

Rexroth IndraMotion MLC 13VRS

Technology Libraries

Library
R911336324

Edition 04



Title Rexroth IndraMotion
MLC 13VRS
Technology Libraries

Type of Documentation Library

Document Typecode DOK-MLC***-TF*LIB**V13-LI04-EN-P

Internal File Reference RS-9cf62c1826f3d2570a6846a5011461e6-3-en-US-3

Change Record

Edition	Release Date	Note
Edition 01	2012-03	First edition for MLC/XLC 13VRS
Edition 02	2013-04	Update
Edition 03	2013-07	Update, RMB_Winder
Edition 04	2015-02	Revision: ML_TechBase, RIL_ParameterChannel, RMB_Winder, RMB_TechCam, ML_TechMotion

Copyright © Bosch Rexroth AG 2015

This document, as well as the data, specifications and other information set forth in it, are the exclusive property of Bosch Rexroth AG. It may not be reproduced or given to third parties without its consent.

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 HBU (vha/MePe)

Table of Contents

	Page
1 About this documentation.....	9
1.1 Validity of the documentation.....	9
1.2 Documentation structure.....	9
1.3 Using safety instructions.....	10
1.3.1 Safety instructions – Structure.....	10
1.3.2 Explaining signal words and safety alert symbol.....	10
1.3.3 Symbols used.....	11
1.4 Required and supplementing documentations.....	11
1.5 Terms and abbreviations.....	17
2 Overview.....	19
2.1 Libraries and target systems.....	19
2.2 Special Features of the IndraMotion MLD.....	19
2.2.1 General Information.....	19
2.2.2 Drive Firmware, Function Package.....	19
3 ML_TechBase.library.....	21
3.1 Introduction and overview.....	21
3.2 Jogging variables (adjustment).....	23
3.2.1 Introduction and overview.....	23
3.2.2 MB_ContinuousAdjustType01.....	26
3.2.3 MB_ContinuousAdjustType02.....	30
3.2.4 MB_IncrementalAdjustType01.....	33
3.3 Backing up and restoring drive parameters (AxisBackupRestore).....	35
3.3.1 Introduction and overview.....	35
3.3.2 ML_AxisBackup.....	35
3.3.3 ML_AxisRestore.....	37
3.3.4 ML_BACKUP_OPTIONS.....	40
3.3.5 ML_BACKUP_PARAMS.....	40
3.4 Two-Point Controllers (Controller).....	41
3.4.1 ML_TwoPosCtrlType01.....	41
3.5 Cam download (LoadCamData).....	42
3.5.1 Introduction and overview.....	42
3.5.2 MB_LoadCamData.....	43
3.5.3 MB_CAMPROF.....	45
3.5.4 MB_CAMTABLE.....	46
3.6 Mathematical functions (MathematicalFunctions).....	46
3.6.1 Introduction and overview.....	46
3.6.2 ML_InterpolationLinear.....	46
3.6.3 ML_MaxValue.....	48
3.6.4 ML_COORD_PAIR_TYPE01.....	49
3.7 Programmable limit switch (PhaseLimitSwitch).....	49
3.7.1 Introduction and Overview.....	49

Table of Contents

	Page
3.7.2 MC_DigitalCamSwitch.....	49
3.7.3 MC_CAMSWITCH_REF with ML_BASIC_CAMSWITCH_REF.....	65
3.7.4 MC_OUTPUT_REF.....	65
3.7.5 MC_TRACK_REF with ML_BASIC_TRACK_REF.....	66
3.8 TemperatureController.....	66
3.8.1 Introduction and overview.....	66
3.8.2 IL_TempControlType01.....	66
3.8.3 IL_TempControlType02	84
3.8.4 IL_TEMPCONTROL_CONTR_PARAM.....	93
3.8.5 IL_TEMPCONTROL_CONTROL_MODE.....	94
3.8.6 IL_TEMPCONTROL_EXTENDED_SETTINGS.....	94
3.8.7 IL_TEMPCONTROL_LIMITS.....	95
3.8.8 IL_TEMPCONTROL_MODEL_PARAM.....	95
3.9 Touch probe function blocks (TouchProbe).....	96
3.9.1 Introduction and overview.....	96
3.9.2 Touch Probe Function Blocks - Components and Parameterization.....	98
3.9.3 MB_InitTouchProbe.....	106
3.9.4 MB_TouchProbe.....	114
3.9.5 MB_TouchProbeContinuous.....	117
3.9.6 MB_WriteExpectWindow.....	120
3.9.7 MB_AbortTrigger.....	123
3.9.8 MB_PROBE_EXECUTION_MODE.....	125
3.10 Trigonometric functions (Trigonometric Functions).....	125
3.10.1 Introduction and overview.....	125
3.10.2 ML_DegToInc.....	125
3.10.3 ML_DegToRad.....	126
3.10.4 ML_IncToDeg.....	126
3.10.5 ML_IncToRad.....	127
3.10.6 ML_RadToDeg.....	127
3.10.7 ML_RadToInc.....	128
3.11 Cyclic data channels, administration (Utilities\CyclicData).....	129
3.11.1 Introduction and overview.....	129
3.11.2 MB_AllocCyclicParameter.....	130
3.11.3 MB_FreeCyclicParameter.....	132
3.11.4 MB_GetCyclicChannelConfig.....	134
3.11.5 MB_GetCyclicParameterHandle.....	135
3.11.6 MB_ReadCyclicParameter.....	137
3.11.7 MB_ReadCyclicRealParameter.....	137
3.11.8 MB_WriteCyclicParameter.....	138
3.11.9 MB_WriteCyclicRealParameter.....	138
3.11.10 MB_GetLastCyclicParameterError.....	139
3.11.11 MB_CYCLIC_PARAM_REF.....	140
3.11.12 MB_CYCLIC_TYPES.....	140
3.11.13 MB_CYCLIC_CHANNELS.....	141
3.11.14 MB_CYCLIC_CHANNEL_CONFIG.....	141
3.12 Logic applications (Utilities\Logic).....	142

Table of Contents

	Page
3.12.1 Introduction and overview.....	142
3.12.2 IL_LinkedList.....	142
3.12.3 IL_LINKED_LIST_MAINTENANCE.....	146
3.12.4 IL_LINKED_LIST_MODE.....	146
3.13 Modulo values (Utilities\Modulo).....	147
3.13.1 Introduction and overview.....	147
3.13.2 MB_GetModuloType01.....	147
3.13.3 MB_GetModuloType02.....	148
3.13.4 MB_GetModuloType03.....	149
3.13.5 MB_GetModuloType04.....	150
3.14 System information (Utilities\SystemInfo).....	152
3.14.1 MB_GetSystemInfo.....	152
3.14.2 MB_SYSTEM_TYPE.....	154
4 ML_TechMotion.library.....	155
4.1 Introduction and overview.....	155
4.2 Function blocks for 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 Distance accumulator (DistanceAccumulator).....	168
4.3.1 Introduction and overview.....	168
4.3.2 MB_DistanceAccumulator.....	169
4.3.3 MB_AxisDistanceAccumulator.....	170
4.4 Flying Shear (FlyingShear).....	171
4.4.1 Introduction and overview.....	171
4.4.2 ML_FlyingShear/MX_FlyingShear.....	171
4.4.3 MB_FS_RETAIN_DATA.....	178
4.4.4 ML_FlyingShear - Special Features of IndraMotion MLC.....	178
4.4.5 MX_FlyingShear - Special Features for the IndraMotion MLD.....	180
4.5 Magic belt (MagicBelt).....	182
4.5.1 Introduction and overview.....	182
4.5.2 MB_MagicBeltType01.....	184
4.5.3 MB_MBELT_MODES.....	206
4.5.4 MB_MBELT_GENERAL.....	206
4.5.5 MB_MBELT_MODEJOG.....	209
4.5.6 MB_MBELT_MODETUNE.....	209
4.5.7 MB_MBELT_MODEAUTO.....	210
4.5.8 MB_MBELT_STARTTYPE.....	211
4.5.9 MB_MBELT_STOPTYPE.....	211
4.5.10 MB_MBELT_INFEED.....	211
4.5.11 MB_MBELT_FEEDTYPE.....	214
4.5.12 MB_MBELT_INFEEDSTEPS.....	214
4.5.13 MB_MBELT_OUTFEED.....	216

Table of Contents

	Page
4.5.14 MB_MBELT_MOVE.....	217
4.5.15 MB_MBELT_STATUS.....	218
4.6 Position monitoring (PositionMonitoring).....	220
4.6.1 Introduction and overview.....	220
4.6.2 MB_PositionMonitoring_GantryCant.....	220
4.6.3 MB_POS_MON_GANTRY_CANT.....	222
4.6.4 MB_PositionMonitoring_PosCollision.....	222
4.6.5 MB_POS_MON_POS_COLLISION.....	224
4.7 Sag controller (SagControl).....	225
4.7.1 Introduction and Overview.....	225
4.7.2 ML_SagControlC.....	226
4.7.3 ML_SagControlA.....	229
4.8 S curve, programming motion sequences (SCurve).....	233
4.8.1 Introduction and overview.....	233
4.8.2 MB_ReachVelocityType02.....	233
4.9 SmartBelt [feed machine (SmartBelt)].....	237
4.9.1 MB_SmartBeltType01.....	237
4.9.2 MB_SMARTBELT_STATISTICS.....	243
4.9.3 Introduction to SmartBelt Usage.....	244
4.9.4 Project Planning.....	248
4.9.5 SmartBelt Limitations and Behavior.....	256
4.9.6 Tips and Tricks.....	257
4.9.7 Common Mistakes and Misunderstandings.....	259
4.10 Cam tables for technology functions (TechFuncCamTable).....	260
4.10.1 Introduction and overview.....	260
4.10.2 MB_PilgrimStepCalcType01.....	260
4.10.3 MB_PrintLengthCorrCalcType01.....	271
4.11 Tension controller (TensionControl).....	275
4.11.1 Introduction and overview.....	275
4.11.2 MB_TensionControlLoadCellType01.....	276
4.11.3 MB_TensionControlLoadCellMatParType01.....	287
4.11.4 MB_TENSION_CONTROL_MATERIAL.....	300
4.11.5 MB_TENSION_CONTROL_MATERIAL_TYPE.....	300
4.11.6 MB_TENSION_CONTROL_MODE.....	301
5 ML_TechRegi.library/MX_TechRegi.lib.....	303
5.1 Introduction and overview.....	303
5.2 Measuring function blocks for register controllers.....	304
5.2.1 Introduction and overview.....	304
5.2.2 MB_RegiMeasuringType01.....	304
5.2.3 MB_RegiMeasuringType02.....	306
5.2.4 MB_RegiMeasuringType03.....	310
5.3 Register quality function block.....	314
5.3.1 Introduction and overview.....	314
5.3.2 IL_RegiQualityType01.....	314
5.3.3 IL_REGI_QUALITY.....	316

Table of Contents

	Page
5.4	Regulating function blocks for register controllers..... 317
5.4.1	Introduction and overview..... 317
5.4.2	MB_RegiRegulateType01..... 317
5.4.3	MB_RegiRegulateType02..... 321
5.4.4	MB_RegiRegulateType03..... 329
5.4.5	MB_RegiRegulateType04..... 335
5.4.6	MB_CORRECTION_MODE..... 343
5.4.7	MB_RAMPING_MODE..... 344
5.5	Register setpoint lock function block..... 344
5.5.1	Introduction and overview..... 344
5.5.2	MB_SetpointLockType01..... 344
5.6	Register Controller Function Block..... 346
5.6.1	Introduction and overview..... 346
5.6.2	MB_RegisterControllerType02..... 346
5.6.3	MB_RegisterControllerType04..... 356
5.6.4	Configuring the Register Controller..... 368
5.6.5	Register Controller – Uses Cases..... 372
5.7	Side Register Controller Function Block..... 375
5.7.1	Introduction and overview..... 375
5.7.2	MB_RegisterControllerSideType01..... 375
5.7.3	Configuring the Side Register..... 382
6	ML_TechCrank.library..... 389
6.1	Introduction and overview..... 389
6.2	Functions and function blocks for the bell crank kinematics (BellCrank)..... 390
6.2.1	Introduction and Overview..... 390
6.2.2	General Definitions..... 391
6.2.3	MB_BellCrankData..... 394
6.2.4	MB_YVirtToWorkRange..... 396
6.2.5	MB_CamTableBellCrank..... 397
6.2.6	MB_CamTableBellCrankSuperimposed..... 400
6.2.7	MB_PhiToYvirt..... 405
6.2.8	MB_YvirtToPhi..... 407
6.2.9	MB_YvirtToPhiPoly5..... 409
6.2.10	MB_YvirtToYmech..... 411
6.3	Functions and function blocks for the crank kinematics (Crank)..... 413
6.3.1	Introduction and Overview..... 413
6.3.2	General Definitions..... 415
6.3.3	MB_CRANK..... 418
6.3.4	MB_CamTableCrank..... 418
6.3.5	MB_CamTableCrankSuperimposed..... 422
6.3.6	MB_MasterToPhi..... 427
6.3.7	MB_PhiToXvirt..... 428
6.3.8	MB_XvirtToXmech..... 429

Table of Contents

	Page
7 ML_TechTensionAdvanced.library.....	431
7.1 Introduction and overview.....	431
7.2 MB_TensionControlLoadCellType02.....	431
7.3 MB_TensionControlLoadCellMatParType02.....	445
7.4 MB_TENSION_CONTROL_CONFIG.....	459
7.5 MB_TENSION_CONTROL_ADJUST.....	460
7.6 MB_TENSION_PRE_CTRL.....	460
7.7 MB_TensionDecouplingNetworkType01.....	461
7.8 MB_TENSION_DECOUPLING_CONTROLLER_VALUE.....	468
7.9 MB_TENSION_DECOUPLING_AXIS_VALUE.....	469
 8 RMB_TechWinder.library/MX_TechWinder.lib.....	 471
8.1 Introduction and overview.....	471
8.2 Closed-loop dancer position control.....	473
8.2.1 Introduction and overview.....	473
8.2.2 MB_DancerControlType03	475
8.2.3 MB_DancerControlType04.....	481
8.2.4 Commissioning of a closed-loop dancer position control.....	487
8.3 Function blocks for the diameter calculation	489
8.3.1 Introduction and overview.....	489
8.3.2 MB_DiameterCalculatorType03.....	489
8.3.3 Commissioning the diameter calculator	497
8.4 Function block to detect an externally measured diameter	498
8.4.1 Introduction and overview.....	498
8.4.2 MB_DiameterMeasurementType01.....	498
8.5 Diameter calculator with dancer	502
8.5.1 Introduction and Overview.....	502
8.5.2 MB_WinderDancerCtrlType01.....	503
8.5.3 MB_WinderDancerCtrlType02.....	512
8.5.4 Commissioning the diameter calculator with dancer	519
8.6 Diameter calculator with open-loop tension control or closed-loop tension control.....	521
8.6.1 Introduction and overview.....	521
8.6.2 MB_WinderTensionCtrlType02.....	522
8.6.3 Commissioning the diameter calculator with open-loop tension control and closed-loop tension control.....	538
8.7 Function Block for the Winder Diameter with Closed-Loop Tension Control and Speed Correction..	541
8.7.1 Introduction and overview.....	541
8.7.2 MB_WinderTensionCtrlSpeedType01.....	542
8.7.3 Commissioning the Winder with Tension Control and Speed Correction.....	555
8.8 Supplementary function blocks.....	558
8.8.1 Introduction and overview.....	558
8.8.2 MB_CalcInertialLimitType02.....	559
8.8.3 MB_UnwindMaterialType02.....	561
8.8.4 MB_WindTaperProfileType01	567
8.8.5 MB_WindSpeedControlAdaptionType01	571
8.8.6 MB_TraverseControlType01.....	574

Table of Contents

	Page
8.9 Parameter overview	593
9 RMB_TechCrossCutCrossSeal.library.....	595
9.1 Introduction and overview.....	595
9.2 MB_CrossCutSealType01.....	595
9.3 MB_CrossCutterCalcType04.....	609
9.4 MB_CAM_TABLE_DATA.....	618
9.5 MB_CORR_PROFILE.....	619
9.6 MB_RESOLUTION.....	620
9.7 MB_PUSH_OUT_CONFIG.....	621
10 RMB_TechCam.library.....	623
10.1 Introduction and overview.....	623
10.2 MB_CoordinatePairCamType01.....	623
10.3 MB_COORDINATE_PAIR_TYPE01.....	630
10.4 MB_COORDINATE_PAIR_CAM_SELECT.....	630
10.5 MB_FeedMovementCalcType01.....	631
10.6 MB_CURVETYPE.....	637
10.7 MB_FeedMovementCalcType02.....	637
10.8 Slotter: Introduction and definitions.....	643
10.9 MB_SlotterCalcType01.....	648
10.10 MB_SlotterCalcType02.....	660
10.11 MB_CAMPROF_SLOTTER.....	671
11 RIL_ParameterChannel.library.....	673
11.1 Introduction and overview.....	673
11.2 IL_ParameterChannel.....	673
11.3 IL_ParameterChannel - Protocol, Error Handling and Commissioning.....	680
11.3.1 Description of the Protocol Used.....	680
11.3.2 Commissioning the function block.....	687
11.3.3 Differences to the Legacy Systems.....	690
12 Service and support.....	693
Index.....	695

1 About this documentation

1.1 Validity of the documentation

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

Required qualification: Individual who is able to assess the tasks assigned and identify possible safety risks owing to qualification in the subject, knowledge and experience. The individual should also be familiar with the standards and regulations.

Overview on target groups and product phases

This documentation supports the user during the following phases:

- Engineering
- Commissioning
- Operation

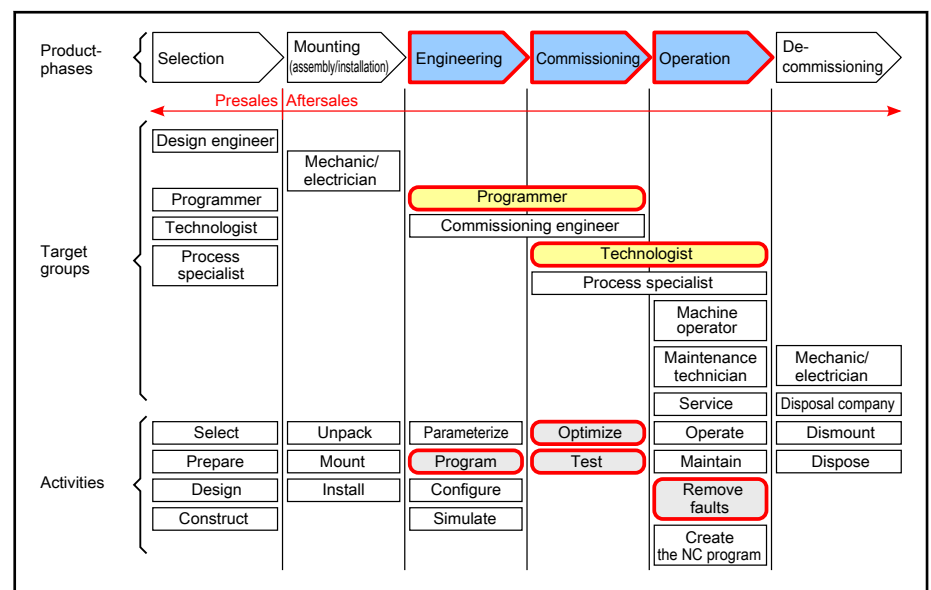


Fig. 1-1: Application phases

1.2 Documentation structure

[Libraries and target systems, page 19](#): Overview on the supported target systems for the technology libraries as well as information on the special features of the target system MLD

Documentations on the following libraries:

- [ML_TechBase.library, page 21](#)
- [ML_TechMotion.library, page 155](#)
- [ML_TechRegi.library/MX_TechRegi.lib, page 303](#)
- [ML_TechCrank.library, page 389](#)
- [ML_TechTensionAdvanced.library, page 431](#)
- [RMB_TechWinder.library/MX_TechWinder.lib, page 471](#)
- [RMB_TechCrossCutCrossSeal.library, page 595](#)
- [RMB_TechCam.library, page 623](#)
- [RIL_ParameterChannel.library, page 673](#)

About this documentation

- For information on the Bosch Rexroth customer service help desk, refer to [Service and support](#), page 693:

1.3 Using safety instructions

1.3.1 Safety instructions – Structure

The safety instructions are structured as follows:

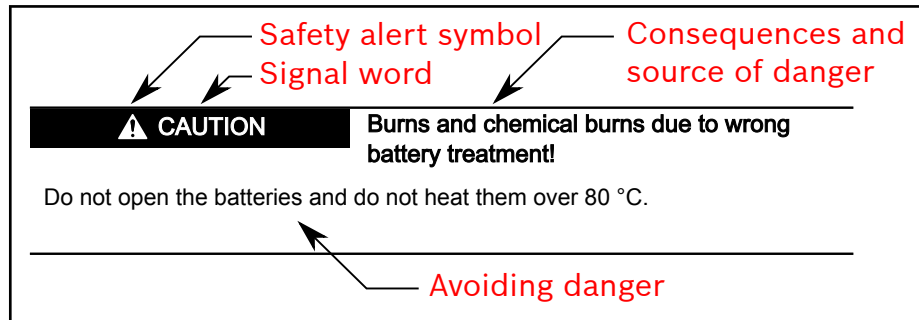


Fig. 1-2: Safety instructions – structure

1.3.2 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 is used to draw attention to the safety instruction and also provides information on the severity of the hazard.

The safety alert symbol (a triangle with an exclamation point), which precedes the signal words danger, warning and caution is used to alert the reader to personal injury hazards.

⚠ DANGER

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

⚠ WARNING

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

⚠ CAUTION

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

NOTICE

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

1.3.3 Symbols used

Pointers are displayed as follows:



This is a note.

Tips are displayed as follows:



This is a tip.

1.4 Required and supplementing documentations

Documentation titles with type codes and parts numbers

IndraWorks		MLC	XLC
/36/	Rexroth IndraWorks 13VRS Software Installation DOK-IWORKS-SOFTINS*V13-CORS-EN-P, R911336880 This documentation describes the IndraWorks installation.	X	X
/5/	Rexroth IndraWorks 13VRS Engineering DOK-IWORKS-ENGINEE*V13-APRS-EN-P, R911336870 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.	X	X
/20/	Rexroth IndraMotion MLC 13VRS Functional Description DOK-MLC***-FUNC****V13-APRS-EN-P, R911336295 This documentation describes wizards, context menus, dialogs, control commissioning, device configuration and functionalities of the IndraMotion MLC.	X	
/20/	Rexroth IndraLogic XLC 13VRS Functional Description DOK-XLC***-FUNC****V13-APRS-EN-P, R911336352 This documentation describes wizards, context menus, dialogs, control commissioning, device configuration and functionalities of the IndraLogic XLC.		X
/7/	Rexroth IndraWorks 13VRS CamBuilder DOK-IWORKS-CAMBUIL*V13-APRS-EN-P, R911336291 This documentation describes the basic principles and operation of the CamBuilder, the cam editing tool.	X	X
/37/	Rexroth IndraLogic XLC IndraMotion MLC 13VRS Automation Interface DOK-XLCMLC-AUT*INT*V13-APRS-EN-P, R911336356 This documentation describes the script-based access to IndraWorks project data via the interface of the Automation Interface.	X	X
/38/	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.	X	X

About this documentation

IndraWorks		MLC	XLC
/29/	Rexroth IndraLogic XLC IndraMotion MLC 13VRS Project Conversion DOK-XLCMLC-PROCONV*V13-APRS-EN-P, R911336366 This documentation describes the project conversion of IndraLogic 04VRS and IndraMotion MLC04VRS on IndraWorks version 12 with IndraLogic 2G. Changes with regard to Motion and PLC are described in detail.	X	X
/28/	Rexroth IndraMotion MLC 13VRS Commissioning DOK-MLC***-STARTUP*V13-CORS-EN-P, R911336308 This documentation describes the steps to commission and service the IndraMotion MLC system. It includes checklists for frequent tasks and a detailed description of the steps.	X	

Tab. 1-1: XCL/MLC documentation overview

Motion		MLC	XLC
/23/	Rexroth IndraLogic XLC IndraMotion MLC 13VRS PLCopen Libraries DOK-XLCMLC-FUNLIB**V13-LIRS-EN-P, R911336360 This documentation describes the function blocks, functions and data types of the RIL_Common-Types, ML_Base and ML_PLCOpen libraries for the IndraLogic XLC/IndraMotion MLC. It also includes the error reactions of function blocks.	X	X
/27/	Rexroth IndraLogic XLC IndraMotion MLC 13VRS Generic Application Template DOK-XLCMLC-TF*GAT**V13-APRS-EN-P, R911336370 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.	X	X
/31/	Rexroth IndraMotion MLC 13VRS RCL Programming Instruction DOK-MLC***-RCL*PRO*V13-APRS-EN-P, R911336297 This documentation provides information on the RobotControl. The programming language RCL (RobotControl Language) is focused. The program structure, variables, functions, motion statements and the required system parameters are described.	X	
/32/	Rexroth IndraMotion MLC 13VRS Robot Control V2 DOK-MLC***-ROCO***V13-RERS-EN-P, R911341035 The documentation provides information about the Robot Control V2. The focus of this documentation is on PLCopen programming. The program structure, variables, functions and motion commands as well as the required system parameters are described. Furthermore, the documentation contains the description of the ML_Robot library.	X	
/21/	Rexroth IndraLogic XLC IndraMotion MLC 13VRS Parameters DOK-XLCMLC-PARAM***V13-RERS-EN-P, R911336364 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.	X	X
/10/	Rexroth IndraDrive Firmware for Drive Controllers MPH-, MPB-, MPD-, MPC-07 DOK-INDRV*-MP*-07VRS**-FKRS-EN-P, R911328670 This documentation describes all functional properties of the IndraDrive firmware in the variants MPH-07, MPB-07, MPD-07 and MPC-07.		
/11/	Rexroth IndraDrive MPx-17 Functions DOK-INDRV*-MP*-17VRS**-APRS-EN-P, R911331236 This documentation describes all functional properties of the IndraDrive firmware in the variants MPB-17, MPM-17, MPC-17 and MPE-17.		

Tab. 1-2: XCL/MLC documentation overview

About this documentation

Field Buses		MLC	XLC
/4/	Rexroth IndraWorks 13VRS Field Buses DOK-IWORKS-FB*****V13-APRS-EN-P, R911336872 This documentation describes the supported field buses and their diagnostic function blocks.	X	X
/4a/	Rexroth IndraWorks 13VRS Field Buses Libraries DOK-IWORKS-FB*LIB**V13-LIRS-EN-P, R911337857 This manual describes the field bus libraries: RIL_ProfibusDP_02, RIL_ProfibusDP Slave, RIL_Profine-tIO, RIL_ProfinetIODevice, RIL_EtherNetIPAdapter, RIL_MappingList, RIL_SERCOSIII, RIL_Inline including their diagnostics and error reactions of the function blocks.	X	X
/63/	SERCOS System Manual for I/O Devices DOK-CONTRL-ILS3*****-APRS-EN-P, R911333512 This documentation describes the configuration, parameterization, commissioning and diagnostics of I/O devices with a SERCOS interface.	X	X

Tab. 1-3: XCL/MLC documentation overview

HMI		MLC	XLC
/8/	Rexroth IndraWorks 13VRS HMI DOK-IWORKS-HMI*****V13-APRS-EN-P, R911336874 This documentation describes the functions, configuration and operation of the user interfaces IndraWorks HMI Engineering and IndraWorks HMI Operation.	X	X
/6/	Rexroth IndraWorks 13VRS WinStudio DOK-IWORKS-WINSTUD*V13-APRS-EN-P, R911336882 This documentation describes the installation of the software, working with WinStudio and the creation and operation of applications.	X	X
/50/	Rexroth IndraLogic XLC IndraMotion MLC 13VRS HMI Connection DOK-XLCMLC-HMI*****V13-APRS-EN-P, R911336362 This documentation describes the visualization systems supported by the IndraLogic XLC and IndraMotion MLC and their connection.	X	X

Tab. 1-4: XCL/MLC documentation overview

PLC		MLC	XLC
/3/	Rexroth IndraWorks 13VRS IndraLogic 2G Programming Instruction DOK-IWORKS-IL2GPRO*V13-APRS-EN-P, R911336876 This documentation describes the PLC programming tool IndraLogic 2G and its use. It includes the basic use, first steps, visualization, menu items and editors.	X	X
/33/	Rexroth IndraWorks 13VRS Basic Libraries IndraLogic 2G DOK-IL*2G*-BASLIB**V13-LIRS-EN-P, R911336285 This documentation describes the system-comprehensive PLC libraries.	X	X

Tab. 1-5: XCL/MLC documentation overview

Technology		MLC	XLC
/30/	Rexroth IndraMotion MLC 13VRS Technology Libraries DOK-MLC***-TF*LIB**V13-LIRS-EN-P, R911336324 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.	X	
/60/	Rexroth IndraMotion MLC 13VRS RegisterControl (Library) DOK-MLC***-REGI*CO*V13-LIRS-EN-P, R911336306 This documentation describes the inputs and outputs of the individual function blocks and provides notes on their use.	X	
/62/	Rexroth IndraMotion MLC 13VRS RegisterControl (Application Manual) DOK-MLC***-REGI*CO*V13-APRS-EN-P, R911336304 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.	X	
/49/	Rexroth IndraMotion MLC 13VRS Winder Function Application DOK-MLC***-TF*WIND*V13-APRS-EN-P, R911336326 This application-related system documentation describes the application of the winder technology functions.	X	

Tab. 1-6: XCL/MLC documentation overview

Hydraulics		MLC	XLC
/40/	Rexroth IndraLogic XLC IndraMotion MLC 13VRS Sequential Programming DOK-XLCMLC-SEQPROG*V13-LIRS-EN-P, R911336368 This documentation includes the library for the sequential programming.	X	X
/41/	Rexroth IndraMotion MLC 13VRS MH_TechHydrBase DOK-MLC***-TF*HBASEV13-LIRS-EN-P, R911336312 No purpose defined.	X	
/42/	Rexroth IndraMotion MLC 13VRS MH_TechHydrMotion DOK-MLC***-TF*HMOT*V13-LIRS-EN-P, R911336316 No purpose defined.	X	
/43/	Rexroth IndraMotion MLC 13VRS MH_HydrControl DOK-MLC***-TFHCLIB*V13-LIRS-EN-P, R911336328 No purpose defined.	X	
/44/	Rexroth IndraMotion MLC 13VRS MH_SynchControl DOK-MLC***-TFHSLIB*V13-LIRS-EN-P, R911336330 No purpose defined.	X	

Tab. 1-7: XCL/MLC documentation overview

About this documentation

Hardware		MLC	XLC
/1/	Rexroth IndraControl L45/L65/L85 Control DOK-CONTRL-ICL45L65L85-PRRS-EN-P, R911332116 This documentation describes the IndraControl L45/L65/L85 controls.	X	X
/2/	Rexroth IndraControl L25 DOK-CONTRL-IC*L25*****-PRRS-EN-P, R911328474 This documentation describes the IndraControl L25 controls.	X	X
/24/	Rexroth IndraControl Lxx 13VRS Function Modules DOK-CONTRL-FM*LXX**V13-APRS-EN-P, R911336283 This documentation describes all function modules of the Lxx controls including engineering and diagnostics.	X	X
/12/	Rexroth IndraDrive Drive Controllers MPx-02 to MPx-07 DOK-INDRV*-GEN-**VRS**-PARS-EN-P, R911297317 This documentation describes all parameters implemented in the firmware for drive controllers of the IndraDrive family. It supports the parameterization of the drive controllers.		
/13/	Rexroth IndraDrive MPx-02 to MPx-07 and HMV DOK-INDRV*-GEN-**VRS**-WARS-EN-P, R911297319 This documentation describes all diagnostics implemented in the following firmwares: • Drive controller firmwares from MPx-02 to MPx-08 and • firmwares of the supply devices of the type "HMV". It supports the operating crew as well as the programmer at troubleshooting.		
/35/	Rexroth IndraDrive Drive Controller Control Sections CSB01, CSH01, CDB01 DOK-INDRV*-CSH*****-PRRS-EN-P, R911295012 This documentation is a project planning manual to select and use the control sections CSB01, CDB01 and CSH01 for drive controllers of the product families Rexroth IndraDrive M and Rexroth IndraDrive C.		

Tab. 1-8: XCL/MLC documentation overview

Diagnostics and service		MLC	XLC
/26/	Rexroth IndraWorks 13VRS IndraMotion Service Tool DOK-IWORKS-IMST****V13-RERS-EN-P, R911337707 This documentation describes the IndraMotion Service Tool (IMST). It is a web-based diagnostic tool to access the control systems IndraMotion MLC L25, L45 or 65 as well as the MLP via an Ethernet high-speed connection. The IMST allows OEMs, end users and service engineers to access and remotely diagnose a system. A PC with Internet Explorer version 6 or 7 is required.	X	X
/22/	Rexroth IndraLogic XLC IndraMotion MLC 13VRS Diagnostics DOK-XLCMLC-DIAG****V13-RERS-EN-P, R911336358 This documentation includes all control diagnostics implemented in the control systems IndraLogic XLC and IndraMotion MLC.	X	X

Tab. 1-9: XCL/MLC documentation overview

About this documentation

System Overview		MLC	XLC
/48/	Rexroth IndraMotion for Printing 13VRS System Overview DOK-IM*PR*-SYSTEM**V13-PRRS-EN-P, R911336289 This documentation describes the product IndraMotion for Printing. It introduces the control systems, drive systems and I/O systems as well as the commissioning and programming.	X	
/48/	Rexroth IndraMotion for Packaging 13VRS System Overview DOK-IM*PA*-SYSTEM**V13-PRRS-EN-P, R911336287 This documentation describes the product IndraMotion for Packaging. It introduces the control systems, drive systems and I/O systems as well as the commissioning and programming.	X	
/9/	Rexroth IndraMotion MLC 13VRS System Overview DOK-MLC***-SYSTEM**V13-PRRS-EN-P, R911336310 This documentation provides an overview on the hardware/software components of the automation system IndraMotion MLC in the mentioned version. It helps assembling a system.	X	
/9/	Rexroth IndraLogic XLC 13VRS System Overview DOK-XLC***-SYSTEM**V13-PRRS-EN-P, R911336354 This documentation provides an overview on hardware/software components of the automation system IndraLogic XLC in the respective version. It helps assembling a system.		X

Tab. 1-10: XCL/MLC documentation overview

First Steps		MLC	XLC
/25/	Rexroth IndraMotion MLC 13VRS, First Steps DOK-MLC***-F*STEP**V13-CORS-EN-P, R911336293 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.	X	
/25/	Rexroth IndraLogic XLC 13VRS, First Steps DOK-XLC***-F*STEP**V13-CORS-EN-P, R911336350 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.		X

Tab. 1-11: XCL/MLC documentation overview

1.5 Terms and abbreviations

Term	Explanation
Libraries	Includes functions and function blocks
Tech-FB	Technology function block
PLS	Program Limit Switch
PLCopen	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.
Target system	Control (e.g. IndraMotion MLC)
SYNTAX	Automation systems for printing and paper-processing machinery

About this documentation

VisualMotion	Programmable multi-axis motion control system
IndraMotion MLC	Compact Motion Logic systems with Motion, Robot and Logic Control functionalities
IndraLogic XLC	Compact PLC system with integrated Motion functionality
IndraMotion MLD	Drive-based Motion Logic system
IndraWorks	Project planning and commissioning tool of Bosch Rexroth
IndraDrive	Drive control unit
LD	Ladder diagram
FBD	Function block diagram
PWM	Pulse-width modulation
MA	IndraMotion MLD Advanced
MLD-S	Motion Logic Drive (single drive)
MLD-M	Motion Logic Drive (multi drive)
n. def	Not defined

Tab. 1-12: Names and abbreviations used

2 Overview

2.1 Libraries and target systems

The following table list all supported target systems for the technology libraries.

Library	Target system IndraMotion MLC	Target system IndraLogic XLC	Target system IndraMotion MLD
ML_TechBase.library, page 21	X	X	
ML_TechMotion.library, page 155	X		
ML_TechRegi.library MX_TechRegi.lib, page 303	X		X
ML_TechCrank.library, page 389	X		
ML_TechTensionAdvanced.library, page 431	X		
RMB_TechWinder.library MX_TechWinder.lib, page 471	X		X
RMB_TechCrossCutCrossSeal.library, page 595	X		
RMB_TechCam.library, page 623	X		
RIL_ParameterChannel.library, page 673	X	X	

Tab. 2-1: Target systems supported by the technology libraries



The technology function blocks (winder, register controller, tension controller...) are not supported for a SercosDrive.
IndraDrive with interpolation in the control or drive is required.

2.2 Special Features of the IndraMotion MLD

2.2.1 General Information

The IndraMotion MLD is an "onboard" control integrated into the hardware of the IndraDrive control section. The PLC firmware is part of the drive firmware. In contrast to the IndraMotion MLC, the PLC engineering environment is the IndraLogic 1.X.



The technology libraries (RMB_TechWinder, ML_TechRegi, ML_TechMotion, ML_TechCam...) do **not** support the PLC development environment of drive firmware MPx18 or higher.

2.2.2 Drive Firmware, Function Package

The drive firmware is available in the following variants starting from 07VRS:

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

Overview

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

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



The additional package "IndraMotion MLD Advanced" is only available for MPC firmware and MPH firmware (not for MPB firmware).

Technology libraries requiring, e.g. the additional package "IndraMotion MLD Advanced":

- MX_TechWinder
- MX_TechRegi

Multi-axis IndraMotion MLD-M The MPC drive firmware is required for the multiple axis control IndraMotion MLD-M starting from the drive firmware 07VRS. The CSH01.3 control unit is required for this firmware.

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



The CSH01.3 control unit is required for the MPC firmware starting from MPX07.

3 ML_TechBase.library

3.1 Introduction and overview

General information	Technology function blocks (Tech FB) extend the basic functionality of the target systems IndraMotion MLC/IndraLogic XLC and IndraMotion 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 to the IndraMotion MLC/IndraLogic XLC via the "ML_TechBase.library" library. This documentation describes the functionality as well as the inputs and outputs of the technology function blocks.
Preconditions	Technology function blocks of the "ML_TechBase.library" 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. The identifier "FP_MODULO" for example is set on the axis parameter A-0-0045 in an IndraMotion MLC/IndraLogic XLC. In an IndraMotion MLD system it is set to the Sercos parameter S-0-0103. Thus, the system can abstract better at application level and can write a generally valid code.

The ML_TechBase function blocks

Function block	Description
Jogging variables (adjustment)	
MB_ContinuousAdjustType01, page 26	Continuous adjustment of a REAL variable via binary inputs
MB_ContinuousAdjustType02, page 30	Continuous adjustment of a DINT variable to be influenced via binary inputs
MB_IncrementalAdjustType01, page 33	Incremental adjustment of a REAL variable via binary inputs
Backing up and restoring axes (AxisBackupRestore)	
ML_AxisBackup, page 35	Parameter backup of an axis on the Compact Flash card
ML_AxisRestore, page 37	Parameter restoration of an axis from the Compact Flash card
Two-position controller (Controller)	
ML_TwoPosCtrlType01, page 41	Implements a two-point controller with hysteresis
Cam table download (LoadCamData)	
MB_LoadCamData, page 43	Downloads cams to the drive and control to encapsulate the existing detail differences.
Mathematical functions (MathematicalFunctions)	
ML_InterpolationLinear, page 46	Cyclically calculates the y-value from the x-value according to the data point table
ML_MaxValue, page 48	Determines the present maximum value from the time characteristics of an input signal
Programmable limit switches (PhaseLimitSwitch)	

ML_TechBase.library

Function block	Description
MC_DigitalCamSwitch, page 49	Implements a freely programmable limit switch with up to 32 switches on 8 outputs
Temperature controller (TemperatureController)	
IL_TempControlType01, page 66	Controls temperatures including the comfort functions such as monitoring, pause and manual mode
IL_TempControlType02, page 84	Controls temperatures, online register setting
Touch probe function blocks (TouchProbe)	
MB_InitTouchProbe, page 106	Initializing the touch probe functionality
MB_TouchProbe, page 114	Recording an axis position, master axis position, time or single-shot measurement
MB_TouchProbeContinuous, page 117	Recording an axis position, master axis position or time at continuous measurement
MB_WriteExpectWindow, page 120	Configures the expectation window for the touch probe functionality
MB_AbortTrigger, page 123	Cancels the MB_TouchProbe function block
Trigonometric functions (TrigonometricFunctions)	
ML_DegToInc, page 125	Converts angle from degree to corresponding incremental value
ML_DegToRad, page 126	Converts angle in degree to angle in radian measure incl. modulo
ML_IncToDeg, page 126	Converts angle specified in increments to degree
ML_IncToRad, page 127	Converts angle specified in increments to the corresponding radian measure
ML_RadToDeg, page 127	Converts an angle in radian measure to the corresponding angle in degree incl. modulo
ML_RadToInc, page 128	Converts an angle in radian measure to the corresponding incremental value
Cyclic data channels (Utilities\CyclicData)	
MB_AllocCyclicParameter, page 130	Checking and revising the configuration of a parameter to be read or written cyclically in the cyclic data channel
MB_FreeCyclicParameter, page 132	Enabling an already configured parameter to be read or written cyclically in the cyclic data channel
MB_GetCyclicChannelConfig, page 134	Reads the configuration of cyclic read and write data channels and saves them in a data structure
MB_GetCyclicParameterHandle, page 135	Searches for a cyclic parameter to be read or written in the cyclic data channel and stores all required information in one handle
MB_ReadCyclicParameter, page 137	Reads a cyclic parameter via the handle and returns the current content as DINT
MB_ReadCyclicRealParameter, page 137	Reads a cyclic parameter via the handle and returns the current content as REAL
MB_WriteCyclicParameter, page 138	Writes a cyclic parameter via the handle. Number of decimals are converted according to Sercos declaration
MB_WriteCyclicRealParameter, page 138	Writes a cyclic parameter via the handle. Number of decimals are rounded

Function block	Description
MB_GetLastCyclicParameterError, page 139	Returns the last error while reading or writing a cyclic value for a handle and deletes the error
Logic applications (Utilities\Logic)	
IL_LinkedList, page 142	Managing a list linked twice
Modulo values (Utilities\Modulo)	
MB_GetModuloType01, page 147	Returns the modulo value (value range between 0 and + modulo value) (data type REAL)
MB_GetModuloType02, page 148	Returns the modulo value (value range between 0 and + modulo value) (data type DINT)
MB_GetModuloType03, page 149	Returns the modulo value (value range between - modulo value and + modulo value)
MB_GetModuloType04, page 150	Returns the modulo value (value range is between - modulo/2 and + modulo/2)
System information (Utilities\SystemInfo)	
MB_GetSystemInfo, page 152	Checks function packages and system options

Tab. 3-1: Function blocks in the ML_TechBase.library

3.2 Jogging variables (adjustment)

3.2.1 Introduction and overview

Adjust (jog) the PLC variables continuously or incrementally via binary inputs using the following function blocks:

- [MB_ContinuousAdjustType01 on page 26](#)
- [MB_ContinuousAdjustType02 on page 30](#)
- [MB_IncrementalAdjustType01 on page 33](#)

When using the above mentioned function blocks to jog PLC variables, consider the following requirements:

- Stop the operation via a binary input ("Enable")
- The variable to be affected can be continuously adjusted using the function blocks MB_ContinuousAdjustType02 and IL_ContinuousAdjustType02 (similar to the long jog for SYNAX)
- The variable to be affected can be incrementally adjusted using the function block MB_IncrementalAdjustType01 (similar to the short jogging for SYNAX)
- The selected variable can either be changed in incremented or decremented steps within the limit values
- When reaching a limit value "HighLimitActive" = TRUE or "LowLimitActive" = TRUE), the continuous adjustment is disabled and thus, the function block does no longer affect the variable. In this case, the respective limit value is displayed at the "HighLimit" input or at the "LowLimit" input

Basics

If the limits "HighLimit" and "LowLimit" have the same value for a specified modulo value, "LowLimit" is set to zero and "HighLimit" is set to the modulo value. Thus, it is possible to adjust the limit values in the entire modulo range.

The change rate describes the rate the affected variable changes during adjustment. This rate depends on the step width and the number of steps per

ML_TechBase.library

second. The change rate is calculated as follows, for example, when changing a position:

$$\text{AlterationVelocity} \left[\frac{\text{Degree}}{s} \right] = \text{Increments} [\text{Degree}] * \text{Increments per Second} \left[\frac{1}{s} \right]$$

Fig. 3-1: Change rate when changing a position

If, for example, the speed of the virtual master axis is changed, the change rate is calculated as follows:

$$\text{AlterationVelocity} \left[\frac{r.p.m.}{s} \right] = \text{Increments} [r.p.m.] * \text{Increments per Second} \left[\frac{1}{s} \right]$$

Fig. 3-2: Change rate when changing the axis speed

As long as the control signal is present, the affected variable is changed with the defined velocity.

Adjustment limits

The variable can only be adjusted via the corresponding inputs within the determined limit values.

Exception:

If the initial value of the variable to be adjusted is not in the specified limits, the affected variable can only be adjusted in the direction of the target range. If a modulo value is specified, the affected variable can be adjusted in both directions. After reaching the target range, it is no longer possible to leave this range.

Preset

The output variable can be pre-assigned to a definite value using the inputs "Preset" and "PresetValue". The "PresetAck" is set as soon as the "PresetValue" was applied. The output variable remains on the "PresetValue" as long as the "Preset" input is set.

Basic rules

- If both control signals ("Inc"/"Dec") are TRUE at the same time, they are evaluated as FALSE
- A control signal changes the polarity and causes an immediate change of direction. The current adjustment is also not completed
- If the adjustment velocity is changed during a motion, the behavior of the following function blocks differ:

MB_ContinuousAdjustType01/MB_ContinuousAdjustType02

MB_IncrementalAdjustType01

The change becomes immediately effective for MB_ContinuousAdjustType01 and MB_ContinuousAdjustType02. In contrast, the motion for the MB_IncrementalAdjustType01 is completed before the new step velocity is applied. A subsequent adjustment is performed with the adjusted variables.

Furthermore, the following rules apply to the function block MB_IncrementalAdjustType01:

- The inputs "Inc" and "Dec" are evaluated with regard to the edges. A rising edge causes an adjustment by the specified step width ("Step-Width")
- A new control signal (positive edge) with the same polarity detected during a running motion causes another motion following the current motion

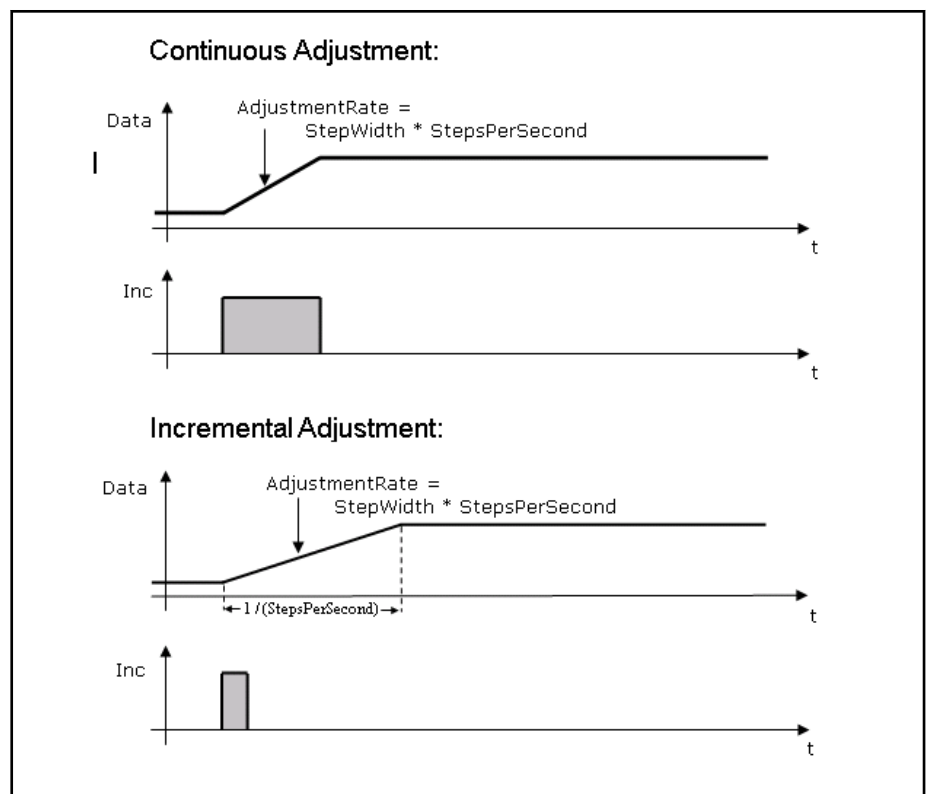


Fig. 3-3: Comparison of the behavior between a continuous and an incremental change

During the first activation ("Enable" = TRUE), the function blocks MB_ContinuousAdjustType01 and MB_ContinuousAdjustType02 initialize the variables required to determine the cycle time (irrespective of the present inputs). In the second cycle, a calculated cycle time is available required to calculate the correct change rate. Subsequently, the cycle time is updated continuously.

By specifying a modulo value, the function block provides an output variable between zero and the specified modulo value. When exceeding the modulo value, the output variable is reset to zero. However, the adjustment continues. The specified valid range can thus be between zero and the modulo value. The range can be adjusted by the zero value.

ML_TechBase.library

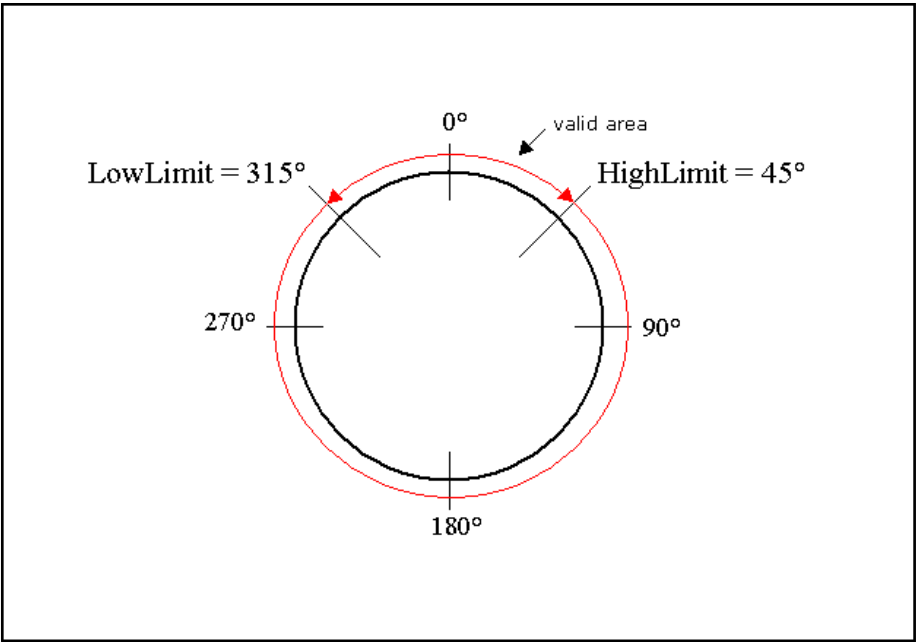


Fig. 3-4: Possible allocation of a valid range when specifying a modulo value of 360°

In addition, the function blocks MB_ContinuousAdjustType01 and MB_ContinuousAdjustType02 allow to adjust the change rate depending on the time by specifying scaling factors and time intervals.

3.2.2 MB_ContinuousAdjustType01

Brief description The MB_ContinuousAdjustType01 function block allows a continuous adjustment of a REAL variable via binary inputs.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-2: Reference table of the MB_ContinuousAdjustType01 function block

Interface description

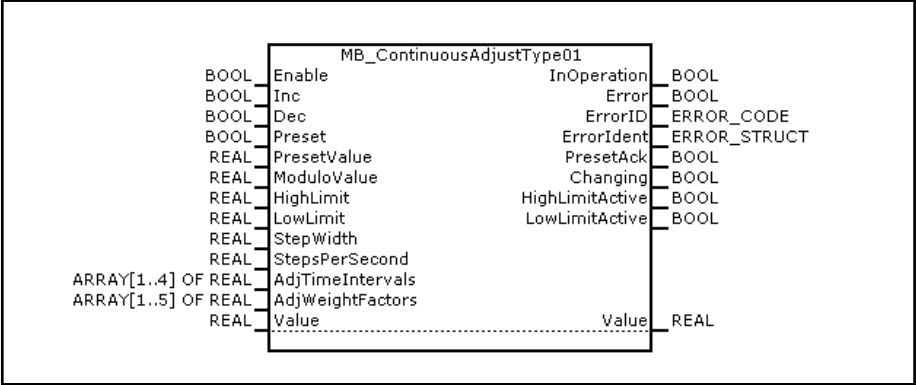


Fig. 3-5: MB_ContinuousAdjustType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enabling the function block (cyclic, state-controlled)
	Inc	BOOL	Increase affected variable
	Dec	BOOL	Decrease affected variable
	Preset	BOOL	Set the preset value (PresetValue)

I/O type	Name	Data type	Description
	PresetValue	REAL	Specified value for the preset value
	ModuloValue	REAL	Modulo value (if 0, then absolute processing)
	HighLimit	REAL	Maximum output value of the variable to be affected
	LowLimit	REAL	Minimum output value of the variable to be affected
	StepWidth	REAL	Step width
	StepsPerSecond	REAL	Number of steps per second
	AdjTimeIntervals	ARRAY [1..4] OF REAL	Time intervals in seconds to scale the change rate
	AdjWeightFactors	ARRAY [1..5] OF REAL	Scaling factors of the change rate
VAR_OUTPUT	InOperation	BOOL	Function block enabled. The outputs "HighLimitActive" and "LowLimitActive" are valid
	Error	BOOL	Calculation of the variable "Value" to be affected was completed with an error. The output variable "ErrorIdent" is valid
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed description of the diagnostics in the event of an error Troubleshooting on page 29
	PresetAck	BOOL	The value of the "PresetValue" input was applied
	Changing	BOOL	The variable to be affected was calculated. Output variable "Value" is valid
	HighLimitActive	BOOL	Maximum output value of the variable to be affected was reached
	LowLimitActive	BOOL	Minimum output value of the variable to be affected was reached
VAR_IN_OUT	Value	REAL	Variable to be affected

Tab. 3-3: Interface variables of the MB_ContinuousAdjustType01 function block

Signal-time diagram

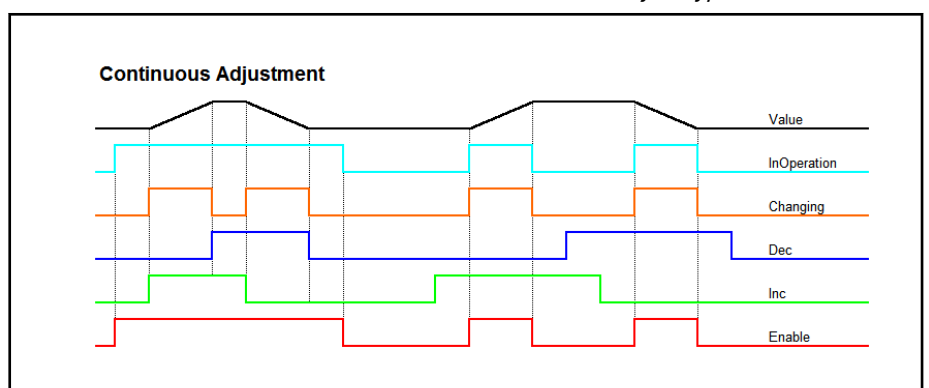


Fig. 3-6: Continuous parameter adjustment

ML_TechBase.library

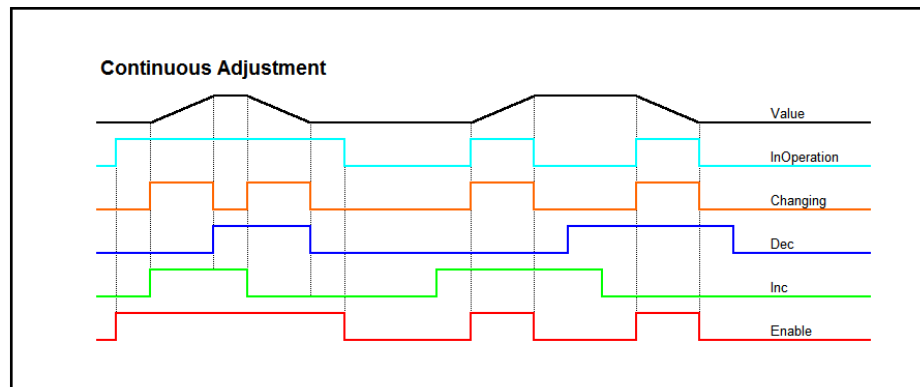


Fig. 3-7: Continuous parameter adjustment

Functional Description

The variable "Value" can be adjusted either in positive or negative direction. A valid range is specified to affect the variable using the limit values. The modulo value specification allows the adjustment in the reference system of the axis. Specify the step width and the number of steps per second to determine the change rate of the affected variable. Additionally, a specified value ("PresetValue") in the permissible operating range can be directly written to the affected variable.

After processing enabled via "Enable", the variable to be affected can be changed via the inputs "Inc" (in positive direction) and "Dec" (in negative direction). While input "Inc" or "Dec" is set, the adjustable variable is continuously increased (or decreased) with the given velocity. ("StepsPerSecond" * "StepWidth").

When initializing the function block, ensure that the input variables "ModuloValue", "HighLimit" and "LowLimit" are provided with valid values.

Specify the scaling factors ("AdjWeightFactors[]") and their respective time intervals ("AdjTimeIntervals[]") to modify the change rate with regard to time. The scaling factors and the time intervals have to be set in addition to the inputs "StepsPerSecond" and "StepWidth". Depending on the current time interval, the change rate is multiplied with the respective scaling factor. The figure below shows the behavior of the function block using an example configuration: The "AdjTimeIntervals" input variables ([T1; T2; T3; T4]) are specified in seconds. T5 is the time interval between the completion of the time interval T4 and the adjustment abort. The "AdjWeightFactors" input variables ([0.5; 1; 1.5; 2; 4]) determine the scaling of the change rate. The last "AdjWeightFactors" value ("AdjWeightFactors [5]") is assigned to the time interval T5.

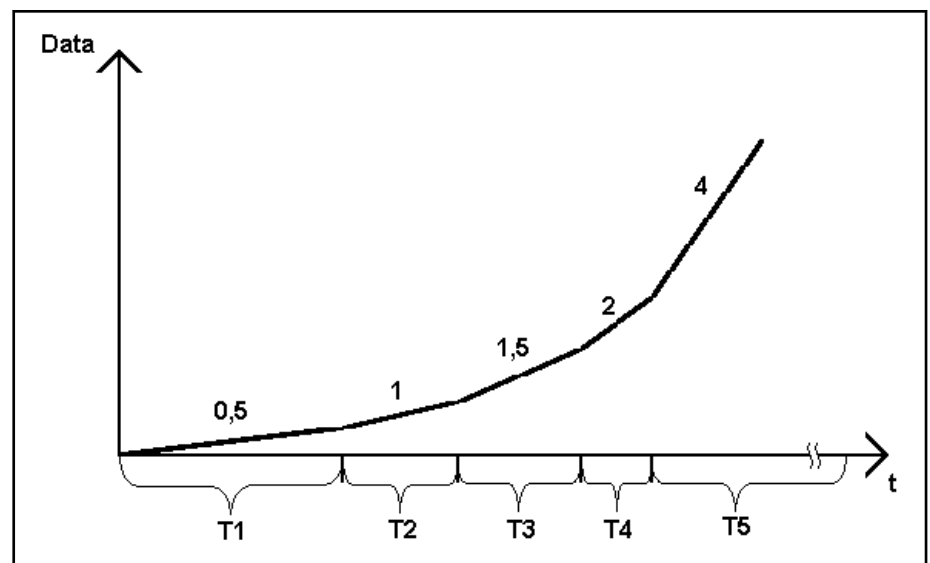


Fig. 3-8: Scaled adjustment



If a time interval of 0 is selected, the following scaling factor is retained until the adjustment is aborted.

Example:

If "AdjTimeIntervals[2]" = 0, the scaling factor of "AdjWeightFactors[3]" is retained until the process is aborted.

The calculation of the elapsed time to define the correct time interval of the scaling factors is reset with each new adjustment. If the inputs "AdjWeightFactors" and "AdjTimeIntervals" are not set, it is continuously adjusted with the change rate specified in the inputs "StepsPerSecond" and "StepWidth".



Positive values have to be specified for the input values of the time intervals ("AdjTimeIntervals"). The scaling factors can also contain negative values.

The variable to be affected can only be set to the specified preset value if it is in the valid range specified by "HighLimit" and "LowLimit".

The "InOperation" output signals that the function block is operating. However, there is no adjustment. The "Changing" output indicates that the adjustment was processed without errors. The value of the adjusted variable "Value" is valid.

The "Error" output indicates if an error occurs during the adjustment. For detailed information on the error, refer to the "ErrorIdent" structure.

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#0002	16#0000	Inputs are not in the valid range
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

Tab. 3-4: Error codes of the MB_ContinuousAdjustType01 function block

ML_TechBase.library

3.2.3 MB_ContinuousAdjustType02

Brief description The MB_ContinuousAdjustType02 function block allows a continuous adjustment of a DINT variable to be influenced via binary inputs.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-5: Reference table of the MB_ContinuousAdjustType02 function block

Interface description

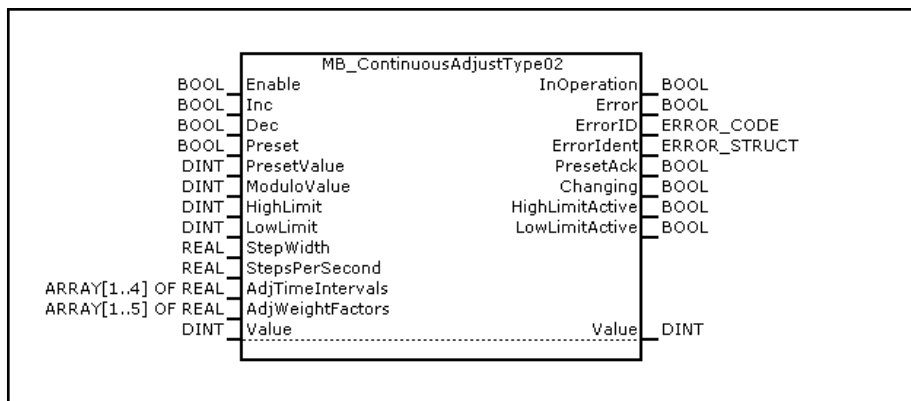


Fig. 3-9: MB_ContinuousAdjustType02 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enabling the function block (cyclic, state-controlled)
	Inc	BOOL	Increase affected variable
	Dec	BOOL	Decrease affected variable
	Preset	BOOL	Set the preset value (PresetValue)
	PresetValue	DINT	Specified value for the preset value
	ModuloValue	DINT	Modulo value (if 0, then absolute processing)
	HighLimit	DINT	Maximum output value of the variable to be affected
	LowLimit	DINT	Minimum output value of the variable to be affected
	StepWidth	REAL	Step width
	StepsPerSecond	REAL	Steps per second
	AdjTimeIntervals	ARRAY [1..4] OF REAL	Time intervals in seconds to scale the change rate
	AdjWeightFactors	ARRAY [1..5] OF REAL	Scaling factors of the change rate
VAR_OUTPUT	InOperation	BOOL	Function block enabled. The outputs HighLimitActive and LowLimitActive are valid
	Error	BOOL	Calculation of the variable "Value" to be affected was completed with an error. The output variable "ErrorIdent" is valid
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed description of the diagnostics in the event of an error Troubleshooting on page 32

I/O type	Name	Data type	Description
	PresetAck	BOOL	The value of the "PresetValue" input was applied
	Changing	BOOL	The variable to be affected was calculated. The output variable "Value" is valid
	HighLimitActive	BOOL	Maximum output value of the variable to be affected was reached
	LowLimitActive	BOOL	Minimum output value of the variable to be affected was reached
VAR_IN_OUT	Value	DINT	Variable to be affected

Tab. 3-6: Interface variables of the MB_ContinuousAdjustType02 function block

Signal-time diagram

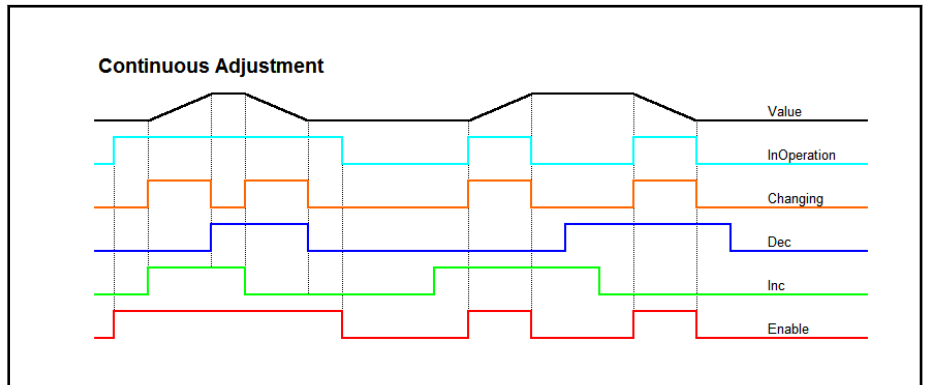


Fig. 3-10: Continuous parameter adjustment

Functional Description

The variable "Value" can be adjusted either in positive or negative direction. A valid range is specified to affect the variable using the limit values. The modulo value specification allows the adjustment in the reference system of the axis. Specify the step width and the number of steps per second to determine the change rate of the affected variable. Additionally, a specified value (PresetValue) in the permissible operating range can be directly written to the affected variable.

After processing enabled via "Enable", the variable to be affected can be changed via the inputs "Inc" (in positive direction) and "Dec" (in negative direction). While input "Inc" or "Dec" is set, the adjustable variable is continuously increased (or decreased) with the given velocity. ("StepsPerSecond" * "StepWidth").

When initializing the function block, ensure that the input variables "Modulo-Value", "HighLimit" and "LowLimit" are provided with valid values.

Specify the scaling factors ("AdjWeightFactors[]") and their respective time intervals ("AdjTimeIntervals[]") to modify the change rate with regard to time. The scaling factors and the time intervals have to be set in addition to the inputs "StepsPerSecond" and "StepWidth". Depending on the current time interval, the change rate is multiplied with the respective scaling factor. The figure below shows the behavior of the function block using an example configuration: The "AdjTimeIntervals" input variables ([T1; T2; T3; T4]) are specified in seconds. T5 is the time interval between the completion of the time interval T4 and the adjustment abort. The "AdjWeightFactors" input variables ([0.5; 1; 1.5; 2; 4]) determine the scaling of the change rate. The last "AdjWeightFactors" value ("AdjWeightFactors [5]") is assigned to the time interval T5.

