordinate pairs
	Select	MB_COORDINATE_PAIR_CAM_SELECT	Selection of the coordinate pair types MB_COORDINATE_PAIR_CAM_SELECT, page 646
	Shape	REAL	Factor for jerk limitation
	CamProf	MB_CAMPROF	Cam profile selection
VAR_OUTPUT	Done	BOOL	Processing completed without errors, output data valid
	Active	BOOL	Processing active, output data invalid
	Error	BOOL	Processing completed with error, output data invalid, error outputs valid
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, it contains a detailed error description (see Troubleshooting, page 646)
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements

Tab. 10-3: MB_CoordinatePairCamType01 function block

Min./max. and default values

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
NoOfPairs	UINT	1 or 2	1023	0	Rising edge at "Execute"
Select	MB_COORDINATE_PAIR_CAM_SELECT			COORDINATE_PAIR_POS_VEL_TYPE01	Rising edge at "Execute"
Shape	REAL	0.0	1.0	0.0	Rising edge at "Execute"
CamProf	MB_CAMPROF			CAM_PROF_STANDARD	Rising edge at "Execute"

Tab. 10-4: Min./max. and default values of the MB_CoordinatePairCamType01 function block

The enumeration [MB_CAMPROF](#), [page 97](#), selects between different cam profiles.



When selecting the cam profile "CAMPROF_ADDITIVE_SEGMENTABLE", the following is to be considered:

- For cam tables for motions with unlimited (infinite) travel range whose first and last element are identical, the linear path has to be ticked and the reduction has to be set to "1", since it is a profile with gear reduction. For all other cam tables ($|\text{last\_Element} - \text{first\_Element}| = 100\%$), the linear path has to be disabled, since they are profiles without gear reduction.
- For cam tables for motions with limited (finite) travel range, the linear path has to be disabled.

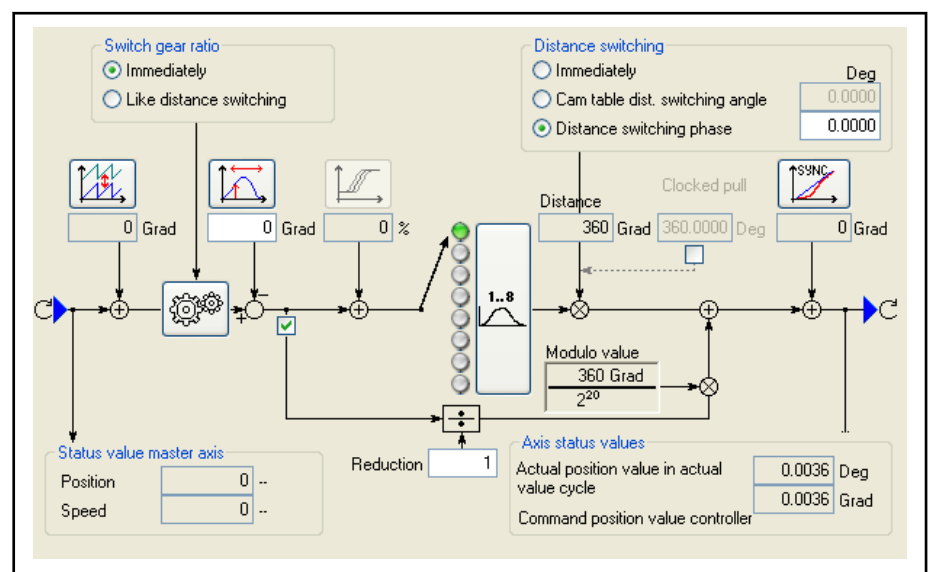


Fig. 10-2: Enabling the linear path

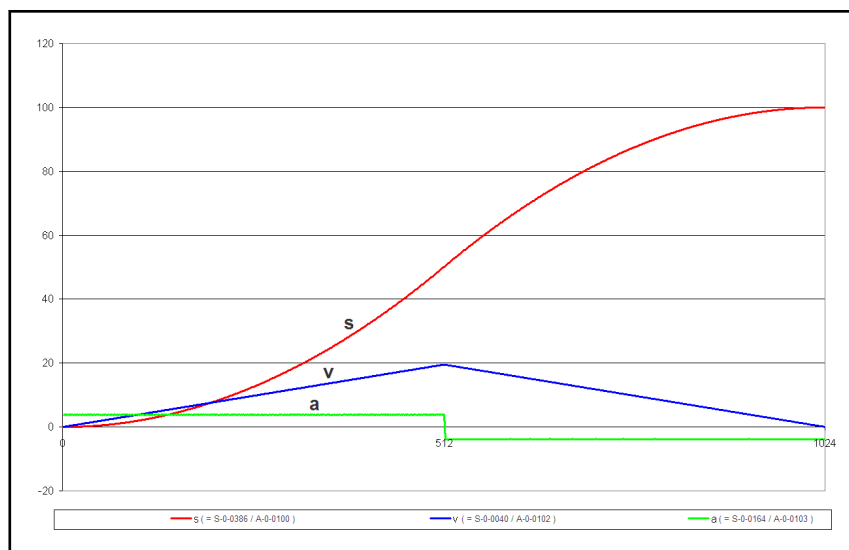
The structure [MB_CAMTABLE](#), [page 97](#), is provided with the "CamTable" array containing the calculated cam points and the "NumberOfElements" variable which shows the number of valid cam points of the "CamTable" array.

Functional Description

With the specification of the coordinate pairs, the ML_CoordinateCamPair-Type01 function block carries out a one-time calculation of a cam table with up to 1,024 data point elements after processing enabled via "Execute" based on the input values.

The number of data point elements of a cam table has to be set by the "NumberOfElements" element of the "CamTable" structure.

- Select: COORDINATE_PAIR_POS_VEL_TYPE01
 - If the data point (0;0) is no specified coordinate pair, it is automatically considered when calculating the cam table.



Coordinate pairs: (180;100)

Fig. 10-3: MB_CoordinatePairCamType01, MotionProfile COORDINATE_PAIR_POS_VEL_TYPE01

- Select: COORDINATE_PAIR_POS_VEL_TYPE02
 - To calculate the cam table, at least two coordinate pairs have to be given.
 - Due to the "Shape" factor provided for the jerk limitation, the velocity is blending.
 - The cam table calculated can only be used as an endless cam if the first data point is (0;0) and the last data point is (360;0). If this is not the case, there are position irregularities at the cam transitions.
 - If the above mentioned data points (0; 0 and 360; 0) are inserted and the "Shape" factor is set to 0%, the created cam is identical to the cam created due to the selection of COORDINATE_PAIR_POS_VEL_TYPE01.

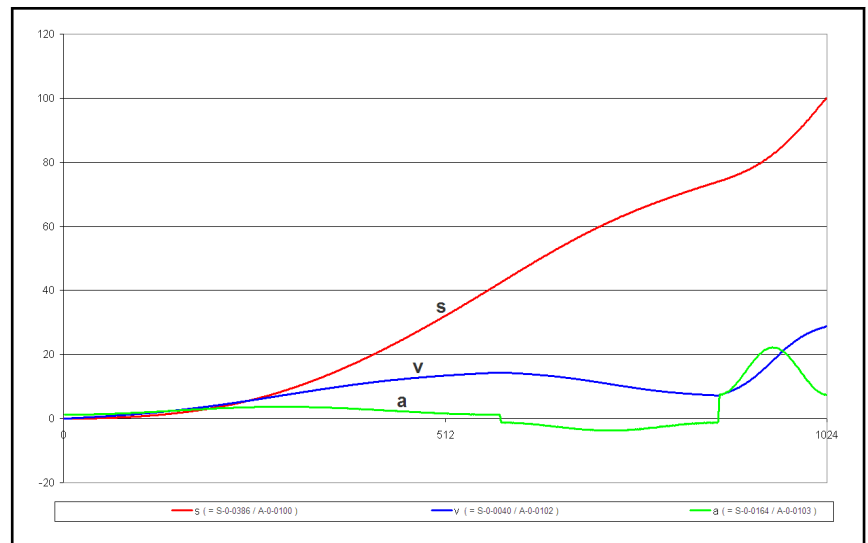


Fig. 10-4: MB_CoordinatePairCamType01, MotionProfile COORDINATE_PAIR_POS_VEL_TYPE02

- Select: COORDINATE_PAIR_POS_ACC
 - This selection has to be used when the desired cam profile is specified by individual data points to be traveled with a "rest-in-rest" motion. The target points are connected with a polynomial 5th order. The consequence is that every target point from the input data is approached with an acceleration/deceleration motion, i.e. the velocity is = 0 at every target point.
 - To calculate the cam table, at least two coordinate pairs have to be given.
 - Due to the "Shape" factor provided for the jerk limitation, the velocity is blending.

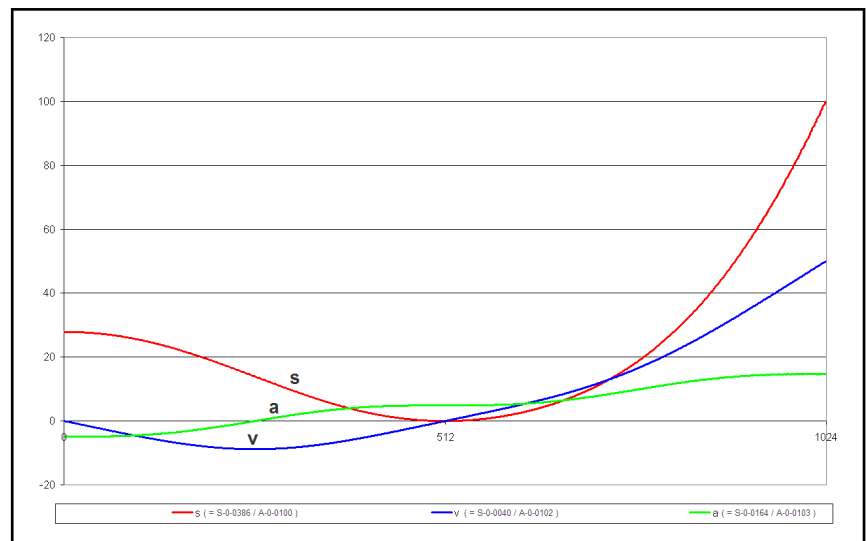
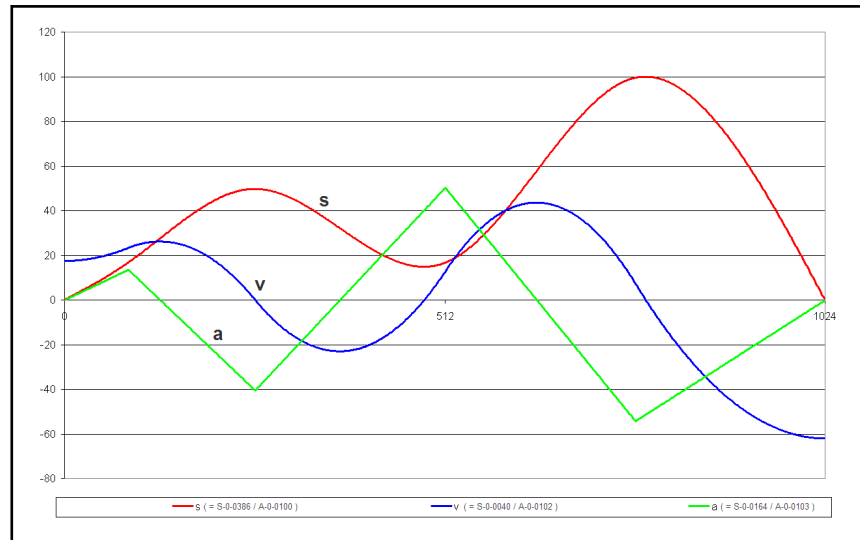


Fig. 10-5: MB_CoordinatePairCamType01, MotionProfile COORDINATE_PAIR_POS_ACC

- Select: COORDINATE_PAIR_POS_POS_TYPE01

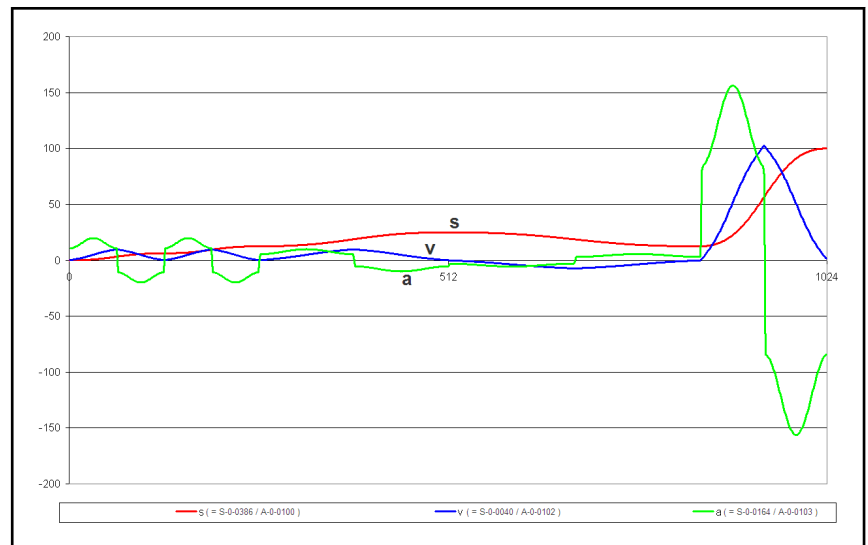
- To calculate the cam table, at least two coordinate pairs have to be given.
- It is traveled through the individual value pairs without rest. The target positions are connected with a spline polynomial 3rd order.



Coordinate pairs: (0;0), (30;10), (90;30), (180;10), (270;60), (360;0)

Fig. 10-6: MB_CoordinatePairCamType01, MotionProfile COORDINATE_PAIR_POS_POS_TYPE01

- Select: COORDINATE_PAIR_POS_POS_TYPE02
 - This selection has to be used when the desired cam profile is specified by individual data points to be traveled with a "rest-in-rest" motion. The target points are connected with a polynomial 5th order. The consequence is that every target point from the input data is approached with an acceleration/deceleration motion, i.e. the velocity is = 0 at every target point.
 - To calculate the cam table, at least two coordinate pairs have to be given.
 - Due to the "Shape" factor provided for the jerk limitation, the velocity is blending.



Coordinate pairs: (0;0), (45;50), (90;100), (180;200), (300;100), (360;800); Shape = 0.3

Fig. 10-7: MB_CoordinatePairCamType01, MotionProfile COORDINATE_PAIR_POS_POS_TYPE02

Downloading the cam to the control or to the drive

The cam table to be calculated is stored in the "CamTable" structure with scaling in per cent. The validity of the cam elements is signaled by the "Done" output. The cam table can be loaded to the control system or drive using the [MB_LoadCamData](#), page 94, function block.

Cam characteristics

The cam table of the ML_CoordinatePairCamType01 provides the following characteristics:

- The algorithm to calculate the cam data points is distributed across several cycles, i.e. calls of the instance to avoid a considerable effect on the PLC cycle time.
- Select: COORDINATE_PAIR_POS_VEL_TYPE01
 - The MotionProfile characterized by the cam table is constant up to the velocity (s, v) and shows a jump in the acceleration (a)
- Select: COORDINATE_PAIR_POS_VEL_TYPE02
 - The MotionProfile characterized by the cam table is constant up to the acceleration (s, v, a) depending on the "Shape" factor and shows a jump in the jerk (j)
- Select: COORDINATE_PAIR_POS_ACC
 - The MotionProfile characterized by the cam table is constant up to the acceleration (s, v, a) depending on the "Shape" factor and shows a jump in the jerk (j)
- Select: COORDINATE_PAIR_POS_POS_TYPE01
 - The MotionProfile characterized by the cam table is constant up to the acceleration (s, v, a) and shows a jump in the jerk (j)
- Select: COORDINATE_PAIR_POS_POS_TYPE02
 - The MotionProfile characterized by the cam table is constant up to the acceleration (s, v, a) depending on the "Shape" factor and shows a jump in the jerk (j)

If an error occurs while processing the function block, it is signaled by the "Error" output. The elements of the "CamTable" structure are not updated in case of error.

Troubleshooting The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR	16#0006	16#0000	Invalid function block state
CALCULATION_ERROR	16#000C	16#0000	Division by zero
INPUT_RANGE_ERROR	16#2200	16#0001	"NumberOfElements" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#2270	16#0001	"NoOfPairs" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#2270	16#0002	"Shape" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#2270	16#0003	Input element of "Table" is < minimum or > maximum
INPUT_INVALID_ERROR	16#2201	16#0001	Address of the "CamTable" array is zero
INPUT_INVALID_ERROR	16#2271	16#0001	Input element of "Table" is < (lower than) the previous input element

Tab. 10-5: Error codes of the MB_CoordinatePairCamType01 function block

10.3 MB_COORDINATE_PAIR_CAM_SELECT

The enumeration "MB_COORDINATE_PAIR_CAM_SELECT" selects between the different types (position, velocity, and acceleration) of the coordinate pairs.

Element	Value	Description
COORDINATE_PAIR_POS_VEL_TYPE01	0	Coordinate pairs from position (X) and velocity (Y)
COORDINATE_PAIR_POS_VEL_TYPE02	1	Coordinate pairs from position (X) and velocity (Y) considering the "Shape" input
COORDINATE_PAIR_POS_ACC	2	Coordinate pairs from position (X) and acceleration (Y) considering the "Shape" input
COORDINATE_PAIR_POS_POS_TYPE01	3	Coordinate pairs from position (X) and position (Y)
COORDINATE_PAIR_POS_POS_TYPE02	4	Coordinate pairs from position (X) and position (Y) considering the "Shape" input

Tab. 10-6: Enumeration type MB_COORDINATE_PAIR_CAM_SELECT

10.4 MB_COORDINATE_PAIR_TYPE01

The MB_COORDINATE_PAIR_TYPE01 structure contains the coordinate pair (X, Y) of one point.

Name	Type	Min. value	Max. value	Default value	Effective	Description
X	LREAL	0.0	=<360.0	0.0	Rising edge at "Execute"	X-coordinate of the coordinate pair (angle)
Y	LREAL	n.def.	n.def.	0.0	Rising edge at "Execute"	Y-coordinate of the coordinate pair (position, velocity or acceleration)

Tab. 10-7: MB_COORDINATE_PAIR_TYPE01 structure with min., default, - and max. value

10.5 MB_FeedMovementCalcType01

Brief description The function block calculates a cam table with up to 1,024 data point elements for a feed motion.

Assignment: Target system/library

MLC 14VRS	RMB_TechCam
TEC function package	

Tab. 10-8: Reference table of the MB_FeedMovementCalcType01

Interface description

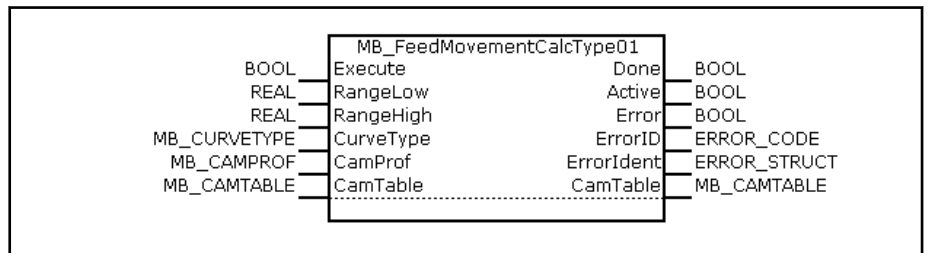


Fig. 10-8: MB_FeedMovementCalcType01 function block

I/O type	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Processing of the function block enabled (once, edge-controlled)
	RangeLow	REAL	Starting point of the feed motion. First point of the MotionProfile
	RangeHigh	REAL	End point of the feed motion. Last point of the Motion-Profile
	CurveType	MB_CURVETYPE	Curve type selection MB_CURVETYPE, page 658
	CamProf	MB_CAMPROF	Cam profile selection
VAR_OUTPUT	Done	BOOL	Processing completed without errors, output data is valid
	Active	BOOL	Processing not yet completed, output data invalid
	Error	BOOL	Processing completed with error, output data invalid, error outputs valid
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, it contains a detailed error description (see Error handling, page 652)
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements

Tab. 10-9: Interface variables of the MB_FeedMovementCalcType01 function block

Min./max. and default values The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
RangeLow	REAL	0.0	<360.0	0.0	Rising edge at "Execute"

Name	Type	Min. value	Max. value	Default value	Effective
RangeHigh	REAL	>0.0	360.0	360.0	Rising edge at "Execute"
CurveType	MB_CURVETYPE			CURVE-TYPE_POLY_5	Rising edge at "Execute"
CamProf	MB_CAMPROF			CAM_PROF_STANDARD	Rising edge at "Execute"

Tab. 10-10: Min./max. and default values of the MB_FeedMovementCalcType01 function block

The enumeration MB_CAMPROF, page 97, selects between different cam profiles.



When selecting the cam profile "CAMPROF_STANDARD", the linear path has to be disabled since it is a profile without gear reduction. The linear path for the profiles "CAMPROF_ADDITIVE" and "CAMPROF_ADDITIVE_SEGMENTABLE" has to be ticked and the reduction has to be set to "1", since they are profiles with gear reduction.

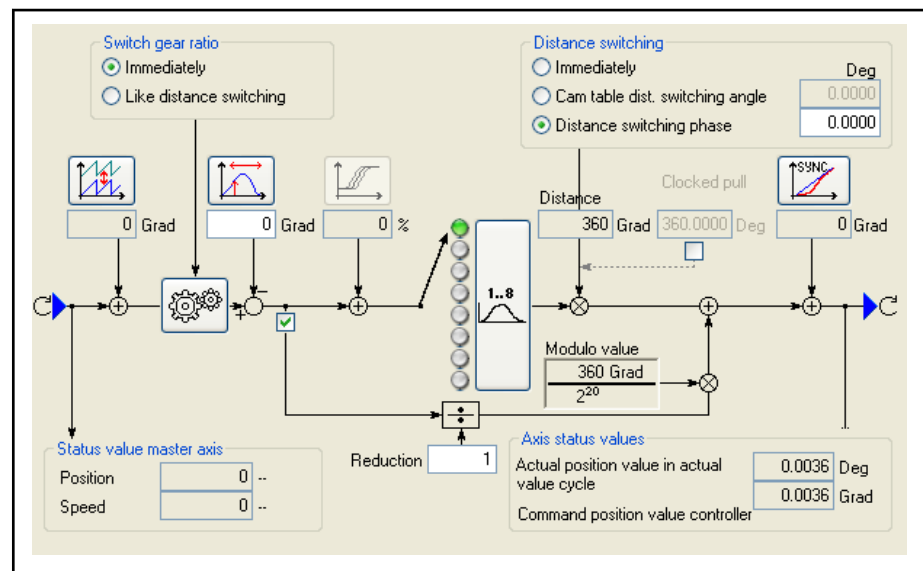


Fig. 10-9: Enabling the linear path

The structure MB_CAMTABLE, page 97, is provided with the "CamTable" array containing the calculated cam points, and the "NumberOfElements" variable, which shows the number of valid cam points of the "CamTable" array.

Functional description

With the specification of a feed range defined by the limits "RangeLow" and "RangeHigh", the ML_FeedMovementCalcType01 function block calculates a cam table with 1,024 data point elements one time after enabling the processing with "Execute" based on the input values. The number of data point elements of a cam table is set by the "NumberOfElements" element of the "CamTable" structure.

The motion from rest-to-rest can be carried out with different compensation motions. The adjustable motions are:

- Square parabola
- Simple sine line

- Polynomial 5th order
- Inclined sine line
- Modified acceleration trapezoid
- Modified sine line

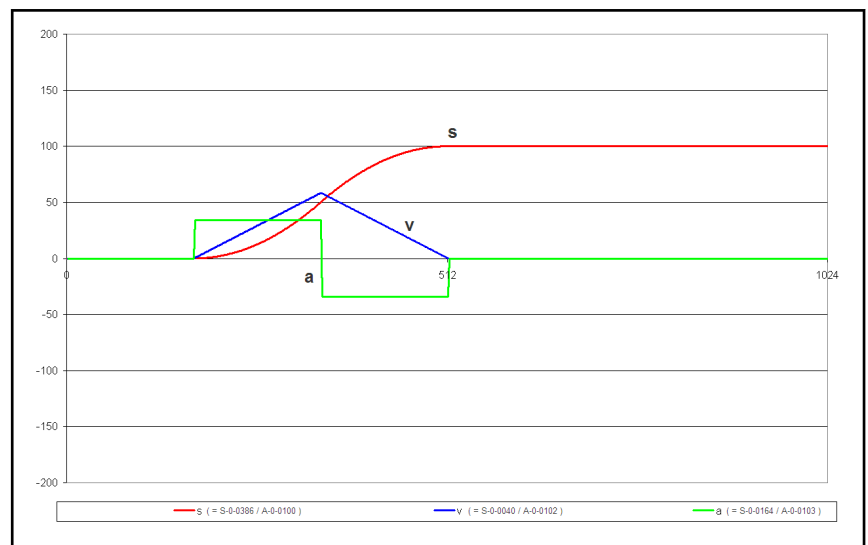
The cam table to be calculated is stored in the "CamTable" structure with scaling in per cent. The validity of the cam elements is signaled by the "Done" output. The cam table can be loaded into a cam in the control or into the drive using the [MB_LoadCamData](#), [page 94](#), function block.

If an error occurs while processing the function block, this is signaled by the "Error" output. The elements of the "CamTable" structure are not updated in case of error.

Cam characteristics

The cam table of the MB_FeedMovementCalcType01 provides the following characteristics:

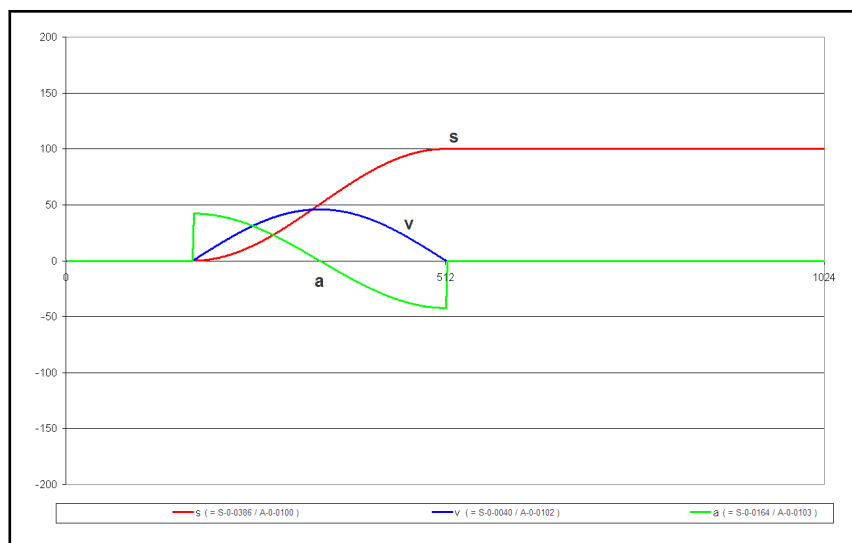
- The cam table generated by the function block describes a rest-in-rest motion
- The algorithm to calculate the cam data points depends on a difference (RangeHigh-RangeLow) of 1,026 cycles max, i.e. calls of the instance to avoid a considerable effect on the PLC cycle time
- The cam profile is provided for the cyclic processing in the drive as parameters P-0-0072, P-0-0092 etc.
- CurveType: CURVETYPE_QUADRATIC_PARABLE
 - The MotionProfile characterized by the cam is constant up to the acceleration (s, v, a) and shows a jump in the jerk (j)



RangeLow = 60; RangeHigh = 180

Fig. 10-10: MB_FeedMovementCalcType01, MotionProfile QUADRATIC_PARABLE

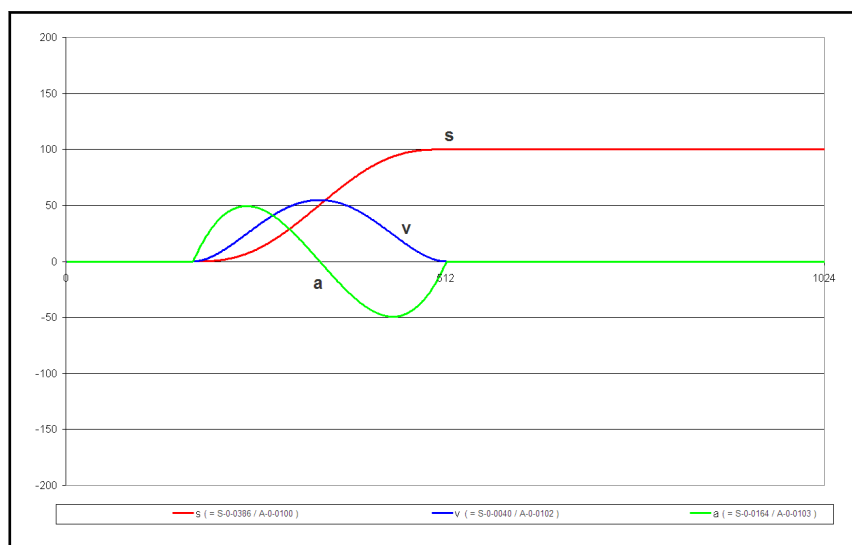
- CurveType: CURVETYPE_SIMPLE_SINUSLINE
 - The MotionProfile characterized by the cam is constant up to the acceleration (s, v, a) and shows a jump in the jerk (j)



RangeLow = 60; RangeHigh = 180

Fig. 10-11: MB_FeedMovementCalcType01, MotionProfile SIMPLE_SINUSLINE

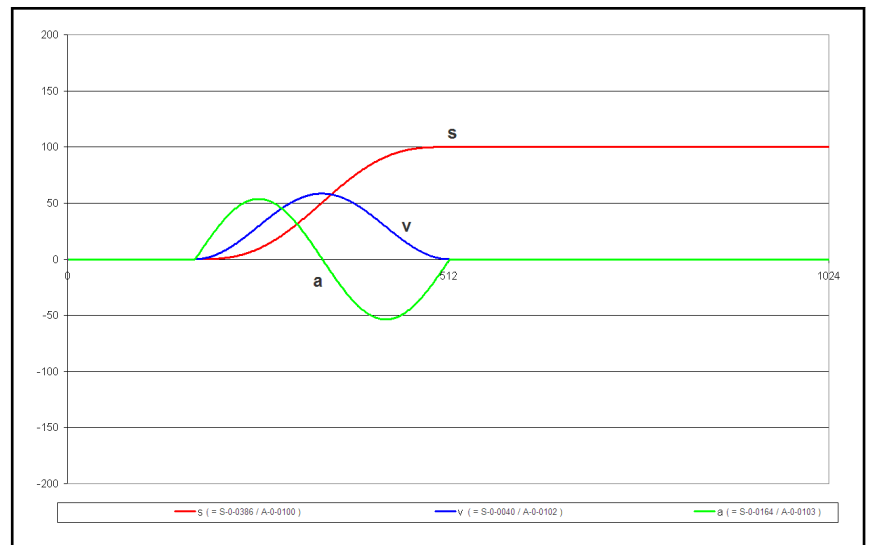
- CurveType: CURVETYPE_POLY_5
 - The MotionProfile characterized by the cam is constant up to the jerk (s, v, a, j)



RangeLow = 60; RangeHigh = 180

Fig. 10-12: MB_FeedMovementCalcType01, MotionProfile POLY_5

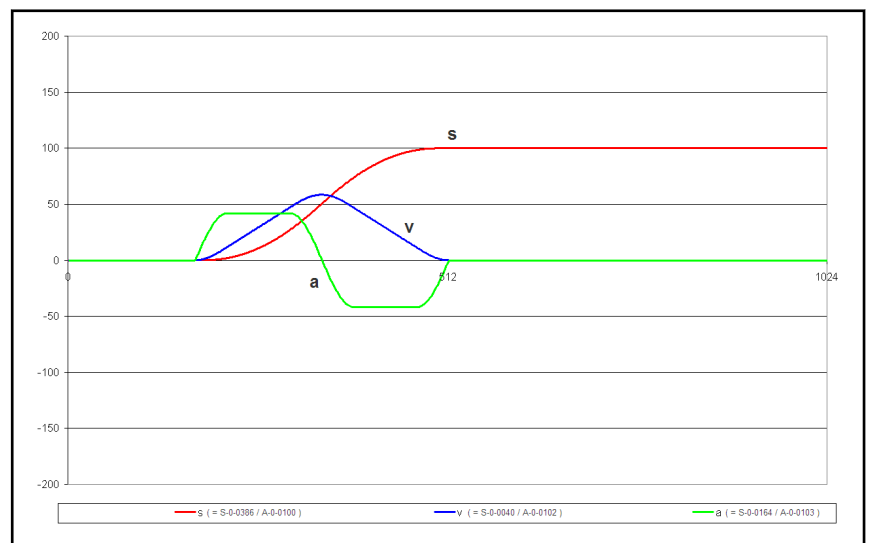
- CurveType: CURVETYPE_SLOPING_SINUSLINE
 - The MotionProfile characterized by the cam is constant up to the jerk (s, v, a, j)



RangeLow = 60; RangeHigh = 180

Fig. 10-13: MB_FeedMovementCalcType01, MotionProfile SLOPING_SINUSLINE

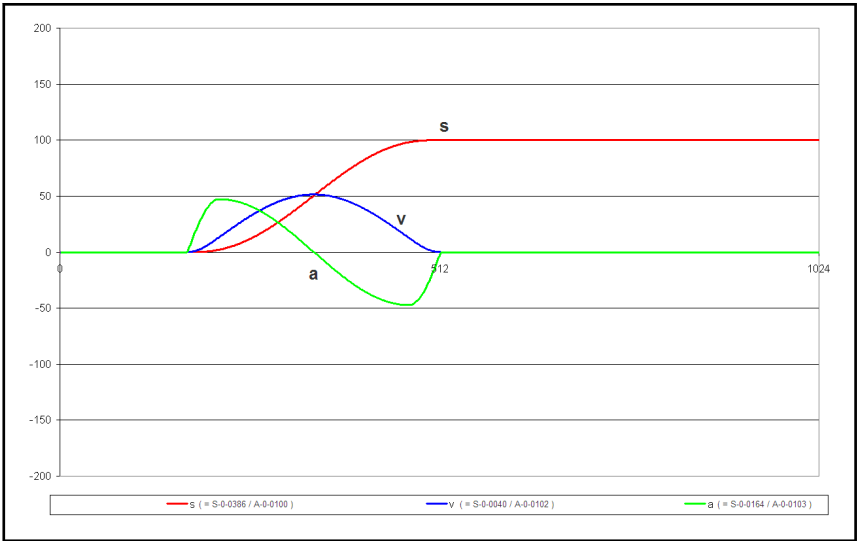
- CurveType: CURVETYPE_MODIFIED_ACCELERATIONTRAPEZE
 - The MotionProfile characterized by the cam is constant up to the jerk (s, v, a, j)



RangeLow = 60; RangeHigh = 180

Fig. 10-14: MB_FeedMovementCalcType01, MotionProfile MODIFIED_ACCELERATIONTRAPEZE

- CurveType: CURVETYPE_MODIFIED_SINUSLINE
 - The MotionProfile characterized by the cam is constant up to the jerk (s, v, a, j)



RangeLow = 60; RangeHigh = 180

Fig. 10-15: MB_FeedMovementCalcType01, MotionProfile MODIFIED_SINUSLINE

Error handling The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2.

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR	16#0006	16#0000	Invalid function block state
CALCULATION_ERROR	16#000C	16#0000	Division by Zero
INPUT_RANGE_ERROR	16#2200	16#0001	"NumberOfElements" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#2220	16#0001	"RangeLow" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#2220	16#0002	"RangeHigh" input is < minimum or > maximum
INPUT_INVALID_ERROR	16#2201	16#0001	Address of the "CamTable" array is zero
INPUT_INVALID_ERROR	16#2221	16#0001	"RangeLow" input is ≥ RangeHigh

Tab. 10-11: Error codes of the MB_FeedMovementCalcType01 function block

10.6 MB_FeedMovementCalcType02

Brief description The function block calculates a cam table with up to 1,024 data points with a polynomial 7th order in which the inputs of the jerk, acceleration, and velocity can be limited in order to ensure optimum processing.

Assignment: Target system/library

MLC 14VRS	RMB_TechCam
TEC function package	

Tab. 10-12: Reference table of the MB_FeedMovementCalcType02

Interface description

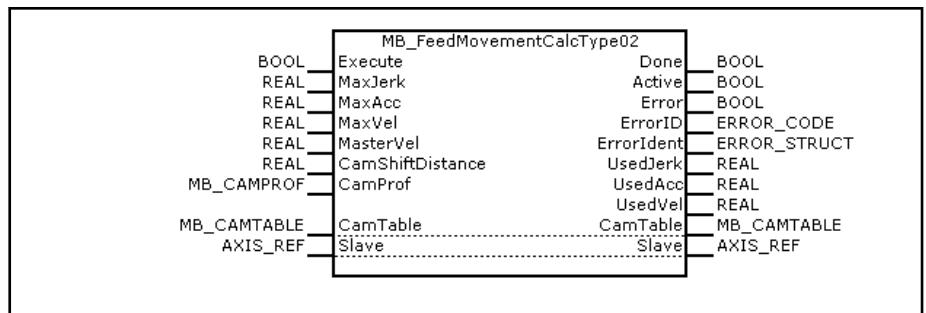


Fig. 10-16: MB_FeedMovementCalcType02 function block

I/O type	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Processing of the function block enabled (once, edge-controlled)
	MaxJerk	REAL	Maximum jerk
	MaxAcc	REAL	Maximum acceleration
	MaxVel	REAL	Maximum velocity
	MasterVel	REAL	Velocity of the master axis
	CamShiftDistance	REAL	Cam length
	CamProf	MB_CAMPROF	Cam profile selection
VAR_OUTPUT	Done	BOOL	Processing completed without errors, output data is valid
	Active	BOOL	Processing not yet completed, output data invalid
	Error	BOOL	Processing completed with error, output data invalid, error outputs valid
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, it contains a detailed error description (see Error handling , page 658)
	UsedJerk	REAL	Jerk used related to the input "MaxJerk" in [%]
	UsedAcc	REAL	Acceleration used related to the input "MaxAcc" in [%]
	UsedVel	REAL	Used velocity in relation to the input "MaxVel" in [%]
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements
VAR_IN_OUT CONSTANT	Slave	AXIS_REF	Reference to the slave axis

Tab. 10-13: Interface variables of the MB_FeedMovementCalcType02 function block

Min./max. and default values of the inputs

The following table lists the min./max. and default values of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
MaxJerk	REAL	>0.0	n.def.	20.0	Rising edge at "Execute"

Name	Type	Min. value	Max. value	Default value	Effective
MaxAcc	REAL	>0.0	n.def.	20.0	Rising edge at "Execute"
MaxVel	REAL	>0.0	n.def.	20.0	Rising edge at "Execute"
MasterVel	REAL	>0.0	n.def.	10.0	Rising edge at "Execute"
CamShaftDistance	REAL	>0.0	n.def.	360.0	Rising edge at "Execute"
CamProf	MB_CAMPROF			CAM_PROF_STANDARD	Rising edge at "Execute"

Tab. 10-14: Min./max. and default values of the MB_FeedMovementCalcType02 function block

The enumeration [MB_CAMPROF](#), [page 97](#), selects between different cam profiles.



When selecting the cam profile "CAMPROF_STANDARD", the linear path has to be disabled since it is a profile without gear reduction. The linear path for the profiles "CAMPROF_ADDITIVE" and "CAMPROF_ADDITIVE_SEGMENTABLE" has to be ticked and the reduction has to be set to "1", since they are profiles with gear reduction.

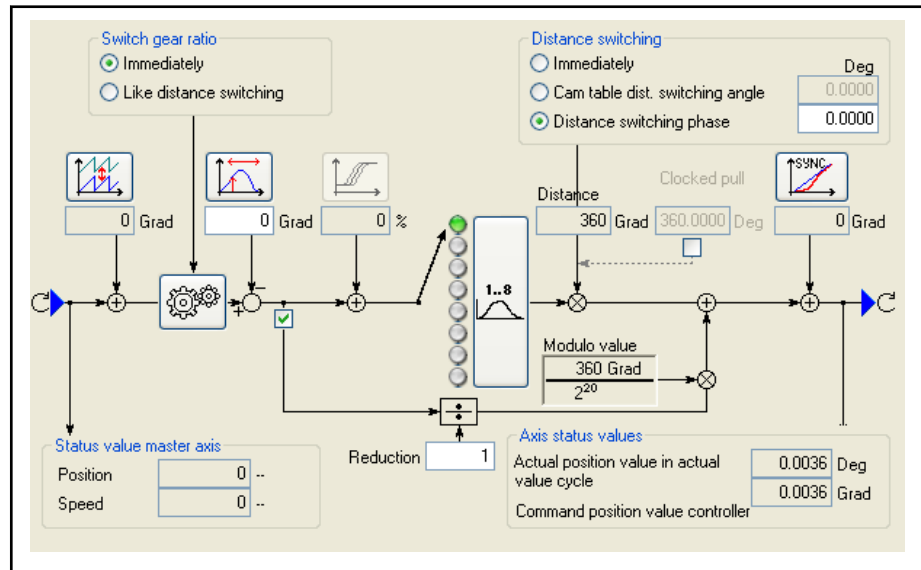


Fig. 10-17: Enabling the linear path

The structure [MB_CAMTABLE](#), [page 97](#), is provided with the "CamTable" array containing the calculated cam points, and the "NumberOfElements" variable, which shows the number of valid cam points of the "CamTable" array.

Functional description

The MB_FeedMovementCalcType02 function block calculates a cam table with up to 1,024 data points. At this cam table, jerk, acceleration and velocity can be limited. The number of data point elements of a cam table is set by the "NumberOfElements" element of the "CamTable" structure.

The user sets the master axis velocity, cam distance, maximum velocity, acceleration and jerk. The function block uses these specified values to calcu-

late a MotionProfile as cam indicating a rest-in-rest motion. The jerk profile either consists of individual parabola parts or calculated as polynomial 7th order. The MotionProfile is calculated from this jerk profile. The MotionProfile allows an endless motion of the slave axis.

Remark:

The variant of the function block for drive cams gets more information from the parameters of the respective drive required for the calculation of the function block. That information includes for example the maximum jerk of the drive or the modulo value which means that parameters are read from the drive while the function block is active. If the specified values are possible, a cam is calculated or an error is output stating that the motion with this specified values is not possible.

The modulo value of the drive has to correspond to the value of the input variable "CamShaftDistance". Otherwise, the function block is completed with error. The parameter "CamShaftDistance" of the command "MC_CamIn" has thus to be set as high as the modulo value of the slave axis. Otherwise, an endless motion is not possible.

The cam table to be calculated is stored in the "CamTable" structure with scaling in per cent. The validity of the cam elements is signaled by the "Done" output. The cam table can be loaded into a cam in the control or into the drive using the [MB_LoadCamData, page 94](#), function block.

If an error occurs while processing the function block, this is signaled by the "Error" output. The elements of the "CamTable" structure as well as the outputs "UsedJerk" and "UsedAcc" and "UsedVel" are not updated in case of error.

Cam characteristics

The cam table of the MB_FeedMovementCalcType02 provides the following characteristics:

- The MotionProfile characterized by the cam table is constant up to the jerk (s, v, a, j)
- The algorithm to calculate the cam data points is distributed across several cycles, i.e. calls of the instance to avoid a considerable effect on the PLC cycle time

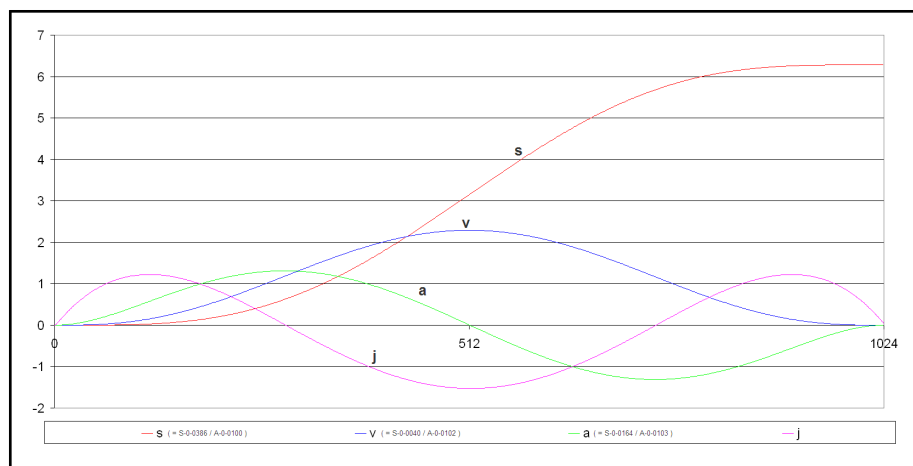
Overview on possible calculation procedures

Switching between the individual variants is executed by the function block. Preferably, a cam with polynomial 7th order is calculated. If this is not possible due to the set limit values, it is calculated using parabola parts instead. If this option does also not allow to maintain the limit values, the "Error" output of the function block is set.

Remark:

When calculating using parabola parts, inaccuracies might occur if the ratio of jerk and acceleration is most unfavorable (high jerk at low acceleration) since the parabola becomes very high for the jerk and very narrow at the same time. Due to the limited accuracy, inaccuracy can result.

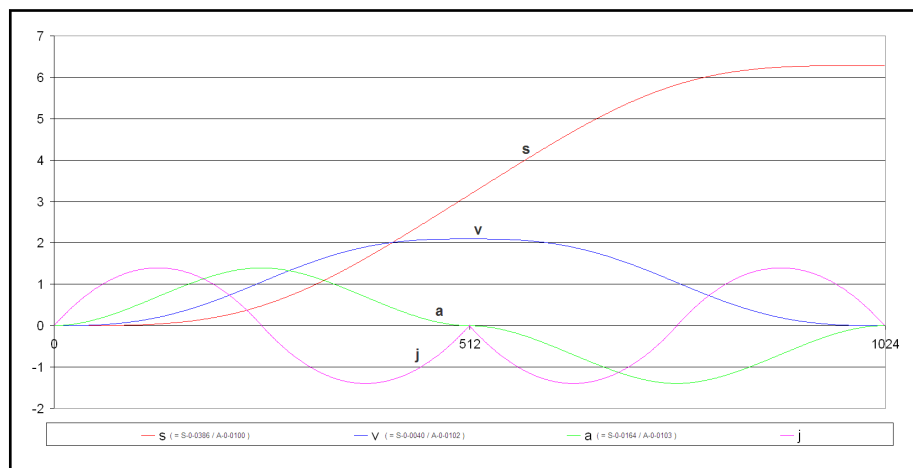
Variant 1: Polynomial 7th order



MaxJerk = 10 rad/s³; MaxAcc = 10 rad/s²; MaxVel = 10 rad/s; MasterVel= 10 rpm; modulo value = 2*Pi rad
Calculation as polynomial 7th order

Fig. 10-18: Cam profile of the MB_FeedMovementCalcType02 [slave = f(master)]

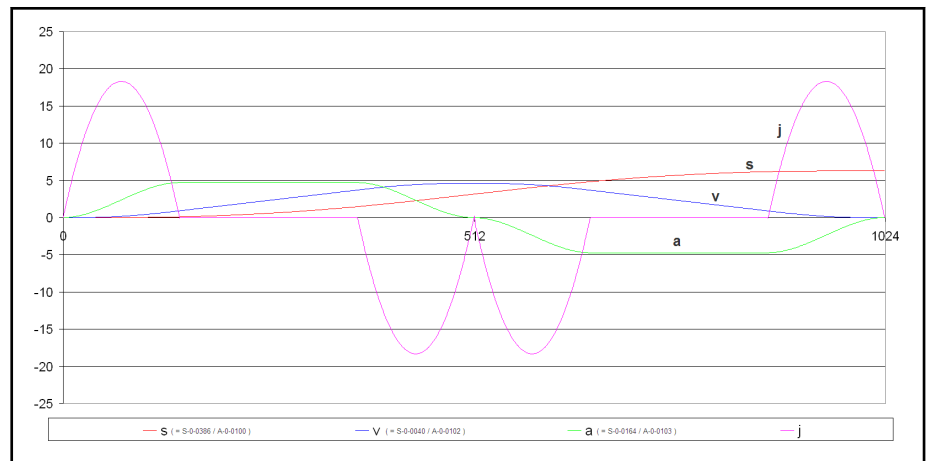
Variant 2: Velocity and acceleration limit values are not reached (variant is rarely used in practice)



MaxJerk = 10 rad/s³; MaxAcc = 10 rad/s²; MaxVel = 10 rad/s; MasterVel= 10 rpm; modulo value = 2*Pi rad
Calculation from parabola parts

Fig. 10-19: Cam profile of the MB_FeedMovementCalcType02 [slave = f(master)]

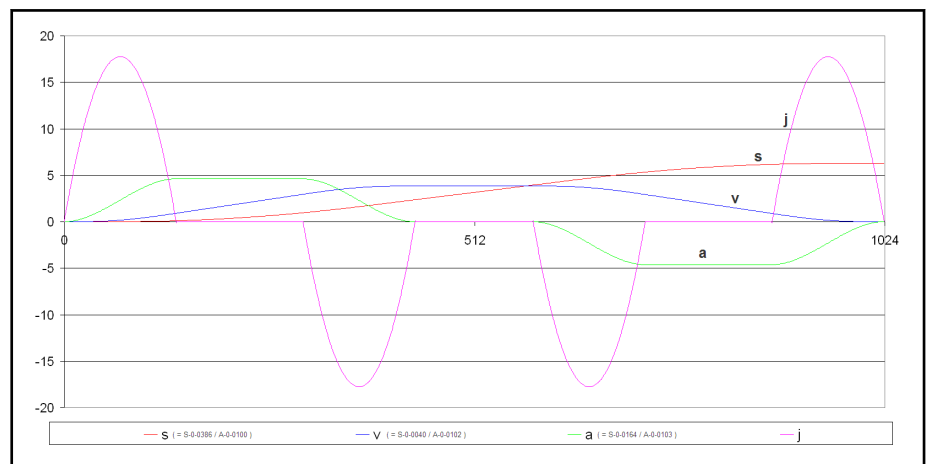
Variant 3: Velocity limit value is not reached, acceleration limit value is reached (variant is rarely used in practice)



MaxJerk = 20 rad/s³; MaxAcc = 5 rad/s²; MaxVel = 5 rad/s;
 MasterVel = 22 rpm; modulo value = 2*Pi rad
 Calculation from parabola parts

Fig. 10-20: Cam profile of the MB_FeedMovementCalcType02 [slave = f(master)]

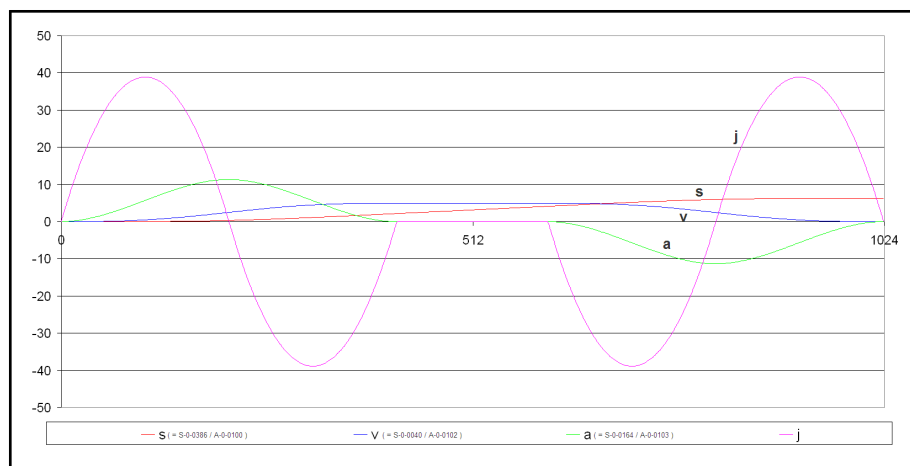
Variant 4: Velocity and acceleration limit values are reached (variant is rarely used in practice)



MaxJerk = 20 rad/s³; MaxAcc = 5 rad/s²; MaxVel = 4 rad/s;
 MasterVel = 21 rpm; modulo value = 2*Pi rad
 Calculation from parabola parts

Fig. 10-21: Cam profile of the MB_FeedMovementCalcType02 [slave = f(master)]

Variant 5: Velocity limit value is reached, acceleration limit value is not reached (variant is rarely used in practice)



MaxJerk = 40 rad/s³; MaxAcc = 20 rad/s²; MaxVel = 5 rad/s;
 MasterVel= 28 rpm; modulo value = 2*Pi rad
 Calculation from parabola parts

Fig. 10-22: Cam profile of the MB_FeedMovementCalcType02 [slave = f(master)]

Error handling The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2.

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR	16#0006	16#0000	Invalid function block state
INPUT_RANGE_ERROR	16#2200	16#0001	"NumberOfElements" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#2230	16#0001	"MaxJerk" input is < minimum
INPUT_RANGE_ERROR	16#2230	16#0002	"MaxAcc" input is < minimum
INPUT_RANGE_ERROR	16#2230	16#0003	"MaxVel" input is < minimum
INPUT_RANGE_ERROR	16#2230	16#0004	"MasterVel" input is < minimum
INPUT_RANGE_ERROR	16#2230	16#0005	"CamShaftDistance" input is < minimum
INPUT_INVALID_ERROR	16#2201	16#0001	Address of the "CamTable" array is zero
INPUT_INVALID_ERROR	16#2231	16#0001	"CamShaftDistance" unequal to the modulo value of the axis reference
INPUT_INVALID_ERROR	16#2231	16#0002	Input values for jerk, acceleration or velocity are higher than preset in the axis parameters
CALCULATION_ERROR	16#2232	16#0001	Velocity of the master axis is too high or the maximum jerk, the maximum acceleration or the maximum velocity is too low.

Tab. 10-15: Error codes of the MB_FeedMovementCalcType02 function block

10.7 MB_CURVETYPE

The "MB_CURVETYPE" enumeration selects between different compensation motions from rest-to-rest.

Element	Value	Description
CURVETYPE_QUADRATIC_PARABOLA	0	Square parabola
CURVETYPE_SIMPLE_SINUSLINE	1	Simple sine line
CURVETYPE_POLY_5	2	Polynomial 5th order
CURVETYPE_SLOPING_SINUSLINE	3	Inclined sine line
CURVETYPE_MODIFIED_ACCELERATIONTRAPEZE	4	Modified acceleration trapezoid
CURVETYPE_MODIFIED_SINUSLINE	5	Modified sine line

Tab. 10-16: Elements of the enumeration type "MB_CURVETYPE"

10.8 Slotters, Introduction and Definitions

A slotter is a rotating roller with one or more knives attached to it which produce slots in the corrugated cardboard sheet in a corrugated cardboard printing machine, thus meaning the knives can cut free the subsequent bottom and lid flaps in a corrugated cardboard box.

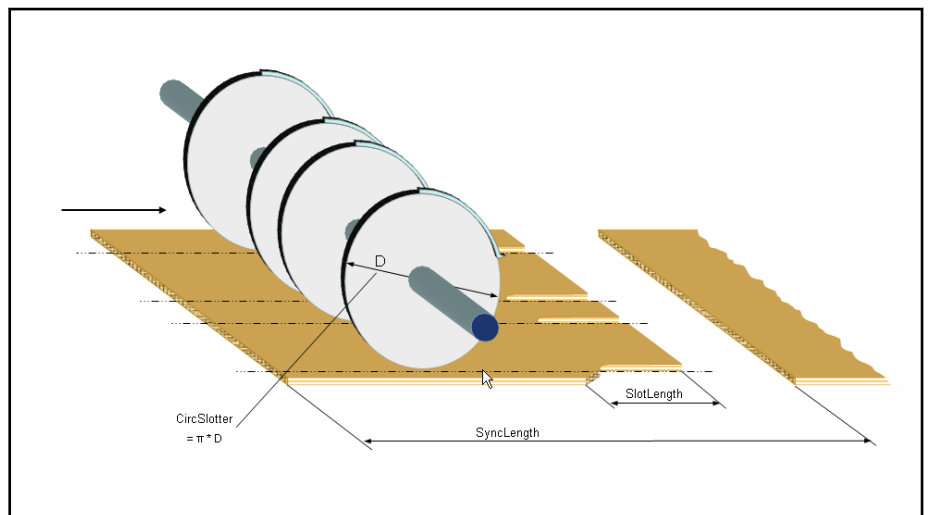


Fig. 10-23: Graphic display of a slotter

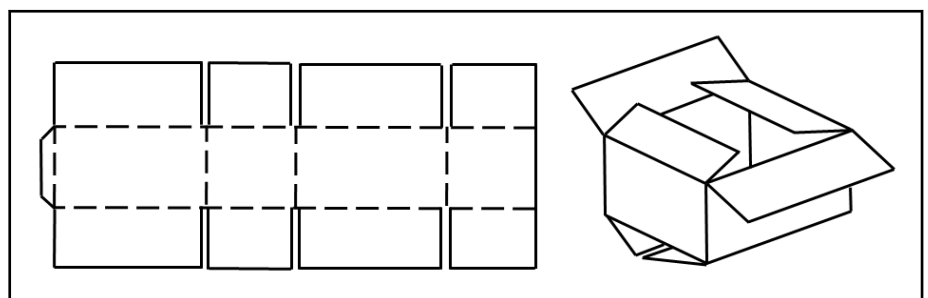


Fig. 10-24: Exemplary depiction of a slotted cardboard box

Several slotters can be arranged in the processing line one after the other in order to increase the number of slots per sheet, or to produce slots longer than the knife for one of the slotter rollers. Single slotters (slotter roller with one knife) and double slotters (slotter roller with two knives) are typical.

Definitions and specifications, whose information is necessary for calculating slotter cams when using the function blocks [MB_SlotterCalcType01](#), [page 664](#), and [MB_SlotterCalcType02](#), [page 676](#), are described in the following.

Direction of rotation, direction of motion, and zero marks

All definitions and references described here assume that the shaft equipped with knives is positioned above the conveying level of the box.

The correct differentiation between the direction of rotation and direction of motion is essential for defining and parameterizing the slotters.

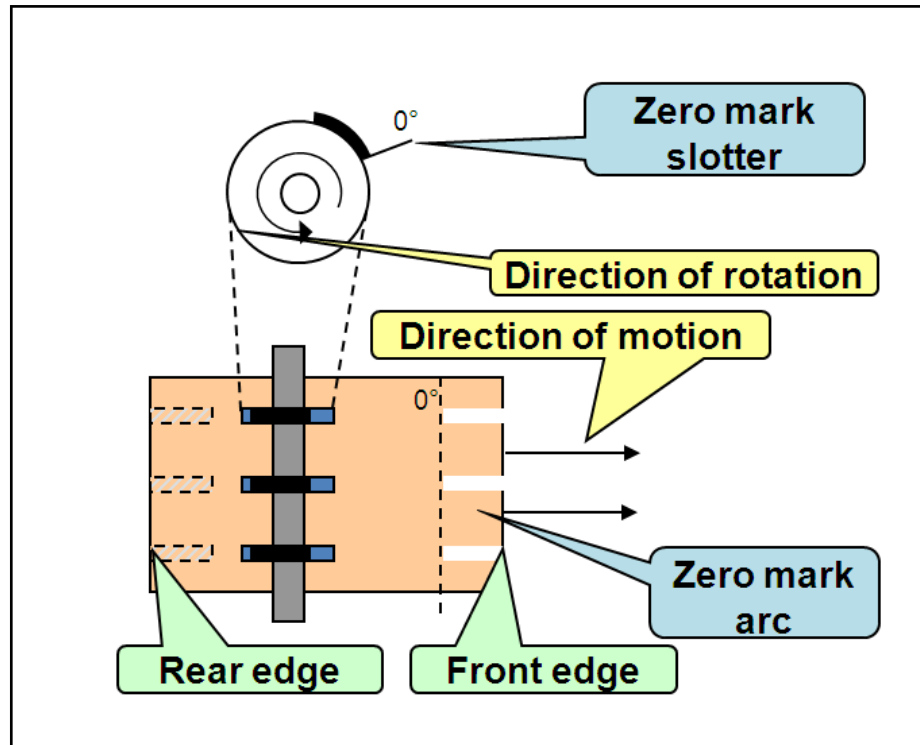


Fig. 10-25: Difference between direction of rotation and direction of motion

Direction of rotation

The direction of rotation refers to the rotary movement of the slotter roller. The direction of motion always refers to the direction of motion of the sheet being slotted. In this case, the counting procedure is always done from the front edge of the sheet - thus from the edge that the slotter roller passes first - in the direction of the rear edge of the sheet. The figure below shows an example of the correct numbering of three points on the slotter roller. According to this, point 1 is the first and point 3 the last point in the direction of rotation.

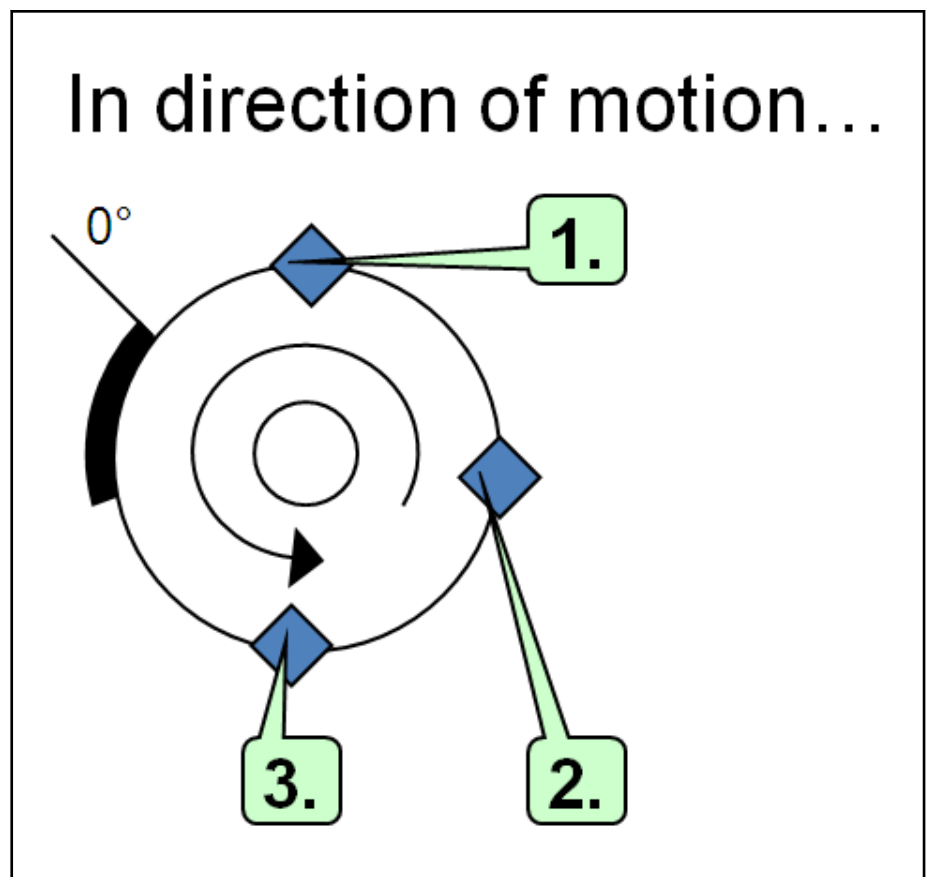


Fig. 10-26: Enumeration method for details in the direction of rotation

Direction of motion

Furthermore, [fig. 10-25 "Difference between direction of rotation and direction of motion" on page 660](#) shows that the direction of motion always refers to the movement of the sheet being slotted. In this case, the counting procedure is always done from the front edge of the sheet - thus from the edge that the slotter roller passes first - in the direction of the rear edge of the sheet. The figure below shows an example of the enumeration method in the direction of motion. According to this, point 1 is the first and point 3 the last point in the direction of motion.

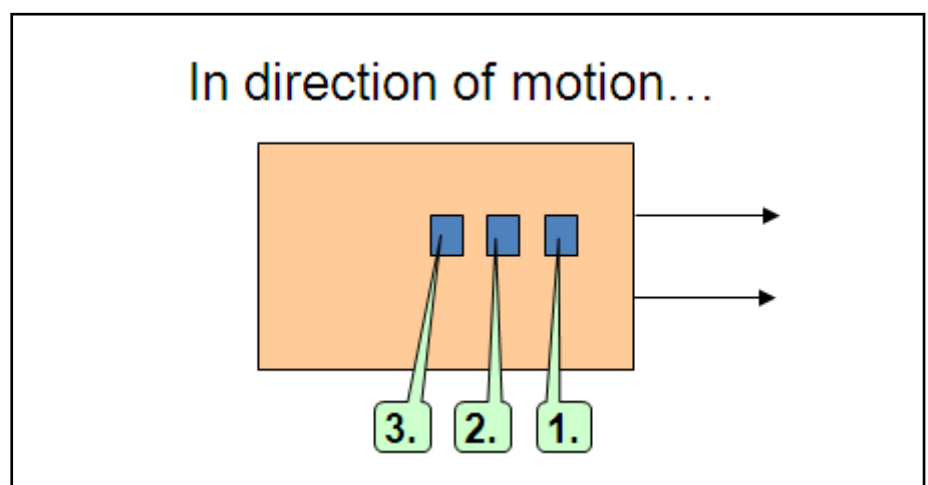


Fig. 10-27: Enumeration method for details in the direction of motion

Slot numbers

One or two slots can be produced per sheet using the slotter function blocks documented [here](#).

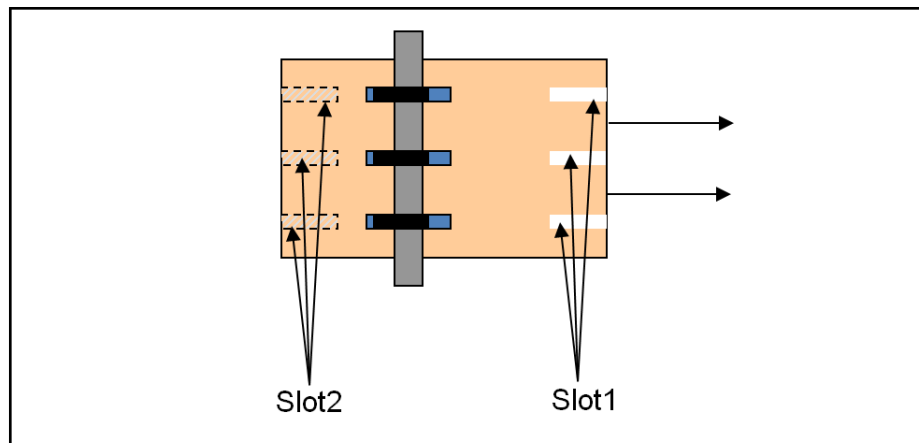
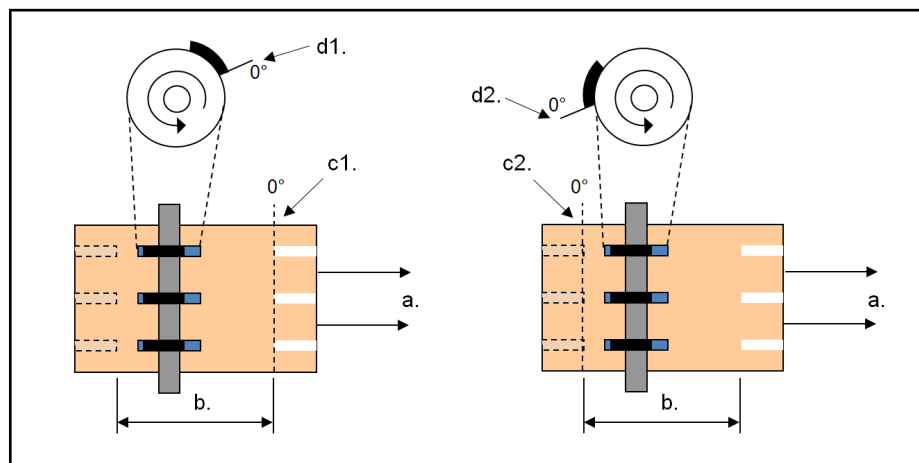


Fig. 10-28: Numbering of slots for several slots in a sheet

The first slot in the direction of motion is always referred to as "Slot 1" and the second slot in the direction of motion as "Slot 2".

Box height and position of the zero marks

The position of the zero marks for the slotter roller defines the relationship between the slotter roller and the sheet being slotted. It is equal to the slotter axis's 0° absolute position.



- a. Direction of motion
- b. Box height ("BoxHeight")
- c1., c2. Possible position of the zero mark for the sheet
- d1., d2. Possible position of the zero mark for the slotter

Fig. 10-29: Definition of the zero mark on the slotter roller and in the sheet being slotted

The position of the zero mark on the sheet defines an edge of the box height ("BoxHeight"). The box height is the area between "Slot1" and "Slot2".

The zero mark always has to be placed next to a knife on the slotter roller. Therefore, in slotters with one knife, there are two possible positions for the zero mark and in slotters with two knives on the cutting roll, there are four possible positions.

By definition, on the sheet being slotted, the zero mark is always placed next to the subsequent box height at Slot 1 or Slot 2. The two different options for positioning the zero mark for a slotter with one knife are depicted in [fig. 10-29 "Definition of the zero mark on the slotter roller and in the sheet being slotted" on page 662](#).

For the slotter function blocks, you need to specify (as the case may be) for the calculation which zero mark position is required on the sheet ("CamProf" input). This defines whether the slotter roller "Slot1" or "Slot2" is slotting.

Knife numbers The slotter function block MB_SlotterCalcType02 can be used for calculating the cams for slotters with two knives on the slotter roller. The zero mark for the slotter roller always having to be located at the start or at the end of a knife defines the knife number.

The knife attached in the direction of rotation in ascending order as the first one on the slotter roller from the position of the zero mark is knife 2, the other one is knife 1. This definition means that Knife 1 always cuts Slot 1 and Knife 2 always cuts Slot 2.

The figure below shows the numbering of the knives for all four possible positions of the zero mark for the cutting roll.

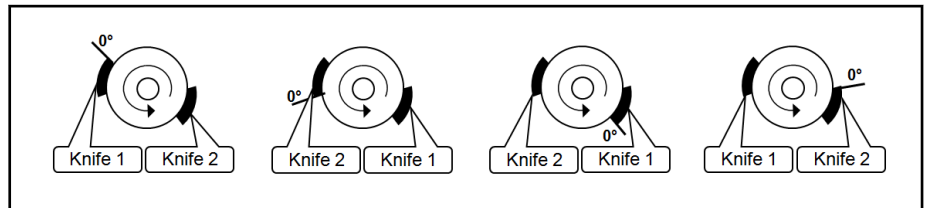


Fig. 10-30: Definition of the knife numbers for slotters with two knives on the slotter roller

Knife distance The slotter function block MB_SlotterCalcType02 can be used for calculating the cams for slotters with two knives on the slotter roller. The knives in this case do not have to be attached on the slotter roller symmetrically, but can have any distance apart from one another.

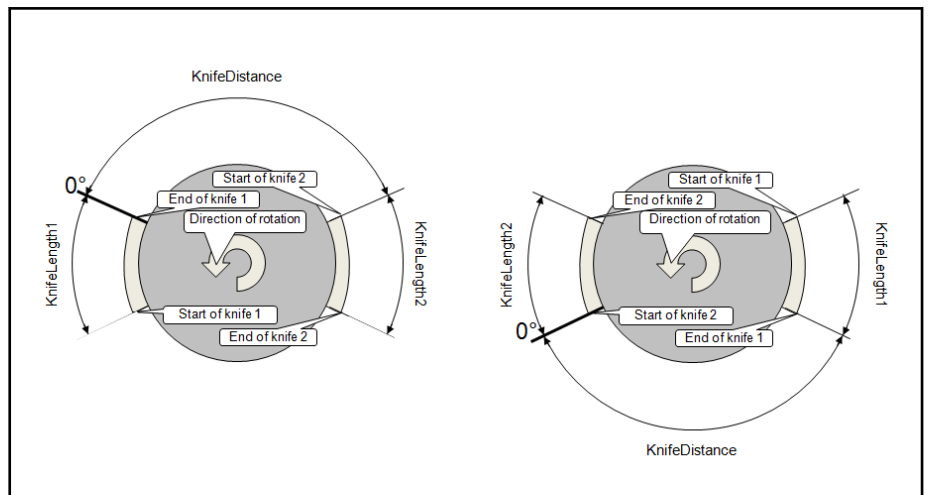


Fig. 10-31: Definition of the knife distance for slotters with two knives on the slotter roller

As shown in [fig. 10-31 "Definition of the knife distance for slotters with two knives on the slotter roller" on page 663](#), the knife distance is always measured from the end of Knife 1 to the start of Knife 2. Since the numbering of the knives is related to the position of the zero mark, the knife distance also depends on the position of the zero mark.

Multislotters If several consecutive slotters together produce an excessively long slot, then these are considered to be multislotters. The definitions and specifications applying to multislotters are depicted graphically in [fig. 10-32 "Definitions and specifications for multislotters" on page 664](#) below.

By contrast to the definition option for a zero mark when using single slotters, for multislotters, the zero mark for the individual slotters is defined in the group. As depicted, Slotter 1 always has its zero mark in the direction of rotation at the end of the knife and Slotter 2 up to n at the start.



The function block cannot intercept incorrect entries at this point. The user is responsible for correctly parameterizing the zero marks for the individual slotters.

The box height is applied between the zero marks of Slotter 1 and Slotter n in the sheet. The zero mark of slotter n-1 is offset by $1/n$ th of "SlotLength2" in the direction of the rear edge of the sheet.