ML_TechBase.library

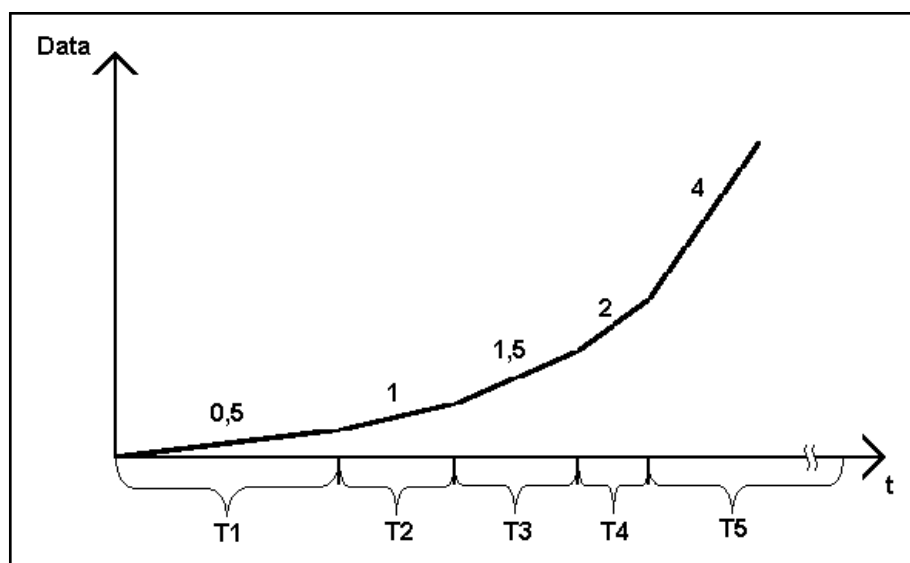


Fig. 3-11: Scaled adjustment



If a time interval of 0 is selected, the following scaling factor is retained until the adjustment is aborted.

Example:

If "AdjTimeIntervals[2]" = 0, the scaling factor of "AdjWeightFactors[3]" is retained until the process is aborted.

The calculation of the elapsed time to define the correct time interval of the scaling factors is reset with each new adjustment. If the inputs "AdjWeightFactors" and "AdjTimeIntervals" are not set, it is continuously adjusted with the change rate specified in the inputs "StepsPerSecond" and "StepWidth".



Positive values have to be specified for the input values of the time intervals (AdjTimeIntervals). The scaling factors can also contain negative values.

The variable to be affected can only be set to the specified preset value if it is in the valid range specified by "HighLimit" and "LowLimit".

The "InOperation" output signals that the function block is operating. However, there is no adjustment. The "Changing" output indicates that the adjustment was processed without errors. The value of the adjusted variable "Value" is valid.

The "Error" output indicates if an error occurs during the adjustment. For detailed information on the error, refer to the "ErrorIdent" structure.

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#0002	16#0000	Inputs are not in the valid range
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

Tab. 3-7: Error codes of the MB_ContinuousAdjustType02 function block

3.2.4 MB_IncrementalAdjustType01

Brief description The MB_IncrementalAdjustType01 function block allows an incremental adjustment of a REAL variable via binary inputs.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-8: Reference table of the MB_IncrementalAdjustType01 function block

Interface description

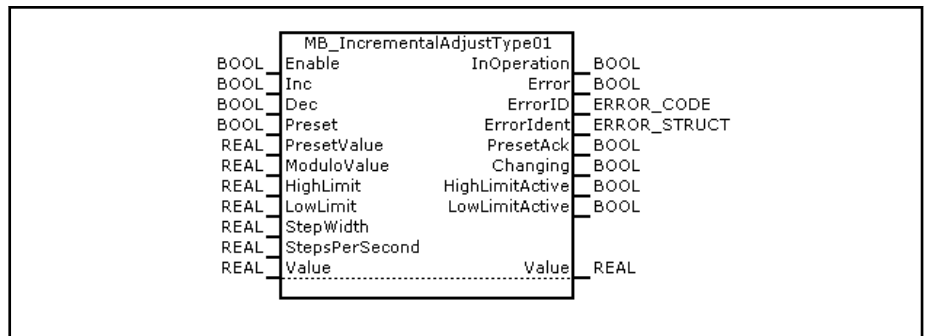


Fig. 3-12: MB_IncrementalAdjustType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Enabling the function block (cyclic, state-controlled)
	Inc	BOOL	Increase affected variable
	Dec	BOOL	Decrease affected variable
	Preset	BOOL	Set the preset value (PresetValue)
	PresetValue	REAL	Specified value for the preset value
	ModuloValue	REAL	Modulo value for rotary calculation
	HighLimit	REAL	Maximum output value of the variable to be affected
	LowLimit	REAL	Minimum output value of the variable to be affected
	StepWidth	REAL	Step width
	StepsPerSecond	REAL	Steps per second
VAR_OUTPUT	InOperation	BOOL	Function block enabled. The outputs "HighLimitActive" and "LowLimitActive" are valid
	Error	BOOL	Calculation of the variable "Value" to be affected was completed with an error. The output variable "ErrorIdent" is valid
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed description of the diagnostics in the event of an error Troubleshooting on page 34
	PresetAck	BOOL	The value of the "PresetValue" input was applied
	Changing	BOOL	The variable to be affected was calculated. Output variable "Value" is valid
	HighLimitActive	BOOL	Maximum output value of the variable to be affected was reached

ML_TechBase.library

I/O type	Name	Data type	Description
	LowLimitActive	BOOL	Minimum output value of the variable to be affected was reached
VAR_IN_OUT	Value	REAL	Variable to be affected

Tab. 3-9: Interface variables of the MB_IncrementalAdjustType01 function block

Signal-time diagram

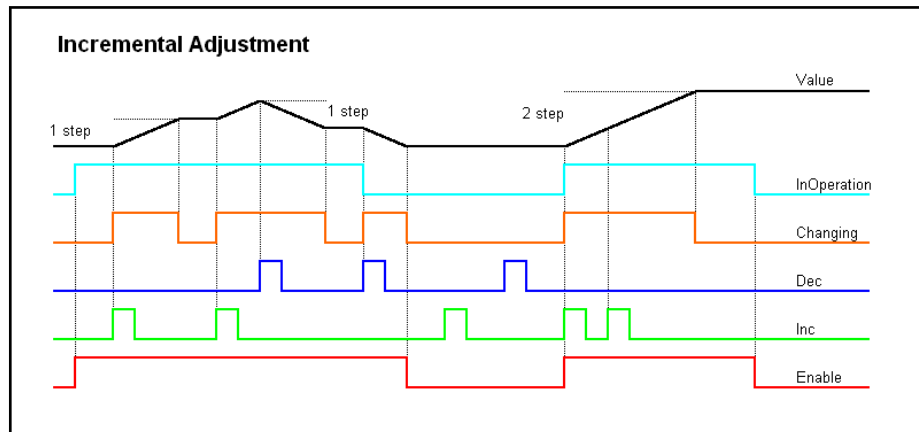


Fig. 3-13: Incremental parameter adjustment

Functional Description

The variable "Value" can be adjusted either in positive or negative direction.

A valid range is specified to affect the variable using the limit values. The modulo value specification allows the adjustment in the reference system of the axis.

Specify the step width and the number of steps per second to determine the change rate of the affected variable.

Additionally, a specified value ("PresetValue") in the permissible operating range can be directly written to the affected variable.

After processing enabled via "Enable", the variable to be affected can be changed via the inputs "Inc" (in positive direction) and "Dec" (in negative direction). While input "Inc" or "Dec" is set, the adjustable variable is continuously increased (or decreased) with the given velocity. ("StepsPerSecond" * "StepWidth").



The variable to be affected can only be set to the specified preset value if it is in the valid range specified by "HighLimit" and "LowLimit".

The "InOperation" output signals that the function block is operating. However, there is no adjustment. The "Changing" output indicates that the adjustment was processed without errors. The value of the adjusted variable "Value" is valid.

If an error occurs during the function block processing, this is signaled by the output "Error" = TRUE. For detailed information on the error, refer to the "ErrorIdent" output structure.

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#0002	16#0000	Inputs are not in the valid range
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

Tab. 3-10: Error codes of the MB_IncrementalAdjustType01 function block

3.3 Backing up and restoring drive parameters (AxisBackupRestore)

3.3.1 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 two function blocks.

[ML_AxisBackup on page 35](#): Creates a parameter backup and stores it as text file on the Compact Flash card of the control.

[ML_AxisRestore on page 37](#): Transfers the parameter backup back to the axis.

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

3.3.2 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.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-11: Reference table of the ML_AxisBackup

Interface description

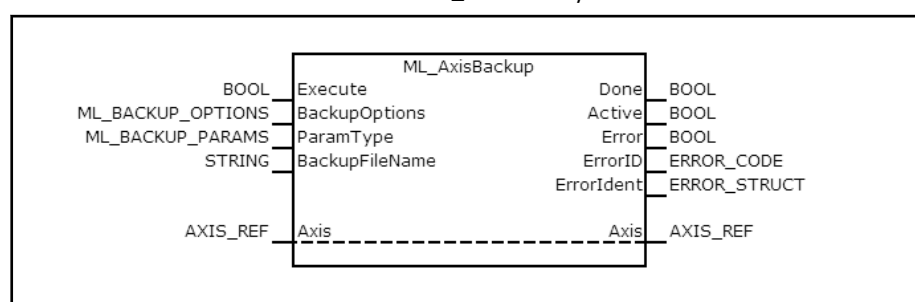


Fig. 3-14: ML_AxisBackup function block

ML_TechBase.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 on page 40
	ParamType	ML_BACKUP_PARAMS	Selection of the parameters to be backed up (all, A, S and P) ML_BACKUP_PARAMS on page 40
	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 the event of an error
	ErrorIdent	ERROR_STRUCT	Error diagnostics (see table Error codes on page 37)
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis

Tab. 3-12: 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. 3-13: Min./max. and default values of the ML_AxisBackup function block

Functional Description

The ML_AxisBackup function block backs up the parameter of an axis in a text file on the user partition of the Compact Flash card of the control. If the file already exists, it is overwritten. The file can be read again by the IL_Axis-Restore 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 are backed up for restoration. This is the presetting



The "BackupOptions" is currently only evaluated for S- and P-parameters. When exporting A-parameters, all parameters 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. 3-14: Error codes of the ML_AxisBackup function block

3.3.3 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.

ML_TechBase.library

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-15: Reference table of the ML_AxisRestore

Interface description

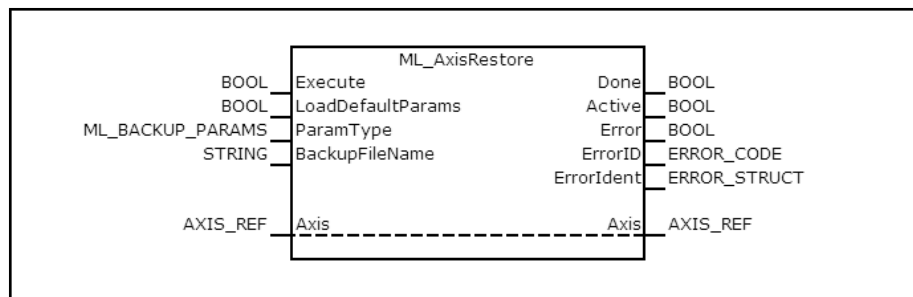


Fig. 3-15: ML_AxisRestore function block

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 on page 40
	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 the event of an error
	ErrorIdent	ERROR_STRUCT	Error diagnostics (see table Error codes on page 40)
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis

Tab. 3-16: 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"
ParamType	ML_BACKUP_PARAMS			ASP_PARAMS	Rising edge at "Execute"
BackupFileName	STRING			"	Rising edge at "Execute"

Tab. 3-17: 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 the parameter file has

been loaded 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 blocks 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-0-0005, 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

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 0x51 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 yourself.

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.

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

ML_TechBase.library

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.

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
OTHER_ERROR	16#00001E15	16#xxxxxxx	Error when parsing the backup file
STATE_MACHINE_ERROR	16#00000006	16#0000FFFFx	Invalid state of the function block

Tab. 3-18: Error codes of the ML_AxisRestore function block

3.3.4 ML_BACKUP_OPTIONS

Type definitions The following definition describes the enumeration type ML_BACKUP_OPTIONS used in the function block ML_AxisBackup ([ML_AxisBackup on page 35](#)).

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

Tab. 3-19: Definition of the enumeration elements of the ML_BACKUP_OPTIONS data type

3.3.5 ML_BACKUP_PARAMS

Type definitions The following definition describes the enumeration type ML_BACKUP_PARAMS used in the function blocks [ML_AxisBackup on page 35](#) and [ML_AxisRestore on page 37](#).

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. 3-20: Definition of the enumeration elements of the ML_BACKUP_PARAMS data type

3.4 Two-Point Controllers (Controller)

3.4.1 ML_TwoPosCtrlType01

Brief description The function block ML_TwoPosCtrlType01 implements a two-point controller with hysteresis and is suitable for simple closed-loop controls.

Assignment: Target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-21: Reference table of ML_TwoPosCtrlType01 function block

Interface description

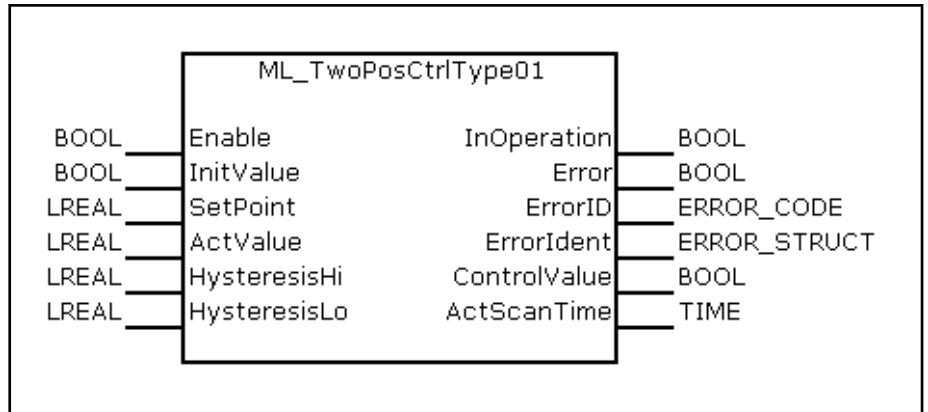


Fig. 3-16: ML_TwoPosCtrlType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Processing of function block enabled (cyclic, state-controlled)
	InitValue	BOOL	Specified value for the control variable y
	SetPoint	LREAL	Reference variable w (command value)
	ActValue	LREAL	Controlled variable x (actual value)
	HysteresisHi	LREAL	Upper response threshold of the hysteresis
	HysteresisLo	LREAL	Lower response threshold of the hysteresis
VAR_OUTPUT	InOperation	BOOL	Calculation of the control variable y completed without error, output variable "ControlValue" valid
	Error	BOOL	Calculation of the control variable "ControlValue" completed with error, output variable "ErrorIdent" valid
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	Diagnostic description in the event of an error Error handling on page 42
	ControlValue	BOOL	Controller output value – control variable y
	ActScanTime	TIME	Controller cycle time [ms]

Tab. 3-22: Interface variables of the ML_TwoPosCtrlType01 function block

Specification The function block ML_TwoPosCtrlType01 can be adjusted in its properties using the following input variables:

"HysteresisHi":

- The input has to be preset with a positive value or zero. Zero switches off the positive hysteresis.

"HysteresisLo":

- The input has to be preset with a negative value or zero. Zero switches off the negative hysteresis.



When calculating the control variable "ControlValue", "FALSE" is dominant. Thus, with "HysteresisLo" == "HysteresisHi" == zero with a control difference of zero ("SetPoint" == "ActualValue"), the output "ControlValue" is always "FALSE".

Functional description

After the function block has been enabled for processing, the function block ML_TwoPosCtrlType01 uses "Enable" to preset the internal controller state in the first controller cycle as well as the controller output to the preset value given at the "InitValue" input and it cyclically calculates the control variable "ControlValue" in the following cycles.

If there are no errors during the function block processing, the validity of the control variable is signaled by the "InOperation" output and the elements of the output structures "ErrorID" and "ErrorIdent" are not updated.

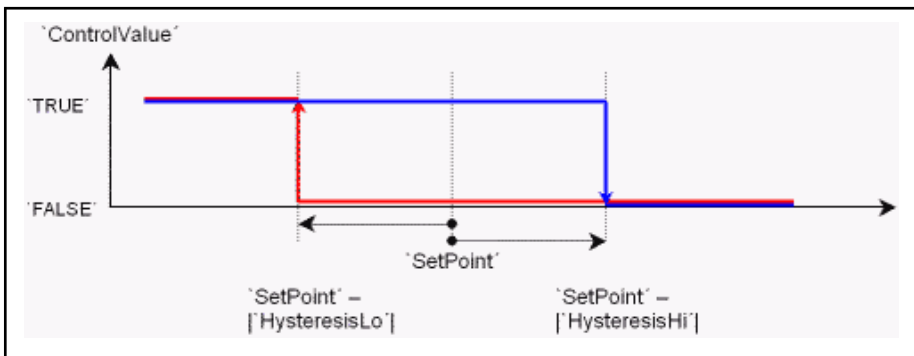


Fig. 3-17: Hysteresis of the ML_TwoPosCtrlType01 function block

If an error occurs during the function block processing, this is signaled by the "Error" output and specified by the elements of the output structures "ErrorID" and "ErrorIdent" which are updated in this case. The outputs "ControlValue" and "ActScanTime" are not updated in the event of an error.

Error handling

The function block uses the error table **MLC_TABLE**, 16#0030. In Additional1 and Additional2 it can generate the following error messages:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#F0260030	0	The inputs "HysteresisHi" or "HysteresisLo" are invalid
STATE_MACHINE_ERROR (16#0005)	16#F0260031	xx	Invalid state of the state machine xx indicates the invalid state

Tab. 3-23: Error codes of the ML_TwoPosCtrlType01 function block

3.5 Cam download (LoadCamData)

3.5.1 Introduction and overview

The "MB_LoadCamData" function block downloads cams to the drive and the control to encapsulate the existing differences in detail. The following "MB_CAMPROF" enumeration and the "MB_CAMTABLE" structure are used in the function blocks to create cam tables.

Also refer to

- [MB_LoadCamData](#), page 43

- [MB_CAMPROF, page 45](#)
- [MB_CAMTABLE, page 46](#)

3.5.2 MB_LoadCamData

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

Assignment: Target system/library

IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-24: Reference table of the MB_LoadCamData

Interface description

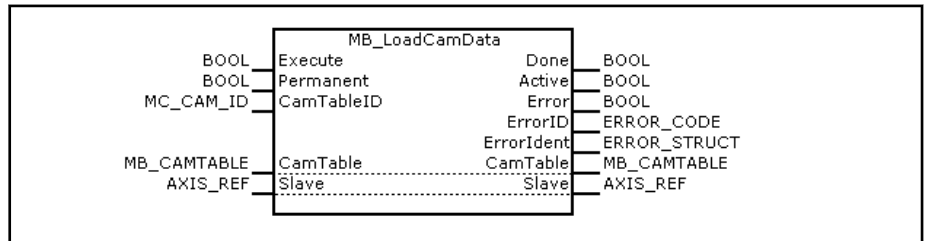


Fig. 3-18: MB_LoadCamData function block

I/O type	Name	Type	Comment
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements MB_CAMTABLE, page 46
	Slave	AXIS_REF	Reference selection (drive or control)
VAR_INPUT	Execute	BOOL	Processing of the function block 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 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 45)

Tab. 3-25: 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.

ML_TechBase.library

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. 3-26: Min./max. and default values of the MB_LoadCamData function block

The enumeration "MC_CAM_ID" 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. 3-27: Elements of the enumeration type MC_CAM_ID

The structure [MB_CAMTABLE](#), [page 46](#) 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

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 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 set parameter S-0-0269 "Saving mode" of the drive to value "1"(non-saving) if the function block is calling with the "Permanent"="false" input.

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 if "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. 3-28: Input assignments of the MB_LoadCamData function block to load the cam to the drive or control

Error handling

The function block uses the F_RELATED_TABLE, 16#0170x error table. 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. 3-29: Error codes of the MB_LoadCamData function block

3.5.3 MB_CAMPFROF

Type definitions

The enumeration selects between different cam profiles. The selections of CAMPFROF_STANDARD and CAMPFROF_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 CAMPFROF_STANDARD and CAMPFROF_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.

ML_TechBase.library

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. 3-30: 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
IrCamTable	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. 3-31: Min./max. and default values of the MB_CAMTABLE structure

3.6 Mathematical functions (MathematicalFunctions)

3.6.1 Introduction and overview

The following function blocks provide mathematical functions. The function block [ML_InterpolationLinear](#) on page 46 calculates intermediate points and linearizations for example.

The function block [ML_MaxValue](#) on page 48 outputs the maximum value of all values up to now at its output upon each call.

3.6.2 ML_InterpolationLinear

Brief description

The function block ML_InterpolationLinear cyclically calculates the y-value associated to a given x-value by linear interpolation based on a defined X-Y node table.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-32: Reference table of ML_InterpolationLinear function block

Interface description

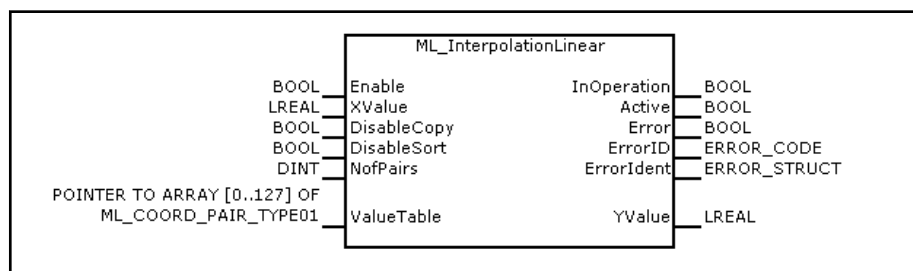


Fig. 3-19: ML_InterpolationLinear function block

I/O type	Name	Type	Description
VAR_INPUT	Enable	BOOL	Enabling the function block (cyclic, state-controlled)
	XValue	LREAL	X value
	DisableCopy	BOOL	Deactivating copying of the data point table
	DisableSort	BOOL	Deactivating sorting of the data point table
	NofPairs	DINT	Number of the connected value pairs in the data point table
	ValueTable	POINTER TO ARRAY [0..127] OF ML_COORD_PAIR_TYPE01	Address of the data point table ML_COORD_PAIR_TYPE01 on page 49
VAR_OUTPUT	InOperation	BOOL	Cyclic calculation of a Y value completed without error, output variable "YValue" valid
	Active	BOOL	Calculation of a Y value active, output variables invalid
	Error	BOOL	Calculation of a Y value completed with error, error outputs "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
	YValue	LREAL	Y value

Tab. 3-33: Interface variables of the ML_InterpolationLinear function block

Functional Description

The function block ML_InterpolationLinear cyclically calculates the Y value "YValue" associated with the input value "XValue" with "Enable" on basis of a specified data point table after the release for processing of the function block.

The data point table is saved on an internal data array depending on the inputs "DisableCopy" and "DisableSort" at a rising edge and sorted in an ascending order afterwards. The time for calculating an associated y-value "YValue" can be minimized if a data point table already sorted in an ascending order is used when the function block is enabled for processing and this is marked by the input "DisableSort". In this case, the data array is not sorted. If in a new processing of the function block, the internally available and already used data point table should be used again, the input "DisableCopy" as well as the input "DisableSort" have to be set to enabled.

The Y value to be calculated is stored on the output "YValue" in case of error-free processing of the function block, whereupon the validity of the value is signaled by the output "InOperation".

If it is not possible to provide the Y value to be calculated immediately after the activation of the function block, this is indicated by the output "Active".

If an error occurs while processing the function block, this is signaled by the "Error" output. The output "YValue" is not updated in case of an error.

ML_TechBase.library

Effect of the inputs "DisableCopy" und "DisableSort" on the runtime behavior of the function block

DisableCopy	DisableSort	Use case
FALSE	FALSE	An unsorted data point table is specified. It was not used like this in the previous calculation. The data point table is copied and sorted
FALSE	TRUE	A data point table already sorted in an ascending order is specified. It was not used like this in the previous calculation. The data point table is copied but not sorted internally
TRUE	FALSE	This combination is not a reasonable state
TRUE	TRUE	The data point table was used in previous calculations. Therefore, it is already known internally and sorted in an ascending order



Sorting the data point table after the activation of the function block might cause that the function block cannot deliver the desired output value in the first or the following cycle due to the restriction to a maximally admissible runtime. This duration at the beginning of the activation is signaled by "Active".

Troubleshooting

The function block uses the error table **MLC_TABLE**, 16#0030. It can generate the following error messages in Additional1 and Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#F0260030	0	Inputs invalid (XValue not within the data points)
STATE_MACHINE_ERROR (16#0005)	16#F0260031	xx	Invalid state of the state machine. xx indicates the invalid state

Tab. 3-34: Error codes of the ML_InterpolationLinear function block

3.6.3 ML_MaxValue**Brief description**

The function block ML_MaxValue determines the actual maximum value from the time dependent course of the input signal.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

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

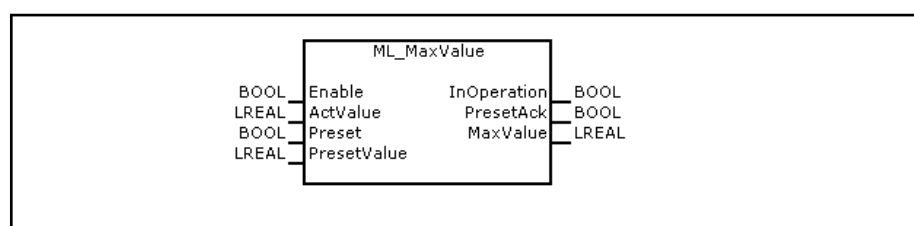
Interface description

Fig. 3-20: ML_MaxValue function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Processing of function block enabled (cyclic, state-controlled)
	ActValue	LREAL	Current value
	Preset	BOOL	Activation of preset value
	PresetValue	LREAL	Specified value

I/O type	Name	Data type	Description
VAR_OUTPUT	InOperation	BOOL	Determination of maximum value completed
	PresetAck	BOOL	Confirmation to apply "PresetValue" value
	MaxValue	LREAL	Maximum value

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

Functional Description

After the release of the function block, the function block ML_MaxValue uses "Enable" to compare the current value of the input variable "ActValue" to the value in "MaxValue". It copies the respectively higher value to the output variable "MaxValue".

If the input "Preset" is activated at function block runtime, the value of the input variable "PresetValue" is copied to the output variable "MaxValue". The input is state-controlled, i.e. as long as the "Preset" input is active, the "PresetValue" is copied to the "MaxValue" output variable.



If a "PresetValue" value is to be specified when the function is activated, the "Preset" input has to be set simultaneously with the "Enable" input.

Error codes The function block ML_MaxValue does not generate any errors.

3.6.4 ML_COORD_PAIR_TYPE01

This data structure defines a two-dimensional coordinate with X- and Y-LREAL values.

Type declaration

```
TYPE ML_COORD_PAIR_TYPE01 :
STRUCT
  X: LREAL := 0.0;           // X coord
  Y: LREAL := 0.0;           // Y coord
END_STRUCT
END_TYPE
```

Usage in

- [ML_InterpolationLinear, page 46](#)

3.7 Programmable limit switch (PhaseLimitSwitch)

3.7.1 Introduction and Overview

The function block for the application "Programmable limit switch" [MC_DigitalCamSwitch on page 49](#) is a PLCopen-compliant function block. It can be used to freely program up to 8 outputs with up to 32 switches.

The data structures used

- [MC_OUTPUT_REF on page 65](#)
- [MC_CAMSWITCH_REF with ML_BASIC_CAMSWITCH_REF on page 65](#)
- [MC_TRACK_REF with ML_BASIC_TRACK_REF on page 66](#)

also fulfill the PLCopen specifications.

3.7.2 MC_DigitalCamSwitch

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

ML_TechBase.library

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-37: Reference table of the MC_DigitalCamSwitch function block

Interface description

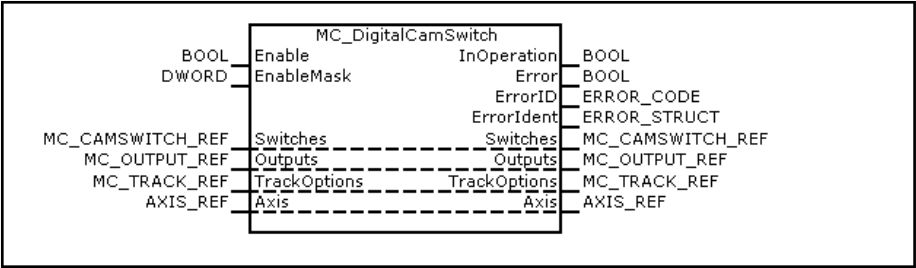


Fig. 3-21: MC_DigitalCamSwitch function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis the programmable limit switch is assigned to
	Switches	MC_CAMS- WITCH_REF	Reference to the switch structure containing all definitions for all switches. MC_CAMSWITCH_REF with ML_BASIC_CAMS- WITCH_REF on 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 on page 65
	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 with ML_BASIC_TRACK_REF on page 66
VAR_INPUT	Enable	BOOL	Enables the switching outputs
	EnableMask	DWORD	32 bits of the 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
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 on page 64
	Busy	BOOL	Is set when the function block is enabled (not in idle mode)

Tab. 3-38: 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

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

Tab. 3-39: Terminology of the programmable limit switch

ML_TechBase.library

- 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](#) with [ML_BASIC_TRACK_REF](#) on page 66, 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
 - [MC_CAMSWITCH_REF](#) with [ML_BASIC_CAMSWITCH_REF](#) on page 65, contains an array of 32 basic structures each describing a single switch
 - [MC_OUTPUT_REF](#) on page 65, contains an array with eight Boolean variables assigned to the outputs of the programmable limit switches
- 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 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:

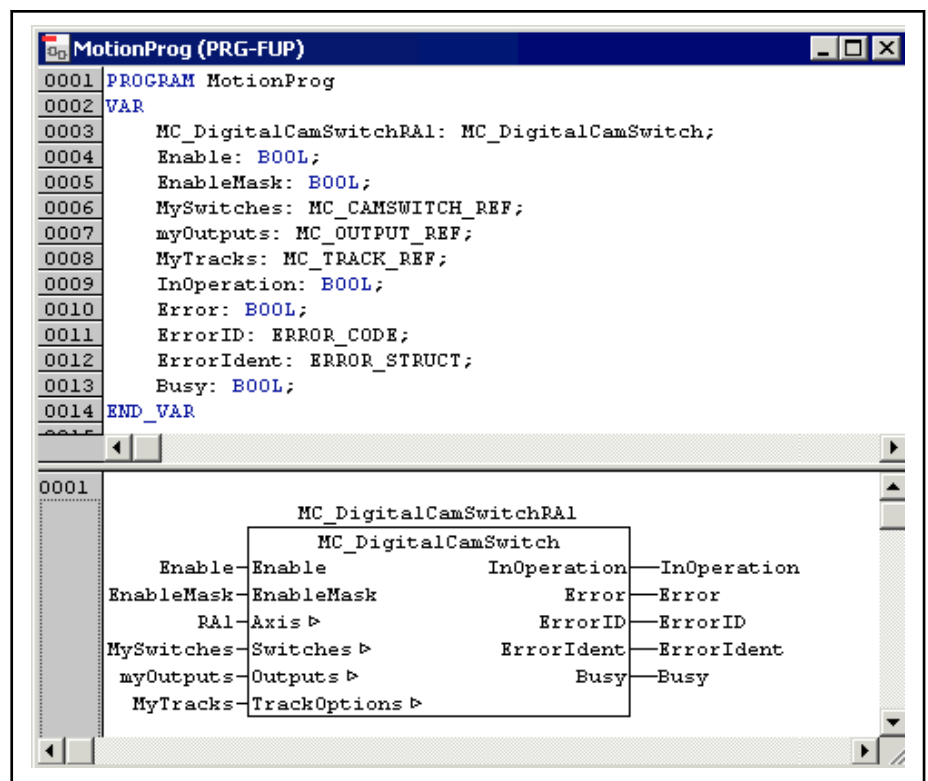


Fig. 3-22: Variable declaration and function block call

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 contents of the structures provided for the results of the example above:

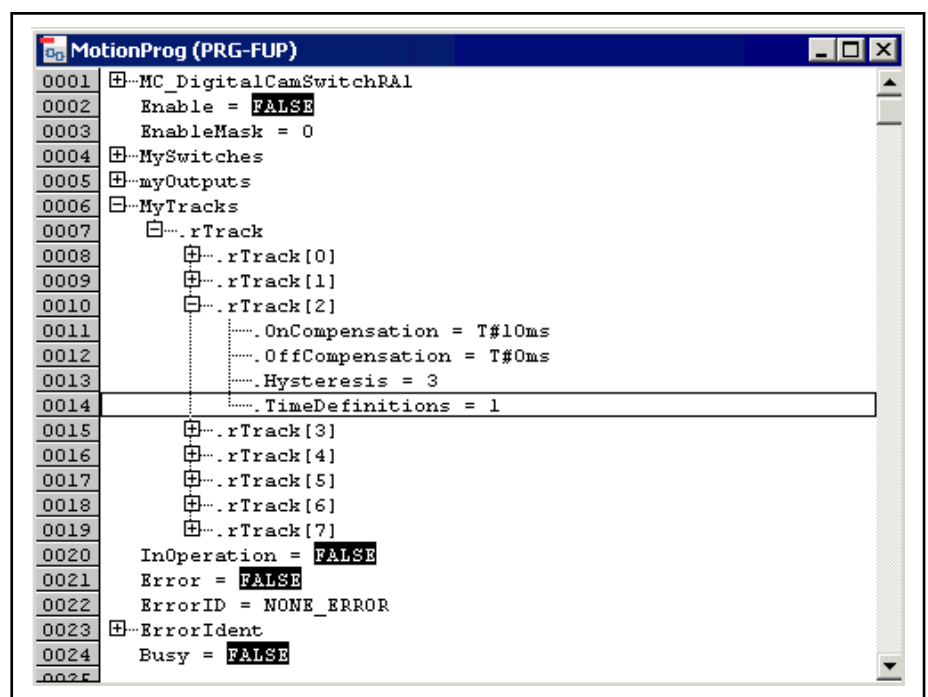


Fig. 3-23: Content of the "MyTracks" structure

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

ML_TechBase.library

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.

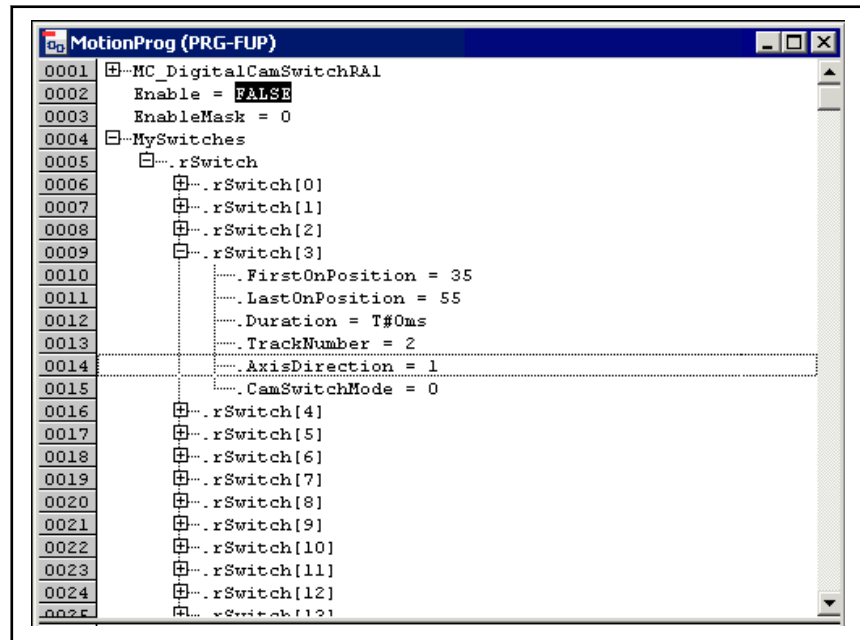


Fig. 3-24: Content of the "MySwitches" structure

The switch is defined (figure above).

1. Initialize the switch structure with index 3 to initialize switch 4, since the array starts with index 0.
2. "FirstOnPosition" is set to the 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 cam **rTrack[2]**.

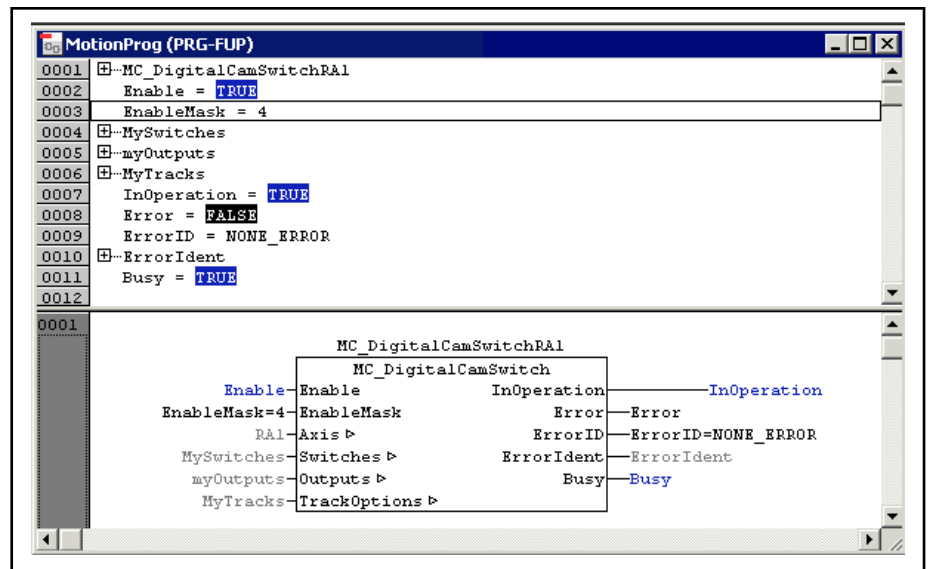


Fig. 3-25: Running function block

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, the "Busy" output remains set and the "InOperation" output becomes 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.

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 restriction 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 regardless whether the current position is located in the area of a switch. If "EnableMask" 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 PLS is only used for rotary processes. However, the PLS can also be used for a linear axis which runs forward and backward. In both cases, the PLS 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 traveled in positive direction. If the "AxisDirection" is set to 2, the output is only set if it is traveled 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 the "FirstOnPosition" and disabled after the specified "Duration".

Duration: Period the time-controlled switch is switched on. 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 maximum possible duration depends on the hardware. If the switch-on duration is exceeded, the error code 16#F0260006 is generated.

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, in each case 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 of all moves in positive direction over the PLS, afterwards in negative and finally in positive direction again. You can see 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.

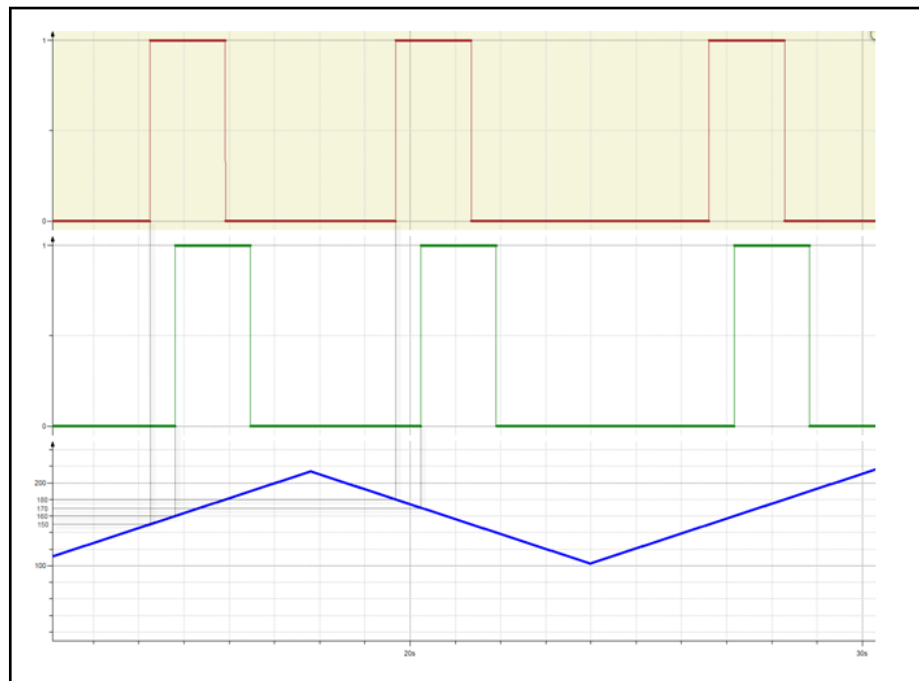


Fig. 3-26: Effect of the hysteresis on the switching points of a switch

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, "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:

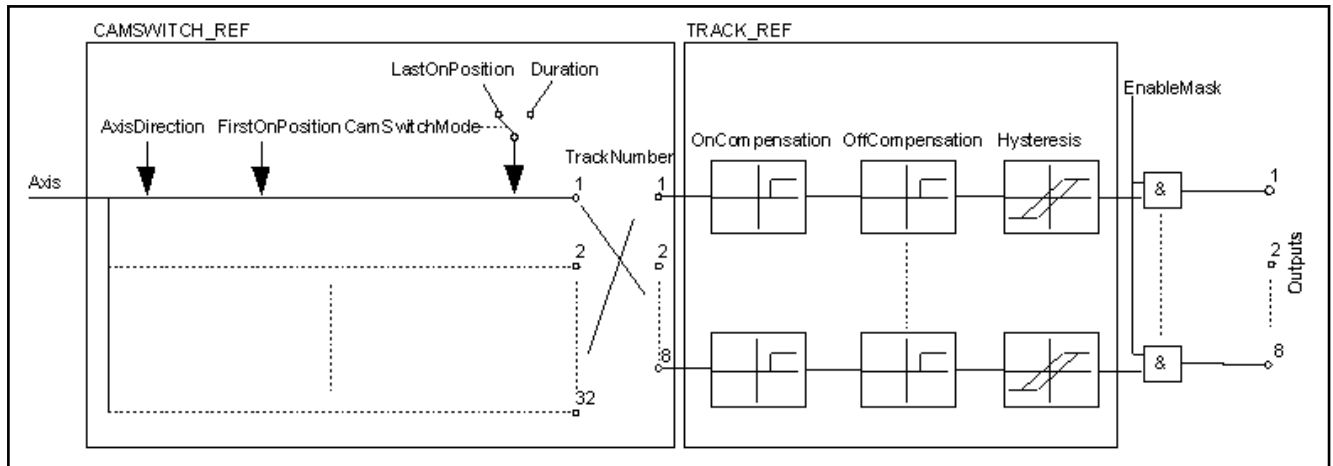


Fig. 3-27: Relation of the structures and the corresponding functionality of the MC_DigitalCamSwitch function block

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.

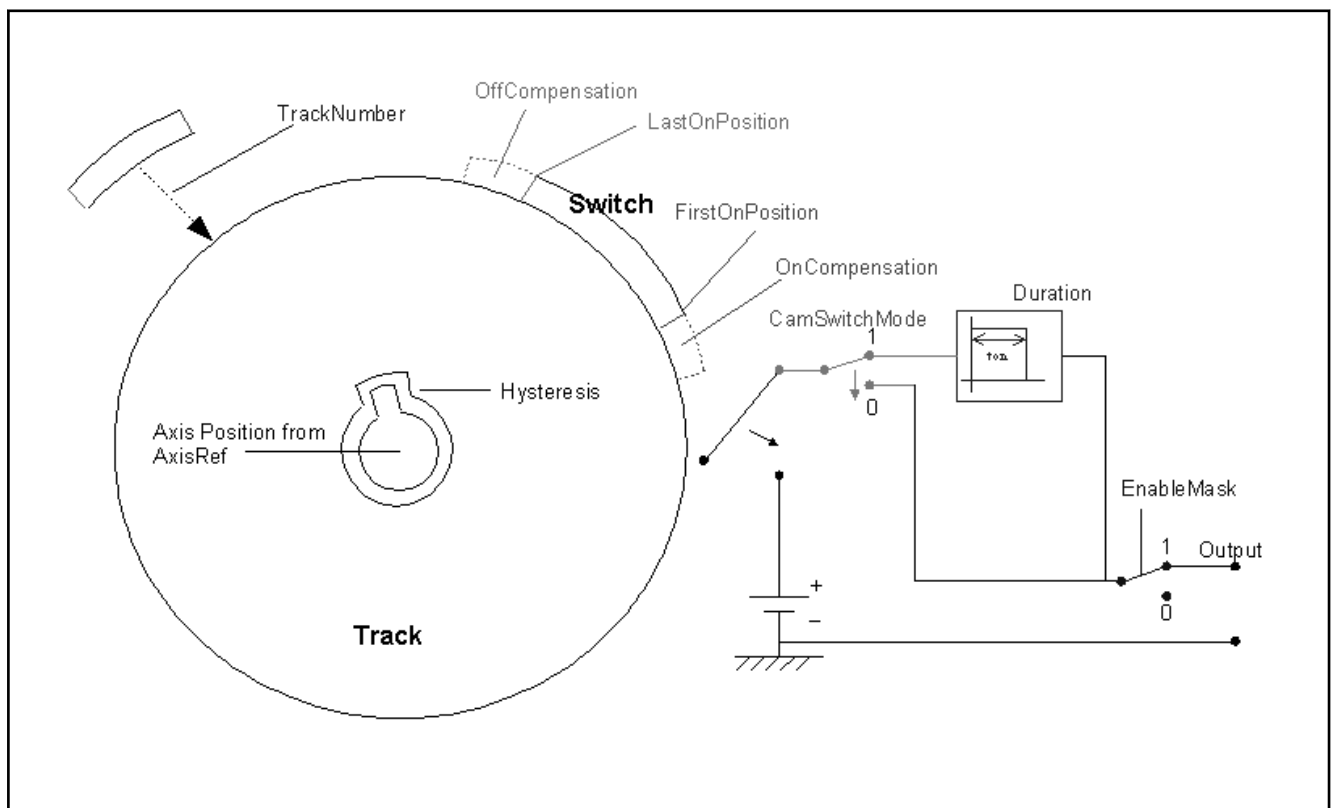


Fig. 3-28: Mechanical equivalent of the digital programmable limit switch

ML_TechBase.library

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

Parameter	Type	Switch01	Switch02	Switch03	Switch04	...	SwitchN
TrackNumber	INT	1	1	1	2		
FirstOnPosition [u]	REAL	2000	2500	4000	3000		
LastOnPosition [u]	REAL	3000	3000	1000	-		
AxisDirection	INT	1=Pos	2=Neg	0=Both	0=Both		
CamSwitchMode	INT	0=Position	0=Position	0=Position	1=Time		
Duration	TIME	-	-	-	1350		

Tab. 3-40: Example of the MC_CAMSWITCH_REF structure

This example uses the values from the example "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).

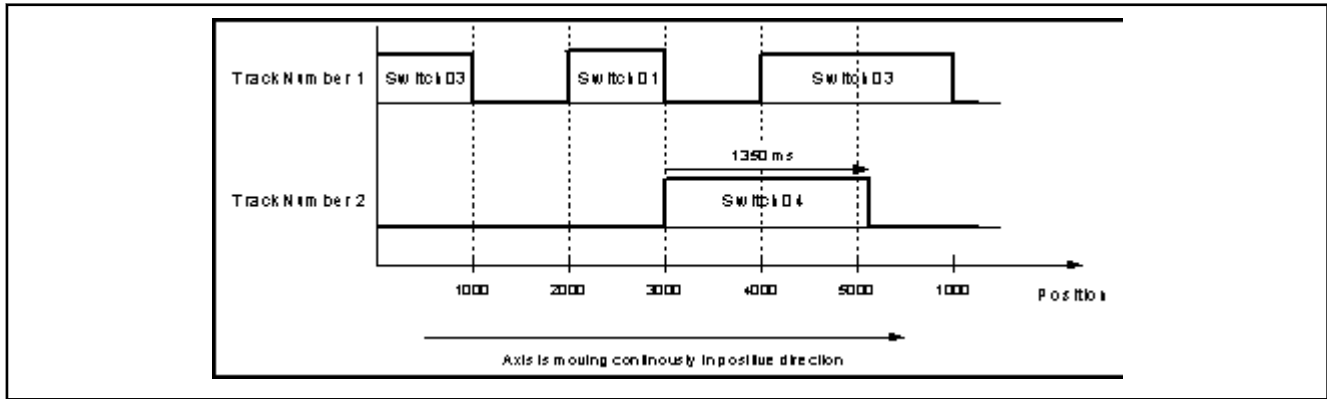


Fig. 3-29: Behavior of the outputs in the example

Detailed description of the switch "Switch01"

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

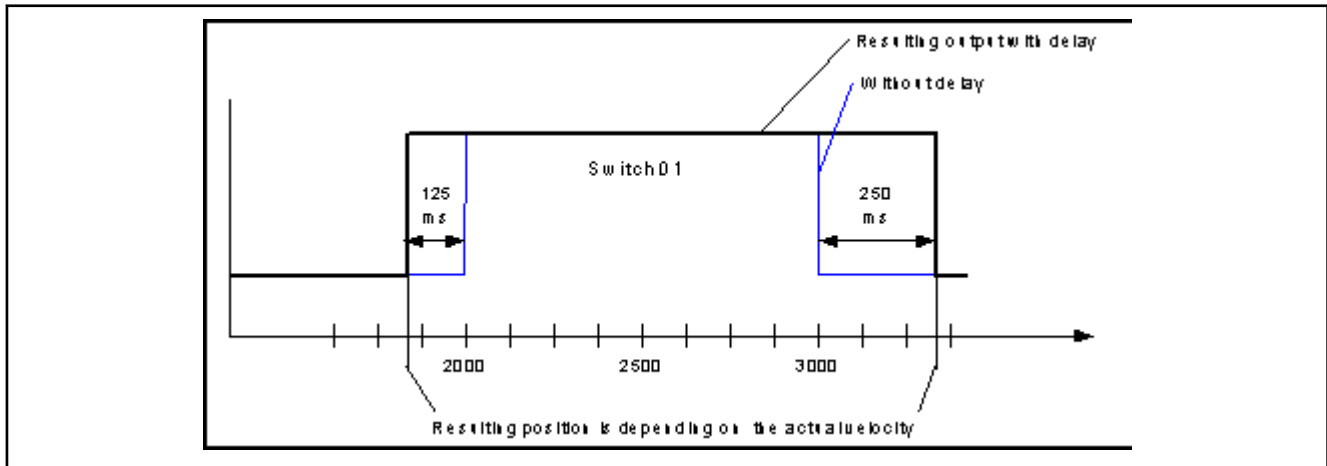


Fig. 3-30: Detailed behavior of Switch01 when moving in positive direction

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

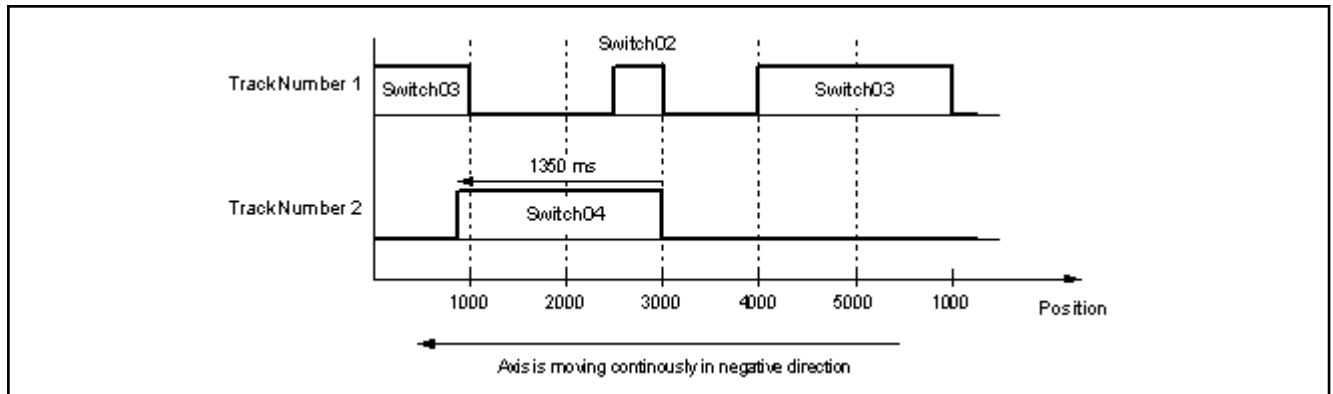


Fig. 3-31: Detailed behavior of the switches when moving in negative direction

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:

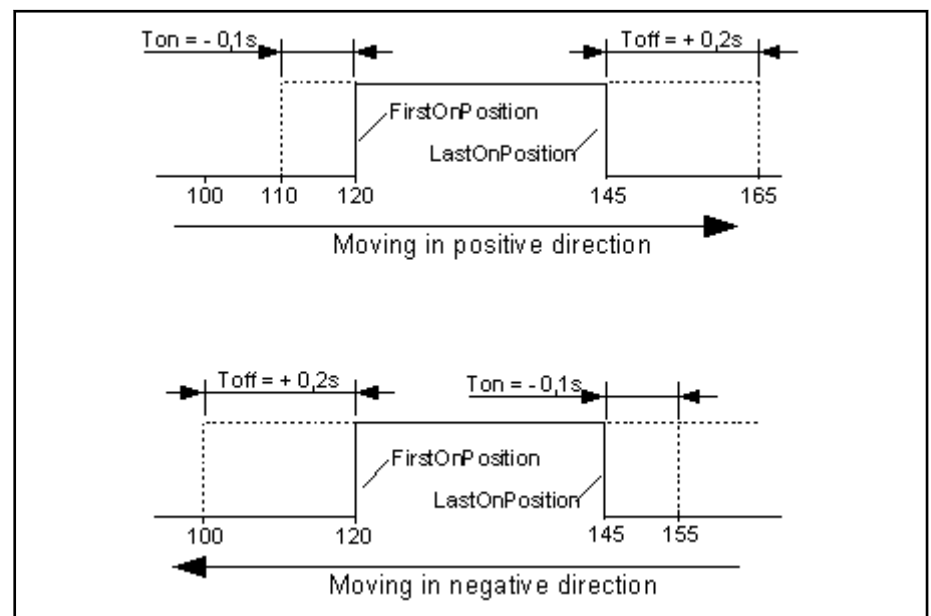


Fig. 3-32: Definition of the OnCompensation and OffCompensation time

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 have 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 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:

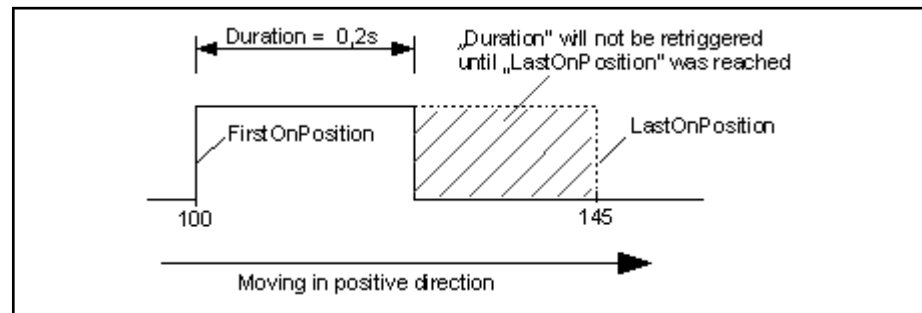


Fig. 3-33: Definition of the duration

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 retrigged (hatched area in the figure above).

In this application, both 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.

PLS restrictions

The restrictions of the PLS are discussed in this section. 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 switch, for which switching actions are lost, are calculated.

Rotary systems:

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

n refers to the revolutions 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 of 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 rate of rotations 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 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 min⁻¹.

Linear systems:

$$v = s / t,$$

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

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

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

or for the velocity:

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

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

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

or for the velocity:

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

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:

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

Switches seem to have the wrong size

The boundary conditions for "lost switching actions" have been 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 4ms (case A) depending on the sampling time of the PLS. A more detailed explanation can be found in the following figure

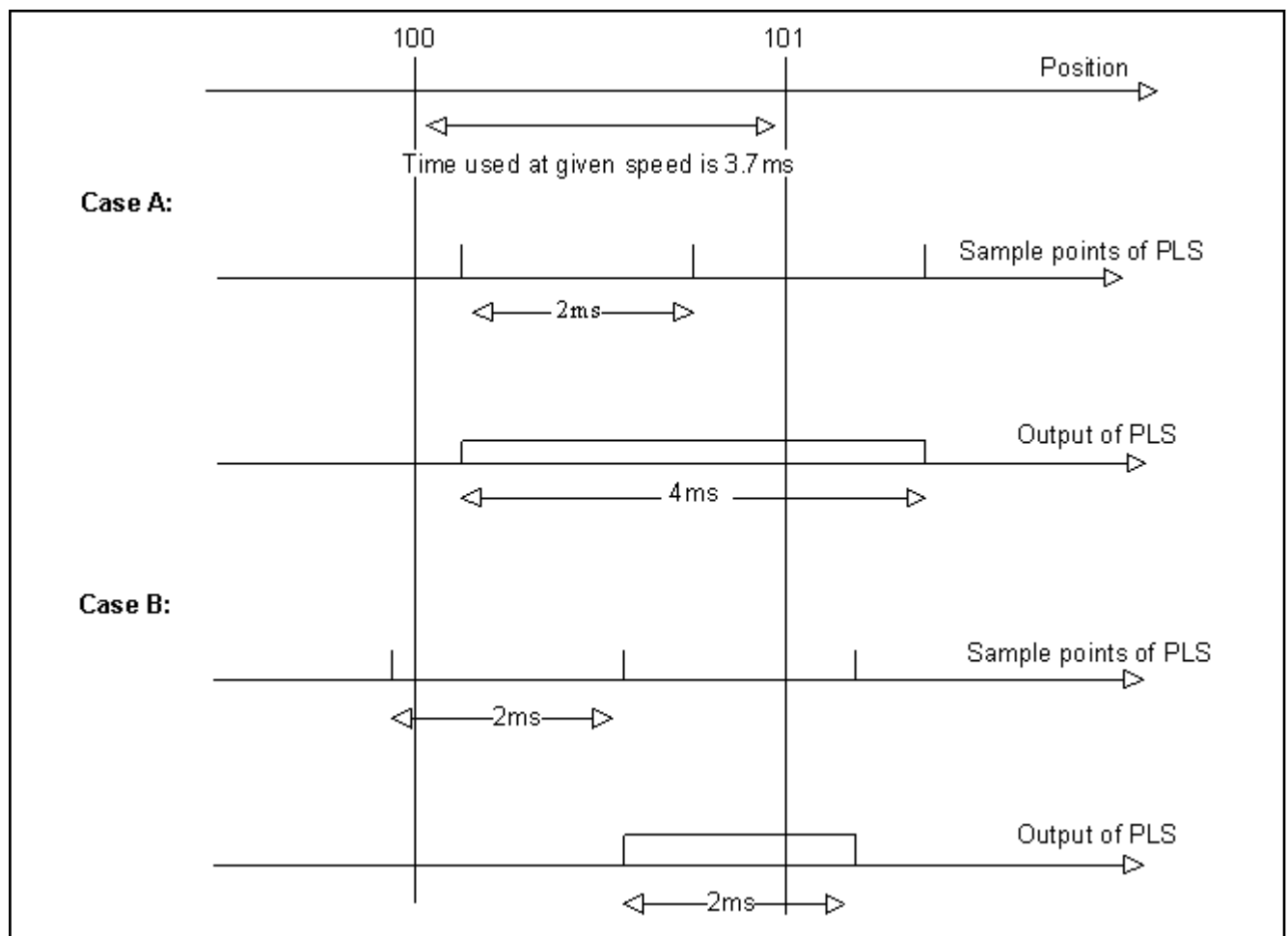


Fig. 3-34: Inexact switch sizes

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:

ML_TechBase.library

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

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.

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

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

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

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 are 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 **MLC_TABLE**, 16#0030.

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

ErrorID	Additional1	Additional2	Description
SYSTEM_ERROR (16#7FFF)	16#F0260001	0	Error while initializing the function block. No reference could be established to the parameter system
COMMUNICATION_ERROR (16#0002)	16#F0260004	xx	Runtime error in function block: Parameter could not be read xx refers to the parameter number (A-parameter)
INPUT_INVALID_ERROR (16#0001)	16#F0260005	0	Runtime error in function block: Invalid combination of units for velocity, time and position

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR (16#0001)	16#F0260006	0	Runtime error in function block: The value "Duration" of a switch is too high
STATE_MACHINE_ERROR (16#0005)	16#F0260007	0	Runtime error in function block: Invalid state of the state machine
RESSOURCE_ERROR	16#F0260040	0	Reference to the basic system could not be found
RESSOURCE_ERROR	16#F0260041	0	The function block could not be instantiated
SYSTEM_ERROR	16#F0260042	0	Function block may not be in the RETAIN data array

Tab. 3-41: Error codes of the MC_DigitalCamSwitch function block

3.7.3 MC_CAMSWITCH_REF with ML_BASIC_CAMSWITCH_REF

Type definitions The following definition describes data types used in the function block [chapter 3.7.2 "MC_DigitalCamSwitch" on page 49](#).

The type definitions are included in the library.

"MC_CAMSWITCH_REF" 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
TrackNumber	INT	"TrackNumber" is the reference for the switch track
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)
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)
Duration	TIME	Coupled with time-related "CamSwitchMode". I.e. the output remains on for the specified time

Tab. 3-42: Definition of the basic elements of the ML_BASIC_CAMSWITCH_REF structure

3.7.4 MC_OUTPUT_REF

Type Definitions The following definition describes data types which are used in the function block [chapter 3.7.2 "MC_DigitalCamSwitch" on page 49](#). The data type definitions are contained in the library.

The data type describes the 8 outputs of the cam switch.

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

ML_TechBase.library

Structural element	Type	Description
OUTPUT	ARRAY [0..7] OF BOOL	Outputs of the cam switch.

Tab. 3-43: Definition of the basic elements of the MC_OUTPUT_REF structure

3.7.5 MC_TRACK_REF with ML_BASIC_TRACK_REF

Type Definitions The following definition describes data types which are used in the function block [chapter 3.7.2 "MC_DigitalCamSwitch" on page 49](#). The data type definitions are contained in the library.

A cam disc is an element describing the characteristics of the cam switch related to an output.

The "MC_TRACK_REF" contains an array with 8 basic structures (i.e. 8 cam discs which are allocated to 8 outputs). The basic structure ("ML_BASIC_TRACK_REF") of a cam disc is as follows:

Structural element	Type	Description
OnCompensation	TIME	Lead time of the cam disc related to the time of turn-on.
OffCompensation	TIME	Lead time of the cam disc 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: Bit 0 set: "OnCompensation" time is negative; Bit 1 set: "OffCompensation" time is negative;

Tab. 3-44: Definition of the basic elements of the MC_TRACK_REF structure

Several cams can be allocated to every cam disc.

3.8 TemperatureController

3.8.1 Introduction and overview

The following function blocks control temperature-controlled systems.

Function block	Description
IL_TempControlType01, page 66	Controller for the temperature-controlled system with automatic offline controller setting
IL_TempControlType02, page 84	Controller for the temperature-controlled system with automatic online controller setting

Tab. 3-45: Function blocks to control temperature-controlled systems

3.8.2 IL_TempControlType01

Brief description The IL_TempControlType01 function block controls the temperature. It can determine the controller parameters either independently or with specifications. Special functions like monitoring, pause and manual mode can also be enabled.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-46: Reference table of the IL_TempControlType01 function block

Interface description

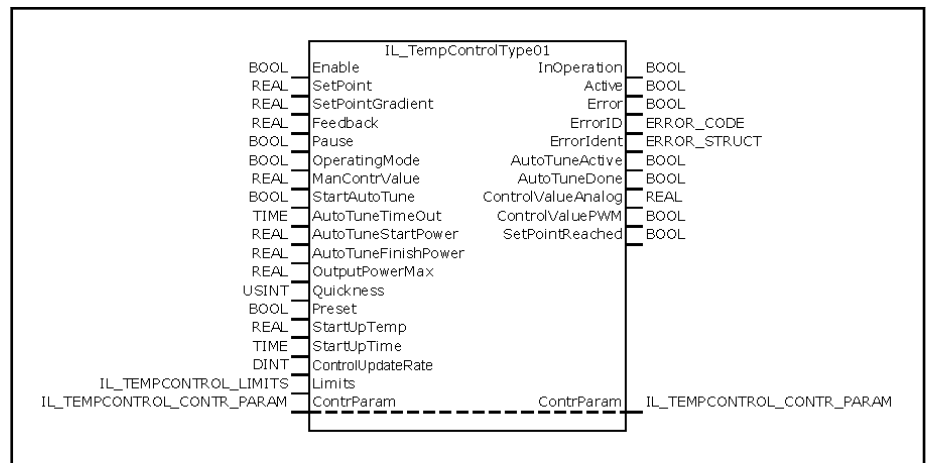


Fig. 3-35: IL_TempControlType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	The function block is enabled if this input is "TRUE". The function block is reset to its initial values with a falling edge. The input is read in each cycle
	SetPoint	REAL	Command temperature Unit in [u]* The input is read in each cycle
	SetPointGradient	REAL	Maximum parameter change "SetPoint" in [rps]. The input is read in each cycle. The preset value is "0 [rps]" (not active)
	Feedback	REAL	Input for the actual temperature Unit in [u] The input is read in each cycle
	Pause	BOOL	This input "freezes" the control variable and the controller calculation. The input is read in each cycle. The preset value is "FALSE" (not active)
	OperatingMode	BOOL	Sets the controller mode from automatic to manual mode: AUTO:=FALSE, MANUAL:=TRUE. The preset value is "FALSE" (AUTO). The input is read in each cycle
	ManContrValue	REAL	Specifies the control variable in [%] for the manual mode. Is only actively read if "OperatingMode:=1". The input is read in each cycle. The preset value is "0%" (no control variable)

ML_TechBase.library

I/O type	Name	Data type	Description
	StartAutoTune	BOOL	A rising edge starts the identification. The PID-controller parameters are calculated subsequently. They are not active until a falling edge is present. The input is read in each cycle. The preset value is "FALSE" (not active)
	AutoTuneTimeOut	TIME	Watchdog responsible for the time monitoring during the identification. The preset value is "10000 • T _A ". The input is read at a rising edge of "StartAutoTune".
	AutoTuneStartPower	REAL	Control variable in [%] representing the lower limit of the identification. The preset value is "0%". The input is read at a rising edge of "StartAutoTune".
	AutoTuneFinishPower	REAL	Control variable in [%] representing the upper limit of the identification. The preset value is "50%". The input is read at a rising edge of "StartAutoTune".
	OutputPowerMax	REAL	Maximum output control variable in [%]. The preset value is "100%". The input is read in each cycle
	Quickness	USINT	Quickness of the controller The preset value is "3" (very slight overshoot). The input can be selected in the range from "1" (quick control, strong overrun) to "5" (slow control, no overshoot). The input is read at a rising edge of "StartAutoTune".
	Preset	BOOL	A rising edge applies the controller parameters of the "ContrParam" structure. The preset value is "FALSE" (not active). The input is read during each cycle and becomes active only if "Enable" = TRUE
	StartUpTemp	REAL	Startup temperature in [u] The input is read at a rising edge of "StartAutoTune". The preset value is "0u" (not active). Becomes active if a value not equal to "0" is assigned to the "StartUpTime" input
	StartUpTime	TIME	Startup time, i.e. the period during which the startup temperature ("StartUpTemp") is not changed until the "normal" closed-loop control is started. The preset value is "T#0s" (not active). The input is read at a rising edge of "Enable"

I/O type	Name	Data type	Description
	ControlUpdateRate	DINT	This factor specifies how much slower the controller works compared to the task of the function block. The sampling time results from multiplying the cycle time (= task interval) with "ControlUpdateRate"
	Limits	IL_TEMPCON- TROL_LIMITS	Structure containing the alarm limits in [u] (units). The preset value is always "0u" (not active). The input is read in each cycle. If values are assigned to the structure, Limits.UpperAlarm has to be > Limits.LowerAlarm and (Setpoint + Limits.UpperAtSetpoint) has to be >= (Setpoint - Limits.LowerAtSetpoint). Otherwise, an error is generated IL_TEMPCONTROL_LIMITS, page 95
VAR_OUTPUT	InOperation	BOOL	The temperature control is active if an output is set
	Active	BOOL	This output is set if "Enable" is set and if the controller is active
	Error	BOOL	An error occurred at runtime. The control variable is set to "0%"
	ErrorID	ERROR_CODE	If the "Error" output is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, this output contains detailed error information
	AutoTuneActive	BOOL	Identification is active
	AutoTuneDone	BOOL	Identification process completed successfully
	ControlValueAnalog	REAL	Analog value of the control variable in %
	ControlValuePWM	BOOL	Digital, pulse-width modulated control variable value; either "0" or "1"
	SetPointReached	BOOL	A status output indicates whether the actual value is within the limits ("Setpoint" + "Limits.UpperAtSetpoint") >= actual value >= (Setpoint - "Limits.LowerAtSetpoint")
VAR_IN_OUT	ContrParam	IL_TEMPCON- TROL_CONTR_PAR- AM	This structure contains the current parameters K_p , K_i and K_D of the controller as well as the dead time T_t IL_TEMPCONTROL_CONTR_PARAM, page 93

*

The unit [u] means "unit". This can be any unit such as [°C] or [°F]. Ensure that the selected unit is always used.