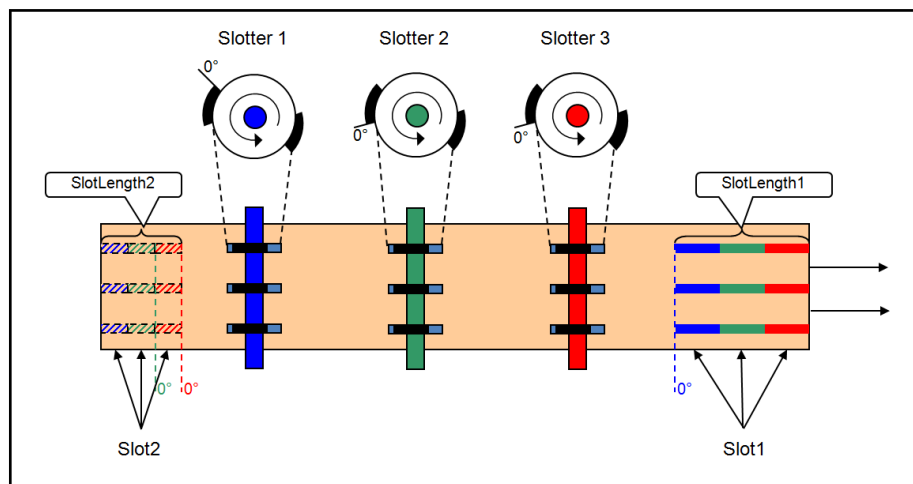


Fig. 10-32: Definitions and specifications for multislotters

- The first slotter that makes contact with the sheet is Slotter 1. The second one, Slotter 2 ... the last slotter in the sequence is Slotter n.
- In order to ensure that the cams are calculated correctly, the function block with the differing input variables has to be called up n times, or n-inherent instances in the function block have to be applied.
- The slots cut free by the multislotters are always specified using their complete length ("SlotLength1" or "SlotLength2"). On no account should the length of an individual partial slots ($\text{SlotLength}/n$) be applied at one function block.
- The individual slotters share "SlotLength1" and "SlotLength2" in equal parts. Each of the n slotters cuts free $1/n$ th of the SlotLength from Slot1 and Slot2.
- The knives for the individual slotters have to be of the same length. In slotters with several knives, the knives that cut together on one slot need to have identical lengths.

10.9 MB_SlotterCalcType01

Brief description

The MB_SlotterCalcType01 function block calculates a cam with 1,024 data point elements for an individual slotter (single slotter) with one knife on the cutting roll, or for one slotter in a group of multislotters, each with one knife on the cutting roll.



Please refer to [Slotters, Introduction and Definitions, page 659](#), for the basis of understanding the function block described below.

The MB_SlotterCalcType01 function block covers the following functionalities:

- One slot per sheet (single slotter). One slot at the start or at the end of the sheet (equivalent to MSV_CamTable14 in Rexroth SYNAX200).

- One slot for each sheet in "Skip feed" mode with one sheet for several machine processes.
- One long slot at the start or at the end of the sheet (multislotter). Two or more slotter rollers operate here on one slot.
- One long slot for each sheet in "Skip feed" mode with one sheet for several machine processes.
- Two slots per sheet (single slotter). One slot at the start of the sheet and one slot at the end of the sheet.
- Two slots for each sheet in "Skip feed" mode with one sheet for several machine processes.
- Two long slots per sheet (multislotter). One long slot at the start of the sheet and one long slot at the end of the sheet.
- Two long slots for each sheet in "Skip feed" mode with one sheet for several machine processes.
- Slotting with slotter discs whose peripheral length (knife cutting circle / "CircSlotter") is less than the machine process ("SyncLength").

This function block always calculates one cam for one slotter. For n-slotters in the multislotter group, the function block has to be called n times, or n-inherent instances in the function block have to be applied. Use the function block (via the "SlotterToCalculate" input) to define which slotters the function block is supposed to calculate the cam for.

Assignment: Target system/library

MLC 14VRS	RMB_TechCam
TEC function package	

Tab. 10-17: Reference table of MB_SlotterCalcType01

Interface description

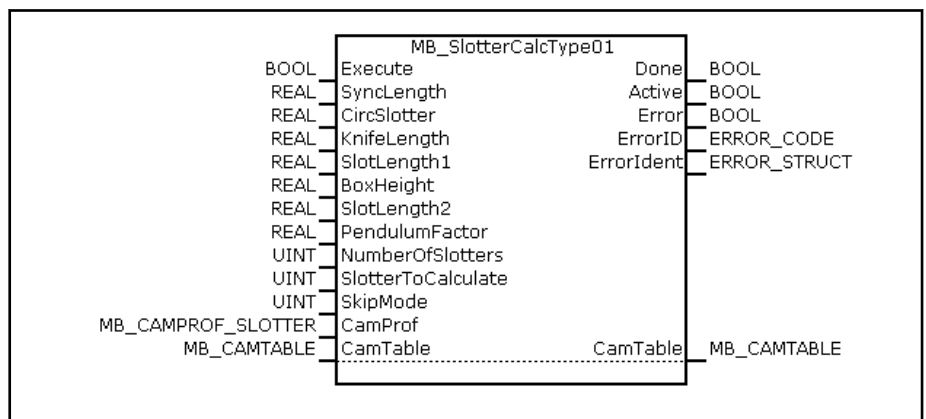


Fig. 10-33: Interface of the function block: MB_SlotterCalcType01

	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Processing of the function block enabled (once, edge-controlled)
	SyncLength	REAL	Synchronous format length, i.e. mechanical processing length of the machine, e.g. circumference of the printing cylinder [u]*
	CircSlotter	REAL	Circumference of the slotter disc or knife cutting circle [u]*
	KnifeLength	REAL	Length of the knife [u]*

	Name	Type	Comment
	SlotLength1	REAL	Total length of the first slot in the sheet in the direction of motion [u]*
	BoxHeight	REAL	Distance between the two slots in the sheet ("SlotLength1" + "BoxHeight" + "SlotLength2" = Sheet length) [u]*
	SlotLength2	REAL	Total length of the second slot in the sheet in the direction of motion [u]*
	PendulumFactor	REAL	Maximum pendulum radius of the compensation motion [%] (without synchronous range) 0%: No pendular motion (standstill at large format length) 100%: Maximum pendular motion (maximum reverse rotation up to the boundary where the knife cuts backwards into the box)
	NumberOfSlotters	UINT	Number of slotters in the multislotter group which together produce the two slots at the start and at the end of the sheet [-]**
	SlotterToCalculate	UINT	Number of the slotter for which a calculation is going to be made for the cam when the function block is called [-]**
	SkipMode	UINT	Number of machine cycles per sheet - 1 [-]**
	CamProf	MB_CAMPROF_SLOTTER	Selection of the cam profile and position of the zero mark on the slotter roller MB_CAMPROF_SLOTTER, page 686
VAR_OUTPUT	Done	BOOL	Calculation of the cam elements completed without errors. List of cam elements valid
	Active	BOOL	Calculation of the cam elements active. List of cam elements invalid
	Error	BOOL	Calculation of the cam elements completed with error. Output variables "ErrorID" and "ErrorIdent" valid
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, the output contains detailed error information Error handling, page 675
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements

* The unit [u] means "unit". This can be any unit such as [mm] or [inches]. Ensure that the selected unit is always used.
 ** The unit [-] stands for a quantity without a unit.

Tab. 10-18: Interface variables of the MB_SlotterCalcType01 function block

Min./max. and default values

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL	FALSE	TRUE	FALSE	Continuous
SyncLength	REAL	>0.0	n.def.	0.0	Rising edge at "Execute"
CircSlotter	REAL	>0.0	n.def.	0.0	Rising edge at "Execute"
KnifeLength	REAL	>0.0	"CircSlotter"	0.0	Rising edge at "Execute"
SlotLength1	REAL	>0.0	"KnifeLength" * "NumberOfSlot- ters"	0.0	Rising edge at "Execute"
BoxHeight	REAL	0.0	n.def.	0.0	Rising edge at "Execute" and Slot- Length2 > 0
SlotLength2	REAL	0.0	"KnifeLength" * "NumberOfSlot- ters"	0.0	Rising edge at "Execute"
PendulumFactor	REAL	0.0	100.0	95.0	Rising edge at "Execute"
NumberOfSlotters	UINT	1	10	1	Rising edge at "Execute"
SlotterToCalculate	UINT	1	10	1	Rising edge at "Execute"
SkipMode	UINT	0	10	0	Rising edge at "Execute"
CamProf	MB_CAM- PROF_SLOTTER	0	3	0	Rising edge at "Execute"

Tab. 10-19: Min./max. and default values of the FB MB_SlotterCalcType01

Signal-time diagram

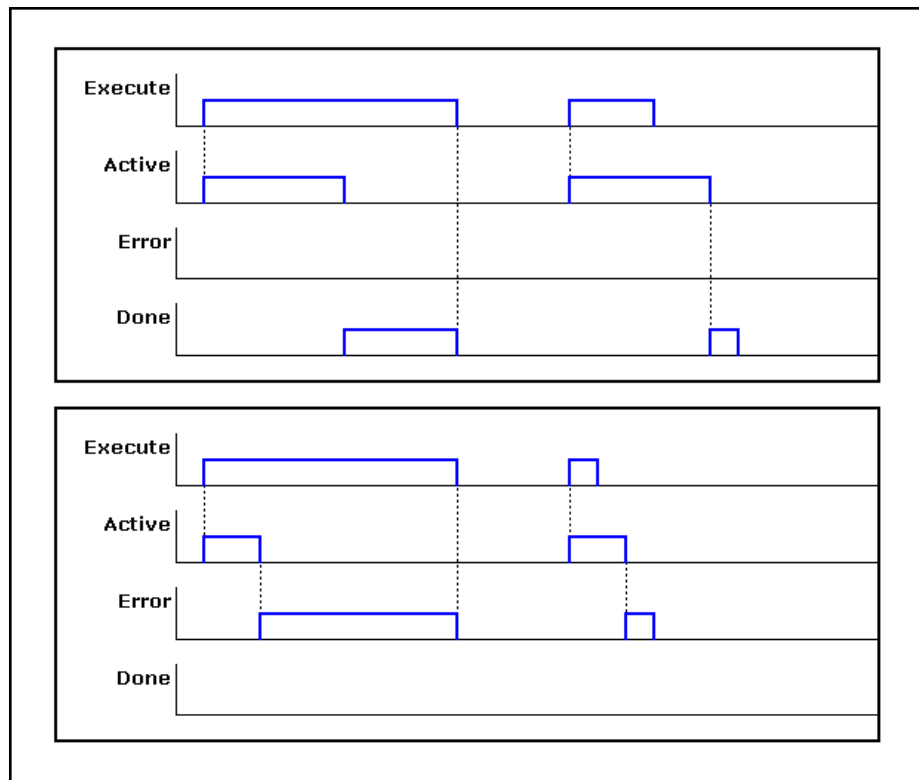


Fig. 10-34: Signal-time diagram of the FB MB_SlotterCalcType01

Cam characteristics

The cam table of the "MB_SlotterCalcType01" function block exhibits the following characteristics:

- For cyclical processing, the cam profile can be loaded as cam parameters into the drive (P-0-0072, P-0-0092, P-0-0780 and P-0-0781) or into the control (C-0-2001 to C-0-2099).
- The MotionProfile characterized by the cam table is constant up to the acceleration (s, v, a) and shows a jump in the jerk (j).
- The function block is able to generate profiles for endlessly rotating cams without gear reduction (P-0-0086 Bit4 = 0) as well as for superimposed profiles with gear reduction (P-0-0086 Bit4 = 1).
- The zero mark for the slotter roller can be (in the direction of rotation) both at the start and at the end of the knife. Please refer to [Box height and position of the zero marks](#), page 662
- In order to avoid a negative impact on the cycle time of the PLC, the number of calculation cycles for an instance of MB_SlotterCalcType01 is automatically determined by the current system load.
- With excessively long synchronous ranges, the compensation motion can lead to a reverse rotation. The maximum reverse rotation for the slot knife can be specified.
- With cams that cut two slots per sheet, please make sure (by parameterizing an electronic gear) that both rotations are contained in the one cam created by the function block. The function block calculates the synchronous ranges such that the slotter roller turns at material velocity in accordance with the electronic gear during the synchronous ranges.
- In SkipFeed mode, it is assumed that a product covers several machine cycles ("synchronous format lengths"). SkipMode n stretches an operation rotation to n+1 machine cycles. Use an electronic gear to ensure

that the cam produced by the function block is extended to the appropriate number of machine cycles. In SkipFeed mode, the function block calculates the synchronous ranges such that the slotter roller turns at material velocity in accordance with the electronic gear during the synchronous ranges.

Features in the multislotter group

When using multislotters, the following further points need to be observed:

- Multislotters that cut a long slot at the start or at the end of a sheet can be implemented on the sheet both by using a common cam ("NumberOfSlotters"=1) as well as shifted zero marks.
- The first slotter making contact with the sheet is Slotter 1, the last one is Slotter n.
- Slotter 1 always initially cuts a window in the sheet to produce Slot 1 (in order to utilize the slotter in an optimally dynamic way). Slotters 2 to n then cut the slot free. On the other side of the sheet (Slot 2), Slotter 1 cuts the piece directly on the edge of the sheet and Slotters 2 to n enlarge the slot up to "SlotLength2".
- The knives for the individual slotters have to be of the same length.
- The individual slotters share "SlotLength1" and "SlotLength2" in equal parts. Each of the n slotters cuts free 1/n th of the SlotLength from Slot1 and Slot2.
- The box height is applied between the zero marks of Slotter 1 and Slotter n in the sheet. The zero mark for slotter n-1 is offset by 1/n th of "SlotLength2" in the direction of the rear edge of the sheet.
- At inputs "SlotLength1" and "SlotLength2", the entire length of Slots 1 and 2 need to be applied and not the length of the partial slot which the slotter being calculated produces.
- At the "NumberOfSlotters" input, a definition is needed as to how many slotters are located in the multislotter group and work together on the two slots.
- At the "SlotterToCalculate" input, a definition is needed for which of the "NumberOfSlotters" slotters the cam is going to be calculated for. The slotters are numbered in the order in which they make contact with the sheet (Slotter 1 makes contact with the sheet firstly, Slotter2 secondly, ...).
- Since Slotter1 cuts a window in the material for Slot1, it has to run synchronously with the material velocity for the entire length of its knife. All other slotters only run synchronously for their respective partial slot lengths. This is why Slotters 2-n are able to respond dynamically more favorably than Slotter 1.

Functional description

The target is to maintain the circumferential velocity of the knife roller synchronous to the speed of the material web in a synchronous range determined by the cutting process (specified by "SlotLength1" and "SlotLength2"). To do this, the function block calculates one time a cam table with 1,024 data point elements on the basis of the input variables after processing has been enabled using "Execute".

The explicit significance of the input variables is shown in the depiction below.

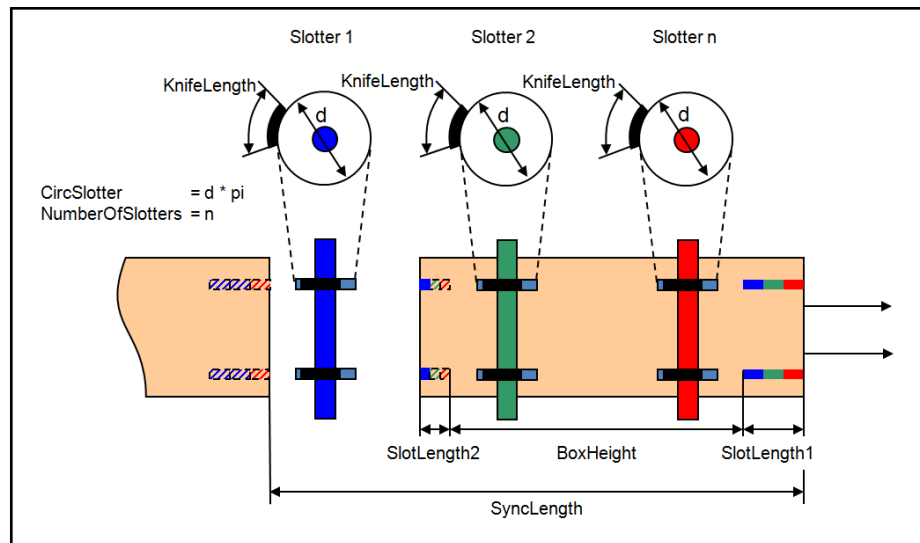


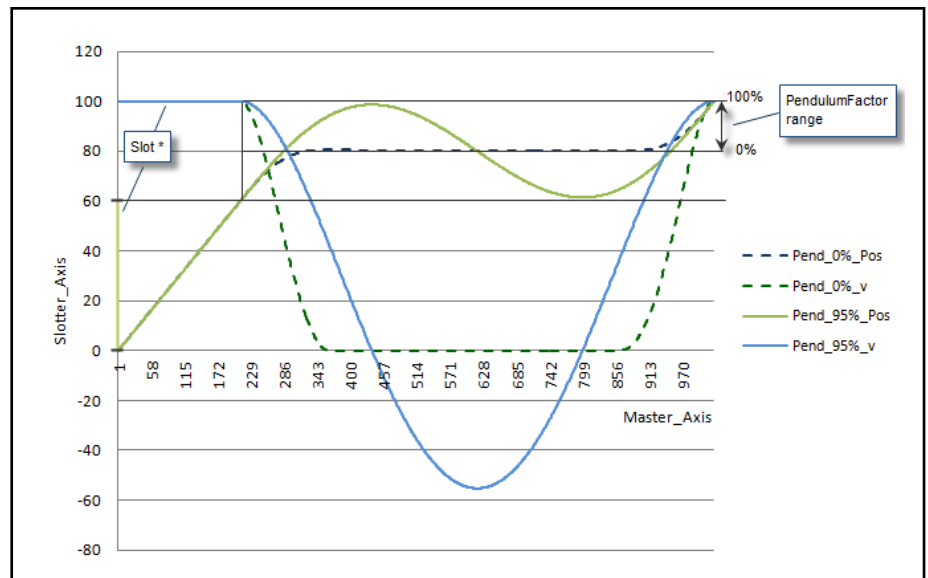
Fig. 10-35: Significance of the input variables for the MB_SlotterCalcType01 function block

In order to be able to cut formats which are longer than the circumference of the cutting roll synchronously with only one cutting roll, there has to be a compensation motion calculated by the function block. This compensation motion is characterized by a continuity up to the acceleration.

PendulumFactor

If the ratios between the box lengths and slot lengths as well as slotter disc circumferences are unfavorable, this may sometimes result in reverse rotations in the compensation motions. These are indeed desired, since the dynamic load of the motor then improves owing to the peak accelerations.

In extreme cases, the reverse rotation can result in the knife plunging into the box against the direction of transport. In order to prevent this from happening, the "PendulumFactor" input is available to the user and can be used to restrict the maximum reverse rotation in the compensation motion ($0\% < \text{"PendulumFactor"} < 100\%$) or to suppress it completely ($\text{"PendulumFactor"} = 0\%$). The compensation motion between the synchronous ranges corresponds with $\pm 100\%$ ($\text{"PendulumFactor"} = \pm 100\%$).



Pend_0%_Pos Cam table when "PendulumFactor" = 0%

Pend_0%_v Velocity when "PendulumFactor" = 0%

Pend_95%_Pos Cam table when "PendulumFactor" = 95%

Pend_95%_v Velocity when "PendulumFactor" = 95%

Slot* Synchronous velocity during material encroachment

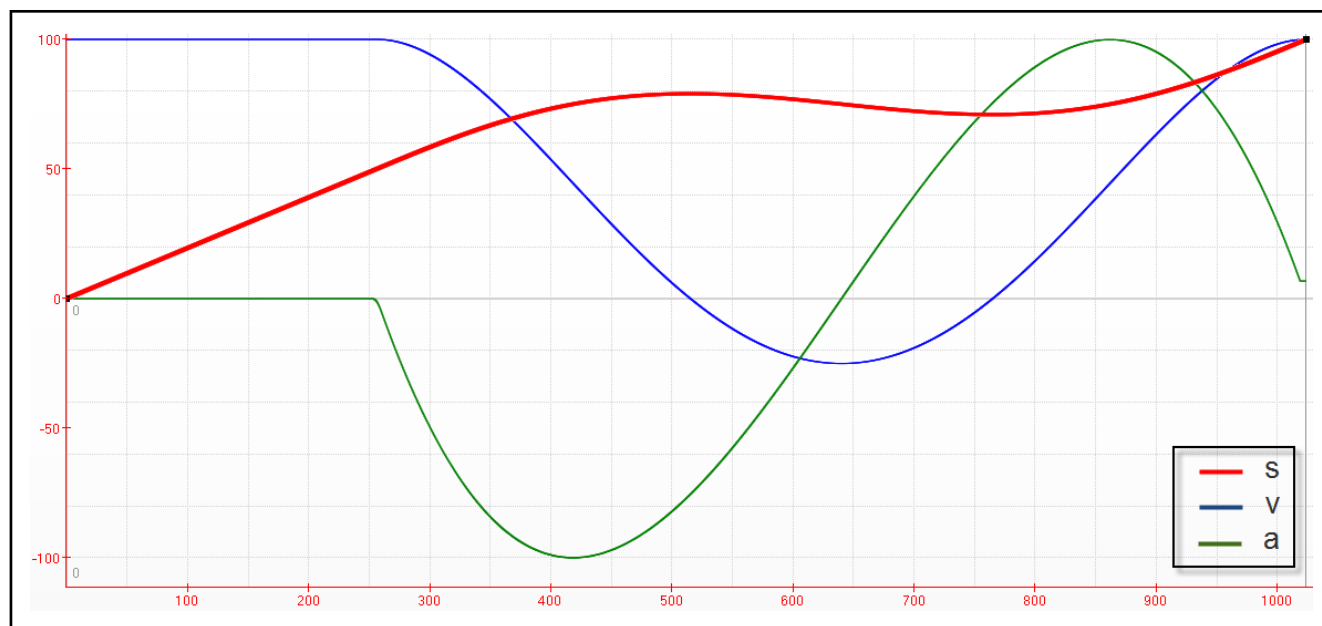
Fig. 10-36: Explaining the "PendulumFactor" input using cam tables

Function variants

Different function variants are possible using function block MB_SlotterCalc-Type01 through the use of various preassignments for the inputs:

Example 1

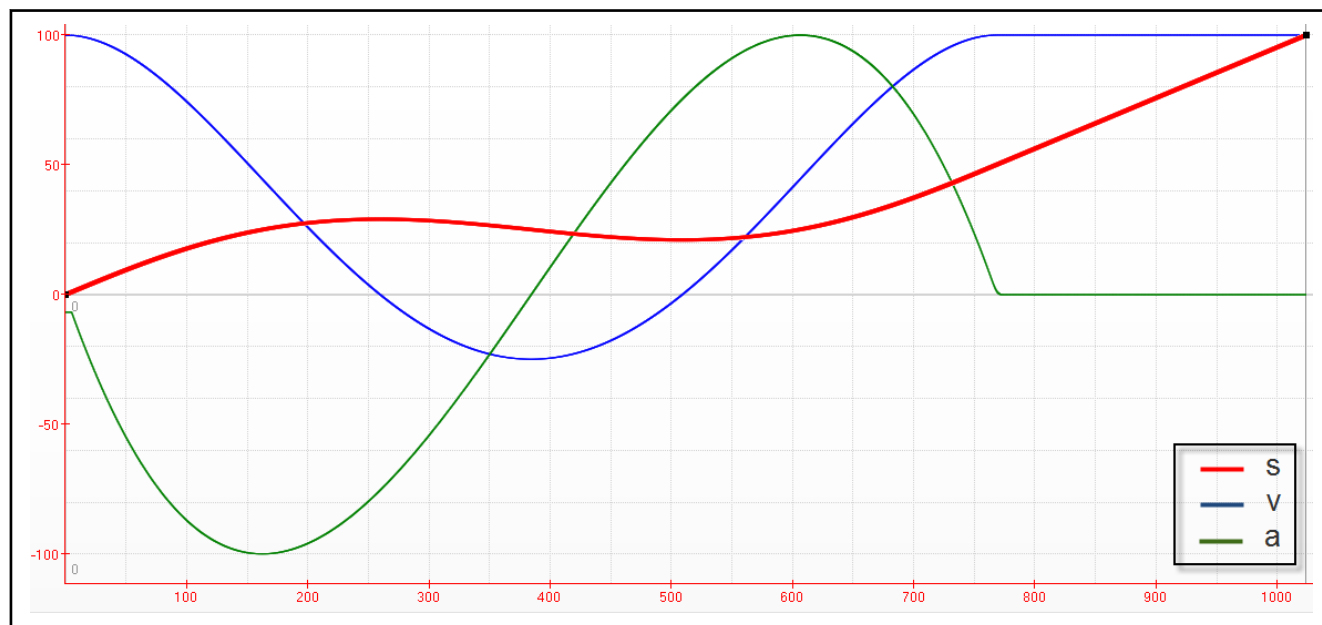
The first variant [Example of motion profile 1, page 672](#), is suitable for using a slotter to cut formats whose cutting circle "CircSlotter" is smaller than the mechanical processing length of the machine "SyncLength". The effective cutter length is adjusted to the material transportation velocity in the tool gearing. The remaining disc circumference of the slotter without cutting knife is defined as the area for a compensation motion.



Example 1: Zero mark in front of the knife, compensation motion at the end (SyncLength = 1024; CircSlotter = 512; SlotLength1 = 256; SkipMode = 0; CamProf = 0)

Fig. 10-37: Example of motion profile 1

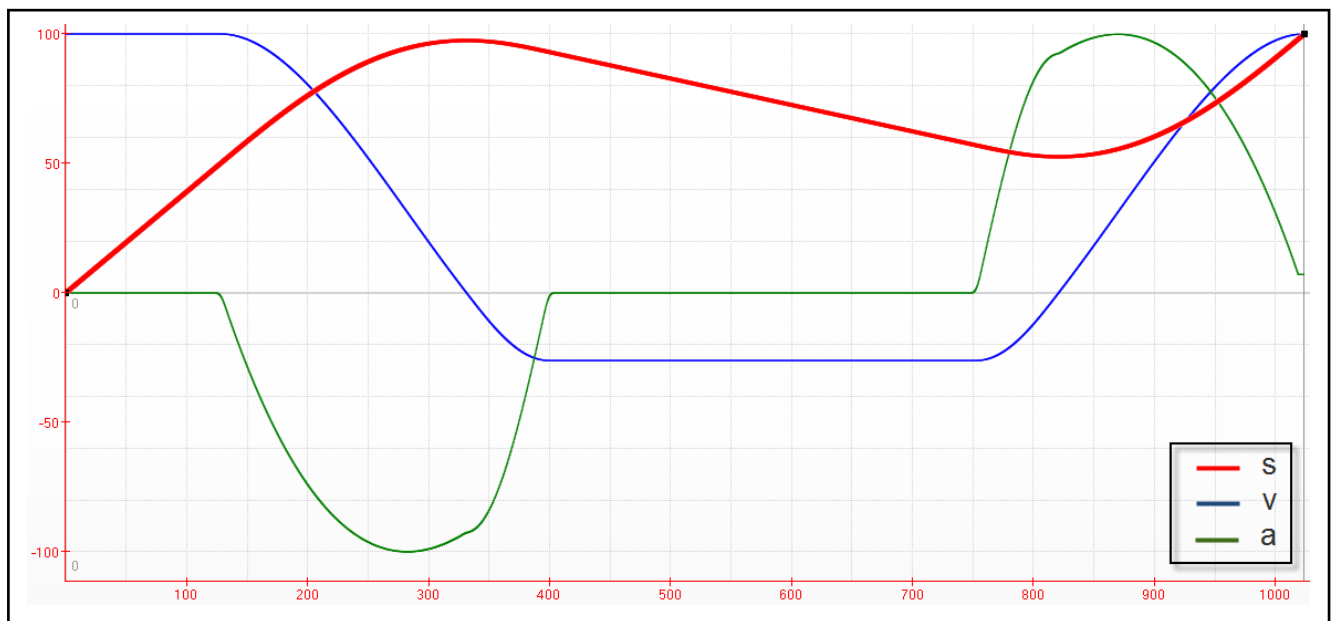
Example 2 In all single slotter variants "NumberOfSlotters" = 1, the zero mark for the slotter roller can be defined as being in front of or behind the knife by parameterizing the "CamProf" input. In the following variant [fig. 10-38 "Example of motion profile 2" on page 672](#), the first variant [fig. 10-37 "Example of motion profile 1" on page 672](#) was taken up again representatively, and a compensation motion was selected at the start and the synchronous motion at the end using the "CamProf" input.



Example 2: Zero mark after the knife, compensation motion at the start (SyncLength = 1024; CircSlotter = 512; SlotLength1 = 256; SkipMode = 0; CamProf = 2)

Fig. 10-38: Example of motion profile 2

Example 3 The third version [fig. 10-39 "Example of motion profile 3" on page 673](#) is suitable for the processing of excessively long formats, i.e. formats larger than the maximum synchronous format length (normal in "SkipMode"). The slotting process is stretched to several machine cycles, whereupon the cutting area of the knife remains synchronous to the surface velocity ("SkipMode" = n). In the course of n+1 machine cycles, the slotter rotates with one mechanical rotation. While the area with the effective cutter length ("Slot-Length1") travels synchronously to the material, a compensation motion is carried out in the remaining motion until the n+1 cycle is completed. In addition, the mode of operation of the pendulum limit can be recognized below. Specifying the "PendulumFactor" = 90 stipulates that the reverse rotation is limited in the compensation range to $75\% + 25\% \cdot 0.9 = 97.5\%$.



Example 3: Zero mark in front of the knife, compensation motion at the end (SyncLength = 1024; CircSlotter = 512; Slot-Length1 = 256; PendulumFactor = 90; SkipMode = 1; CamProf = 0)

Fig. 10-39: Example of motion profile 3

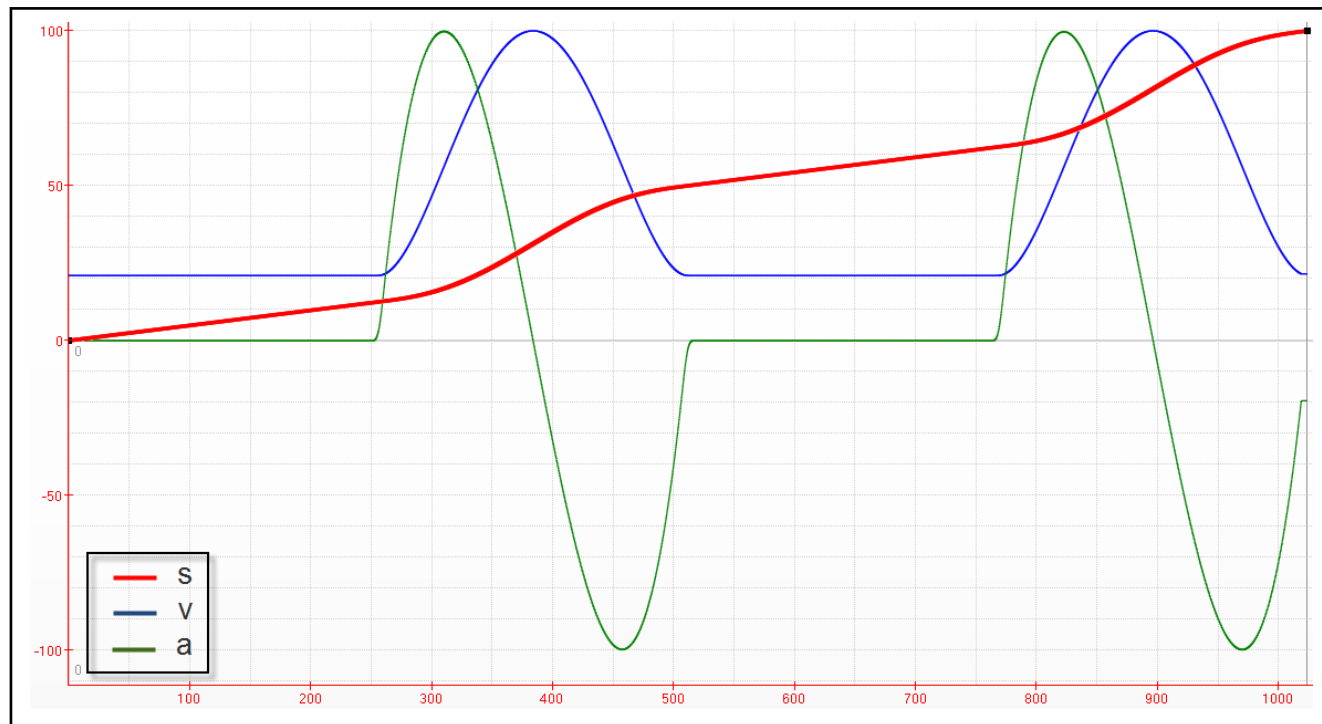
Example 4 In the fourth variant [fig. 10-40 "Example of motion profile 4" on page 674](#), a cam is shown generating a slot at the start of the sheet and a slot at the end of the sheet. With cams that cut two slots per sheet, the slotter roller has to turn completely two times whilst the sheet passes it. In order to be able to achieve two rotations per master axis revolution, the two rotations have to be included in the one cam produced by the function block.



In format lengths which are almost equivalent to the synchronous format, Slot2 in the first sheet and Slot1 in the following sheet are located directly one after the other. The slotter roller has to carry out an almost complete revolution in a very short period of time so that the knife can penetrate the second sheet at the correct point in time. This puts great demands on the acceleration capacity of the drives and motors used, and is not dynamically favorable.

Under this condition, it is recommended that the function block is not operated when "SkipFeed" = 0.

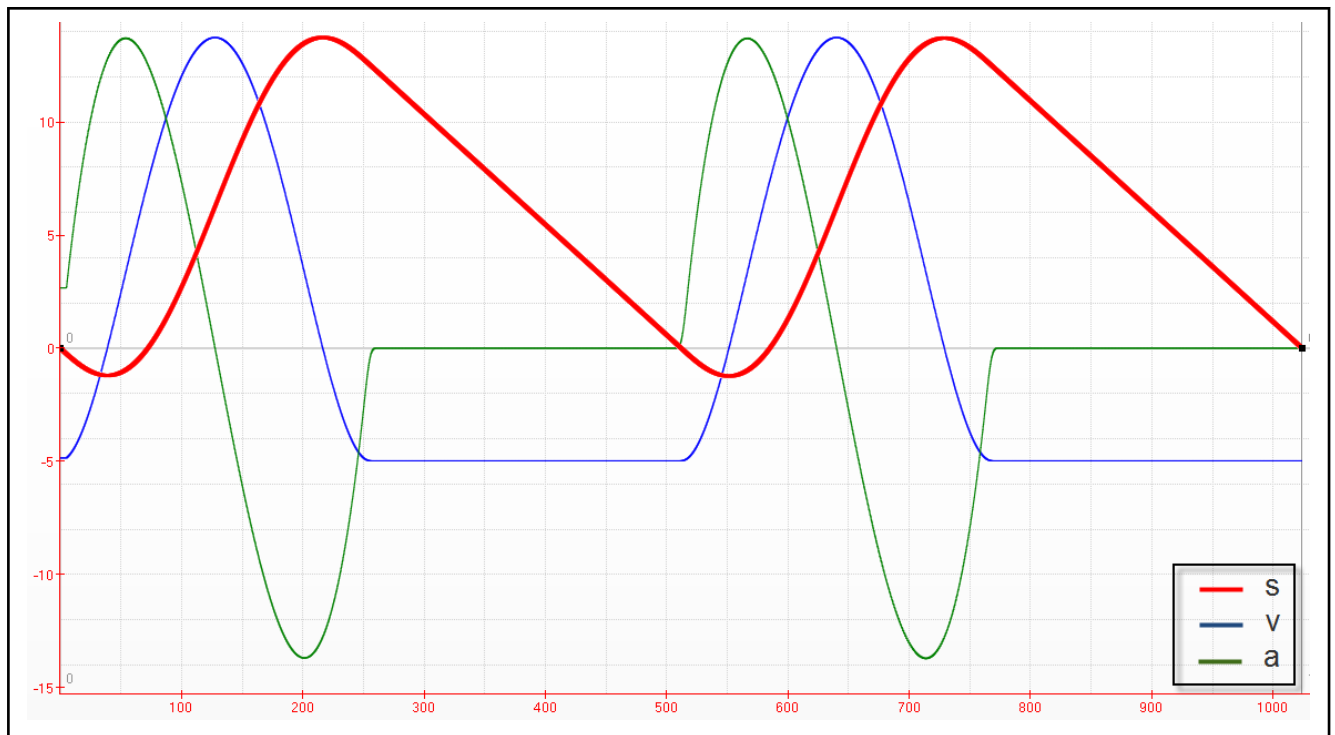
The cam is characterized both by two synchronous ranges (specified by "SlotLength1" and "SlotLength2"), and two compensation motions between the synchronous ranges. While in the synchronous ranges, the knife attached on the slotter roller finds itself in material encroachment and cuts the two slots.



Example 4: Zero mark in front of the knife (SyncLength = 1024; CircSlotter = 1024; KnifeLength = 256; BoxHeight = 256; SlotLength1 = SlotLength2 = 256; SkipMode = 0; CamProf = 0)

Fig. 10-40: Example of motion profile 4

Example 5 Based on the previous example 4 [fig. 10-40 "Example of motion profile 4" on page 674](#), example 5 [fig. 10-41 "Example of motion profile 5" on page 675](#) representatively follows for an additive curve. By using the "CamProf" input, however, this time both a superimposed profile with gear reduction and a compensation motion were selected at the start and a synchronous motion at the end.



Example 5: Zero mark after of the knife (SyncLength = 1024; CircSlotter = 1024; KnifeLength = 256; BoxHeight = 256; SlotLength1 = SlotLength2 = 256; SkipMode = 0; CamProf = 3)

Fig. 10-41: Example of motion profile 5

The cam table to be calculated is saved in the "CamTable" structure in case of an error-free processing of the function block, whereupon the validity of the cam elements is signaled by the output "Done".

If an error occurs while processing the function block, this is signaled by the "Error" output. The elements of the "CamTable" structure are not updated in case of error.

Error handling

The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2.

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR, 16#0005	16#0006	16#0000	Invalid function block state
CALCULATION_ERROR, 16#0007	16#000C	16#0000	Division by Zero
INPUT_INVALID_ERROR, 16#0001	16#2201	16#0001	Address of the "CamTable" array is ZERO
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0001	"SyncLength" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0002	"CircSlotter" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0003	"KnifeLength" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0004	"BoxHeight" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0005	"SlotLength1" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0006	"SlotLength2" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0007	"PendulumFactor" is smaller than the minimum or higher than the maximum value

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0008	"SkipMode" is smaller than the minimum or higher than the maximum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0009	"CamProf" is smaller than the minimum or higher than the maximum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0010	"NumberOfSlotters" is smaller than the minimum or higher than the maximum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0011	"SlotterToCalculate" is smaller than the minimum value or higher than "NumberOfSlotters"
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0012	"CamTable.NumberOfElements" is not equal to 1024
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0001	"KnifeLength" is greater than "CircSlotter"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0002	"SlotLength1" is greater than "KnifeLength" * "NumberOfSlotters"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0003	"SlotLength2" is greater than "KnifeLength" * "NumberOfSlotters"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0004	"SlotLength1" + "BoxHeight" + "SlotLength2" is greater than "SyncLength" * ("SkipMode" + 1)
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0005	"CamProf" value is not valid

Tab. 10-20: Error codes of the MB_SlotterCalcType01 function block

10.10 MB_SlotterCalcType02

Brief description The MB_SlotterCalcType02 function block calculates a cam with 1,024 data point elements for an individual slotter (single slotter) with two knives on the cutting roll, or for one slotter in a group of multislotters, each with two knives on the cutting roll.

The MB_SlotterCalcType02 function block covers the following functionalities:

- Two slots per sheet (single slotter). One slot at the start of the sheet and one slot at the end of the sheet.
- Two slots for each sheet in "Skip feed" mode with one sheet for several machine processes.
- Two long slots per sheet (multislotter). One long slot at the start of the sheet and one long slot at the end of the sheet.
- Two long slots for each sheet in "Skip feed" mode with one sheet for several machine processes.

This function block always calculates only one cam for one slotter. For n-slotters in the multislotter group, the function block has to be called n times. Use the function block (via the "SlotterToCalculate" input) to define which slotters the function block is supposed to calculate the cam for.

Assignment: Target system/library

MLC 14VRS TEC function package	RMB_TechCam

Tab. 10-21: Reference table of MB_SlotterCalcType02

Interface description Type02

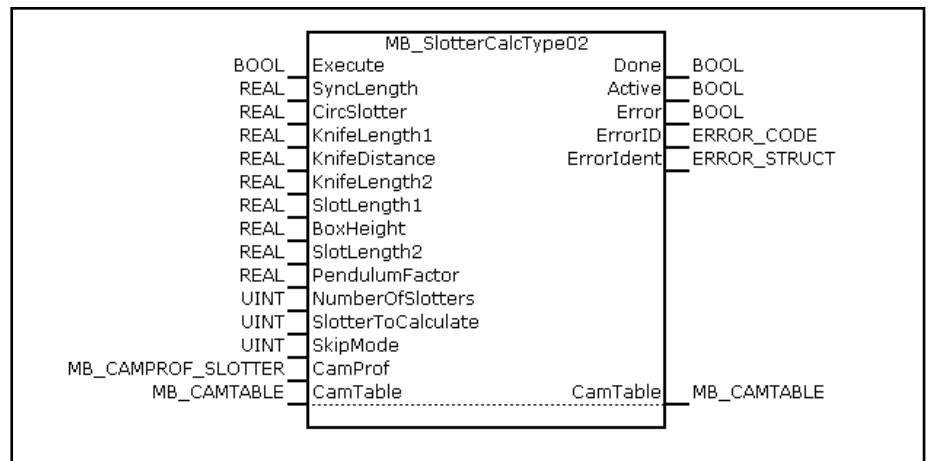


Fig. 10-42: Function Block: MB_SlotterCalcType02

	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Processing of the function block enabled (once, edge-controlled)
	SyncLength	REAL	Synchronous format length, i.e. mechanical processing length of the machine, e.g. circumference of the printing cylinder [u]*
	CircSlotter	REAL	Circumference of the slotter disc or knife cutting circle [u]*
	KnifeLength1	REAL	Length of the first knife [u]*
	KnifeDistance	REAL	Distance between the end of Knife 1 and the start of Knife 2 [u]*
	KnifeLength2	REAL	Length of the second knife [u]*
	SlotLength1	REAL	Total length of the first slot in the sheet in the direction of motion [u]*
	BoxHeight	REAL	Distance between the two slots in the sheet ("SlotLength1" + "BoxHeight" + "SlotLength2" = Sheet length) [u]*
	SlotLength2	REAL	Total length of the second slot in the sheet in the direction of motion [u]*
	PendulumFactor	REAL	Maximum pendulum radius [%] 0%: No pendular motion (standstill at large format length) 100%: Maximum pendular motion (maximum reverse rotation at large format length)
	NumberOfSlotters	UINT	Number of slotters in the multislotter group which together produce the two slots at the start and at the end of the sheet [-]
	SlotterToCalculate	UINT	Number of the slotter for which a calculation is going to be made for the cam when the function block is called [-]
	SkipMode	UINT	Number of machine cycles per sheet - 1 [-]
	CamProf	MB_CAMPROF_SLOTTER	Selection of the cam profile and position of the zero mark on the slotter roller MB_CAMPROF_SLOTTER, page 686
VAR_OUTPUT	Done	BOOL	Calculation of the cam elements completed without errors. List of cam elements valid
	Active	BOOL	Calculation of the cam elements active. List of cam elements invalid

	Name	Type	Comment
	Error	BOOL	Calculation of the cam elements completed with error. Output variables "ErrorID" and "ErrorIdent" valid
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, the output contains detailed error information Error handling, page 685
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements

*

The unit [u] means "unit". This can be any unit such as [mm] or [inches]. Ensure that the selected unit is always used.

Tab. 10-22: Interface variables of the FB MB_SlotterCalcType02

Min./max. and default values

The following table lists the min./max. and default values of the function block inputs.

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL	FALSE	TRUE	FALSE	Continuous
SyncLength	REAL	>0.0	n.def.	0.0	Rising edge at "Execute"
CircSlotter	REAL	>0.0	n.def.	0.0	Rising edge at "Execute"
KnifeLength1	REAL	>0.0	n.def.	0.0	Rising edge at "Execute"
KnifeDistance	REAL	>0.0	n.def.	0.0	Rising edge at "Execute"
KnifeLength2	REAL	>0.0	n.def.	0.0	Rising edge at "Execute"
SlotLength1	REAL	>0.0	n.def.	0.0	Rising edge at "Execute"
BoxHeight	REAL	>0.0	n.def.	0.0	Rising edge at "Execute" and SlotLength2 > 0
SlotLength2	REAL	>0.0	n.def.	0.0	Rising edge at "Execute"
PendulumFactor	REAL	0.0	100.0	95.0	Rising edge at "Execute"
NumberOfSlotters	UINT	1	10	1	Rising edge at "Execute"
SlotterToCalculate	UINT	1	10	1	Rising edge at "Execute"
SkipMode	UINT	0	10	0	Rising edge at "Execute"
CamProf	MB_CAM-PROF_SLOTTER	0	3	0	Rising edge at "Execute"

Tab. 10-23: Min./max. and default values of the FB MB_SlotterCalcType02

Signal-time diagram

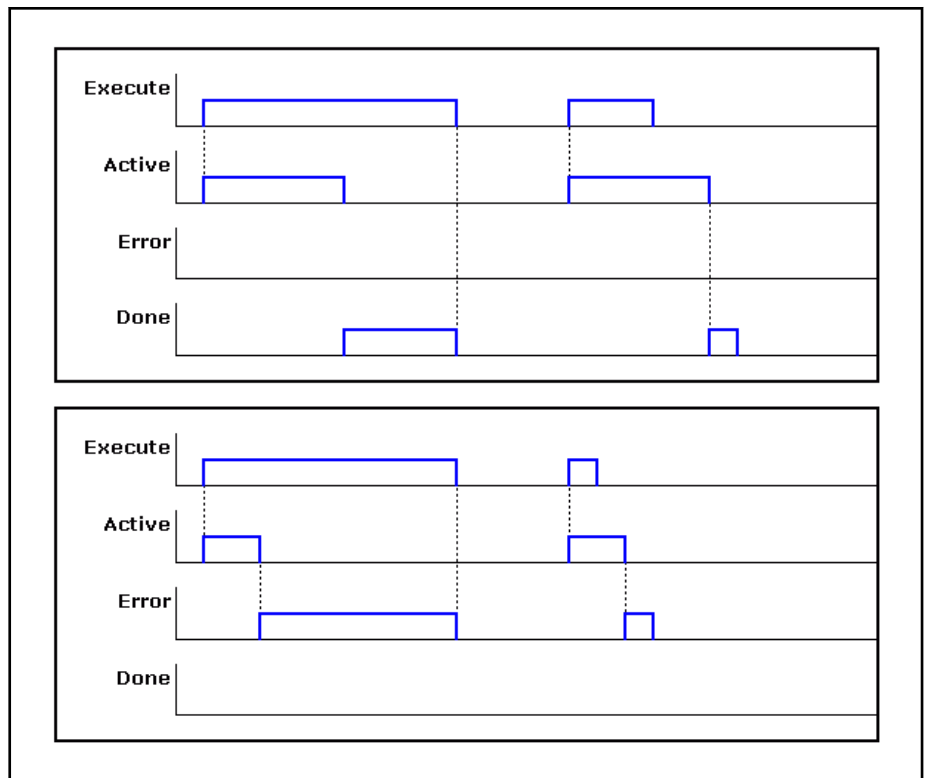


Fig. 10-43: Signal-time diagram of the MB_SlotterCalcType02 function block

Cam characteristics

The cam table of the "MB_SlotterCalcType02" function block exhibits the following characteristics:

- For cyclical processing, the cam profile can be loaded as cam parameters into the drive (P-0-0072, P-0-0092, P-0-0780 and P-0-0781) or into the control (C-0-2001 to C-0-2099).
- The MotionProfile characterized by the cam table is constant up to the acceleration (s, v, a) and shows a jump in the jerk (j).
- The function block is able to generate profiles for endlessly rotating cams without gear reduction (P-0-0086 Bit4 = 0) as well as for superimposed profiles with gear reduction (P-0-0086 Bit4 = 1).
- The zero mark for the slotter roller can be (in the direction of rotation) both at the start and at the end of the knife.
- The algorithm to calculate the cam data points is distributed across several cycles, i.e. calls of the instance to avoid a considerable effect on the PLC cycle time.
- With excessively long synchronous areas, the compensation motion can lead to a reverse rotation. The maximum reverse rotation for the slot knife can be specified.
- In SkipFeed mode, it is assumed that a product covers several machine cycles ("synchronous format lengths"). SkipMode n stretches an operation rotation to n+1 machine cycles. Use an electronic gear to ensure that the cam produced by the function block is extended to the appropriate number of machine cycles.

Features in the multislotter group

When using multislotters, the following further points need to be observed:

- The first slotter making contact with the sheet is Slotter 1, the last one is Slotter n.

- Slotter 1 initially always cuts a window in the sheet in order to produce Slot 1. Slotters 2 to n then cut the slot free. On the other side of the sheet (Slot 2), Slotter 1 cuts the piece directly on the edge of the sheet and Slotters 2 to n enlarge the slot up to "SlotLength2".
- The knives for the individual slotters have to be of the same length.
- The individual slotters share "SlotLength1" and "SlotLength2" in equal parts. Each of the n slotters cuts free 1/n th of the SlotLength from Slot1 and Slot2.
- The box height is applied between the zero marks of Slotter 1 and Slotter n in the sheet. The zero mark for slotter n-1 is offset by 1/n th of "SlotLength2" in the direction of the rear edge of the sheet.
- At inputs "SlotLength1" and "SlotLength2", the entire length of Slots 1 and 2 need to be applied and not the length of the partial slot which the slotter being calculated produces.
- At the "NumberOfSlotters" input, a definition is needed as to how many slotters are located in the multislotter group and work together on the two slots.
- At the "SlotterToCalculate" input, a definition is needed for which of the "NumberOfSlotters" slotters the cam is going to be calculated for. The slotters are numbered in the order in which they make contact with the sheet (Slotter 1 makes contact with the sheet firstly, Slotter2 secondly, ...).
- Since Slotter1 cuts a window in the material for Slot1, it has to run synchronously with the material velocity for the entire length of its knife. All other slotters only run synchronously for their respective partial slot lengths. This is why Slotters 2-n are able to respond dynamically more favorably than Slotter 1.

Functional description

With the target to maintain the circumferential velocity of the knife roller synchronous to the speed of the material web in a synchronous range determined by the cutting process (specified by "SlotLength1" and "SlotLength2"), the function block calculates one time a cam table with 1,024 data point elements on the basis of the input variables after processing has been enabled using "Execute".

The explicit significance of the input variables is shown in the depiction below.

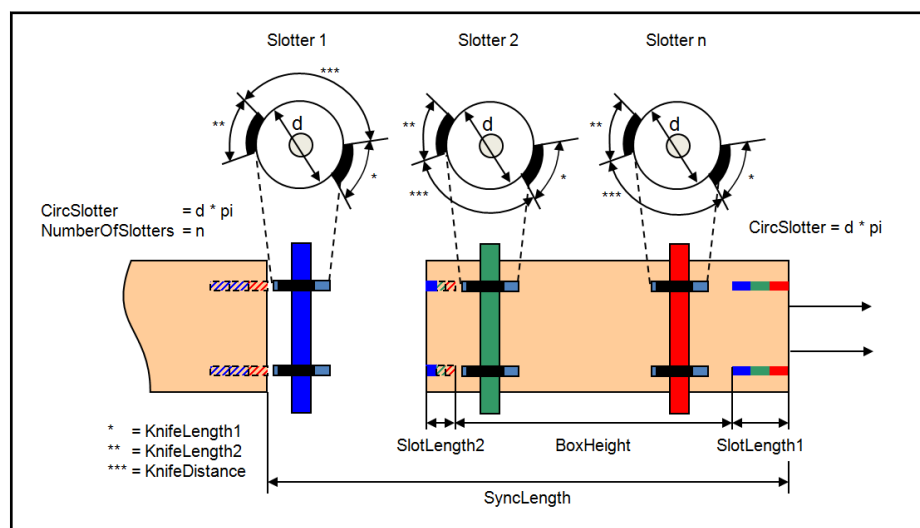
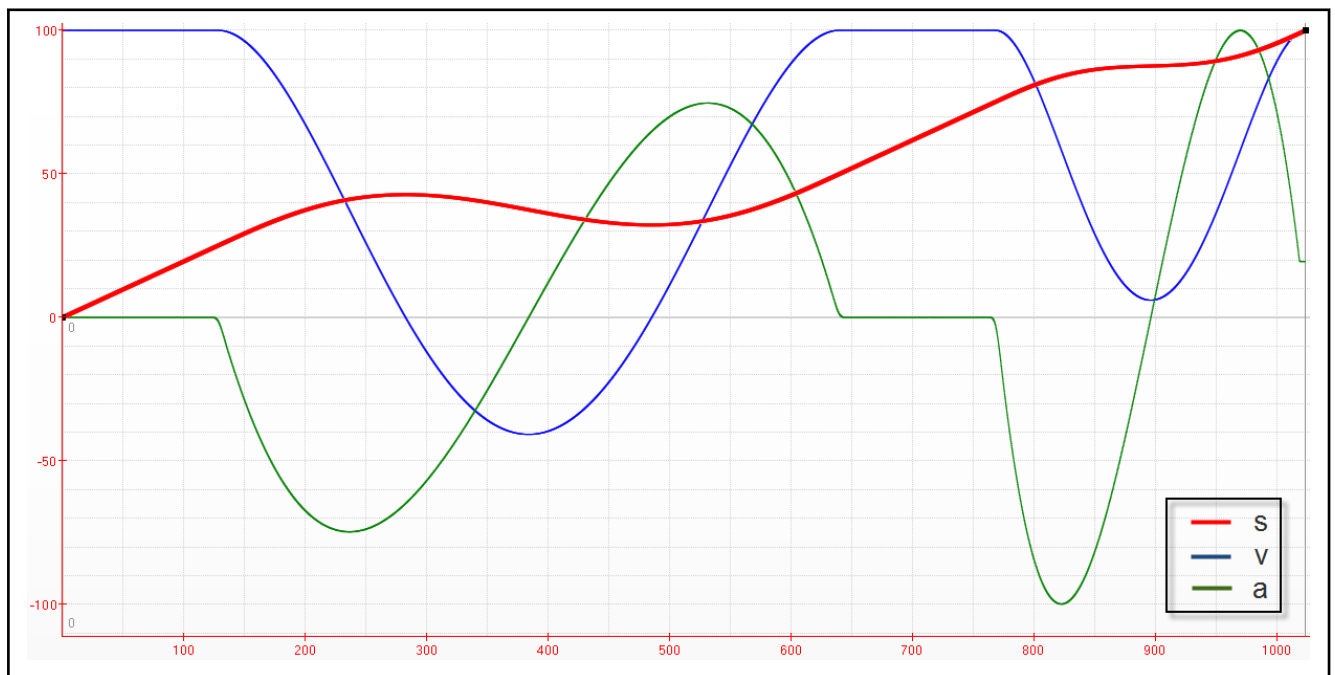


Fig. 10-44: Significance of the input variables for the MB_SlotterCalcType02 function block

In accordance with the two synchronous ranges specified, two compensation motions calculated by the function block have to take place. These compensation motions are characterized by a continuity up to the acceleration.

Different function variants are possible using function block MB_SlotterCalc-Type02 through the use of various preassignments for the inputs:

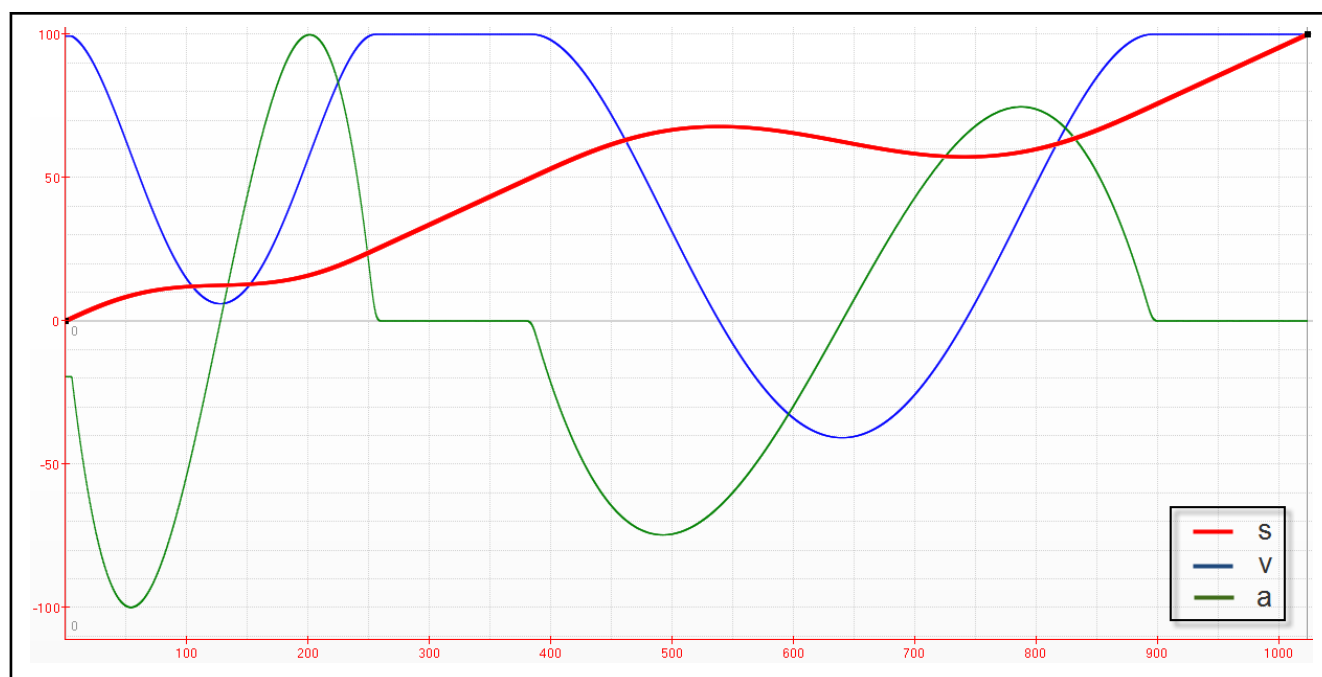
The first variant [fig. 10-45 "Example of motion profile 1" on page 681](#) is suitable for using a slotter to cut formats whose cutting circle "CircSlotter" is smaller than the mechanical processing length of the machine "SyncLength". The effective cutter lengths are adjusted to the material transportation velocity in the tool gearing. The remaining disc circumference of the slotter without cutting knife is defined as the area for the compensation motions.



Example 1: Zero mark in front of the knife, compensation motion at the end (SyncLength = 1024; CircSlotter = 512; SlotLength1 = SlotLength2 = 128; BoxHeight = 256; KnifeLength1 = KnifeLength2 = KnifeDistance = 128; SkipMode = 0; CamProf = 0)

Fig. 10-45: Example of motion profile 1

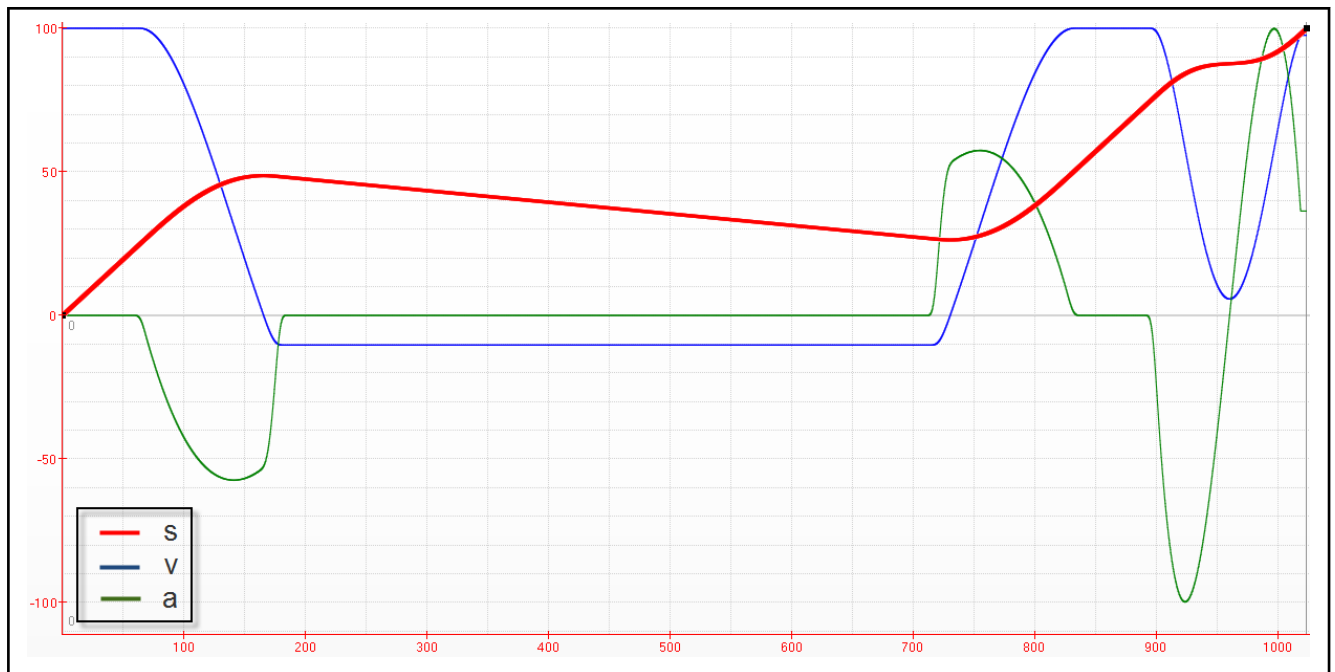
In all single slotter variants "NumberOfSlotters" = 1, the zero mark for the slotter roller can be defined as being at the end of the first knife or at the start of the second knife by parameterizing the "CamProf" input. In the following variant [fig. 10-46 "Example of motion profile 2" on page 682](#), the first variant [fig. 10-45 "Example of motion profile 1" on page 681](#) was taken up again representatively, and a compensation motion was selected at the start and the synchronous motion at the end using the "CamProf" input.



Example 2: Zero mark after the knife, compensation motion at the start (SyncLength = 1024; CircSlotter = 512; SlotLength1 = SlotLength2 = 128; BoxHeight = 256; KnifeLength1 = KnifeLength2 = KnifeDistance = 128; SkipMode = 0; CamProf = 2)

Fig. 10-46: Example of motion profile 2

The third variant [fig. 10-47 "Example of motion profile 3" on page 683](#) is suitable for the processing of excessively long formats, i.e. formats larger than the maximum synchronous format length. The slotting process is stretched to several machine cycles, whereupon the cutting areas of the knives remain synchronous to the surface velocity ("SkipMode" = n). In the course of n+1 machine cycles, the slotter rotates with one mechanical rotation. While the areas with the effective cutter lengths stay synchronous to the material, a compensation motion is carried out in the remaining motion until the n+1 cycle is completed. In addition, the mode of operation of the pendulum limit can be recognized below. Specifying the "PendulumFactor" = 90 stipulates that the reverse rotation is limited in the compensation range to $37.5\% + 12.5\% \cdot 0.9 = 48.75\%$.



Example 3: Zero mark in front of the knife, compensation motion at the end (SyncLength = 1024; CircSlotter = 512; SlotLength1 = SlotLength2 = 128; BoxHeight = 256; KnifeLength1 = KnifeLength2 = KnifeDistance = 128; PendulumFactor = 90; SkipMode = 1; CamProf = 0)

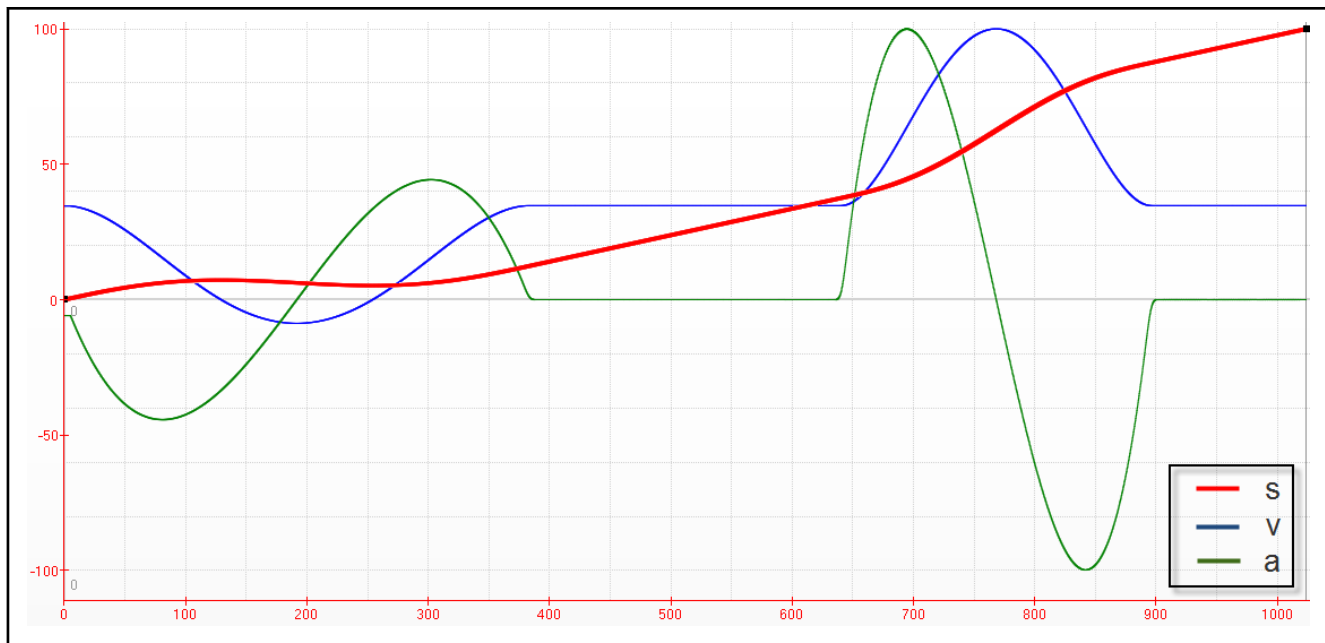
Fig. 10-47: Example of motion profile 3

In the fourth variant, cams are shown generating a long slot at the start of the sheet and a long slot at the end of the sheet (multislotters).



By contrast to the definition option for a zero mark when using single slotters, for multislotters, the zero marks for the individual slotters are defined according to [fig. 10-32 "Definitions and specifications for multislotters" on page 664](#). Slotter 1 (NumberOfSlotter = 1) always has its zero mark at the end of the knife (CamProf = 2, 3) in the direction of rotation, and Slotters 2 to n (NumberOfSlotter = 2,...,n) at the start (CamProf = 0, 1).

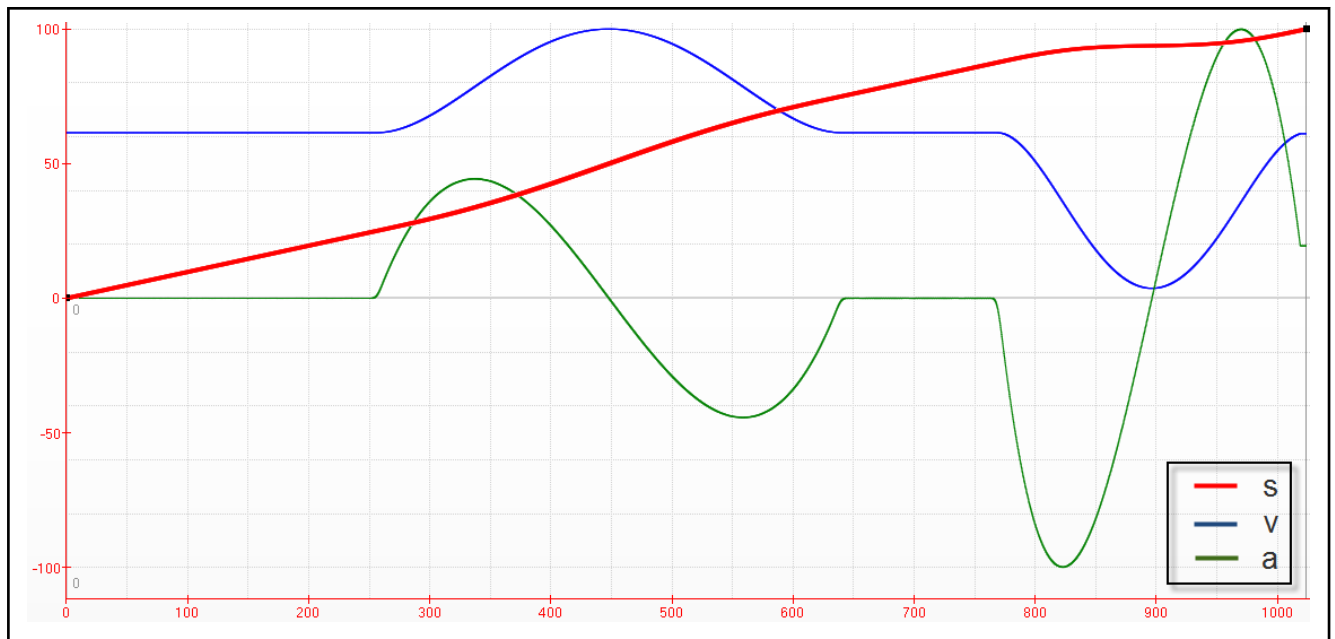
The cam for the first slotter is characterized in accordance with the position of the zero mark by a cam that starts with a compensation motion and ends with a synchronous range.



Example 4: First slotter in the multi-slotted group (SyncLength = 1024; CircSlotter = 1024; KnifeLength1 = 128; KnifeLength2 = 256; KnifeDistance = 128; BoxHeight = 128; SlotLength1 = 256; SlotLength2 = 512; NumberOfSlotters = 2, SlotterToCalculate = 1; SkipMode = 0; CamProf = 2)

Fig. 10-48: Example of motion profile 4

The cam for the second slotter is characterized in accordance with the position of the zero mark by a cam that starts with a synchronous range and ends with a compensation motion.



Example 5: Second slotter in the multislotter group (SyncLength = 1024; CircSlotter = 1024; KnifeLength1 = 128; KnifeLength2 = 256; KnifeDistance = 128; BoxHeight = 128; SlotLength1 = 256; SlotLength2 = 512; NumberOfSlotters = 2, SlotterToCalculate = 2; SkipMode = 0; CamProf = 0)

Fig. 10-49: Example of motion profile 5

The cam table to be calculated is saved in the "CamTable" structure in case of an error-free processing of the function block, whereupon the validity of the cam elements is signaled by the output "Done".

If an error occurs while processing the function block, this is signaled by the "Error" output. The elements of the "CamTable" structure are not updated in case of error.

Error handling

This function block uses the error table **F_RELATED_TABLE**, 16#170. The function block can generate the following error messages in Additional1/Additional2.

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR, 16#0005	16#0006	16#0000	Invalid function block state
CALCULATION_ERROR, 16#0007	16#000C	16#0000	Division by Zero
INPUT_INVALID_ERROR, 16#0001	16#2201	16#0001	Address of the "CamTable" array is ZERO
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0001	"SyncLength" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0002	"CircSlotter" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0004	"BoxHeight" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0005	"SlotLength1" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0006	"SlotLength2" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0007	"PendulumFactor" is smaller than the minimum or higher than the maximum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0008	"SkipMode" is smaller than the minimum or higher than the maximum value

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0009	"CamProf" is smaller than the minimum or higher than the maximum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0010	"NumberOfSlotters" is smaller than the minimum or higher than the maximum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0011	"SlotterToCalculate" is smaller than the minimum value or higher than "NumberOfSlotters"
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0012	"CamTable.NumberOfElements" is not equal to 1024
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0013	"KnifeLength1" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0014	"KnifeDistance" is smaller than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0015	"KnifeLength2" is smaller than the minimum value
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0004	"SlotLength1" + "BoxHeight" + "SlotLength2" is greater than "SyncLength" * ("SkipMode" + 1)
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0005	"CamProf" value is not valid
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0006	"KnifeLength1" is greater than "CircSlotter"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0007	"KnifeLength2" is greater than "CircSlotter"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0008	"SlotLength1" is greater than "KnifeLength1" * "NumberOfSlotters"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0009	"SlotLength2" is greater than "KnifeLength2" * "NumberOfSlotters"
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0010	"KnifeLength1" + "KnifeDistance" + "KnifeLength2" is greater than "CircSlotter"

Tab. 10-24: Error codes of the FB MB_SlotterCalcType02

10.11 MB_CAMPROF_SLOTTER

The enumeration "MB_CAMPROF_SLOTTER" selects between different cam profiles and zero marks.

Element	Value	Description
CAMPROF_STANDARD_SEGMENTABLE_ZERO_KNIFE	0	Cam profile without gear reduction. Synchronous area at the beginning, compensation motion at the end
CAMPROF_ADDITIVE_SEGMENTABLE_ZERO_KNIFE	1	Superimposed profile with gear reduction. Synchronous area at the beginning, compensation motion at the end
CAMPROF_STANDARD_SEGMENTABLE_KNIFE_ZERO	2	Cam profile without gear reduction. Synchronous area at the end, compensation motion at the beginning
CAMPROF_ADDITIVE_SEGMENTABLE_KNIFE_ZERO	3	Superimposed profile with gear reduction. Synchronous area at the end, compensation motion at the beginning

Tab. 10-25: Elements of the enumeration type MB_CAMPROF_SLOTTER

11 RIL_ParameterChannel.library

11.1 Introduction and Overview

Using its function block [IL_ParameterChannel](#), page 687, the RIL_ParameterChannel library allows acyclic communication via the cyclic channel of a field bus connection.



The library requires the PLC basic package for the control.

11.2 IL_ParameterChannel

Brief description

The function block allows an acyclic communication via the cyclic data channel. The data is exchanged via this function block on the control as well as via a corresponding function block in the field bus master.