Tab. 3-47: Interface variables of the IL_TempControlType01 function block

Time diagram

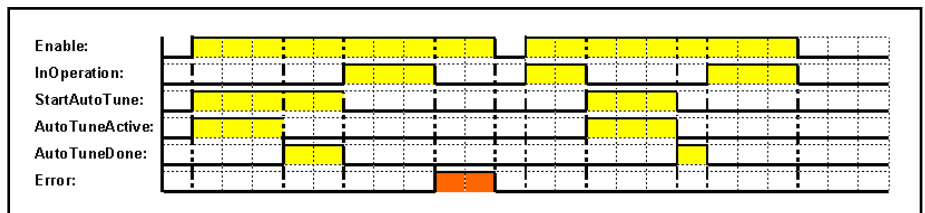


Fig. 3-36: Time diagram

Min./max. and default values

The following table lists the min./max. and default values as well as the transfer behavior of the function block inputs:

ML_TechBase.library

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL	FALSE	TRUE	FALSE	Cyclic
SetPoint	REAL	N. def.	N. def.	0.0	Cyclic
SetPointGradient	REAL	0.0	N. def.	0.0	Cyclic
Feedback	REAL	N. def.	N. def.	0.0	Cyclic
Pause	BOOL	FALSE	TRUE	FALSE	Cyclic
OperatingMode	BOOL	FALSE	TRUE	FALSE	Cyclic
ManContrValue	REAL	0.0	100.0	0.0	Cyclic
StartAutoTune	BOOL	FALSE	TRUE	FALSE	Cyclic
AutoTuneTime-Out	TIME	T#0s	"10000 • T _A ".	"10000 • T _A ". (is automatically set when T#0s is set)	At a rising edge at "StartAutoTune"
AutoTuneStart-Power	REAL	0.0	95.0	0.0	At a rising edge at "StartAutoTune"
AutoTuneFinish-Power	REAL	5.0	100.0	50.0	At a rising edge at "StartAutoTune"
OutputPowerMax	REAL	>0.0	100.0	100.0	Cyclic
Quickness	USINT	1	5	3	At a rising edge at "StartAutoTune"
Preset	BOOL	FALSE	TRUE	FALSE	Cyclic
StartUpTemp	REAL	N. def.	N. def.	0.0	At a rising edge at "Enable"
StartUpTime	TIME	T#0s	N. def.	T#0s	At a rising edge at "Enable"
ControlUpdateRate	DINT	1	1000	100	At a rising edge at "Enable"
Limits	IL_TEMPCONTROL_LIMITS	See "Data structure"	See "Data structure"	See "Data structure"	Cyclic
ContrParam	IL_TEMPCONTROL_CONTR_PARAM	See "Data structure"	See "Data structure"	See "Data structure"	Data is written by the autotune process at a rising edge at "AutoTuneDone". Data is applied at a rising edge at "Preset".

Tab. 3-48: Min./max. and default values as well as the transfer behavior of the inputs of the function block IL_TempControlType01

Application The function block can be used for the IndraLogic XLC and IndraMotion MLC using the Inline temperature modules (IL TEMP 2 RTD). Each value can be used as temperature value, since the temperature controller does not directly access the hardware. Thus, it is also possible to obtain temperature values via field bus connections for example.

Identification The automatic identification, also known as "AutoTune", identifies the parameters of the controlled system to determine the parameters of the controller.

First, certain model parameters are calculated. Subsequently, the model is compared to the real controlled system to avoid incorrect parameterizations of the controller parameters.

Before the jump is activated, it has to be ensured that the controlled system is in passive state. Therefore, a time monitoring is started. The preset value for this time monitoring is 5,000 multiplied with the root of the sampling time in [ms]. The actual value must not exceed the tolerance limits of the defined tolerances up to the end of the time monitoring. Otherwise, the time monitoring is restarted and the system waits until the specified timeout. The limit for the temperature window containing the actual value for the time of the time monitoring is calculated as follows:

$$|\text{IdentTemp} - \text{ActTemp}| < \sqrt{\frac{|\text{IdentTemp}|}{10}} + 0.5$$

The variable "IdentTemp" is the value around which the tolerance window of the actual temperature is created. Waiting for the stationary final value is performed identically.

Example:

Estimation of the time required for identification

It is assumed, that the function block is operated in a PLC task with a cycle time of 10ms. The "ControlUpdateRate" input is connected to 100. What is the minimum time for the "Autotune" process?

The sampling time can be calculated multiplying the PLC task cycle time and the "ControlUpdateRate": $T_A = \text{PLC task cycle time} * \text{"ControlUpdateRate"} = 10\text{ms} * 100 = 1000\text{ms}$

Use the formula above to calculate the waiting time: $\text{Waiting time} = 5000 * \text{SQRT}(T_A) = 158114\text{ms}$ or 2min and 38s. As it has to be waited for double the period for a jump response, a total waiting time of at least 5min and 16s results. The actual time required to reach the stationary state has to be added.

The quality of the determined model is estimated. If the deviation is too big, the function block outputs the error

"16#1603" (CALCULATION_ERROR)

with the identifier "16#0008" (invalid process model).

Setting the controller parameters

Quickness = 1...5;

1 = very fast control with large overshoot;

5 = very slow control without overshoot;

Preset value is quickness = 3.

The calculated controller parameters are entered into the provided structure "IL_TEMPCONTROL_CONTR_PARAM".

The entry

KP	corresponds to the proportional gain factor
KI	corresponds to the integration constant
KD	corresponds to the differentiation constant and
Tt	corresponds to the dead time

If these parameters are specified by the user, at least an estimated value is to be entered for the dead time to avoid the error

"Error in the control loop, sensor possibly displaced or destroyed"

(Additional1 = 16#1602; Additional2 = 16#0006),

Control algorithm An additive PIDT₁ algorithm with anti-wind-up of the I-value and PT₁ filtering of the D-part is implemented. The additive form allows to disable certain parts of the controller. Thus, it can be switched to the structure of a PI-controller by parameterizing the D-value to "zero" for example.

The I-value is limited using an anti-wind-up delimiter which is internally preset to 90 % of the control variable.

The saturation limit of the controller is the maximum output value (VAR_IN OutputPowerMax) of the controller. The preset value is 100%.

Controller structure In the following, the PIDT₁ controller structure with anti-wind-up limitation and limitation of the maximum control variable are displayed. Due to the additive structure, the values of the controller can be individually disabled. Thus, pure P-, P-I or PDT₁-controllers can be parameterized as well. However, the individual values have to be disabled by the user. For this purpose, the desired values have to be entered into the "contrParam" structure. At a rising edge at the "Preset" input, these new parameters are enabled in the controller. By default, the controller works with the complete PIDT₁ structure.

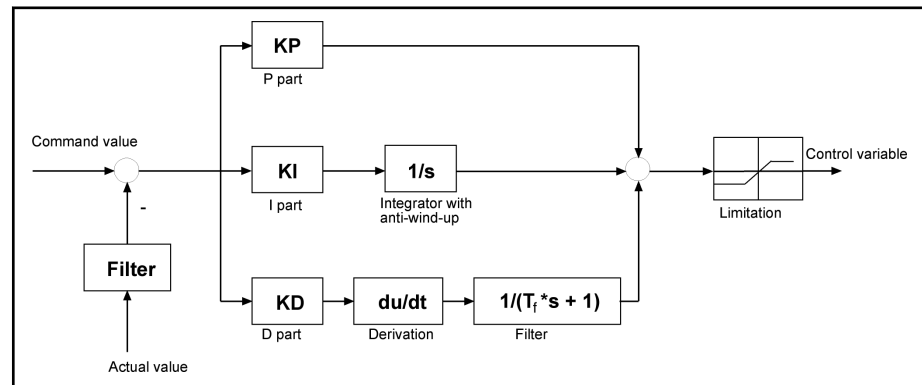


Fig. 3-37: Controller structure

The actual temperature value is smoothed with a PT₁ filter during the identification and during the control to avoid stochastic failures. The time constant of the filter is fixed to 20 times of the cycle time.

Control loop monitoring The function block is provided with a control loop monitoring. If a control variable of "100%" is output (full power), the temperature has to change in the direction of the sign set in K_p. Otherwise, a timer is enabled displaying an error at timeout. The timer is reset as soon as the temperature moves into the correct direction. The timer is assigned with the following values:

If an I-value exists, $(2000/K_i) + \text{dead time}$ is used. If no I-value is used, only $2000 * \text{sample time}$ is preset. The timer is reset even if the control variable becomes again less than "100%". It is important to specify the dead time (T_t) in the structure for the controller parameters as well if the controller parameters are specified by the "Preset" input. If K_i is high and there is a high dead time, monitoring could be triggered otherwise. Due to this monitoring, sensor errors (e.g. by being shifted) or actuating element errors can be detected.

In addition, the function block also monitors the gradient of the actual value. However, this monitoring function becomes only active in case of large devia-

tions. There can be deviations for example if the sensor displays an error code and if this error code is not caught/evaluated by the user. The gradient is fixedly set to 10 rms. An exceedance is tolerated once before an error is generated.

Using the function block

The function block should be operated in a cyclic task. The function block automatically detects the task interval (= cycle time) and adjusts the control algorithm accordingly. The controller works with the sampling time resulting from the task interval multiplied with the input variable "ControlUpdateRate". If "ControlUpdateRate" = 1, the sampling time corresponds to the task interval. The sampling time is used when identifying the controlled system and if the controller is enabled. All other functions of the function block (reading actual values, gradient monitoring, PWM output, manual closed-loop control) use the task interval. Thus, the control/identification process can be separated from other functions of the function block (with regard to time). This is especially important to separate the output of the control variable from the control/identification process using the PWM.

The pulse-width modulation (PWM) is based on the task interval, i. e. the task interval is the smallest possible time slice which can be set to "ON" or "OFF" during the PWM. One PWM period is 20 times longer than the task interval. Thus, the control variable can be changed in steps of 5 % between 0 % and 100 %.

Example:

The task interval is parameterized to 20 ms for the cyclic task. This means that a new value for the control variable can be transferred to the PWM every 400 ms. Thus, the possible switch-on/switch-off times for the PWM are:

Control variable [%]	Switch-on period [ms]	Switch-off period [ms]
0	0	400
5	20	380
10	40	360
...		
50	200	200
...		
95	380	20
100	400	0

Tab. 3-49: Example of switch on/off times

The task interval and "ControlUpdateRate" always have to be selected with regard to the control element and the controlled system. If the task intervals are too short, the control element wears very quickly (such as relay modules). If the task intervals/sampling times are too long, the accuracy of the closed-loop control reduces.

Furthermore, the sampling time generating and error if it exceeds/falls below by more than 20 % an setting the "Error" output to TRUE is monitored.

Recommendation for the task interval selection

Set the task interval/cycle time to ensure that the PWM does not cause any visible temperature changes in a PWM period of 20*task interval. If, the control variable is 20 % for example, the temperature of the controlled system strongly increases during the heating and reduces while the heating is switched off. This means that the task interval is too long. In contrast, it is also not

ML_TechBase.library

Recommendation for the sampling time	recommended to choose a task interval that is too short, since the control element is subject to higher loads (e. g. the relay module has to switch very often). Tests determine an appropriate task interval. First, select the task interval as described in the previous section. The sampling time can be set to the desired value using the "ControlUpdateRate" input. The following sampling time is recommended: • 10 ms for quick/dynamic controlled systems (time constant for the controlled system from 200 ms to 2 s) • 100 ms for average controlled systems (time constant for the controlled system from 2 s to 20 s) • 1 s to 30 s for slow/static controlled systems (time constant for the controlled system from 20 s to 100 min). The sampling time should be approximately 20 to 200 times faster than the "dominating time constant" of the controlled system. The "dominating time constant" is approximately the time a controlled system requires to pass 63% of an added jump. However, the sampling time can be freely selected by the user. If the selected sampling time is too short, the error "Time overflow during identification (watchdog)" (Additional1 = 16#1605; Additional2 = 16#0009) can occur during the identification. In this case, a longer sampling time has to be selected, since the controlled systems was not in steady state when the time monitoring elapsed. If the selected sampling time is too long, the error "Identified controlled process parameters invalid, self-identification of the controlled process failed" (Additional1 = 16#1603; Additional2 = 16#0007) can occur during the identification. In this case, select a shorter sampling time, since the intervals the controlled system behavior was measured were too long.
Recommendation for the identification	For the identification, some rules are to be observed to achieve good results. These are explained in detail in the following. In general, temperature-controlled system are not linear. For low temperatures, less power is required to achieve a certain temperature increase than at higher temperatures. At low temperatures, a controller needs different settings than at high temperatures. Thus, it is particularly important to carry out the identification as well in the temperature range in which the controller is to operate later on. There are two options: 1. For a very precise closed-loop control in a small temperature range, the power required in the manual mode is to be specified first so that the desired temperature results. Subsequently, perform the identification in which the value for "AutoTuneStartPower" is a bit lower than the calculated value and the value for "AutoTuneFinishPower" is a bit higher than the calculated value. The controller parameters are determined precisely for this temperature range. However, outside the temperature range, the closed-loop control is less precise. 2. For a wide closed-loop control range, the power required for the lowest and highest temperature in manual mode is to be determined. These are used as values for "AutoTuneStartPower" and "AutoTuneFinishPower". The closed-loop control is averaged over the entire temperature range. In general, this results in a more dynamic control at low temperatures and more diminished closed-loop control at higher temperatures.

Describing the individual functions

With regard to the following use cases, the variables have to be specified in the declaration part of the user program to assign the function block.

Program:

```
VAR
fb_TempControl:IL_TempControlType01;      (* Instance of the function blocks *)
strContrParam: IL_TEMPCONTROL_CONTR_PARAM; (* Structure containing the controller parameters *)
strLimits: IL_TEMPCONTROL_LIMITS;         (* Alarm/status limits *)
rActualTemp:REAL;                         (* Actual temperature*)
rActualPower:REAL;                        (* Control variable, analog *)
bActualPower:BOOL;                        (* Control variable, digital (PWM) *)
bAutoTuneDone:BOOL;                       (* Identification state *)
bSetPointReached:BOOL;                    (* Actual value is within the command value window *)
bError:BOOL;                              (* State of the error output *)
iState:INT;                               (* Value in SFC *)
END_VAR
```

Simple closed-loop control with a controller parameter specification

For an easier representation, no errors are considered.

In the "strContrParam" structure, the function block is parameterized with the desired controller parameters. At a rising edge at the "Preset" input, new controller parameters can also be parameterized during the closed-loop control. The command value ("Setpoint") has to be preset and the actual value ("Feedback") has to be returned. The control variable can either be retrieved at the analog output ("ControlValueAnalog") or at the digital output ("ControlValuePWM").

Program:

```
(* Setting the controller parameters *)
strContrParam.KP := 1.5;
strContrParam.KI := 2;
strContrParam.KD := 0.6;
strContrParam.Tt := T#1s;

(* Starting the function block at simultaneous transfer *)
(* of the controller parameters *)
fb_TempControl( Enable:=TRUE,
                Setpoint:=50,
                Feedback:=rActualTemp,
                Preset:=TRUE,
                ContrParam:=strContrParam,
                ControlValueAnalog=>rActualPower,
                ControlValuePWM=>bActualPower);
```

The controller parameters are only applied once and the controller works with the specified values.

Simple identification of a controlled system with a subsequent transition to the closed-loop control

First, the function block is set to the identification or AutoTune mode. After the successful identification, the closed-loop control is started by generating a falling edge at the input "StartAutoTune".

The calculated controller parameters are saved in the "strContrParam" structure and are thus available to the user. These controller parameters can be saved in the retain area of the control (refer to the example [Retain backup of the controller parameters, page 80](#)). Another identification is thus no longer required when starting the closed-loop control again. The simple control with parameter specification can be started immediately.

With the optional "Quickness" input, a slightly quicker setting was specified for the controller (quicker control but higher possibility for overshoots).

The identification is performed with a jump from 0 % ("AutoTuneStartPower") to 10 % ("AutoTuneFinishPower") of the output power

Program:

```
CASE iState OF
1: (* Identification *)
    fb_TempControl( Enable:=TRUE,
```

ML_TechBase.library

```

        StartAutoTune := TRUE
        Feedback:=rActualTemp,
        AutoTuneStartPower:= 0,
        AutoTuneFinishPower:= 10,
        Quickness:=2,
        ContrParam:=strContrParam,
        ControlValueAnalog=>rActualPower,
        ControlValuePWM=>bActualPower,
        AutoTuneDone=>bAutoTuneDone);
    IF (bAutoTuneDone = TRUE) THEN
        iState := 2;
    END_IF

2: (* Transition to closed-loop control *)
    fb_TempControl( Enable:=TRUE,
                    StartAutoTune := FALSE,
                    Setpoint:=50,
                    Feedback:=rActualTemp,
                    ContrParam:=strContrParam,
                    ControlValueAnalog=>rActualPower,
                    ControlValuePWM=>bActualPower);

```

Identifying the controlled system in the operating point

During a process control, another identification in the operating point can be desired. In the following, the corresponding procedure is described based on the previous example "Simple identification of a controlled process with subsequent transition to a closed-loop control". Setting the "AutoTuneStart" input starts the identification. The initial and final parameters of the control variable parameters are transferred to "AutoTuneStartPower" and "AutoTuneFinishPower". After a successful identification, the new controller parameters are available in the "strContrParam" structure.

In the example, the operating range is calculated based on the actual value of the control variable (in %). The prerequisite is a controlled system in a steady (stationary) state. In the example, the operating range is selected from "control variable -10 %" to "control variable +10 %". The identification in the operating point starts when changing to state "3".

Program:

```

CASE iState OF
2: (* Closed-loop control *)
    fb_TempControl( Enable:=TRUE,
                    StartAutoTune := FALSE,
                    Setpoint:=50,
                    Feedback:=rActualTemp,
                    ContrParam:=strContrParam,
                    ControlValueAnalog=>rActualPower);

    If (bIdent = TRUE) THEN
        iState := 3;
    End_IF;

3: (* Identification in the operating point *)
    fb_TempControl( Enable:=TRUE,
                    StartAutoTune := TRUE,
                    AutoTuneStartPower:=rActualPower-10,
                    AutoTuneFinishPower:=rActualPower+10,
                    Feedback:=rActualTemp,
                    ContrParam:=strContrParam,
                    ControlValueAnalog=>rActualPower,
                    AutoTuneDone=>bAutoTuneDone);

    IF (bAutoTuneDone = TRUE) THEN
        iState := 2;
    END_IF

```

Defining a startup ramp

If a process prior to the actual temperature control requires a pause at a certain temperature for a certain period (startup ramp), this function is available via the inputs "StartUpTemp" and "StartUpTime".

First, the controller controls until the startup temperature. If this temperature is not achieved, the startup time "StartUpTime" elapses and the controller keeps the startup temperature constant. After the startup time "StartUpTime"

has elapsed, the startup ramp is completed and the controller shows its normal behavior. The startup ramp is executed once after each change from "Enable" to "TRUE".

In the example, the controller controls to a temperature of "100". Then, it waits for 5 minutes and 30 seconds and controls it to the desired temperature of "150" (see [Using a command value gradient and a startup ramp, page 79](#)).

Program:

```
fb_TempControl(    Enable:=TRUE,
                  Setpoint:=150,
                  Feedback:=rActualTemp,
                  Preset:=TRUE,
                  StartUpTemp:=100,
                  StartUpTime:=t#5m30s,
                  ContrParam:=strContrParam,
                  ControlValueAnalog=>rActualPower);
```

Defining alarm and status limits

Alarm and status limits can be created for process monitoring purposes.

The "Limits" structure contains the values "UpperAlarm" and "LowerAlarm". When exceeding the alarm limits, an error is generated. The monitoring function becomes active if the current process temperature ranges between the values of "UpperAlarm" and "LowerAlarm" in order not to generate an error during process start. These values are absolute values.

The "Limits" structure also contains the values "UpperAtSetPoint" and "LowerAtSetPoint". The "SetPointReached" output is set as soon as the condition $(\text{Setpoint} - \text{Limits.LowerAtSetPoint}) \leq \text{actual value} \leq (\text{Setpoint} + \text{Limits.UpperAtSetPoint})$ is met. These values are relative values that relate to the "Setpoint". The "Setpoint" can thus be modified without changing "Limits.LowerAtSetPoint" and "Limits.UpperAtSetPoint". Based on this status display, it can be determined whether the desired temperature has already been reached.

Program:

```
strLimits.UpperAlarm:=250;
strLimits.LowerAlarm:=70;
strLimits.UpperAtSetpoint:=20;
strLimits.LowerAtSetPoint:=20;

fb_TempControl(    Enable:=TRUE,
                  Setpoint:=150,
                  Feedback:=rActualTemp,
                  Limits:=strLimits,
                  ContrParam:=strContrParam,
                  ControlValueAnalog=>rActualPower,
                  SetPointReached=>bSetPointReached);
```

ML_TechBase.library

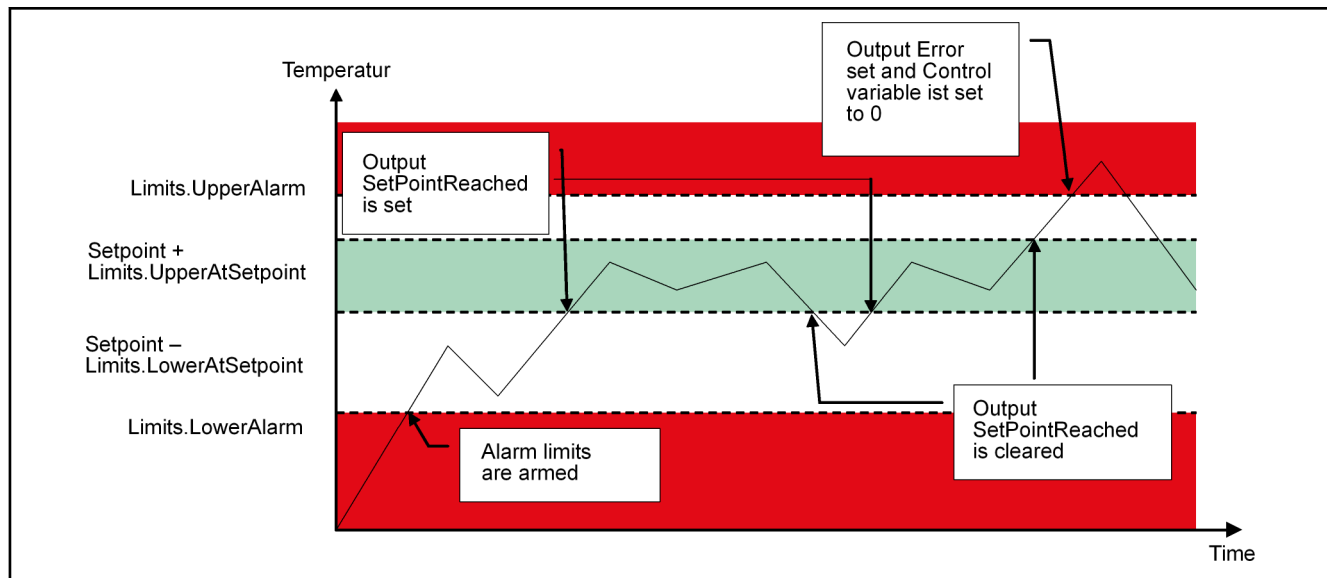


Fig. 3-38: Mode of operation of the alarm and status limits

Pausing the closed-loop control (PAUSE mode)

The closed-loop control can be paused via the "Pause" input. The control variable remains constant and the controller as well as all its values are "frozen". After the "Pause" input is reset, the closed-loop control continues.

In the example, the temperature is controlled to "150" in state "4". Switch to state 5 and the closed-loop control is "frozen" and no new control variables are output. Return to state "4" and the closed-loop control is continued and new control variables are output again. A jump can occur if the actual value has significantly changed in the meantime.

Program:

```

CASE iState OF
4:
fb_TempControl(   Enable:=TRUE,
                  Pause:=FALSE,
                  Setpoint:=150,
                  Feedback:=rActualTemp,
                  Preset:=TRUE,
                  ContrParam:=strContrParam,
                  Limits:=strLimits,
                  ControlValueAnalog=>rActualPower);

5:
fb_TempControl(   Enable:=TRUE,
                  Pause:=TRUE,
                  Setpoint:=150,
                  Feedback:=rActualTemp,
                  Preset:=TRUE,
                  ContrParam:=strContrParam,
                  Limits:=strLimits,
                  ControlValueAnalog=>rActualPower);

```

Analog/digital control variable output

The function block is provided with two control variable outputs: one analog output ("ControlValueAnalog") displaying REAL values between "0%" (no power) and "100%" (full power) and one digital output ("ControlValuePWM") displaying the BOOL values "FALSE" (no power) or "TRUE" (full power). The digital output can pulse the power when using relay modules. Pulsing takes place using a pulse-width modulation whose cycle time is 20 times slower than the time of the cyclic task in which the function block is running. Thus, control variables can be modulated in steps of 5 % with a resolution between 0 % to 100 %.

The analog output can be used for an analog output or to monitor the current control variable.

Program:

```
fb_TempControl( Enable:=TRUE,
                Pause:=FALSE,
                Setpoint:=150,
                Feedback:=rActualTemp,
                Preset:=TRUE,
                ContrParam:=strContrParam,
                Limits:=strLimits,
                ControlValueAnalog=>rActualPower,
                ControlValuePWM=>bActualPower);
```

Specifying a command value gradient

The function block can limit (ramp) the increase of the command value. This reduces the overshoot width and prevents the unit from damages. For each new specification of the input "SetPoint", the command value is internally changed in ramps (not jumps) until the desired command value is reached. "SetPointGradient" is entered in [rps].

In the example, the command temperature is controlled to "150" with an increase of "0.5 per second".

Program:

```
fb_TempControl( Enable:=TRUE,
                Pause:=FALSE,
                Setpoint:=150,
                Setpointgradient:=0.5,
                Feedback:=rActualTemp,
                Preset:=TRUE,
                ContrParam:=strContrParam,
                Limits:=strLimits,
                ControlValueAnalog=>rActualPower);
```

Ramping with command value gradient and an additional startup ramp shows the following development of the command and actual value. It has been controlled to a command temperature ("Setpoint:=200") of "200°C" with a gradient ("Setpointgradient:=1") of "1°C/s". The startup temperature ("StartUp-Temp:=100") is "100°C" and the startup time ("StartUpTime:=T#50s") is "50s".

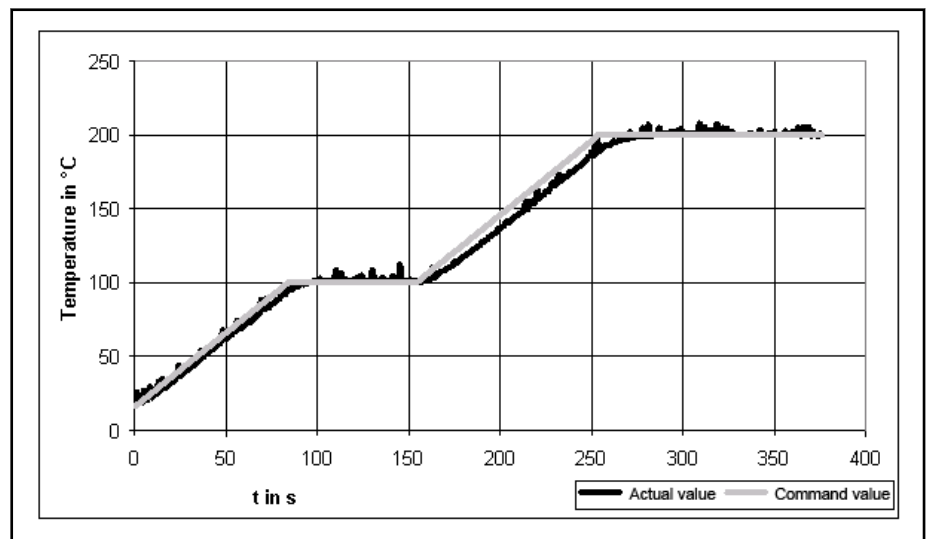


Fig. 3-39: Using a command value gradient and a startup ramp

Specifying a manual control variable

A manual control variable can be defined to control the temperature manually. At the same time, this variable represents the control variable at the function block output. A manual control variable can be required to record manual testing functions or to enable an emergency mode in case of a sensor error.

In the example, the function block is manually controlled with a control variable of "50%".

ML_TechBase.library

Program:

```
fb_TempControl(  Enable:=TRUE,
                 Pause:=FALSE,
                 OperatingMode:=TRUE,
                 ManContrValue:=50,
                 ContrParam:=strContrParam,
                 Limits:=strLimits,
                 ControlValueAnalog=>rActualPower);
```

Retain backup of the controller parameters

To ensure that the controller parameters are safely stored even in the case of a power failure, they can be backed up in the retain area. Thus, no identification is required when switching on the controller and the closed-loop control can be started immediately.

For this purpose, create the "strContrParam" structure as retain variables ("VAR RETAIN") containing the values of the individual controller parameters. At the same time, create a mechanism to differentiate at the initial startup whether an identification has already been performed.

Additional retain variables have to be created in the header.

Program:

```
VAR RETAIN
    strContrParam:  STRUCT_CONTR_PARAM;
    bIdentification: BOOL := FALSE;
END_VAR

If (bIdentification = FALSE) THEN
    fb_TempControl(  Enable:=TRUE,
                    StartAutoTune := TRUE,
                    Feedback:=rActualTemp,
                    Quickness:=2,
                    ContrParam:=strContrParam,
                    ControlValueAnalog=>rActualPower,
                    ControlValuePWM=>bActualPower,
                    AutoTuneDone=>bAutoTuneDone);

    IF (bAutoTuneDone = TRUE) THEN
        bIdentification:= TRUE;
    END_IF
ELSE
    fb_TempControl(  Enable:=TRUE,
                    StartAutoTune := FALSE
                    Preset:=TRUE,
                    Setpoint:=150,
                    Feedback:=rActualTemp,
                    ContrParam:=strContrParam,
                    Limits:=strLimits,
                    ControlValueAnalog=>rActualPower);

END_IF;
```

In the example above, the values of the controller are stored in the retain variables in each cycle. After a deactivation/power failure, the code checks if there has already been an identification using the Boolean variable "bIdentification". If this is not the case, an identification is performed. Otherwise, the controller can be restarted with the backed up (old) parameters. A rising edge has to be present at the input "Preset".

Troubleshooting with the substitute value strategy

By default, the function block displays the control variable "0%" (no power) in the event of an error. This is done for the system safety. If the user wants to use a control with a certain power, for example to avoid a strong cool down of a system, it is recommended to use the substitute value strategy described in the following example.

Errors can be checked in state "6" with "bError". If this condition is "TRUE", the function block is reset ("Enable:=FALSE") and state "7" is enabled.

In state "7", the function block is restarted with a rising edge at the input "Enable" and operated in manual mode (see [Specifying a manual control variable, page 79](#)).

In the example, a control variable of "50%" is output.

Further information on the error is output at the outputs "ErrorID" and "ErrorIDent" which were not analyzed in this example.

Program:

```
CASE iState OF:
6:
fb_TempControl(  Enable:=TRUE,
                  Setpoint:=50,
                  Feedback:=rActualTemp,
                  Preset:=TRUE,
                  ContrParam:=strContrParam,
                  Limits:=strLimits,
                  ControlValueAnalog=>rActualPower,
                  Error=>bError);

IF (bError = TRUE) THEN
  fb_TempControl( Enable:=FALSE,
                  ContrParam:=strContrParam,
                  Limits:=strLimits);

  iState := 7;
END_IF

7:
fb_TempControl(  Enable:=TRUE,
                  Pause:=FALSE,
                  OperatingMode:=TRUE,
                  ManContrValue:=50,
                  ContrParam:=strContrParam,
                  Limits:=strLimits,
                  ControlValueAnalog=>rActualPower);
```

It has to be stated that the control variable output in the example above is set to 0 for one cycle (if the error is detected and if Enable:=FALSE). In state "7", the manual value is used as control variable.

Troubleshooting

The function block generates the following error messages in Additional1/Additional2 for the table **F_RELATED_TABLE**, 16#0170:

ML_TechBase.library

ErrorID	Additional1	Additional2	Description
STATE_MACHINE_ERROR	16#1600	Current state	Error in the state machine; invalid state
INPUT_RANGE_ERROR	16#1601	16#0001	Input values for a maximum output value "OutputPowerMax", "AutoTuneStartPower" or "AutoTuneFinishPower") not in the valid range. Value has to be higher than or equal to 0 [%] up to max. 100 [%]
		16#1602	Input values for alarm limits invalid (Condition: Limits.UpperAlarm > Limits.LowerAlarm)
		16#1603	Input value for manual control variable invalid. Value has to be between 0[%] and 100[%]
		16#0004	Controller parameter invalid
		16#000A	Initial and final value for identification purposes are too close to each other. The minimum difference is 5[%]
		16#000B	The "SetpointGradient" input is invalid. Value has to be greater than 0
		16#000C	Input values invalid for status limits (Condition: (Setpoint + Limits.UpperAtSetPoint) > (Setpoint - Limits.LowerAtSetPoint))
		16#000D	Time for identification is too long ("AutoTuneTimeOut")
		16#000E	Value for "Quickness" is not in the valid limits (1 - 5)
		16#000F	Value for "ControlUpdateRate" is not in the valid limits (1 - 1000)
SYSTEM_ERROR	16#1602	Sampling time	Jittering the sampling time > 20[%]. Error becomes only active after two successive cycle timeouts
		16#0005	Actual value gradient is too high; possibly due to sensor error (diagnostics)
		16#0006	Error in the control loop, sensor possibly displaced or destroyed
CALCULATION_ERROR	16#1603	16#0007	Identified controlled system parameters are invalid, self-identification of the controlled system failed
		16#0008	The identified process model is not correct/invalid
ACCESS_ERROR	16#1604	Field number of the array	Incorrect access to internal variable

ErrorID	Additional1	Additional2	Description
OTHER_ERROR	16#1605	16#0009	Time overflow during identification (watchdog)
DEVICE_ERROR	16#1606	Temperature	Temperature error limit exceeded; temperature not in the error limits

Tab. 3-50: Error codes of the IL_TempControlType01 function block

Information on specific errors

- Error "Gradient of the actual value is too high"**
(Additional1 = 16#1602; Additional2 = 16#0005):

The gradient of the actual value is monitored. If the difference between two consecutive actual values is higher than 10 rms, this error is generated. This can be the case when using an Inline temperature model, since error analyses are displayed as measured values with values higher than 32768 (bit 15 set). If this diagnostics is not evaluated by the user, this error message can be output.
- Error "Error in the control loop, sensor possibly displaced or destroyed"**
(Additional1 = 16#1602; Additional2 = 16#0006):

It is monitored whether the actual value approaches the command value at a maximum output performance. For this purpose, time monitoring is started which is at an active I-value

$$2000/K_i + T_t \text{ and without the I-value is } 2000 * \text{ sampling time} + T_t.$$

If the actual value does not approach the command value during this period, the error is generated. The sensor might be defective/displaced or the heating/cooling system does not work. The error can also occur if no dead time (Tt) is specified when specifying the controller parameters via the structure "IL_TEMPCONTROL_CONTR_PARAM". In this case, the value of the dead time is to be estimated and the controller parameters are to be set again. This is performed automatically when identifying the controlled system.
- Error "Identified controlled system parameters are invalid, self-identification of controlled system failed"**
(Additional1 = 16#1603; Additional2 = 16#0007):

The controlled system parameters resulted in invalid values. This can result due to an incorrect sampling time or incorrect actual values (sensor not connected). Also refer to [Recommendation for the sampling time, page 74](#)
- Error "The identified process model is not correct/invalid"**
(Additional1 = 16#1603; Additional2 = 16#0008):

When identifying the controlled system, the calculated model is compared to the measured values. The comparison did not show any satisfying results. The model shows a deviation of more than 20 % from the measured values. The sampling time chosen might have been incorrect or the controlled system does not show an aperiodic behavior. This can also be caused by disturbance variables occurring during the identification.
- Error "Time elapsed during identification (watchdog)"**
(Additional1 = 16#1605; Additional2 = 16#0009):

When identifying the controlled system, the system waits until the actual value is in a steady state. Subsequently, a command value step change

is performed and the system waits again until the actual value is in the steady state. If this requires too much time, the error is generated. The reason is either a strongly varying actual value or an incorrectly selected sampling time (see also next section)

- Error "Time for identification is too long ('AutoTuneTimeOut')"
(Additional1 = 16#1601; Additional2 = 16#000D):
When identifying the controlled system, the time monitoring can be set. This value can have a maximum of 10,000 * sampling time. In this case, the time monitoring has to be set to 0 (maximum possible time for identification at a given sampling time) or the sampling time has to be increased accordingly

3.8.3 IL_TempControlType02

Brief description The IL_TempControlType02 function block controls temperatures such as the IL_TempControlType01 function block (IL_TempControlType01, page 66). It can determine the controller parameters either independently or with specifications.

- The advantages of the two variants are:
- Shorter identification period ("StartAutoTune")
 - Changed controller parameterization ("Preset")
 - Changed fine controller adjustment ("DampingFactor")
 - Input/output of the controlled system parameters ("ModelParam")
 - Individual controller design ("ControlMode")

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-51: Reference table of the IL_TempControlType02 function block

Interface description

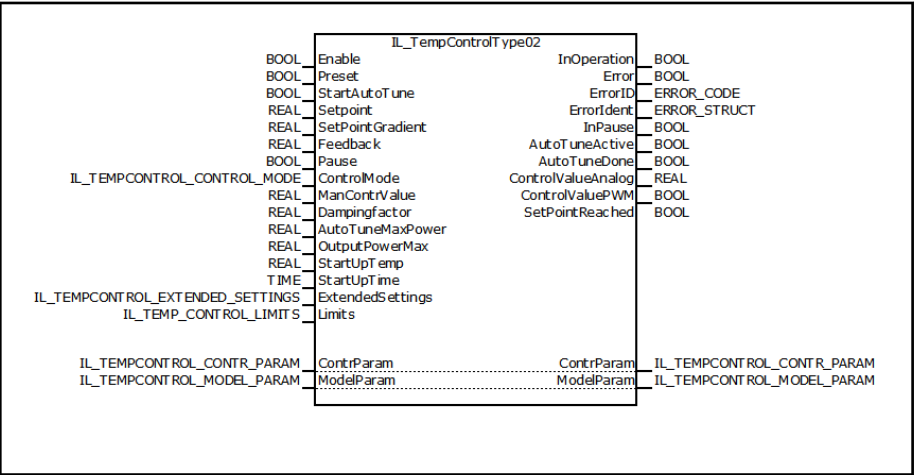


Fig. 3-40: IL_TempControlType02 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	The function block is active if this input is TRUE. The function block outputs are reset to their default value with a falling edge.
	Preset	BOOL	A rising edge at this input applies the controller or controlled system parameters of the structures "ContrParam" and/or "ModelParam"
	StartAutoTune	BOOL	A rising edge at this input starts the self-adjustment. If this input is set together with "Enable", the controller parameters do not have to be specified
	SetPoint	REAL	Command temperature. Unit [u]*
	SetPointGradient	REAL	Maximum change of the "Setpoint" parameter. Unit [u/s]
	Feedback	REAL	Actual temperature. Unit [u]
	Pause	BOOL	This input "freezes" the control variable and the controller calculation
	ControlMode	IL_TEMPCONTROL_CONTROL_MODE	Selection of the control loop response. Either the open-loop control in the open control loop or the closed-loop control optimize the command value response or the disturbance response in the closed control loop IL_TEMPCONTROL_CONTROL_MODE, page 94
	ManContrValue	REAL	This input specifies the control variable when controlling manually. Unit [%]
	DampingFactor	REAL	This input affects the control loop response. It can be steplessly selected between a slow aperiodic and a fast oscillating response. Unit [-]
	AutoTuneMaxPower	REAL	Control variable limitation at the beginning of the self-adjustment. Unit [%]
	OutputPowerMax	REAL	Limitation of the controller output. If a control signal limitation in the controlled system is smaller than the limitation of the controller output, the anti-windup function of the controller can be restricted. Unit [%]
	StartUpTemp	REAL	Startup temperature. Is only active if a value not equal to "0" is assigned to the "StartUpTime" input. Unit [u]
	StartUpTime	TIME	Startup time, i.e. the period during which the startup temperature ("StartUpTemp") is maintained until the "normal" closed-loop control starts. The preset value is "T#0s" (not active)

ML_TechBase.library

I/O type	Name	Data type	Description
	ExtendedSettings	IL_TEMPCONTROL_EXTENDED_SETTINGS	Structure containing the extended setting options of the self-adjustment. The default settings are sufficient for the common controlled systems IL_TEMPCONTROL_EXTENDED_SETTINGS, page 94
	Limits	IL_TEMPCONTROL_LIMITS	Structure containing the alarm limits. The preset value is always "0u" (not active). If values are assigned to the structure, Limits.UpperAlarm has to be > Limits.LowerAlarm and (Setpoint + Limits.UpperAtSetpoint) has to be >= (Setpoint - Limits.LowerAtSetpoint). Unit [u] IL_TEMPCONTROL_LIMITS, page 95
VAR_OUTPUT	InOperation	BOOL	This output is set if "Enable" is set and if the function block is active
	Error	BOOL	An error occurred at runtime. The control variable is set to "0"[%]
	ErrorID	ERROR_CODE	If the "Error output" is set, it contains a broad error classification
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, this output contains a detailed error information Troubleshooting, page 92
	InPause	BOOL	The "ControlValueAnalog" output is "frozen" and the "ControlValuePWM" output remains active with this value
	AutoTuneActive	BOOL	The self-adjustment is active
	AutoTuneDone	BOOL	The self-adjustment has been successfully completed
	ControlValueAnalog	REAL	Analog value of the control variable. Unit [%]
	ControlValuePWM	BOOL	Digital, pulse width-modulated control variable value. Unit [-]
	SetPointReached	BOOL	Status output whether the actual value is in the limits ("Setpoint " + "Limits.UpperAtSetpoint") >= Actual value >= (Setpoint - "Limits.LowerAtSetpoint")
VAR_IN_OUT	ContrParam	IL_TEMPCONTROL_CONTR_PARAM	This structure contains the parameters K_P , K_I and K_D of the controller as well as the controlled system delay time T_t IL_TEMPCONTROL_CONTR_PARAM, page 93
	ModelParam	IL_TEMPCONTROL_MODEL_PARAM	This structure contains the parameters K_S and T_S of the controlled system IL_TEMPCONTROL_MODEL_PARAM, page 95

*

The unit [u] means "unit". This can be any unit such as [°C] or [°F]. Ensure that the selected unit is always used.

Tab. 3-52: Interface variables of the IL_TempControlType02 function block

Min./max. and default values

The following table lists the min./max. and default values as well as the transfer behavior of the function block inputs:

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL	FALSE	TRUE	FALSE	Cyclic
Preset	BOOL	FALSE	TRUE	FALSE	Cyclic
StartAutoTune	BOOL	FALSE	TRUE	FALSE	Cyclic
SetPoint	REAL	N. def.	N. def.	0	Cyclic
SetPointGradient	REAL	0	N. def.	0	Cyclic
Feedback	REAL	N. def.	N. def.	0	Cyclic
Pause	BOOL	FALSE	TRUE	FALSE	Cyclic
ControlMode	IL_TEMPCONTROL_ CONTROL_MODE	CONTROL_TEMP- MODE_ MANUAL	CONTROL_TEMP- MODE_ OPT_DISTURB- ANCE	CONTROL_TEMP- MODE_ OPT_DISTURB- ANCE	Cyclic
ManContrValue	REAL	0	100	0	Cyclic
DampingFactor	REAL	20	200	100	Cyclic
AutoTuneMax- Power	REAL	>0	100	80	At a rising edge at "StartAutoTune"
OutputPowerMax	REAL	>0	100	100	Cyclic
StartUpTemp	REAL	N. def.	N. def.	0	At a rising edge at "Enable"
StartUpTime	TIME	T#0s	N. def.	T#0s	At a rising edge at "Enable"
ExtendedSettings	IL_TEMPCONTROL_ EXTENDED_SETTINGS	See "Data structure"	See "Data structure"	See "Data structure"	See "Data structure"
Limits	IL_TEMPCONTROL_ LIMITS	See "Data structure"	See "Data structure"	See "Data structure"	See "Data structure"
ContrParam	IL_TEMPCONTROL_ CONTR_PARAM	See "Data structure"	See "Data structure"	See "Data structure"	See "Data structure"
ModelParam	IL_TEMPCONTROL_ MODEL_PARAM	See "Data structure"	See "Data structure"	See "Data structure"	See "Data structure"

Tab. 3-53: Min./max. and default values as well as the transfer behavior of the inputs of the function block IL_TempControlType02

Functional description

The function block runs on the IndraLogic XLC and the IndraMotion MLC. The actual temperature to be controlled can be determined using Inline temperature modules (IL TEMP 2 RTD) or provided differently via a field bus connection for example).

Except for the mode of operation of the auto tuning ("StartAutoTune"), the controller parameterization ("Preset"), the controller fine adjustment ("DampingFactor"), the input/output structure "ModelParam" and the option of the

Controller/controlled system parameters

controller design ("ControlMode"), the "IL_TempControlType02" function block is identical to the "IL_TempControlType01" function block.

The internal PID-controller can either be parameterized as P-, PI-, PD- or PID-controller or the function block can automatically determine "StartAutoTune" controller parameters via an auto-tuning.

Compared to the IL_TempControlType01" function block, the identification of the self-adjusting temperature controller function block "IL_TempControlType02" was modified in such a way that control loop is now closed when starting the self-adjustment and that the controller parameters follow the currently identified controlled system parameters continuously. For this functionality, set the "StartAutoTune" input with the controller design procedure "ControlMode". The self-adjustment is not required for known controller or controlled system parameters. In this case, the values are specified via the inputs "ContrParam" and/or "ModelParam" and applied with a rising edge at the "Preset" input.

A PT_1 -filter continuously smoothes the actual temperature value ("Feedback") in the function block to reduce disturbances. The time constant of the filter is nearly 20 times the cycle time ($T_{Filt}=20 \cdot T_{Task}$). Due to this and further secondary delays caused by the process, the temperature controlled system can be assumed as non-oscillating PT_2 controlled systems with a dominant compensation constant T_s and a secondary delay T_t .

$$\frac{T_{[C]}(s)}{u_{[%]}(s)} \approx \frac{K_s}{(1+s \cdot T_s) \cdot (1+s \cdot T_t)}$$

The following jump response of a temperature-controlled system is an example assuming that the controlled system parameter shows $K_s=1^\circ\text{C}/\%$, $T_s=10\text{ s}$ and $T_t=1\text{ s}$.

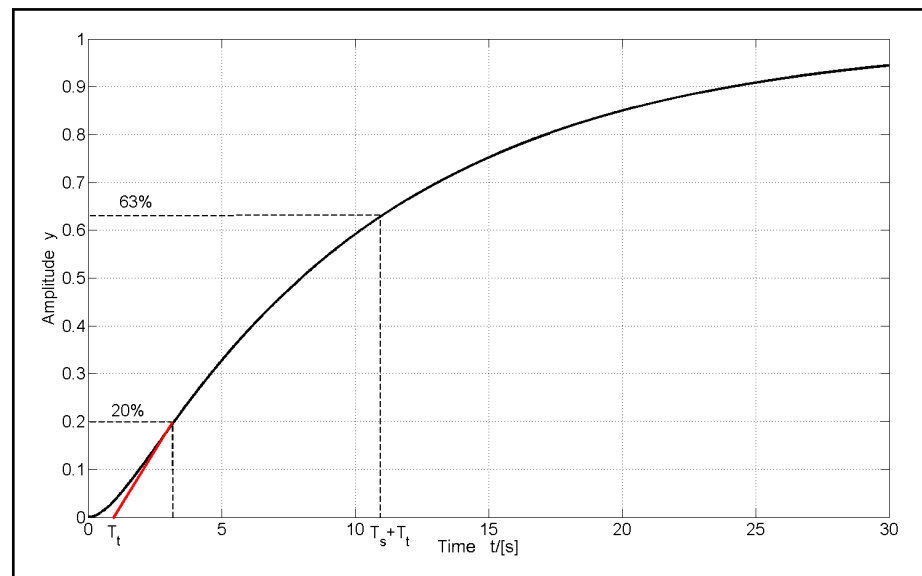


Fig. 3-41: Jump response of a non-oscillating PT_2 controlled system

In a self-adjustment, the control loop is directly closed with a P-controller with a rising edge at the "StartAutoTune" input to cause a controlled system reaction. The initial, proportional controller parameter K_P is used either from the "ContrParam" structure, calculated from the present controlled system parameters (K_s, T_s, T_t) or pre-assigned with a positive and internally stored value. While the actual value approximates the command value, the controlled sys-

tem response is continuously identified and the controller settings follow according to the controlled system information. After the identification, the "AutoTuning" is automatically completed and the determined controller and controlled system parameters are applied to the "ModelParam" (K_s, T_s) and "ContrParam" (K_p, K_i, K_d, T_t) structures.

Cooling

If there is a cooling, the K_p input has to be pre-assigned with a negative value for a first system initiation. Due to the control signal limitation and the controller adaptation, $K_p = -10000$ can be specified for the initial identification.

Control signal limitation

If the actual value is changed by 20 % of the command value change, there is already sufficient information on the dynamic controlled system response to calculate the controller parameters. Thus, at this point in time, the control signal limitation "AutoTuneMaxPower" of the automatically set PI-controller is increased to the maximum controller output "OutputPowerMax" to achieve the fastest correction time possible.

Preset; controller parameter

The internal PID-controller can also be used as P-controller, PI-controller, PD-controller or PID-controller if individual controller parts are set to zero. The controller parameterization as purely integrating I-controller is not supported with regard to the expected controlled system response.

$$K_p = \text{ContrParam}.K_p, \quad K_i = \text{ContrParam}.K_i, \quad K_d = \text{ContrParam}.K_d$$

Fig. 3-42: Formal dependency of the controller parameters at a "Preset"

Preset; controlled system parameter

In addition to the controller parameterization using the specified controller parameters, it can also be parameterized using specified controlled system parameters. If all controller parameters (K_p , K_i and K_d) of the "ContrParam" structure are zero when there is a rising edge at the "Preset" input, it is checked internally whether all controlled system parameters (K_s , T_s and T_t) are present in the "ModelParam" and "ContrParam" structures. After the plausibility check of these controlled system parameters, a PI-controller is designed on this basis. This controller parameterization has the advantage that the set controller can be switched between the disturbance and command value response and that the controller can be re-adjusted using the "DampingFactor".

$$K_p = f(\text{ModelParam}.K_s, \text{ModelParam}.T_s, \text{ModelParam}.T_t, \text{DampingFactor}, \text{ControlMode})$$

$$K_i = f(\text{ModelParam}.K_s, \text{ModelParam}.T_s, \text{ModelParam}.T_t, \text{DampingFactor}, \text{ControlMode})$$

Fig. 3-43: Formal dependency of the controller parameters at a "Preset" with controlled system parameters



To archive useful controlled system and/or controller parameters, it is recommended to back them up in the retain area. The structures "ContrParam" and/or "ModelParam", which contain the controller and/or controlled system parameters, can be created as retain variables ("VAR RETAIN").

AutoTuneMaxPower

Due to the dominant time constant T_s and a secondary delay time T_t , a jump response without control signal limitation results up to the point in time $t \sim T_s/3$. An IT_1 system can also write to the jump response (see [fig. 3-41 "Jump response of a non-oscillating \$PT_2\$ controlled system" on page 88](#)). To be able to determine controller parameters at this early stage of the self-adjustment (up to a change of 20% of the actual value), it might be required to limit the controller output signal additionally ("AutoTuneMaxPower"). Whether this is required depends on the relation of the controlled system constants. Thus, the transient of the delay time T_t can already be completed at this point in time

ML_TechBase.library

	(20% of the actual value change) even at a low quotient of the time constants $T_s/T_t < 10$. This results in sufficient information for a controller design.
DampingFactor	This input can be used to re-adjust the automatically ("StartAutoTune") or manually ("Preset" with specification of the controlled system parameters K_s, T_s, T_t) determined controller parameters. Changing and deviating parameters not covered in a fast and slow control-loop response for ideal controlled system parameters can thus also be considered. Therefore, the linear interpolation of the controller parameters between an oscillating and aperiodic transient performing an extrapolation was extended. For ideal controlled system parameters, a "DampingFactor" of 100 corresponds exactly to an aperiodic boundary case and a "DampingFactor" of 120 corresponds to an transient oscillation with approximately 20 % overshoot. Thus, the control loop response can decelerate accordingly with a "DampingFactor" of up to 20 or can be accelerated with a "DampingFactor" of up to 200 respectively. If there are controlled system parameters or a system response deviating from the real behavior, the desired dynamic control loop response can be corrected via this input.
StartUpTemp/StartUpTime	If a process prior to the actual temperature control requires a pause at a certain temperature for a certain period (startup ramp), this function is available via the inputs "StartUpTemp" and "StartUpTime". First, the controller controls up to the startup temperature "StartUpTemp". If this temperature is reached, the startup time "StartUpTime" expires and the controller keeps the startup temperature constant. After the startup time "StartUpTime" has elapsed, the startup ramp is completed and the controller shows its normal behavior. The startup ramp is executed once with each rising edge at the "Enable" input.
Limits	Alarm and status limits can be created for process monitoring purposes. The "Limits" structure contains the values "UpperAlarm" and "LowerAlarm". When exceeding the alarm limits, an error is generated. The monitoring function becomes active when the current process temperature ranges between the values of "UpperAlarm" and "LowerAlarm" so that no error is generated during process start. These values are absolute values. The "Limits" structure also contains the values "UpperAtSetPoint" and "LowerAtSetPoint". The "SetPointReached" is set as soon as the condition ("Setpoint" - "Limits.LowerAtSetPoint") $\leq$ "Feedback" $\leq$ ("Setpoint" + "Limits.UpperAtSetPoint") is met. These values are relative values related to the "Setpoint". The "Setpoint" can thus be modified without changing "Limits.LowerAtSetPoint" and "Limits.UpperAtSetPoint". Based on this status display, it can be determined whether the desired temperature has already been reached.
Pause	Pausing the closed-loop control can be implemented via the "Pause" input. The control variable remains constant and the controller as well as all its values are "frozen". After the "Pause" input is reset, the closed-loop control continues. A jump can occur if the actual value has significantly changed in the meantime.
SetPointGradient	The function block can limit the increase of the command value. This reduces the overshoot width and prevents the unit from damages. For each new specification of the "SetPoint" input, the command value is internally changed in ramps (not jumps) until the desired command value is reached. "SetPoint-Gradient" is entered in [rps].
ControlMode	This input can be used to select between a control in the open control loop and a closed-loop control (optimized to the command value or disturbance response) in a closed control loop. The manual control variable "ManContrValue" can for example be used before a self-adjustment to control the control-

	led variable in the operating point or when an error occurs in the actual value detection.
Control loop monitoring	The function block is provided with a gradient monitoring of the actual value reporting in case of very high deviations. There can be deviations for example if the sensor displays an error code and if this error code is not caught/evaluated by the user. The gradient is fixed to 10 [rms]. A single deviation is tolerated before an error is generated.
Task interval/PWM	The pulse-width modulation (PWM) is based on the task interval, i. e. the task interval is the smallest possible time slice which can be set to ON or OFF during the PWM. One PWM period is 20 times longer than the task interval. Thus, control variables can be modulated in steps of 5 % with a resolution between 0 % to 100 %. Set the task interval/cycle time with regard to the control element and the controlled system to ensure that the PWM does not cause any visible temperature changes in one PWM period of (20*task interval). Thus, if the temperature gradient changes noticeably between the switched-on and switched-off state of the PWM period due to the control variable, the selected task interval is too high. Do not use a task interval that is too small, since the control element (such as a relay module) is then subject to an increased load. Furthermore, the sampling time is monitored function block-internally. This sampling time generates an error if it exceeds/falls below by more than 20 % and sets the "Error" output to TRUE.
Self-adjustment	Note the following for an optimum self-adjustment: 1. Temperature-controlled systems are mostly not linear over a wide range. Thus, if there is an identical temperature change, the following applies: The higher the temperature, the higher the control signal change required and vice versa. That results in temperature-dependent settings that are required by the controller. Thus, the identification in the temperature range is performed at a later stage in the controller. 2. At runtime, the self-adjustment can be started again at a rising edge at the "StartAutoTune" input. This can for example be reasonable for a level change or if there are any other modifications at the controlled system. Ensure a sufficient system initiation (actual value change) for an optimum identification. 3. The self-adjustment can only be made if the command value is higher than the actual temperature for the heating and if the actual temperature is higher than the command value for a cooling.
Special influencing variables of the self-adjustment	In special cases, the following measures can be taken to improve the "Auto-Tuning": 1. The "ExtendedSettings.IdentAccuracy" can influence the abort criteria. It is aborted either when reaching this change rate or at double the correction time with this limit depending on which criterion is met first. 2. Use the "ExtendedSettings.OblivionFactor" input to consider changeable controlled system parameters at the "AutoTuning". For controlled system parameters varying in time, a reduction of this input is reasonable. For controlled system parameters not varying in time, an increase of this input is reasonable. 3. The "ExtendedSettings.MinDelayTime" input specifies a minimum delay time T_t as a priori information. This value is used for the "AutoTuning" if no value is specified at the "ContrParam.Tt" input. This value corresponds to the minimum controlled system delay time possible which

ML_TechBase.library

cannot be lower than the sum of all secondary transport, delay and filter time constants.

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#00001600	Current state	Error in the state machine; invalid state
INPUT_RANGE_ERROR	16#00001601	16#00001601	Input values for maximum output value ("OutputPowerMax") is not in the valid range. Value has to be higher than 0 [%] up to max. 100 [%]
INPUT_RANGE_ERROR	16#00001601	16#00001602	Input values for alarm limits invalid. (Condition: Limits.UpperAlarm > Limits.LowerAlarm)
INPUT_RANGE_ERROR	16#00001601	16#00001603	Input value for manual control variable ("ManContrValue") invalid. Value has to be between 0[%] and 100[%]
INPUT_RANGE_ERROR	16#00001601	16#00000004	Controller parameter invalid. There are either no controller parameters or the controller parameters are invalid
INPUT_RANGE_ERROR	16#00001601	16#00000005	Controlled system constant is invalid. Either the time constant "ModelParam.Ts" or "ContrParam.Tt" is invalid
INPUT_RANGE_ERROR	16#00001601	16#00000006	"OblivionFactor" input value is invalid. Value has to be between 0.9 and 1
INPUT_RANGE_ERROR	16#00001601	16#00000007	"MinDelayTime" input value is invalid. The value has to be higher than T#0s
INPUT_RANGE_ERROR	16#00001601	16#00000008	"IdentAccuracy" input value is invalid. Value has to be higher than 0 [%/s]
INPUT_RANGE_ERROR	16#00001601	16#00000009	"DampingFactor" input value is invalid. Value has to be between 20 and 200
INPUT_RANGE_ERROR	16#00001601	16#0000000B	"SetPointGradient" input value is invalid. Value has to be higher than 0 [u/s]
INPUT_RANGE_ERROR	16#00001601	16#0000000C	Input values invalid for status limits (condition: (Setpoint + Limits.UpperAtSetpoint) > (Setpoint - Limits.LowerAtSetpoint))
INPUT_RANGE_ERROR	16#00001601	16#00000010	"ContrMode" input value is invalid
INPUT_RANGE_ERROR	16#00001601	16#00000011	"AutoTuneMaxPower" input value is invalid. Value has to be between 0[%] and 100[%]
SYSTEM_ERROR	16#00001602	Sampling time	Jittering the sampling time > 20[%]. Error becomes only active after two successive timeouts
SYSTEM_ERROR	16#00001602	16#00000005	Gradient of the actual value is too high; maybe due to a sensor error
CALCULATION_ERROR	16#00001603	16#00000007	Identified controlled system parameters are invalid, self-identification of the controlled system failed
DEVICE_ERROR	16#00001606	Temperature	Temperature error limit exceeded; temperature not in the error limits

Tab. 3-54: Error codes of the IL_TempControlType02 function block

Information on specific errors

- "Controller parameter invalid"

(Additional1 = 16#1601; Additional2 = 16#0004):

Cause: Either there are no controller parameters or the controller parameters created in the input/output structure "ContrParam" are incorrect. In order that the temperature controller can create a reasonable control signal in either the operation mode "CONTROL_TEMP_MODE_OPT_COMMAND" or "CONTROL_TEMP_MODE_OPT_DISTURBANCE", at least one proportional controller with a K_p unequal to zero is required. Its number depends on the control direction (positive = heating and negative = cooling).

Error recovery: The temperature controller can either determine the controlled system and controller parameters itself or the controller parameters can be specified or the controller can determine the controller parameters itself using the controlled system parameters. For the self-determination without specified controller parameters, it is required to start the self-adjustment with a rising edge at "StartAutoTune". To specify the controller parameters, it is required to use the desired controller parameters at the input/output structure "ContrParam" and to apply them with a rising edge at the "Preset" input. To specify the controlled system parameters, it is required to set the proportional controller parameter ("ContrParam.KP") to zero in the input/output structures "ModelParam" and "ContrParam", to use the controlled system parameters ("ModelParam.Ks", "ModelParam.Ts" and "ContrParam.Tt") and to start the controller calculation with a rising edge at the "Preset" input

- "Identified controlled system parameters are invalid, self-identification of the controlled system failed"

(Additional1 = 16#1603; Additional2 = 16#0007):

Cause: The identification of the controlled system response and the resulting controller parameterization failed. This can cause a disadvantageous specified controlled system delay time "ContrParam.Tt", a task interval that is too high, a controlled system that is not sufficiently transient at the identification point in time, a controlled system initiation that is too low or an incorrect actual value detection (sensor not connected).

Error recovery: If the self-adjustment failed due to an incorrect sampling time, a rough estimation of the controlled system delay time can be specified as information for the self-adjustment at the "ContrParam.Tt" input and/or the task interval can be reduced. If the self-adjustment of the temperature controller failed due to the controlled system not being sufficiently transient, the self-adjustment is to be repeated with a stationary transient controlled system. If the self-adjustment failed due to an insufficient controlled system initiation, the self-adjustment is to be repeated with a bigger command value change. If the temperature detection is defective, the error has to be recovered

3.8.4 IL_TEMPCONTROL_CONTR_PARAM

The structure IL_TEMPCONTROL_CONTR_PARAM contains the current parameters K_P , K_I and K_D of the controller as well as the dead time T_t

ML_TechBase.library

Name	Type	Min. value	Max. value	Default value	Effective	Description
KP	REAL	n.def	n.def	10000	At a rising edge at "Preset" or "StartAutoTune"	Proportional gain of the controller
KI	REAL	n.def	n.def	0	At a rising edge at "Preset"	Integration constant of the controller
KD	REAL	n.def	n.def	0	At a rising edge at "Preset"	Differentiation constant of the controller
Tt	TIME	T#0s	n.def	T#0s	At a rising edge at "Preset", "StartAutoTune" or a change in the controller setting ("DampingFactor", "ControlMode")	Delay of the controlled system

Tab. 3-55: Structural description of the IL_TEMPCONTROL_CONTR_PARAM

3.8.5 IL_TEMPCONTROL_CONTROL_MODE

The enumeration type IL_TEMPCONTROL_CONTROL_MODE selects the control loop response. Either the open-loop control in the open control loop or the closed-loop control optimize the command value response or the disturbance response in the closed control loop

Enumeration element	Number	Description
CONTROL_TEMP-MODE_MANUAL	1	The function block controls the actual controlled system value using the value specified at the "ManContrValue" input
CONTROL_TEMP-MODE_OPT_COMMAND	2	The function block optimizes the command value response of the "Feedback" value to the "Setpoint" value
CONTROL_TEMP-MODE_OPT_DISTURBANCE	3	The function block optimizes the disturbance response of the "Feedback" value to the "Setpoint" value (default)

Tab. 3-56: IL_TEMPCONTROL_CONTROL_MODE enumeration type

3.8.6 IL_TEMPCONTROL_EXTENDED_SETTINGS

The structure IL_TEMPCONTROL_EXTENDED_SETTINGS contains the extended setting options of the self-adjustment. The default settings are sufficient for the common controlled systems.

Name	Type	Min. value	Max. value	Default value	Effective	Description
IdentAccuracy	REAL	>0	n.def	3.0	Cyclic	Abort criterion of the identification
OblivionFactor	REAL	0.9	1.0	0.98	Cyclic	Oblivion factor for changeable controlled systems
MinDelayTime	TIME	>T#0s	n.def	T#1s	At a rising edge of "StartAutoTune"	Minimum delay of the controlled system

Tab. 3-57: Structural description of the IL_TEMPCONTROL_EXTENDED_SETTINGS

3.8.7 IL_TEMPCONTROL_LIMITS

The structure IL_TEMPCONTROL_LIMITS contains the alarm limits in [u] (units). Initial value is "0u" (not active).