Assignment: Target system/library

Target system	Libraries
MLC 14VRS PLC basic package	RIL_ParameterChannel

Tab. 11-1: Reference table for the IL_ParameterChannel function block

Interface description

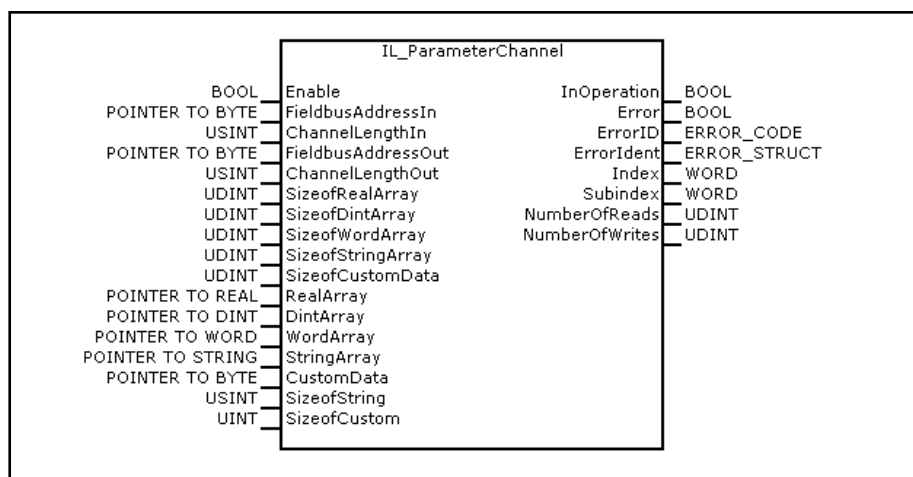


Fig. 11-1: IL_ParameterChannel function block

	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Processing of the function block enabled (once, edge-controlled)
	FieldbusAddressIn	POINTER TO BYTE	Address of the input range for the cyclic data traffic
	ChannelLengthIn	USINT	Length of the cyclic inputs for the parameter channel in bytes.
	FieldbusAddressOut	POINTER TO BYTE	Address of the output range for the cyclic data traffic
	ChannelLengthOut	USINT	Length of the cyclic outputs for the parameter channel in bytes.
	SizeofRealArray	UDINT	Number of REAL data accessible via the RealArray address in bytes.
	SizeofDintArray	UDINT	Number of DINT data accessible via the DintArray address in bytes.

	Name	Data type	Description
	SizeofWordArray	UDINT	Number of WORD data accessible via the WordArray address in bytes.
	SizeofStringArray	UDINT	Number of STRING data accessible via the StringArray address in bytes.
	SizeofCustomData	UDINT	Number of the complete Custom data accessible via the Custom-Data address in bytes.
	RealArray	POINTER TO REAL	A pointer to an array of REAL values. The array can be provided with 4,845 elements
	DintArray	POINTER TO DINT	A pointer to an array of DINT values. The array can be provided with 4,845 elements
	WordArray	POINTER TO WORD	A pointer to an array of WORD values. The array can have 2,295 elements
	StringArray	POINTER TO STRING	A pointer to an array of STRING values. The array can have 2,040 elements
	CustomData	POINTER TO BYTE	A pointer to an array of BYTE values. The array can have data of up to 255 * 8 KB
	SizeofString	USINT	STRING size in the StringArray. Caution: This is the size given in the definition of the string (without end ID).
	SizeofCustom	UINT	Size of a data array (1 – 8,192 bytes)
VAR_OUTPUT	InOperation	BOOL	The function block operates correctly. The output data is valid
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in the event of an error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Error handling, page 692
	Index	WORD	Last successfully used index
	Subindex	WORD	Last successfully used subindex
	NumberOfReads	UDINT	Number of successful read accesses by the master. The value can exceed (from 65535 to 0).
	NumberOfWrites	UDINT	Number of successful write accesses by the master. The value can exceed (from 65535 to 0).

Tab. 11-2: Interface description of the IL_ParameterChannel function block

Min./max. default values and applying the inputs

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
FieldbusAddressIn	POINTER TO BYTE	16#00	16#FFFFFFFF	16#00	Rising edge at "Enable"
ChannelLengthIn	USINT	6	18	12	Rising edge at "Enable"

Name	Type	Min. value	Max. value	Default value	Effective
FieldbusAddressOut	POINTER TO BYTE	16#00	16#FFFFFFFF	16#00	Rising edge at "Enable"
ChannelLengthOut	USINT	6	18	12	Rising edge at "Enable"
SizeofRealArray	UDINT	0	19380 (4845 * 4)	0	Rising edge at "Enable"
SizeofDintArray	UDINT	0	19380 (4845 * 4)	0	Rising edge at "Enable"
SizeofWordArray	UDINT	0	4590 (2295 * 2)	0	Rising edge at "Enable"
SizeofStringArray	UDINT	0	522240 (2040 * 256)	0	Rising edge at "Enable"
SizeofCustomData	UDINT	0	2088960 (255 * 8192)	0	Rising edge at "Enable"
RealArray	POINTER TO REAL	16#00	16#FFFFFFFF	16#00	Rising edge at "Enable"
DintArray	POINTER TO DINT	16#00	16#FFFFFFFF	16#00	Rising edge at "Enable"
WordArray	POINTER TO WORD	16#00	16#FFFFFFFF	16#00	Rising edge at "Enable"
StringArray	POINTER TO STRING	16#00	16#FFFFFFFF	16#00	Rising edge at "Enable"
CustomData	POINTER TO BYTE	16#00	16#FFFFFFFF	16#00	Rising edge at "Enable"
SizeofString	USINT	0	255	80	Rising edge at "Enable"
SizeofCustom	UINT	0	8192	128	Rising edge at "Enable"

Tab. 11-3: Min./max./default values of the IL_ParameterChannel function block

Functional description

The function block allows an acyclic communication via a cyclic channel. Therefore, the range from the cyclic data channel is used for the communication. The inputs "FieldbusAddressIn" and "ChannelLengthIn" transfer data from the master to the slave. The inputs "FieldbusAddressOut" and "ChannelLengthOut" are used for the direction from the slave to the master. The terms "out" and "in" refer to the direction seen from the slave. The data ranges can be at any position in the cyclic data and can also differ in length. The minimum length of the parameter channel is 6 bytes and the maximum length in both directions is 18 bytes.

PLC data can only be exchanged via data arrays. The different data types have to be considered.

Transferable data types

- REAL (up to 4845 data arrays)
- DINT (up to 4845 data arrays)
- WORD (up to 2295 data arrays)
- STRING (up to 2040 data arrays)
- Custom (up to 255 data arrays)

All data areas are assigned as pointers to the function block. The size available has to be specified. If either the data size or the assigned pointer is 0, the function block interprets it as a non-existing data range and outputs an error to these data requests. The data size has to be in bytes, e.g. the `SIZEOF()` operator can be used.

Apart from the array size, the string size has to be specified for strings at the "SizeofString" input. This is required since a string can be defined in different sizes. The size of the string has to be uniform in the array and may not consist of more than 255 characters.



The size of the string is the size given when defining the string. IndraLogic attaches one more byte for the end ID. For the definition

```
strExample: STRING(20);
```

21 bytes are assigned in the PLC, since the end ID still has to be added after the 20 characters.

The `IL_ParameterChannel` function block considers this automatically when checking the sizes. Thus, "`SIZEOF(strExample)`" had to be specified as value for "SizeofString" in the previous example.

The `CustomData` can be used to consistently transfer a higher number of data. The `CustomData` is like a data structure that is to be completely transferred. Up to 255 of these structures can exist. To use this functionality, a pointer has to be assigned ("`CustomData`"). The size of the individual structure has to be known as well. This is executed at the "SizeofCustom" input. Finally, the complete size is required ("`SizeofCustomData`").

Since there is no information on the data content in the function block, the data is copied byte by byte. The respective application has to ensure that the data format is consistent on slave and master.



Since in `CustomData`, a data set always has to be copied completely, the time required for copying takes on significant importance in the runtime of a function block.

Runtime for copying and processing the protocol

- 256 bytes, approx. 12µs
- 4 Kbytes, approx. 20µs
- 8 Kbytes, approx. 40µs on an L65

Please take this into consideration for the configuration of the task.



If functions are used to check the validity of pointers and array limits (e.g. `RIL_CheckRtv.library`), the runtime of the function block increases drastically.

In the case of 4 Kbyte `CustomData`, up to 30ms will then be required instead of 20µs!



The applied pointers and lengths are not allowed to change as long as the "Enable" input is TRUE.

This needs to be taken into account particularly in the event of an online change if the sizes of the applied arrays are changed.

If the pointers or lengths change during the transmission, then an error will be generated.

The outputs "NumberOfReads" and "NumberOfWrites" identify an action in the slave performed by the master and to react accordingly. Thus, the respective values of "NumberOfReads" and "NumberOfWrites" can be saved on the slave in the PLC. If there is a change, an action can be performed. Thus, the value just written by the master can be processed after a change of "NumberOfWrites" via the specified "index" and "subindex" for example.

The outputs are only updated if reading or writing was successful. If an error occurred, the counters are not increased. The counters are of type UDINT. Their value range is between 0 and 65535. If the counter is at the maximum value of 65535 and it has to be incremented again, there is an overflow in the counter and it starts again at 0.

Boundary conditions and restrictions:

- The function block may only be instantiated once
- Only the parameter set 0 can be read for sercos parameters
- sercos parameters can only be read as 16 bit "Ident Number" (IDN). "Extended Ident Number" (EIDN) cannot be read.
- In the sercos parameters, only the sercos element 7 (data) can be read

The data can be queried from the master via the index or subindex. The following list specifies the respective values for the index/subindex.

Starting index (16 bits)	End index (16 bits)	Subindex (8 bits)	Calculation rule
16#0000	16#1FFF		Reserved
16#2000	16#2FFF	Drive # (1-64)	Index = 16#2000 + S-parameter (only parameter set 0 is possible, no EIDN)
16#3000	16#3FFF	Drive # (1-64)	Index = 16#3000 + P-parameter (only parameter set 0 is possible, no EIDN)
16#4000	16#4FFF	Axis # (1-64)	Index = 16#4000 + A-parameter (only parameter set 0 is possible, no EIDN)
16#5000	16#5AEC		Free
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]; sub-index = (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]; sub-index = (STRING array element) – [(index - 16#5AF8) * 255]; (element 1 - 2040)
16#5B00	16#5CEB	16#00 – 16#FF	Free

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	Free
16#6000	16#6FFF	16#00	Index = 16#6000 + C-parameter (only parameter set 0 is possible, no EIDN)
16#7000	16#7FFF	Kinematics # (1-16)	Index = 16#7000 + K-parameter (only parameter set 0 is possible, no EIDN)
16#8000	16#8FFF	Probe # (1-99)	Index = 16#8000 + M-parameter (only parameter set 0 is possible, no EIDN)
16#9000	16#9FFF	PLS # (1-4)	Index = 16#9000 + N-parameter (only parameter set 0 is possible, no EIDN)
16#A000	16#AFFF	Osci # (1-16)	Index = 16#A000 + O-parameter (only parameter set 0 is possible, no EIDN)
16#B000	16#FFFF		Free

Tab. 11-4: Assigned address ranges for index/subindex

Error handling

There are two different error types for the error handling.

- Error when accessing data and error when transferring data
- Error in the configuration of the function block

Errors in the configuration (e.g. parameter channel too short/long) are displayed as errors at the function block itself and have to be analyzed by the PLC program.

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0001	"FieldbusAddressIn" is either not assigned or is provided with 0.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0002	"ChannelLengthIn" is either not assigned or is provided with an invalid value (valid values are in the range 6 <= "ChannelLengthIn" <= 18).
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0003	"FieldbusAddressOut" is either not assigned or is provided with 0.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0004	"ChannelLengthOut" is either not assigned or provided with an invalid value (valid values are in the range 6 <= "ChannelLengthOut" <= 18).
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0005	"SizeofRealArray" is provided with an invalid value (valid values are in the range 0 <= "SizeofRealArray" <= 19380).
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0006	"SizeofDintArray" is provided with an invalid value (valid values are in the range 0 <= "SizeofDintArray" <= 19380).
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0007	"SizeofWordArray" is provided with an invalid value (valid values are in the range 0 <= "SizeofWordArray" <= 4590).
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0008	"SizeofStringArray" is provided with an invalid value (valid values are in the range 0 <= "SizeofStringArray" <= 52240).

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0009	"SizeofCustomData" exceeds the value allowed (2088960).
INPUT_RANGE_ERROR (16#0001)	16#0002	16#000A	"SizeofRealArray" is provided with an invalid value. "SizeofRealArray" has to be divisible by 4 without remainder.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#000B	"SizeofRealArray" is provided with an invalid value. "SizeofDintArray" has to be divisible by 4 without remainder.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#000C	"SizeofWordArray" is provided with an invalid value. "SizeofWordArray" has to be divisible by 2 without remainder.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#000D	"SizeofStringArray" or "SizeofString" is provided with an invalid value. "SizeofStringArray" has to be divisible by "SizeofString" + 1 without remainder.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#000E	"SizeofCustomData" or "SizeofCustom" is provided with an invalid value. "SizeofCustomData" has to be divisible by "SizeofCustom" without remainder.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#000F	"SizeofCustom" is provided with an invalid value (valid values are in the range $0 \leq \text{"SizeofCustom"} \leq 8192$).
STATE_MACHINE_ERROR (16#0005)	16#0006	16#000x	An invalid state has been detected in the function block. The function block must be restarted.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0010	The pointer in "FieldbusAddressIn" or the length "ChannelLengthIn" has changed.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0011	The pointer in "FieldbusAddressOut" or the length "ChannelLengthOut" has changed.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0012	The pointer in "RealArray" or the length "SizeofRealArray" has changed.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0013	The pointer in "DintArray" or the length "SizeofDintArray" has changed.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0014	The pointer in "WordArray" or the length "SizeofWordArray" has changed.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0015	The pointer in "StringArray", the length "SizeofStringArray", or the length "SizeofString" has changed.
INPUT_RANGE_ERROR (16#0001)	16#0002	16#0016	The pointer in "CustomArray", the length "SizeofCustomArray", or the length "SizeofCustom" has changed.

Tab. 11-5: Error codes of FB IL_ParameterChannel

Errors while accessing data (such as read/write non-existent index) and transferring data (such as incorrect length specification) are not visible at the function block output. These are labeled as errors in the parameter channel and error values are specified. The master has to analyze the error values.

ErrorCode	Text
16#8500	The data volume to be transferred is too high for the internal cache
16#8B00	The format specification in the control word is incorrect (has to be 16#0C)
16#8C00	The given length is longer than the parameter channel
16#9500	Data was sent even though a read request was made

ErrorCode	Text
16#9600	Bit "C1" is not set in the control word
16#9700	The "last" bit was not sent even though it is a read request
16#1F00	Control-specific error. Further error codes (see error table of the control) follow in the next four words. The error code "Additional1" is in the words 2 and 3 and the error code "Additional2" is in the words 4 and 5.
16#F200	The given index is unknown
16#F300	The given subindex is unknown

Tab. 11-6: Error codes within the parameter channel that can be created by the IL_ParameterChannel function block.

11.3 IL_ParameterChannel - Protocol, Error Handling, and Commissioning

11.3.1 Description of the Protocol Used

Parameter Channel in the Cyclic Data Channel

The protocol used in the parameter channel only uses a control word to send data from the master to the slave.

The slave uses the status word to be able to send data to the master.

The master can either write data or read data. A data transfer can be only activated by the master. The control word and the status word are always a fixed part of the parameter channel as shown below:

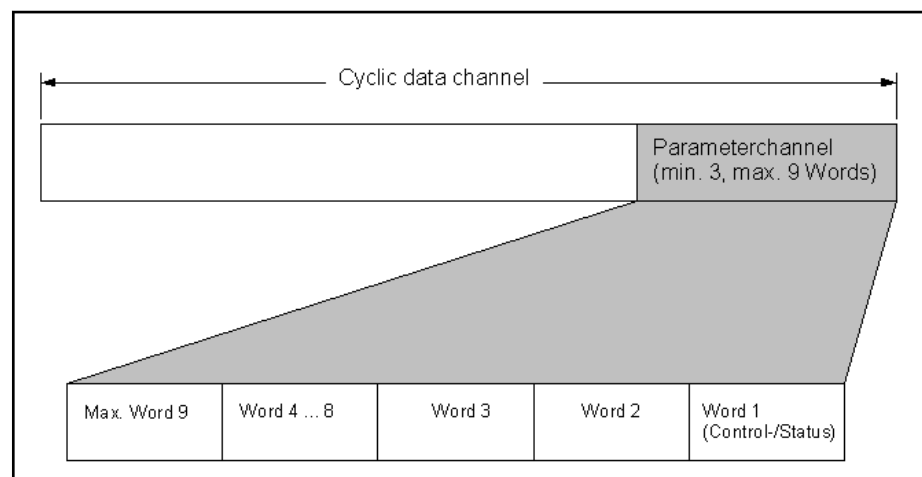


Fig. 11-2: Data content of the field bus in the parameter channel

Control Word/Status Word

The control word is assigned as follows:

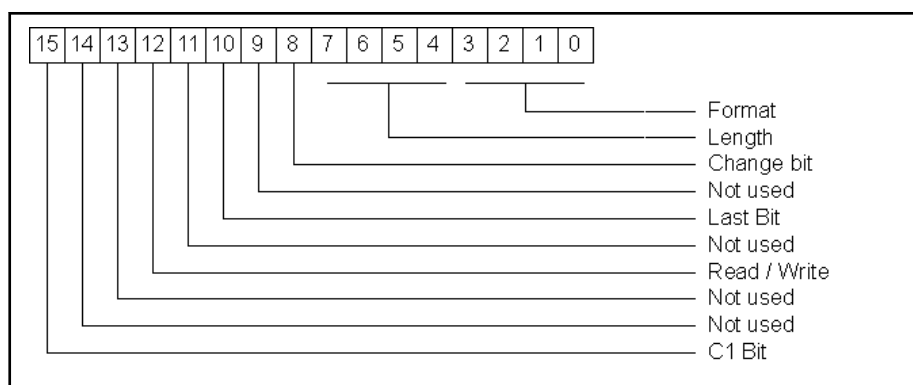


Fig. 11-3: Assignment of the control word in the parameter channel

The status word is assigned as follows:

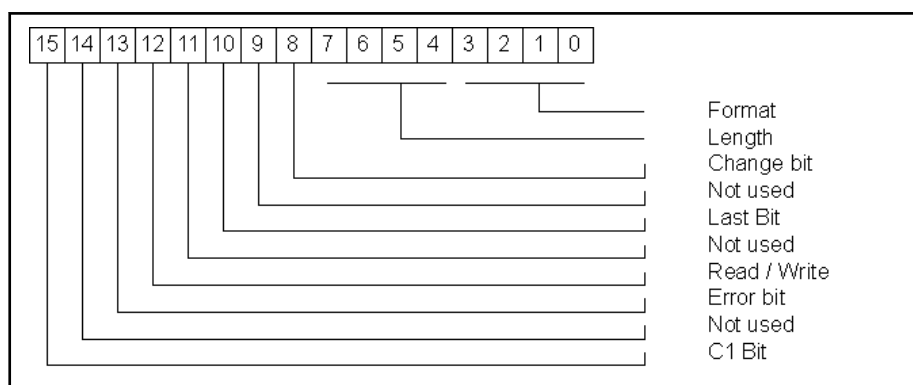


Fig. 11-4: Assignment of the status word in the parameter channel

The individual bits have the following meaning:

- Format:** These bits indicate the purpose and meaning of the following data words in the parameter channel. The value is always 1100b except if the communication is aborted or initialized.
- Length:** These four bits specify the length of the valid data in bytes excluding the control word. The data in the remaining part of the parameter channel is not defined.
- Toggle bit (T):** This bit is only then toggled if a new set of data is sent. It creates a handshake between the master and the slave. The master may only toggle this bit if the toggle bit in the status word has the same state as the toggle bit sent in the control word.
- Last bit (L):** The last bit is set if the last part of the data set is sent. It is used for fragmentation.
- Read/write bit (R/W):** "Read" corresponds to the bit value 1 which means that the master wants to read data. If it should be written, this bit has to be 0.
- C1:** Due to the compatibility with the legacy systems, this bit has to be set to 1.
- Error bit (E):** This bit signals when an error occurred in the slave. The error cause (error code) is specified with the following data.



Non-used bits are provided with a value of 0.

Index and Subindex

For the first request by the master, one word is provided for the index and one word for the subindex after the control word. Additionally, all available data can be addressed. The assignment of index/subindex to the control parameters is as follows:

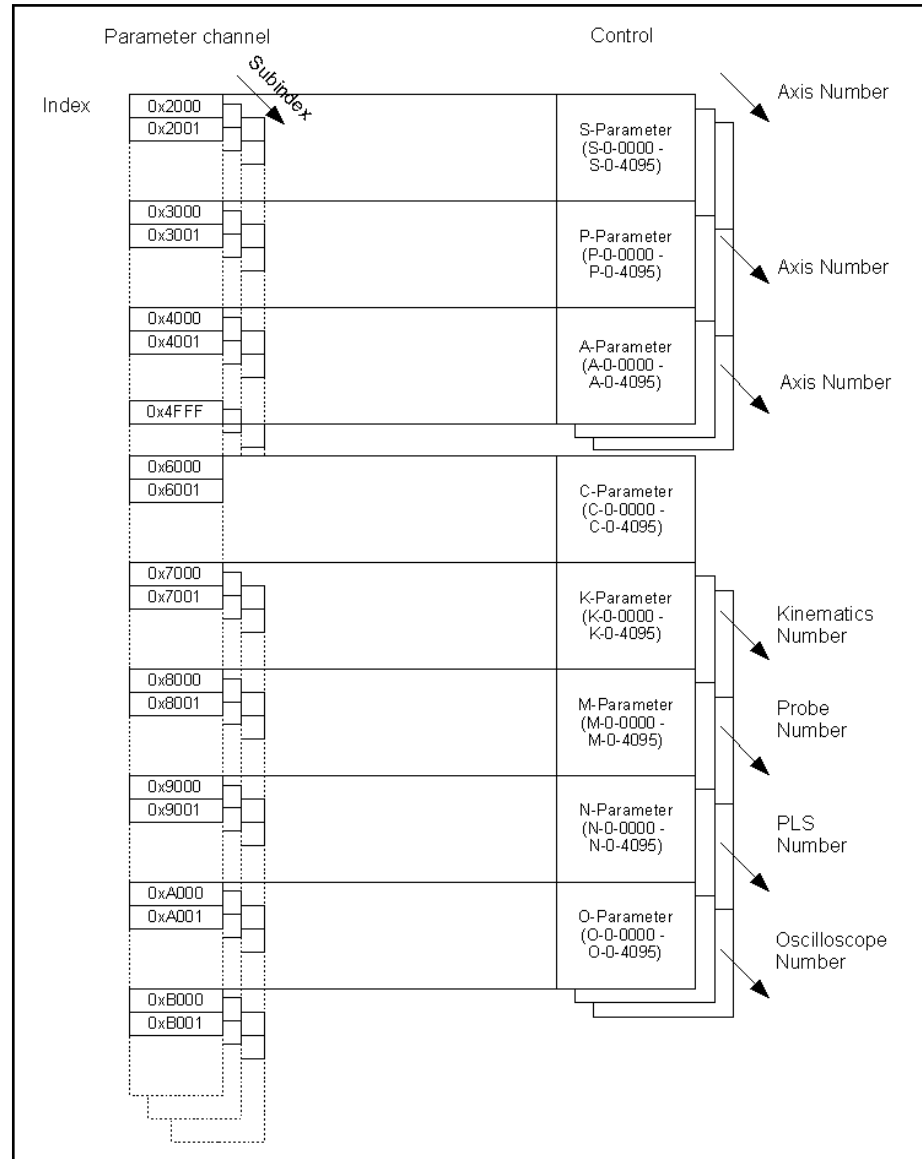


Fig. 11-5: Assignment of index/subindex to the parameters

The assignment is very simple. To get from a parameter to the desired index/subindex, only the parameter number has to be added to the start offset of the index. The subindex results from the respective number of the axis, the kinematics etc.

Example:

The axis parameter A-0-0100 of the axis 22 should be read.

In this case, $16\#4000 + 100 = 16\#4064$ has to be selected for the index and the axis number 22 for the subindex.

The assignment of index/subindex to the PLC variables is shown below. It is important that the subindex 0 is not used. Thus, it is not shown below.

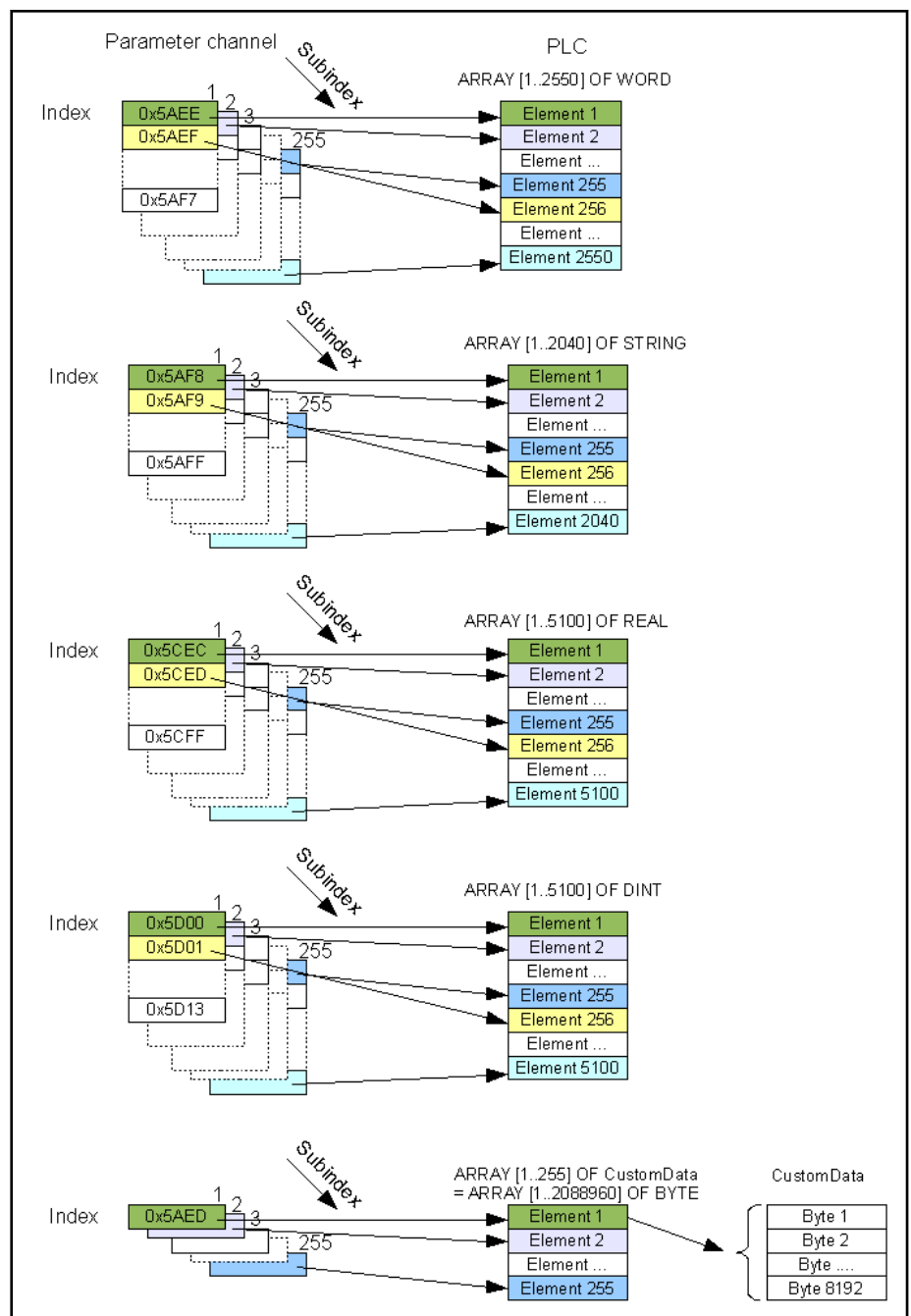


Fig. 11-6: Assignment of index/subindex to PLC variables

The mapping rules are more comprehensive. To get from a desired element number to the index/subindex, the following conversion formulae are used for the data types STRING, REAL, DINT and WORD:

Index = start index + [(element number - 1) / 255] (The division is integer which means the result is calculated without decimal places)

Subindex = element number - [(index - start index) * 255]

Example:

The 257th element of the REAL array should be read.

In this case, the start index is 16#5CEC and the element number is 257.

The following results for the index:

$\text{Index} = 16\#5\text{CEC} + (257 - 1) / 255 = 16\#5\text{CEC} + (256 / 255) = 16\#5\text{CEC} + 1 = 16\#5\text{CED}$

The following results for the subindex:

$\text{Subindex} = 257 - [(16\#5\text{CED} - 16\#5\text{CEC}) * 255] = 257 - [1 * 255] = 2$

To convert the index/subindex to the element number, the following formula applies:

Element number = [(index - starting index) * 255] + subindex

Example:

Which element is requested by index 16#5D02 and by subindex 3?

The index is in the range from 16#5D00 to 16#5D13 and thus, it is an element in the DINT array. The element number is:

$\text{Element number} = [(16\#5\text{D02} - 16\#5\text{D00}) * 255] + 3 = 2 * 255 + 3 = 513$

The same mapping rules as above apply to the CUSTOM data. Since only one index is used, the mapping can be simplified respectively.

$\text{Index} = \text{start index} = 16\#5\text{AED}$

$\text{Subindex} = \text{element number}$

To allow the calculation in the PLC program in reverse order (determine the element number from the index and the subindex), the following formulae can be used:

Element number = (index - starting index) * 255 + subindex

Example:

The function block indicates on its output that 16#5AF9 is written on the index and 5 on the subindex. On which element was thus written?

16#5AF9 is in the range 16#5AF8 to 16#5AFF. Thus, it is a string.

The element number results from:

$\text{Element number} = (16\#5\text{AF9} - 16\#5\text{AF8}) * 255 + 5 = 16\#1 * 255 + 5 = 260$

It was written on the 260th element in the "STRING array".

Initializing the Communication

To initialize the communication or to cancel a pending query, the master can send a special request. The request is provided with the following content in the control word: 2#1000 000x 0000 1111. The value of the toggle bit (T) is not important.

If the master sends this control word, the slaves aborts all the processing and goes into a state in which it can process a new request. It responds in the status word with the content 2#1000 000x 0000 1111.

Error Telegrams

If the slave detects an error, it sends a response telegram to the master in which the error bit (E) is set in the status word. There is an error code in the second word. This error code specifies the cause (also refer to [Error handling, page 692](#)).

The error code 16#1F00 is a special code. It specifies the error that occurred in the control. In this case, the words 3 and 4 contain an error code of the control that can be found in the documentation of the control. Additional information on the error can be found in the words 5 and 6. If the parameter channel is shorter than 6 words, the telegram is fragmented.

Example:

Procedure of a read request

The following example is based on the assumptions below:

- The parameter channel is 8 bytes = 4 words long in both directions.
- The toggle bit is 0 at the beginning
- A PLC STRING variable (element 1) should be requested. Thus, the index 16#5AF8 and the subindex 1 have to be selected
- In this case, the slave responds with only one telegram. If the response is longer, the "last bit" would not be set in the status bit and the master would have to request telegrams until the "last bit" is set in the status word

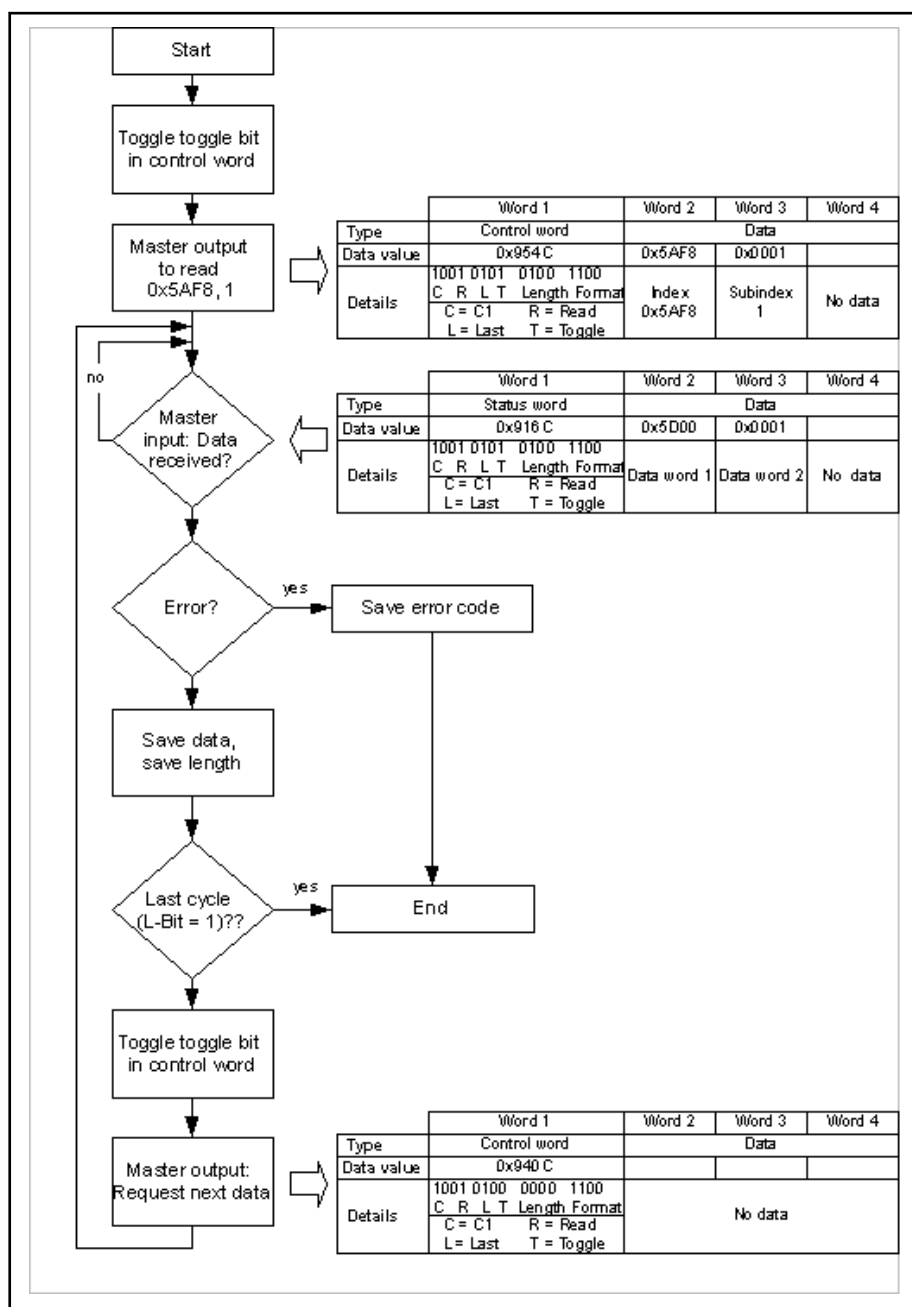


Fig. 11-7: Example on how to read a PLC STRING variable

Example:

Procedure of a write request

The following example is based on the assumptions below:

- The parameter channel is 8 bytes = 4 words long in both directions.
- The toggle bit is 0 at the beginning
- The axis parameter "A-0-0002, Axis name" of the axis 22 should be written. Thus, the index 16#4002 and the subindex 22 have to be selected

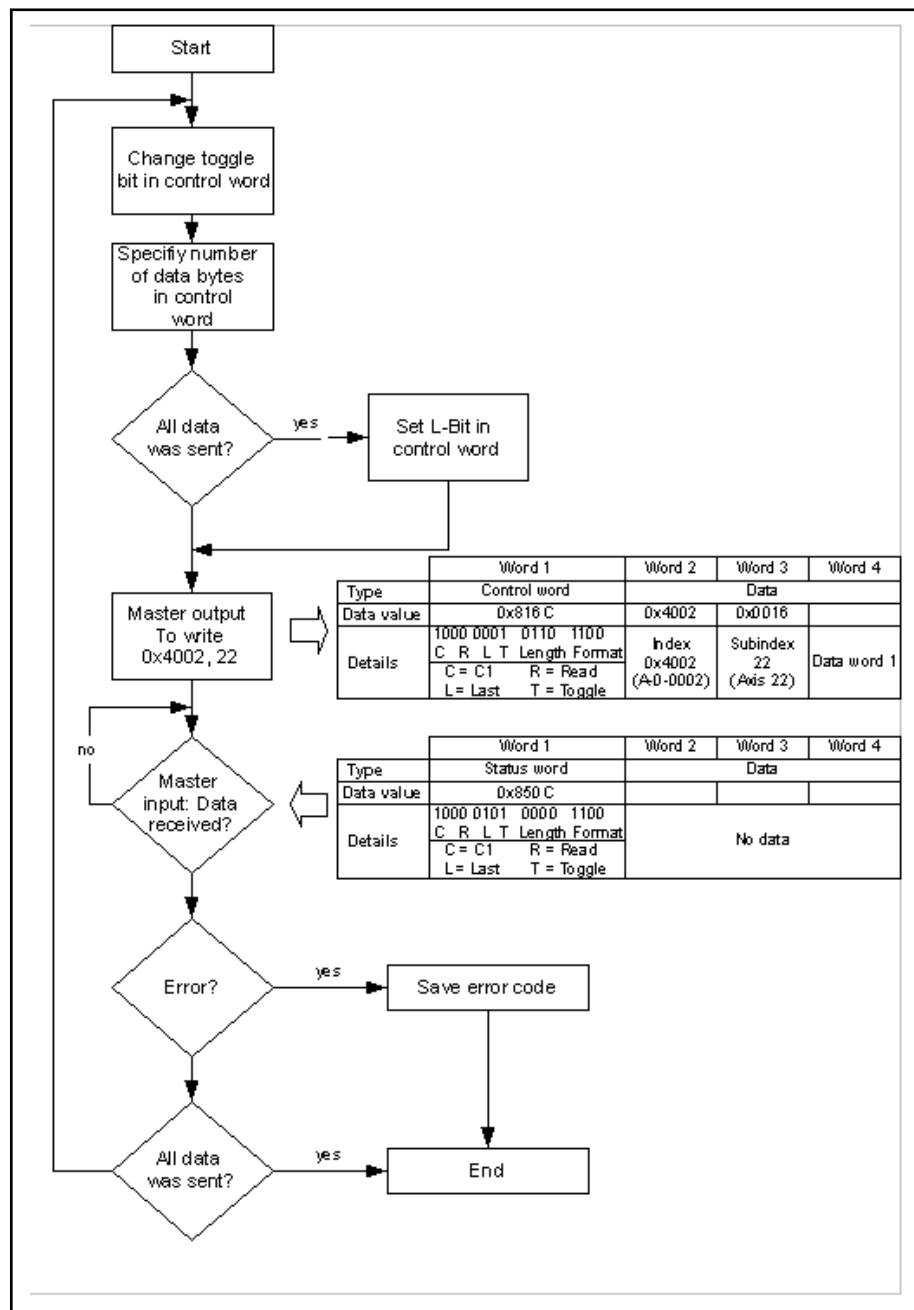


Fig. 11-8: Example on how to write an axis parameter

11.3.2 Commissioning of the Function Block

Before using the function block, establish a field bus connection. The example of the PROFIBUS below shows how it is configured for operation.

Individual modules are configured for the PROFIBUS. It is recommended for the parameter channel to use own modules to achieve a structuring. Thus, there are modules of 8 and 16 bytes (parameter channel can obtain a length from 6 to 18 bytes). Alternatively, only parts of a module can also be used. This has to be considered when assigning the parameter channel function block. It is important that the inserted modules are modules with byte order. The following figure defines one module each with 16 byte inputs and outputs ("ParameterChannel_In" and "ParameterChannel_Out") for the parameter channel.

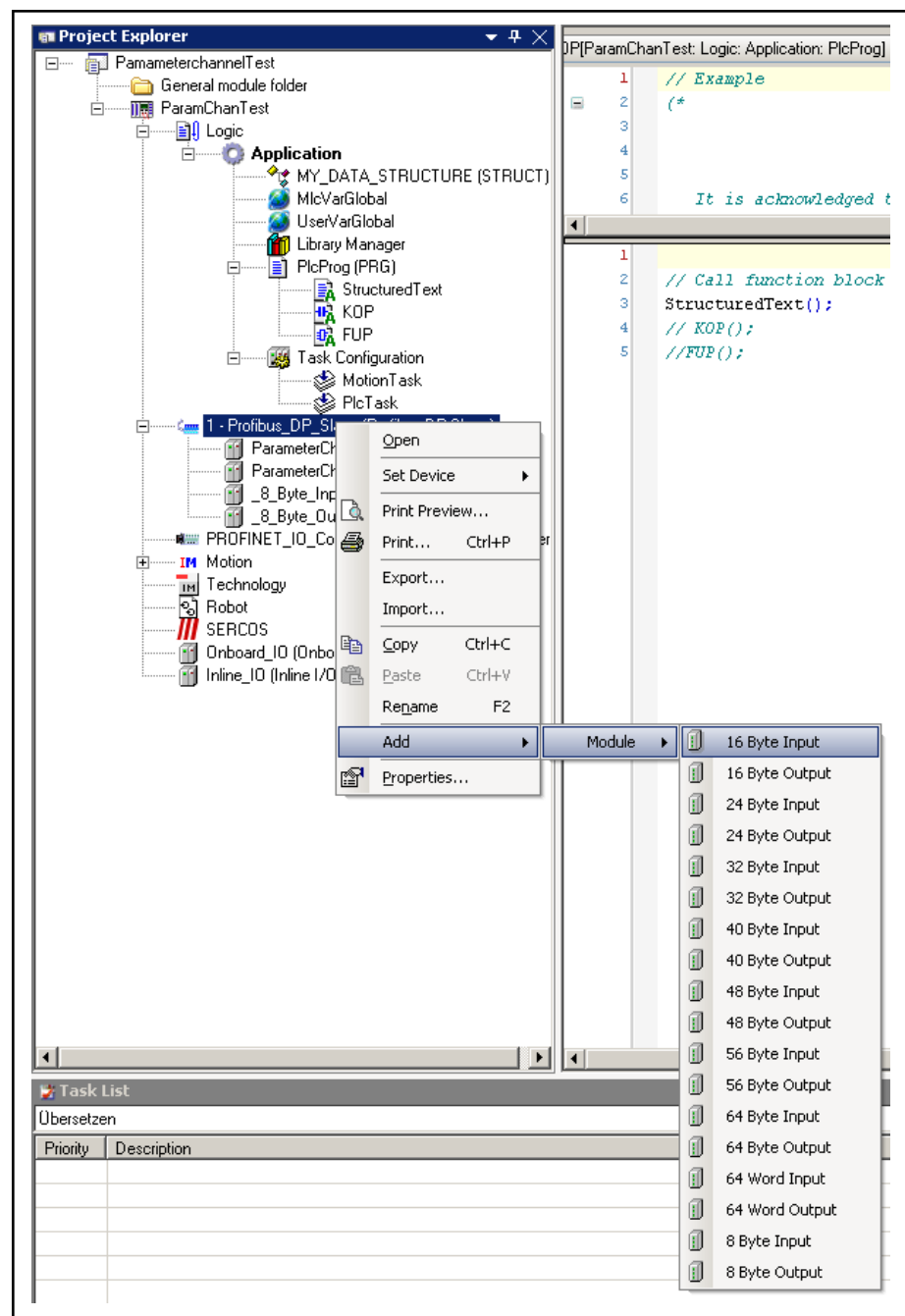


Fig. 11-9: PROFIBUS module used as parameter channel

To operate the function block, the addresses for the field bus and the respective length have to be specified. There are several possibilities. A symbolic name can be given or the address of the I/O range can be used.

In the following figure, the name "myFieldbusOut " was specified for the output data of the field bus module "ParameterChannel_Out". This is carried out by double-clicking on the desired module and in the opening window in the tab "DP Modules I/O Mapping". The tab might have different names for different field buses. Alternatively, the address can also be used (here "%QB1").

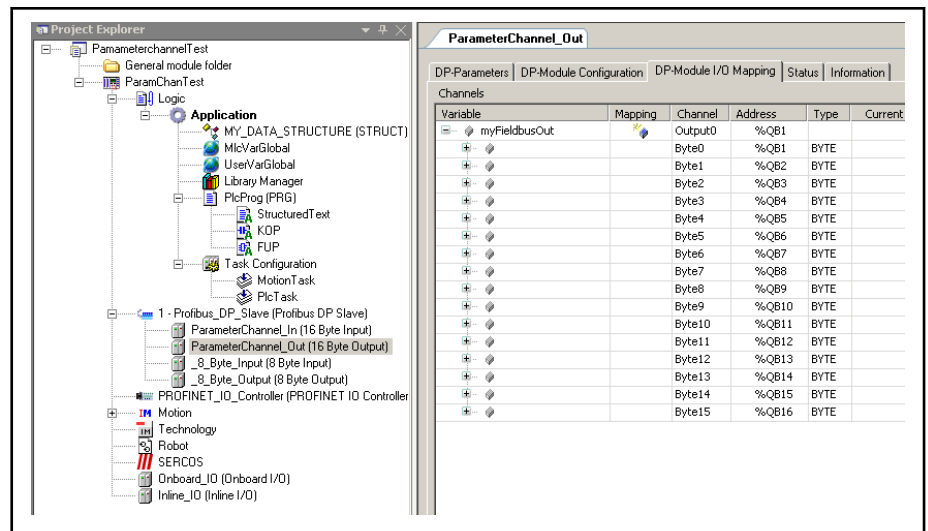


Fig. 11-10: Name and address of a Profibus module

To assign the IL_ParameterChannel function block, default inputs/outputs have to be assigned. In addition, inputs for PLC objects have to be assigned. Therefore, a pointer as well as the length of the data assigned are required. The STRING and CUSTOM data is something special since these can have user-specific lengths. Thus, the length of a string has also to be specified. In any case, the arrays always have to start with 1. Examples of the declaration of PLC variables are shown in the following source code. The maximum lengths are respectively specified for the arrays. The default inputs and outputs as well as the function block is declared.

Declaring an "IL_ParameterChannel" function block with data set

```

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

```

The function block is called as follows:

```

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

```

Fig. 11-11: Use of the IL_ParameterChannel function block in structured text

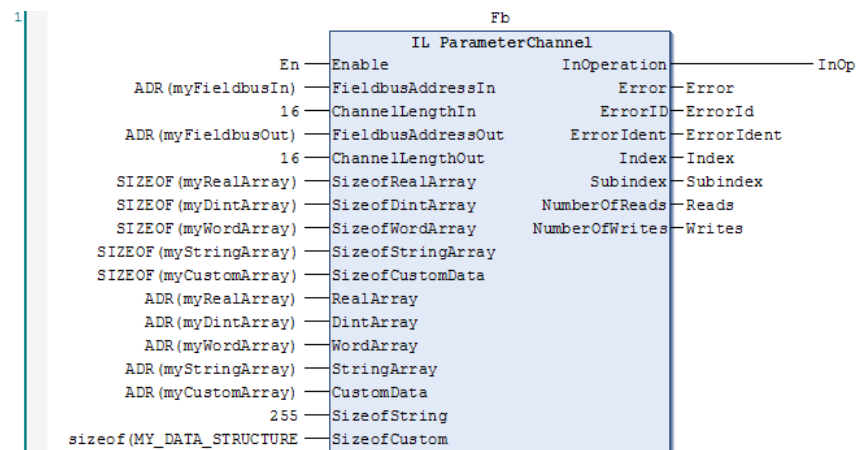


Fig. 11-12: Use of the IL_ParameterChannel function block in FBD

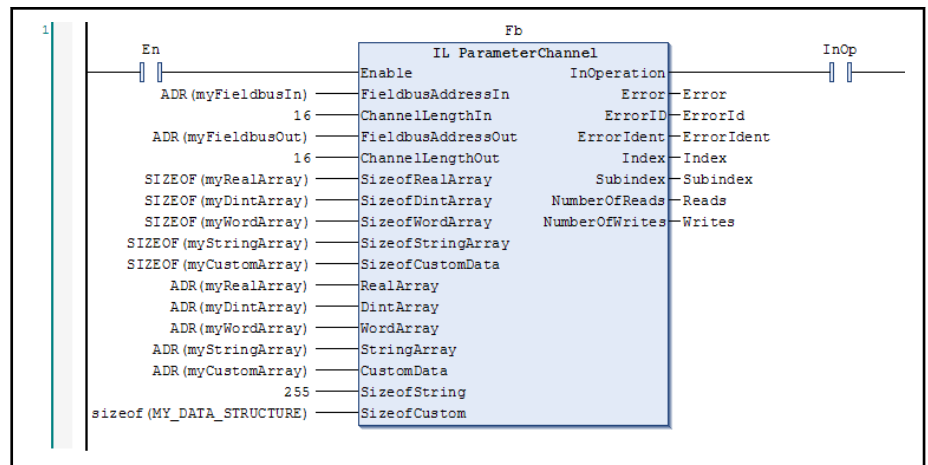


Fig. 11-13: Use of the IL_ParameterChannel function block in LD

In the examples above, the method with the symbolic address ("myFieldbus-Out") or the method with the direct addressing ("%IB1") was used for the addresses of the field bus data in the structured text.

11.3.3 Differences to the Legacy Systems

General Information

The function block is a substitute for the already existing system VisualMotion. Existing differences are now described.

On the Field Bus Slave

To operate the parameter channel, the field bus has to be setup on the slave so that the additional cyclic data is configured for the parameter channel. This was also the case for the legacy system.

Additionally, the IL_ParameterChannel function block as well as different arrays have to be created for the MLC/XLC.

The function block may only be instantiated once. It is started by setting the "Enable" input and acts with regard to the default outputs ("InOperation", "Error" etc.) like other MLC/XLC function blocks.

The arrays to be created represent the variables and the registers in the VisualMotion. In any case, the arrays always have to start at the index 1. The arrays of strings and CustomData are additionally available and did not have comparable variables in the VisualMotion.

MLC/XLC		VisualMotion	
Declaration	Data type	Name	Data type
myReals: ARRAY [1..4845] of REAL	REAL (32-bit)	Program floats (4845 pieces)	FLOAT (32-bit)
myDints: ARRAY [1..4845] of DINT	DINT (32-bit)	Program Ints (4845 pieces)	INT (32-bit)
myWord: ARRAY [1..2295] of WORD	WORD (16 bit)	Register (2295 pieces)	INT (16 bit)
myString: ARRAY [1..255] of STRING	STRING (myStringSize)	--	--

Tab. 11-7: Comparison of the PLC data MLC/XLC/VisualMotion

On the Field Bus Master

The existing function blocks on the side of the master can still be used. But adaptations in the function block assignment have to be executed.

Addressing via "Index" and "Subindex" was changed or extended.

The S-parameters and P-parameters are identical to the VisualMotion system.

Only the data contents change for registers, program floats and program ints:

- "Index" 16#5AEE – 16#5AF7 addressed the registers in VisualMotion. Now, the WORD array is accessed in the PLC
- "Index" 16#5CEC – 16#5CFF addressed the program floats in VisualMotion. Now, the REAL array is accessed in the PLC
- "Index" 16#5D00 – 16#5D13 addressed the program integers in VisualMotion. Now, the DINT array is accessed in the PLC

The data types and the calculation of the "index" and the "subindex" do not change.

The A-parameters were extended and remained compatible. Up to now, only the A-parameters till the number 2047 are possible. This function block can also be used to access A-parameters up to number 4096.

All the other data is empty data or additional data that has just been added.

Starting index (16 bits)	End index (16 bits)	Subindex (8 bits)	Calculation rule
16#0000	16#1FFF		Reserved
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		Free
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	Free
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	Free
16#6000	16#6FFF	16#00	Index = 16#6000 + C-parameter
16#7000	16#7FFF	Kinematics # (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		Free

Tab. 11-8: Assigned address ranges for index/subindex

The error codes have (partially) been changed in the parameter channel. The differences are shown in the table below:

ErrorCode MLC/XLC	ErrorCode VisualMotion	Description
16#8500	16#8500	Data too long (VisualMotion: >128 bytes; MLC > 64kBytes)
--	16#8800	Error not available on the MLC/XLC
16#8B00	16#8B00	Identical [the format specification in the control word is wrong (has to be 16#0C)]
--	16#8D00	Error case is already checked on the MLC/XLC while starting the function block
16#8C00	16#8C00	Identical (the given length is longer than the parameter channel itself)
--	16#9000	The format bits 0 to 3 of the control word have been changed during the transmission. This error is indicated on the MLC/XLC with the error number 16#8B00
16#9500	16#9500	Identical (data was sent even though a read request was made)
16#9600	--	C1 bit has not been set. The error does not exist on VisualMotion, since the old formats of the parameter channel are still supported
16#9700	--	The "last" bit was not sent even though it is a read request. This error can only occur if a read request can be transferred in one fragment. This is not guaranteed for the VisualMotion since the parameter channel may only contain 2 words maximum as well. However, it is specified on the MLC/XLC that the parameter channel has to contain at least 3 words. Due to that reason, a read request can always be transferred with a telegram
16#1F00	16#1Fxx	Control-specific error. In VisualMotion, the control-specific error is indicated in the lower byte. In the MLC/XLC, 8 more bytes follow after the error code. These contain the error codes of the function blocks used to read/write parameters. The "Additional1" error codes is in the words 3 and 4 of the parameter channel, "Additional2" in the words 5 and 6
16#F200	16#F200	Identical (the given index is unknown)
16#F300	16#F300	Identical (the given subindex is unknown)

Tab. 11-9: Differences between the MLC/XLC and VisualMotion error codes in the parameter channel

12 ML_JobComm.library

12.1 Introduction and overview

The library ML_JobComm permits up to 255-byte sized telegrams, known here as jobs, to be sent to other controls.

Data is transmitted via the SCL protocol, which requires UDP. The transmission layer ensures that the data arrives at the receiver correctly and in the right order.

The job communication is real-time capable due to the adjustable maximum transmission time for jobs. This ensures that a data package arrives at the receiver within the required time. The transmitter is informed if this is not achieved.

Advantages The use of job communication provides the following advantages:

- Data transmission is real-time compatible, that is to say a job must arrive at the communication partner in a specified time
- Data transmission is guaranteed by a checksum
- The sequence of the jobs is guaranteed
- All jobs reach the communication partner
- The transmitter recognizes transmission errors
- It can communicate with multiple communication partners, without these mutually affecting each other
- Jobs can be sent or received decentrally at a number of points in the PLC program
- Jobs can be received in a filtered manner from the receiver buffer by means of a signature
- The implementation permits simultaneous use in multiple PLC tasks

Communication The communication across the job communication level between multiple controls takes place the Ethernet.

The UDP protocol is set up for this. A free UDP port on the control is needed. The JobComm uses port 7101 as the default port. It is possible to use other ports.

The following ports should not be used:

- Ports with a port number less than or equal to 1024, as these are used by the operating system for numerous services.
- Port 1740: PLC, IndraLogic
- Port 6040: ComServer

Data packages are transmitted and also received via a UDP port. No second UDP port is opened.

12.2 Architecture of the JobComm library

The JobComm library makes available multiple function blocks. All function blocks can be instantiated several times. The function block inputs are evaluated and the outputs are activated in line with Bosch Rexroth's general rules for function blocks.

Central communication The central function block ML_SendRecvJobAss takes on the main work ([ML_SendRecvJobAss, page 711](#)). The function block should run with high priority and a short cycle time in a proprietary PLC task. No data transmission

	is possible without the cyclic request of this function block. It provides a handle. Other function blocks can access the instance via that handle. All operations of the other function blocks are triggered by this handle.
Connections	A connection has to be established before job telegrams can be transmitted between two communication partners. An instance of the function block ML_SendRecvJobAss can administer multiple connections. The connections are controlled by the function block ML_CtrlJobConnect (ML_CtrlJobConnect, page 719). An attempted connection can either be active or passive. With the active establishment of a connection (<code>Command = ML_CONNECT_CMD_JOB</code>), an attempt is made to contact the communication partner directly. With a passive connection (<code>Command = ML_SYNC_CMD_JOB</code>) it waits until the remote host attempts to establish an active connection. The function block ML_CtrlJobConnect also performs the disconnections (<code>Command = ML_DISCONNECT_JOB</code>). As the actual connection and disconnection is performed by the instance of the function block ML_SendRecvJobAss, the ML_CtrlJobConnect function block does not return immediately with a result. Rather, several requests are generally required before the end of the action is signaled. The function block ML_ShowJobConnect requests the list of the known connections with the respective connection status. The connection status of the control can easily be checked by this.
Transmission	The function block ML_SendJobTlg is used to transmit jobs to a communication partner (ML_SendJobTlg, page 714). During the processing of the block, the job is merely transmitted to the transmission buffer. The actual transmission takes place in the instance of the function block ML_SendRecvJobAss. If a job cannot be sent within a specified period of time, then it is moved to the transmission junk buffer. The cause of a job not being sent can be a missing connection to the communication partner.
Receipt	The function block ML_RecvJobTlg searches through the receiver buffer for a data package with the specified signature. The actual receipt of the data packages takes place in the instance of the function block ML_SendRecvJobAss. Only one package is always read. If a job cannot be retrieved from the receiver buffer within a specified period of time, then it is moved to the receiver junk buffer. This generally happens if no request by the ML_RecvJobTlg function block fits the signature of the job.
Error analysis	To facilitate the error analysis, the function blocks ML_DebugJobConnect and ML_DebugJobConnect_Cyclic allow an output of the internal buffer of the ML_SendRecvJobAss instance (ML_DebugJobConnect, page 724 or ML_DebugJobConnect_Cyclic, page 727).. <i>The following buffers can be viewed:</i> • SendJobBuff: Contains all jobs ready to be sent • RecvJobBuff: Contains all received and not yet retrieved jobs • SendJobAssBuff: Contains the SCL package currently to be sent • SendUdpBuff: Contains the SCL package currently to be sent • RecvUdpBuff: Contains the SCL package last received

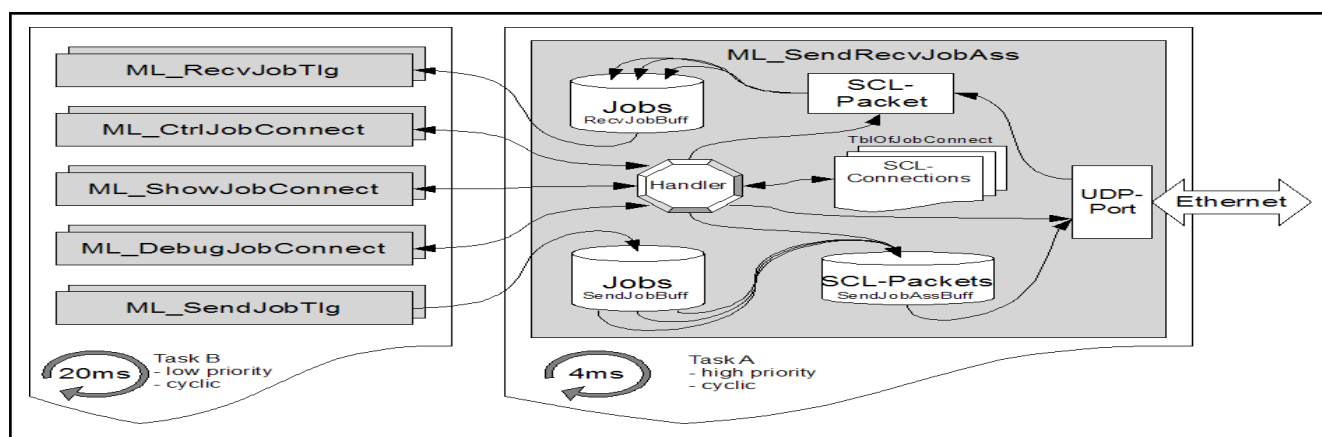


Fig. 12-1: Communication model of ML_JobComm

12.3 Function blocks

12.3.1 Introduction and overview

The "ML_JobComm" library implements the SCL protocol. The implementation has been designed in such a way that access to the data to be transmitted is as simple as possible. Jobs can be sent at any time. Received jobs can be taken from the receiver buffer by means of a signature. This enables jobs to be processed decentrally.

ML_JobComm function blocks

Function blocks	Description
ML_SendRecvJobAss, page 711	Used to prepare, transmit, receive and post-process consolidated orders (jobs).
ML_SendJobTlg, page 714	Used to command a job into the output buffer for job telegrams.
ML_RecvJobTlg, page 716	Used for the defined search for and reading of a job from the input buffer for job telegrams.
ML_CtrlJobConnect, page 719	Used for the local administration of a bidirectional port connection. Enables a connection or disconnection to be commanded and the status of a connection to be monitored.
ML_ShowJobConnect, page 722	Used for the one-time reading of the status of all local bidirectional port connections.
ML_DebugJobConnect, page 724	Used for the one-time transmission of the monitoring data from internal buffers.
ML_DebugJobConnect_Cyclic, page 727	Used for the cyclic transmission of the monitoring data from internal buffers.

Tab. 12-1: ML_JobComm library

12.3.2 ML_SendRecvJobAss

Brief description

The function block ML_SendRecvJobAss is used to prepare, transmit, receive and post-process consolidated orders (jobs). The function block is intended for use in a high-priority, cyclic time-controlled PLC user task.

Assignment: Target system/library

Target system	Library
MLC from 10VRS	ML_JobComm
MLD/MPx18	MX_JobComm.lib
MLD/MPC07	MX_JobComm.lib

Tab. 12-2: Reference table of the ML_SendRecvJobAss function block

Interface description

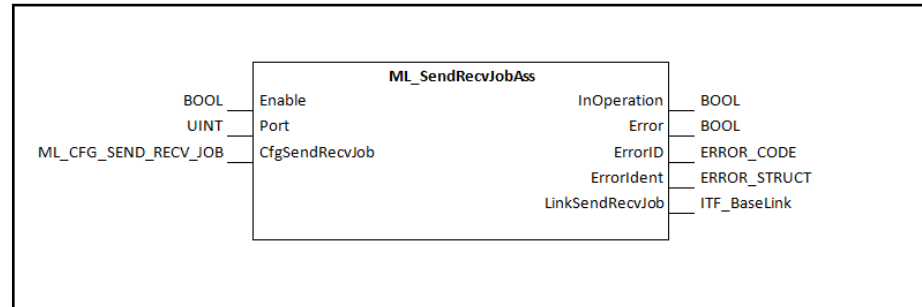


Fig. 12-2: ML_SendRecvJobAss function block

	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Processing enabled
	Port	UINT	UDP port for communication
	CfgSendRecvJob	ML_CFG_SEND_RECV_JOB	Configuration of the job telegrams and connections ML_CFG_SEND_RECV_JOB, page 731
VAR_OUTPUT	InOperation	BOOL	TRUE, if the cyclic communication is enabled. The output LinkSendRecvJob is valid.
	Error	BOOL	Indicates that an error occurred in the function block.
	ErrorID	ERROR_CODE	Error ID
	ErrorIdent	ERROR_STRUCT	For the error structure with further error subdivision, see Error codes, page 714 .
	LinkSendRecvJob	ITF_BaseLink	Handle to the instance of the function block

Tab. 12-3: ML_SendRecvJobAss interface

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
Port	UINT	0	65535	7101	Rising edge at "Enable"

Tab. 12-4: Input behavior of the ML_SendRecvJobAss

Functional description

The function block ML_SendRecvJobAss is used to prepare, transmit, receive and post-process consolidated orders (jobs). The function block is intended for use in a high-priority, cyclic time-controlled PLC user task.

Initialization

If there is a rising edge at "Enable", all of the local and global buffers will be deleted, the specified port and the CfgSendRecvJob configuration data will be checked for validity. The UDP port will then be opened. Following successful completion and preparation for the cyclic job communication, the "In-Operation" output is TRUE.

Normal mode

The following steps are performed in a cyclic request if the "Enable" input is active:

1. Check for time-out of the connections and UDP telegrams

All connections in the TblOfJobConnect table are checked for time-out. Incorrect connections are signaled on the synchronized instances of the ML_CtrlJobConnect block and are deleted from the list of open connections TblOfJobConnect without a coordinated disconnection. UDP telegrams, the connection of which has been deleted, will be identified in the SendJobAssBuff transmission buffer as invalid. An incorrect connection does not have any further effect on other connections of the instance of the enabled function block.

2. Check for time-out of job telegrams

All valid job telegrams are checked for time-out. Job telegrams with an exceeded dwell time are deleted from the respective buffer, that is to say are identified as being invalid. The header of a deleted job telegram and the corresponding IP are copied to the memory provided for this by the user (ML_CFG_JUNK_JOB). The telegram fragment contains the IP address (4 bytes) and a configurable number of bytes of the job telegram. The memory addressed via the "CfgSendRecvJob" input must have a minimum size of (4+number of JunkBuffer size) bytes and must be an integer multiple of the minimum size. The size of the telegram fragment is determined in the configuration structure ML_CFG_JUNK_JOB. The default value is 12.

3. Check and process the command buffer

One command is cyclically removed and processed from the command buffers in each case.

4. Summary and transmission of job telegrams

Depending on a corresponding open connection, the job telegrams from the transmission buffer are evaluated, consolidated into UDP telegrams and transmitted to the UDP telegram buffer.

5. Reading the Ethernet receiver buffer

The Ethernet receiver buffer is read and the telegrams or structure and content are evaluated. The read access is repeated until the Ethernet receiver buffer is empty or the maximum number of read accesses has been reached.

Depending on the content and status of the connection (Data, ACK, SYN, SYN ACK, FIN, FIN ACK, RST, ...), the received telegrams determine different reactions, such as storage of jobs, direct, one-time reaction telegrams or repeatable reaction telegrams. If the UDP telegram contains orders (jobs) then these are transmitted to the receiver buffer. Consolidated UDP telegrams, that is to say reaction telegrams with orders, are discarded as invalid.

6. Transmission of the UDP telegrams

Once the time-out has been checked, all of the valid UDP telegrams are sent, depending on the elements in the connection table.

7. Defragmenting the transmission and receiver buffer for job telegrams

Once the block has been disabled via "Enable" or, in the event of an error, no further telegrams are accepted and opened connections are closed without a coordinated disconnection.

Port

The port numbers to be used are specified at the "Port" input. This port may not be used for another purpose. If the port is already assigned or not usable,

the function block generates a corresponding error code when there is a rising edge at "Enable". For usable UDP ports, refer to [Communication, page 709](#).

CfgSendRecvJob

The configuration for the communication is created at CfgSendRecvJob ([ML_CFG_SEND_RECV_JOB, page 731](#)).

Error codes The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR, 16#0001	16#0000F790	16#00900001	The "Port" input is invalid.
INPUT_INVALID_ERROR, 16#0001	16#0000F790	16#00900002	Incorrect configuration of the junk properties of the transmission buffer
INPUT_INVALID_ERROR, 16#0001	16#0000F790	16#00900003	Incorrect configuration of the junk properties of the receiver buffer
DEVICE_ERROR, 16#0008	16#0000F790	16#00910001	Storage of a job telegram failed - buffer overflow in the receiver buffer for job telegrams
DEVICE_ERROR, 16#0008	16#0000F790	16#0095nnnn	Error when creating a communication socket (create) - error code Additional2, LowWord nnnn see operating system
DEVICE_ERROR, 16#0008	16#0000F790	16#0096nnnn	Error when closing a communication socket (close) - error code Additional2, LowWord nnnn see operating system
DEVICE_ERROR, 16#0008	16#0000F790	16#0097nnnn	Error when configuring a communication socket (ioctl) - error code Additional2, LowWord nnnn see operating system
DEVICE_ERROR, 16#0008	16#0000F790	16#0098nnnn	Error during the BIND process of a communication socket (bind) - error code Additional2, LowWord nnnn see operating system
DEVICE_ERROR, 16#0008	16#0000F790	16#0099nnnn	Error on receipt of data from the Ethernet receiver buffer (recvfrom) - error code Additional2, LowWord nnnn see operating system

Tab. 12-5: Error codes of the ML_SendRecvJobAss function block

12.3.3 ML_SendJobTlg

Brief description The function block ML_SendJobTlg is used to command a job into the output buffer for job telegrams. The function block is intended for use in a cyclic time-controlled PLC user task of medium priority.

Assignment: Target system/library

Target system	Library
MLC from 10VRS	ML_JobComm
MLD/MPx18	MX_JobComm.lib
MLD/MPC07	MX_JobComm.lib

Tab. 12-6: Reference table of the ML_SendJobTlg function block

Interface description

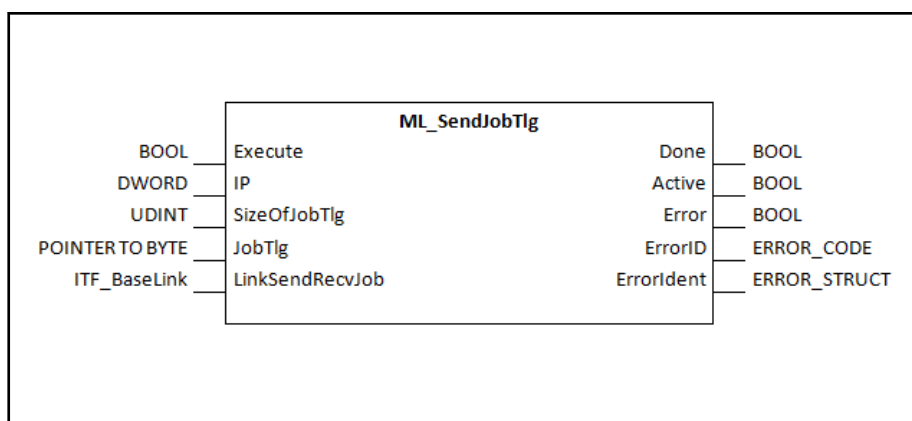


Fig. 12-3: ML_SendJobTlg function block

	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Processing enabled
	IP	DWORD	IP address of the remote device
	SizeOfJobTlg	UDINT	Number of valid bytes in the "JobTlg" data range, which contain the order to be transmitted
	JobTlg	POINTER TO BYTE	Pointer to the data range, which contains the order to be transmitted
	LinkSendRecvJob	ITF_BaseLink	Handle of an "ML_SendRecvJobAss" instance ML_SendRecvJobAss, page 711
VAR_OUTPUT	Done	BOOL	Transmission of the order has been completed without error
	Active	BOOL	Transmission of the order enabled
	Error	BOOL	Indicates that an error occurred in the function block.
	ErrorID	ERROR_CODE	Error ID
	ErrorIdent	ERROR_STRUCT	For the error structure with further error subdivision, see Error codes, page 716 .

Tab. 12-7: ML_SendJobTlg interface

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
IP	DWORD	0	16#FFFFFFFF	0	rising edge at "Execute"
SizeOfJobTlg	UINT	1	255	0	rising edge at "Execute"
JobTlg	POINTER TO BYTE			0	rising edge at "Execute"
LinkSendRecvJob	ITF_BaseLink			0	rising edge at "Execute"

Tab. 12-8: Input behavior of the ML_SendJobTlg

Functional description

The function block ML_SendJobTlg is used to command a job into the output buffer for job telegrams SendJobBuff. The function block is intended for use in a cyclic time-controlled PLC user task of medium priority.

When there is a rising edge at "Execute", the function block transmits the data of the order defined by the "JobTlg" and "SizeOfJobTlg" inputs into the output buffer for job telegrams once. An enabled transmission cannot be interrupted and replaced by a further one (no retrigger support).

The error free commanding of the order into the output buffer for job telegrams is signaled by the "Done" output. The "ErrorID" output and the elements of the "ErrorIdent" structure are not updated in this case.

If an error occurs while processing the function block, this is signaled by the "Error" output and specified by the outputs "ErrorID" and "ErrorIdent" updated in this case.

IP

The "IP" input provides the IP address of the communication partner as a number. The IP address in string form can be converted into the form required for the "IP" input via the "SysSocketInetAddr" function of the SysSocket library.

JobTlg, SizeOfJobTlg

At the "SizeOfJobTlg" input, the number of the bytes to be transmitted in the data range addressed by the "JobTlg" input is specified by the user.

LinkSendRecvJob

The "LinkSendRecvJob" input specifies the link to the corresponding instance of the function block ML_SendRecvJobAss and enables, among other things, the communication path and priority to be selected ([ML_SendRecvJobAss, page 711](#)).

Error codes

The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR, 16#0001	16#0000F770	16#00700001	Invalid telegram address (zero pointer)
INPUT_INVALID_ERROR, 16#0001	16#0000F770	16#00700002	Invalid telegram size (0)
INPUT_INVALID_ERROR, 16#0001	16#0000F770	16#00700003	Invalid telegram size (>255)
INPUT_INVALID_ERROR, 16#0001	16#0000F770	16#00700004	Invalid IP address
INPUT_INVALID_ERROR, 16#0001	16#0000F770	16#00700005	Invalid link
DEVICE_ERROR, 16#0008	16#0000F770	16#00710001	Buffer overflow in the transmission buffer for job telegrams

Tab. 12-9: Error codes of the ML_SendJobTlg function block

12.3.4 ML_RecvJobTlg

Brief description

The function block ML_RecvJobTlg is used for the defined search for and reading of a job from the input buffer for job telegrams. The function block is intended for use in a cyclic time-controlled PLC user task of medium priority.