If values are assigned to the structure,

Limits.UpperAlarm > Limits.LowerAlarm)

and

(Setpoint + Limits.UpperAtSetpoint) > (Setpoint - Limits.LowerAtSetpoint)

Otherwise an error is generated

Name	Type	Min. value	Max. value	Default value	Effective	Description
UpperAlarm	REAL	n.def	n.def	0	Cyclic	Upper actual value limit when an alarm is triggered (absolute)
LowerAlarm	REAL	n.def	n.def	0	Cyclic	Lower actual value limit when the alarm is triggered (absolute)
UpperAtSetpoint	REAL	n.def	n.def	0	Cyclic	Upper actual limit for SetpointReached (relative)
LowerAtSetpoint	REAL	n.def	n.def	0	Cyclic	Lower actual limit for SetpointReached (relative)

Tab. 3-58: Structural description of the IL_TEMPCONTROL_LIMITS

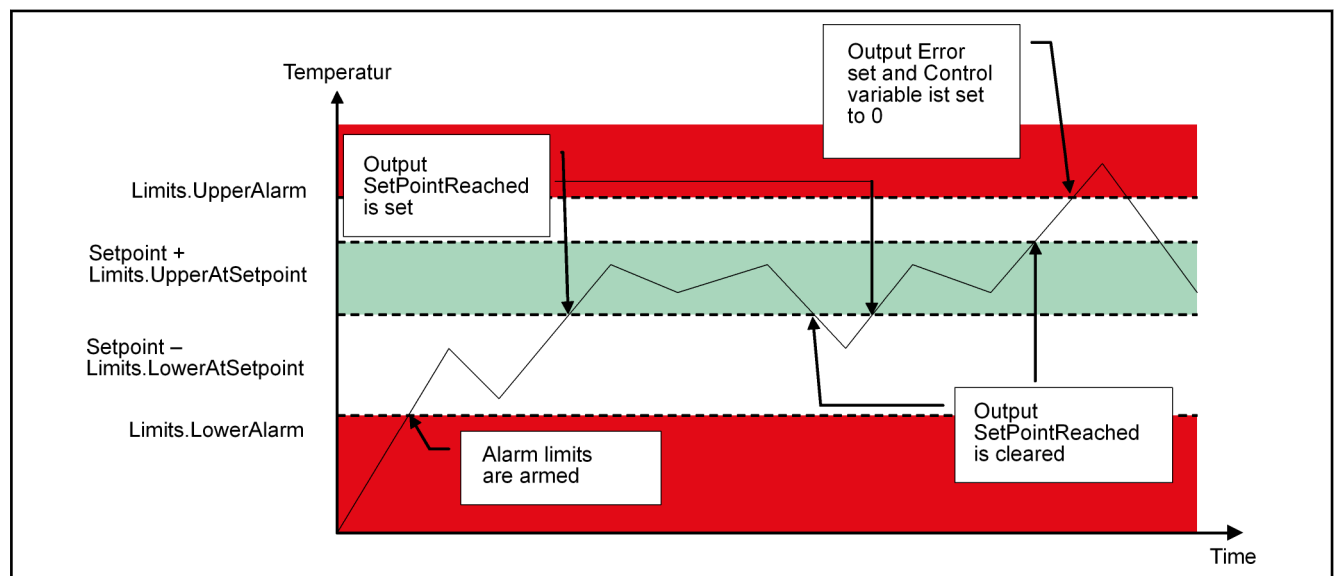


Fig. 3-44: Mode of operation of the alarm and status limits

3.8.8 IL_TEMPCONTROL_MODEL_PARAM

The structure IL_TEMPCONTROL_MODEL_PARAM contains the parameters K_S and T_S of the controlled system

ML_TechBase.library

Name	Type	Min. value	Max. value	Default value	Effective	Description
Ks	REAL	n.def	n.def	0	At a rising edge at "Preset", "StartAutoTune" or a change in the controller setting ("DampingFactor", "ControlMode")	Gain of the controlled system
Ts	TIME	T#0s	n.def	T#0s	Rising edge at "Preset", "StartAutoTune" or a change in the controller setting ("DampingFactor", "ControlMode")	Compensation time constant of the controlled system

Tab. 3-59: Structural description of the IL_TEMPCONTROL_MODEL_PARAM

3.9 Touch probe function blocks (TouchProbe)

3.9.1 Introduction and overview

The touch probe functionality is supported in this IndraWorks version as follows:

- The touch probe functionality of the **Rexroth IndraMotion control hardware** is provided via the following function blocks of the ML_PLCOpen library
- The touch probe functionality of the **Rexroth IndraDrive drives** is provided via the following function blocks of the ML_TechBase library:

The touch probe functionality of the Rexroth IndraMotion control hardware

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
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

The touch probe functionality of the Rexroth IndraDrive drives

Function block	Description
MB_InitTouchProbe, page 106	Enables and configures the touch probe in the drive
MB_WriteExpectWindow, page 120	Configures the expectation window for the touch probe functionality

Function block	Description
MB_TouchProbe, page 114	Enables the selected touch probe, analyzes the status and provides the measured values once after the trigger event
MB_TouchProbeContinuous, page 117	Enables the selected touch probe, analyzes the status and provides the measured values continuously after the trigger event
MB_AbortTrigger, page 123	Cancels an active measurement

The function blocks of both groups use data types declared in the RIL_CommonType library as well as their own data types.

Common data types in the RIL_CommonTypes

Name	Description
AXIS_REF	Contains information on the desired control and the axis controlled by it.
MB_DIFF_EVAL	Supports the setting of the methods for difference measuring.
MB_EXPECT_WINDOW_MODE	Supports the configuration and activation of the expectation window.
MB_INTERPOLATION_MODE	Supports the activation of the interpolation.
MB_POSITION_EVAL	Selection and activation of the trigger signal.
MB_PROBE_CONFIG	Used for registering the data of a touch probe function.
MB_PROBE_DATA	Supports the selection of the reference signal source of the touch probe for the touch probes of the XLC/MLC (A-parameters) and for the touch probes of the drive (S-/P-parameter).
MB_PROBE_MODULE	Supports the selection of the trigger source for the touch probe.
MB_PROBE_INPUT	Supports the selection of the desired input of the selected trigger source.
TOUCHPROBE_REF	Contains information on the desired control and the object (touch probe) controlled by it.

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 "Rexroth IndraMotion MLC 13VRS Functional Description", "Rexroth IndraLogic XLC 13VRS Functional Description" and "Rexroth IndraLogic XLC / IndraMotion MLC, PLCopen Function Blocks".

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

ML_TechBase.library

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:

- IndraMotion MLC/IndraLogic 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-...)
- 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 13VRS
- Drive firmware MPx 07VRS or higher

Parameterizing the Touch Probe Function Blocks for IndraMotion MLC/IndraLogic XLC and IndraMotion MLD

The following drive parameterization steps are required in order to use the MB_InitTouchProbe, MB_TouchProbe, and MB_TouchProbeContinuous technology function blocks:

IndraMotion MLC/IndraLogic XLC configuration

The following drive parameter settings are required when using the touch probe functionality in an IndraMotion MLC/IndraLogic 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

	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

Apply

Fig. 3-45: 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. 3-46: 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

ML_TechBase.library

- 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

Fig. 3-47: Configuring a cyclic AT telegram

Configuring the IndraMotion MLD-S or IndraMotion MLD-M (master axis)

The following drive parameter settings are required to use the touch probe functionality as master axis in an IndraMotion MLD-S or IndraMotion 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

Fig. 3-48: 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. 3-49: 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

ML_TechBase.library

- 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. 3-50: Command value data bits

- Set P-0-1367, bit 6 (activate update of the axis data structure, only IndraMotion MLD-M)

Configuring the IndraMotion MLD-M (slave axis)

The following drive parameter settings are required when using the touch probe functionality as slave axis in an IndraMotion 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. 3-51: 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_TechBase.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. 3-52: 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. 3-53: Configuring a cyclic AT telegram

- Set P-0-1367, bit 6 (activate update of the axis data structure, only IndraMotion 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. 3-60: 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. 3-61: 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

ML_TechBase.library

	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

Apply

Fig. 3-54: 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

Apply

Fig. 3-55: 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. 3-56: Example of a cyclic sercos channel

3.9.3 MB_InitTouchProbe

Brief description The MB_InitTouchProbe function block initializes the touch probe functionality.

Target system	Library
IndraMotion MLC 13VRS	ML_TechMotion
IndraMotion MLD with MPx07/08/17	MX_Technology07.lib

Interface description

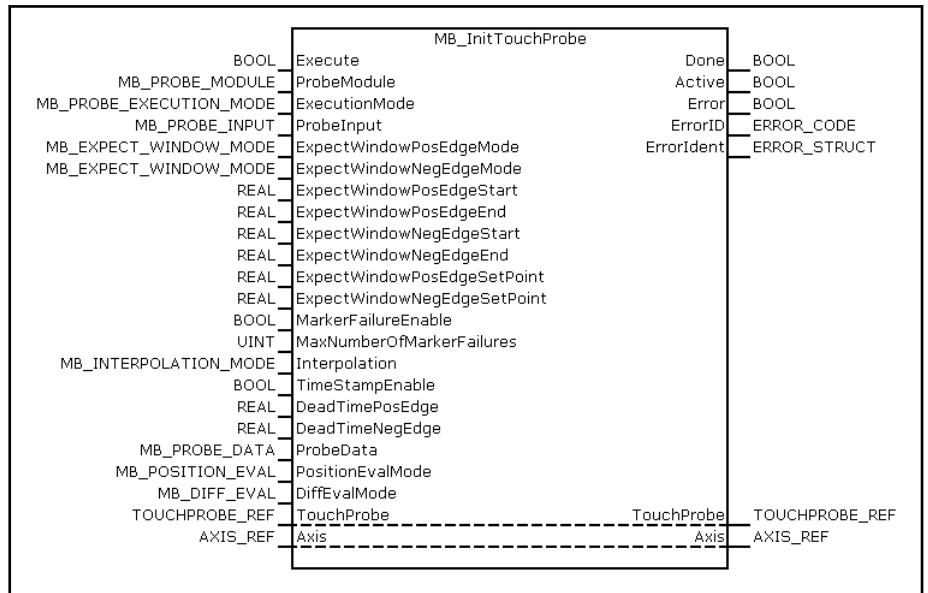


Fig. 3-57: *MB_InitTouchProbe* function block

I/O type	Name	Data type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
	TouchProbe	TOUCHPROBE_REF	Reference to the signal trigger source
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)
	ExecutionMode	MB_PROBE_EXECUTION_MODE	Selection whether single-shot or continuous measurement MB_PROBE_EXECUTION_MODE, page 125
	ProbeInput	MB_PROBE_INPUT	Selects the digital input for the MLC touch 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
	ExpectWindowNegEdgeStart	REAL	Start of the expectation window if the negative edge is evaluated

ML_TechBase.library

I/O type	Name	Data type	Comment
	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
	TimeStampEnable	BOOL	Enabling time stamp for MLC touch 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

Tab. 3-63: 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"
ProbeInput	MB_PROBE_INPUT			D_INPUT_1	Rising edge at "Execute"
ExpectWindowPosEdgeMode	MB_EXPECT_WINDOW_MODE			DISABLED_EXP	Rising edge at "Execute"

Name	Type	Min. value	Max. value	Default value	Effective
ExpectWindow-NegEdgeMode	MB_EXPECT_WINDOW_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"
MarkerFailureEnable	BOOL			FALSE	Rising edge at "Execute"
MaxNumberOfMarkerFailure	UINT			0	Rising edge at "Execute"
TimeStampEnable	BOOL			FALSE	Rising edge at "Execute"
DeadTimePosEdge	REAL	0.0		0.0	Rising edge at "Execute"
DeadTimeNegEdge	REAL	0.0		0.0	Rising edge at "Execute"
ProbeData	MB_PROBE_DATA			TP_ACT_POS_FEEDBACK	Rising edge at "Execute"
PositionEvalMode	MB_POSITION_EVAL			DISABLED_EVAL	Rising edge at "Execute"
DiffEvalMode	MB_DIFF_EVAL			DISABLED_DIFF	Rising edge at "Execute"

Tab. 3-64: Min./max. and default values for the MB_TouchProbeContinuous inputs

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 114](#), and [MB_TouchProbeContinuous](#), [page 117](#),.

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, DiffValMode

The "ProbeData" input defines the measured signal of the touch probe function. The following signals can be selected:

ENUM MB_PROBE_DATA (a choice of elements)

Signal	Description
TP_ACT_POS_FEEDBACK1	Actual position value position feedback 1
TP_ACT_POS_FEEDBACK2	Actual position value position feedback 2
TP_ACT_POSIN_ACT_CYCLE	Position actual value in actual value cycle
TP_SYNC_CMD_POS	ELS, synchronous position command value
TP_PROBE_TIME	Time measurement
TP_ACT_MEASURING_ENC_POS	Actual position value of measuring encoder
TP_RES_MASTER_AXIS_POS	ELS, resulting master axis position
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. 3-65: Possible values of the "ProbeData" input

The "DiffValMode" 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.

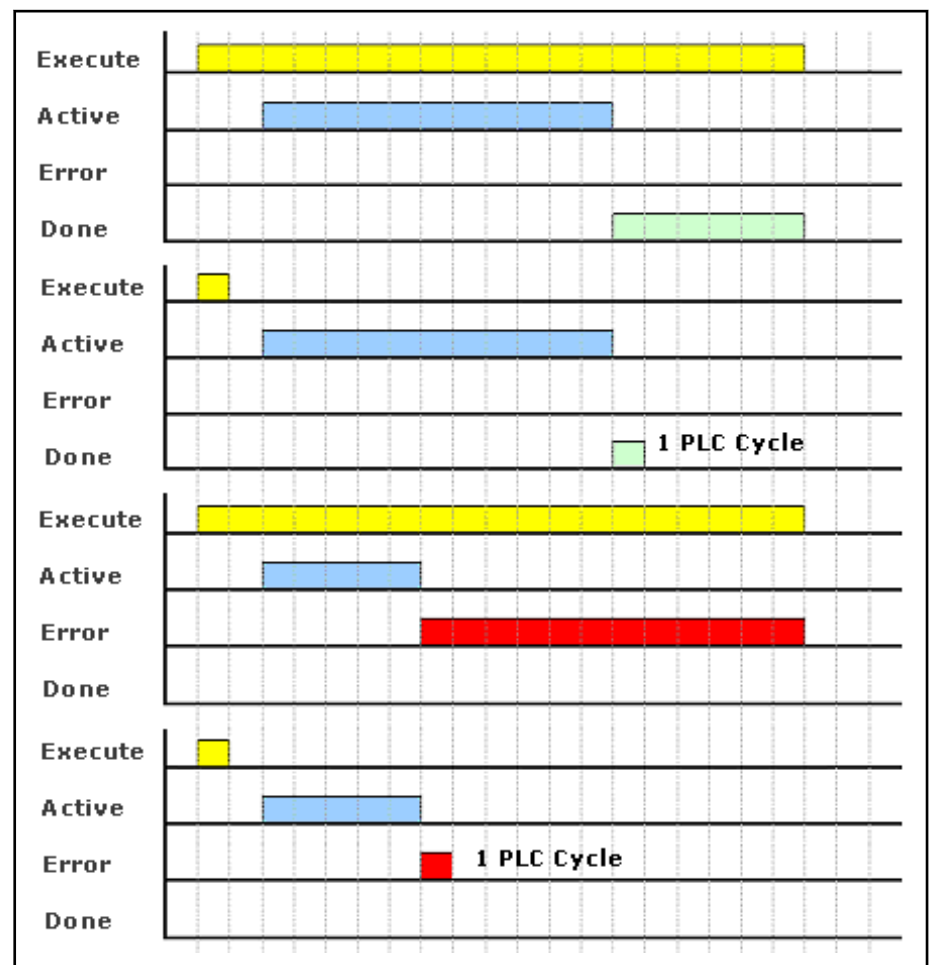


Fig. 3-58: Timing diagram of the MB_InitTouchProbe function block

Troubleshooting

The function block generates the following error messages in Additional1/ Additional2 for the "F_RELATED_TABLE" 16#0170:

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

ML_TechBase.library

ErrorID	Additional1	Additional2	Description
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 "DelExpectWindowPosEdge-Modeay" input is invalid
INPUT_INVALID_ERROR (16#0001)	16#0405	16#0002	The "ExpectWindowNegEdge-Mode" 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 "ExpectWindowPosEdgeSet-Point" input is not in the valid range
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0008	The "ExpectWindowNegEdgeSet-Point" 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 "AxisData" structure

Tab. 3-66: 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

ML_TechBase.library

- 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_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 MB_InitTouchProbe function block ([MB_InitTouchProbe](#), page 106).

Target system	Library
IndraMotion MLC 13VRS	ML_TechMotion
IndraMotion MLD with MPx07/08/17	MX_Technology07.lib

Tab. 3-67: MB_TouchProbe reference table

Interface description

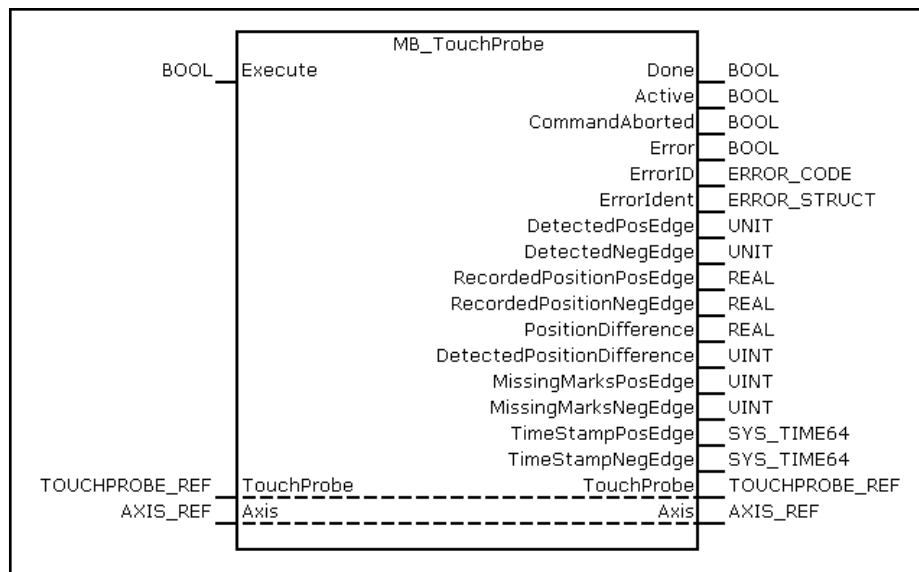


Fig. 3-59: MB_TouchProbe function block

I/O type	Name	Data type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
	TouchProbe	TOUCHPROBE_REF	Reference to the signal trigger source
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

I/O type	Name	Data type	Comment
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 116
	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)

Tab. 3-68: 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. 3-69: 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. 3-60: 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

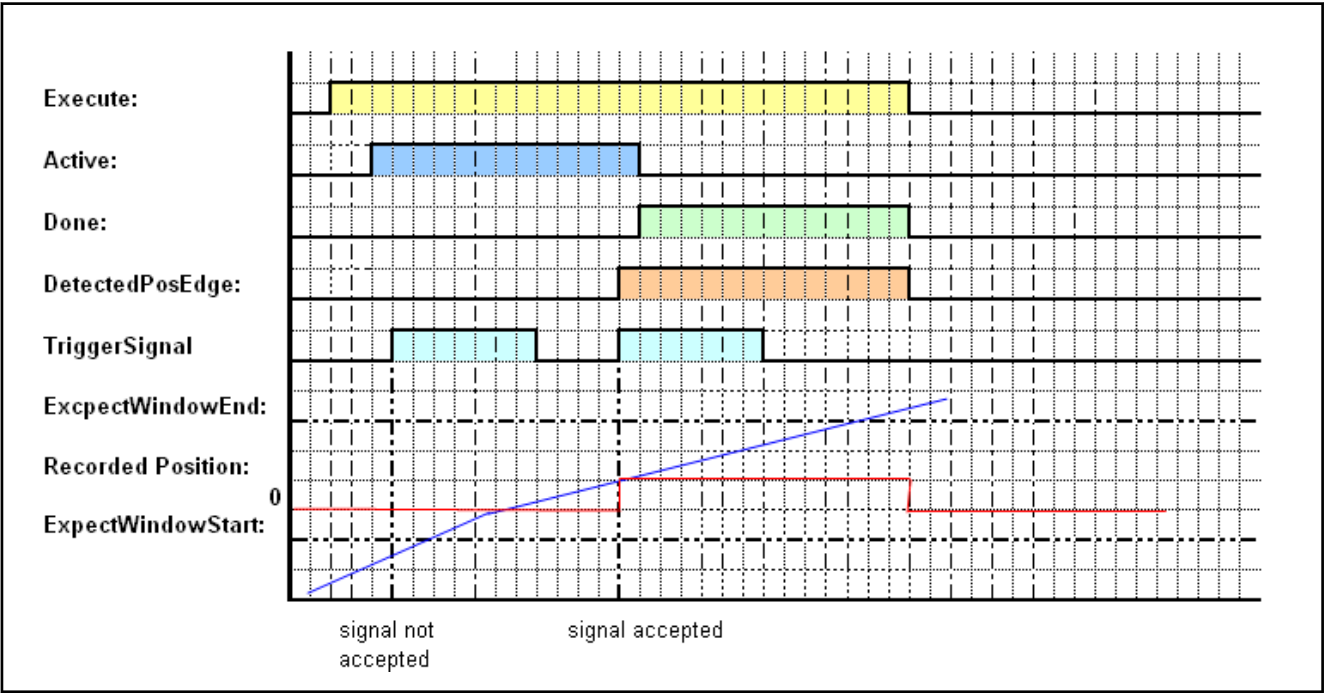


Fig. 3-61: Timing diagram of the MB_TouchProbe function block

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
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

ErrorID	Additional1	Additional2	Description
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 ATDeadTime-Comp
ACCESS_ERROR (16#0003)	16#0406	16#0006	Empty handle for SCWEnableP-robe
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. 3-70: Error codes of the MB_TouchProbe function block

3.9.5 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 MB_InitTouchProbe function block ([MB_InitTouchProbe, page 106](#)).

Target system	Library
IndraMotion MLC 13VRS	ML_TechMotion
IndraMotion MLD with MPx07/08/17	MX_Technology07.lib

Tab. 3-71: Reference table of the MB_TouchProbeContinuous

Interface description

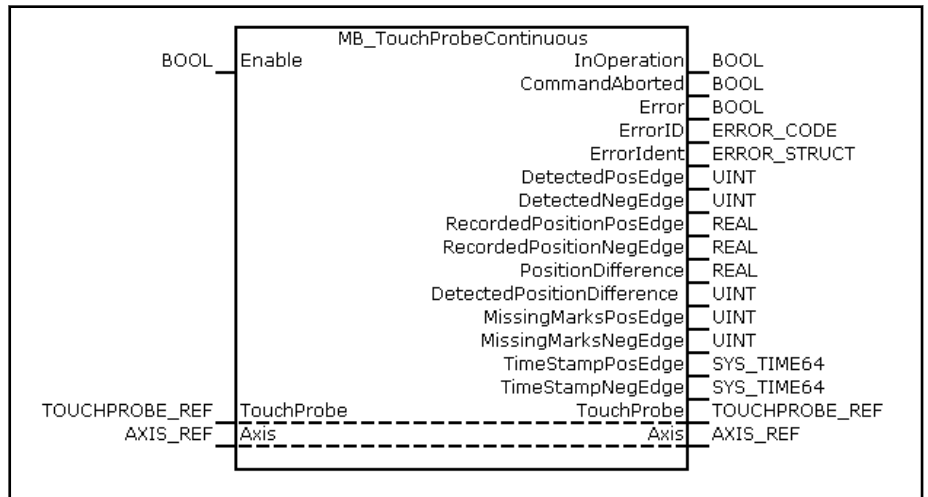


Fig. 3-62: MB_TouchProbeContinuous function block

I/O type	Name	Data type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
	TouchProbe	TOUCHPROBE_REF	Reference to the signal trigger source
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

ML_TechBase.library

I/O type	Name	Data type	Comment
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 119
	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)

Tab. 3-72: 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. 3-73: 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:

$$\text{PositionDifference} = |(S-0-0130) - (S-0-0131)|$$

$$\text{PositionDifference} = |(S-0-0132) - (S-0-0133)|$$

Fig. 3-63: 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

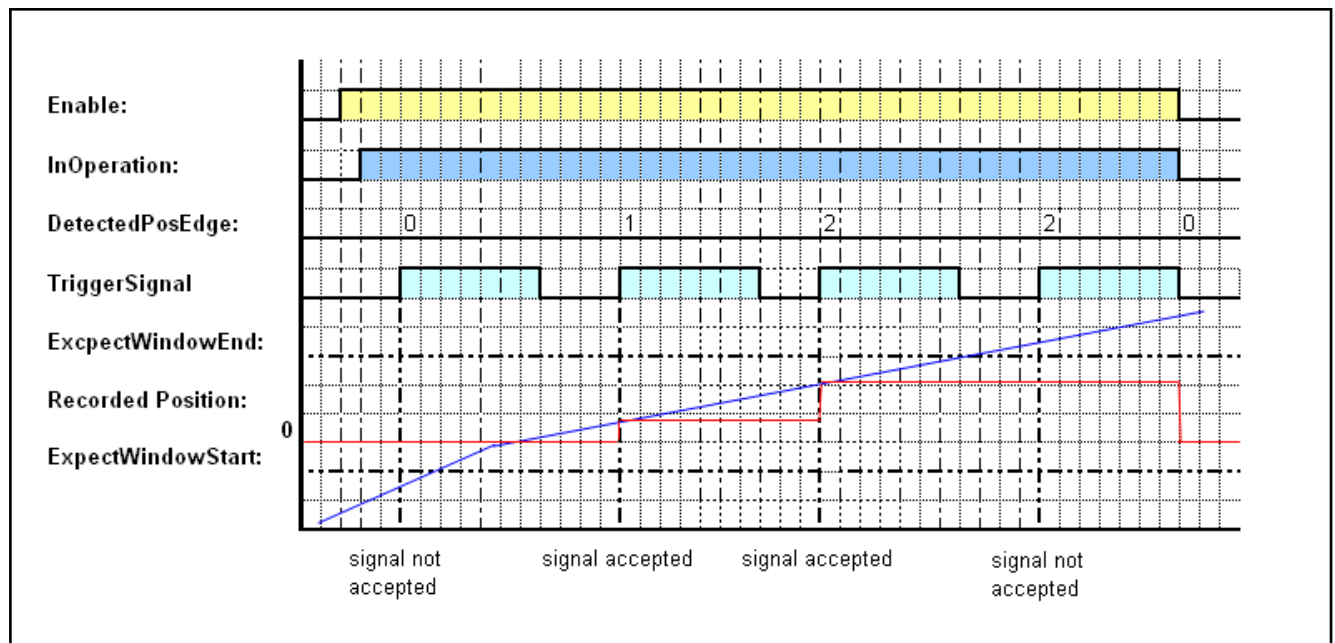


Fig. 3-64: Timing diagram of the MB_TouchProbeContinuous function block

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

ML_TechBase.library

ErrorID	Additional1	Additional2	Description
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 ATDeadTime-Comp
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. 3-74: Error codes of the MB_TouchProbeContinuous function block

3.9.6 MB_WriteExpectWindow

Brief description The MB_WriteExpectWindow function block configures the expectation window for the touch probe functionality.

Target system	Library
IndraMotion MLC 13VRS	ML_TechMotion
IndraMotion MLD with MPx07/08/17	MX_Technology07.lib

Tab. 3-75: Reference table of the MB_WriteExpectWindow

Interface description

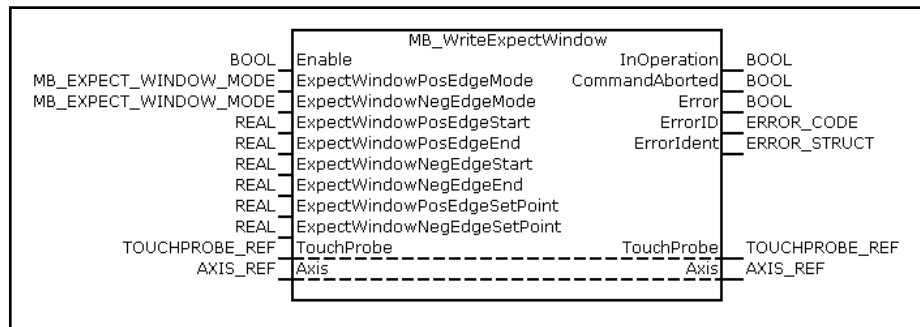


Fig. 3-65: MB_WriteExpectWindow function block

I/O type	Name	Data type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
	TouchProbe	TOUCHPROBE_REF	Reference to the signal trigger source
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

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
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 122

Tab. 3-76: 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. 3-77: 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 "ExpectWindowPosEdge-

ML_TechBase.library

Mode", "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

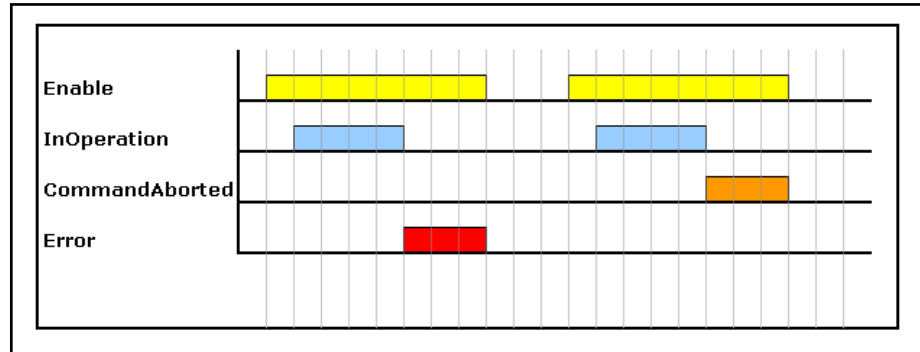


Fig. 3-66: Timing diagram of the MB_WriteExpectWindow function block

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

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR (16#0006)	16#0405	16#0008	The "ExpectWindowNegEdgeSet-Point" 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 "AxisData" structure.

Tab. 3-78: Error codes of the MB_WriteExpectWindow function block

3.9.7 MB_AbortTrigger

Brief description The MB_AbortTrigger function block aborts the MB_TouchProbe function block ([MB_TouchProbe](#), page 114).

Target system	Library
IndraMotion MLC 13VRS	ML_TechMotion
IndraMotion MLD with MPx07/08/17	MX_Technology07.lib

Tab. 3-79: Reference table of the MB_AbortTrigger

Interface description

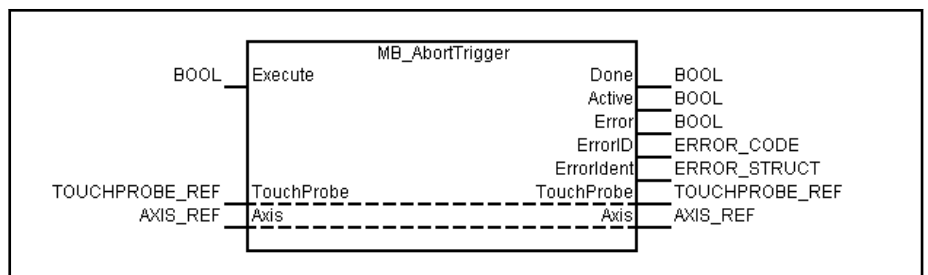


Fig. 3-67: MB_AbortTrigger function block

I/O types	Name	Data type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
	TouchProbe	TOUCHPROBE_REF	Reference to the signal trigger source
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 124

Tab. 3-80: 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. 3-81: 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

ML_TechBase.library

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.

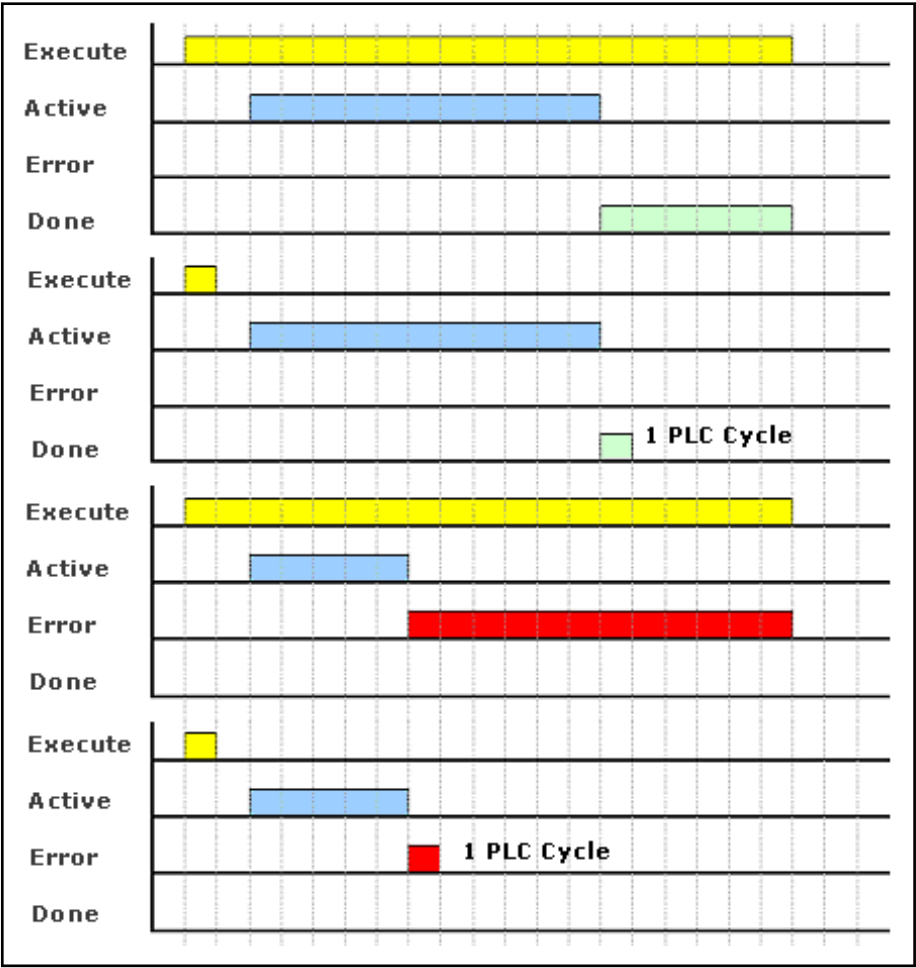


Fig. 3-68: Timing diagram of the MB_AbortTrigger function block

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. 3-82: Error codes of the MB_AbortTrigger function block

3.9.8 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. 3-83: ENUM MB_PROBE_EXECUTION_MODE

3.10 Trigonometric functions (Trigonometric Functions)

3.10.1 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.

3.10.2 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
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-84: Reference table of the ML_DegToInc function

Interface description

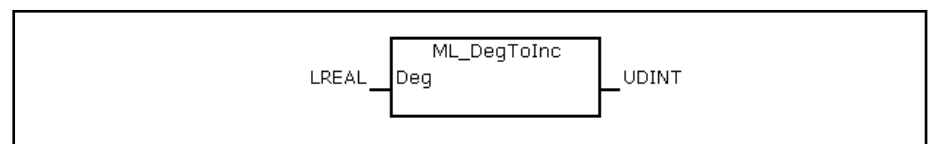


Fig. 3-69: ML_DegToInc function

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

Tab. 3-85: 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. 3-86: 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.

ML_TechBase.library

3.10.3 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
	IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-87: Reference table of the ML_DegToRad function

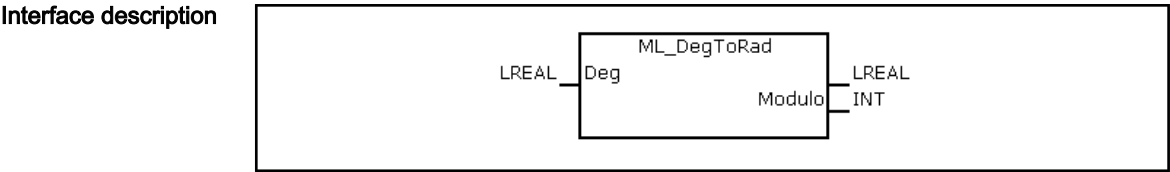


Fig. 3-70: ML_DegToRad function

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

Tab. 3-88: 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. 3-89: 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.

3.10.4 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
	IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-90: Reference table of the ML_IncToDeg function

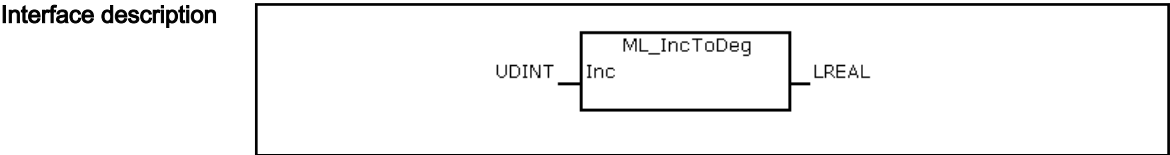


Fig. 3-71: ML_IncToDeg function

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

Tab. 3-91: 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.

3.10.5 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 ($0 \leq x < 2\pi$).

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-92: Reference table of the ML_IncToRad function

Interface description

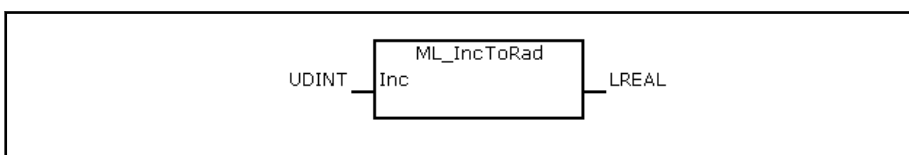


Fig. 3-72: ML_IncToRad function

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

Tab. 3-93: 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 ($0 \leq x < 2\pi$). 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.

3.10.6 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
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-94: Reference table of the ML_RadToDec function

Interface description

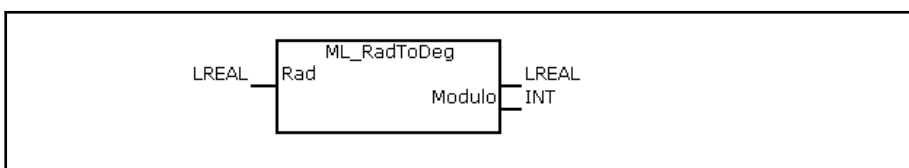


Fig. 3-73: ML_RadToDeg function

ML_TechBase.library

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

Tab. 3-95: 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. 3-96: 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 the return value and for "Modulo".

Error codes The function ML_RadToDeg does not generate any errors.

3.10.7 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
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-97: Reference table of the ML_RadToInc function

Interface description

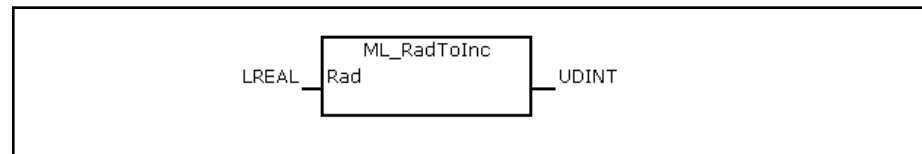


Fig. 3-74: ML_RadToInc function

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

Tab. 3-98: 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. 3-99: 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.

3.11 Cyclic data channels, administration (Utilities\CyclicData)

3.11.1 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 132. 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.

Configure also 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 if other parameters are requested later at runtime with `MB_AllocCyclicParameter`, page 130,, the new parameters are distributed to the free channels. The current configuration of the cyclic applied parameters can be read with `MB_GetCyclicChannelConfig`, page 134,.

In the operating mode (BB), the function block `MB_GetCyclicParameterHandle`, page 135, 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 137, and `MB_WriteCyclicParameter`, page 138,.

The functions `MB_ReadCyclicRealParameter`, page 137, and `MB_WriteCyclicRealParameter`, page 138, 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 139,.

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`.

ML_TechBase.library

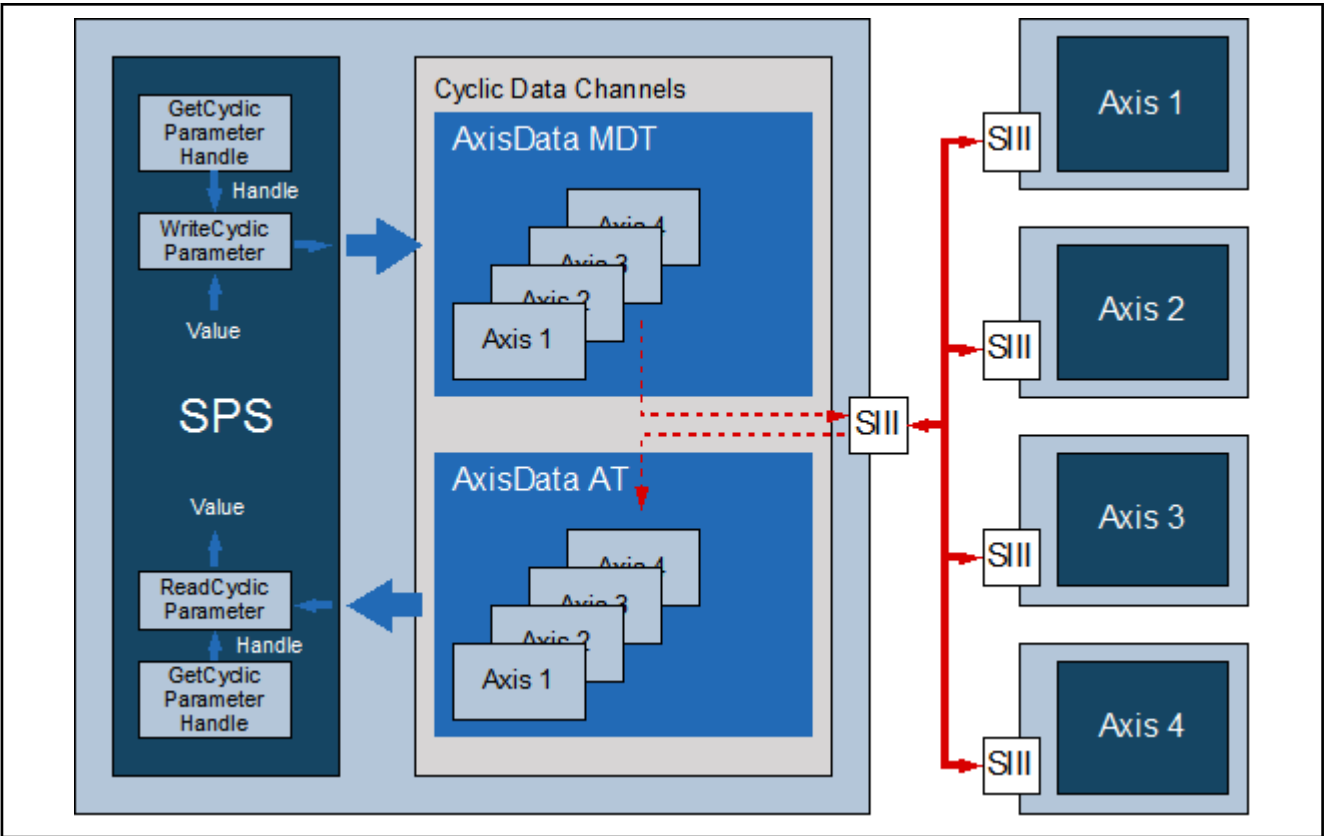


Fig. 3-75: Accessing the cyclic data using the "CyclicParameter" function blocks and functions

3.11.2 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.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx07 without MA function package	MX_Technology_07.lib

Tab. 3-100: Reference table of the MB_AllocCyclicParameter

Interface description

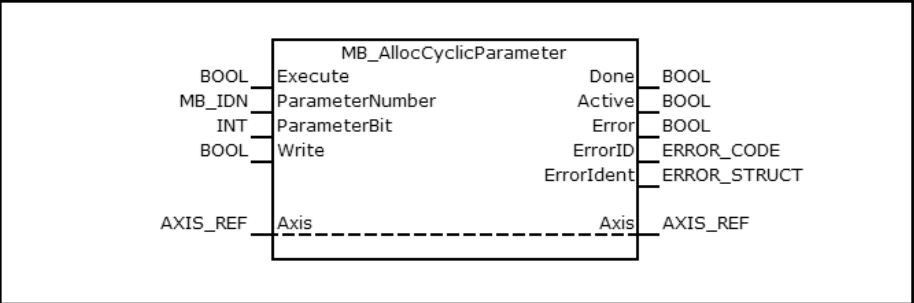


Fig. 3-76: Function block MB_AllocCyclicParameter

I/O type	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Parameter axis
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 131

Tab. 3-101: 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. 3-102: Input behavior of the MB_AllocCyclicParameter

Functional Description	This function block only operates in Sercos phase 2. At a rising edge at "Execute", the inputs "ParameterNumber", "ParameterBit" and "Write" 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 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:

ML_TechBase.library

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. 3-103: Error codes of the MB_AllocCyclicParameter function block

3.11.3 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.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx07 without MA function package	MX_Technology_07.lib

Tab. 3-104: Reference table of the MB_FreeCyclicParameter

Interface description

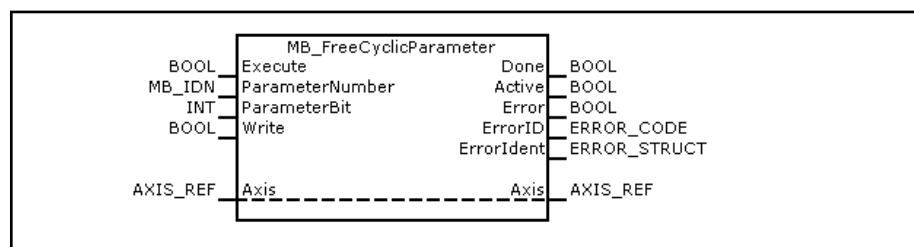


Fig. 3-77: MB_FreeCyclicParameter function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Axis	AXIS_REF	Parameter axis
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

I/O type	Name	Data type	Description
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 133

Tab. 3-105: Interface variables of the MB_FreeCyclicParameter 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. 3-106: 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 inputs "ParameterNumber", "ParameterBit" and "Write" 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. 3-107: Error codes of the MB_FreeCyclicParameter function block

ML_TechBase.library

3.11.4 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.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-108: Reference table of the MB_GetCyclicChannelConfig

Interface description

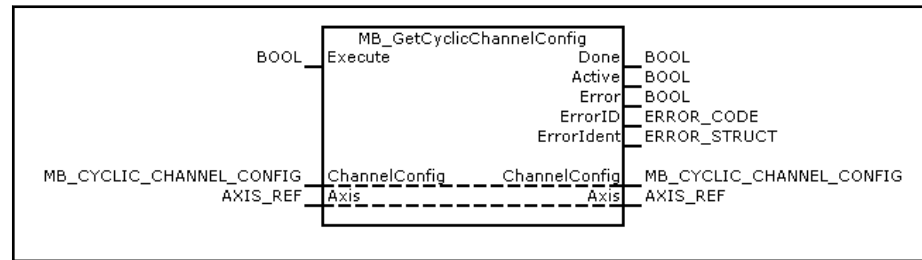


Fig. 3-78: MB_GetCyclicChannelConfig function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Axis	AXIS_REF	Parameter axis
	ChannelConfig	MB_CYCLIC_CHANNEL_CONFIG	Read-out configuration MB_CYCLIC_CHANNEL_CONFIG, page 141
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 135

Tab. 3-109: Interface variables of the MB_GetCyclicChannelConfig function block

Name	Type	Min. value	Max. value	Default value	Effective
Execute	BOOL			FALSE	Continuous

Tab. 3-110: Input behavior of the MB_GetCyclicChannelConfig

Functional Description

This function block only operates in the 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. 3-111: Error codes of the MB_GetCyclicChannelConfig function block

3.11.5 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.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx07 without MA function package	MX_Technology_07.lib

Tab. 3-112: Reference table of the MB_GetCyclicParameterHandle

Interface description

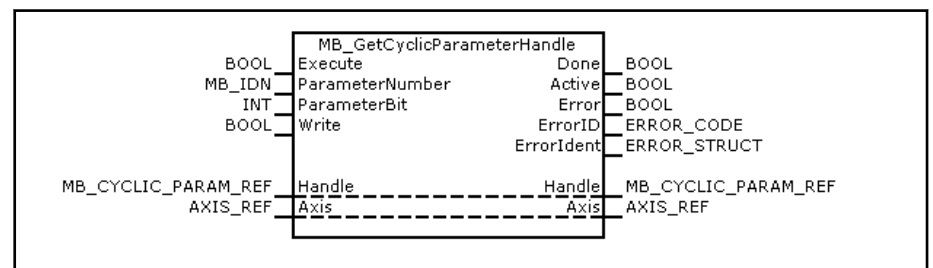


Fig. 3-79: MB_GetCyclicParameterHandle function block

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 140
	Axis	AXIS_REF	Parameter axis
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

ML_TechBase.library

I/O type	Name	Type	Comment
VAR_OUTPUT	Done	BOOL	Parameter was found and "Handle" is valid
	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 136

Tab. 3-113: Interface variables of the MB_GetCyclicParameterHandle 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. 3-114: 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 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

ErrorID	Additional1	Additional2	Description
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. 3-115: Error codes of the MB_GetCyclicParameterHandle function block

3.11.6 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.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx07 without MA function package	MX_Technology_07.lib

Tab. 3-116: Reference table of the MB_ReadCyclicParameter

Interface description

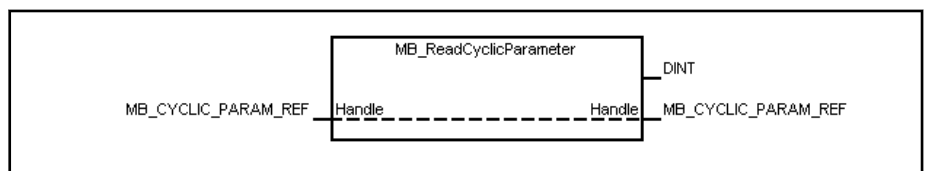


Fig. 3-80: MB_ReadCyclicParameter function

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 140
Return value		DINT	Parameter value

Tab. 3-117: Interface variables of the MB_ReadCyclicParameter function

3.11.7 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.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx07 without MA function package	MX_Technology_07.lib

Tab. 3-118: Reference table of the MB_ReadCyclicRealParameter

Interface description

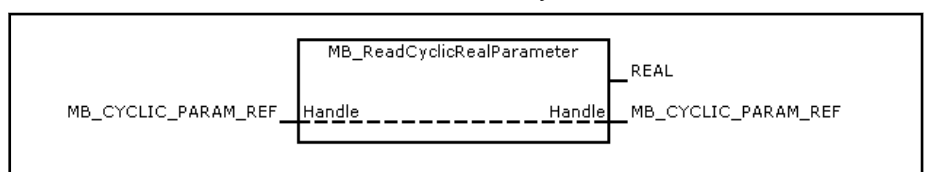


Fig. 3-81: MB_ReadCyclicRealParameter function

ML_TechBase.library

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 140
Return value		REAL	Parameter value

Tab. 3-119: Interface variables of the MB_ReadCyclicRealParameter function

3.11.8 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.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx07 without MA function package	MX_Technology_07.lib

Tab. 3-120: Reference table of the MB_WriteCyclicParameter

Interface description

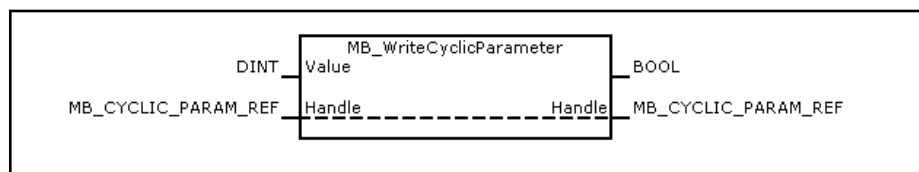


Fig. 3-82: MB_WriteCyclicParameter function

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 140
Return value		BOOL	Dummy output without meaning; is always TRUE

Tab. 3-121: Interface variables of the MB_WriteCyclicParameter function

3.11.9 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.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx07 without MA function package	MX_Technology_07.lib

Tab. 3-122: Reference table of the MB_WriteCyclicRealParameter

Interface description

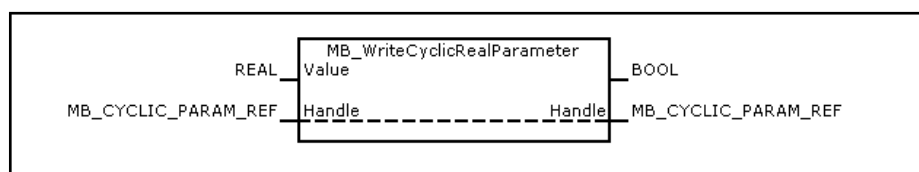


Fig. 3-83: MB_WriteCyclicRealParameter function

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 140
Return value		BOOL	Dummy output without meaning; is always TRUE

Tab. 3-123: Interface variables of the MB_WriteCyclicRealParameter function

3.11.10 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.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx07 without MA function package	MX_Technology_07.lib

Tab. 3-124: Reference table of the MB_GetLastCyclicParameterError

Interface description

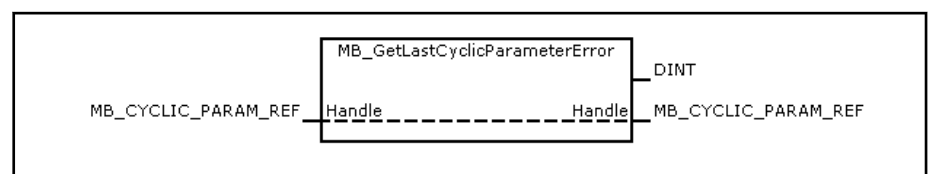


Fig. 3-84: MB_GetLastCyclicParameterError function

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 140
Return value		DINT	Error code of the last reading or writing action

Tab. 3-125: 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. 3-126: Error codes of the GetLastCyclicParameterError function

ML_TechBase.library

3.11.11 MB_CYCLIC_PARAM_REF

Type definitions The following definition describes data types used in the function block MB_GetCyclicParameterHandle ([MB_GetCyclicParameterHandle](#), [page 135](#))..

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.

Structure element	Type	Description
Axis	AXIS_REF	Parameter axis
ParameterNumber	MB_IDN	FP number of the parameter
bBit	INT	Bit number in the parameter
ParamType	MB_CYCLIC_TYPES	Parameter data type MB_CYCLIC_TYPES , page 140
AccessChannel	MB_CYCLIC_CHANNELS	Data channel in which the parameter was found MB_CYCLIC_CHANNELS , page 141
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	DINT	For the error code of the last error, see MB_GetLastCyclicParameterError , page 139

Tab. 3-127: Definition of the basic elements of the MB_CYCLIC_PARAM_REF structure

3.11.12 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

Element	Value
tDV_LINT	13
tDV_ULINT	14
tDV_REAL	15
tDV_LREAL	16

Tab. 3-128: ENUM MB_CYCLIC_TYPES

3.11.13 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. 3-129: ENUM MB_CYCLIC_CHANNELS

3.11.14 MB_CYCLIC_CHANNEL_CONFIG

The "MB_CYCLIC_CHANNEL_CONFIG" structure is described by the MB_GetCyclicChannelConfig function block and contains the applied parameters in the cyclic data channel of the axis ([MB_GetCyclicChannelConfig](#), page 134).

ML_TechBase.library

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
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. 3-130: Definition of the MB_CYCLIC_CHANNEL_CONFIG structure

3.12 Logic applications (Utilities\Logic)

3.12.1 Introduction and overview

The following function block was designed for applications at which a double-linked list is required. Elements can be added and deleted. The function block maintain the list. Thus, ring buffers, FIFOs and LIFOs can be designed.

3.12.2 IL_LinkedList

Brief description

The function block maintains a double-linked list. The list content can be specified by the user.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase

Tab. 3-131: Reference table of the IL_LinkedList

Interface description

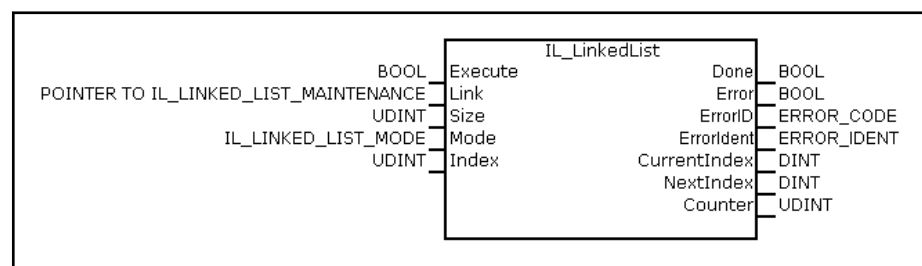


Fig. 3-85: IL_LinkedList function block

I/O type	Name	Data type	Description
VAR_INPUT	Execute	BOOL	The function block starts its functionality at a rising edge
	Link	POINTER TO IL_LINKED_LIST_MAINTENANCE	Pointer to an array of IL_LINKED_LIST_MAINTENANCE structures required to maintain the list IL_LINKED_LIST_MAINTENANCE on page 146
	Size	UDINT	Number of list elements to be maintained
	Mode	IL_LINKED_LIST_MODE	Mode that executes the function block, e.g. adding or removing an element IL_LINKED_LIST_MODE on page 146
	Index	DINT	Index on which the operation is to be applied
VAR_OUTPUT	Done	BOOL	If this output is TRUE, the function block has successfully completed its operation
	Error	BOOL	Indicates that an error occurred in the function block. The output data is invalid
	ErrorID	ERROR_CODE	Error diagnostics (see table Error codes on page 146)
	ErrorIdent	ERROR_STRUCT	Error structure with further subdivision of error classes
	CurrentIndex	DINT	Current index
	NextIndex	DINT	Next index
	Counter	UDINT	Number of elements in the list

Tab. 3-132: Interface variables of the IL_LinkedList 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
Link	0	0	16#FFFFFFFF	0
Size	UDINT	0	16#FFFFFFFF	0
Mode	IL_LINKED_LIST_MODE	ModeInitClear	ModeCheckIndex	ModeInitClear
Index	UDINT	0	16#FFFFFFFF	0

Tab. 3-133: Min./max. and default values of the IL_LinkedList function block

Functional Description

The function block maintains a double-linked list. This list is stored in an array in which only the indices have to be maintained. To maintain the list, create an array of type IL_LINKED_LIST_MAINTENANCE that has to be as large as its own data ([IL_LINKED_LIST_MAINTENANCE on page 146](#)).

The list is manipulated by calling the function block. The manipulation type can be defined by the "Mode" input ([IL_LINKED_LIST_MODE on page 146](#)).

Using the IL_LINKED_LIST_MAINTENANCE structure

The data structure IL_LINKED_LIST_MAINTENANCE is used by the function block and may not be overwritten. Since users can determine the length of the linked list, they have to create the maintenance structures for the list as well.

Therefore, define an array of the type [IL_LINKED_LIST_MAINTENANCE, page 146](#).

The structure can be used for debugging.

ML_TechBase.library

"Prev" is a pointer to the preceding list element

"Next" is a pointer to the following list element.

If the value is 0, it is the end or the beginning of the array or an unused element.

The inputs "Link" and "Size" of the function block may not be modified after initialized by "Mode ModeInitClear". This also applies when changing online e.g. if the length of the data structure is changed.

Example Example of the function block use:

Defining variables:

Program:

```
(* Maximum array size *)
uiMAX_ARRAY_SIZE: INT := 10;
(* My_data Structure array *)
arMyData: ARRAY [1..uiMAX_ARRAY_SIZE] OF MY_STRUCTURE;
(* List management array*)
arLinkedList: ARRAY [1..uiMAX_ARRAY_SIZE] OF IL_LINKED_LIST_MAINTENANCE;
(* Function block instance *)
fbMyList: IL_LinkedList;
(* Index variable*)
udiIndex: DINT;
```

Call for initialization:

Program:

```
fbMyList(Execute:= TRUE,
  Link:= ADR(arLinkedList),
  Size:= uiMAX_ARRAY_SIZE,
  Mode:= ModeInitClear,
  Index:= ,
  Done=> ,
  Error=> ,
  ErrorId =>,
  ErrorIdent=> ,
  CurrentIndex=> ,
  NextIndex=>);
```

Adding a head to the list:

Program:

```
(* Reset "Execute" input first (value of mode is not important) *)
fbMyList(Execute:= FALSE,
  Link:= ADR(arLinkedList),
  Size:= uiMAX_ARRAY_SIZE,
  Mode:= ModeInitClear,
  Index:= ,
  Done=> ,
  Error=> ,
  ErrorId =>,
  ErrorIdent=> ,
  CurrentIndex=> ,
  NextIndex=>);

(* Add head *)
fbMyList(Execute:= TRUE,
  Link:= ADR(arLinkedList),
  Size:= uiMAX_ARRAY_SIZE,
  Mode:= ModeAddHead,
  Index:= ,
  Done=> ,
  Error=> ,
  ErrorId =>,
  ErrorIdent=> ,
  CurrentIndex=> udiIndex,
  NextIndex=>);

(* Return value in CurrentIndex contains index for list head or
-1 if an error occurred. *)
```

```
IF udiIndex <> -1 THEN
    arMyData[udiIndex].StructureMember := MyValue;
END_IF;
```

Scroll through a list:

Program:

```
(* Reset "Execute" input first (value of mode is not important) *)
fbMyList(Execute:= FALSE,
    Link:= ADR(arLinkedList),
    Size:= uiMAX_ARRAY_SIZE,
    Mode:= ModeGetHead,
    Index:= ,
    Done=> ,
    Error=> ,
    ErrorId =>,
    ErrorIdent=> ,
    CurrentIndex=> ,
    NextIndex=>);

(* Determine head *)
fbMyList(Execute:= TRUE,
    Link:= ADR(arLinkedList),
    Size:= uiMAX_ARRAY_SIZE,
    Mode:= ModeGetHead,
    Index:= ,
    Done=> ,
    Error=> ,
    ErrorId =>,
    ErrorIdent=> ,
    CurrentIndex=> udiIndex,
    NextIndex=>);

(* Check return value (-1 means error) *)
WHILE (udiIndex > 0) DO
    (* Return value in CurrentIndex contains index in list *)
    arMyData[udiIndex].StructureMember := MyValue;

    (* Reset "Execute" input again *)
    fbMyList(Execute:= FALSE,
        Link:= ADR(arLinkedList),
        Size:= uiMAX_ARRAY_SIZE,
        Mode:= ModeGetNext,
        Index:= ,
        Done=> ,
        Error=> ,
        ErrorId =>,
        ErrorIdent=> ,
        CurrentIndex=> ,
        NextIndex=>);

    (* Get next element*)
    fbMyList(Execute:= TRUE,
        Link:= ADR(arLinkedList),
        Size:= uiMAX_ARRAY_SIZE,
        Mode:= ModeGetNext,
        Index:= ,
        Done=> ,
        Error=> ,
        ErrorId =>,
        ErrorIdent=> ,
        CurrentIndex=> udiIndex,
        NextIndex=>);
END_WHILE
```

Output assignment depending on the "Mode" input

Depending on the "Mode" input assignment, the "Index" input and the "CurrentIndex" and "NextIndex" are configured accordingly. The following table lists the respective values. An X stands for a value that is unchanged with respect to the previous call. If an error occurs, -1 is returned for "CurrentIndex".

Mode	Index	CurrentIndex	NextIndex
ModeInitClear	X	-1	Next free index
ModeAddTail	X	Tail index	Next free index
ModeAddHead	X	Head index	Next free index

ML_TechBase.library

Mode	Index	CurrentIndex	NextIndex
ModeRemoveTail	X	New tail index	Next free index
ModeRemoveHead	X	New head index	Next free index
ModeRemoveIndex	Index for removal	Previous element index	Next free index
ModeGetHead	X	Head index	X
ModeGetTail	X	Tail index	X
ModeGetNext	Previous index	Subsequent element index	X
ModeGetPrev	Next index	Previous element index	X
ModeGetEmptyIndex	X	X	Next free index
ModeCheckIndex	Testing index	1: Found -1: Not found	-1

Tab. 3-134: Function block outputs with respect to "Mode"

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#1900	16#0	The "Index" input is not in the valid range
INPUT_INVALID_ERROR	16#1901	16#0	"Link" input is a zero pointer
ACCESS_ERROR	16#1902	16#xxx	An index is still linked in the list although it is empty (internal error)
ACCESS_ERROR	16#1903	16#0	An element in the list is to be cleared, but the list is already empty
RESOURCE_ERROR	16#1904	16#0	List is full
STATE_MACHINE_ERROR	16#1905	16#0	"Mode" input has an invalid value
RESOURCE_ERROR	16#1906	16#0	"Size" input changed again after initialization (e.g. by online change).

Tab. 3-135: Error codes of the IL_LinkedList function block

3.12.3 IL_LINKED_LIST_MAINTENANCE

The structure is used by the IL_LinkedList function block ([IL_LinkedList](#), page 142).

"Prev" is a pointer for the previous list element, "Next" for the subsequent element. If the value is 0, it is the end or the beginning of the array or an unused element.

Due to this structure, debugging lists is facilitated.

Type definition

```

TYPE IL_LINKED_LIST_MAINTENANCE :
STRUCT
    Prev: POINTER TO DINT;
    Next: POINTER TO DINT;
END_STRUCT
END_TYPE

```

3.12.4 IL_LINKED_LIST_MODE

Use the enumeration type to define the action to be executed.

The enumeration is used by the function block [IL_LinkedList](#), page 142.

Enumeration	Significance
ModeInitClear	The linked list is initialized. All entries are deleted
ModeAddTail	An element is added at the tail of the list. If the list is empty, the first element is created
ModeAddHead	An element is added at the head of the list. If the list is empty, the first element is created
ModeRemoveTail	The last element in the list is removed
ModeRemoveHead	The first element in the list is removed
ModeRemoveIndex	The element in the index "Index" is removed
ModeGetHead	Returns the index of the head of the list
ModeGetTail	Returns the index of the tail of list
ModeGetNext	Returns the subsequent element in the list
ModeGetPrev	Returns the previous element in the list
ModeGetEmptyIndex	Returns the next free index in the list
ModeCheckIndex	Checks whether the specified index is used in the linked list. The result is returned in the "CurrentIndex": 1: Index is used -1: Index is not used

Tab. 3-136: Enumerations and their meaning in the function block [IL_LinkedList](#)

3.13 Modulo values (Utilities\Modulo)

3.13.1 Introduction and overview

The following functions calculate a modulo value. Depending on the selected function, different value ranges are used.

Areas of application and functions/function blocks of ML_TechBase

Area of application (folder in library)	
Function blocks	Description
Modulo value calculation (Utilities/Modulo)	
MB_GetModuloType01 , page 147	Calculates a modulo value. Value range between 0.0 and + modulo value (REAL)
MB_GetModuloType02 , page 148	Calculates a modulo value. Value range between 0 and + modulo value (DINT)
MB_GetModuloType03 , page 149	Calculates a modulo value. Value range between - modulo value and + modulo value (REAL)
MB_GetModuloType04 , page 150	Calculates a modulo value. Value range between - modulo/2 and + modulo value/2 (REAL)

Tab. 3-137: [RIL_Uilities](#), Modulo value calculation

3.13.2 MB_GetModuloType01

Brief description The function returns the modulo value. The returned value range is between - modulo value and + modulo value.

ML_TechBase.library

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx18	MX_Technology07.lib

Tab. 3-138: Reference table of the MB_GetModuloType01

Interface description

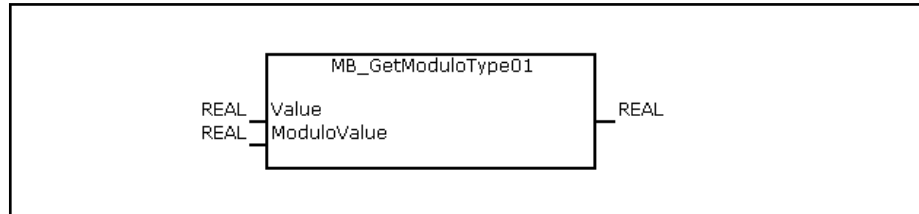


Fig. 3-86: MB_GetModuloType01 function

I/O type	Name	Data type	Description
VAR_INPUT	Value	REAL	Value at which the modulo function is to be applied
	ModuloValue	REAL	Modulo value to be used for the modulo operation
RETURN		REAL	Modulo value: The value range returned is between 0.0 and + modulo value.

Tab. 3-139: Interface variables of the MB_GetModuloType01 function

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
Value	REAL	- 1.175494e-38	3.402823e+38	0.0	With every call
ModuloValue	REAL	- 1.175494e-38	3.402823e+38	360.0	With every call

Tab. 3-140: Min./max. and default values of the MB_GetModuloType01 function

Functional Description

The function returns the modulo value. The value range returned is between 0.0 and + modulo value. The positive and the negative limit value are not part of the value range (also see the following diagram which shows the value transferred to the function on the x-coordinates and the value returned from the function on the y-coordinates).

If 0.0 is transferred for "ModuloValue", the function returns the value "Value".

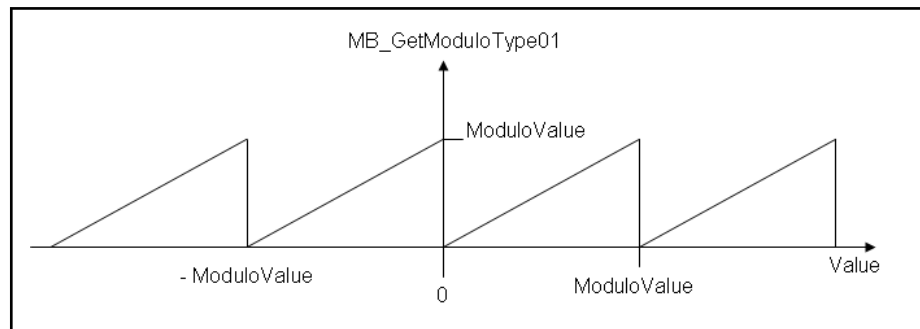


Fig. 3-87: Representation of the modulo function

Troubleshooting

The function does not generate any errors.

3.13.3 MB_GetModuloType02

Brief description

The function returns the modulo value. The returned value range is between - modulo value and + modulo value.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx07 without MA function package	MX_Technology_07.lib

Interface description

Tab. 3-141: Reference table of the MB_GetModuloType02

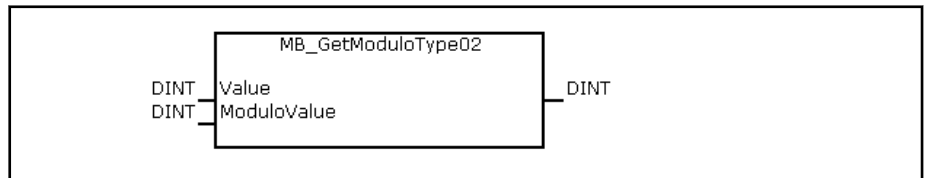


Fig. 3-88: MB_GetModuloType02 function

I/O type	Name	Data type	Description
VAR_INPUT	Value	DINT	Value at which the modulo function is to be applied
	ModuloValue	DINT	Modulo value to be used for the modulo operation
RETURN		DINT	Modulo value: The value range returned is between 0 and + modulo value

Tab. 3-142: Interface variables of the MB_GetModuloType02 function

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
Value	DINT	- 2147483648	2147483647	0	With every call
ModuloValue	DINT	- 2147483648	2147483647	360	With every call

Tab. 3-143: Min./max. and default values of the MB_GetModuloType02 function

Functional Description

The function returns the modulo value. The value range returned is between 0 and + modulo value. The positive and the negative limit value are not part of the value range (also see the following diagram which shows the value transferred to the function on the x-coordinates and the value returned from the function on the y-coordinates).

If 0 is transferred for "ModuloValue", the function returns the value "Value".

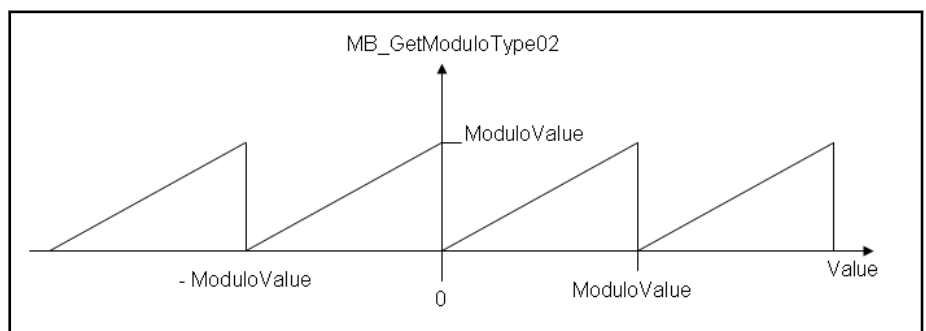


Fig. 3-89: Representation of the modulo function

Troubleshooting

The function does not generate any errors.

3.13.4 MB_GetModuloType03

Brief description

The function returns the modulo value. The returned value range is between - modulo value and + modulo value.

ML_TechBase.library

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx18	MX_Technology07.lib

Tab. 3-144: Reference table of the MB_GetModuloType03

Interface description

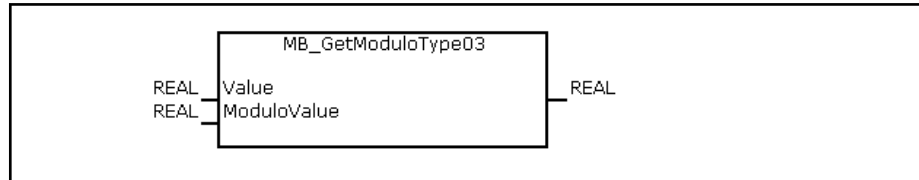


Fig. 3-90: MB_GetModuloType03 function

I/O type	Name	Data type	Description
VAR_INPUT	Value	REAL	Value at which the modulo function is to be applied
	ModuloValue	REAL	Modulo value to be used for the modulo operation
RETURN		REAL	Modulo value: The returned value range is between - modulo value and + modulo value.

Tab. 3-145: Interface variables of the MB_GetModuloType03 function

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
Value	REAL	- 1.175494e-38	3.402823e+38	0.0	With every call
ModuloValue	REAL	- 1.175494e-38	3.402823e+38	360.0	With every call

Tab. 3-146: Min./max. and default values of the MB_GetModuloType03 function

Functional Description

The function returns the modulo value. The returned value range is between - modulo value and + modulo value. The positive and the negative limit value are not part of the value range (also see the following diagram which shows the value transferred to the function on the x-coordinates and the value returned from the function on the y-coordinates).

If 0.0 is transferred for "ModuloValue", the function returns the value "Value".

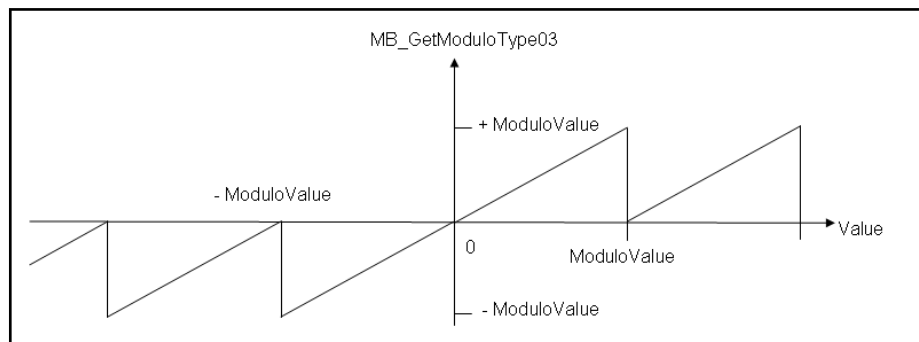


Fig. 3-91: Representation of the modulo function

Troubleshooting

The function does not generate any errors.

3.13.5 MB_GetModuloType04

Brief description

The function returns the modulo value. The returned value range is between - modulo / 2 and + modulo / 2.

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx18	MX_Technology07.lib

Tab. 3-147: Reference table of the MB_GetModuloType04

Interface description

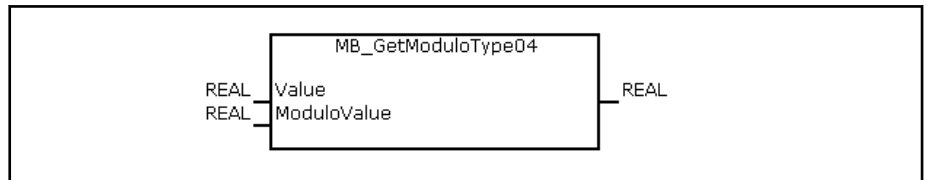


Fig. 3-92: MB_GetModuloType04 function

I/O type	Name	Data type	Description
VAR_INPUT	Value	REAL	Value at which the modulo function is to be applied
	ModuloValue	REAL	Modulo value to be used for the modulo operation
RETURN		REAL	Modulo value: The returned value range is between - modulo / 2 and + modulo / 2.

Tab. 3-148: Interface variables of the MB_GetModuloType04 function

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
Value	REAL	- 1.175494e-38	3.402823e+38	0.0	With every call
ModuloValue	REAL	- 1.175494e-38	3.402823e+38	360.0	With every call

Tab. 3-149: Min./max. and default values of the MB_GetModuloType04 function

Functional Description

The function returns the modulo value. The returned value range is between - modulo * 0.5 and + modulo * 0.5. The positive and the negative limit value are part of the value range (also see the following diagram which shows the value transferred to the function on the x-coordinates and the value returned from the function on the y-coordinates).

If 0.0 is transferred for "ModuloValue", the function returns the value "Value".

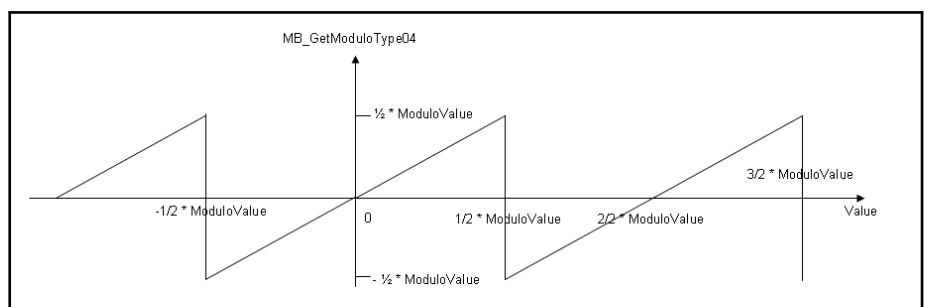


Fig. 3-93: Representation of the modulo function

Troubleshooting

The function does not generate any errors.

ML_TechBase.library

3.14 System information (Utilities\SystemInfo)

3.14.1 MB_GetSystemInfo

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

Assignment target system/library

Target system	Library
IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechBase
IndraMotion MLD with MPx07 without MA function package	MX_Technology_07.lib

Tab. 3-150: Reference table of the MB_GetSystemInfo

Interface description

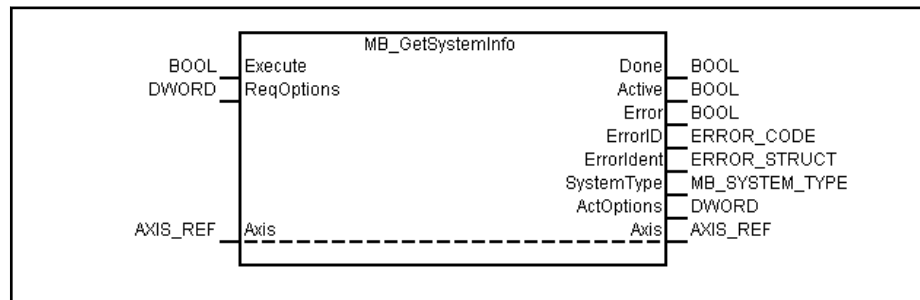


Fig. 3-94: MB_GetSystemInfo function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Axis	AXIS_REF	Axis to be checked
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	For the error structure with further error division, see Error codes, page 154
	SystemType	MB_SYSTEM_TYPE	System type MB_SYSTEM_TYPE, page 154
	ActOptions	DWORD	Actually enabled system options

Tab. 3-151: 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. 3-152: Input behavior of the ML_GetSystemInfo

Functional Description

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 ([Error codes, page 154](#)).

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	IndraMotion XLC	MLC/IndraLogic	IndraMotion MLD	Value	Description
MB_OPEN_LOOP	Axis		Axis	16#00000001	Basic package "Open Loop"
MB_CLOSED_LOOP	Axis		Axis	16#00000002	Basic package "ClosedLoop"
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. 3-153: Possible basic packages, function packages and options in "ReqOptions" and "ActOptions"

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.



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

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.

ML_TechBase.library

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. 3-154: Error codes of the MB_GetSystemInfo function block

3.14.2 MB_SYSTEM_TYPE

This enumeration type contains the currently available control systems.

System type		Description
MB_SYSTEM_UNKNOWN	0	Unknown system
MB_SYSTEM_MLC	1	IndraMotion MLC
MB_SYSTEM_MLD	2	IndraMotion MLD
MB_SYSTEM_MTX	3	IndraMotion MTX
MB_SYSTEM_MLP	4	IndraMotion MLP
MB_SYSTEM_XLC	5	IndraLogic XLC

Tab. 3-155: Possible system types

4 ML_TechMotion.library

4.1 Introduction and overview

The function blocks of the ML_TechMotion

Function block	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
Distance accumulator (DistanceAccumulator)	
MB_DistanceAccumulator, page 169	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 170	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.
Flying Shear (FlyingShear)	
ML_FlyingShear/MX_FlyingShear, page 171	Sequencing of a "FlyingShear"
Magic belt (MagicBelt)	
MB_MagicBeltType01, page 184	Contains functionalities to control a chain of a MagicBelt application
Position monitoring (PositionMonitoring)	
MB_PositionMonitoring_GantryCant, page 220	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 222	Monitors the absolute and relative position of two axes with regard to indentation and collision
Sag controller (SagControl)	
ML_SagControlA, page 229	Sag controller with analog actual value measurement of the sag
ML_SagControlC, page 226	Sag controller with binary min./max. contacts
S curve (SCurve)	
MB_ReachVelocityType02, page 233	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
Feed machine (SmartBelt)	
MB_SmartBeltType01, page 237	Executes a phase correction at a product using a triangular velocity profile

ML_TechMotion.library

Function block	Description
Cam tables for technology functions (TechFuncCamTable)	
MB_PilgrimStepCalcType01, page 260	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 271	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)
Tension controller (TensionControl)	
MB_TensionControlLoadCellType01, page 276	Depending on the measured value of a load cell, the tension controller controls the tension to the command value. The tension controller is directly parameterized via controller parameters
MB_TensionControlLoadCellMatParType01, page 287	Depending on the measured value of a load cell, the tension controller controls the tension to the command value. The tension controller is parameterized via material and distance parameters

Tab. 4-1: ML_TechMotion.library

4.2 Function blocks for cycle-synchronous lock-on and lock-off

4.2.1 Introduction and overview

By providing the CamLock technology function blocks, the IndraMotion functionality was extended by "Lock On/Lock Off".

[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.

[MB_CamLock, page 161](#): 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 slowed down in order to synchronize it to the master. The following figure shows a typical packaging machine for filling and sealing:

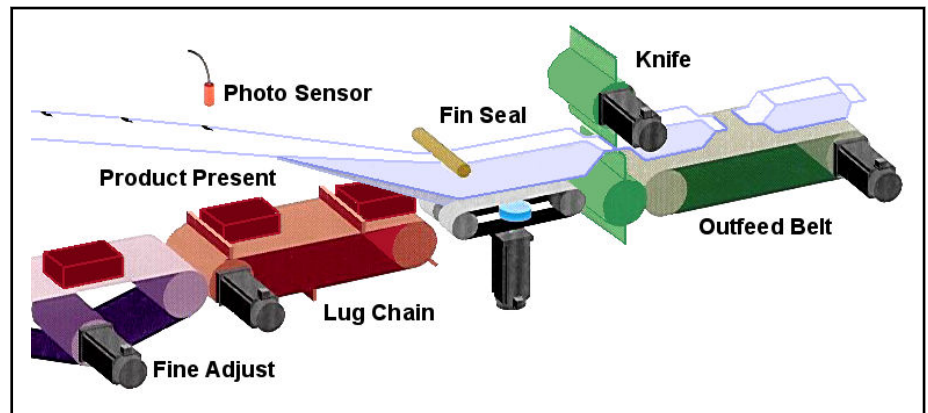


Fig. 4-1: Horizontal version of a packaging machine for filling and sealing

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.

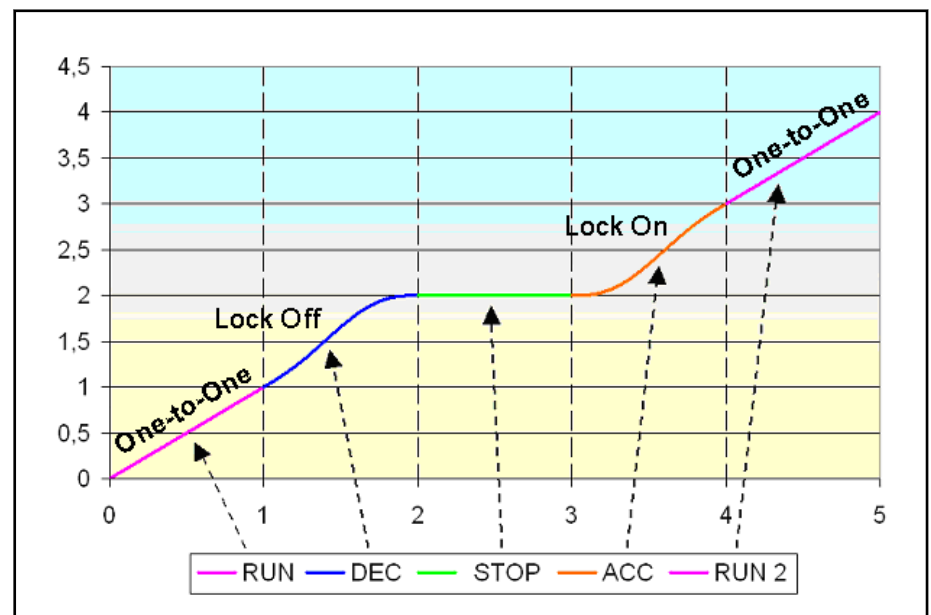


Fig. 4-2: CamLock cam profile

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.

ML_TechMotion.library

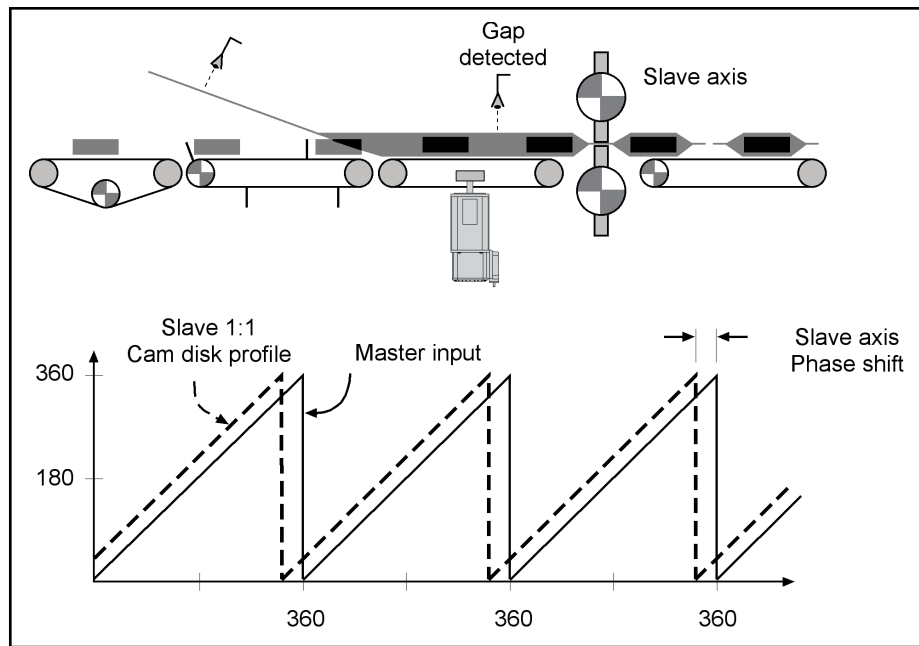


Fig. 4-3: Processing cam active, normal operation of the packaging application

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.

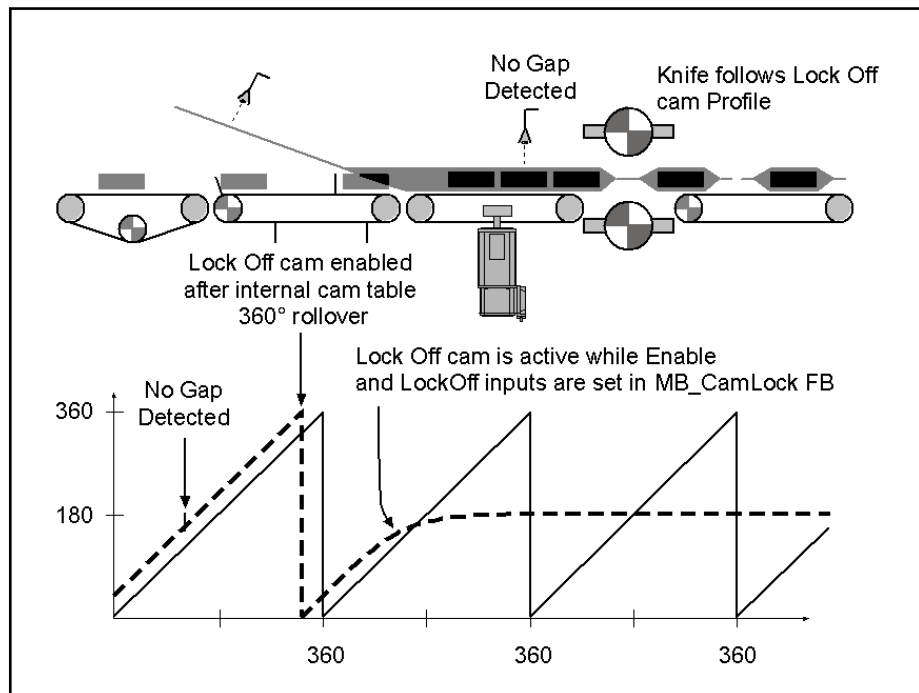


Fig. 4-4: Lock Off cam active, No Product - No Seal

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.

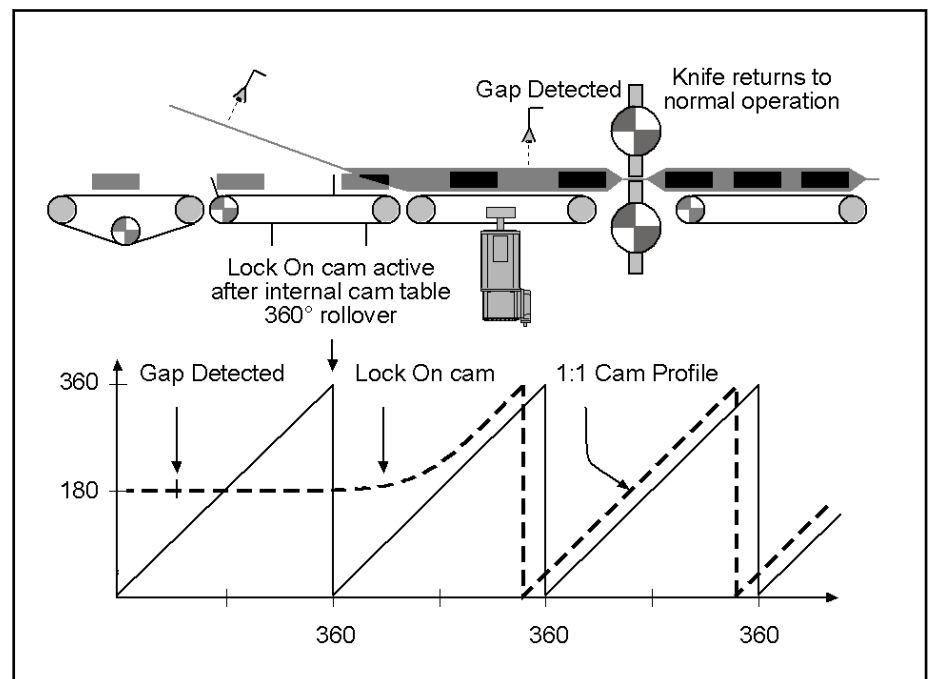


Fig. 4-5: Lock On cam active, product present

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	Library
IndraMotion MLC 13VRS	ML_TechMotion

Tab. 4-2: Reference table of the MB_PrepareCams

ML_TechMotion.library

Interface description

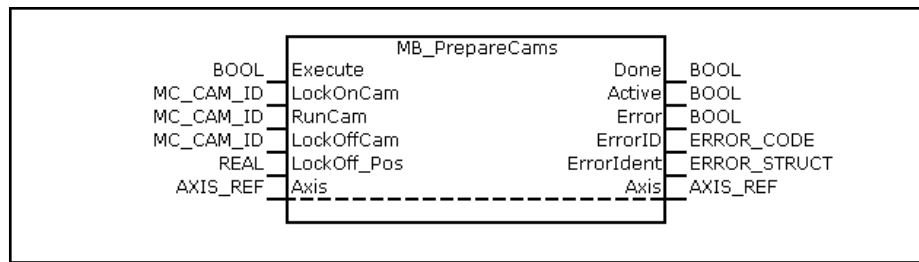


Fig. 4-6: MB_PrepareCams function block



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 IndraMotion 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 AXIS VAR_IN_OUT is still required.

The calculated cam profiles are stored in control parameters.

I/O type	Name	Data type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the slave axis for which cam profiles are saved
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 MC_CAM_ID from RIL_CommonTypes.library
	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 Troubleshooting, page 161

Tab. 4-3: Interface variables of the MB_PrepareCams function block

Functional Description

An IndraMotion MLC system 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 the MB_CamLock function block ([MB_CamLock](#), page 161).

Troubleshooting

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. 4-4: Error codes of the MB_PrepareCams function block

4.2.4 MB_CamLock

Brief description

The CamLock function block MB_CamLock The MB_CamLock function block is used to enable the Run, Lock On and Lock Off cam profiles calculated and stored by the MB_PrepareCams 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

ML_TechMotion.library

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechMotion

Tab. 4-5: Reference table of the MB_CamLock

Interface description

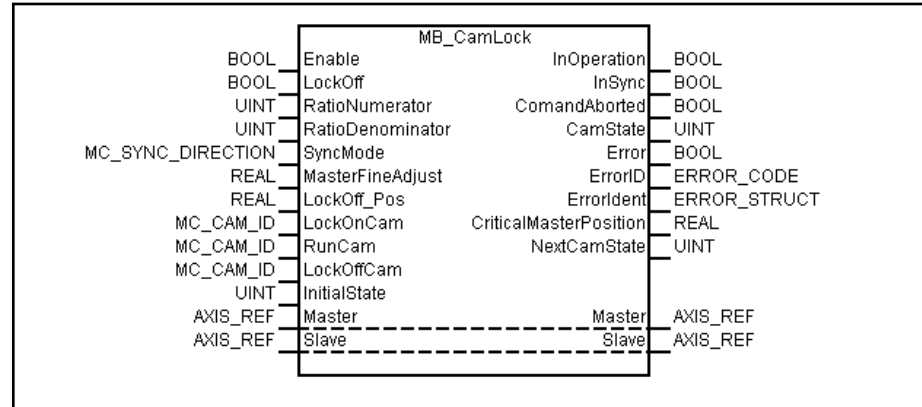


Fig. 4-7: MB_CamLock function block



For an IndraMotion MLC control system, 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. 4-6: Normal "CamLock" operation

I/O type	Name	Data type	Description
VAR_IN_OUT	Slave	AXIS_REF	Real or virtual axis
	Master	AXIS_REF	Real or virtual axis
VAR_INPUT	Enable	BOOL	Enables the MB_CamLock FB. A rising edge dynamically synchronizes the slave 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 locks off (LockOff)
	RatioNumerator	UINT	Electronic gear ratio

I/O type	Name	Data type	Description
	RatioDenominator	UINT	Electronic gear ratio
	SyncMode	MC_SYNC_DIRECTION	0: = Shortest distance 1: = positive direction 2: = negative direction MC_SYNC_DIRECTION from RIL_CommonTypes.library
	MasterFineAdjust	REAL	This input is required for the MB_MotionProfile function block
	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 MC_CAM_ID from RIL_CommonTypes.library
	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: In Standstill (stop state) 5: in LockOn (locked on state) 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 Troubleshooting, page 167
	CriticalMasterPosition	REAL	Provides the critical master position required for the switching state transition
	NextCamState	UINT	Indicates the next output to be processed

Tab. 4-7: Interface variables of the MB_CamLock function block

ML_TechMotion.library

Signal-time diagram

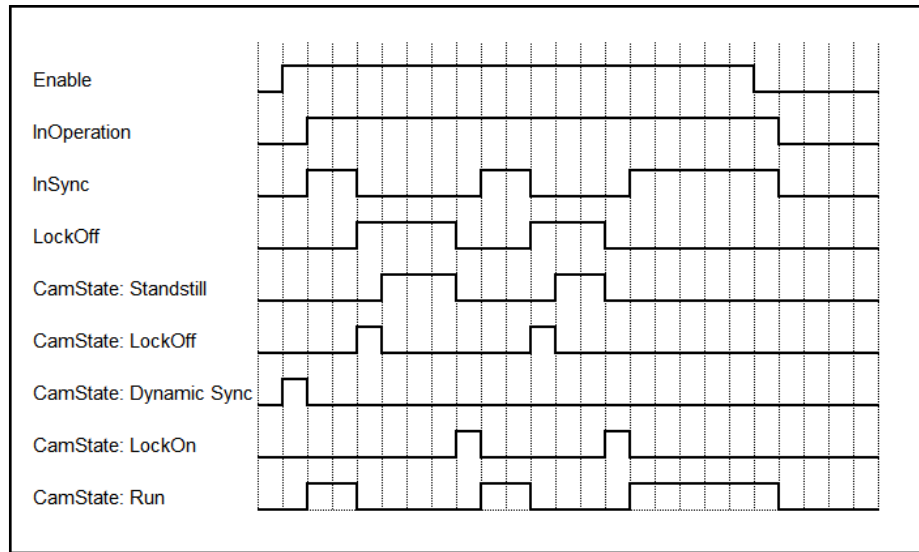


Fig. 4-8: Signal-time diagram of the MB_CamLock

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 remain 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.

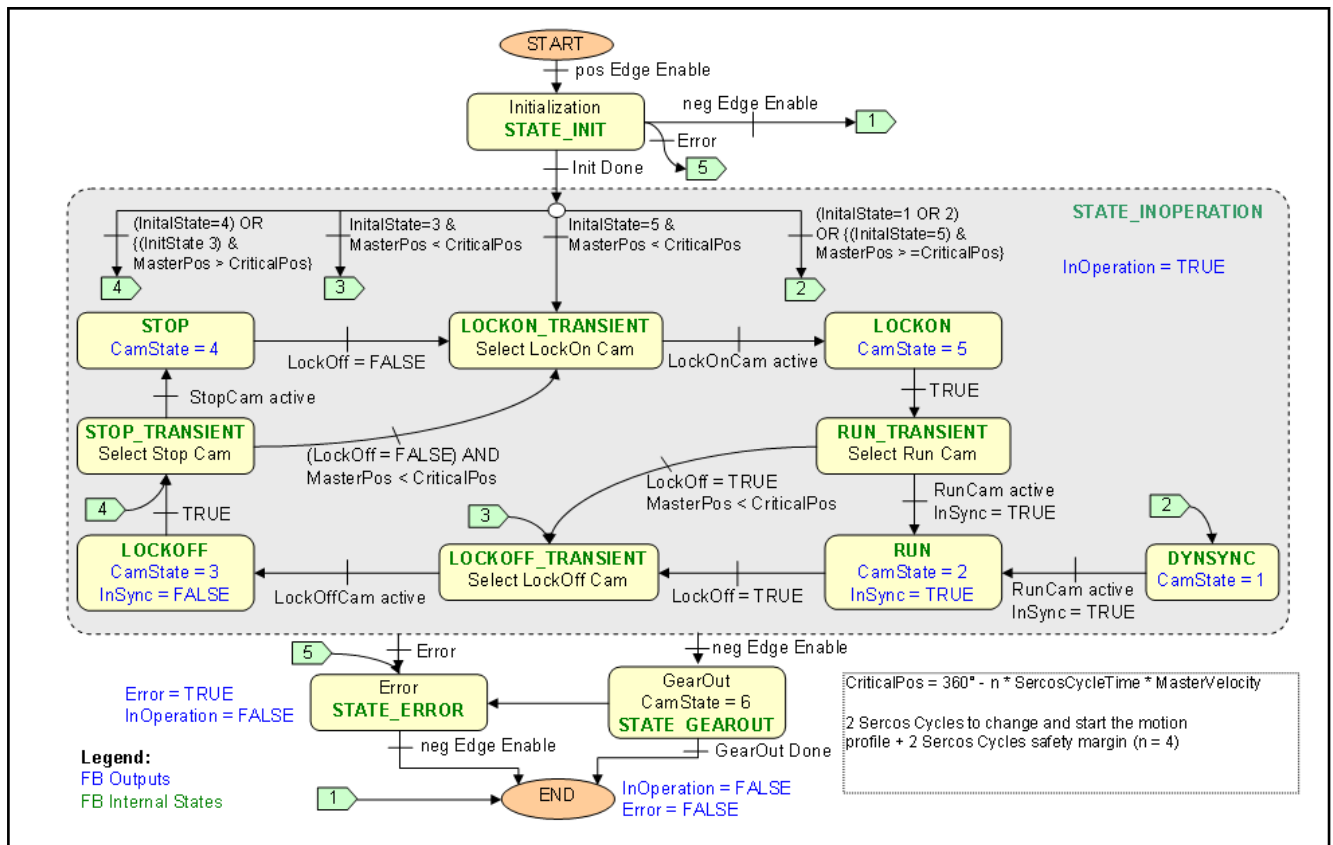
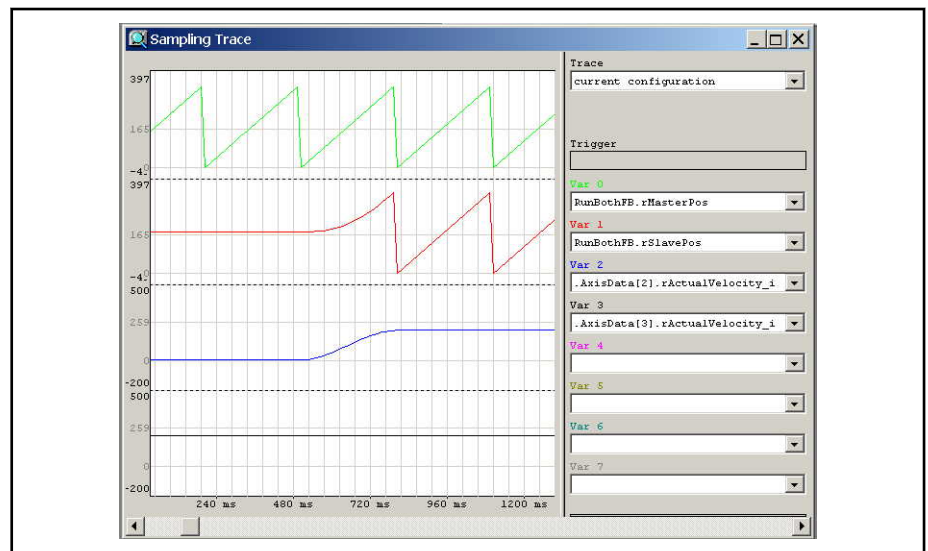


Fig. 4-9: CamLock state machine

LockOn/LockOff sequence examples



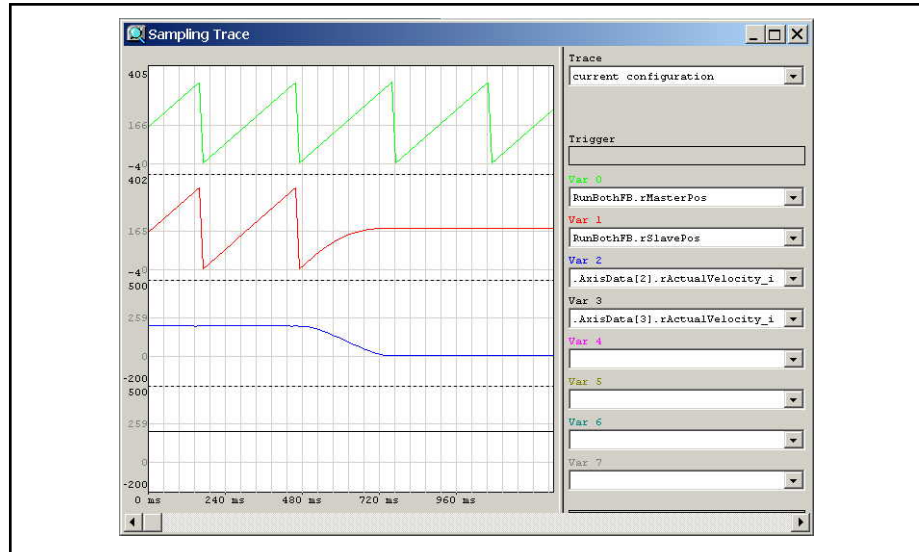
Channel 0 (green) = master position
Channel 1 (red) = slave position
Channel 2 (blue) = slave velocity
Master velocity = 200 rpm

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

ML_TechMotion.library

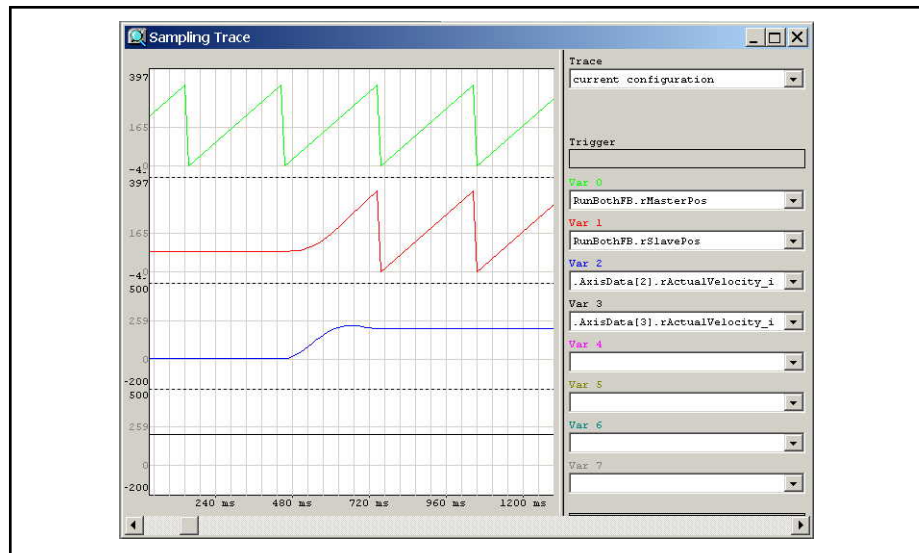
shows the position and velocity profile during the "LockOff" profile without jumps and overshoots. The "LockOff" position is 180 degrees (default).



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.

Troubleshooting

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

ML_TechMotion.library

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 User-Cam_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. 4-8: Error codes of the MB_CamLock function block

4.2.5 MB_CamLock Parameterization

The CamLock function blocks 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 MB_PrepareLockCams function block 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.
8. Note: The CamID 99 is always used as stop Cam.

4.3 Distance accumulator (DistanceAccumulator)

4.3.1 Introduction and overview

The function block [MB_DistanceAccumulator](#), page 169, sums up **any** relatively traveled distance via an encoder.

The function block [MB_AxisDistanceAccumulator](#), page 170, sums up the relatively traveled distance **of one axis** via an encoder.

4.3.2 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.

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechMotion

Tab. 4-9: Reference table of the MB_DistanceAccumulator

Interface description

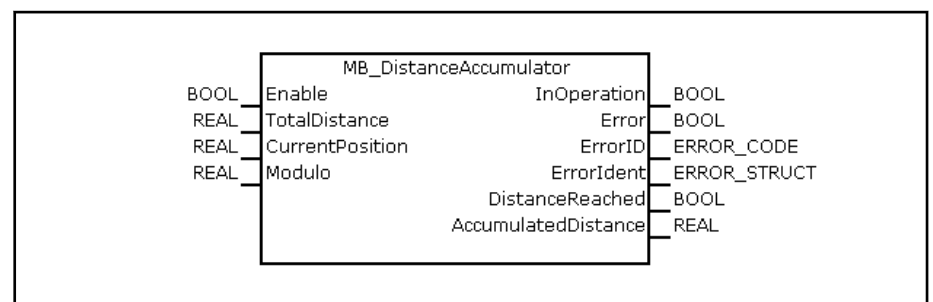


Fig. 4-14: MB_DistanceAccumulator function block

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: rotary 0: linear) In case of a Servo axis, this value can be read from A-0-0045 and/or S-0-0103
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 169
	DistanceReached	BOOL	TotalDistance was reached and the processing of the function block is completed.
	AccumulatedDistance	REAL	Indicates the currently traveled distance

Tab. 4-10: 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**.

ML_TechMotion.library

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. 4-11: Error codes of the MB_DistanceAccumulator function block

4.3.3 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.

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechMotion

Tab. 4-12: Reference table of the MB_AxisDistanceAccumulator

Interface description

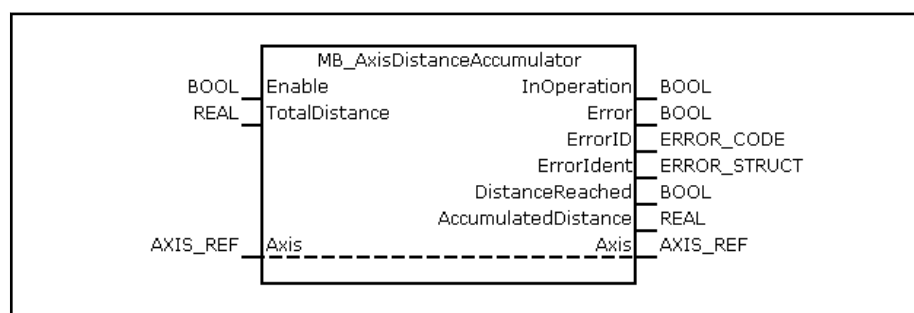


Fig. 4-15: MB_AxisDistanceAccumulator function block

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 171
	DistanceReached	BOOL	TotalDistance was reached and the processing of the function block is completed
	AccumulatedDistance	REAL	Indicates the currently traveled distance
VAR_IN_OUT	Axis	AXIS_REF	Axis reference

Tab. 4-13: 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. 4-14: Error codes of the MB_AxisDistanceAccumulator function block

4.4 Flying Shear (FlyingShear)

4.4.1 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 171](#), 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 IndraMotion MLD: Function block MX_FlyingShear with IndraLogic 1.x
- Target system IndraMotion MLC: Function block ML_FlyingShear with IndraLogic 2G

The following description refers to both function blocks.

For the system-specific settings, refer to

- Target system IndraMotion MLD: see [MX_FlyingShear - Special Features for the IndraMotion MLD, page 180](#)
- Target system IndraMotion MLC: see [ML_FlyingShear - Special Features of IndraMotion MLC, page 178](#)

4.4.2 ML_FlyingShear/MX_FlyingShear

Brief description The function block ML_FlyingShear/MX_FlyingShear contains the typical procedure of a FlyingShear and performs the following steps once the input "Start" was set to TRUE:

ML_TechMotion.library

- 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 IndraMotion MLC, note the commissioning notes in [ML_FlyingShear - Special Features of IndraMotion MLC](#), page 178.

and for the target system IndraMotion MLD, note the commissioning notes in [MX_FlyingShear - Special Features for the IndraMotion MLD](#), page 180.

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechMotion
IndraMotion MLD with MPx07/08/17 with MA function package	MX_Technology07.lib

Tab. 4-15: Reference table of the ML_FlyingShear/MX_FlyingShear

Interface description

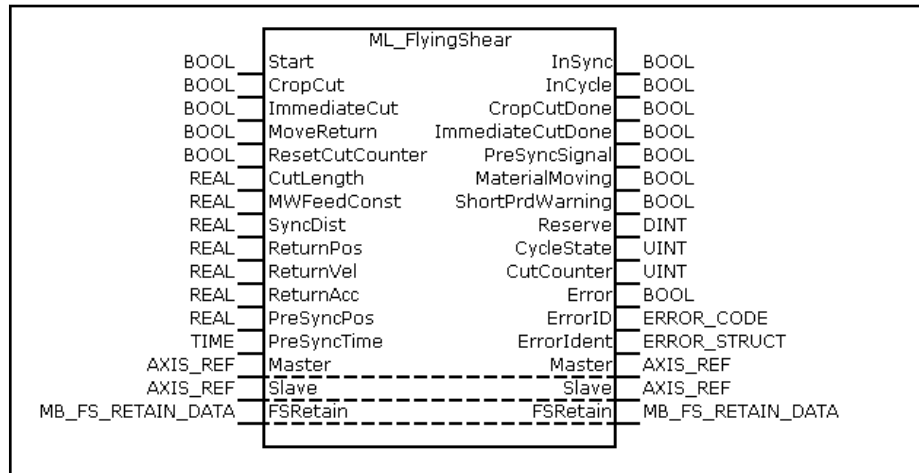


Fig. 4-16: Function block MX(L)_FlyingShear

I/O type	Name	Data type	Explanation
VAR_IN_OUT	Master	AXIS_REF	Reference to the master axis
	Slave	AXIS_REF	Reference to the slave axis (machining station)
	FSRetain	MB_FS_RETAIN_DATA	Reference to the required retain data of this function block MB_FS_RETAIN_DATA , page 178

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
	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

ML_TechMotion.library

I/O type	Name	Data type	Explanation
	MWFeedconst	REAL	Feed constant of the measuring wheel per revolution *1*2
	SyncDist	REAL	Synchronization distance required by the slave axis to synchronous 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 "Return-Pos" with the velocity ReturnVel *1*2
	ReturnAcc	REAL	The slave axis returns to the starting position "Return-Pos" with the acceleration ReturnAcc *1*2
	PreSyncPos	REAL	The output signal "PreSyncSignal" is set before the synchronous position "PreSyncPos" with the distance given prepare before the synchronous run*2
	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
	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") • Shorten 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 stand-still is only possible if MaterialMoving=FALSE

I/O type	Name	Data type	Explanation
	Reserve	DINT	Reserved increments of the synchronization procedure for analysis purposes. A synchronization is not possible if $\text{Reserve} \leq 0$ -> 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	INT (Enum)	ERROR_CODE: Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description

*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. 4-16: 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):

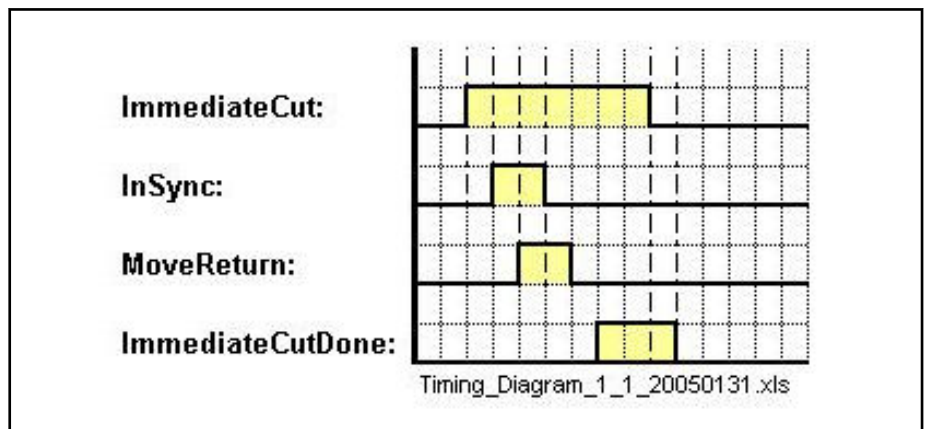


Fig. 4-17: Signal-time diagram: Immediate cut at material standstill

The following chart shows the complete signal flow of the function block "FlyingShear" with continuously passing material and immediate cut:

ML_TechMotion.library

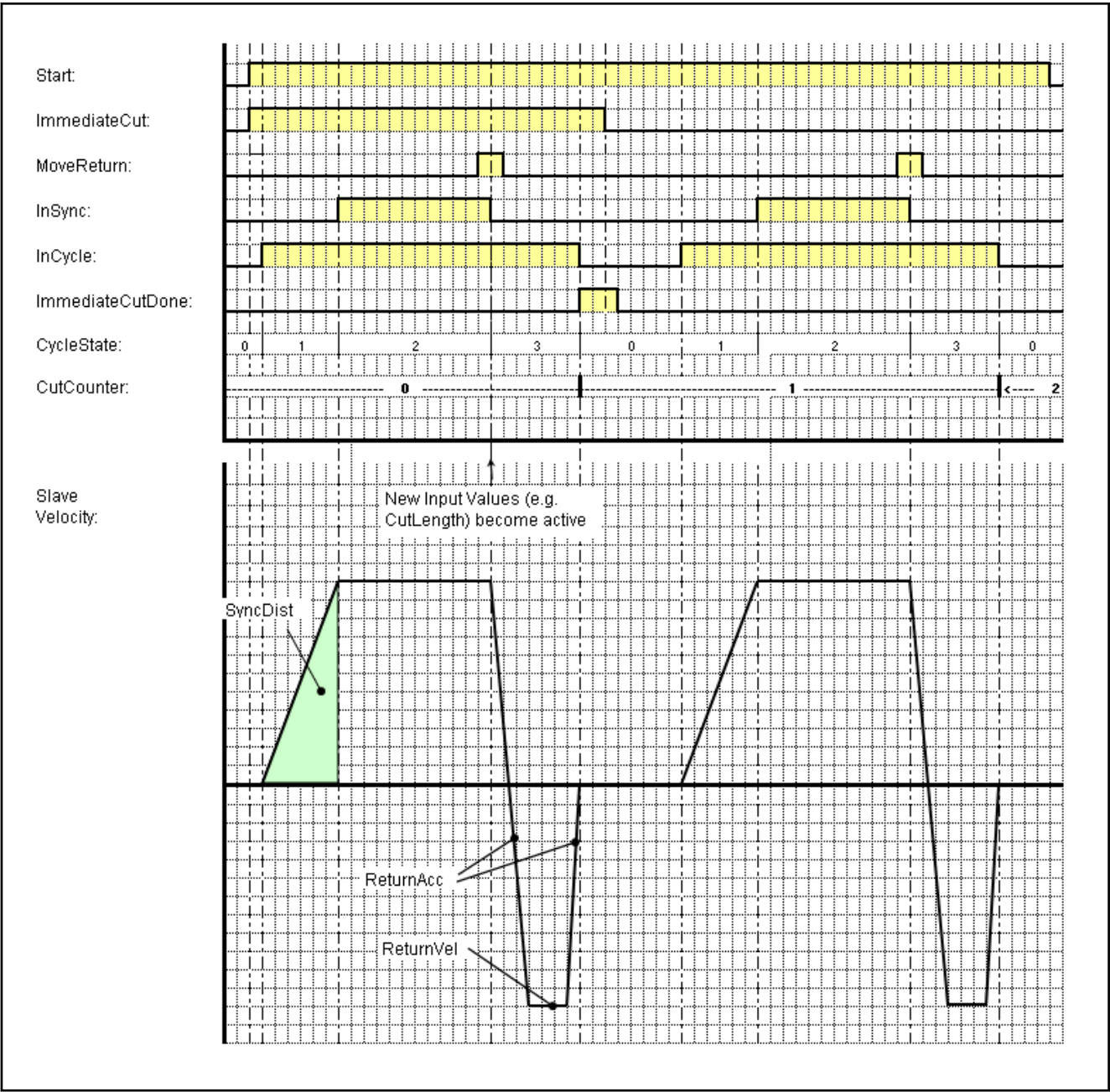


Fig. 4-18: Complete signal flow of the function block "FlyingShear" with immediate cut

Troubleshooting The function block "MX(L)_FlyingShear" generates the following error messages in Additional1/Additional2 for the table "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 AxisData 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#xxx01000
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

ML_TechMotion.library

ErrorID	Additional1	Additional2	Description
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
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. 4-17: Error codes of the ML_FlyingShear/MX_FlyingShear function block