Assignment: Target system/library

Target system	Library
MLC from 10VRS	ML_JobComm
MLD/MPx18	MX_JobComm.lib
MLD/MPC07	MX_JobComm.lib

Tab. 12-10: Reference table of the ML_RecvJobTlg function block

Interface description

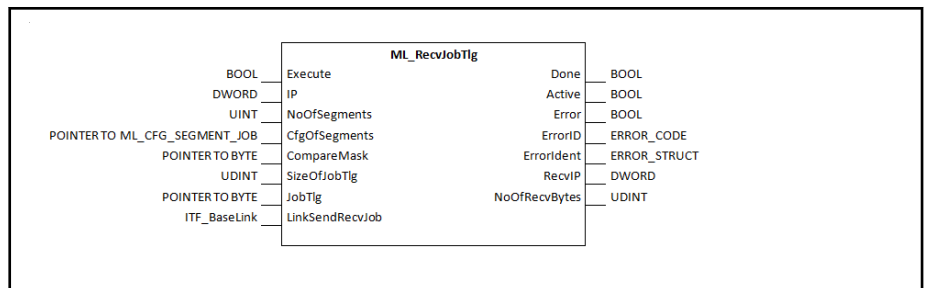


Fig. 12-4: ML_RcvJobTlg function block

	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Processing enabled
	IP	DWORD	IP address of the remote device
	NoOfSeg-ments	UINT	Number of segments of the search mask (number of array elements)
	CfgOfSeg-ments	POINTER TO ML_CFG_SEGMENT_JOB	Definition of the segments of the search mask ML_CFG_SEGMENT_JOB, page 731
	Compare-Mask	POINTER TO BYTE	Search mask
	SizeOfJobTlg	UDINT	Number of valid bytes in the "JobTlg" data range, which contain the order to be transmitted
	JobTlg	POINTER TO BYTE	Pointer to the data range, which contains the order to be transmitted
	LinkSendRecvJob	ITF_BaseLink	Handle of an "ML_SendRecvJobAss" instance ML_SendRecvJobAss, page 711
VAR_OUTPUT	Done	BOOL	Transmission of the order has been completed without error
	Active	BOOL	Transmission of the order enabled
	Error	BOOL	Indicates that an error occurred in the function block.
	ErrorID	ERROR_CODE	Error ID
	ErrorIdent	ERROR_STRUCT	For the error structure with further error subdivision, see Error codes, page 719 .
	RecvIP	DWORD	IP address of the transmitter
	NoOfRecv-Bytes	UDINT	Number of valid bytes in the data range addressed by JobAdr

Tab. 12-11: ML_RcvJobTlg interface

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
IP	DWORD	0	16#FFFFFFFF	0	rising edge at "Execute"
NoOfSegments	UINT	0	20	0	rising edge at "Execute"
CfgOfSegments	POINTER TO ML_CFG_SEGMENT_JOB			0	rising edge at "Execute"
CompareMask	POINTER TO BYTE			0	rising edge at "Execute"

Name	Type	Min. value	Max. value	Default value	Effective
SizeOfJobTlg	UDINT	1	255	0	rising edge at "Execute"
JobTlg	POINTER TO BYTE			0	rising edge at "Execute"
LinkSendRecvJob	ITF_BaseLink			0	rising edge at "Execute"

Tab. 12-12: Input behavior of the ML_RecvJobTlg

Functional description

The function block ML_RecvJobTlg is used for the defined search for and reading of a job from the input buffer for job telegrams RecvJobBuff. The function block is intended for use in a cyclic time-controlled PLC user task of medium priority.

After the processing of the function block has been enabled by "Execute", the function block ML_RecvJobTlg searches for a defined telegram within a specified range. The telegram searched for is specified via the inputs "NoOfSegments", "CfgOfSegments" and "CompareMask". At the "NoOfBytes" input, the number of bytes available in the data range addressed by the "JobTlg" input is specified by the user. An enabled transmission cannot be interrupted and replaced by a further one (no retrigger support).

The error-free finding and reading of the order from the input buffer for job telegrams is signaled by the "Done" output. The "ErrorID" output and the elements of the "ErrorIdent" structure are not updated in this case. The output "NoOfRecBytes" provides information on how many bytes have been read and, in the event of the error-free processing of the function block in the data range addressed by the "JobTlg" input, indicates how many are valid. If no telegram has been found with the corresponding specification, the "NoOfRecBytes" output is 0.

If an error occurs while processing the function block, this is signaled by the "Error" output and specified by the outputs "ErrorID" and "ErrorIdent" updated in this case.

The read accesses of the instances of the function block ML_RecvJobTlg to the input buffer for job telegrams are protected, that is to say at any one time only one PLC task job telegram can be read from the read buffer.

IP

The "IP" input provides the IP address of the communication partner as a number. The IP address in string form can be converted into the form required for the "IP" input via the "SysSocketInetAddr" function of the SysSocket library.

NoOfSegments, CfgOfSegments, CompareMask

The search mask "CompareMask" for an order is specified by its segment definition ("NoOfSegments", "CfgOfSegments"). A segment is defined from a starting point and the length of the segment in bytes. The first element of the search mask is byte 0.

The segment definition defines which bytes of a job are checked. Every segment is defined by its start position and length ([ML_CFG_SEGMENT_JOB](#), [page 731](#)).

The search mask "CompareMask" contains the search template. If multiple segments are defined, then the search templates of the individual segments are stored continuously, that is without separation, in "CompareMask". The number of the bytes in "CompareMask" must be the sum of all the segment sizes defined in "CfgOfSegments".

Example:

Two segments are defined. The first segment contains 4 bytes and starts at byte 0; the second contains 2 bytes and starts at byte 12.

This results in the following definition:

```
NoOfSegment := 2
```

```
CfgOfSegments := [(0,4), (12,2)]
```

The search masks correspondingly consists of 6 bytes and is defined as follows in this example:

```
CompareMask := [47, 12, 8, 15, 3, 98]
```

A received job data package could look like the following:

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 telegram with comparison mask

The telegram would therefore fit the search mask in this example and would thus be transmitted by the function block from the receiver buffer into the "JobTlg" data range.

JobTlg, SizeOfJobTlg

At the "SizeOfJobTlg" input, the number of the bytes to be transmitted in the data range addressed by the "JobTlg" input is specified by the user.

LinkSendRecvJob

The "LinkSendRecvJob" input specifies the link to the corresponding instance of the function block ML_SendRecvJobAss and enables, among other things, the communication path and priority to be selected.

Error codes

The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR, 16#0001	16#0000F780	16#00800001	Invalid telegram address (zero pointer)
INPUT_INVALID_ERROR, 16#0001	16#0000F780	16#00800002	Invalid telegram size (0)
INPUT_INVALID_ERROR, 16#0001	16#0000F780	16#00800003	Invalid IP address
INPUT_INVALID_ERROR, 16#0001	16#0000F780	16#00800004	Invalid link
INPUT_INVALID_ERROR, 16#0001	16#0000F780	16#00800005	Invalid search mask definition (number of segments equals zero or greater than 20, zero pointer of the mask and / or the configuration)
INPUT_INVALID_ERROR, 16#0001	16#0000F780	16#00800006	Invalid search mask definition (mask length details or telegram length)
DEVICE_ERROR, 16#0008	16#0000F780	16#00810001	Insufficient memory available to store the telegram

Tab. 12-14: Error codes of the ML_RecvJobTlg function block

12.3.5 ML_CtrlJobConnect

Brief description

The function block ML_CtrlJobConnect is used for the local administration of a bidirectional port connection. The function block enables a connection or disconnection to be commanded and the status of a connection to be monitored. The function block is intended for use in a cyclic time-controlled PLC user task of medium priority.

Assignment: Target system/library

Target system	Library
MLC from 10VRS	ML_JobComm
MLD/MPx18	MX_JobComm.lib
MLD/MPC07	MX_JobComm.lib

Tab. 12-15: Reference table of the ML_CtrlJobConnect function block

Interface description

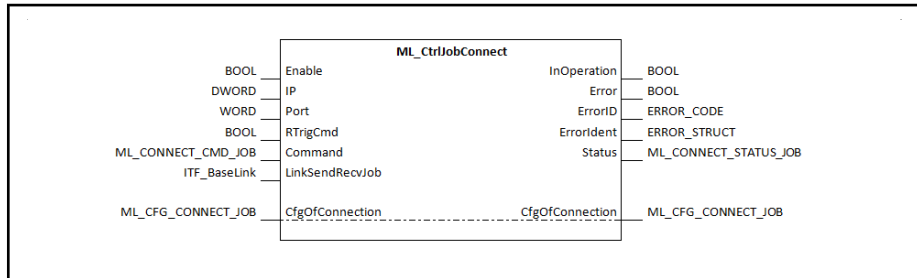


Fig. 12-5: ML_CtrlJobConnect function block

	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Processing enabled (cyclic, state-controlled)
	IP	DWORD	IP address of the remote device
	Port	WORD	UDP port of the remote device
	RTrigCmd	BOOL	Transfer of a new command at a rising edge
	Command	ML_CONNECT_CMD_JOB	Command ML_CONNECT_CMD_JOB , page 731..
	LinkSendRecvJob	ITF_BaseLink	Handle of an "ML_SendRecvJobAss" instance ML_SendRecvJobAss , page 711
VAR_OUTPUT	InOperation	BOOL	The function blocks is busy with its actual task
	Error	BOOL	Indicates that an error occurred
	ErrorID	ERROR_CODE	Error ID
	ErrorIdent	ERROR_STRUCT	For the error structure with further error subdivision, see Error codes , page 722.
	Status	ML_CONNECT_STATUS_JOB	Status of the connection ML_CONNECT_STATUS_JOB , page 731
VAR_IN_OUT	CfgOfConnection	ML_CFG_CONNECT_JOB	Configuration of the connection ML_CFG_CONNECT_JOB , page 730

Tab. 12-16: ML_CtrlJobConnect interface

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
IP	DWORD	0	16#FFFFFFFF	0	Rising edge at "Enable"
Port	WORD	0	65535	0	Rising edge at "Enable"
RTrigCmd	BOOL			FALSE	continuous if "Enable" is TRUE

Name	Type	Min. value	Max. value	Default value	Effective
Command	ML_CONNECT_CMD_JOB			ML_SYNC_CMD_JOB	rising edge at "Enable" or "RTrigCmd"
LinkSendRecvJob	ITF_BaseLink			0	Rising edge at "Enable"

Tab. 12-17: Input behavior of the ML_CtrlJobConnect

Functional description

The function block ML_CtrlJobConnect is used for the local administration of a bidirectional port connection. The function block enables a connection or disconnection to be commanded and the status of a connection to be monitored. The function block is intended for use in a cyclic time-controlled PLC user task of medium priority.

The function block can be instantiated several times for the structuring of the PLC program; however, a connection can always only be monitored by one block instance simultaneously. The function block can synchronize itself to a connection for the first time or resynchronize itself to it providing this connection already exists and no other instance has synchronized itself.

After the processing of the function block has been enabled by "Enable", the function block ML_CtrlJobConnect transmits the command specified at the "Command" input ([ML_SendRecvJobAss, page 711](#)). The "LinkSendRecvJob" input specifies the link to the corresponding instance of the function block ML_SendRecvJobAss and enables, among other things, the communication path and priority to be selected.

The active monitoring of a connection by the function block is signaled by "InOperation". The "ErrorID" output and elements of the "ErrorIdent" structure are not updated in this case.

If an error occurs while processing the function block, this is signaled by the "Error" output and specified by the outputs "ErrorID" and "ErrorIdent" updated in this case.

IP

The "IP" input provides the IP address of the communication partner as a number. The IP address in string form can be converted into the form required for the "IP" input via the "SysSockInetAddr" function of the SysSocket library.

RTrigCmd, Command

The "Command" input determines the action to be executed. A new command will be applied if there is a rising edge at Enable or if there is a rising edge at RTrigCmd.

The following commands are available:

- ML_SYNC_CMD_JOB: Enables synchronization to an existing connection (default). If no connection exists for the given IP, an error will be returned.
- ML_CONNECT_CMD_JOB: Enables the synchronization or opening of a connection to the specified IP (3-way handshake). If no connection already exists to the target address and if no other instance is already synchronized, then there will simply be synchronization to the existing connection. If the connection to the target address is already monitored by another instance, then an error will be returned.
- ML_DISCONNECT_CMD_JOB: Enables the closure and enabling of a connection to the specified IP (3-way handshake). If the connection

does not exist or synchronizes another instance, an error will be returned.

- **ML_FREE_CMD_JOB:** Facilitates the enabling of the administration rights of the proprietary connection or a connection administered by another block instance. If the administration rights of another block instance are enabled, then it will be notified and signals this, for its part, by setting the error outputs.

Status

The "Status" output issues the status of the connection specified via the "IP" input.

The following output values are possible:

- **ML_INVALID_STATUS_JOB:** Signals an invalid status that cannot be evaluated.
- **ML_DISCONNECTED_STATUS_JOB:** Signals that there is no existing connection to the specified target address (default).
- **ML_CONNECTING_STATUS_JOB:** Signals the establishment of a connection to the specified target address.
- **ML_CONNECTED_STATUS_JOB:** Signals an established connection to the specified target address.
- **ML_DISCONNECTING_STATUS_JOB:** Signals a disconnection from the specified target address.

CfgOfConnection

Values for the connection timeout and the timeout behavior of data packages are filed at the CfgOfConnection input ([ML_CFG_CONNECT_JOB](#), page 730).

Error codes The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR, 16#0001	16#0000F750	16#00500001	Invalid command
INPUT_INVALID_ERROR, 16#0001	16#0000F750	16#00500002	Invalid configuration
INPUT_INVALID_ERROR, 16#0001	16#0000F750	16#00500003	Invalid IP address
INPUT_INVALID_ERROR, 16#0001	16#0000F750	16#00500004	Invalid link
INPUT_INVALID_ERROR, 16#0001	16#0000F750	16#00500005	The connection is controlled by another function block instance.

Tab. 12-18: Error codes of the *ML_CtrlJobConnect* function block

12.3.6 ML_ShowJobConnect

Brief description The function block *ML_ShowJobConnect* is used for the one-time reading of the statuses of all local bidirectional port connections. The function block is intended for use in a cyclic time-controlled PLC task of medium priority.

Assignment: Target system/library

Target system	Library
MLC from 10VRS	ML_JobComm
MLD/MPx18	MX_JobComm.lib
MLD/MPC07	MX_JobComm.lib

Tab. 12-19: Reference table of the *ML_ShowJobConnect* function block

Interface description

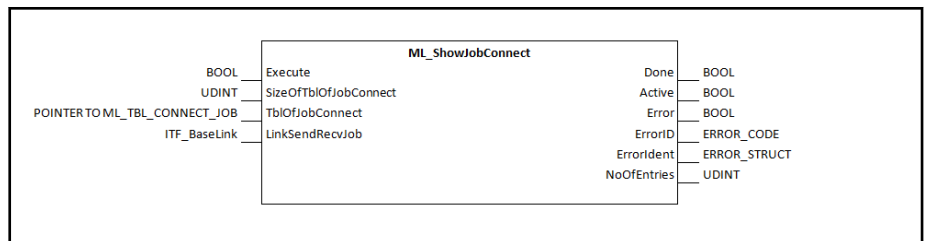


Fig. 12-6: ML_ShowJobConnect function block

	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Processing enabled
	SizeOfTbOfJobConnect	UDINT	Number of available bytes in the data range addressed by "TbOfJobConnect"
	TbOfJobConnect	POINTER TO ML_TBL_CONNECT_JOB	Pointer to the available data range for the connections to be read ML_TBL_CONNECT_JOB , page 735..
	LinkSendRecvJob	ITF_BaseLink	Handle of an "ML_SendRecvJobAss" instance ML_SendRecvJobAss , page 711
VAR_OUTPUT	Done	BOOL	Transmission of the connections has been completed without error
	Active	BOOL	Transmission of the connections enabled
	Error	BOOL	Indicates that an error occurred in the function block.
	ErrorID	ERROR_CODE	Error ID
	ErrorIdent	ERROR_STRUCT	For the error structure with further error subdivision, see Error codes , page 724.
	NoOfEntries	UDINT	Number of valid connections

Tab. 12-20: ML_ShowJobConnect interface

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
SizeOfTbOfJobConnect	UDINT	0	16#FFFFFFFF	0	rising edge at "Execute"
TbOfJobConnect	POINTER TO ML_TBL_CONNECT_JOB			0	rising edge at "Execute"
LinkSendRecvJob	ITF_BaseLink			0	rising edge at "Execute"

Tab. 12-21: Input behavior of the ML_ShowJobConnect

Functional description

The function block ML_ShowjobConnect is used for the one-time reading of the statuses of all local bidirectional port connections. The function block is intended for use in a cyclic time-controlled PLC task of medium priority.

If the reading of the current status of the connections has been started by ML_ShowJobConnect, precisely one connection is transmitted to the target range with each request of the ML_SendRecvJobAss instance, that is to say the entries in the target range are not consistent with each other ([ML_SendRecvJobAss](#), page 711).

The function block can be instantiated several times and simultaneously enabled, as, however, only one connection can always be transmitted to a target range, the transmission period of all instances is lengthened accordingly.

After the processing of the function block has been enabled by "Execute", the function block ML_ShowJobConnect transmits an order for reading of the connection table. An enabled transmission can be interrupted and replaced by a new one (retrigger support is provided).

At the "NoOfBytes" input, the number of the bytes available in the data range addressed by the "TblOfJobConnect" input is specified by the user. The data range provided for the order is specified by the "TblOfJobConnect" input and has to be referenced to an array of the data type ML_TBL_CONNECT_JOB ([ML_TBL_CONNECT_JOB](#), [page 735](#)). The "LinkSendRecvJob" input specifies the link to the corresponding instance of the function block ML_SendRecvJobAss and facilitates the selection of the communication path and the priority.

The error-free transmission of all connections is signaled by the "Done" output. The ErrorID output and elements of the "ErrorIdent" structure are not updated in this case. The "NoOfEntries" output provides information on how many entries have been read and, in the event of the error-free processing of the function block in the data range addressed by the "TblOfJobConnect" input, on how many are valid.

The "Active" output indicates the duration of transmission of the data; during this period of time, the further outputs are invalid and cannot be evaluated.

If an error occurs during processing of the function block, this is signaled by the "Error" output and specified by the "ErrorID" output and the structure elements of "ErrorIdent", which are updated in this case.

SizeOfTblOfJobConnect, TblOfJobConnect

The memory addressed by the "TblOfJobConnect" input must have a minimum size of sizeof (ML_TBL_CONNECT_JOB) bytes and must be an integer multiple of sizeof (ML_TBL_CONNECT_JOB).

TblOfJobConnect is a pointer to an array of the data type "ML_TBL_CONNECT_JOB" ([ML_TBL_CONNECT_JOB](#), [page 735](#)).

Error codes

The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR, 16#0001	16#0000F760	16#00600001	Invalid data range (zero pointer)
INPUT_INVALID_ERROR, 16#0001	16#0000F760	16#00600002	Invalid size of data range (0)
INPUT_INVALID_ERROR, 16#0001	16#0000F760	16#00500003	Invalid link

Tab. 12-22: Error codes of the ML_ShowJobConnect function block

12.3.7 ML_DebugJobConnect

Brief description

The function block ML_DebugJobConnect is used for the one-time transmission of the monitoring data from internal buffers.

Assignment: Target system/library

Target system	Library
MLC from 10VRS	ML_JobComm
MLD/MPx18	MX_JobComm.lib
MLD/MPC07	MX_JobComm.lib

Tab. 12-23: Reference table of the ML_DebugJobConnect function block

Interface description

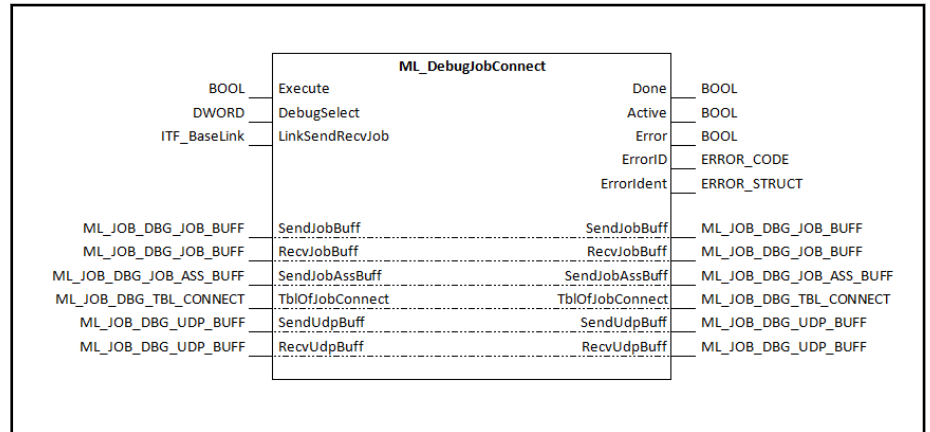


Fig. 12-7: ML_DebugJobConnect function block

	Name	Data type	Description
VAR_INPUT	Execute	BOOL	Processing enabled
	DebugSelect	DWORD	Selection of debug views
	LinkSendRecvJob	ITF_BaseLink	Handle of an "ML_SendRecvJobAss" instance ML_SendRecvJobAss, page 711
VAR_OUTPUT	Done	BOOL	Transmission of the monitoring data has been completed without error
	Active	BOOL	Transmission of the monitoring data enabled
	Error	BOOL	Indicates that an error occurred in the function block.
	ErrorID	ERROR_CODE	Error ID
	ErrorIdent	ERROR_STRUCT	For the error structure with further error subdivision, see Error codes, page 727 .
VAR_IN_OUT	SendJobBuff	ML_JOB_DBG_JOB_BUFF	Monitoring data of the SendJobBuff ML_JOB_DBG_JOB_BUFF, page 734..
	RecvJobBuff	ML_JOB_DBG_JOB_BUFF	Monitoring data of the RecvJobBuff ML_JOB_DBG_JOB_BUFF, page 734..
	SendJobAssBuff	ML_JOB_DBG_JOB_ASS_BUFF	Monitoring data of the SendJobAssBuff ML_JOB_DBG_JOB_ASS_BUFF, page 734..
	TblOfJobConnect	ML_JOB_DBG_TBL_CONNECT	Monitoring data of the TblOfJobConnect ML_JOB_DBG_TBL_CONNECT, page 734..
	SendUdpBuff	ML_JOB_DBG_UDP_BUFF	Monitoring data of the SendUdpBuff ML_JOB_DBG_UDP_BUFF, page 735..
	RecvUdpBuff	ML_JOB_DBG_UDP_BUFF	Monitoring data of the RecvUdpBuff ML_JOB_DBG_UDP_BUFF, page 735..

Tab. 12-24: ML_DebugJobConnect interface

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous
DebugSelect	DWORD	16#00	16#3F	0	rising edge at "Execute"
LinkSendRecvJob	ML_LINK_SEND_RECV_JOB			0	rising edge at "Execute"

Tab. 12-25: Input behavior of the ML_DebugJobConnect

Functional description

After the processing of the function block has been enabled by "Execute", the function block ML_DebugJobConnect transmits an order for transmission of the monitoring data (view). An enabled transmission can be interrupted and replaced by a new one (retrigger support is provided).

The error-free transmission of the selected monitoring data (views) is signaled by the "Done" output. The "ErrorID" output and elements of the "ErrorIdent" structure are not updated in this case.

The "Active" output indicates the duration of transmission of the data. During this period, the other outputs are invalid and cannot be evaluated.

If an error occurs during processing of the function block, this is signaled by the "Error" output and specified by the "ErrorID" output and the structure elements of "ErrorIdent", which are updated in this case.

DebugSelect

At the "DebugSelect" input, the views to be generated are specified by the user, whereby global constants are available for selection, which can be OR-linked.

The following constants are supported:

- 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

The "LinkSendRecvJob" input specifies the link to the corresponding instance of the function block "ML_SendRecvJobAss" and facilitates the selection of the communication path and the priority ([ML_SendRecvJobAss, page 711](#)).

SendJobBuff

The reference "SendJobBuff" can accept information via the job transmission buffer. It is therefore possible to monitor the actual status of the processing of the job. "SendJobBuff" is defined by the data type "ML_JOB_DBG_JOB_BUFF" ([ML_JOB_DBG_JOB_BUFF, page 734](#)).

RecvJobBuff

The reference "RecvJobBuff" can accept information via the job receiver buffer. It is therefore possible to monitor the actual status of the processing of the job. "RecvJobBuff" is defined by the data type "ML_JOB_DBG_JOB_BUFF" ([ML_JOB_DBG_JOB_BUFF, page 734](#)).

SendJobAssBuff

The reference "SendJobAssBuff" can accept information via the UDP transmission buffer. It is therefore possible to monitor the current UDP package. SendJobAssBuff is defined by the data type "ML_JOB_DBG_JOB_ASS_BUFF" (ML_JOB_DBG_JOB_ASS_BUFF, page 734).

TblOfJobConnect

The reference "TblOfJobConnect" can accept information via the connections to other communication partners. The functionality is comparable to the function block "ML_ShowJobConnect" (ML_ShowJobConnect, page 722). TblOfJobConnect is defined by the data type "ML_JOB_DBG_TBL_CONNECT" (ML_JOB_DBG_TBL_CONNECT, page 734).

SendUdpBuff

The reference "SendUdpBuff" can accept information via the UDP transmission buffer. It is therefore possible to monitor the UDP package currently being sent. SendUdpBuff is defined by the data type "ML_JOB_DBG_UDP_BUFF" (ML_JOB_DBG_UDP_BUFF, page 735).

RecvUdpBuff

The reference "RecvUdpBuff" can accept information via the UDP reception buffer. It is therefore possible to monitor the UDP packages received. RecvUdpBuff is defined by the data type "ML_JOB_DBG_UDP_BUFF" (ML_JOB_DBG_UDP_BUFF, page 735).

Error codes

The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR, 16#0001	16#0000F7A0	16#00A00001	Invalid selection of debug views
INPUT_INVALID_ERROR, 16#0001	16#0000F7A0	16#00A00003	Invalid link

Tab. 12-26: Error codes of the ML_DebugJobConnect function block

12.3.8 ML_DebugJobConnect_Cyclic

Brief description

The function block ML_DebugJobConnect_Cyclic is used for the cyclic transmission of the monitoring data from internal buffers.

Assignment: Target system/library

Target system	Library
MLC from 14VRS	ML_JobComm

Tab. 12-27: Reference table of the FB ML_DebugJobConnect_Cyclic

Interface description

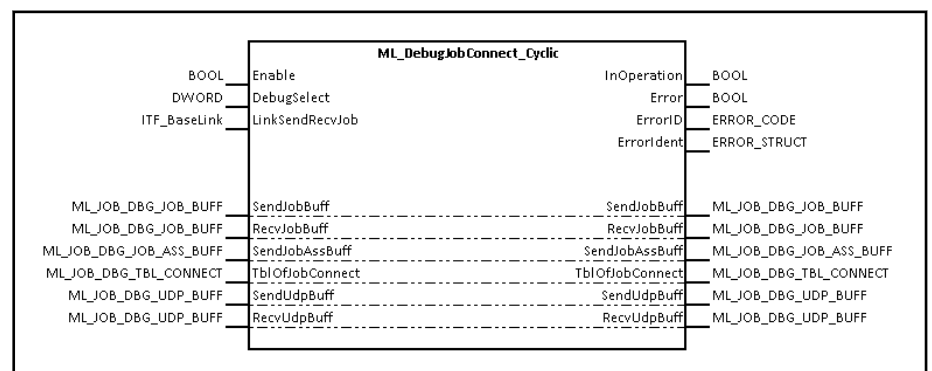


Fig. 12-8: ML_DebugJobConnect_Cyclic function block

	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Processing enabled
	DebugSelect	DWORD	Selection of debug views
	LinkSendRecvJob	ITF_BaseLink	Handle of an "ML_SendRecvJobAss" instance
VAR_OUTPUT	InOperation	BOOL	Transmission of the monitoring data has been completed without error
	Error	BOOL	Indicates that an error occurred in the function block.
	ErrorID	ERROR_CODE	Error ID
	ErrorIdent	ERROR_STRUCT	For the error structure with further error subdivision, see Error codes, page 729 .
VAR_IN_OUT	SendJobBuff	ML_JOB_DBG_JOB_BUFF	Monitoring data of the SendJobBuff ML_JOB_DBG_JOB_BUFF, page 734..
	RecvJobBuff	ML_JOB_DBG_JOB_BUFF	Monitoring data of the RecvJobBuff ML_JOB_DBG_JOB_BUFF, page 734..
	SendJobAssBuff	ML_JOB_DBG_JOB_ASS_BUFF	Monitoring data of the SendJobAssBuff ML_JOB_DBG_JOB_ASS_BUFF, page 734..
	TblOfJobConnect	ML_JOB_DBG_TBL_CONNECT	Monitoring data of the TblOfJobConnect ML_JOB_DBG_TBL_CONNECT, page 734..
	SendUdpBuff	ML_JOB_DBG_UDP_BUFF	Monitoring data of the SendUdpBuff ML_JOB_DBG_UDP_BUFF, page 735..
	RecvUdpBuff	ML_JOB_DBG_UDP_BUFF	Monitoring data of the RecvUdpBuff ML_JOB_DBG_UDP_BUFF, page 735..

Tab. 12-28: ML_DebugJobConnect_Cyclic interface

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
DebugSelect	DWORD	16#00	16#3F	0	Continuous
LinkSendRecvJob	ML_LINK_SEND_RECV_JOB			0	Rising edge at "Enable"

Tab. 12-29: Input behavior of the ML_DebugJobConnect_Cyclic

Functional description

After the processing of the function block has been enabled via "Enable", the function block ML_DebugJobConnect_Cyclic transmits the monitoring data ordered (view). Monitoring data is cyclically output.

The error-free transmission of the selected monitoring data (views) is signaled by the "InOperation" output. The "ErrorID" output and elements of the "ErrorIdent" structure are not updated in this case.

If an error occurs during processing of the function block, this is signaled by the "Error" output and specified by the "ErrorID" output and the structure elements of "ErrorIdent", which are updated in this case.

DebugSelect

At the "DebugSelect" input, the views to be generated are specified by the user, whereby global constants are available for selection, which can be OR-linked.

The following constants are supported:

- 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

The "LinkSendRecvJob" input specifies the link to the corresponding instance of the function block ML_SendRecvJobAss and facilitates the selection of the communication path and the priority ([ML_SendRecvJobAss, page 711](#))

SendJobBuff

The reference "SendJobBuff" can accept information via the job transmission buffer. It is therefore possible to monitor the actual status of the processing of the job. "SendJobBuff" is defined by the data type "ML_JOB_DBG_JOB_BUFF" ([ML_JOB_DBG_JOB_BUFF, page 734](#)).

RecvJobBuff

The reference "RecvJobBuff" can accept information via the job receiver buffer. It is therefore possible to monitor the actual status of the processing of the job. "RecvJobBuff" is defined by the data type "ML_JOB_DBG_JOB_BUFF" ([ML_JOB_DBG_JOB_BUFF, page 734](#)).

SendJobAssBuff

The reference "SendJobAssBuff" can accept information via the UDP transmission buffer. It is therefore possible to monitor the current UDP package. SendJobAssBuff is defined by the data type "ML_JOB_DBG_JOB_ASS_BUFF" ([ML_JOB_DBG_JOB_ASS_BUFF, page 734](#)).

TblOfJobConnect

The reference "TblOfJobConnect" can accept information via the connections to other communication partners. The functionality is comparable to the function block "ML_ShowJobConnect" ([ML_ShowJobConnect, page 722](#)). TblOfJobConnect is defined by the data type "ML_JOB_DBG_TBL_CONNECT" ([ML_JOB_DBG_TBL_CONNECT, page 734](#)).

SendUdpBuff

The reference "SendUdpBuff" can accept information via the UDP transmission buffer. It is therefore possible to monitor the UDP package currently being sent. SendUdpBuff is defined by the data type "ML_JOB_DBG_UDP_BUFF" ([ML_JOB_DBG_UDP_BUFF, page 735](#)).

RecvUdpBuff

The reference "RecvUdpBuff" can accept information via the UDP reception buffer. It is therefore possible to monitor the UDP packages received. RecvUdpBuff is defined by the data type "ML_JOB_DBG_UDP_BUFF" ([ML_JOB_DBG_UDP_BUFF, page 735](#)).

Error codes

The function block uses the error table F_RELATED_TABLE, 16#0170. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR, 16#0001	16#0000F7A0	16#00A00001	Invalid selection of debug views
INPUT_INVALID_ERROR, 16#0001	16#0000F7A0	16#00A00003	Invalid link

Tab. 12-30: Error codes of the FB ML_DebugJobConnect_Cyclic

12.4 Data types

12.4.1 ML_CFG_CONNECT_JOB

Type definitions The structure "ML_CFG_CONNECT_JOB" is used within the structure "ML_CFG_SEND_RECV_JOB" to configure the function blocks "ML_SendRecvJobAss" and "ML_CtrlJobConnect".

The "ML_CFG_CONNECT_JOB" structure contains the configuration of the connections.

Structure element	Type	Description
TimeOutOfExchange	TIME	Waiting time without data transfer before the connection is closed
TimeOutUntilRepeat	TIME	Waiting time without confirmation before the UDP package is sent again
MaxNofRepeat	UDINT	Maximum number of repetitions for an UDP package

Tab. 12-31: Elements of the ML_CFG_CONNECT_JOB structure

Also refer to

- [ML_CFG_SEND_RECV_JOB, page 731](#)
- [ML_SendRecvJobAss, page 711](#)
- [ML_CtrlJobConnect, page 719](#)

12.4.2 ML_CFG_JUNK_JOB

Type definitions The structure "ML_CFG_JUNK_JOB" is used within the structure ML_CFG_SEND_RECV_JOB to configure the function block "ML_SendRecvJobAss".

The "ML_CFG_JUNK_JOB" structure contains the configuration for the junk buffer. It has to be written to the fields "SizeOfStorage", "AdrOfStorage" and "TimeOutOfJunk" to enable the junk buffer

Structure element	Type	Default	Description
SizeOfStorage	UDINT	0	Complete size of the junk job buffer. SizeOfStorage=Number of entries*(4+SizeOfJunkJob)
AdrOfStorage	POINTER TO BYTE	0	Pointer to the junk job buffer
ActIndex	UDINT	0	Active index of the junk job buffer
TimeOutOfJunk	TIME	t#0s	Maximum period of time before a job is removed from the buffer
SizeOfJunkJob	UDINT	12	Number of bytes of the job that should be copied into the buffer.

Tab. 12-32: Elements of the ML_CFG_JUNK_JOB structure

Also refer to

- [ML_CFG_SEND_RECV_JOB, page 731](#)

- [ML_SendRecvJobAss, page 711](#)

12.4.3 ML_CFG_SEGMENT_JOB

Type definitions The structure "ML_CFG_SEGMENT_JOB" is used by the function block ML_RecvJobTlg ([ML_RecvJobTlg, page 716](#)).

The structure "ML_CFG_SEGMENT_JOB" contains the configuration of the comparison mask.

Structure element	Type	Description
OffsetOfSegment	UDINT	Segment offset within the mask flow
SizeOfSegment	UDINT	Size of the segment

Tab. 12-33: Elements of the ML_CFG_SEGMENT_JOB structure

12.4.4 ML_CFG_SEND_RECV_JOB

Type definitions The structure "ML_CFG_SEND_RECV_JOB" is used for the configuration of the function block "ML_SendRecvJobAss" ([ML_SendRecvJobAss, page 711](#)).

The structure "ML_CFG_SEND_RECV_JOB" contains the configuration for the connections and the junk buffer.

Structure element	Type	Description
SendJunk	ML_CFG_JUNK_JOB structure	Configuring the send junk buffer ML_CFG_JUNK_JOB, page 730
RecvJunk	ML_CFG_JUNK_JOB structure	Configuring the reception junk buffer ML_CFG_JUNK_JOB, page 730
CfgOfConnection	ML_CFG_CONNECT_JOB	Configuration of the connection ML_CFG_CONNECT_JOB, page 730

Tab. 12-34: Elements of the ML_CFG_SEND_RECV_JOB structure

12.4.5 ML_CONNECT_CMD_JOB

The enumeration type "ML_CONNECT_CMD_JOB" is used in the function block "ML_CtrlJobConnect" ([ML_CtrlJobConnect, page 719](#)).

Element	Value	Description
ML_SYNC_CMD_JOB	0	Passive establishment of connection
ML_CONNECT_CMD_JOB	1	Active establishment of connection
ML_DISCONNECT_CMD_JOB	2	Disconnection
ML_FREE_CMD_JOB	3	Delete connection without disconnecting

Tab. 12-35: Definition of the enumeration type ML_CONNECT_CMD_JOB

12.4.6 ML_CONNECT_STATUS_JOB

The enumeration type "ML_CONNECT_STATUS_JOB" is used in the function block "ML_CtrlJobConnect" ([ML_CtrlJobConnect, page 719](#)).

Element	Value	Description
ML_ERROR_MSG_STATUS_JOB	-2	An error has occurred
ML_INVALID_STATUS_JOB	-1	No valid status is available
ML_DISCONNECTED_STATUS_JOB	0	No connection

Element	Value	Description
ML_CONNECTING_STATUS_JOB	1	Connection is currently being established
ML_CONNECTED_STATUS_JOB	2	Connection has been established
ML_DISCONNECTING_STATUS_JOB	3	Currently disconnecting

Tab. 12-36: Definition of the enumeration type `ML_CONNECT_STATUS_JOB`

12.4.7 ML_JOB_ASS_BUFF

Type definitions The structure "ML_JOB_ASS_BUFF" is used by the data structure "ML_JOB_DBG_JOB_ASS_BUFF" ([ML_JOB_DBG_JOB_ASS_BUFF](#), page 734).

The structure "ML_JOB_ASS_BUFF" contains information on the job assembly buffer.

Structure element	Type	Description
DataValid	UDINT	Data valid
Length	UDINT	SCL telegram length
Destination	ML_JOB_ASS_BUFF_ADDR	Target address ML_JOB_ASS_BUFF_ADDR , page 732
Scl	ML_JOB_ASS_BUFF_SCL	Monitoring data ML_JOB_ASS_BUFF_SCL , page 733
Data	ML_JOB_ASS_BUFF_DATA	Job data ML_JOB_ASS_BUFF_DATA , page 732

Tab. 12-37: Elements of the `ML_JOB_ASS_BUFF` structure

12.4.8 ML_JOB_ASS_BUFF_ADDR

Type definitions The structure "ML_JOB_ASS_BUFF_ADDR" is used by the data structure "ML_JOB_ASS_BUFF" ([ML_JOB_ASS_BUFF](#), page 732).