4.4.3 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

```

4.4.4 ML_FlyingShear - Special Features of IndraMotion 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 a proper functionality of the function block "MX(L)_FlyingShear".

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 FlyingShear 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 function block "MX(L)_FlyingShear" contains high-prior Motion commands and should therefore be cyclically called in a high-prior, cyclical PLC task with a cycle time $\leq 4\text{ms}$ (MotionTask)

ML_TechMotion.library

- Before starting the function block, "MC_Power" has to be processed without errors

4.4.5 MX_FlyingShear - Special Features for the IndraMotion 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

Note:

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 the 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.

4. Configure the mechanical axis settings of the flying shear (linear axis, mo modulo, travel range, range limits etc.) of the given application accordingly.
5. Configure the additional second (measuring) encoder.
- Switch IndraWorks to online mode and open the axis dialog **Measuring Systems ▶ Optional Encoder**
6. Ensure that the measuring encoder moves towards the positive direction since the synchronization mechanism can only be used for the positive direction.

ML_TechMotion.library

- 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. 4-18: Relevant parameters for the MX_FlyingShear

4.5 Magic belt (MagicBelt)

4.5.1 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 ([MB_MagicBeltType01](#), page 184). The function blocks communicate with each other using the applied data structures.

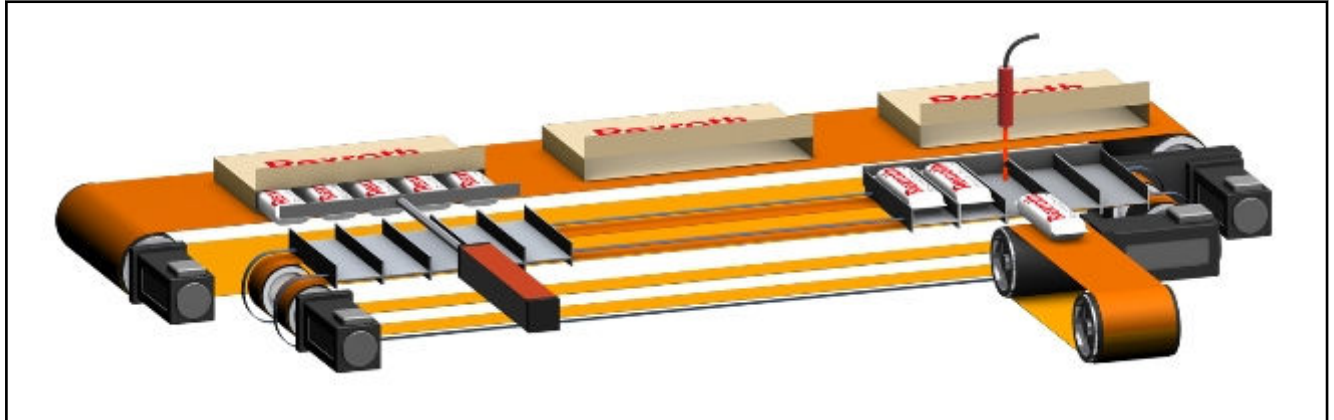


Fig. 4-19: MagicBelt application

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:

$$\text{ModuloValue} = \frac{\text{ChainLength}}{\text{TrainsPerChain}}$$

Fig. 4-20:

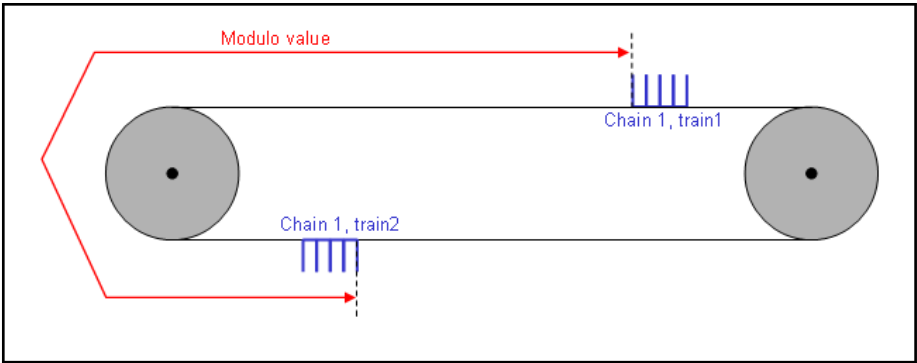


Fig. 4-21: MagicBelt definitions: Modulo value

4.5.2 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:** The trains can be jogged individually or synchronously, optionally continuous, or incrementally
- **Tune mode:** To determine dynamic limit values (such as the maximum acceleration), the trains are synchronized and continuously traveled by a set distance
- **Automatic mode:** 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, note the commissioning instructions at the end of this chapter.

Assignment: Target system/library

Target system	Libraries
IndraMotion MLC 13VRS	ML_TechMotion
TEC function package	

Tab. 4-19: 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

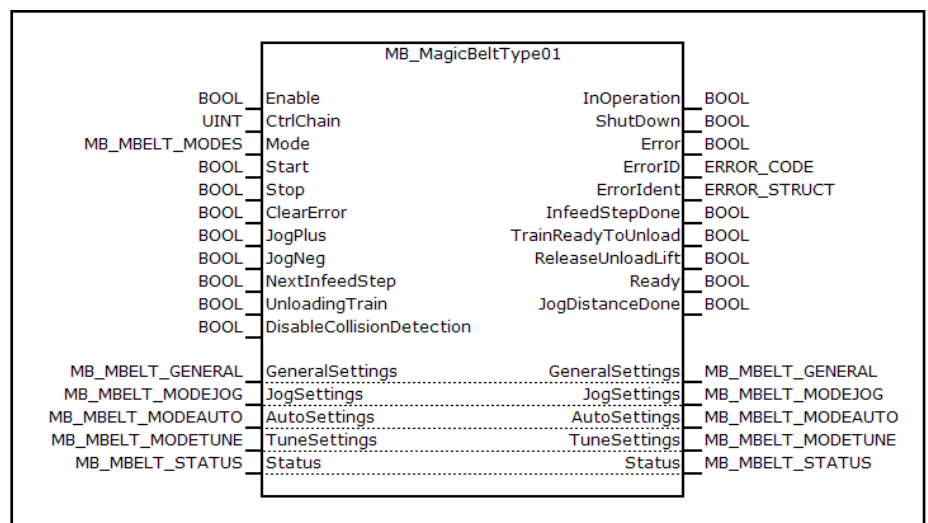


Fig. 4-22: MB_MagicBeltType01 function block

The user program controls the chains using the function block inputs (e.g. Start, Stop...). The configuration data for operation are 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 206
	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.

ML_TechMotion.library

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 away 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 203
	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 206
	JogSettings	MB_MBELT_MODEJOG	Jog mode settings MB_MBELT_MODEJOG, page 209
	AutoSettings	MB_MBELT_MODEAUTO	Automatic mode settings MB_MBELT_MODEAUTO, page 210
	TuneSettings	MB_MBELT_MODETUNE	Tune mode settings MB_MBELT_MODETUNE, page 209
	Status	MB_MBELT_STATUS	Status information on the chains MB_MBELT_STATUS, page 218

Tab. 4-20: 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. 4-21: Min./max. and default values of the MB_MagicBeltType01 function block

Data structures used

Data Structures Used

Structure // Substructure	Purpose
MB_MBELT_MODES, page 206	Operation modes of MagicBelt function block
MB_MBELT_GENERAL, page 206	Overall operation mode settings (e.g. chain length)
MB_MBELT_MODEJOG, page 209	Settings for Jog operation mode (e.g. Jog speed)
MB_MBELT_MODETUNE, page 209	Settings for Tune operation mode (e.g. Tune speed)
MB_MBELT_MODEAUTO, page 210	Settings for Auto operation mode (e.g. Tune speed)
MB_MBELT_STARTTYPE, page 211	Start type (e.g. START_AT_BEGIN)
MB_MBELT_STOPTYPE, page 211	Stop type (e.g. STOP_IMMEDIATE)
MB_MBELT_INFEED, page 211	Settings for loading in Auto operation mode
MB_MBELT_FEEDTYPE, page 214	Loading procedure (indexed, synchronized, with velocity)
MB_MBELT_INFEEDSTEPS, page 214	Specific chain data for loading in Automatic mode
MB_MBELT_OUTFEED, page 216	Settings for unloading in Auto operation mode
MB_MBELT_MOVE, page 217	Settings for traveling between Infeed ↔ Outfeed in Auto operation mode
MB_MBELT_STATUS, page 218	Chain-comprehensive status information

ML_TechMotion.library

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.

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 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.

"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 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 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 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".

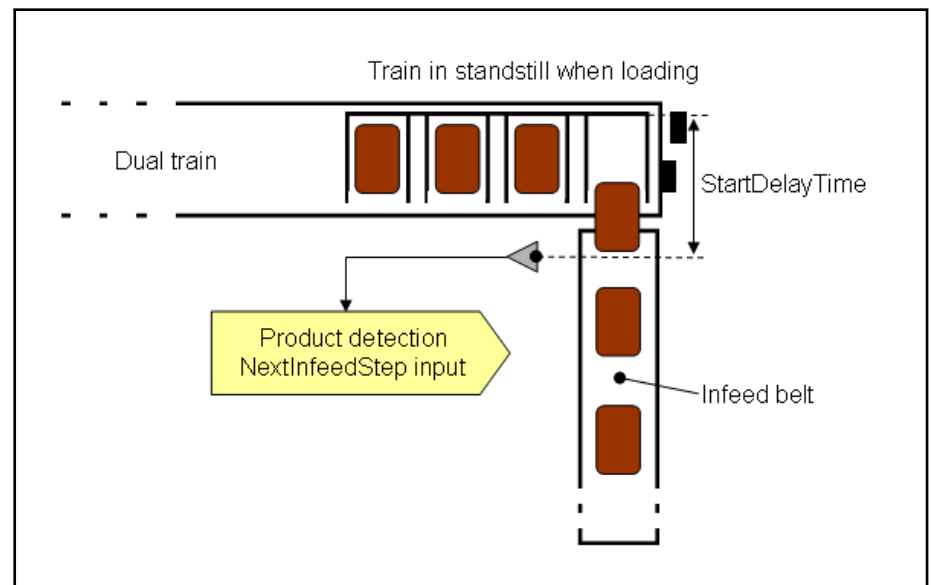


Fig. 4-23: Indexed product feed/loading at standstill from the front

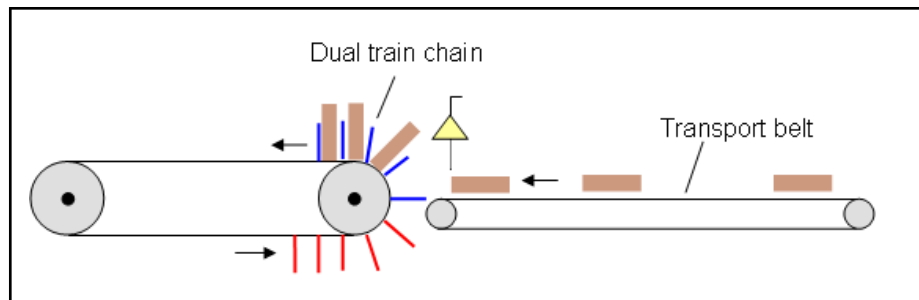


Fig. 4-24: Indexed product feed/loading at standstill from the side

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.

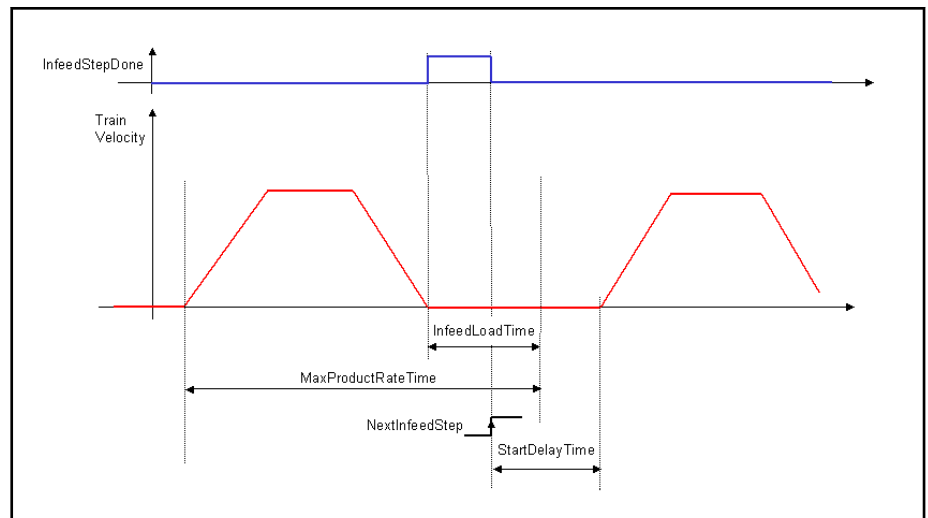


Fig. 4-25: Overview and mode of operation of "InfeedLoadTime", "StartDelayTime" and "MaxProductRate"

MaxMoveVelocity, MaxMoveAcceleration, MaxMoveJerk setting

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:

X_FIT_ACC_TRAPEZE_JLIM

If the motion law **X_FIT_ACC_TRAPEZE_JLIM** is used for the Infeed motion, the **MaxMove...** input values have to be set as follows:

- MaxMoveVelocity $\geq$ MaxInfeedVelocity
- MaxMoveAcceleration $\geq$ MaxInfeedAcceleration
- MaxMoveJerk $\geq$ MaxInfeedJerk

VELOCITY_IN_VELOCITY_MOD_SINE

If the motion law **VELOCITY_IN_VELOCITY_MOD_SINE** is used for the Infeed motion, the **MaxMove...** input values 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. 4-26: Calculation of MaxMoveVel
 $S_{\text{max\_infeed}}$ = Maximum Infeed step width

Setting MaxMoveAcceleration:

$$\text{MaxMoveAcceleration} \left[\frac{\text{mm}}{\text{s}^2} \right] \geq 5,53 \cdot \left(\frac{S_{\text{max\_infeed}} [\text{mm}]}{\left(\frac{60 \left[\frac{\text{s}}{\text{min}} \right]}{\text{MaxProductRate} [\text{min}^{-1}]} - \frac{\text{InfeedLoadTime} [\text{ms}]}{1000 \left[\frac{\text{ms}}{\text{s}} \right]} \right)^2} \right)^2$$

Fig. 4-27: Calculation of MaxMoveAcceleration
 $S_{\text{max\_infeed}}$ = Maximum Infeed step width

Setting MaxMoveJerk:

ML_TechMotion.library

$$\text{MaxMoveJerk} \left[\frac{\text{mm}}{\text{s}^3} \right] \geq 69,47 \cdot \left(\frac{S_{\text{max\_infeed}} [\text{mm}]}{\left(\frac{60 \left[\frac{\text{s}}{\text{min}} \right]}{\text{MaxProductRate} [\text{min}^{-1}]} - \frac{\text{InfeedLoadTime} [\text{ms}]}{1000 \left[\frac{\text{ms}}{\text{s}} \right]} \right)^3} \right)$$

Fig. 4-28: 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 **MaxMove...** input values 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. 4-29: 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

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).

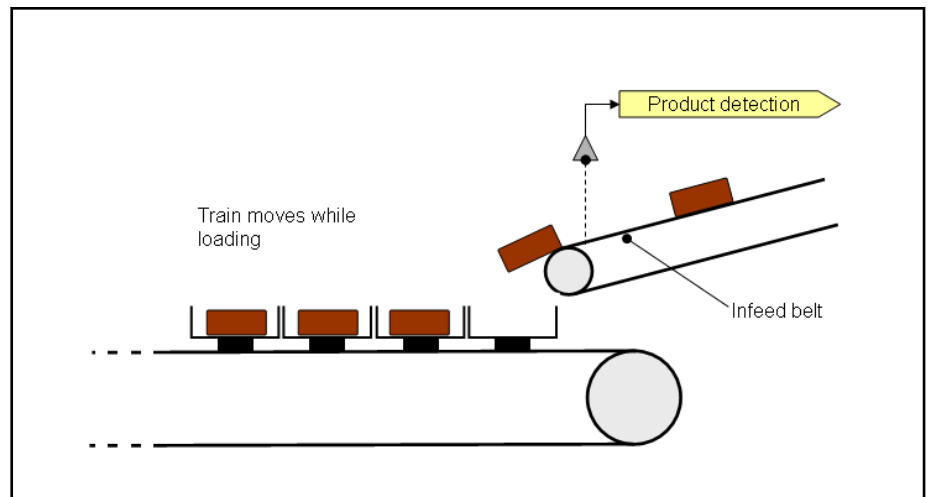


Fig. 4-30: Indexed product feed/loading when the train is moving



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

In principle, 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 status 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

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:

$$\varphi_{MaxProdWindowEnd} [^\circ] = 360^\circ - 3 \cdot t_{SercosCycle} [ms] \cdot v_{InfeedMaster} [min^{-1}] \cdot 0,006$$

Fig. 4-31: MB_MBELT_STATUS.MaxProdExpectationWindowEnd calculation

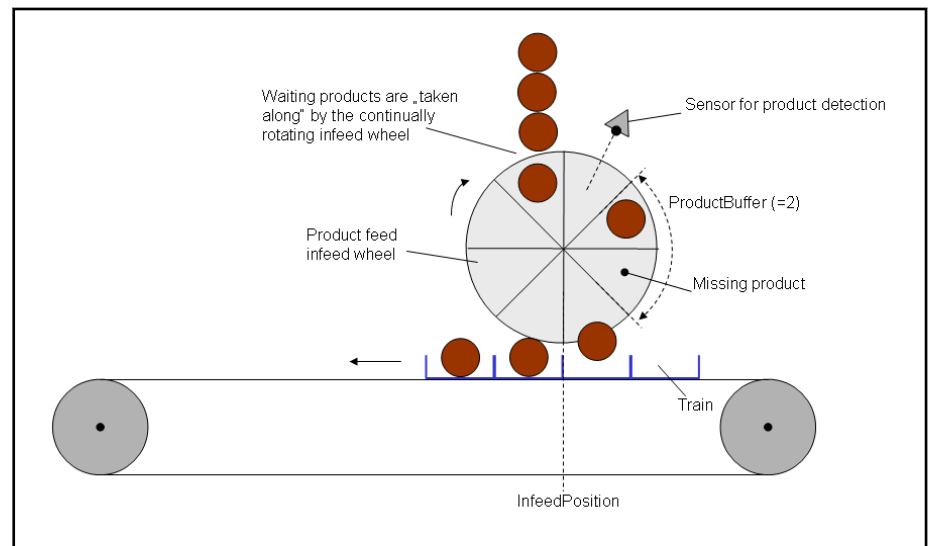


Fig. 4-32: Synchronous product feed

Example of synchronous product feed (temporal sequence):

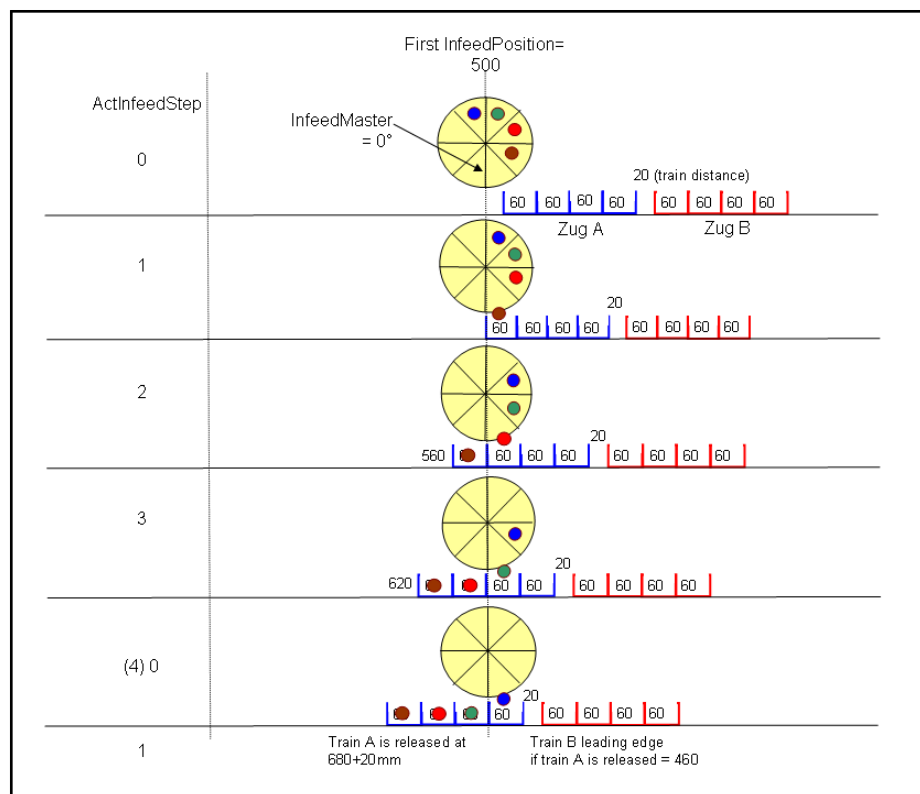


Fig. 4-33: Synchronous product feed (temporal sequence)

Motions between the loading and unloading stations

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

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".

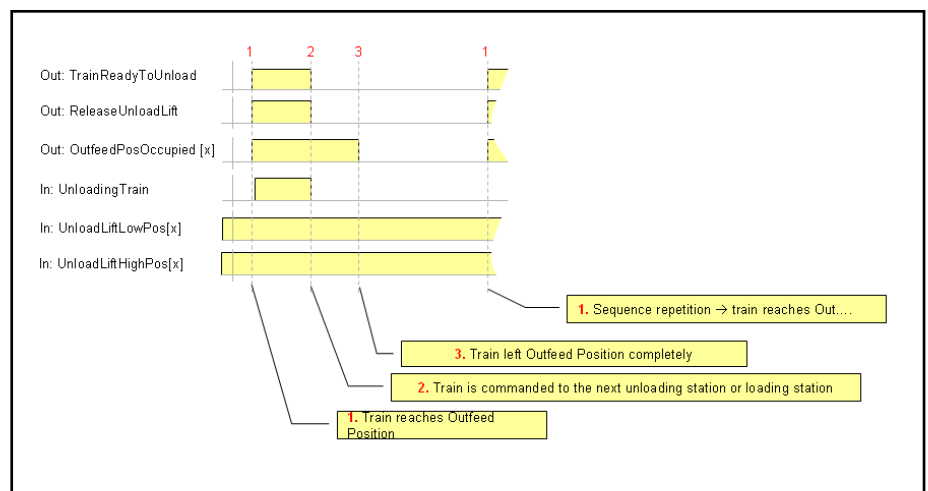


Fig. 4-34: Signal-time diagram when unloading without unload handshake



13VRS or 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.

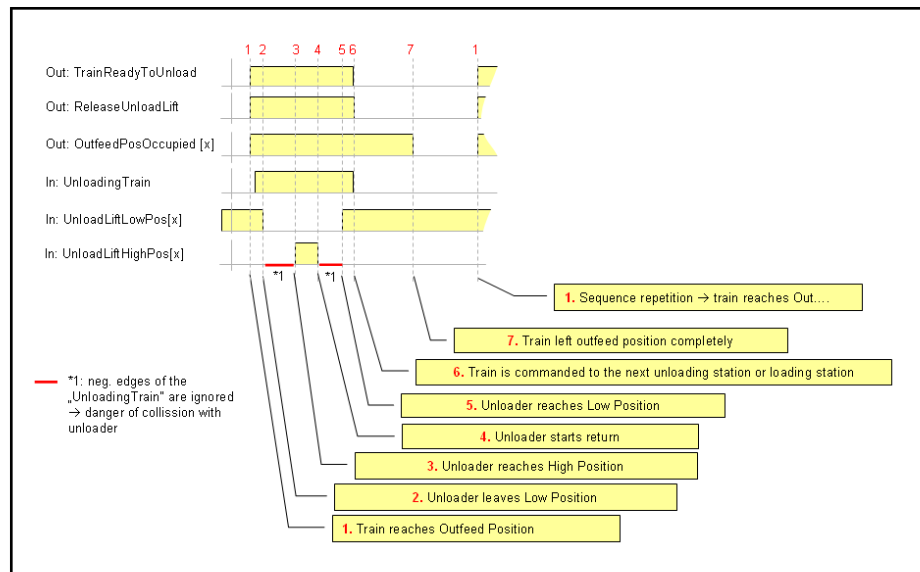


Fig. 4-35: Signal-time diagram to unload with handshake/continuing train traveling when the unloader is retracted

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.

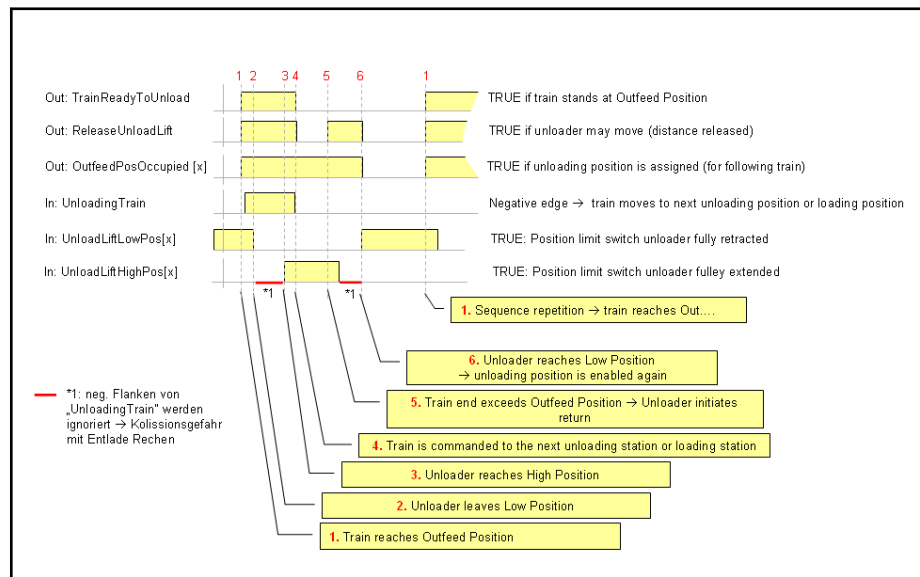


Fig. 4-36: Signal-time diagram when unloading with handshake/continuing train traveling when the unloader is extended

Stop 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).

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.Sto-

ML_TechMotion.library

pRamp"). 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 are 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 do 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.

Error handling 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 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.

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	Infeed positioning time ≤ 0 ms
INPUT_INVALID_ERROR	16#1A00	16#0002	Products per pocket = 0
INPUT_INVALID_ERROR	16#1A00	16#0003	Feedrate override ≤ 0 OR > 100
INPUT_INVALID_ERROR	16#1A00	16#0004	CtrlChain ≤ 0 OR > 3
INPUT_INVALID_ERROR	16#1A00	16#0005	ChainLength ≤ 0
INPUT_INVALID_ERROR	16#1A00	16#0006	NoOfChains ≤ 0 OR > 3
INPUT_INVALID_ERROR	16#1A00	16#0007	TrainsPerChain ≤ 0 OR > 3
INPUT_INVALID_ERROR	16#1A00	16#0008	ProductBuffer ≤ 0
INPUT_INVALID_ERROR	16#1A00	16#0009	InfeedAxis AxisRef.AxisNo 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

ML_TechMotion.library

ErrorID	Additional1	Additional2	Description
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
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 AxisRef 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

ErrorID	Additional1	Additional2	Description
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
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. 4-22: Error codes of the MB_MagicBeltType01 function block

Commissioning Notes

The following has to be considered when commissioning the "MB_MagicBeltType01" 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 for the chain drive **Chain A, B, (C)** have to be created using IndraWorks

Note the following parameterization:

- Axis type: Real or virtual (for testing purposes)
- When the drive is added, "Interpolation in the drive" **has to be de-selected** (command position value generation on the control)

ML_TechMotion.library

- 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
Note 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
Note 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 then have to be assigned to the corresponding axes (e.g. "ChainAxes") (via AxisRef) using the data structure [MB_MBELT_GENERAL](#), page 206,

4.5.3 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. 4-23: MB_MBELT_MODES enumeration type elements

4.5.4 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

Structure element	Type	Default value	Effective	Description
TrainsPerChain	UINT	1	Pos. edge "Enable"	Trains per chain Minimum = 1 Maximum = 4
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

ML_TechMotion.library

Structure element	Type	Default value	Effective	Description
Feedrate	UINT	100	Continuous	All velocities in the "Auto Mode" are reduced accordingly if the "FeedrateOverride" < 100% is specified. Minimum: 10% Maximum: 100%
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. 4-24: Interface description and default values of the MB_MBELT_GENERAL data structure

4.5.5 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. 4-25: Interface description and default values of the MB_MBELT_MODEJOG data structure

4.5.6 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.

ML_TechMotion.library

Structure element	Type	Default value	Effective	Description
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. 4-26: Interface description and default values of the MB_MBELT_MODE-TUNE data structure

4.5.7 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 211
StopType	MB_BELT_STOPTYPE	STOP_CYCLE_COMPLETE	Defines the stop behavior for Automatic mode MB_MBELT_STOPTYPE, page 211 Application at positive "Stop" or negative edge at "Enable"
Infeed	MB_BELT_INFEED		Contains data to load the trains MB_MBELT_INFEED, page 211
Outfeed	MB_BELT_OUTFEED		Contains data to unload the trains MB_MBELT_OUTFEED, page 216
MoveTo	MB_BELT_MOVE		Contains data for the motion between loading and unloading MB_MBELT_MOVE, page 217

Tab. 4-27: Interface description and default values of the MB_MBELT_MODEAUTO data structure

4.5.8 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. 4-28: Element of the MB_MBELT_STARTTYPE enumeration type

4.5.9 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. 4-29: Element of the MB_MBELT_STOPTYPE enumeration type

4.5.10 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 214)
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

ML_TechMotion.library

Structure element	Type	Default value	Effective	Description
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
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

Structure element	Type	Default value	Effective	Description
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
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°

ML_TechMotion.library

Structure element	Type	Default value	Effective	Description
FirstInfeedPosition	REAL	500.0	Continuous	First (absolute) loading position Minimum = 0 Maximum = ModuloValue
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 214

Tab. 4-30: Interface description and default values of the MB_MBELT_INFEED data structure

4.5.11 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. 4-31: Elements of the MB_MBELT_FEEDTYPE enumeration type

4.5.12 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. 4-32: 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

ML_TechMotion.library

ChainA_InfeedData[4]. ProductsPerPocket = don't mind

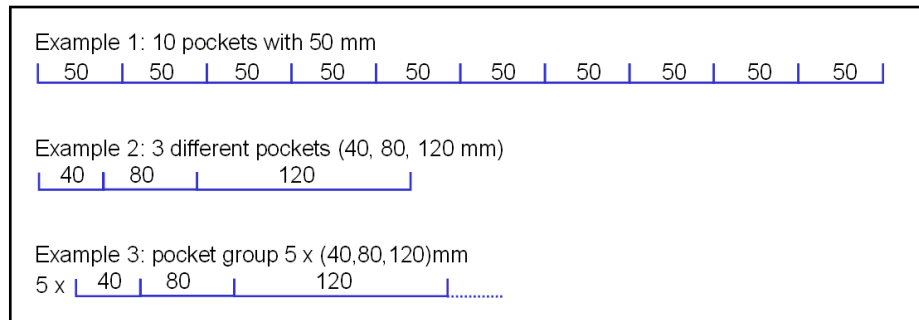


Fig. 4-37: Train configurations of the examples shown above

4.5.13 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. 4-33: Interface description and default values of the MB_MBELT_OUTFEED data structure

4.5.14 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: $\text{MaxMoveAcceleration} * \text{LoadTrainAcceleration} [\%] / 100 [\%]$

ML_TechMotion.library

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. 4-34: Interface description and default values of the MB_MBELT_MOVE data structure

4.5.15 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 206
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

ML_TechMotion.library

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. 4-35: Interface description and default values of the MB_MBELT_STATUS data structure

4.6 Position monitoring (PositionMonitoring)

4.6.1 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 220.: Monitoring of the relative position of two axes with regard to tilting.
2. [MB_PositionMonitoring_PosCollision](#), page 222.: Monitoring of the absolute and relative positions of two axes with regard to indentation and collision.

4.6.2 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.

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
IndraMotion MLC 13VRS	ML_TechMotion

Tab. 4-36: Reference table of the MB_PositionMonitoring_GantryCant

Interface description

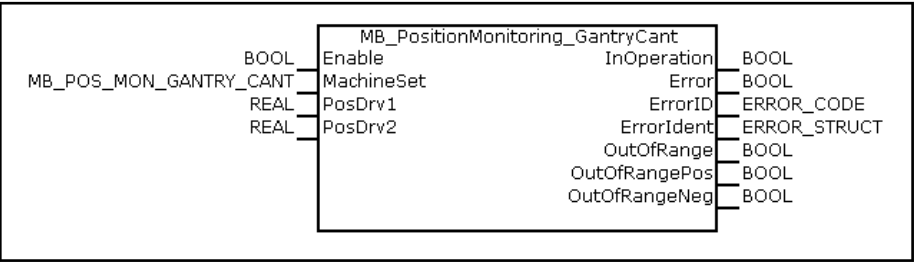


Fig. 4-38: MB_PositionMonitoring_GantryCant function block

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 222
	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 Troubleshooting , page 221
	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. 4-37: 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. 4-38: Error codes of the MB_PositionMonitoring_GantryCant function block

ML_TechMotion.library

4.6.3MB_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	$\geq 2 * \text{Tolerance}$	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. 4-39: Data structure of the MB_POS_MON_GANTRY_CANT

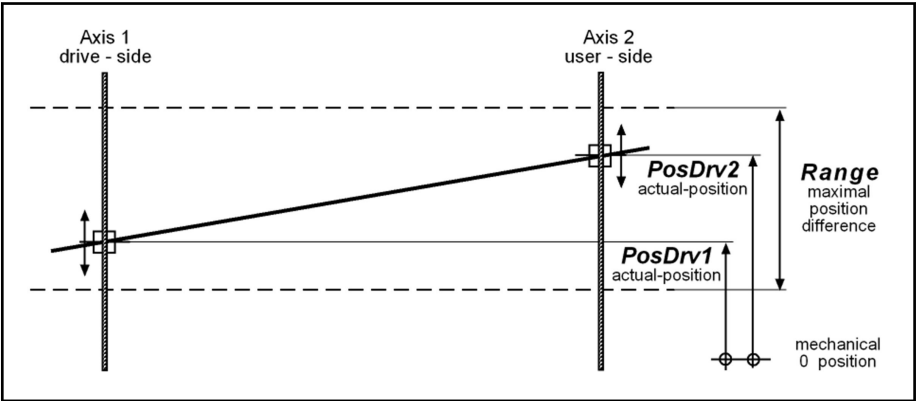


Fig. 4-39: Printing unit top view – Maximum position difference allowed (range)

The specified tolerance has to fulfill the following condition:

$$\text{Range} > 2 * \text{Tolerance}$$

Fig. 4-40: Condition machine set type 01

4.6.4MB_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.

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechMotion

Tab. 4-40: Reference table of the MB_PositionMonitoring_PosCollision

Interface description

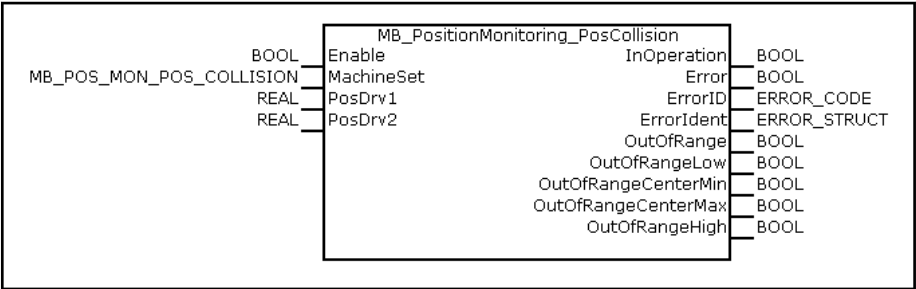


Fig. 4-41: MB_PositionMonitoring_PosCollision function block

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 224
	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 224
	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. 4-41: 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.

ML_TechMotion.library

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. 4-42: Error codes of the MB_PositionMonitoring_PosCollision function block

4.6.5 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	$\leq \text{DeltaCenter-Max} - 2 * \text{Tolerance}$	2.0	Minimum distance between the axes 1 and 2
DeltaCenter-Max	REAL	$\geq 2 * \text{Tolerance}$	n.def.	5.0	Maximum distance between the axes 1 and 2
DeltaHigh	REAL	$\geq 2 * \text{Tolerance}$	n.def.	5.0	Limitation of axis position 2
Tolerance	REAL	0.0	n.def.	1.0	Tolerance of the limits

Tab. 4-43: Data structure of the MB_POS_MON_POS_COLLISION

The specified tolerance has to fulfill the following condition:

$$\text{Parameters} > 2 * \text{Tolerance}$$

Fig. 4-42: Conditions for the machine set PosCollision

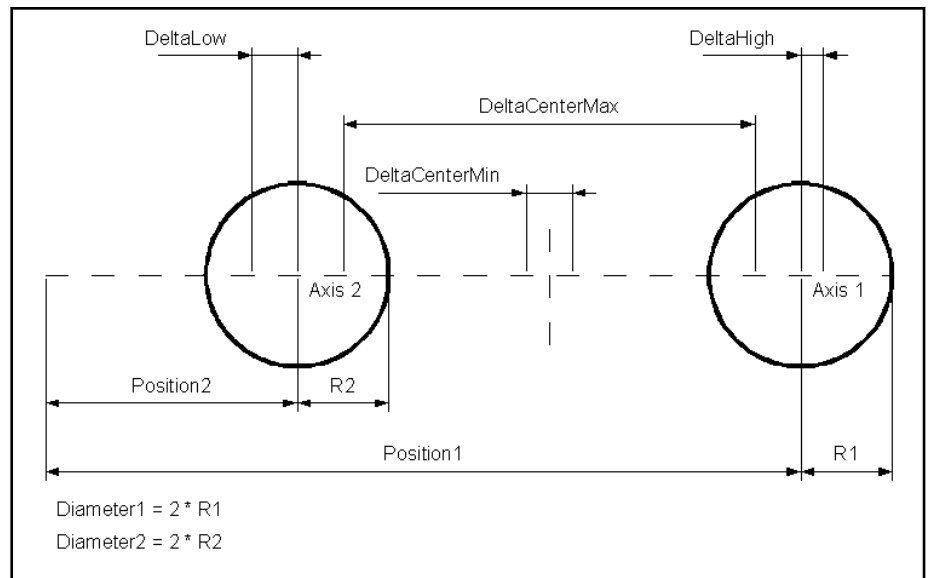


Fig. 4-43: Machine set of the MB_PositionMonitoring_PosCollision function block

4.7 Sag controller (SagControl)

4.7.1 Introduction and Overview

Sag controls are used to decouple system components.

The following belt system is a typical use case.

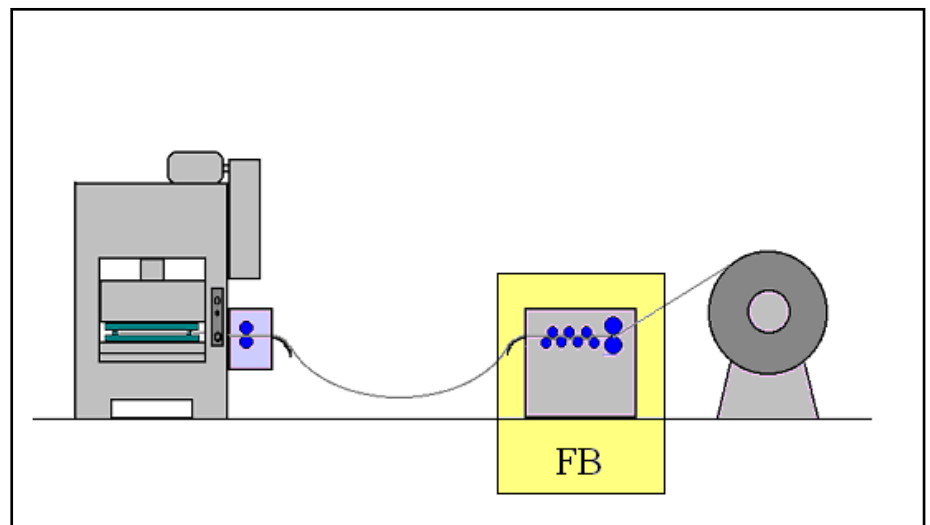


Fig. 4-44: Belt system, concept sketch

The sag is located between the punching machine feed rolls and the upstream 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:

ML_TechMotion.library

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 226, controls the sag with min./max. contacts.
- [ML_SagControlA](#), page 229, controls the sag with analog switch signals.

4.7.2 ML_SagControlC

Brief description The "ML_SagControlC" function block contains a sag controller with binary min./max. contacts.

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechMotion

Tab. 4-44: Reference table of the ML_SagControlA

Interface description

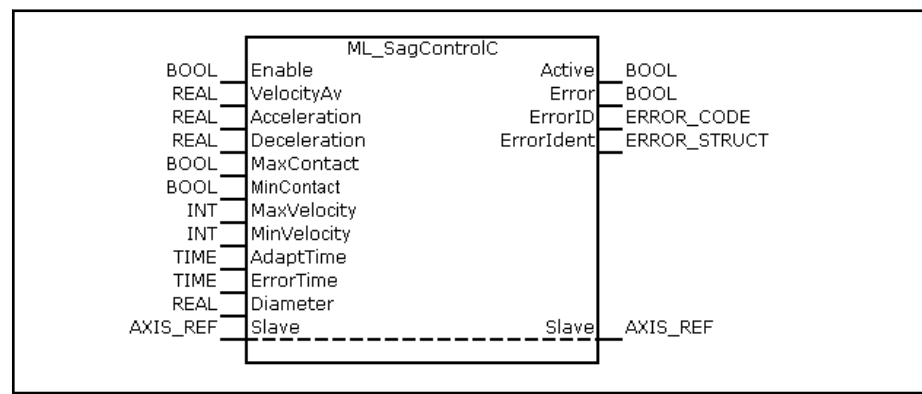


Fig. 4-45: ML_SagControlC function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Slave	AXIS_REF	Slave axis
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

I/O type	Name	Data type	Description
	ErrorID	ERROR_CODE	INT (Enum): Brief error description
	ErrorIdent	ERROR_STRUCT	Detailed error description depending on the error table Troubleshooting, page 227

Tab. 4-45: 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 "Te", the greater velocity is used for operation for the rest of the adaptation time "Ta" 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 "Te", the drive continues to operate at the reduced velocity for the remaining time "Ta". Afterwards, the operation continues at the specified average velocity (see v-t diagram).

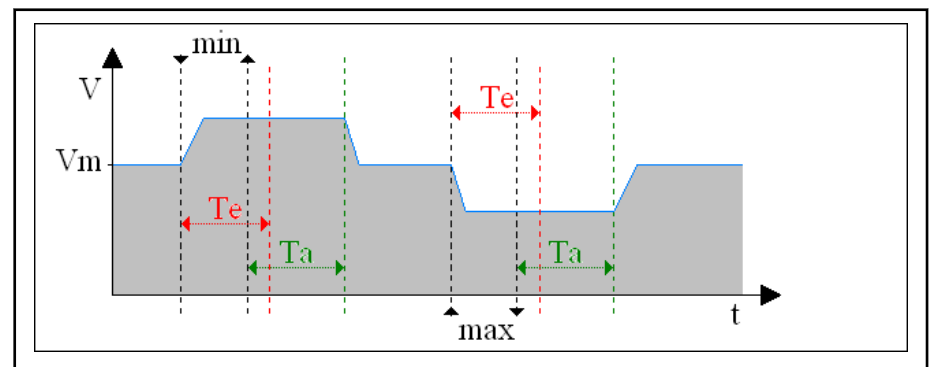


Fig. 4-46: Correct function block behavior

Troubleshooting

If the minimum contact is pending for a period longer than "Te", 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 "Ta" and then at the average velocity specified. However, if the maximum contact is pending for the period "Te", velocity is reduced to zero. If the maximum contact switches again, the drive remains stopped for the period "Ta" and then accelerates again to the specified average velocity "Vm" and continues to operate at this velocity.

ML_TechMotion.library

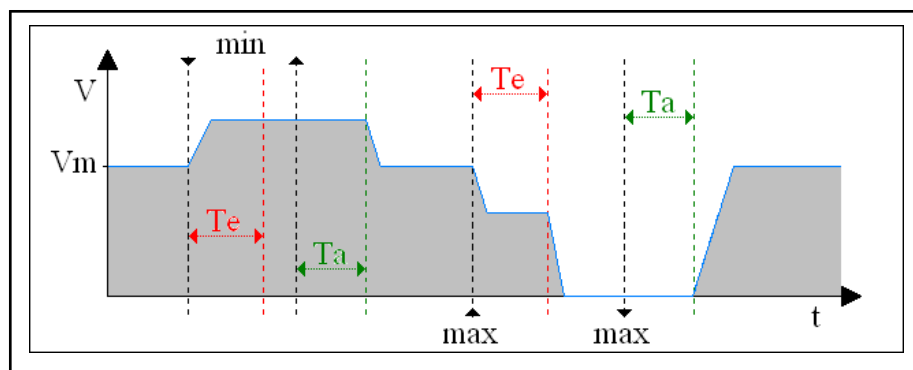


Fig. 4-47: Incorrect function block behavior

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#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. 4-46: 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 13VRS

Application example 1

In the simplest form 2 contacts, 1 minimum and 1 maximum contact are available as closed-loop control parameters.

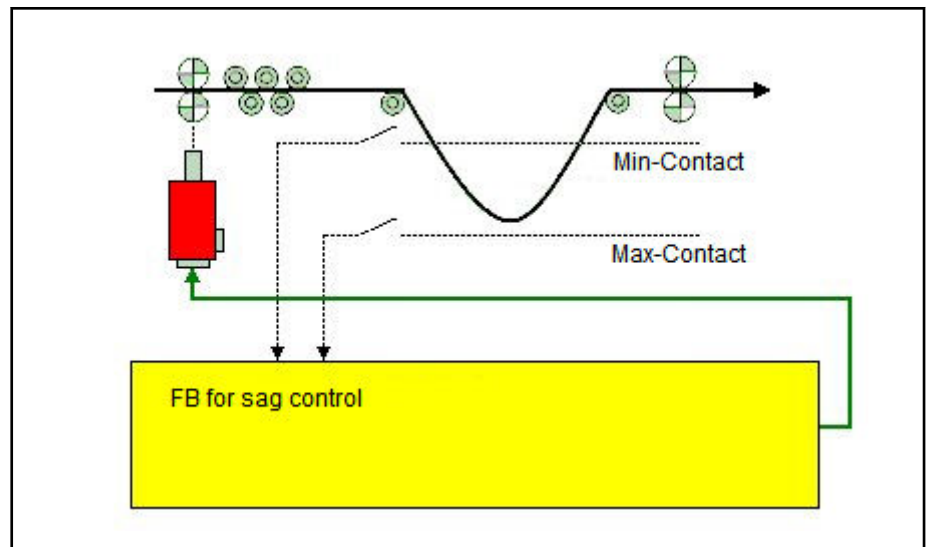


Fig. 4-48: Application 1, ML_SagControlC

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 added apart from the min./max. contacts.

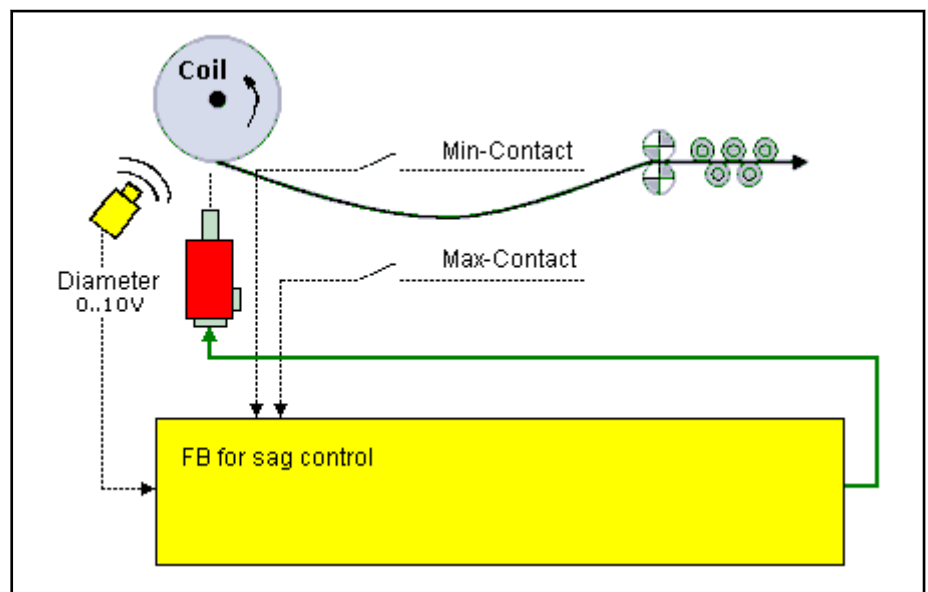


Fig. 4-49: Application 2, ML_SagControlC

4.7.3 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	Library
IndraMotion MLC 13VRS	ML_TechMotion

Tab. 4-47: Reference table of the ML_SagControlA

ML_TechMotion.library

Interface description

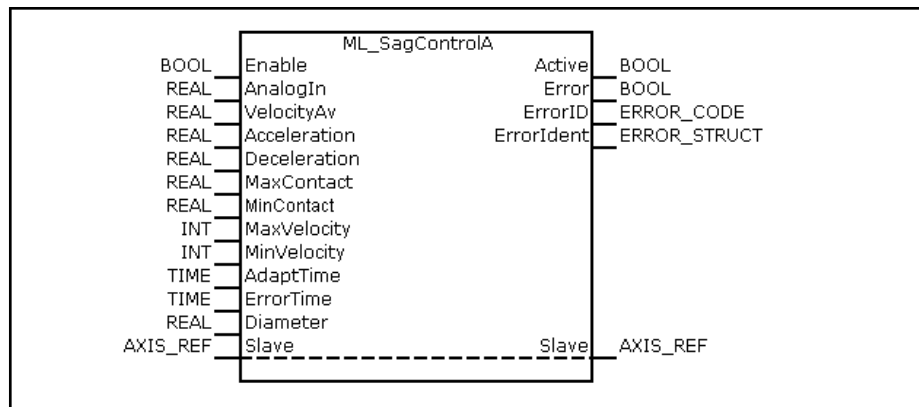


Fig. 4-50: ML_SagControlA function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Slave	AXIS_REF	Slave axis
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)
	CoilDiameter	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 Troubleshooting, page 231

Tab. 4-48: 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 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).

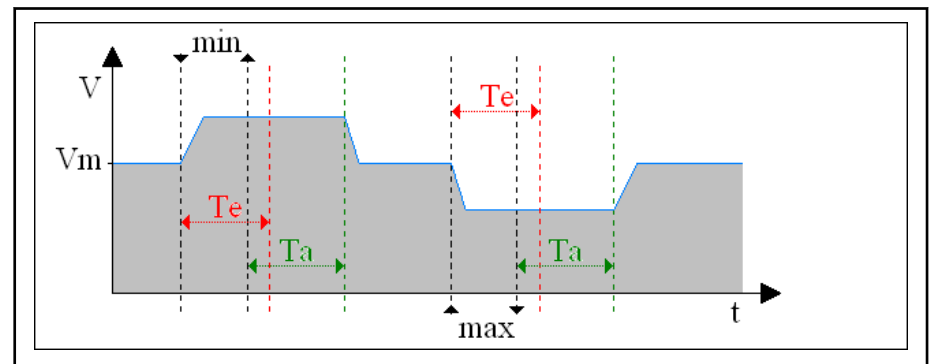


Fig. 4-51: Correct function block behavior

Troubleshooting

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.

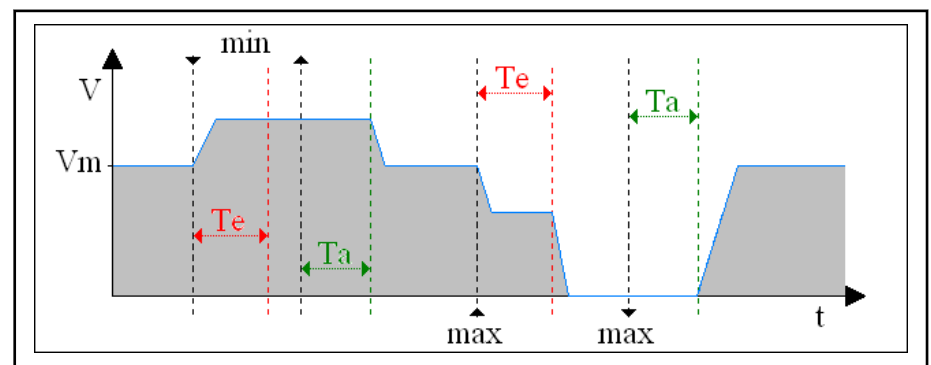


Fig. 4-52: Incorrect function block behavior

The function block generates the following error messages in Additional1/Additional2 for the table **F_RELATED_TABLE**, 16#0170:

ML_TechMotion.library

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. 4-49: 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 13VRS

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.

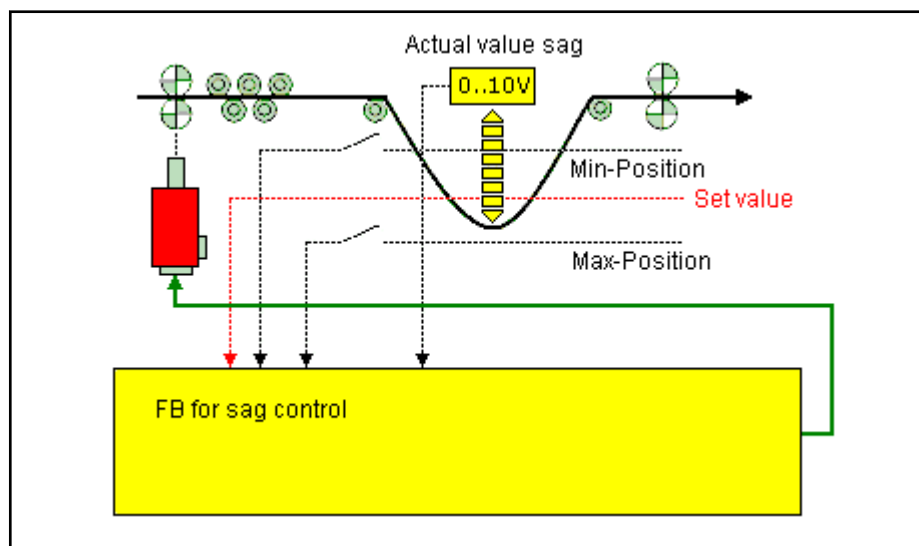


Fig. 4-53: Application 1, ML_SagControlA

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 added apart from the analog input value.

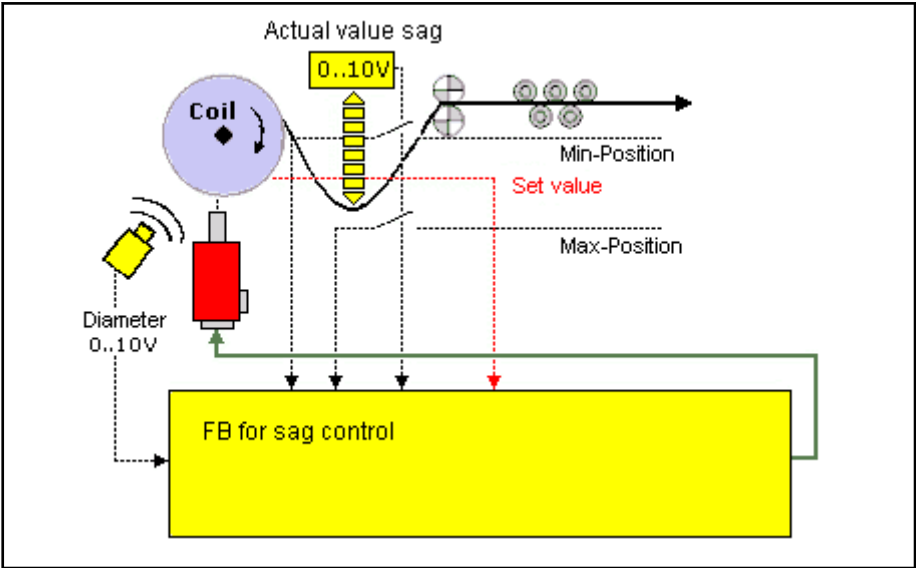


Fig. 4-54: Application 2, ML_SagControlA

4.8 S curve, programming motion sequences (SCurve)

4.8.1 Introduction and overview

The function block [MB_ReachVelocityType02](#), [page 233](#), 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.

4.8.2 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.

Assignment: Target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechMotion

Tab. 4-50: Reference table of the MB_ReachVelocityType02

ML_TechMotion.library

Interface description

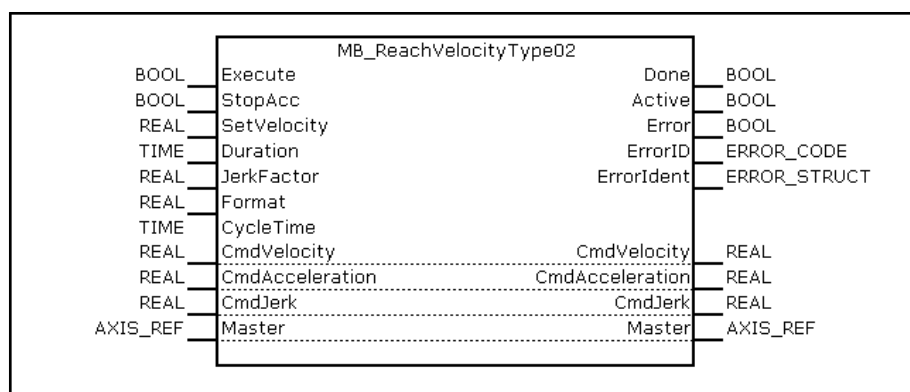


Fig. 4-55: MB_ReachVelocityType02 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	CmdVelocity	REAL	Command velocity [rpm]
	CmdAcceleration	REAL	Command acceleration [rad/s ²]
	CmdJerk	REAL	Command jerk [rad/s ³]
	Master	AXIS_REF	Reference to the master axis It is recommended to create a virtual master axis
VAR_INPUT	Execute	BOOL	Processing of function block enabled
	StopAcc	BOOL	It stops further acceleration
	SetVelocity	REAL	Target velocity [m/min]
	Duration	TIME	Acceleration period