The structure "ML_JOB_ASS_BUFF_ADDR" contains information on the job assembly buffer.

Structure element	Type	Description
Family	INT	Family
Port	UINT	Port
Addr	UDINT	IP address
Dummy	ARRAY[0..7] OF BYTE	Dummy

Tab. 12-38: Elements of the `ML_JOB_ASS_BUFF_ADDR` structure

12.4.9 ML_JOB_ASS_BUFF_DATA

Type definitions The structure "ML_JOB_ASS_BUFF_DATA" is used by the data structure "ML_JOB_ASS_BUFF" ([ML_JOB_ASS_BUFF](#), page 732).

The structure "ML_JOB_ASS_BUFF_DATA" contains information on the data of the job assembly buffer.

Structure element	Type	Description
Data	ARRAY [1..K_ML_DBG_JOB_SIZE_OF_UDP_DATAGRAM] OF BYTE	UDP package

Tab. 12-39: Elements of the `ML_JOB_ASS_BUFF_DATA` structure

12.4.10 ML_JOB_ASS_BUFF_SCL

Type definitions The structure "ML_JOB_ASS_BUFF_SCL" is used by the data structure "ML_JOB_ASS_BUFF" ([ML_JOB_ASS_BUFF](#), page 732).

The structure "ML_JOB_ASS_BUFF_SCL" contains information on the control structure.

Structure element	Type	Description
SeqNoLocal	UDINT	Local sequence number
SeqNoRemote	UDINT	Sequence number of the remote host
Ctrl	WORD	Check word
ErrorCode	WORD	Error code
Res01	WORD	Reserved
Res02	WORD	Reserved
TimeStampLow	DWORD	Time stamp (low 64 bits)
TimeStampHigh	DWORD	Time stamp (high 64 bits)
Res03	DWORD	Reserved
Version	UINT	Protocol version
Res04	WORD	Reserved
NofDataSet	UINT	Number of job telegrams
Res05	WORD	Reserved

Tab. 12-40: Elements of the ML_JOB_ASS_BUFF_SCL structure

12.4.11 ML_JOB_BUFF

Type definitions The structure "ML_JOB_BUFF" is used by the data structure "ML_JOB_DBG_JOB_BUFF" ([ML_JOB_DBG_JOB_BUFF](#), page 734).

The structure "ML_JOB_BUFF" contains information on the job buffer.

Structure element	Type	Description
Management	ML_JOB_BUFF_MANAGEMENT	Management information of the job telegram ML_JOB_BUFF_MANAGEMENT , page 734
Data	ML_JOB_BUFF_DATA	Job telegram data ML_JOB_BUFF_DATA , page 733

Tab. 12-41: Elements of the ML_JOB_BUFF structure

12.4.12 ML_JOB_BUFF_DATA

Type definitions The structure "ML_JOB_BUFF_DATA" is used by the data structures "ML_JOB_BUFF" and "ML_JOB_BUFF_MANAGEMENT".

The structure "ML_JOB_BUFF_DATA" contains the job data.

Structure element	Type	Description
Ip	DWORD	IP address of the target
Length	UDINT	Length of the job telegram
TimeStamp	UDINT	Time stamp of the job telegram
Job	ARRAY[1..256] OF BYTE	Job telegram data

Tab. 12-42: Elements of the ML_JOB_BUFF_DATA structure

Also refer to

- [ML_JOB_BUFF](#), page 733
- [ML_JOB_BUFF_MANAGEMENT](#), page 734

12.4.13 ML_JOB_BUFF_MANAGEMENT

Type definitions The structure "ML_JOB_BUFF_MANAGEMENT" is used by the data structure "ML_JOB_BUFF" ([ML_JOB_BUFF](#), page 733).

The structure "ML_JOB_BUFF_MANAGEMENT" contains information on the job buffer.

Structure element	Type	Description
DataValid	UDINT	1, data is valid 0, data is invalid
Data	POINTER TO ML_JOB_BUFF_DATA	Pointer to data range where the job telegram data is saved. ML_JOB_BUFF_DATA , page 733

Tab. 12-43: Elements of the ML_JOB_BUFF_MANAGEMENT structure

12.4.14 ML_JOB_DBG_JOB_ASS_BUFF

Type definitions The structure "ML_JOB_DBG_JOB_ASS_BUFF" is used by the function block "ML_DebugJobConnect" ([ML_DebugJobConnect](#), page 724).

The structure "ML_JOB_DBG_JOB_ASS_BUFF" contains information on the UDP transmission buffer.

Structure element	Type	Description
Buffer	POINTER TO ML_JOB_ASS_BUFF	Pointer to the UDP buffer ML_JOB_ASS_BUFF , page 732
NofElements	UDINT	Number of jobs in the UDP buffer

Tab. 12-44: Elements of the ML_JOB_DBG_JOB_ASS_BUFF structure

12.4.15 ML_JOB_DBG_JOB_BUFF

Type definitions The structure "ML_JOB_DBG_JOB_BUFF" is used by the function block "ML_DebugJobConnect" ([ML_DebugJobConnect](#), page 724).

The structure "ML_JOB_DBG_JOB_BUFF" contains information on the job buffer.

Structure element	Type	Description
Buffer	POINTER TO ML_JOB_BUFF	Pointer to the job buffer ML_JOB_BUFF , page 733
NofElements	UDINT	Number of lines in the job buffer
RdIdx	POINTER TO UDINT	Read index of the job buffer
WrIdx	POINTER TO UDINT	Write index of the job buffer

Tab. 12-45: Elements of the ML_JOB_DBG_JOB_BUFF structure

12.4.16 ML_JOB_DBG_TBL_CONNECT

Type definitions The structure "ML_JOB_DBG_TBL_CONNECT" is used by the function block "ML_DebugJobConnect" ([ML_DebugJobConnect](#), page 724).

The structure "ML_JOB_DBG_TBL_CONNECT" contains information on the connection table.

Structure element	Type	Description
Buffer	POINTER TO ML_TBL_CONNECT_JOB	Pointer to the connection table ML_TBL_CONNECT_JOB, page 735
NofElements	UDINT	Number connections

Tab. 12-46: Elements of the ML_JOB_DBG_TBL_CONNECT structure

12.4.17 ML_JOB_DBG_UDP_BUFF

Type definitions The structure "ML_JOB_DBG_UDP_BUFF" is used by the function block "ML_DebugJobConnect" ([ML_DebugJobConnect, page 724](#)).

The structure "ML_JOB_DBG_UDP_BUFF" contains information on the UDP input and output buffer.

Structure element	Type	Description
Buffer	POINTER TO ML_JOB_ASS_BUFF	Pointer to the UDP buffer ML_JOB_ASS_BUFF, page 732
NofBytes	UDINT	Size of the current UDP package in bytes

Tab. 12-47: Elements of the ML_JOB_DBG_UDP_BUFF structure

12.4.18 ML_LINK_SEND_RECV_JOB

The data type "ML_LINK_SEND_RECV_JOB" is output by the function block "ML_SendRecvJobAss" at the output "CfgSendRecvJob" and contains a handle to the instance of the "ML_SendRecvJobAss" function block ([ML_SendRecvJobAss, page 711](#)).

Other function blocks of ML_JobComm need this handle to control the function block "ML_SendRecvJobAss".

The internal definition of the data type "ML_LINK_SEND_RECV_JOB" is not relevant to the application.

12.4.19 ML_TBL_CONNECT_JOB

Type definitions The structure "ML_TBL_CONNECT_JOB" is used by the function block ML_ShowJobConnect ([ML_ShowJobConnect, page 722](#)).

The structure "ML_TBL_CONNECT_JOB" contains a connection entry.

Structure element	Type	Description
IP	DWORD	IP address of the communication partner
Port	UDINT	Port on the target computer
Status	ML_CONNECT_STATUS_JOB	Connection status ML_CONNECT_STATUS_JOB, page 731
StatusOld	ML_CONNECT_STATUS_JOB	Last connection status ML_CONNECT_STATUS_JOB, page 731
ActCntLostTlg	UDINT	Lost UDP packages
SeqNoRemote	UDINT	Sequence number of the remote host
SeqNoLocal	UDINT	Local sequence number

Structure element	Type	Description
TimeOutOfExchange	TIME	Connection timeout without data traffic
TimeOfLastExchange	TIME	Last data traffic
TimeOutUntilRepeat	TIME	Timeout before a UDP package is resent
TimeOfLastRepeat	TIME	Time of the last repeat
MaxNofRepeat	UDINT	Maximum repeats
ActCntRepeat	UDINT	Repeat counter
CtrlJobConnect	ITF_CTRLJOBCONNECT	Handle to the controlling end instance of a ML_CtrlJobConnect function block ML_SendRecvJobAss, page 711

Tab. 12-48: Elements of the ML_JOB_DBG_UDP_BUFF structure

12.5 SCL Protocol

12.5.1 Introduction and overview

The SCL (Secure Communication Layer) protocol forms a secure transport mechanism for so-called job telegrams. A job telegram is a data package of up to 255 bytes in size. UDP via Ethernet is used for the data exchange. The SCL protocol works connection-oriented, i.e. before transmitting data, a transport path has to be opened explicitly. Data transmission is assured, meaning that any packages that may have been lost are repeated. The number of repeats and the time period to repeat of the transmission can be set.

Specifications

The SCL protocol has been developed with the following specifications:

- All jobs must reach the communication partner (secured data transmission).
- The sequence of the jobs must be maintained.
- A transmission error must be recognized by the transmitter, however may not lead to blockage of the entire system.
- Faults or missing connections to a communication partner may not affect the communication to other communication partners.
- Guarantee of transmitted data by a checksum.

12.5.2 Sequence of communication

The follow description of the communication sequence is not necessary for the use of the SCL protocol. It serves only for the purpose of understanding the communication sequence should errors have to be analyzed.

Establishing a connection

A connection must be established before transmission of the data. There are two options for this:

1. Active establishment of a connection: an attempt is made to establish a connection with the specified communication partner. To obtain a connection, the communication partner must also actively or passively establish a connection.
2. Passive establishment of a connection: it waits on the active establishment of a connection by the communication partner. To obtain a connection, the communication partner must actively establish a connection.

The connection is established by a 3-way handshake. To do so the active communication partner sends a "SYN" package with a freely selected sequence number and the confirmation number 0.

The addressed communication partner responds with "SYN"|"ACK". A randomly selected number is entered as a sequence number. The confirmation number is the sequence number of the communication partner.

This package is then confirmed by the initiator of the connection with "ACK". The sequence number is 0 with this package and the confirmation number is the sequence number of the communication partner. Thereafter the connection is established and is available for data transmission. The connection status is set to "connected".

If the communication partner does not respond, then at the end of the waiting period, another attempt is made to establish a connection. If this still fails after the number of configured connection attempts, it is terminated.

If an error occurs, the "ERR" is set in the "Ctrl" field and the corresponding error code is entered in the "Error code" field.

If the communication partner reports an existing connection (error code 12), then the initiator of the new connection can reset the old connection by setting the "RST" flag in the "Ctrl" field without a controlled disconnection.

Disconnection

The connection is disconnected by a 3-way handshake. It is started by the "FIN" flag in the "Ctrl" field. The sequence number is the sequence number of the last data package sent increased by one. The confirmation number is 0.

The remote host responds with "FIN" and the "ACK" flag in the "Ctrl" field. The sequence number is the sequence number of the last data package sent increased by one. The confirmation number is the sequence number of the communication partner.

This package is then confirmed by the initiator of the disconnection with "ACK". The sequence number is 0 with this package and the confirmation number is the sequence number of the communication partner. The connection is then disconnected. No more data packages can be exchanged between these two communication partners. The connection status is set to "disconnected".

If the communication partner does not respond, then at the end of the waiting period, another attempt is made to disconnect the connection. If this still fails after the number of configured connection attempts, then it is terminated and the connection is set to "disconnected" without a controlled disconnection.

If an error occurs, the "ERR" is set in the "Ctrl" field and the corresponding error code is entered in the "Error code" field.

Data transfer

A data package is thus identified, which contains the value 0 in the "Ctrl" field. The sequence number is increased by one with each new data package. The confirmation number is always 0. The number of jobs to be sent is shown in the field "Number of data sets". The jobs are referenced by the offset entries of the offset table. An offset always counts from the start of the SCL header. The length of the jobs is determined by the difference between two successive offsets. The last offset points to the end of the SCL package.

A correctly received package is confirmed by the receiver by an "ACK" flag placed in the "Ctrl" field. The confirmation number is the sequence number of the received data package. The sequence number of the confirmation package is always 0.

If the transmitter does not receive confirmation within the configured time frame, then the package will be sent again. If no confirmation is received after the configured number of repeat attempts, then the connection will be as-

sumed to be interrupted. The connection status is set to "disconnected" in this case without a controlled disconnection.

If the received data package is faulty or if another error occurs, then the transmitter is notified by placing the "ERR" flag in the "Ctrl" field and entering the corresponding error code in the "Error code" field.

12.5.3 Structure of the UDP datagram

The following table presents the structure of a UDP telegram including additional protocol level without detailed data sets. The size of the overall frame (IP headers: 20 bytes + UDP header: 8 bytes + SCL header: 36 bytes + data) may not exceed 1500 bytes.

Proto- col	Offset	WORD	WORD
IP	0		Length
	4	Identification	Flags
	8	TTL	Header checksum
	12	Source address	
	16	Destination address	
UDP	0	Source port	Target port
	4	Length	Checksum

Protocol	Offset	WORD	WORD
SCL	0	Sequence number	
	4	Confirmation number	
	8	Ctrl Bit 0: SYN Bit 1: ACK Bit 2: FIN Bit 3: ERR Bit 4: RST all other bits are not used and are thus 0	Error codes of the SCL protocols , page 739
	12	Reserve	Reserve
	16	Time stamp (low)	
	20	Time stamp (high)	
	24	Reserve	Reserve
	28	Version number	Reserve
	32	Number of data sets	Reserve
	36	Offset data set 1	Offset data set 2
	40	Offset data set 3	Offset data set 4
	...	...	...
	...	Offset data set n	Offset after last data set
	Offset data set 1	Data set 1	
	Offset data set 2	Data set 2	
	...	...	
	Offset data set n	Data set n	

Tab. 12-49: Structure of an SCL data package via Ethernet

12.5.4 Error codes of the SCL protocols

The following table provides an overview of the SCL field "Error code".

Code	Description
0	Check that telegram has been completed successfully
1	The received telegram has an invalid telegram length
2	The received telegram contains an incompatible telegram version
3	The reserve fields are not 0
4	The SCL element "Ctrl" contains invalid control bits
5	The SCL element "Error Code" contradicts the "ERR" control bit of the SCL element "Ctrl"
6	The SCL element "Ctrl" contradicts the SCL element "Number of data sets".

Code	Description
7	The SCL element "Number of data sets" with the value 0 contradicts the resulting minimum telegram length
8	The SCL element "Number of data sets" contradicts the resulting minimum telegram length
9	The SCL element "Offset data set x" is greater than the SCL element "Offset data set x+1".
10	The SCL element "Offset data set n+1" (last element) does not agree with the telegram length
11	The element "Sequence Number" does not match the expected value
12	There is already a connection to the device with the requested IP address No further connection can be established to the same IP address
13	No further connection can be established as no further resources are available
14	Not used
15	The element "Sequence Number" and / or "Confirmation Number" does not match the expected value
16	Not used
17	Not used

Tab. 12-50: SCL protocol error codes

13 Service and support

Our worldwide service network provides an optimized and efficient support. Our experts offer you advice and assistance should you have any queries. You can contact us **24/7**.

Service Germany Our technology-oriented Competence Center in Lohr, Germany, is responsible for all your service-related queries for electric drive and controls.

Contact the **Service Hotline** and **Service Helpdesk** under:

Phone:	+49 9352 40 5060
Fax:	+49 9352 18 4941
E-mail:	service.svc@boschrexroth.de
Internet:	http://www.boschrexroth.com

Additional information on service, repair (e.g. delivery addresses) and training can be found on our internet sites.

Service worldwide Outside Germany, please contact your local service office first. For hotline numbers, refer to the sales office addresses on the internet.

Preparing information To be able to help you more quickly and efficiently, please have the following information ready:

- Detailed description of malfunction and circumstances
- Type plate specifications of the affected products, in particular type codes and serial numbers
- Your contact data (phone and fax number as well as your e-mail address)

Index

A

Abbreviations.....	15
ANSI Z535.6-2006.....	7

B

Backing up and restoring drive parameters.....	33
Bell crank kinematics.....	382

C

Cam tables	
Parameters in MLC and MLD.....	633
CamLock.....	156
Application example.....	156
CamLock Function Blocks	
MB_CamLock.....	161
MB_PrepareCams.....	159
Closed-loop dancer position control.....	477
Commissioning	
Dancer position controller.....	492
Diameter calculator.....	502
Diameter calculator with closed-loop tension control.....	540
Diameter calculator with dancer.....	523
Diameter calculator with load cell.....	540
Diameter calculator without sensor.....	540
Complaints.....	15
Crank kinematics.....	405
Mechanical translatory position.....	409
Single-axis mode.....	406
Synchronous Operation with Cam.....	406
Variables and counting directions.....	407
Virtual translatory position.....	409
Criticism.....	15
CrossCutter.....	614
CrossSealer.....	612
Customer Feedback.....	15
Cycle-synchronous lock-on and lock-off	156
Cyclic data channels.....	139

D

Dancer position controller	
Commissioning.....	492
Decoupling network.....	464
Detecting an externally measured diameter.....	503
Diameter calculation, winder.....	494
Diameter calculator with dancer.....	508
Command winding axis.....	523
Commissioning.....	523
Diameter calculator with open-loop and closed loop tension control	
Command winding axis.....	556
Diameter calculator with open-loop and closed-loop tension control.....	524
Commissioning.....	540
Drive parameters	

Backup.....	33, 39
Restore.....	36, 42

E

Error codes	
IL_ParameterChannel.....	693
IL_ParameterChannel, parameter channel.....	694
IL_RegiQualityType01.....	308
MB_AbortTrigger.....	139
MB_AllocCyclicParameter.....	143
MB_AxisDistanceAccumulator.....	49
MB_BellCrankData.....	388
MB_CalcInertiaLimitType02.....	561
MB_CamTableBellCrank.....	393
MB_CamTableBellCrankSuperimposed.....	398
MB_CamTableCrank.....	414
MB_CamTableCrankSuperimposed.....	419
MB_CoordinatePairCamType01.....	646
MB_CrossCutSealType01.....	619
MB_CrossCutterCalcType04.....	626
MB_DancerCalcControlParamType01.....	593
MB_DancerControlType03.....	484
MB_DancerControlType04.....	490
MB_DiameterCalculatorType03.....	500
MB_DiameterMeasurementType01.....	506
MB_DigitalCamSwitchOvs.....	89
MB_DistanceAccumulator.....	48
MB_FeedMovementCalcType01.....	652
MB_FeedMovementCalcType02.....	658
MB_FreeCyclicParameter.....	145
MB_GetCyclicChannelConfig.....	146
MB_GetCyclicParameterHandle.....	148
MB_GetLastCyclicParameterError.....	151
MB_GetSystemInfo.....	30
MB_InitTouchProbe.....	127
MB_LoadCamData.....	97
MB_MagicBeltType01.....	189
MB_PhiToXvirt.....	421
MB_PhiToYvirt.....	400
MB_PilgrimStepCalcType01.....	260
MB_PositionFilterType01.....	100
MB_PositionMonitoring_GantryCant.....	102
MB_PositionMonitoring_PosCollision.....	105
MB_PrepareCams.....	161
MB_PrintLengthCorrCalcType01.....	265
MB_ReachVelocityType02.....	110
MB_RegiMeasuringType01.....	298
MB_RegiMeasuringType02.....	301
MB_RegiMeasuringType03.....	305
MB_RegiRegulateType01.....	313
MB_RegiRegulateType02.....	321
MB_RegiRegulateType03.....	327
MB_RegiRegulateType04.....	335
MB_RegiSetpointLockType01.....	338

Index

MB_RegisterControllerSideType01.....	373	Helpdesk.....	741
MB_RegisterControllerType02.....	347	Hotline.....	741
MB_RegisterControllerType04.....	359		
MB_SlotterCalcType01.....	676	I	
MB_SlotterCalcType02.....	686	IL_ParameterChannel.....	687
MB_SmartBeltType01.....	233	Commissioning of the Function Block.....	701
MB_TensionControlLoadCellMatPar- Type01.....	292	Control word/Status word.....	694
MB_TensionControlLoadCellMatPar- Type02.....	462	Error Telegrams.....	698
MB_TensionControlLoadCellType01.....	278	Index and Subindex.....	696
MB_TensionControlLoadCellType02.....	448	Initializing the Communication.....	698
MB_TensionDecouplingNetworkType01.....	471	Parameter channel in the cyclic data channel.....	694
MB_TensionIdentModelParamType01.....	432	IL_REGI_QUALITY.....	308
MB_TouchProbe.....	131	IL_RegiQualityType01.....	305
MB_TouchProbeContinuous.....	134	IMC interface.....	19
MB_TraverseControlType01.....	589		
MB_UnwindMaterialType02.....	566	L	
MB_WinderDancerCtrlType01.....	514	Libraries	
MB_WinderDancerCtrlType02.....	521	RIL_ParameterChannel.....	687
MB_WinderTensionCtrlSpeedType01.....	553	Library	
MB_WinderTensionCtrlType02.....	537	ML_JobComm.....	709
MB_WindSpeedControlAdaptionType01.....	573	ML_TechBase.....	17
MB_WindTaperProfileType01.....	570	ML_TechCrank.....	381
MB_WriteExpectWindow.....	137	ML_TechMotion.....	155
MB_XvirtToXmech.....	423	ML_TechMotionBase.....	31
MB_YvirtToPhi.....	402	ML_TechRegi.....	295
MB_YvirtToPhiPoly5.....	404	ML_TechTensionAdvanced.....	425
MB_YvirtToYmech.....	390	MX_TechBase.....	17
MC_DigitalCamSwitch.....	64	MX_TechMotion.....	155
ML_AxisBackup.....	35	MX_TechMotionBase.....	31
ML_AxisBackup_Type02.....	41	MX_TechRegi.lib.....	295
ML_AxisRestore.....	39	MX_TechWinder.lib.....	475
ML_AxisRestore_Type02.....	45	MX_TechWinder.library.....	475
ML_CtrlJobConnect.....	722	RMB_TechCam.....	639
ML_DebugJobConnect.....	727	RMB_TechCrossCutCrossSeal.....	605
ML_DebugJobConnect_Cyclic.....	730	RMB_TechWinder.....	475
ML_FlyingShear.....	176		
ML_RecvJobTlg.....	719	M	
ML_SagControlC.....	222	MagicBelt.....	180
ML_SendJobTlg.....	716	Automatic mode.....	204
ML_SendRecvJobAss.....	714	Jog mode.....	203
ML_ShowJobConnect.....	724	Operation modes.....	203
MX_FlyingShear.....	176	Tune mode.....	204
Error codes,		MB_AbortTrigger.....	137
MB_CrossCutterCalcType05.....	633	MB_AllocCyclicParameter.....	141
		MB_AxisDistanceAccumulator.....	48
F		MB_BellCrankData.....	386
Feedback.....	15	MB_CalcInertiaLimitType02.....	559
Filtering incremental positions.....	97	Required parameterization.....	561
FlyingShear.....	168	MB_CAM_TABLE_DATA.....	633
		MB_CamLock.....	161
G		MB_CamLock Parameterization.....	168
Gear setting - calculation.....	624	MB_CAMPROF.....	97
		MB_CAMPROF_SLOTTER.....	686
H		MB_CAMSWITCH_REF_OVS.....	92
Hazard warnings.....	7	MB_CAMTABLE.....	97
		MB_CamTableBellCrank.....	390

MB_CamTableBellCrankSuperimposed.....	394	MB_LoadCamData.....	94
MB_CamTableCrank.....	410	MB_MagicBeltType01.....	182
MB_CamTableCrankSuperimposed.....	415	Commissioning notes.....	218
MB_COORDINATE_PAIR_CAM_SELECT.....	646	MB_MasterToPhi.....	421
MB_COORDINATE_PAIR_TYPE01.....	646	MB_MBELT_FEEDTYPE.....	197
MB_CoordinatePairCamType01.....	639	MB_MBELT_GENERAL.....	189
MB_CORR_PROFILE.....	635	MB_MBELT_INFEED.....	194
MB_CORRECTION_MODE.....	335	MB_MBELT_INFEEDSTEPS.....	197
MB_CRANK.....	414	MB_MBELT_MODEAUTO.....	193
MB_CrossCutSealType01.....	605	MB_MBELT_MODEJOG.....	191
CrossCutter use case.....	614	MB_MBELT_MODES.....	189
CrossSealer use case.....	612	MB_MBELT_MODETUNE.....	192
MB_CrossCutterCalcType04.....	619	MB_MBELT_MOVE.....	200
Special features of MLC.....	626	MB_MBELT_OUTFEED.....	199
MB_CrossCutterCalcType05.....	627	MB_MBELT_STARTTYPE.....	193
Special features for the MLC.....	633	MB_MBELT_STATUS.....	201
Special features for the MLD.....	633	MB_MBELT_STOPTYPE.....	194
MB_CURVETYPE.....	658	MB_PhiToXvirt.....	420
MB_CYCLIC_CHANNEL_CONFIG.....	153	MB_PhiToYvirt.....	398
MB_CYCLIC_CHANNELS.....	152	MB_PilgrimStepCalcType01.....	250
MB_CYCLIC_PARAM_REF.....	151	MB_POS_MON_GANTRY_CANT.....	102
MB_CYCLIC_TYPES.....	152	MB_POS_MON_POS_COLLISION.....	105
MB_DANCER_CONTROL_MODE.....	602	MB_PositionFilterType01.....	98
MB_DancerCalcControlParamType01.....	590	MB_PositionMonitoring_GantryCant.....	101
MB_DancerControlType03.....	479	MB_PositionMonitoring_PosCollision.....	103
Special features for the MLC.....	485	MB_PrepareCams.....	159
Special features for the MLD.....	485	MB_PrintLengthCorrCalcType01.....	260
MB_DancerControlType04.....	485	MB_PROBE_EXECUTION_MODE.....	128
Special features for the MLC.....	491	MB_PUSH_OUT_CONFIG.....	636
Special features for the MLD.....	491	MB_RAMPING_MODE.....	335
MB_DiameterCalculatorType03.....	494	MB_ReachVelocityType02.....	106
Special features for the MLC.....	501	MB_ReadCyclicParameter.....	148
Special features for the MLD.....	501	MB_ReadCyclicRealParameter.....	149
MB_DiameterMeasurementType01.....	503	MB_RegiMeasuringType01.....	296
Special features for the MLC.....	507	MB_RegiMeasuringType02.....	298
Special features for the MLD.....	507	MB_RegiMeasuringType03.....	301
MB_DigitalCamSwitchOvs.....	66	MB_RegiRegulateType01.....	308
MB_DistanceAccumulator.....	47	MB_RegiRegulateType02.....	313
MB_FeedMovementCalcType01.....	647	MB_RegiRegulateType03.....	321
MB_FeedMovementCalcType02.....	652	MB_RegiRegulateType04.....	327
MB_FreeCyclicParameter.....	143	MB_RegiSetpointLockType01.....	336
MB_FS_RETAIN_DATA.....	176	MB_RegisterControllerSideType01.....	366
MB_GetCyclicChannelConfig.....	145	MB_RegisterControllerType02.....	338
MB_GetCyclicParameterHandle.....	146	MB_RegisterControllerType04.....	347
MB_GetLastCyclicParameterError.....	150	MB_RESOLUTION.....	637
MB_GetSystemInfo.....	27	MB_SlotterCalcType01.....	664
MB_IMC_ADMIN_STATUS.....	21	MB_SlotterCalcType02.....	676
MB_IMC_ADMINISTRATION.....	20	MB_SMARTBELT_STATISTICS.....	233
MB_IMC_CONTROL_MODE_STATUS.....	22	MB_SmartBeltType01.....	227
MB_IMC_CONTROL_TYPE01.....	20	MB_SYSTEM_TYPE.....	30
MB_IMC_DIAGNOSIS.....	23	MB_TENSION_CONTROL_ADJUST.....	463
MB_IMC_ERROR_SEVERITY.....	23	MB_TENSION_CONTROL_CONFIG.....	462
MB_IMC_MODE_CONTROL.....	21	MB_TENSION_CONTROL_MATERIAL.....	292
MB_IMC_MODE_STATUS.....	22	MB_TENSION_CONTROL_MATERI- AL_TYPE.....	292
MB_IMC_SERCOS_PHASE.....	22	MB_TENSION_CONTROL_MODE.....	293
MB_IMC_STATUS_TYPE01.....	21	MB_TENSION_DECOUPLING_AXIS_VAL- UE.....	472
MB_IMCInterface.....	19		
MB_InitTouchProbe.....	120		

Index

MB_TENSION_DECOUPLING_CONTROL- LER_VALUE.....	471	MB_YvirtToYmech.....	388
MB_TENSION_IDENT_EXTENDED_SET- TINGS.....	432	MC_CAMSWITCH_REF.....	65
MB_TENSION_PRE_CTRL.....	463	MC_DigitalCamSwitch.....	50
MB_TensionControlLoadCellMatParType01.....	279	MC_OUTPUT_REF.....	66
MB_TensionControlLoadCellMatParType02.....	448	MC_TRACK_REF.....	65
MB_TensionControlLoadCellType01.....	266	Measure diameter externally.....	503
MB_TensionControlLoadCellType02.....	433	Measurement encoder support.....	46
MB_TensionDecouplingNetworkType01.....	464	Measuring function blocks for register con- trollers.....	296
MB_TensionIdentModelParamType01.....	425	ML_AxisBackup.....	33
MB_TouchProbe.....	128	ML_AxisBackup_Type02.....	39
MB_TouchProbeContinuous.....	131	ML_AxisRestore.....	36
MB_TRACK_REF_OVS.....	93	ML_AxisRestore_Type02.....	42
MB_TRAVERSE_ADAPTION.....	594	ML_BACKUP_ERRORS.....	46
MB_TRAVERSE_DIRECTION.....	595	ML_BACKUP_OPTIONS.....	45
MB_TRAVERSE_MODE_LAYER_COUN- TER.....	595	ML_BACKUP_PARAMS.....	46
MB_TRAVERSE_UNWIND.....	595	ML_BASIC_CAMSWITCH_REF.....	65
MB_TRAVERSE_UNWIND_CONTROL.....	596	ML_BASIC_CAMSWITCH_REF_OVS.....	93
MB_TRAVERSE_UNWIND_MODE.....	597	ML_BASIC_TRACK_REF.....	66
MB_TraverseControlType01.....	574	ML_BASIC_TRACK_REF_OVS.....	93
Special features for the MLC.....	590	ML_CFG_CONNECT_JOB.....	730
MB_UnwindMaterialType02.....	562	ML_CFG_JUNK_JOB.....	730
Required parameterization.....	566	ML_CFG_SEGMENT_JOB.....	731
MB_WIND_CTRL_TYPE.....	597	ML_CFG_SEND_RECV_JOB.....	731
MB_WIND_DANCER_LIMITS.....	597	ML_CONNECT_CMD_JOB.....	731
MB_WIND_DIAM_CALC_MODE.....	598	ML_CONNECT_STATUS_JOB.....	731
MB_WIND_DIAMETER.....	598	ML_CtrlJobConnect.....	719
MB_WIND_DIRECTION.....	600	ML_DebugJobConnect.....	724
MB_WIND_FRICTION.....	600	ML_DebugJobConnect_Cyclic.....	727
MB_WIND_TAPER_PROFILE.....	601	ML_DegToInc.....	24
MB_WIND_TENS_DIAG.....	601	ML_DegToRad.....	24
MB_WIND_TENSION.....	601	ML_FlyingShear.....	169
MB_WIND_TENSION_MODE.....	601	Special features of MLC.....	176
MB_WIND_TORQUE_LIMITS.....	602	ML_IncToDeg.....	25
MB_WinderDancerCtrlType01.....	509	ML_IncToRad.....	25
Special features for the MLC.....	515	ML_JOB_ASS_BUFF.....	732
Special features for the MLD.....	515	ML_JOB_ASS_BUFF_ADDR.....	732
MB_WinderDancerCtrlType02.....	516	ML_JOB_ASS_BUFF_DATA.....	732
Special features for the MLC.....	522	ML_JOB_ASS_BUFF_SCL.....	733
Special features for the MLD.....	522	ML_JOB_BUFF.....	733
MB_WinderTensionCtrlSpeedType01.....	544	ML_JOB_BUFF_DATA.....	733
Special features for the MLC.....	554	ML_JOB_BUFF_MANAGEMENT.....	734
Special features for the MLD.....	279, 554	ML_JOB_DBG_JOB_ASS_BUFF.....	734
MB_WinderTensionCtrlType02.....	526	ML_JOB_DBG_JOB_BUFF.....	734
Special features for the MLC.....	538	ML_JOB_DBG_TBL_CONNECT.....	734
Special features for the MLD.....	539	ML_JOB_DBG_UDP_BUFF.....	735
MB_WindSpeedControlAdaptionType01.....	570	ML_JobComm.....	
Special features for the MLC.....	574	Error codes of the SCL protocols.....	739
MB_WindTaperProfileType01.....	567	ML_JobComm.library.....	709
MB_WriteCyclicParameter.....	149	ML_LINK_SEND_RECV_JOB.....	735
MB_WriteCyclicRealParameter.....	150	ML_RadToDeg.....	26
MB_WriteExpectWindow.....	134	ML_RadToInc.....	27
MB_XvirtToXmech.....	422	ML_RecvJobTlg.....	716
MB_YvirtToPhi.....	400	ML_SagControlA.....	223
MB_YvirtToPhiPoly5.....	402	ML_SagControlC.....	219
MB_YVirtToWorkRange.....	404	ML_SendJobTlg.....	714
		ML_SendRecvJobAss.....	711
		ML_ShowJobConnect.....	722

ML_TBL_CONNECT_JOB.....	735	Supplementary function blocks.....	559
ML_TechBase.library.....	17	Winder with closed-loop tension control and speed correction.....	544
ML_TechCrank.library.....	381	RMB_TechWinder.library.....	475
ML_TechMotion.library.....	155	S	
ML_TechMotionBase.library.....	31	S curve, programming motion sequences.....	106
ML_TechRegi.library.....	295	Safety instructions.....	7
ML_TechTensionAdvanced.library.....	425	Sag control.....	218
MX_FlyingShear.....	169	Service hotline.....	741
Components and Parameterization.....	177	Side register	
MX_TechCrossCutCrossSeal.....	605	Configuration.....	373
MX_TechMotion.....	155	Side register controller function block.....	366
MX_TechRegi.lib.....	295	Signal alert symbol.....	7
MX_TechWinder		Signal words.....	7
Closed-loop dancer position control.....	477	Sloters.....	659
Data types.....	594	SmartBelt	
Diameter calculation	494	Accumulating conveyor.....	235
Diameter calculator with dancer.....	508	Calculate the ProbeSetpoint.....	245
Diameter calculator with open-loop and closed-loop tension control.....	524	Calculation of the SmartBeltPitch.....	243
Parameter overview.....	602	Common mistakes and misunderstand- ings	249
Supplementary function blocks.....	559	Degree of order.....	236
Winder with closed-loop tension control and speed correction.....	544	Drive parameter settings.....	240
MX_TechWinder.lib.....	475	Drive requirements.....	239
MX_TechWinder.library.....	475	Friction.....	237
O		PLC programming.....	242
Operation modes, MagicBelt.....	203	Project planning.....	238
Overview on libraries and target systems.....	2	Requirements for belt and machine.....	238
P		SmartBelt limitations and behavior.....	246
Position monitoring.....	100	Tips and tricks.....	247
Probe		Use.....	234
Overview.....	110	Special features for the MLD.....	3
Programmable limit switches.....	49	Suggestions.....	15
R		Support.....	741
Register controller function block.....	338	Symbols.....	9
Register controllers		System information.....	27
Example of punching in label printing.....	363	T	
Use case for a flow wrapper.....	365	Tension controller.....	265
Use case for insetter control.....	364	Terms and abbreviations.....	15
Register quality function block.....	305	Touch probe	
Register setpoint lock function block.....	336	Components.....	111
Regulating function blocks for register con- trollers.....	308	Parameterization.....	112
RIL_ParameterChannel.library.....	687	Trigonometric functions.....	23
RMB_TechCam.library.....	639	V	
RMB_TechCrosscutCrossseal.library.....	605	Validity of the documentation.....	1
RMB_TechWinder		W	
Closed-loop dancer position control.....	477	Warnings.....	7
Data types.....	594	Winder, diameter calculation	
Diameter calculation.....	494	Commissioning.....	502
Diameter calculator with dancer.....	508		
Diameter calculator with open-loop and closed-loop tension control.....	524		
Parameter overview.....	602		

Notes

Notes

Bosch Rexroth AG

Electric Drives and Controls

P.O. Box 13 57

97803 Lohr, Germany

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr, Germany

Phone +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911341511