	CycleTime	TIME	Cycle time of the task in which the function block is called
	JerkFactor	REAL	Ramping the acceleration profile [%]
	Format	REAL	Format [mm]
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 Error handling, page 237

Tab. 4-51: 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

Name	Type	Min. value	Max. value	Default value	Effective
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. 4-52: 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 velocity period (Duration), the path 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 (VAR_IN_OUT) has to be a virtual axis to ensure the desired velocity. 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:

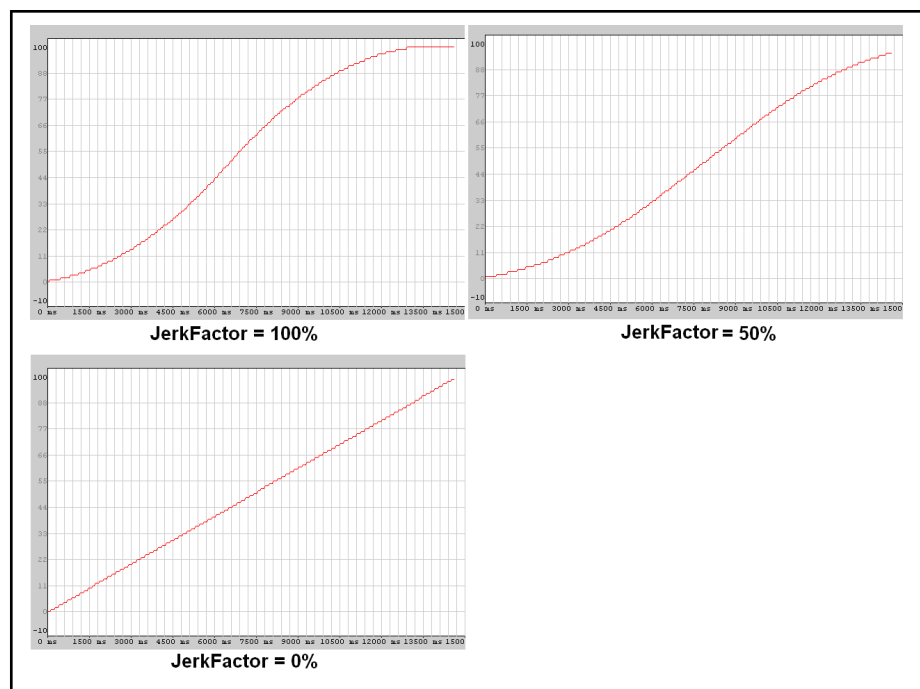


Fig. 4-56: Acceleration phases with different JerkFactor

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.

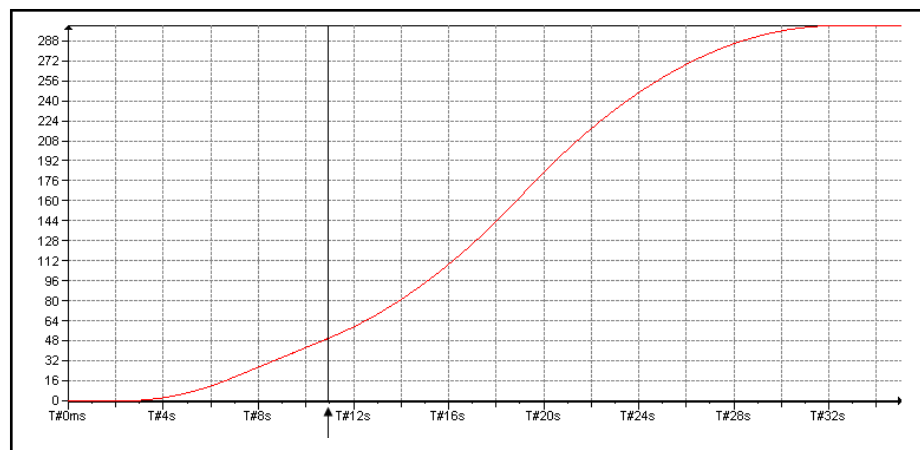


Fig. 4-57: Effects of target velocity changes

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":

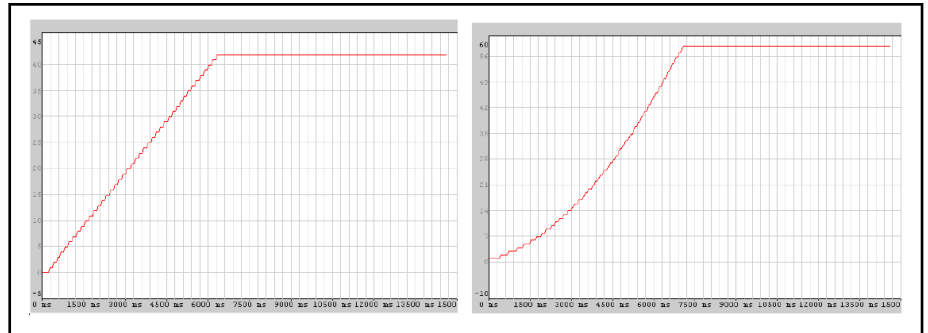


Fig. 4-58: Aborting the acceleration phases (left an abort at 6300 ms, right an abort at 7000 ms)

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
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. 4-53: Error codes of MB_ReachVelocityType02

4.9 SmartBelt [feed machine (SmartBelt)]

4.9.1 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	Library
IndraMotion MLC 13VRS	ML_TechMotion
IndraMotion MLD with MPx07/08/17 with MA function package	MX_Packaging_07.lib

Tab. 4-54: Reference table of the MB_SmartBeltType01

ML_TechMotion.library

Interface description

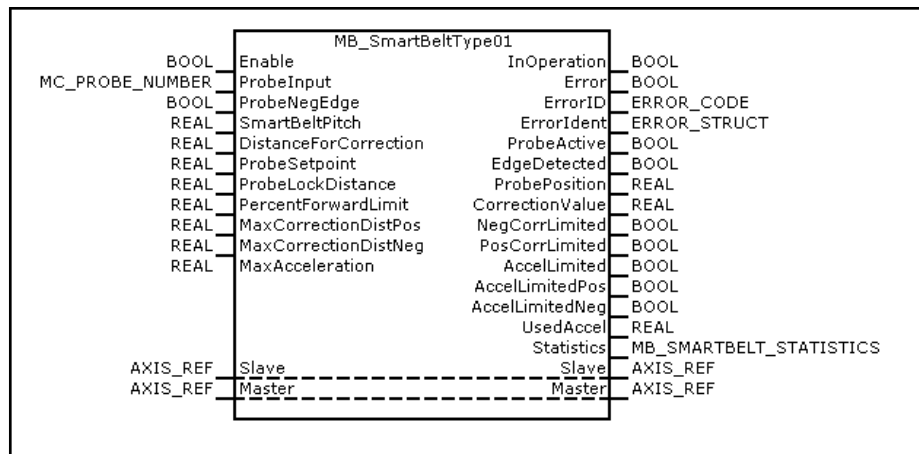


Fig. 4-59: MB_SmartBeltType01 Function Block

I/O type	Name	Data type	Description
VAR_IN_OUT	Slave	AXIS_REF	Address of SmartBelt axis
	Master	AXIS_REF	Address of master axis
VAR_INPUT	Enable	BOOL	Rising edge enables function block
	ProbeInput	MC_PROBE_NUMBER	Selection of touch probe input 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 (in or 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 the touch probe (in or mm). Read cyclically
	ProbeLockDistance	REAL	Distance within the touch probe input is ignored (in or 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 (in or mm). Read cyclically
	MaxCorrectionDistNeg	REAL	Maximum correction distance for negative correction (in or 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
	ErrorStruct	ERROR_STRUCT	Detailed error description Troubleshooting, page 242
	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 243

Tab. 4-55: Interface variables of the MB_SmartBeltType01 function block

ML_TechMotion.library

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. 4-56: Overview on min./max. values and initial values of the inputs

Functional Description

Detailed description of some input and output variables

The measuring units have to be equal for all entries (in or 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:

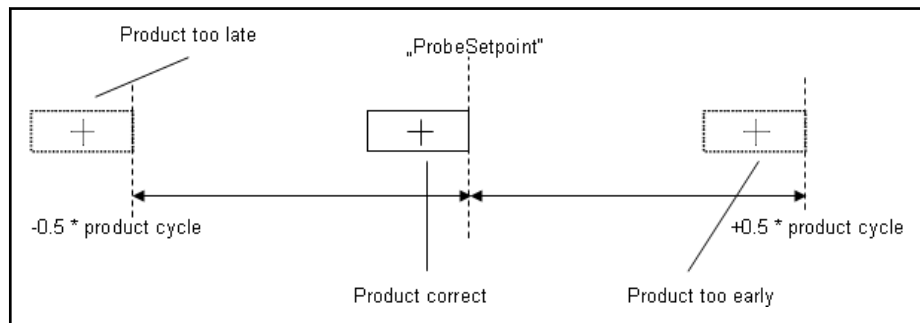


Fig. 4-60: Meaning of the correction in positive or negative direction

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:

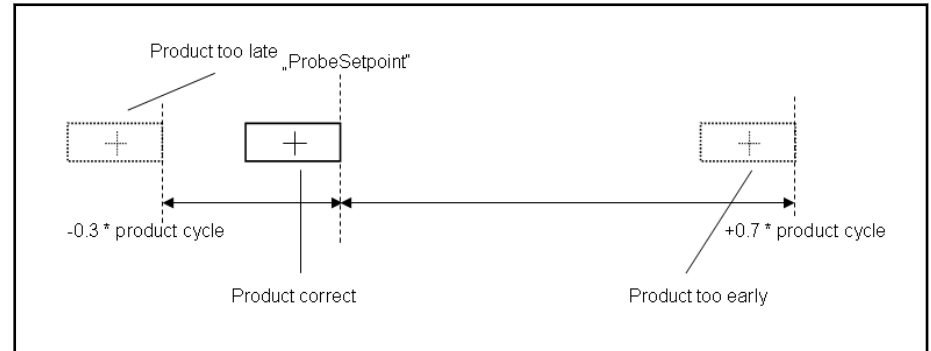


Fig. 4-61: PercentForwardLimit

"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:

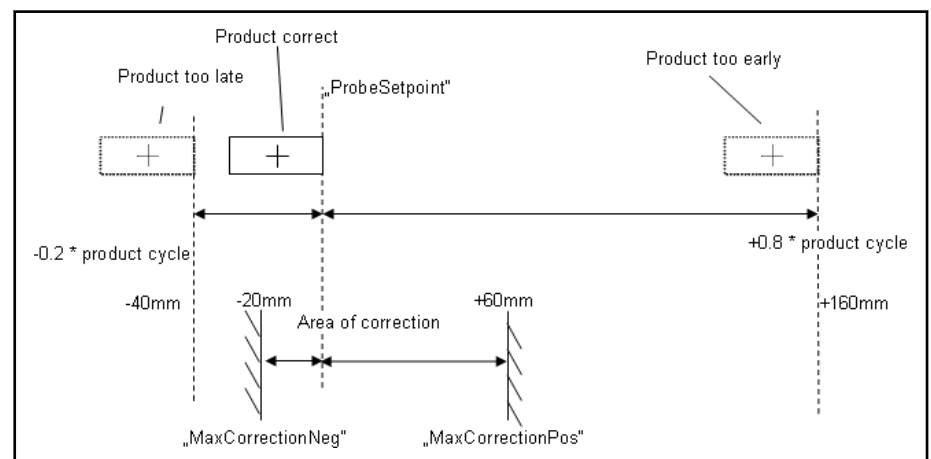


Fig. 4-62: MaxCorrectionDistancePos and MaxCorrectionDistanceNeg

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

ML_TechMotion.library

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.

Troubleshooting

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 a drive "x" indicates the address of the incorrect axis
ACCESS_ERROR (16#0004)	16#0003	16#0000	The internally used function block MB_TouchProbeContinuous 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")
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)

ErrorID	Additional1	Additional2	Description
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 (IndraMotion MLD)/A-0-2793 (IndraMotion MLC) has to be set to "Shortest Path"
INPUT_INVALID_ERROR	16#0F02	16#001D	Input "ProbeSetpoint" is out of limits (0..Master Modulo)
INPUT_INVALID_ERROR	16#0F02	16#001F	"ProbeLockDistance" input 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(Max-CorrectionDistNeg) is higher 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. 4-57: 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.

4.9.2 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

ML_TechMotion.library

Structure element	Type	Default value	Description
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. 4-58: 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.

4.9.3 Introduction to SmartBelt Usage

SmartBelt Usage

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).

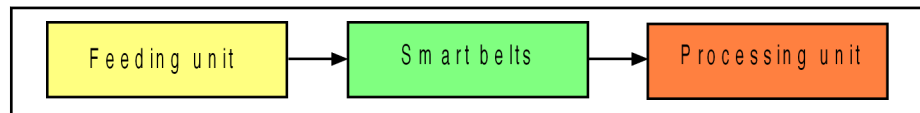


Fig. 4-63: Example a flow wrapper

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

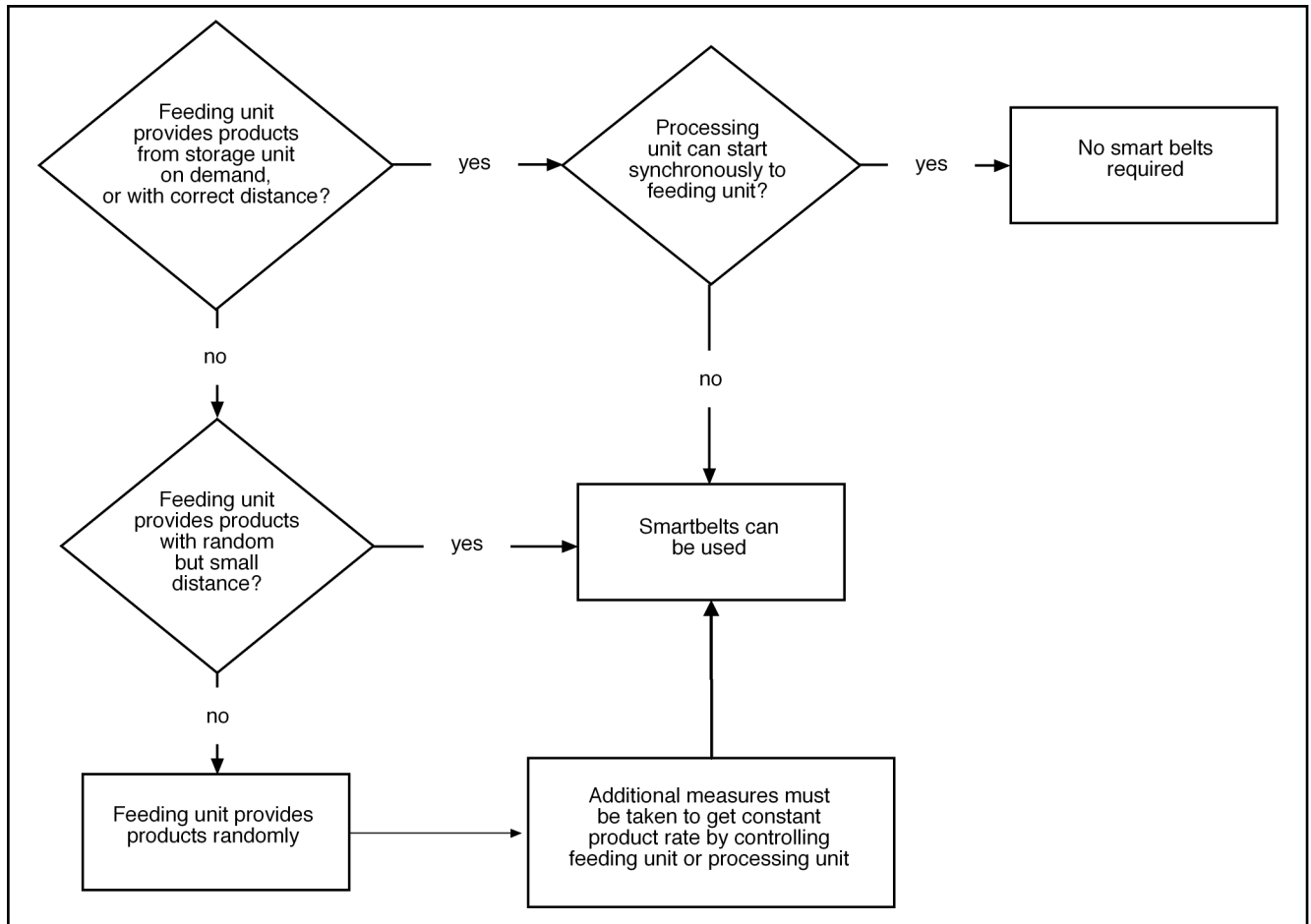


Fig. 4-64: Usage of SmartBelts

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.

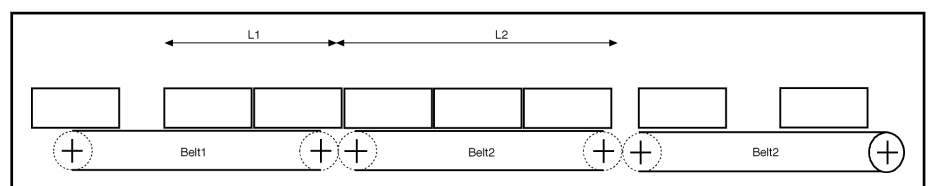


Fig. 4-65: Accumulating Conveyor

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:

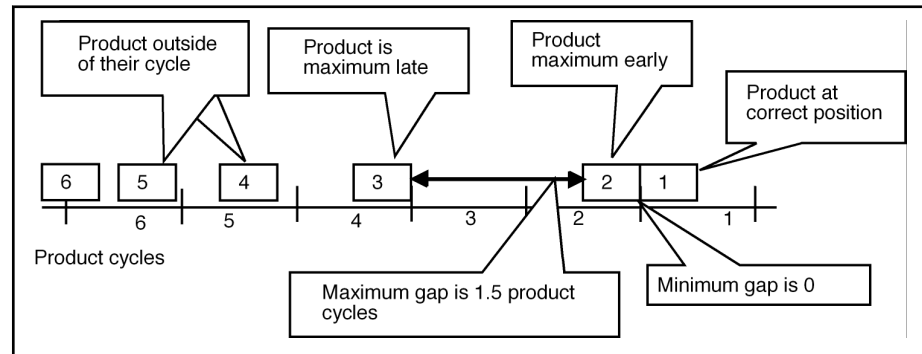


Fig. 4-66: Assessment using Product Cycles and Gaps

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:

$$\text{Correction} = \min \left(L_{gap max}, \left(1.5 * L_{cycl} - L_{Prod} + L_{gap} * (N_{gap} - 1) - (N_{gap} * L_{gap min}) \right) \right)$$

For this example, the following values apply:

Assessment using Product Rate and Duration

$$\begin{aligned}\text{Correc-} &= \min (135 \text{ mm}, (1.5 * 90 \text{ mm} - 40 \text{ mm} + 50 \text{ mm} * (2 - 1) - (2 * 0 \text{ mm}))) \\ \text{tion} &= \min (135 \text{ mm}, 145 \text{ mm}) \\ &= 135 \text{ mm}\end{aligned}$$

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.

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:

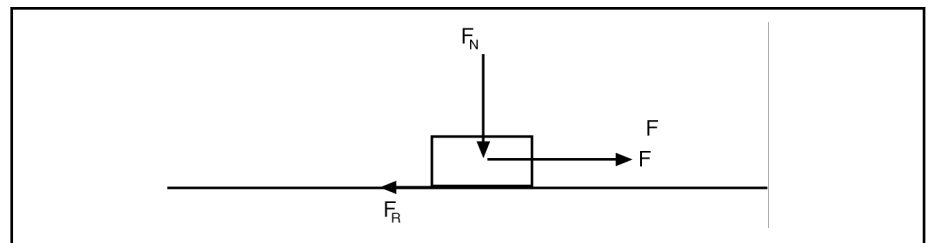


Fig. 4-67: Friction

Static friction is defined as:

$$F_R = \mu_H F_N$$

with

$$F_N = F_G = m * g$$

and substituting the later by the first one

$$F_R = \mu_H * m * g$$

The necessary force for acceleration is:

$$F = m * a$$

ML_TechMotion.library

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.

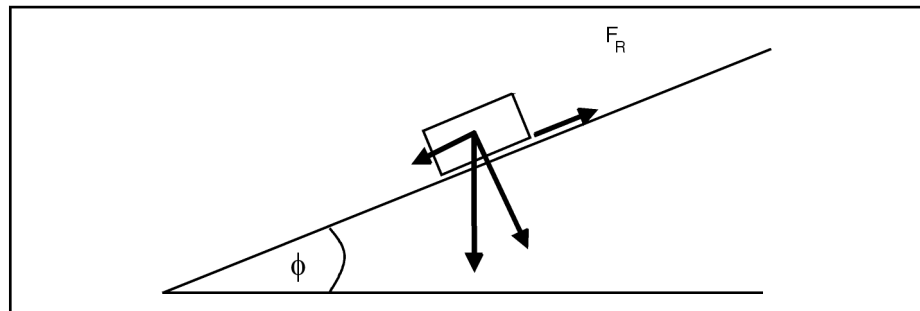


Fig. 4-68: Relation between friction coefficient and angle

4.9.4 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. 4-69 "Touch probe with detection of positive edge" on page 249](#)).

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. 4-70 "Touch probe with detection of negative edge" on page 249](#).

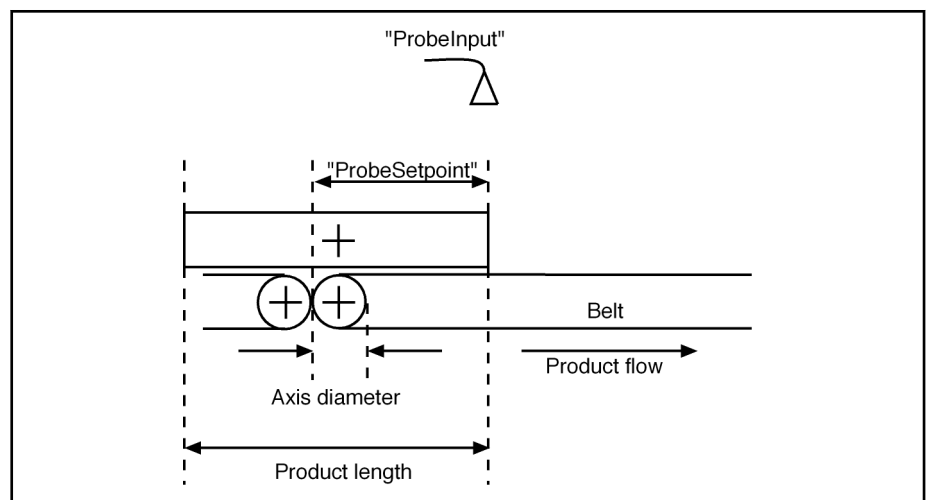


Fig. 4-69: Touch probe with detection of positive edge

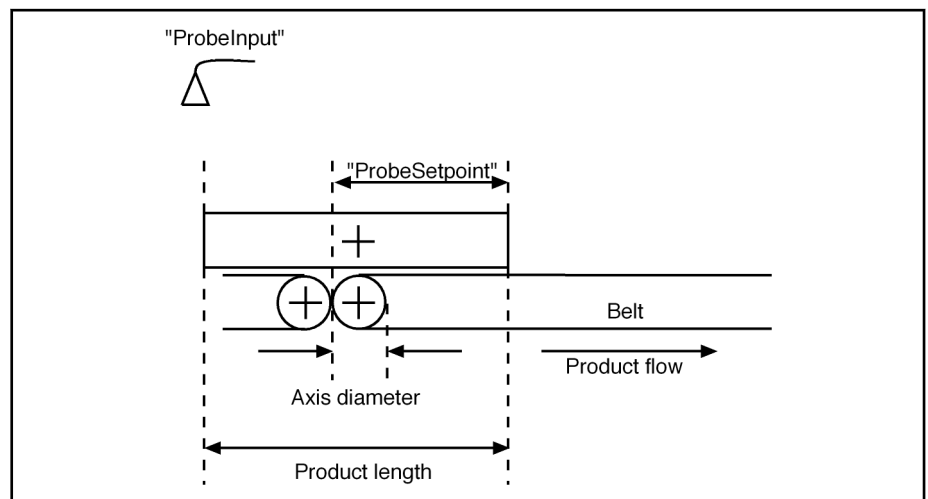


Fig. 4-70: Touch probe with detection of negative edge

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 IndraMotion 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 IndraMotion 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.

ML_TechMotion.library

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.

IndraMotion 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

IndraMotion 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 down-load 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

IndraMotion 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 as follows:

- "RatioNumerator"/"InputRevolution" should be set to 1
- "RatioDenominator"/"OutputRevolution" should be set to 3

IndraMotion MLD parameter settings

ML_TechMotion.library

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:

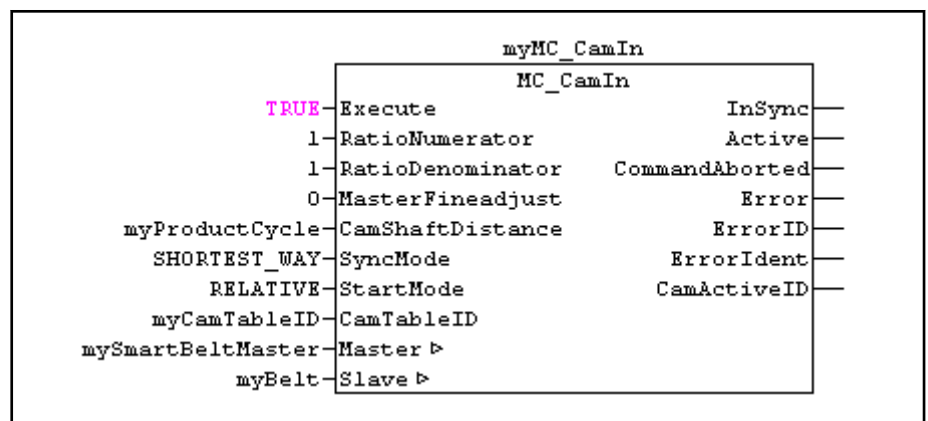


Fig. 4-71: MC_CamIn settings for use with SmartBelt

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 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.

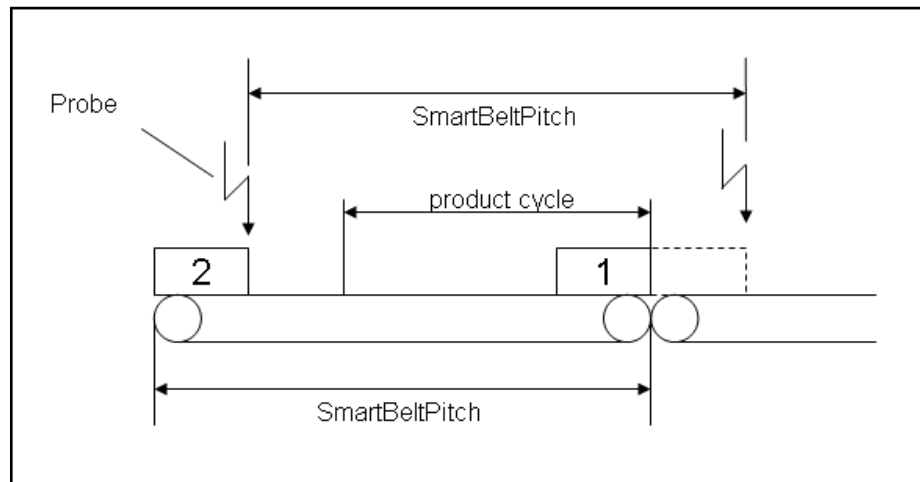


Fig. 4-72: SmartBelt with physical length approximately corresponds to the product cycle

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.

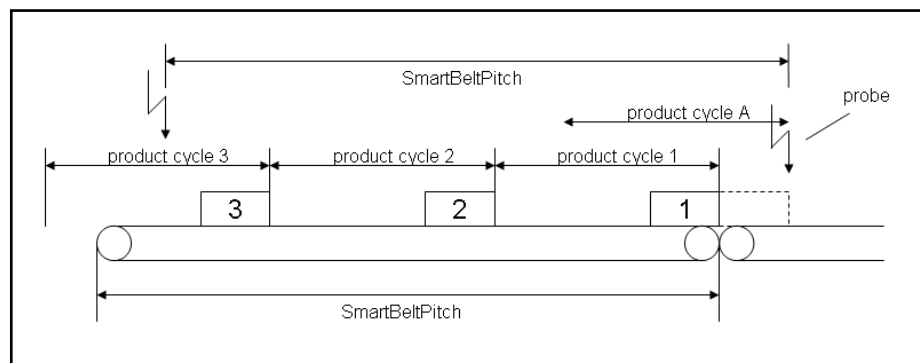


Fig. 4-73: SmartBelt with physical length corresponding to a multiple of the product cycle

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.

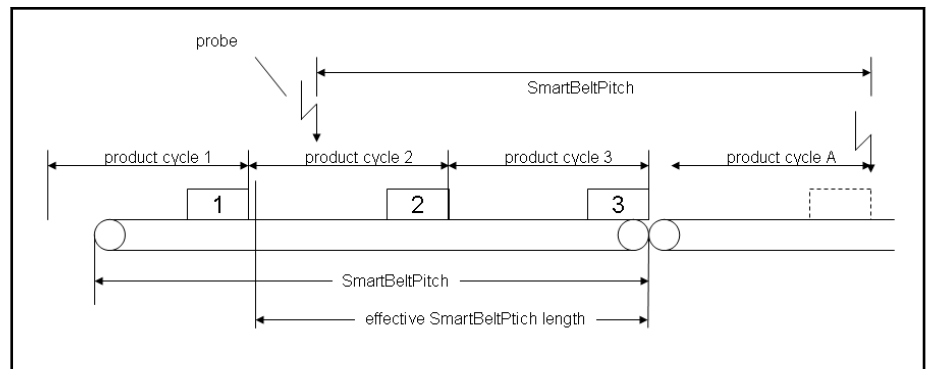


Fig. 4-74: SmartBelt with touch probe not mounted at the beginning of the belt

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:

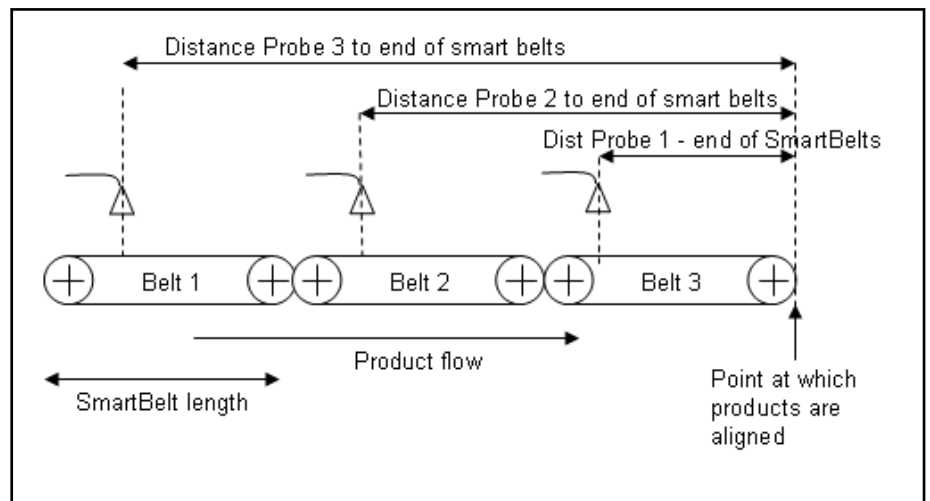


Fig. 4-75: Measurements to calculate "ProbeSetpoint"

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:

ML_TechMotion.library

$$\text{ProbeSetpoint}_k = L_{\text{cycl}} - \left(\left(L_{\text{SmartBelt}_n} - L_{\text{ProbePos}_n} \right) + \sum_{n=1}^{k-1} L_{\text{SmartBelt}_n} \right) \text{MOD} (L_{\text{cycl}})$$

Fig. 4-76: 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. 4-77: 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: $k=3$; $\text{ProbeSetpoint}_3 = 120 - (((150-120) + 300) \text{MOD} (120)) = 120 - (330 \text{MOD} 120) = 30$

4.9.5 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.

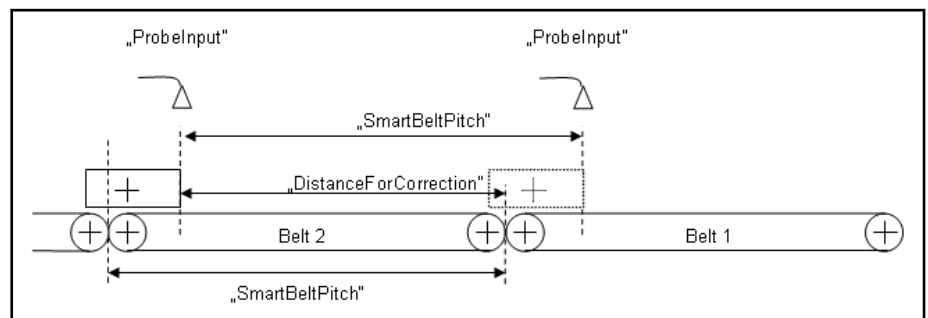


Fig. 4-78: Two or More Products on One Belt

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 [chapter "PLC Programming" on page 252](#) for details.



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!

4.9.6 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.

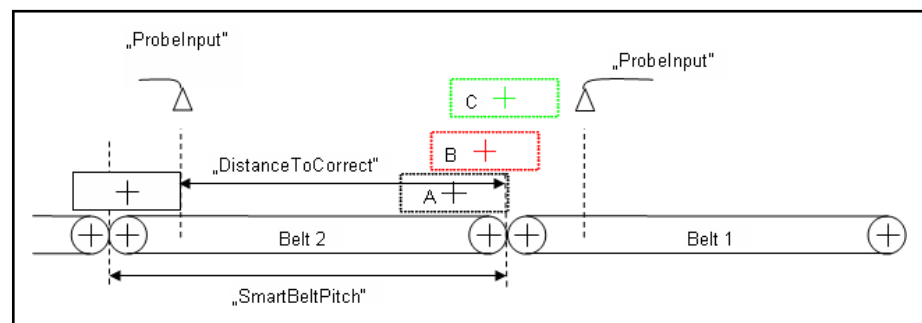


Fig. 4-79: Different Distances for Correction

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 relationship between possible acceleration and friction is given in [chapter "Friction" on page 247](#) 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.

4.9.7 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 (IndraMotion MLC) or P-0-0142 and P-0-0143 (IndraMotion MLD)
Error F2002028 (IndraMotion MLC) or F2028 (IndraMotion MLD)"Excessive control deviation" occurs permanently.	Set the parameters A-0-0032, A-0-0033 and A-0-0034 (IndraMotion MLC) or S-0-0038, S-0-0039 and S-0-0138 (IndraMotion 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

ML_TechMotion.library

Question	Answer:
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 SmartBelt 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

4.10 Cam tables for technology functions (TechFuncCamTable)

4.10.1 Introduction and overview

The function blocks create cam tables.

[MB_PilgrimStepCalcType01 on page 260](#): 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 on page 271](#): 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).

4.10.2 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	IndraMotion MLC/IndraLogic XLC 13VRS	ML_TechMotion

Tab. 4-59: Reference table of the MB_PilgrimStepCalcType01

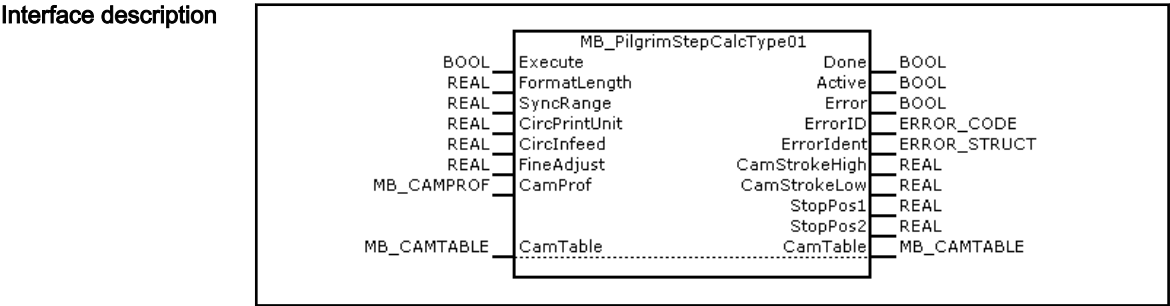


Fig. 4-80: MB_PilgrimStepCalcType01 function block

I/O type	Name	Type	Comment
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements MB_CAMTABLE, page 46
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_CAMPF	Cam profile selection MB_CAMPF, page 45
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 270)
	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

Tab. 4-60: 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"

ML_TechMotion.library

Name	Type	Min. value	Max. value	Default value	Effective
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. 4-61: Min./max. and default values of the MB_PilgrimStepCalcType01 function block

The enumeration MB_CAMPROF, ML_TechBase 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.

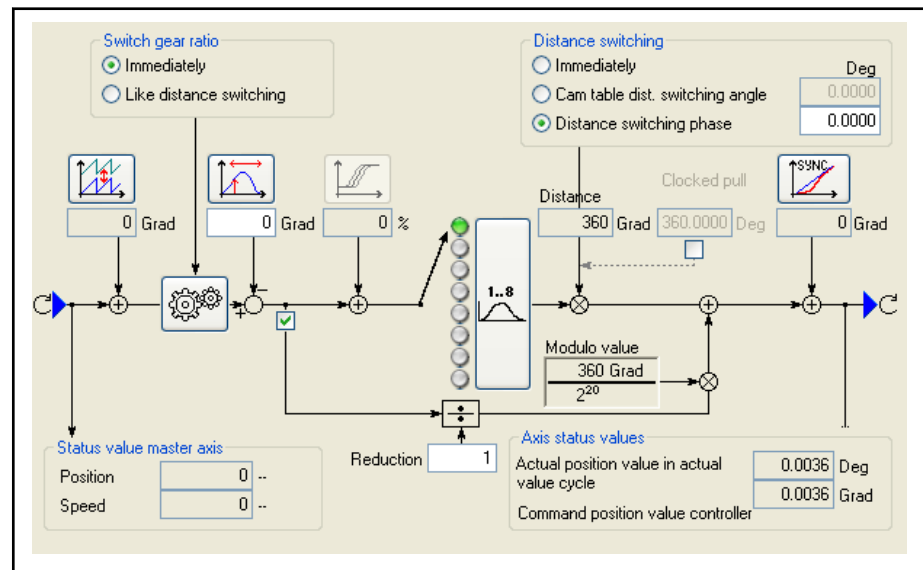


Fig. 4-81: Enabling the linear path

The structure MB_CAMTABLE, ML_TechBase, 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

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.

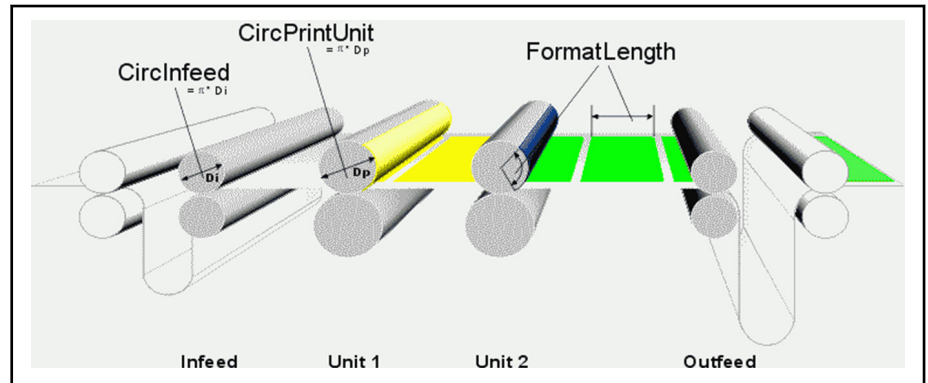


Fig. 4-82: Schematic diagram of the step-back

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 function 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. 4-83: 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. 4-84: *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 or drive using the MB_LoadCamData, ML_TechBase 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.

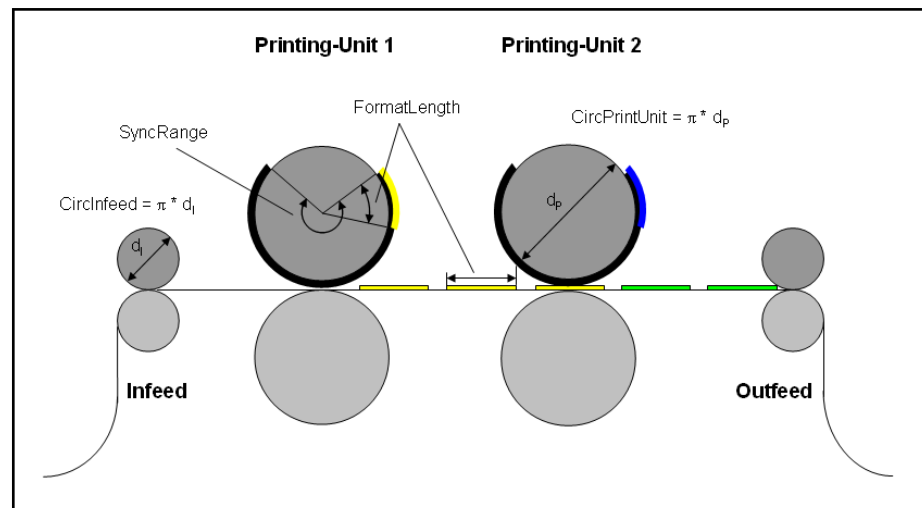


Fig. 4-85: *Process diagram (FormatLength < SyncRange)*

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.

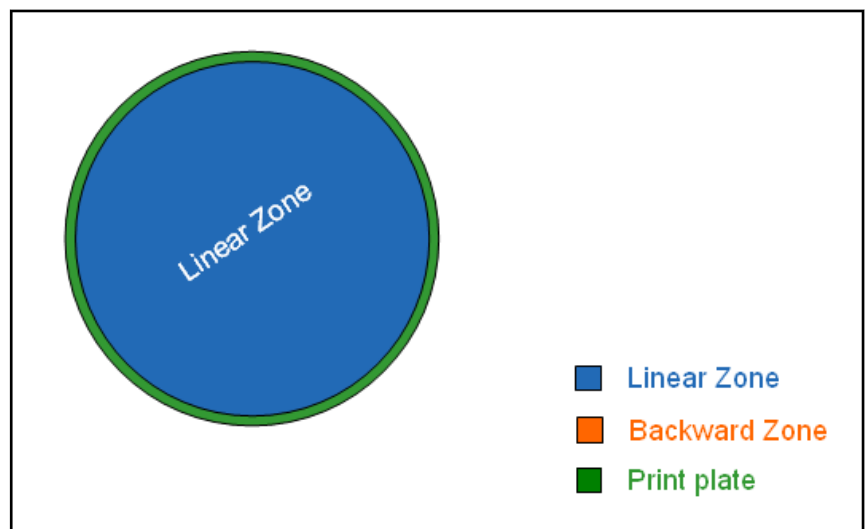
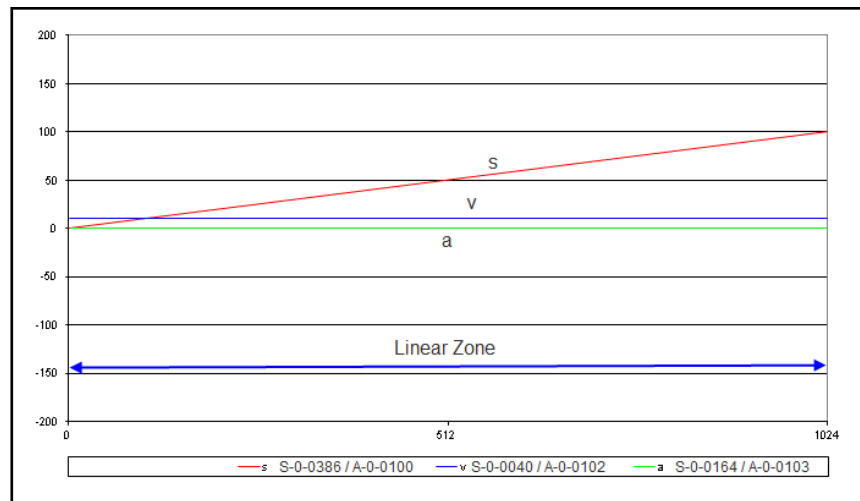


Fig. 4-86: MB_PilgrimStepCalcType01 with FormatLength = SyncRange = CircPrintUnit

ML_TechMotion.library



FormatLength = 2000.0; SyncRange = 2000.0; CircPrintUnit = 2000.0; CircInFeed = 2000.0

Fig. 4-87: 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.

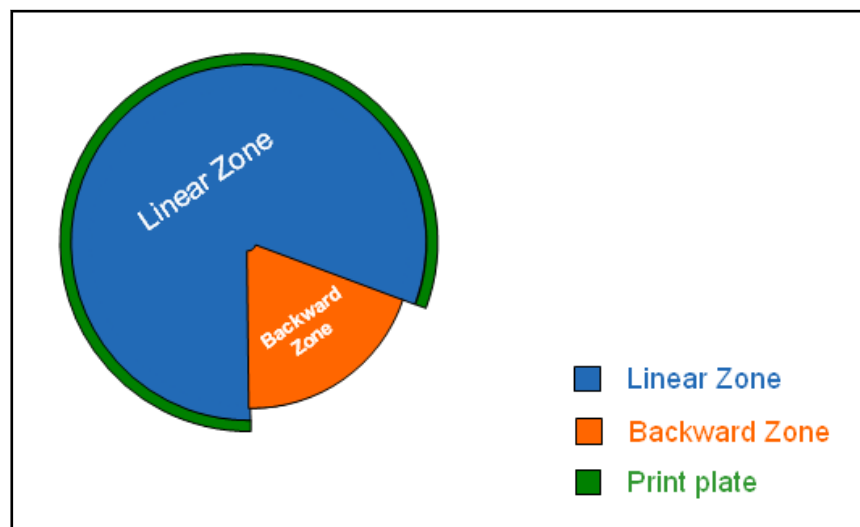


Fig. 4-88: MB_PilgrimStepCalcType01 with FormatLength = SyncRange

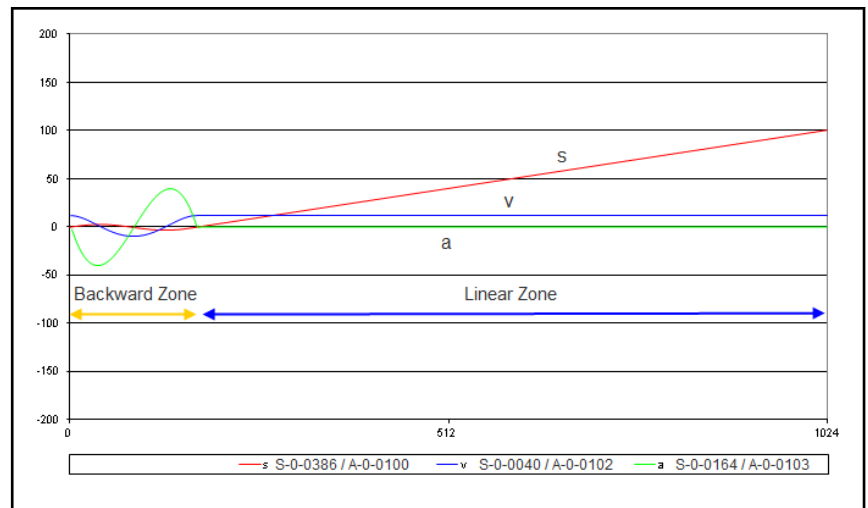


Fig. 4-89: MB_PilgrimStepCalcType01 MotionProfile

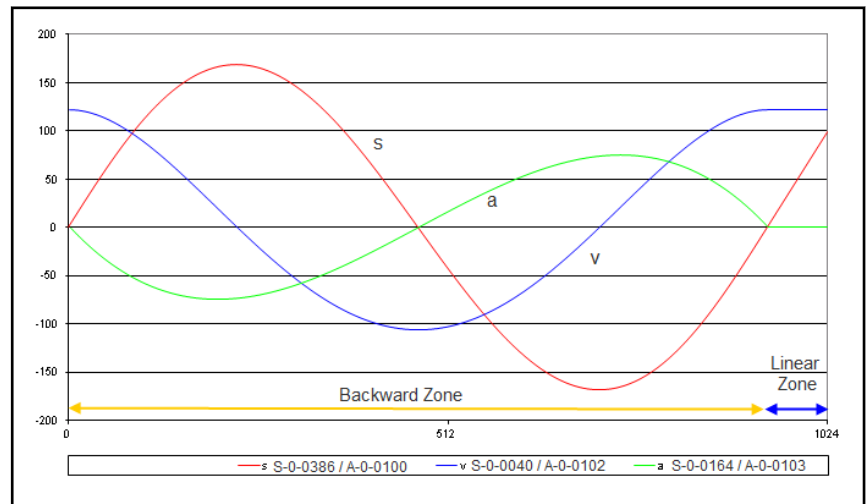


Fig. 4-90: MB_PilgrimStepCalcType01 MotionProfile

- FormatLength < SyncRange**

The printing plate or printing block only covers partially 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.

ML_TechMotion.library

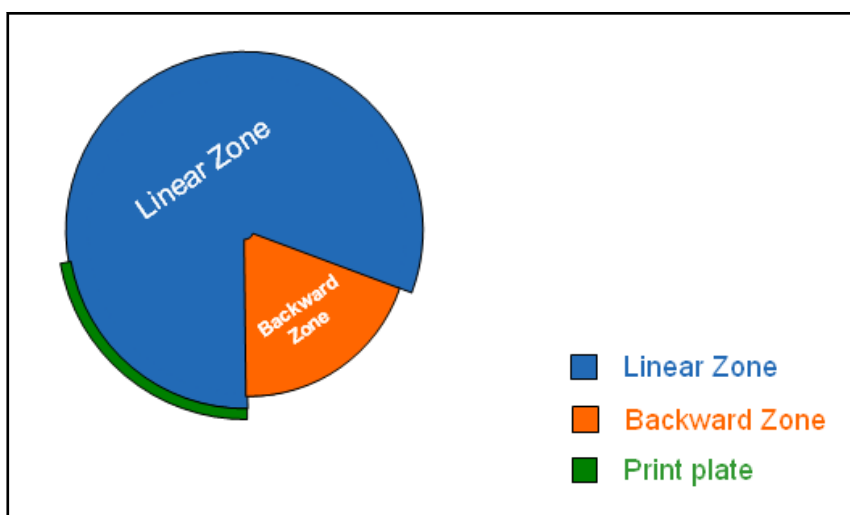
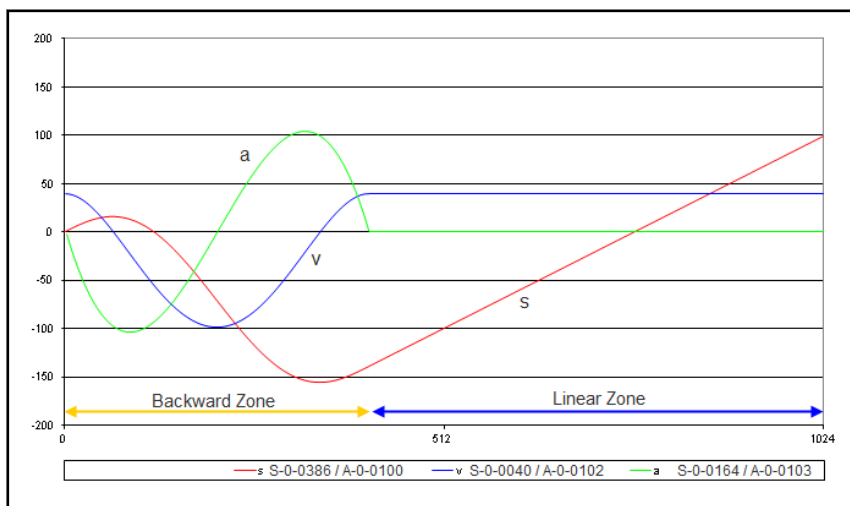
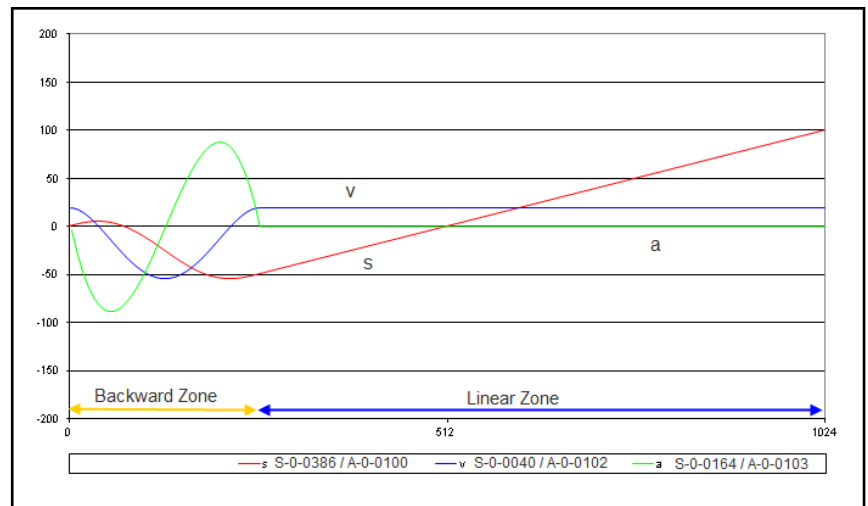


Fig. 4-91: MB_PilgrimStepCalcType01 with FormatLength < SyncRange



FormatLength = 500.0; SyncRange = 1200.0; CircPrintUnit = 2000.0; CircInFeed = 3000.0

Fig. 4-92: MB_PilgrimStepCalcType01 MotionProfile

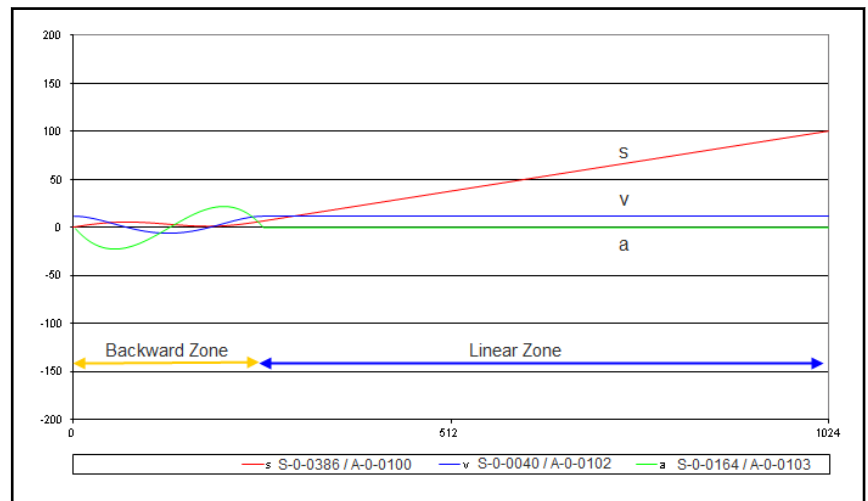


FormatLength = 1000.0; SyncRange = 1500.0; CircPrintUnit = 2000.0; CircInFeed = 3000.0

Fig. 4-93: 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 = ; CircInFeed =

Fig. 4-94: MB_PilgrimStepCalcType01 MotionProfile

Use case special product printed with two printing units:



Fig. 4-95: Standard product (format length 350 mm):

ML_TechMotion.library



Fig. 4-96: Special product (format length 700 mm):

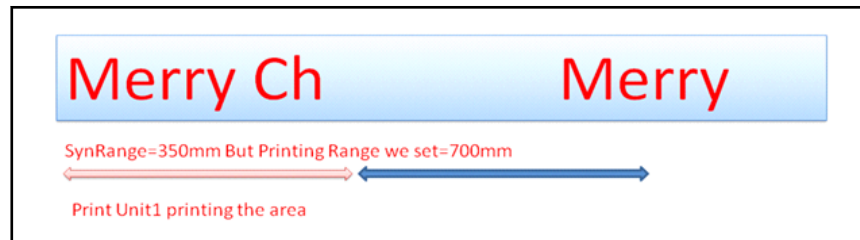


Fig. 4-97: Printing unit 1 for printing the red area



Fig. 4-98: Printing unit 2 for printing the blue area

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

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR	16#00002200	16#00000001	"NumberOfElements" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#00002260	16#00000001	"FormatLength" input is < minimum
INPUT_RANGE_ERROR	16#00002260	16#00000002	"SyncRange" input is < minimum
INPUT_RANGE_ERROR	16#00002260	16#00000003	"CircPrintUnit" input is < minimum
INPUT_RANGE_ERROR	16#00002260	16#00000004	"CircInfeed" input is < minimum
INPUT_RANGE_ERROR	16#00002260	16#00000005	"FineAdjust" input is < minimum or > maximum
INPUT_INVALID_ERROR	16#00002261	16#00000001	Address of the "CamTable" array is zero
INPUT_INVALID_ERROR	16#00002261	16#00000001	"FormatLength" input is < minimum
INPUT_INVALID_ERROR	16#00002261	16#00000002	"SyncRange" input is < minimum
INPUT_INVALID_ERROR	16#00002261	16#00000003	Input "FormatLength" is > (higher than) SyncRange
INPUT_INVALID_ERROR	16#00002261	16#00000004	SyncRange = CircPrintUnit but FormatLength != SyncRange (no space for compensation motions)

Tab. 4-62: Error codes of the MB_PilgrimStepCalcType01 function block

4.10.3 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

IndraMotion MLC 13VRS	ML_TechMotion

Tab. 4-63: Reference table of the MB_PrintLengthCorrCalcType01

Interface description

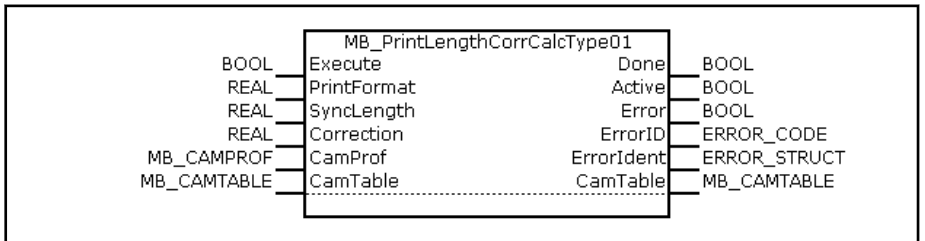


Fig. 4-99: MB_PrintLengthCorrCalcType01 function block

I/O type	Name	Type	Comment
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements MB_CAMTABLE, page 46
VAR_INPUT	Execute	BOOL	Function block processing 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 45
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 275)

Tab. 4-64: 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"

ML_TechMotion.library

Name	Type	Min. value	Max. value	Default value	Effective
Correction	REAL	-9999.9	9999.9	100.0	Rising edge at "Execute"
CamProf	MB_CAMPROF			CAM_PROF_STANDARD	Rising edge at "Execute"

Tab. 4-65: Min./max. and default values of the MB_PrintLengthCorrCalcType01

The enumeration MB_CAMPROF, ML_TechBase 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.

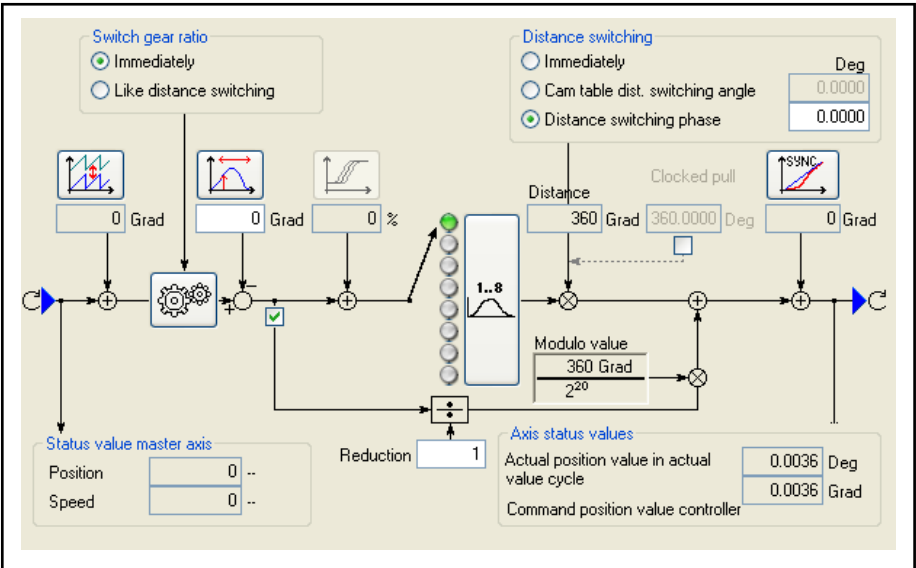


Fig. 4-100: Enabling the linear path

The structure MB_CAMTABLE, ML_TechBase, 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

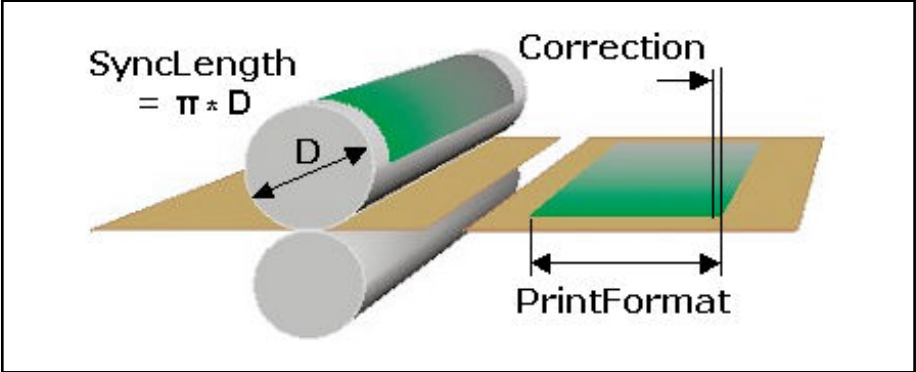


Fig. 4-101: Schematic diagram of the printing length correction

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.

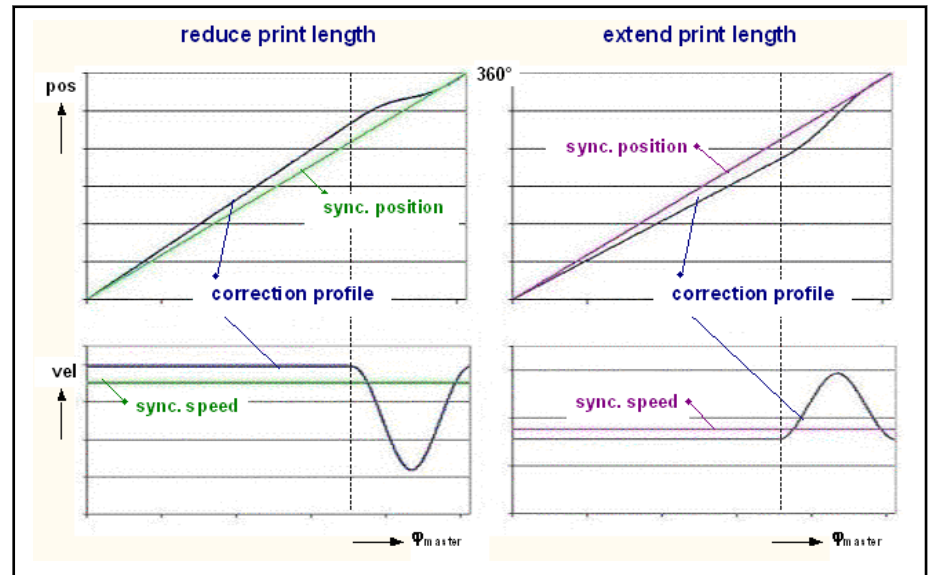


Fig. 4-102: Profile of the printing length correction

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_TechBase 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 ve-

ML_TechMotion.library

locity 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

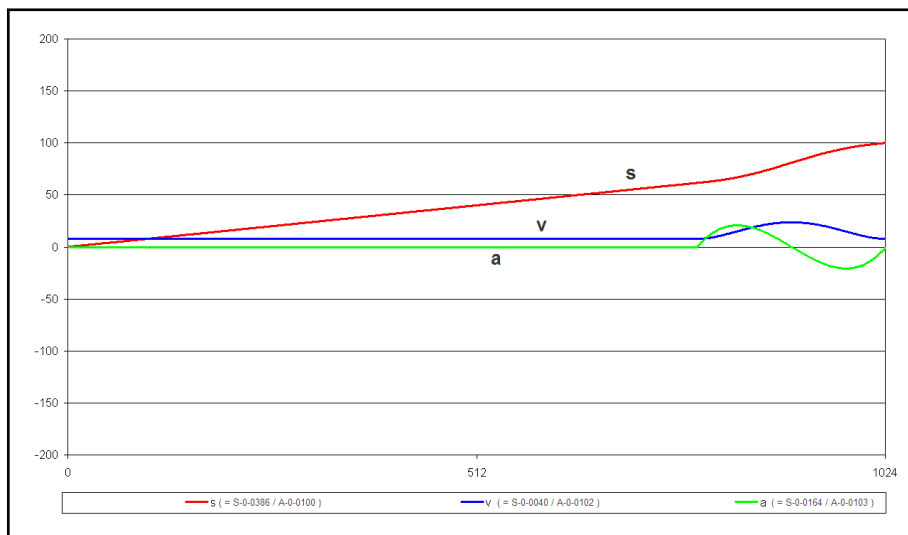


Fig. 4-103: MB_PrintLengthCorrCalcType01 MotionProfile

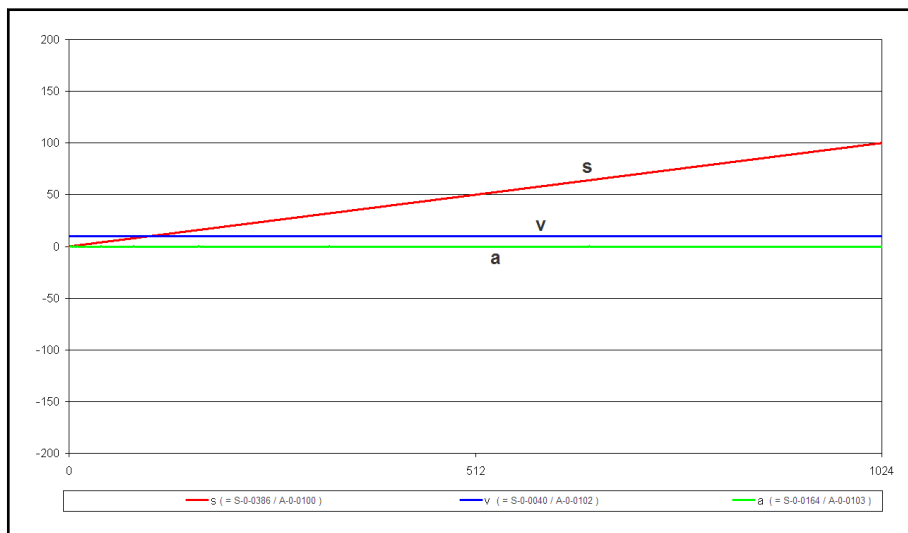
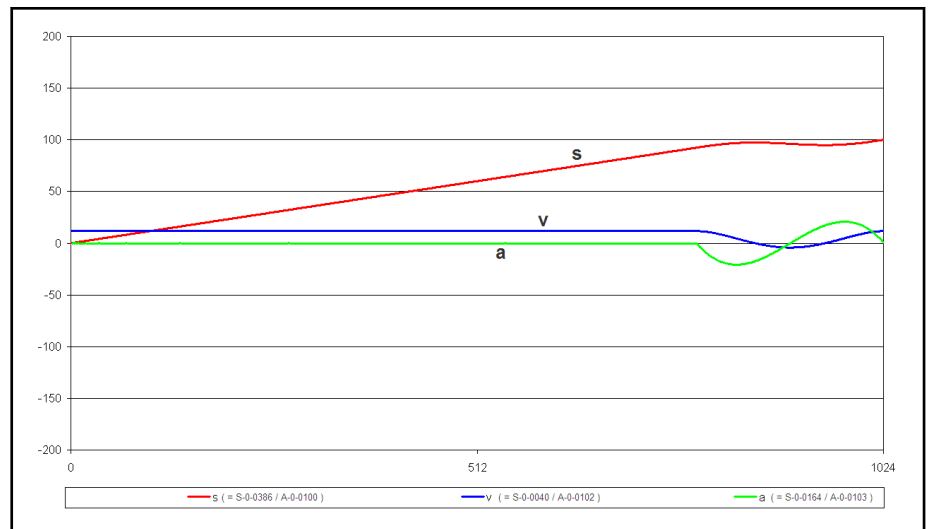


Fig. 4-104: MB_PrintLengthCorrCalcType01 MotionProfile



SyncLength: 1.3 * PrintLength; Correction: 200

Fig. 4-105: MB_PrintLengthCorrCalcType01 MotionProfile

Troubleshooting

The function block uses the error table F_RELATED_TABLE, 16x0170x. It can generate the following error messages in Additional1/Additional2:

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR	16#00002200	16#00000001	"NumberOfElements" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#00002240	16#00000001	"PrintFormat" input is < minimum
INPUT_RANGE_ERROR	16#00002240	16#00000002	"SyncLength" input is < minimum
INPUT_RANGE_ERROR	16#00002240	16#00000003	"Correction" input is < minimum or > maximum
INPUT_INVALID_ERROR	16#00002201	16#00000001	Address of the "CamTable" array is zero
INPUT_INVALID_ERROR	16#00002241	16#00000001	Invalid calculation: (PrintFormat + Correction) > SyncLength or SyncLength <= PrintFormat

Tab. 4-66: Error codes of the MB_PrintLengthCorrCalcType01 function block

4.11 Tension controller (TensionControl)

4.11.1 Introduction and overview

Depending on the measured value of a load cell, the tension controllers 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.

- [MB_TensionControlLoadCellType01 on page 276](#): Depending on the measured value of a load cell, it controls the tension to the desired command value.
- [MB_TensionControlLoadCellMatParType01 on page 287](#): Depending on the measured value of a load cell, it controls the web tension to the desired command value.

ML_TechMotion.library

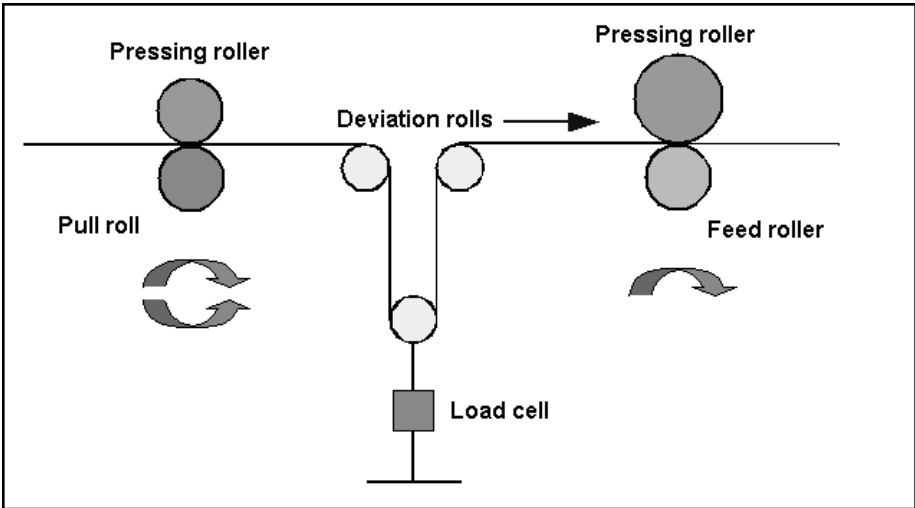


Fig. 4-106: Schematic diagram for the closed-loop tension control

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.

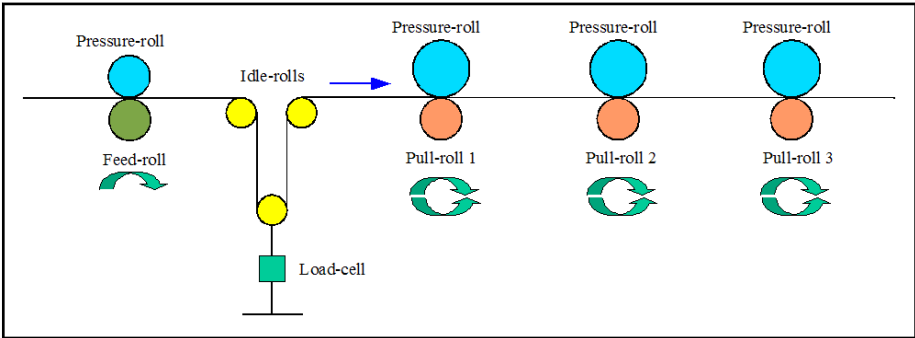


Fig. 4-107: Multi-axis tension controller

4.11.2 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
IndraMotion MLC 13VRS	ML_TechMotion
IndraMotion MLD with MPx07/08/17 with MA function package	MX_Technology_07.lib

Tab. 4-67: Reference table of the MB_TensionControlLoadCellType01



IndraMotion MLC 12VRS or higher also supports slave axes with interpolation on the control.

Interface description

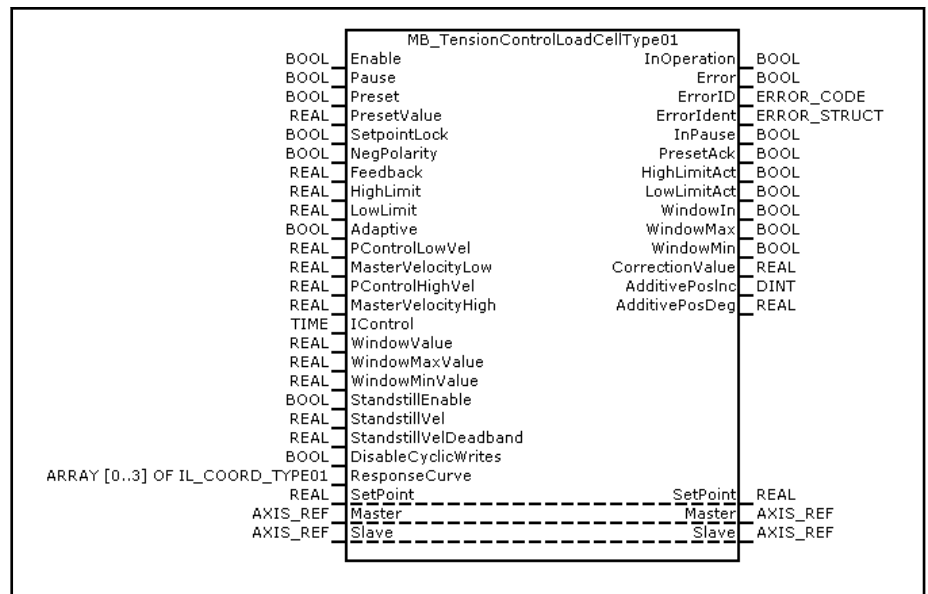


Fig. 4-108: MB_TensionControlLoadCellType01 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Master	AXIS_REF	Reference to the master axis
	Slave	AXIS_REF	Reference to the slave axis
	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_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
	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
	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

ML_TechMotion.library

I/O type	Name	Data type	Description
	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
	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 from 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	For a detailed error description according to the error table (Troubleshooting on page 286)

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
	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
	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) [-]
	AdditivePosDec	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) [°]

Tab. 4-68: 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
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

ML_TechMotion.library

Name	Type	Min. value	Max. value	Default value	Effective
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
StandstillDeadband	REAL	0.0	n.def.	1.0	Continuous

Tab. 4-69: 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.

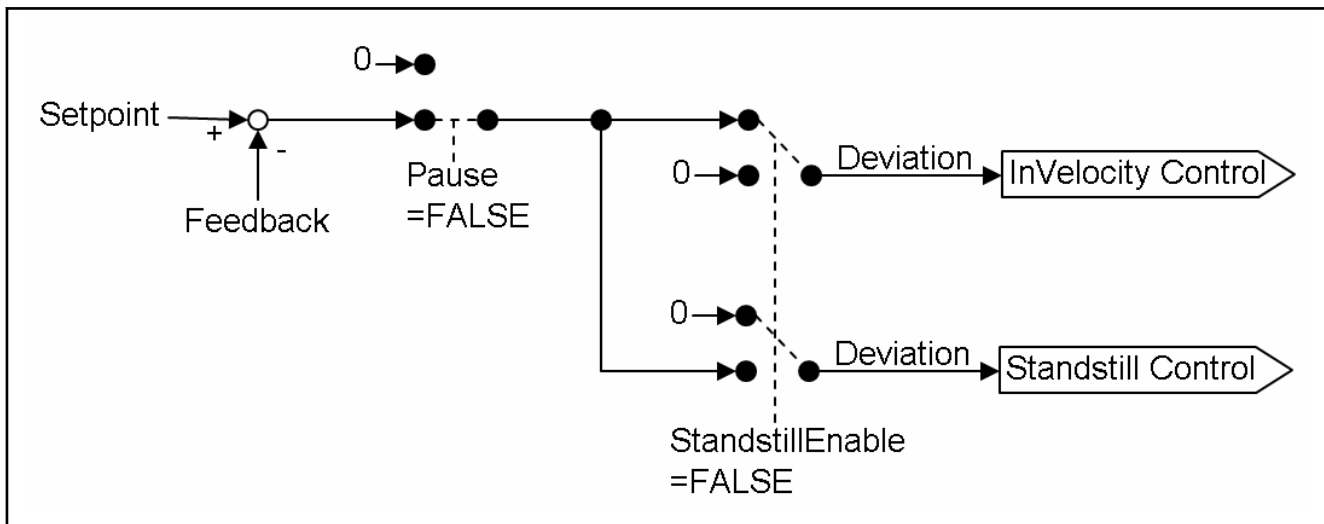


Fig. 4-109: Flowchart of the tension controller with branch for the control at standstill/in velocity

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.

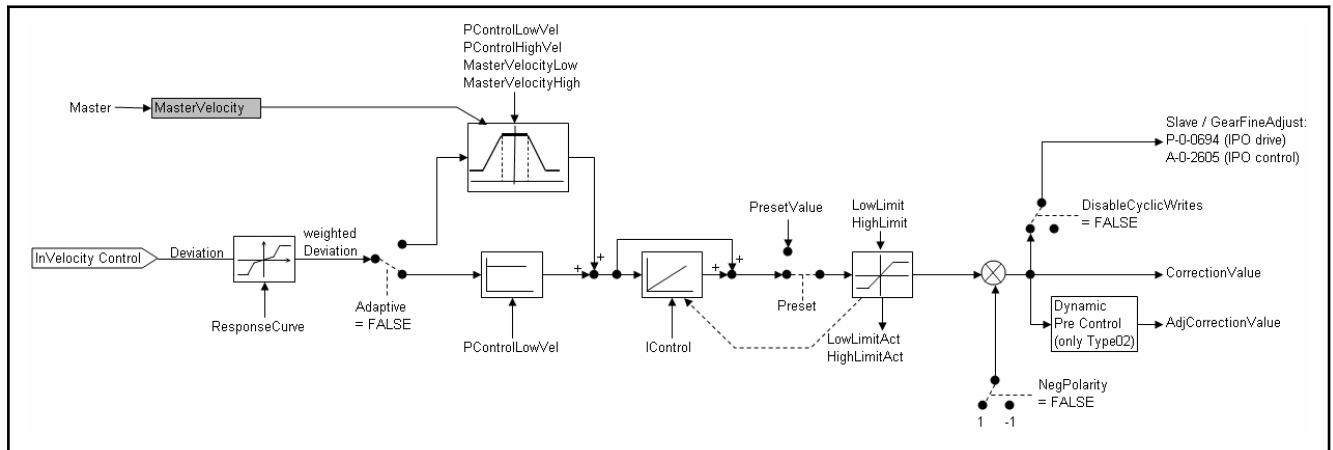


Fig. 4-110: Flowchart of the tension controller for the "InVelocity Control" branch

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.

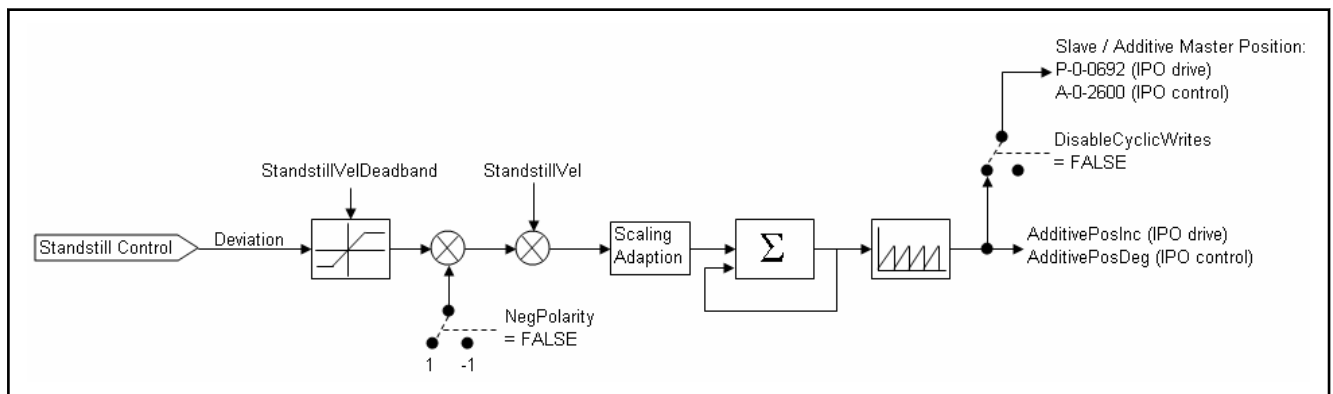


Fig. 4-111: Flowchart of the tension controller for the closed-loop control at standstill (Standstill Control)

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 IndraMotion 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" 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.

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:

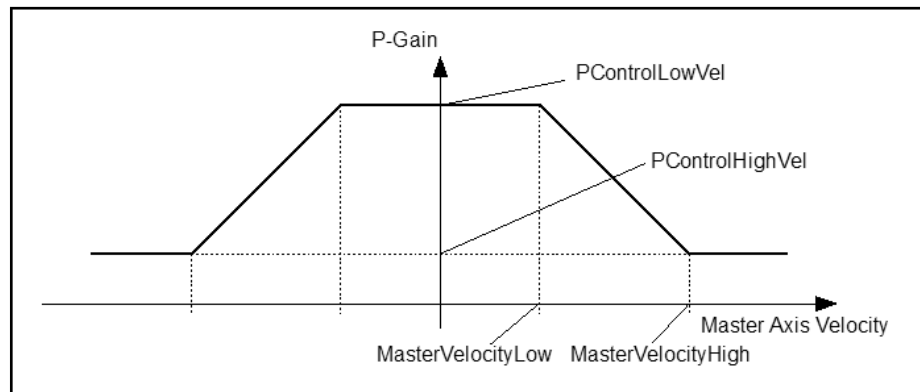


Fig. 4-112: Characteristic curve for the adaptive P-gain

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 section "Gear ratio fine adjustment" on page 285). 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:

$$\left(\text{SetPoint} - |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right) \leq \text{Feedback} \leq \left(\text{SetPoint} + |\text{SetPoint}| \cdot \frac{\text{WindowValue}}{100\%} \right)$$

Fig. 4-113: WindowValue, WindowIn

WindowMaxValue, WindowMax

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.

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. 4-114: 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. 4-115: 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 the 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 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:

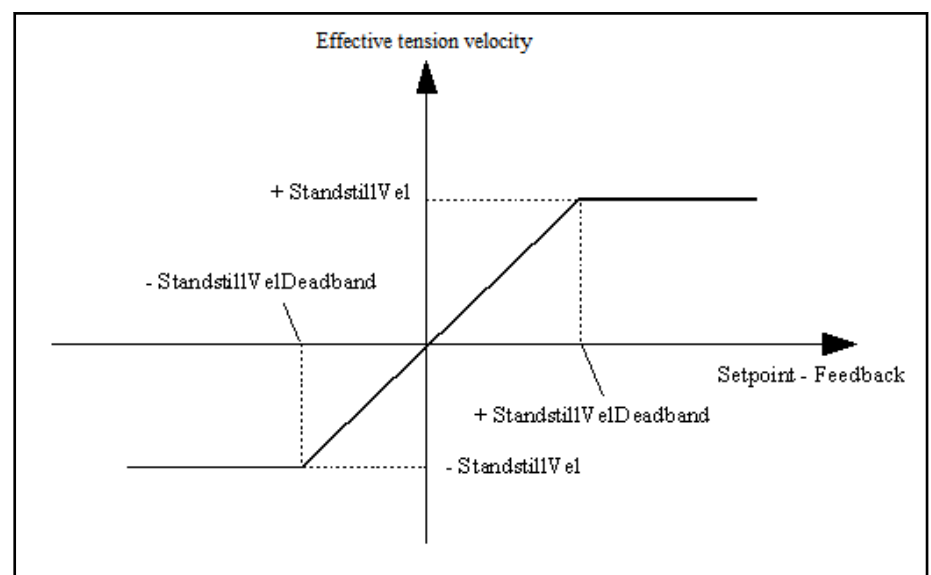


Fig. 4-116: Modified signum function to limit the added "StandstillVel"

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.

ML_TechMotion.library

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 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 as well. 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, 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 are applied. 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 ("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

ErrorID	Additional1	Additional2	Description
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 IndraMotion MLD

Tab. 4-70: Error codes of the MB_TensionControlLoadCellType01 function block

4.11.3 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, 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
IndraMotion MLC 13VRS	ML_TechMotion

Tab. 4-71: MB_TensionControlLoadCellMatParType01 reference table

ML_TechMotion.library

Interface description

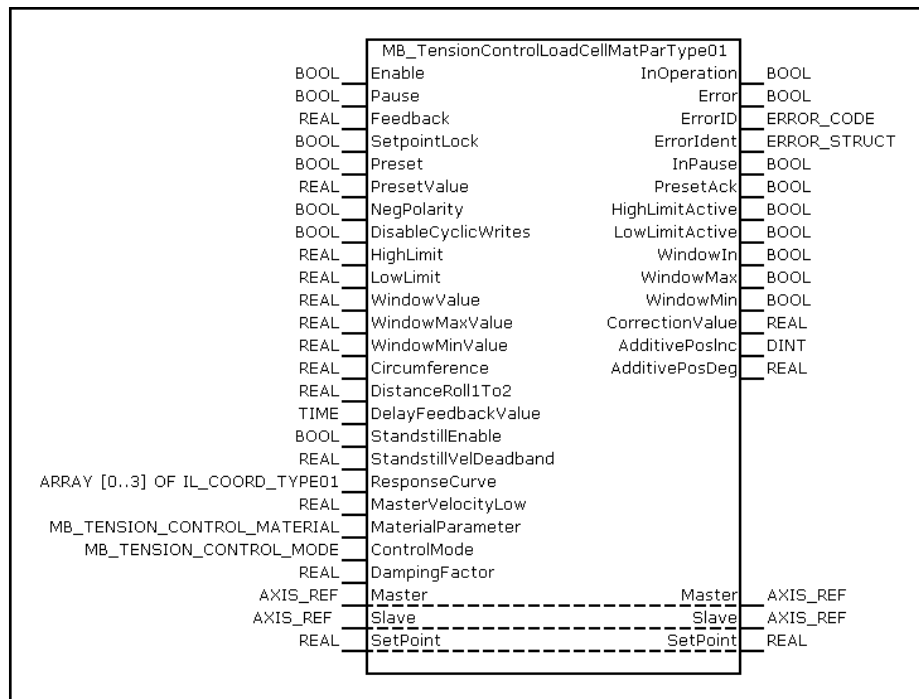


Fig. 4-117: MB_TensionControlLoadCellMatParType01 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Master	AXIS_REF	Reference to the master axis
	Slave	AXIS_REF	Reference to the slave axis
	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")
VAR_IN-PUT	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 "Preset-Value" 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 [%]

I/O type	Name	Data type	Description
	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) [%]
	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 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 "InVelocity Control" branch IL_COORD_TYPE01 from RIL_LoopControl.library
	MasterVelocityLow	REAL	Master axis velocity below which the controller adaption 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 on page 300

ML_TechMotion.library

I/O type	Name	Data type	Description
	ControlMode	MB_TENSION_CONTROL_MODE	Selection of the control loop response. Optimized for command value or disturbance response MB_TENSION_CONTROL_MODE on page 301
	DampingFactor	REAL	This input affects the control loop response [%]. It can be steplessly selected between a slow aperiodic and a fast periodic response.
VAR_OUT PUT	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	For a detailed error description according to the error table (see Troubleshooting on page 298).
	InPause	BOOL	"Pause" is active
	PresetAck	BOOL	If this output is active, the "Preset" input was analyzed and the "CorrectionValue" output applied the "PresetValue" 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 "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 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]

*

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. 4-72: 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
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_CO-ORD_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"

ML_TechMotion.library

Name	Type	Min. value	Max. value	Default value	Effective
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. 4-73: 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.

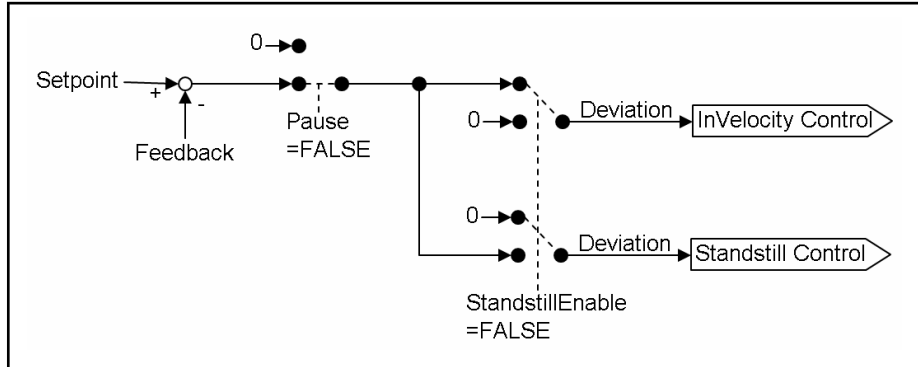


Fig. 4-118: Flowchart of the tension controller with branch for the control at standstill/in velocity

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).

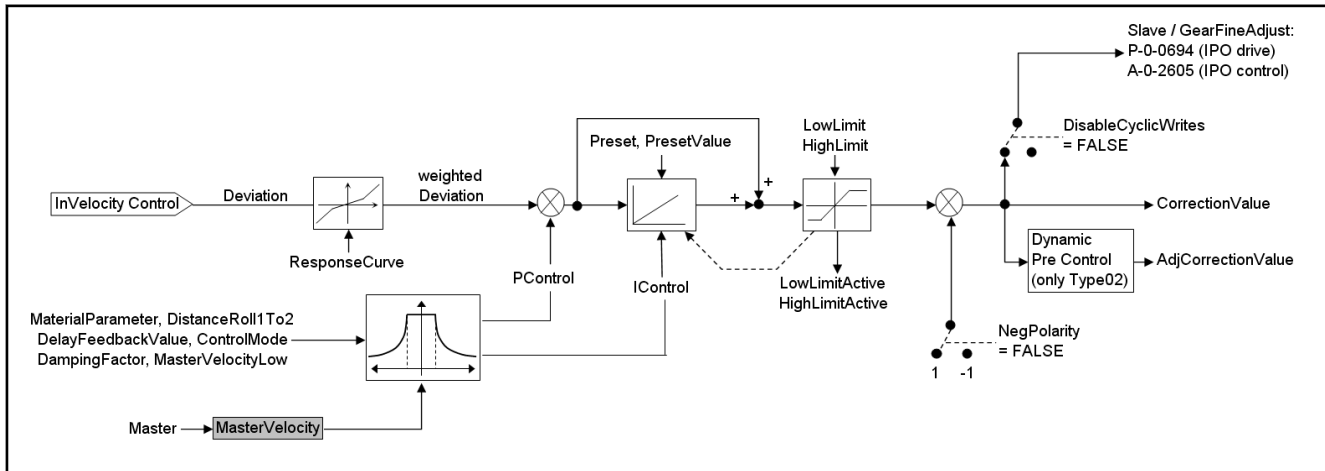


Fig. 4-119: Flowchart of the tension controller for the (InVelocity Control) branch

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.

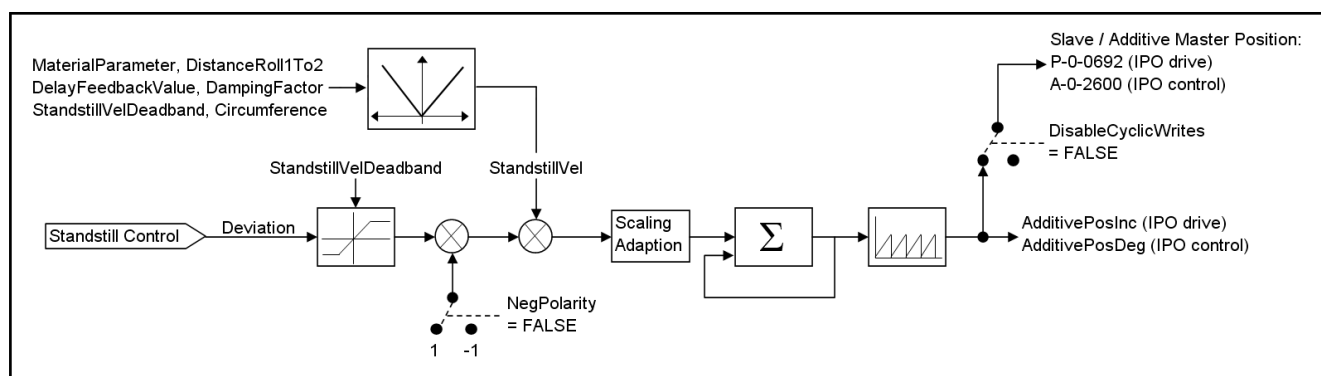


Fig. 4-120: Flowchart of the tension controller for the closed-loop control at standstill (Standstill Control)

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 section ["Gear ratio fine adjustment" on page 298](#)). 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. 4-121: 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:

$$Feedback > \left(SetPoint + |SetPoint| \cdot \frac{WindowMaxValue - 100}{100} \right)$$

Fig. 4-122: 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. 4-123: 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 total delay time of the web tension determination (excluding the Sercos cycle time). The transport dead times resulting from the Sercos transmission are considered function block-internally. Thus, this input corresponds to the sum of all filter and transmission times (exclusively the Sercos cycle times) in the actual value detection of the web tension.

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. This is achieved with a characteristic curve with up to four data points. If the input is not assigned, a 1-to-1 characteristic curve is used as default value. That means that it is not scaled.

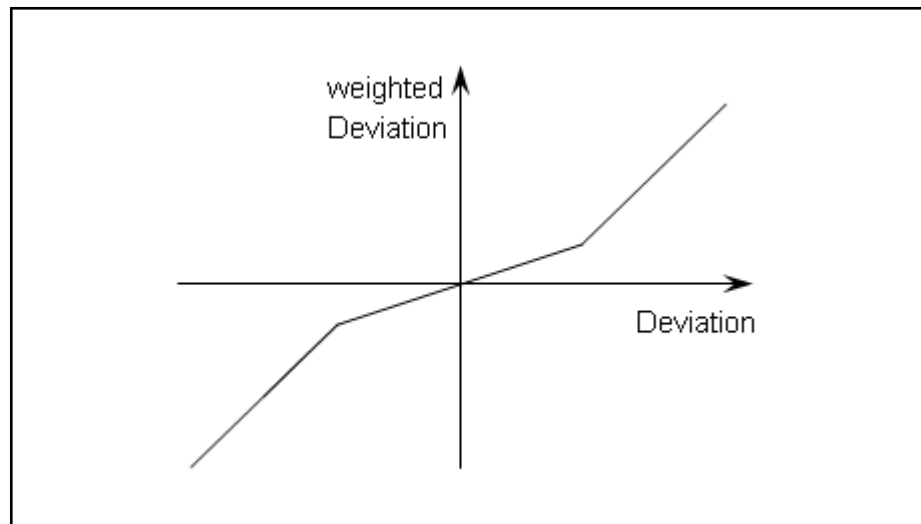


Fig. 4-124: Scaling the control deviation using the "ResponseCurve" input

The function can for example 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).

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.

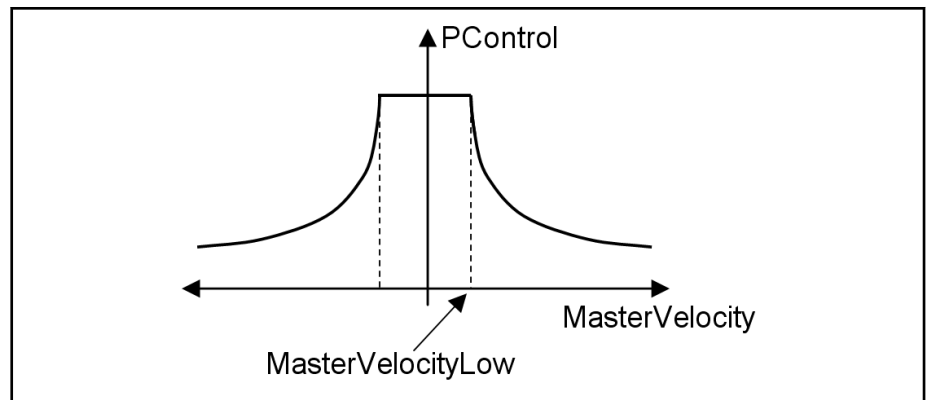


Fig. 4-125: Dependency of the controller gain from the web velocity

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. 4-126: 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. 4-127: 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

ML_TechMotion.library

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 IndraMotion MLD

Tab. 4-74: Error codes of the MB_TensionControlLoadCellMatParType01 function block

4.11.4 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 300	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/m ²]
MaterialEModule	REAL	>0.0 (if MaterialType is CUSTOM_...)	n.def.	0	Continuous	Modulus of elasticity of web material [N/mm ²]*

*

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. 4-75: Structural description of the MB_TENSION_CONTROL_MATERIAL

4.11.5 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)

Enumeration element	Number	Description
PET_FILM	7	Polyethylene terephthalate foil (PET film)
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. 4-76: Enumeration type MB_TENSION_CONTROL_MATERIAL_TYPE

4.11.6 MB_TENSION_CONTROL_MODE

The following table lists the characteristics of the enumeration type MB_TENSION_CONTROL_MODE.

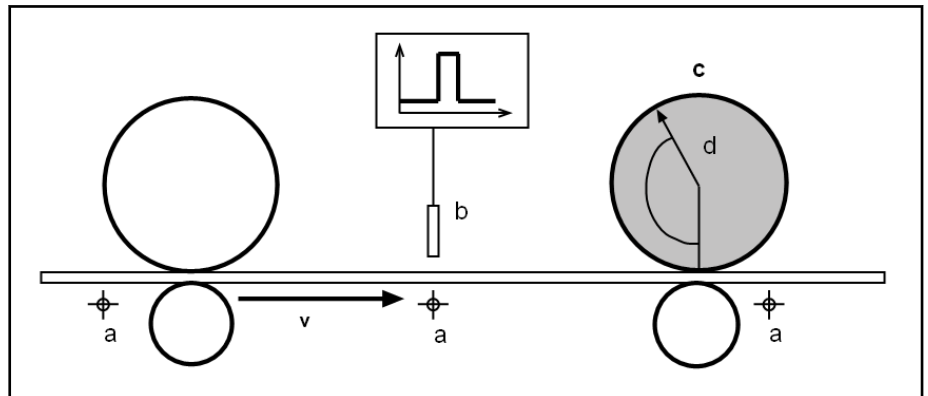
Enumeration element	Number	Description
CONTROL_MODE_OPT_DISTURBANCE	1	The function block optimizes the disturbance response of the "Feedback" value to the "Setpoint" value (default)
CONTROL_MODE_OPT_COMMAND	2	The function block optimizes the command value response of the "Feedback" value to the "Setpoint" value

Tab. 4-77: 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: ϕ print

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 following function blocks are supported:

ML_TechRegi (IndraMotion MLC)/ MX_TechRegi (IndraMotion MLD) function blocks

Area of application (folder in library)		
Function block	MLC	MLD
Editing measured touch probe signals (RegiMeasuring)		
MB_RegiMeasuringType01, page 304	X	X
MB_RegiMeasuringType02, page 306	X	X
MB_RegiMeasuringType03, page 310	X	X
Quality evaluation of printed images (RegiQuality)		
IL_RegiQualityType01, page 314	X	X
Write a control value to the drive (RegiRegulate)		
MB_RegiRegulateType01, page 317	X	X
MB_RegiRegulateType02, page 321	X	X
MB_RegiRegulateType03, page 329	X	X
MB_RegiRegulateType04, page 335	X	

ML_TechRegi.library/MX_TechRegi.lib

Locking a value for a register controller (RegiSetpointLock)		
MB_SetpointLockType01, page 344	X	X
Register controller function block (RegisterController)		
MB_RegisterControllerType02, page 346	X	X
MB_RegisterControllerType04, page 356	X	X
Register controller function block (RegisterControllerSide)		
MB_RegisterControllerSideType01, page 375	X	X

Tab. 5-1: Overview on function block support by the library

ML_TechRegi/MX_TechRegi function blocks

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.

The function block subtracts the "Setpoint" and "Feedback" inputs ("Setpoint" - "Feedback").

The result of the subtraction is applied to the "Deviation" output.

Target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechRegi
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-2: Reference table of the MB_RegiMeasuringType01

Interface description

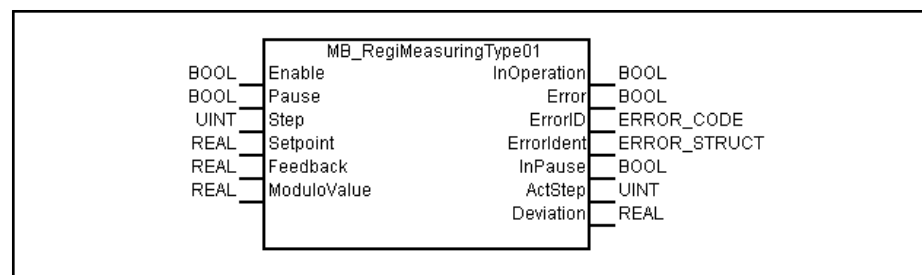


Fig. 5-2: MB_RegiMeasuringType01 function block

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

I/O type	Name	Data type	Description
	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 Troubleshooting, page 306
	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

Time diagram

Timing diagram according to the PLCopen specification.

ML_TechRegi.library/MX_TechRegi.lib

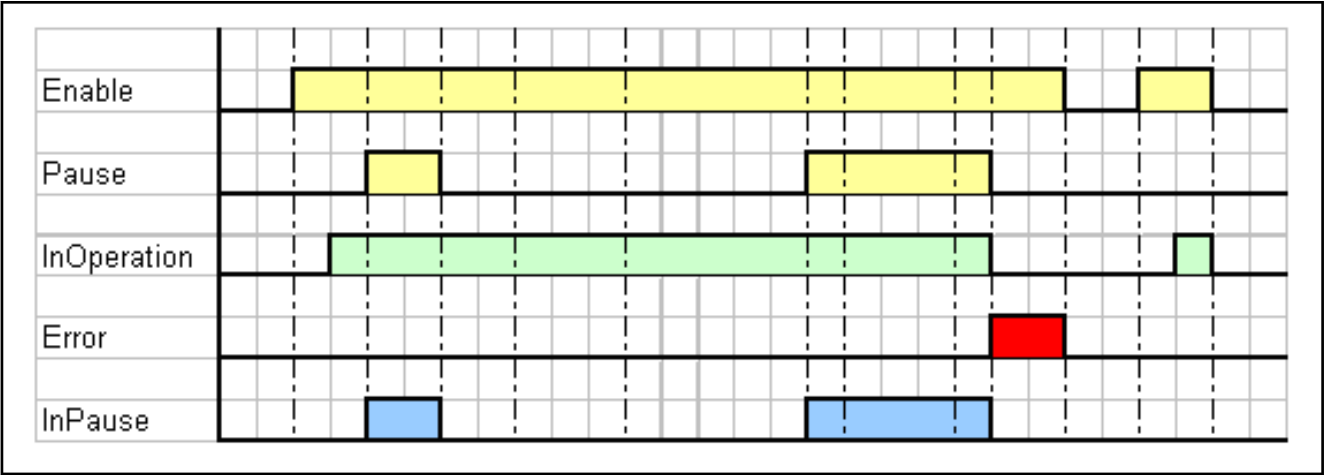


Fig. 5-3: Timing diagram of the MB_RegiMeasuringType01

- 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.
- 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
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.
The function block subtracts the "Setpoint" and "Feedback" inputs ("Setpoint" - "Feedback").
If the shift register is disabled, the result of the subtraction is applied to the "Deviation" output.
If the shift register is activated, it is delayed by the number of "ProbeDistance" when applied.

ML_TechRegi.library/MX_TechRegi.lib

Target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechRegi
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-6: Reference table of the MB_RegiMeasuringType02

Interface description

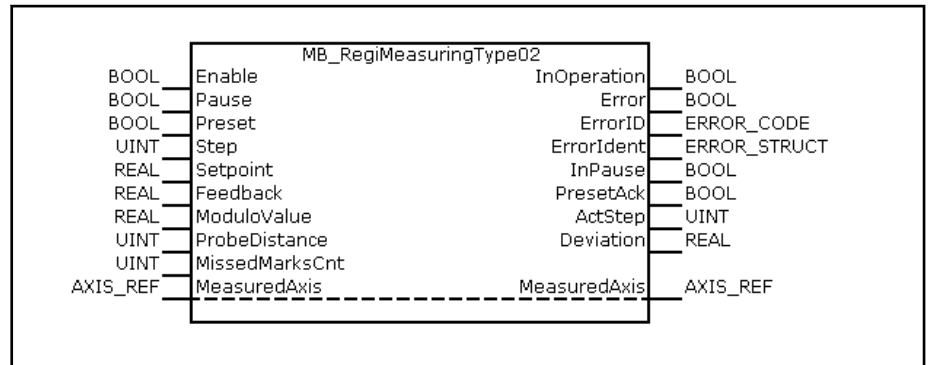


Fig. 5-4: MB_RegiMeasuringType02 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	MeasuredAxis	AXIS_REF	Reference to the measured axis
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 309
	InPause	BOOL	"Pause" is active. The "Deviation" output maintains its value

ML_TechRegi.library/MX_TechRegi.lib

I/O type	Name	Data type	Description
	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

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

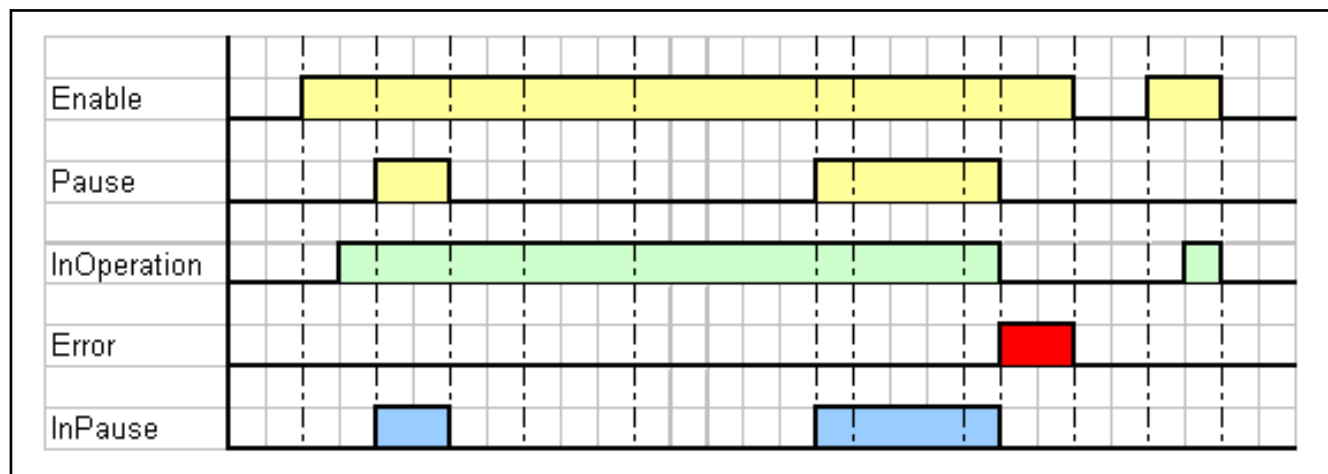
Time diagram Timing diagram according to the PLCopen specification.

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 lengths is 30. If "ProbeDistance" is 0, the shifting register is deactivated. The "Deviation" output is set to "Setpoint" "Feedback". If "ProbeDistance" is not 0, then 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 input "ProbeDistance".

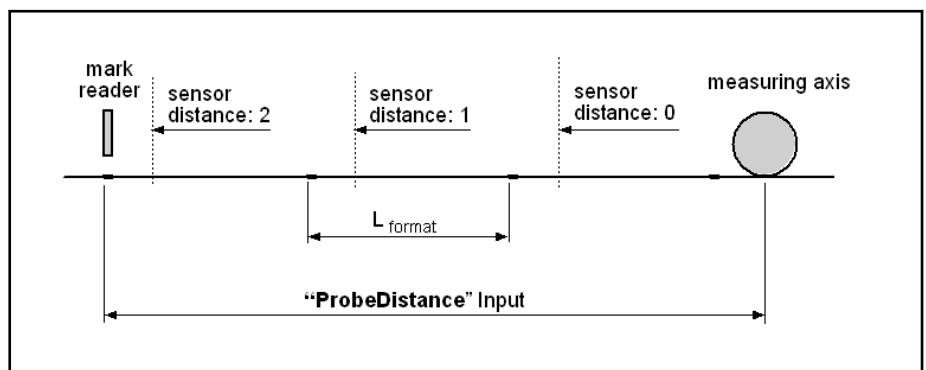


Fig. 5-6: Setting "ProbeDistance" Input

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"

ML_TechRegi.library/MX_TechRegi.lib

ErrorID	Additional1	Additional2	Description
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.

The function block subtracts the "Setpoint" and "Feedback" inputs ("Setpoint" - "Feedback").

If the shift register is disabled, the result of the subtraction is applied to the "Deviation" output.

If the shift register is activated, it is delayed by the number of "ProbeDistance" when applied.

Only every nth mark is for the registration with the mark gating feature of the function block.

Target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechRegi
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-10: Reference table of the MB_RegiMeasuringType03

Interface description

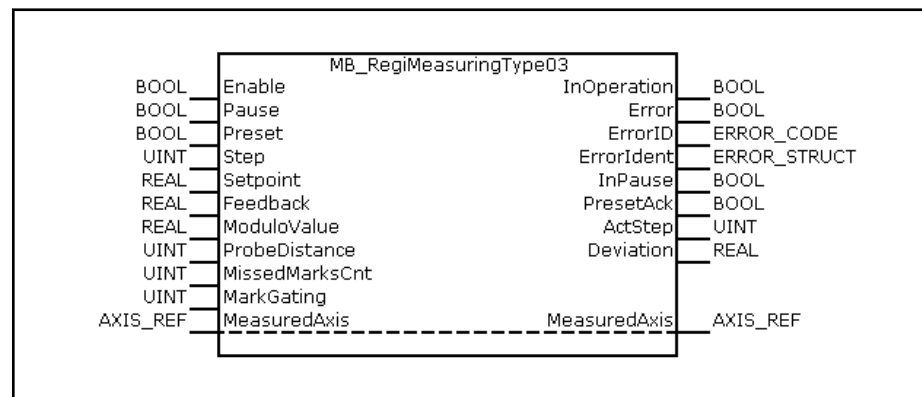


Fig. 5-7: MB_RegiMeasuringType03 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	MeasuredAxis	AXIS_REF	Reference to the measured axis
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

I/O type	Name	Data type	Description
	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 313
	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 at the "Deviation" output
	Deviation	REAL	"Setpoint" - "Feedback" delayed if shift register is active

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

ML_TechRegi.library/MX_TechRegi.lib

Name	Type	Min. value	Max. value	Default value	Effective
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.

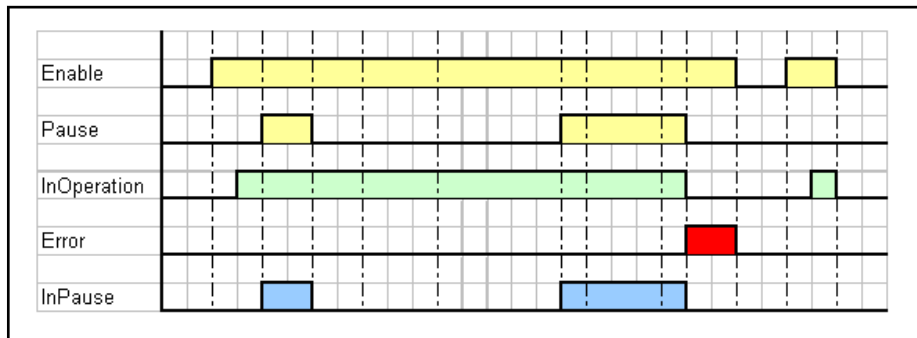


Fig. 5-8: Timing diagram of the MB_RegiMeasuringType03

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.

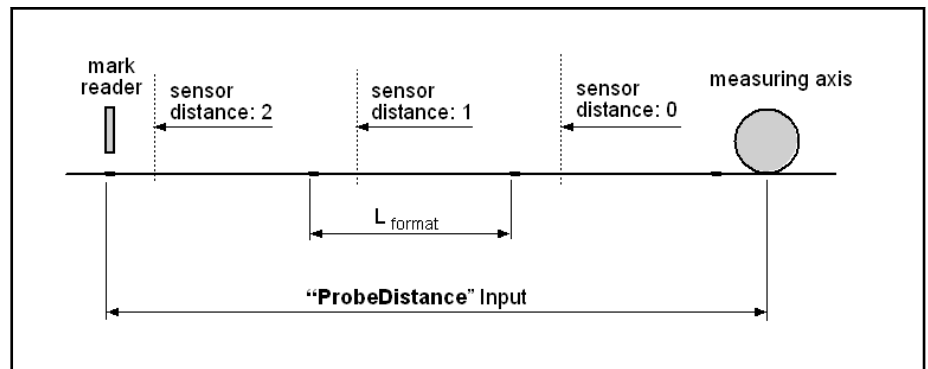


Fig. 5-9: Setting the "ProbeDistance" input

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 N-th 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 appears several times. In this situation, the mark distances between consecutive marks are not constant. For example, the mark distances between every N-th consecutive mark is constant. The RegiMeasuring function block only uses every N-th mark for the register control.

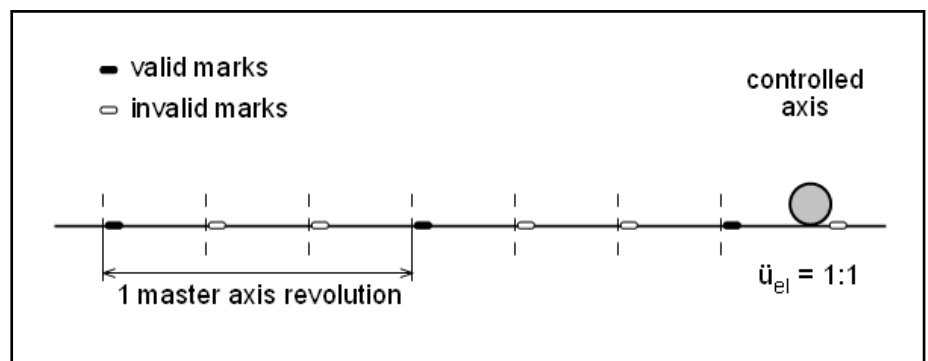


Fig. 5-10: Mark gating

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"

ML_TechRegi.library/MX_TechRegi.lib

ErrorID	Additional1	Additional2	Description
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 (IndraMotion MLC) and MX_TechRegi.lib (IndraMotion 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	Library
IndraMotion MLC 13VRS	ML_TechRegi
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-14: Reference table of the IL_RegiQualityType01

Interface description

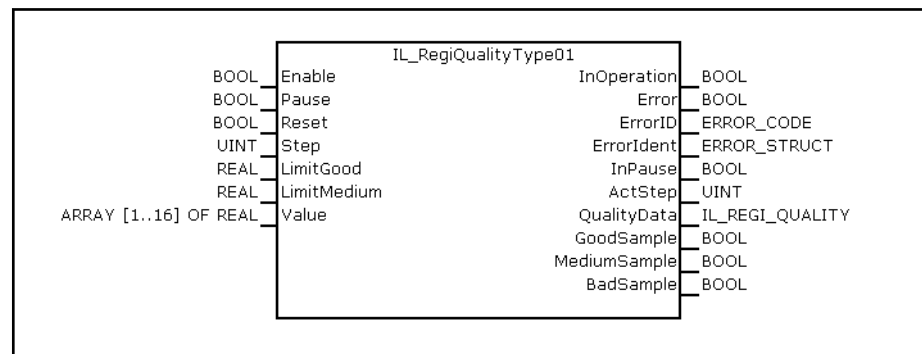


Fig. 5-11: IL_RegiQualityType01 function block

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"

ML_TechRegi.library/MX_TechRegi.lib

I/O type	Name	Data type	Description
	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 316
	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 316
	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 a classified in "good" and "bad" (refer to the following figure).

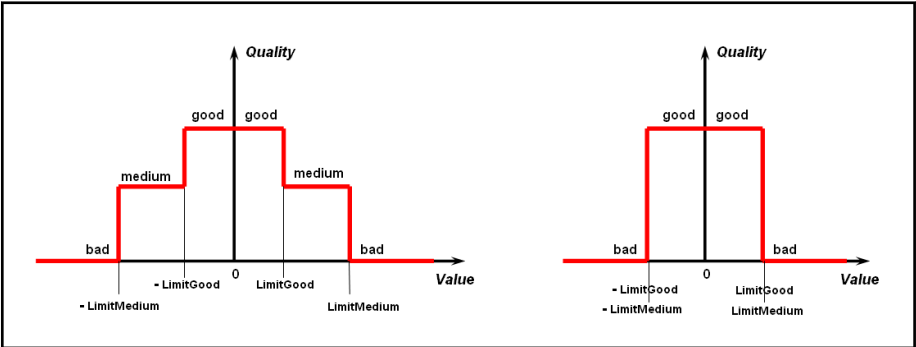


Fig. 5-12: Quality assessment principle for the IL_RegiQualityType01

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 314.

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

Name	Type	Description
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
IndraMotion MLC 13VRS	ML_TechRegi
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-19: Reference table of the MB_RegiRegulateType01

Interface description

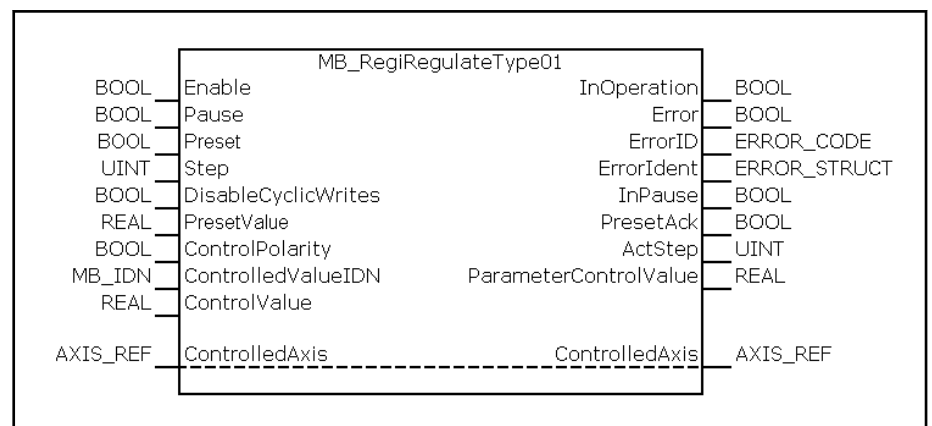


Fig. 5-13: MB_RegiRegulateType01 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	ControlledAxis	AXIS_REF	Reference to the controlled axis
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

ML_TechRegi.library/MX_TechRegi.lib

I/O type	Name	Data type	Description
	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 320
	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 input "DisableCyclicWrites" = FALSE

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"

Name	Type	Min. value	Max. value	Default value	Effective
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.

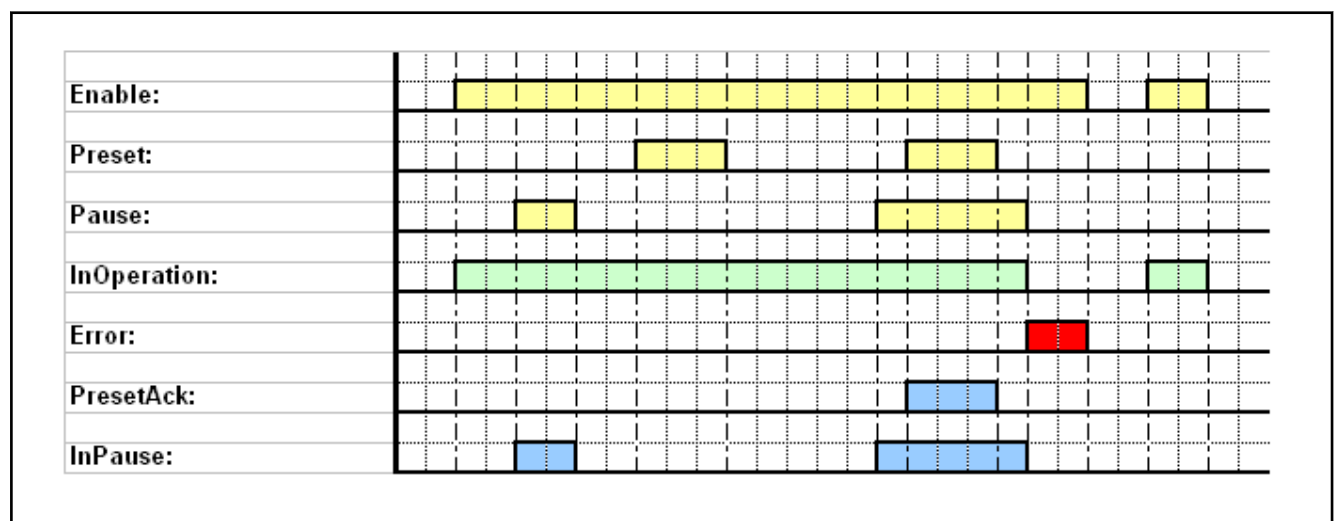


Fig. 5-14: Timing diagram of the MB_RegiRegulateType01

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.

ML_TechRegi.library/MX_TechRegi.lib

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 **IndraMotion 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 or a **virtual axis** is used as controlled 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

IndraMotion 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

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
IndraMotion MLC 13VRS	ML_TechRegi
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-23: Reference table of the MB_RegiRegulateType02

Interface description

ML_TechRegi.library/MX_TechRegi.lib

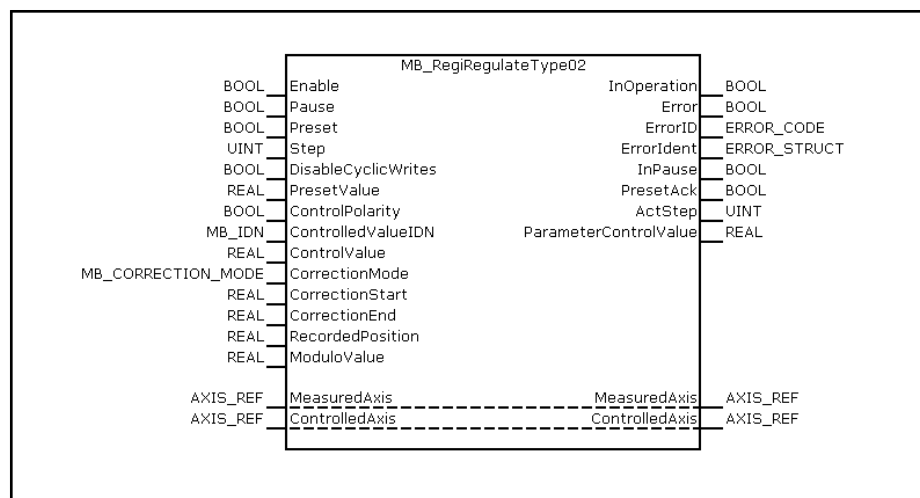


Fig. 5-15: MB_RegiRegulateType02 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	MeasuredAxis	AXIS_REF	Reference to the measured axis
	ControlledAxis	AXIS_REF	Reference to the controlled axis
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_PType02 for example
	CorrectionMode	MB_CORRECTION_MODE	Properties of the correction window MB_CORRECTION_MODE, page 343
	CorrectionStart	REAL	Start of the correction window
	CorrectionEnd	REAL	End of the correction window
	RecordedPosition	REAL	Measured value

I/O type	Name	Data type	Description
	ModuloValue	REAL	Modulo value of the measured 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 328
	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

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

ML_TechRegi.library/MX_TechRegi.lib

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.

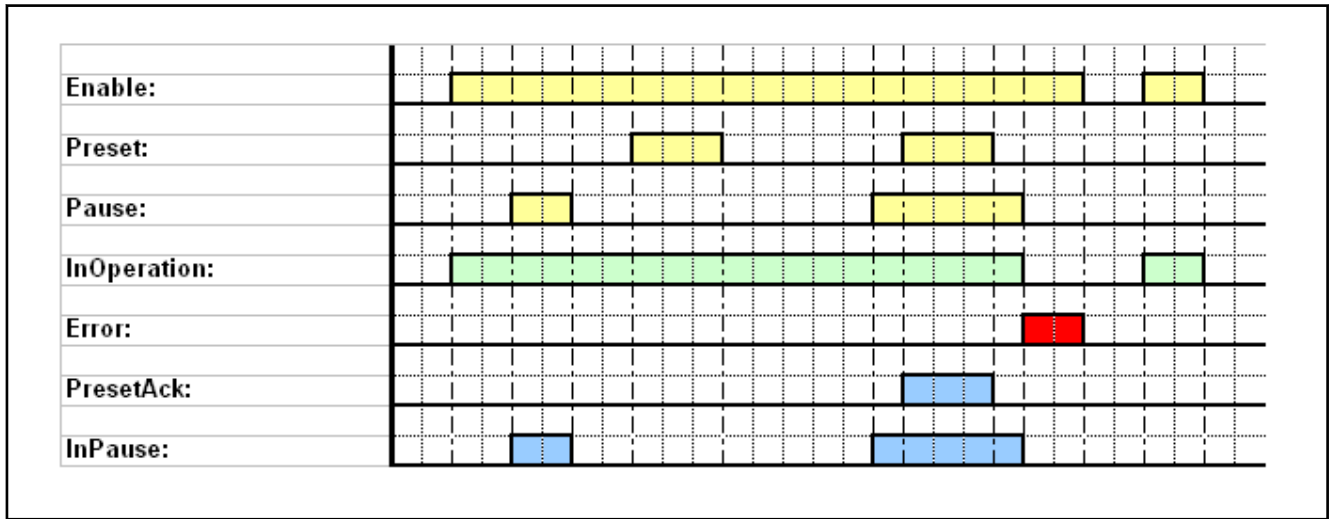


Fig. 5-16: Timing diagram of the MB_RegiRegulateType02

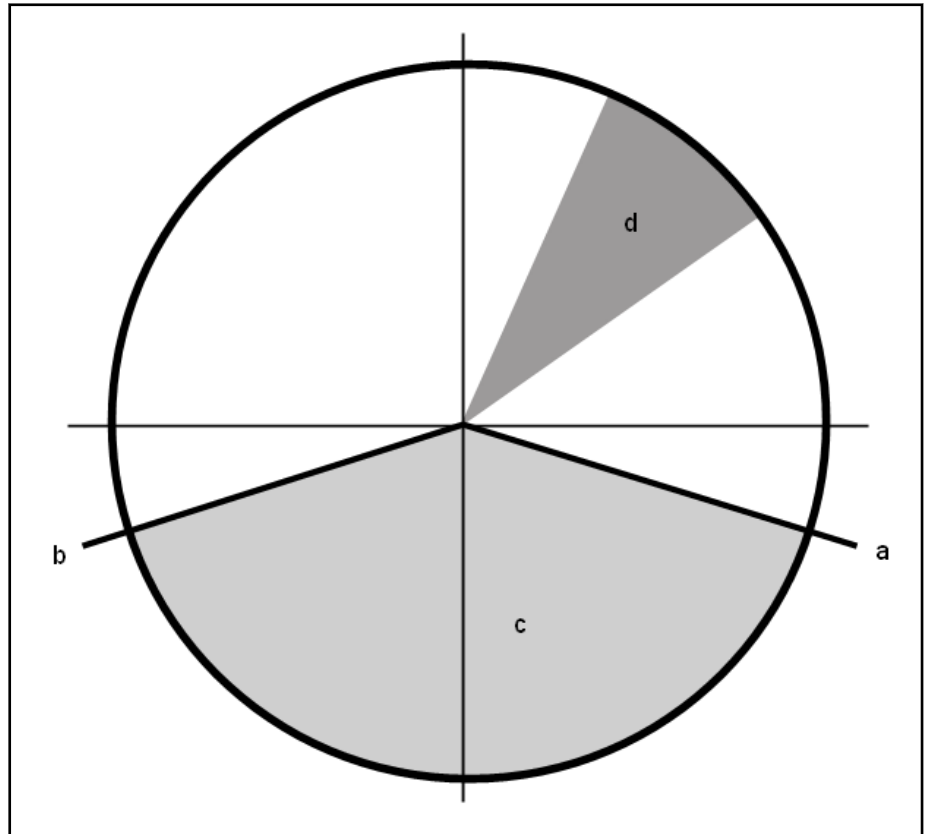
Functional Description	The function block writes control values to a parameter in an axis. Depending on the target parameter, "ControlledValueIDN", the function block includes an integration of the "ParameterControlValue" input. The following "Controlled-ValueIDN" values are integrated: • P-0-0061, A-0-2730 • 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.

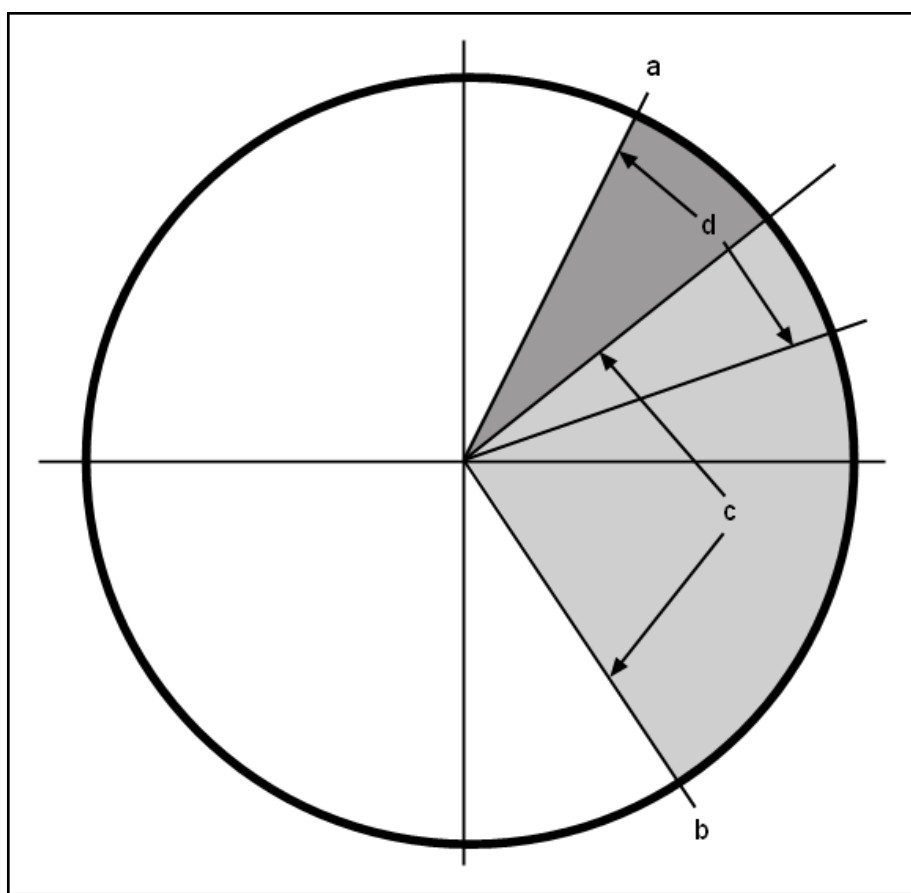
The correction window supports only a positive modulo range (e.g., 0...+360 degrees) of the measuring axis.



- a:** Start correction window
- b:** End correction window
- c:** Correction Window
- d:** Expectation window

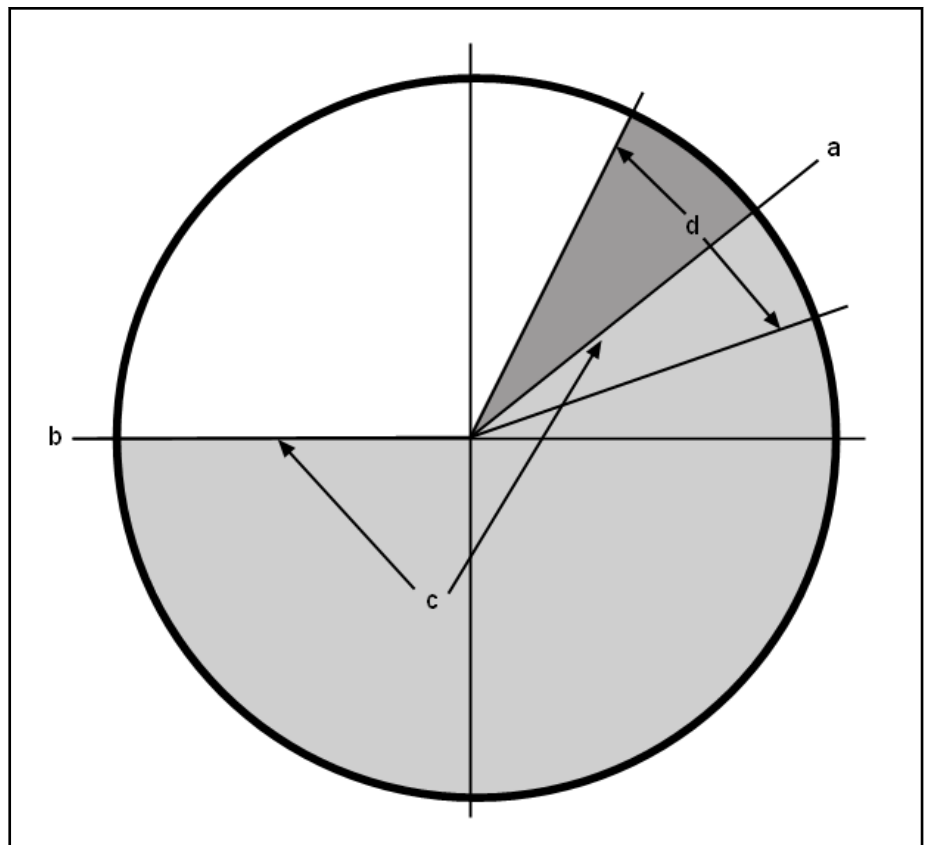
Fig. 5-17: Correction Window Absolute

ML_TechRegi.library/MX_TechRegi.lib



- 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 RegiRegulateType function block only calculates a new value if the "Step" Input changes. The controlled and measured axis must be modulo axes.

Required Parameterization

IndraMotion 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 or a **virtual axis** is used as controlled 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

IndraMotion 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:

ML_TechRegi.library/MX_TechRegi.lib

- P-0-0061, P-0-0690, P-0-0691, P-0-0692, P-0-0694, P-0-0695

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"

ErrorID	Additional1	Additional2	Description
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
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
IndraMotion MLC 13VRS	ML_TechRegi
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-27: Reference table of the MB_RegiRegulateType03

Interface description

ML_TechRegi.library/MX_TechRegi.lib

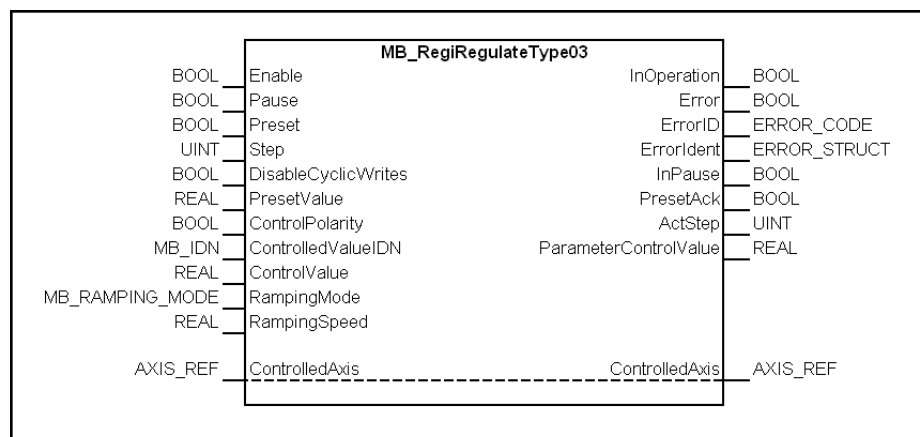


Fig. 5-20: MB_RegiRegulateType03 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	ControlledAxis	AXIS_REF	Reference to the controlled axis
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
	RampingMode	MB_RAMPING_MODE	Specification of the ramping velocity type. MB_RAMPING_MODE, page 344
	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 334
	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

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.

ML_TechRegi.library/MX_TechRegi.lib

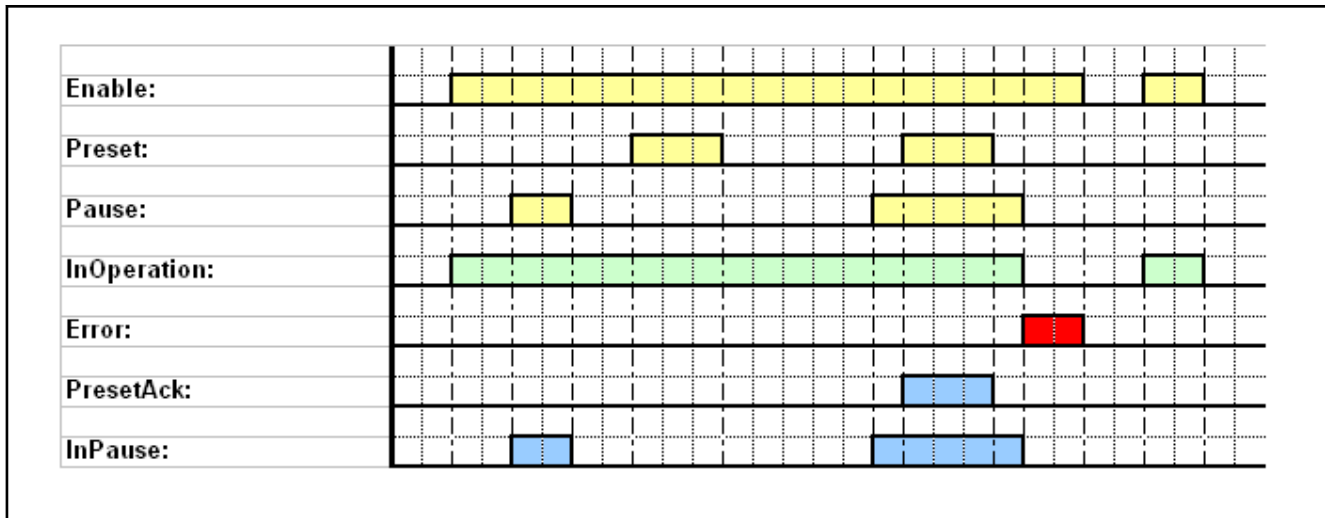


Fig. 5-21: Timing diagram of the MB_RegiRegulateType03

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.

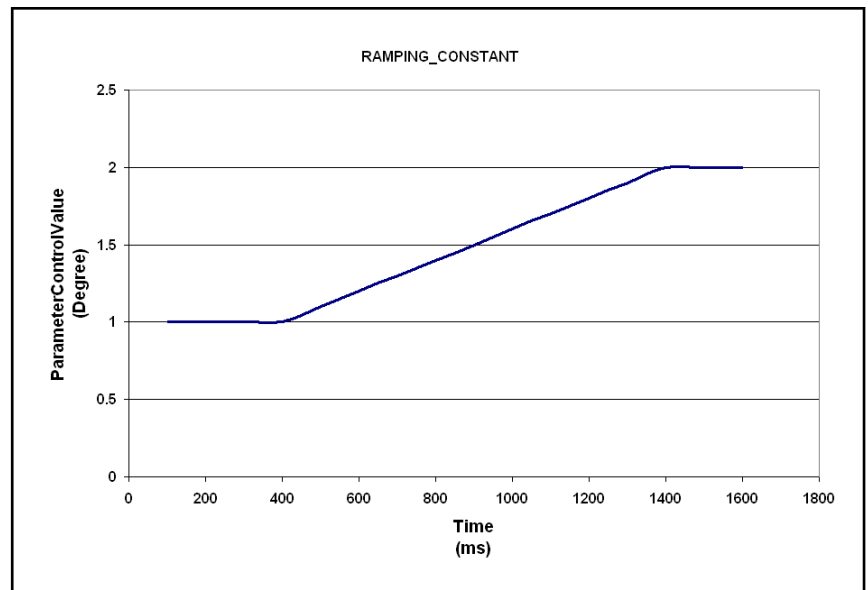


Fig. 5-22: Ramping constant

- 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.

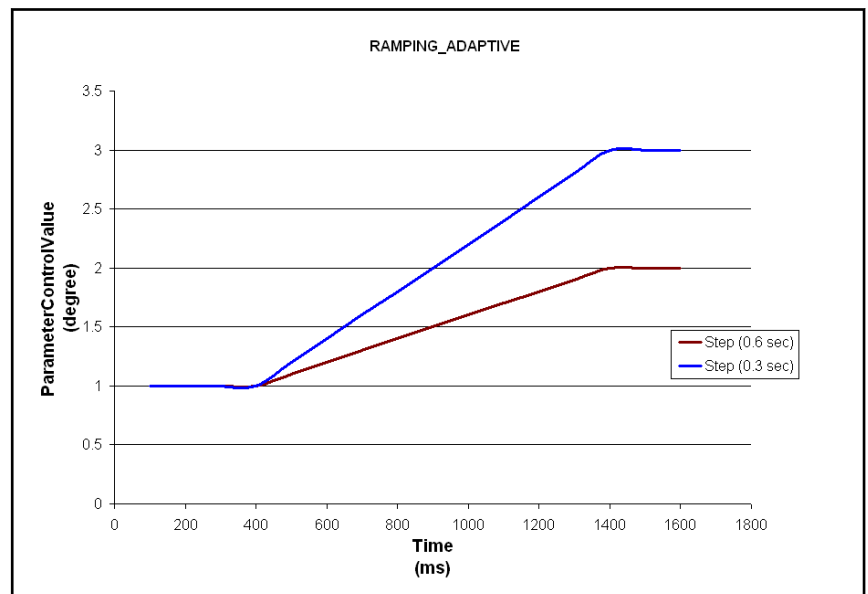


Fig. 5-23: Adaptive ramping

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	IndraMotion 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:

ML_TechRegi.library/MX_TechRegi.lib

- 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 or a **virtual axis** is used as controlled 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

IndraMotion 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

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
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)

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#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
IndraMotion MLC 13VRS	ML_TechRegi

Tab. 5-31: Reference table of the MB_RegiRegulateType04

Interface description

ML_TechRegi.library/MX_TechRegi.lib

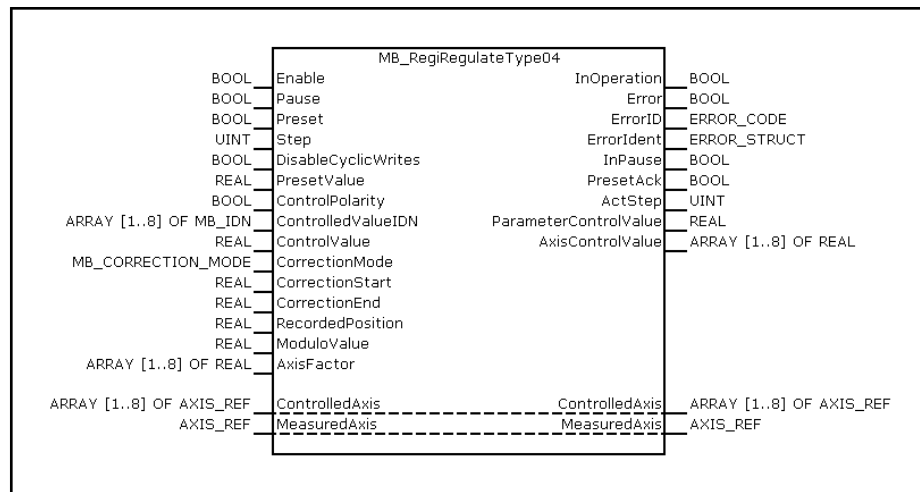


Fig. 5-24: MB_RegiRegulateType04 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	MeasuredAxis	AXIS_REF	Reference to the measured axis
	ControlledAxis	ARRAY [1..8] OF AXIS_REF	Reference to the controlled axes
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: 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
	CorrectionMode	MB_CORRECTION_MODE	Properties of the correction window MB_CORRECTION_MODE, page 343
	CorrectionStart	REAL	Start of the correction window
	CorrectionEnd	REAL	End of the correction window

I/O type	Name	Data type	Description
	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
	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

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
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"

ML_TechRegi.library/MX_TechRegi.lib

Name	Type	Min. value	Max. value	Default value	Effective
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.

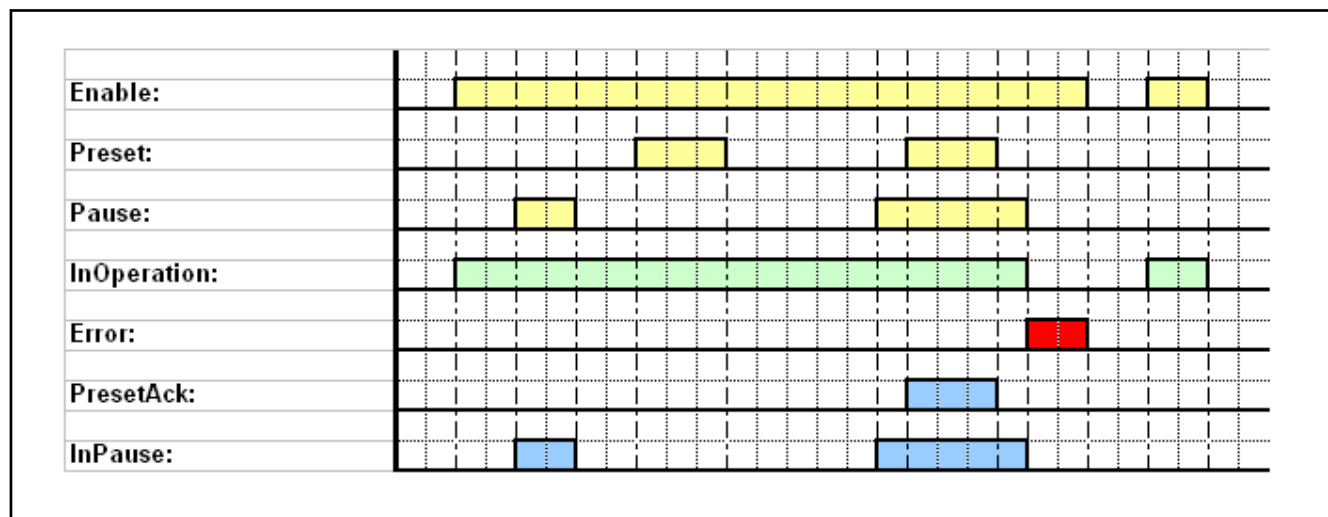


Fig. 5-25: Timing diagram of the MB_RegiRegulateType04

Functional Description

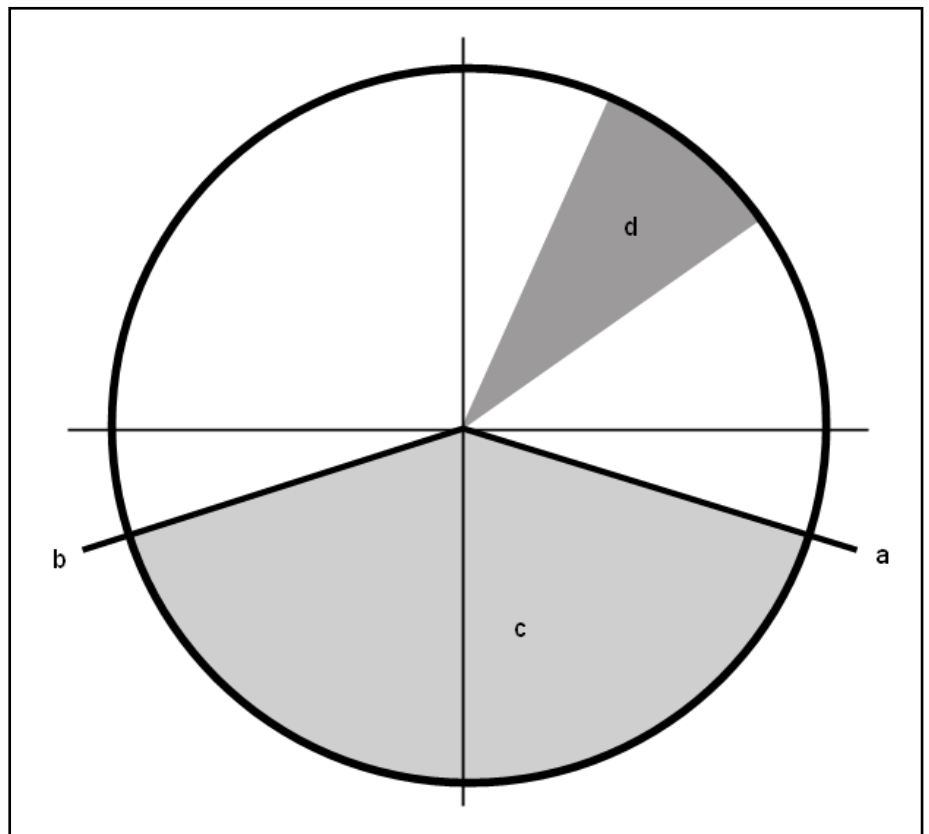
The function block writes control values to a parameter in an axis. Depending on the parameter, "ControlledValueIDN", the function block includes the integration of the "ParameterControlValue" input. This integration is done for the following ControlledValueIDN values:

- 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.

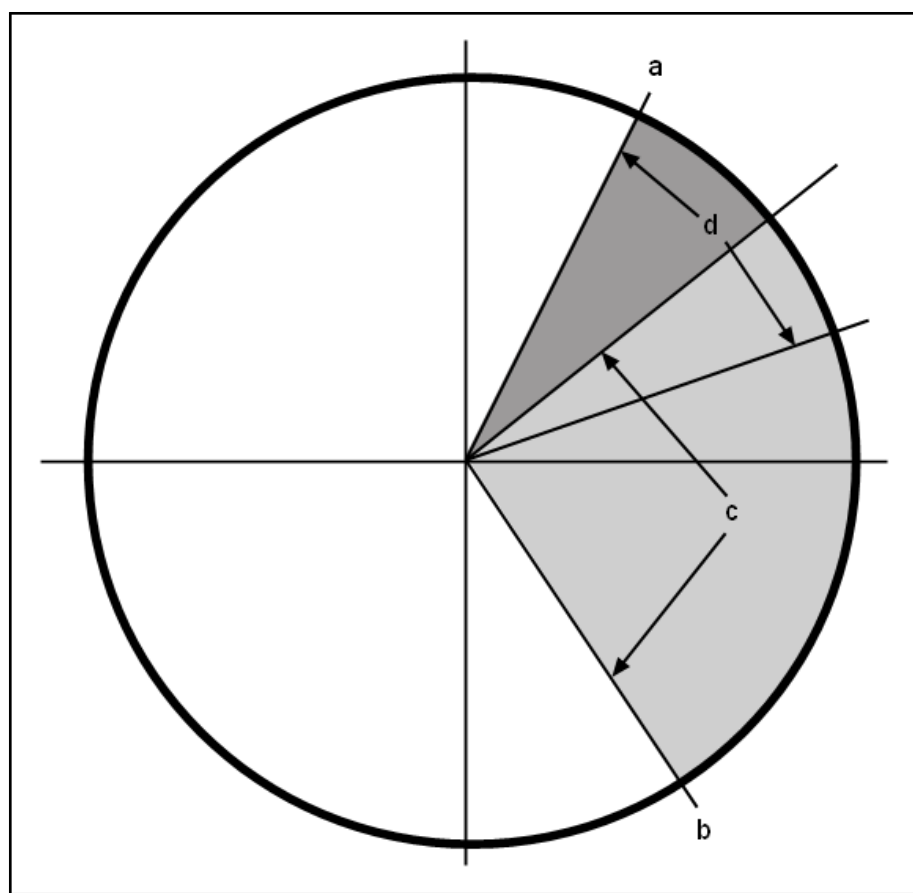
- Preset** The "PresetAck" output is active if the value in "PresetValue" is taken into account. The "Preset" input has priority over "Pause". The value in "PresetValue" is taken into account on a rising edge of "Preset" if the "Enable" input is active. If successful, the "PresetAck" output is set TRUE 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

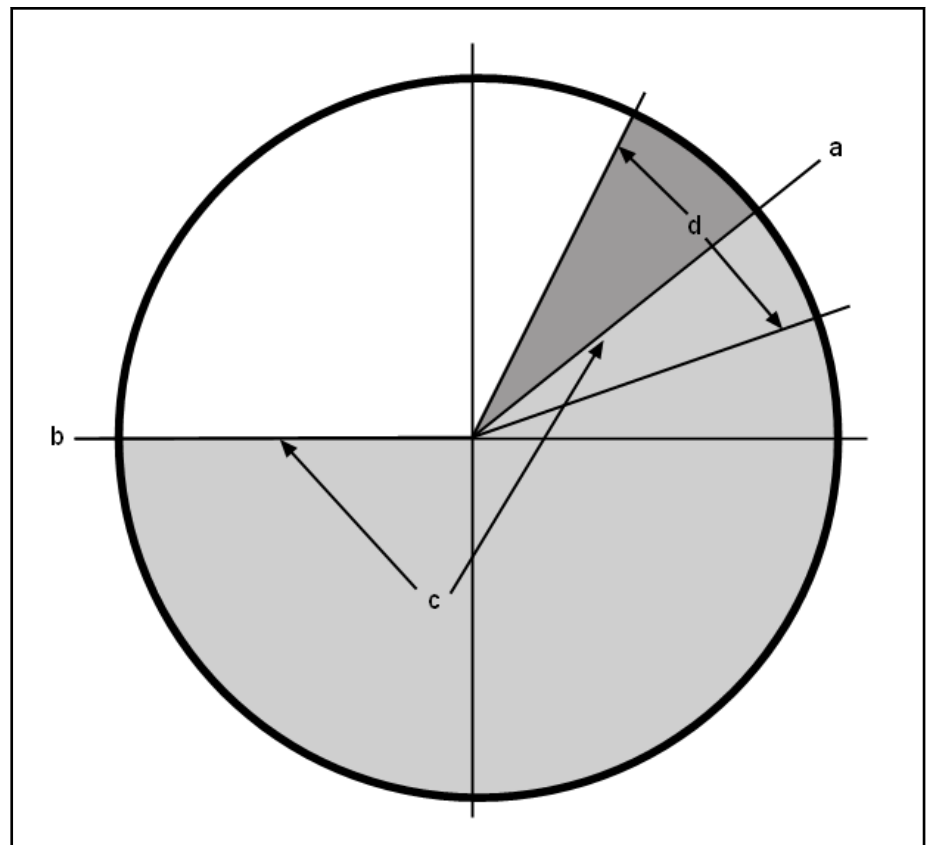
Fig. 5-26: Correction Window Absolute

ML_TechRegi.library/MX_TechRegi.lib



- 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 RegiRegulateType function block only calculates a new value if the "Step" Input changes. The controlled and measured axis must be modulo axes.
Required Parameterization	IndraMotion 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 or a **virtual axis** is used as controlled 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

ML_TechRegi.library/MX_TechRegi.lib

IndraMotion 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

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 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

ErrorID	Additional1	Additional2	Description
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)
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](#), [Seite 321](#), and [MB_RegiRegulateType04](#), [page 335](#).

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"]

ML_TechRegi.library/MX_TechRegi.lib

Name	Value	Description
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 329](#), 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 (IndraMotion MLC) and MX_TechRegi.lib (IndraMotion MLD).

5.5.2 MB_SetpointLockType01

Brief description The MB_RegiSetpointLockType01 function block latches a value for a register controller.

Target system	Library
IndraMotion MLC 13VRS	ML_TechRegi
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-37: Reference table of the MB_SetpointLockType01

Interface description

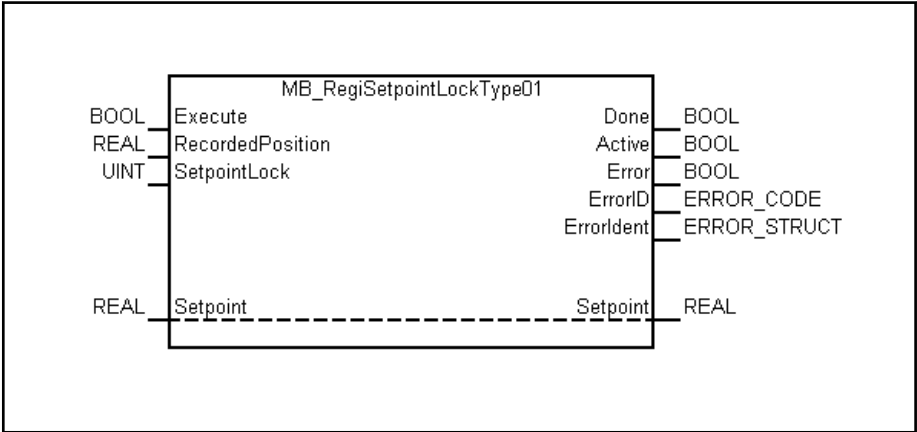


Fig. 5-29: MB_SetpointLockType01 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Setpoint	REAL	Setpoint
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
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 346

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.

ML_TechRegi.library/MX_TechRegi.lib

- Pause function
- Min./max. limitation of the calculated control value
- Correction window function
- Lock setpoint on demand

Target system	Library
IndraMotion MLC 13VRS	ML_TechRegi
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-41: Reference table of the MB_RegisterControllerType02

Interface description

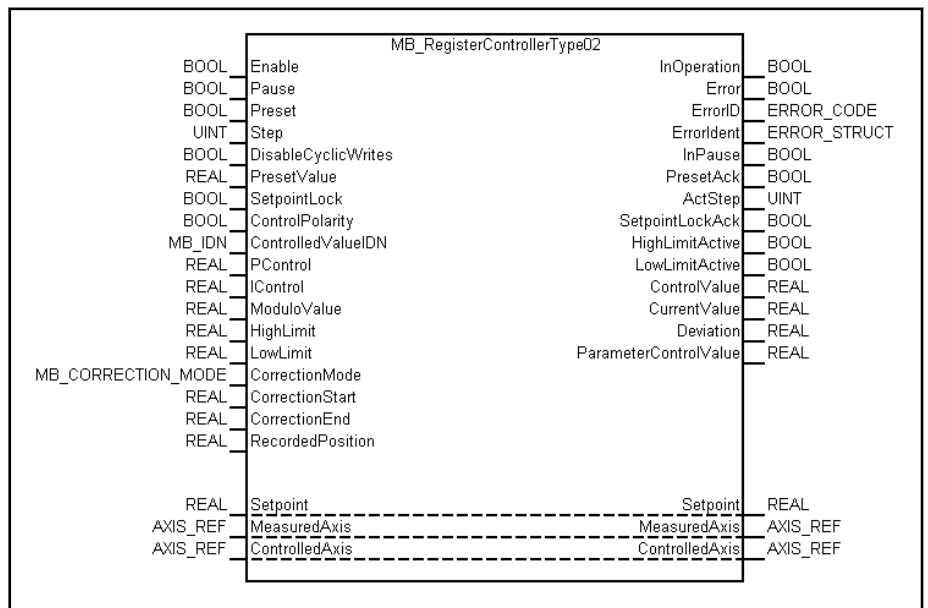


Fig. 5-31: MB_RegisterControllerType02 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Setpoint	REAL	Command value. This value is compared to "RecordedPosition"
	MeasuredAxis	AXIS_REF	Reference to the measured axis
	ControlledAxis	AXIS_REF	Reference to the controlled axis
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

ML_TechRegi.library/MX_TechRegi.lib

I/O type	Name	Data type	Description
	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: 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
	PControl	REAL	Proportional gain of the PI-controller
	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 343
	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 356
	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"

I/O type	Name	Data type	Description
	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

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

ML_TechRegi.library/MX_TechRegi.lib

Name	Type	Min. value	Max. value	Default value	Effective
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.

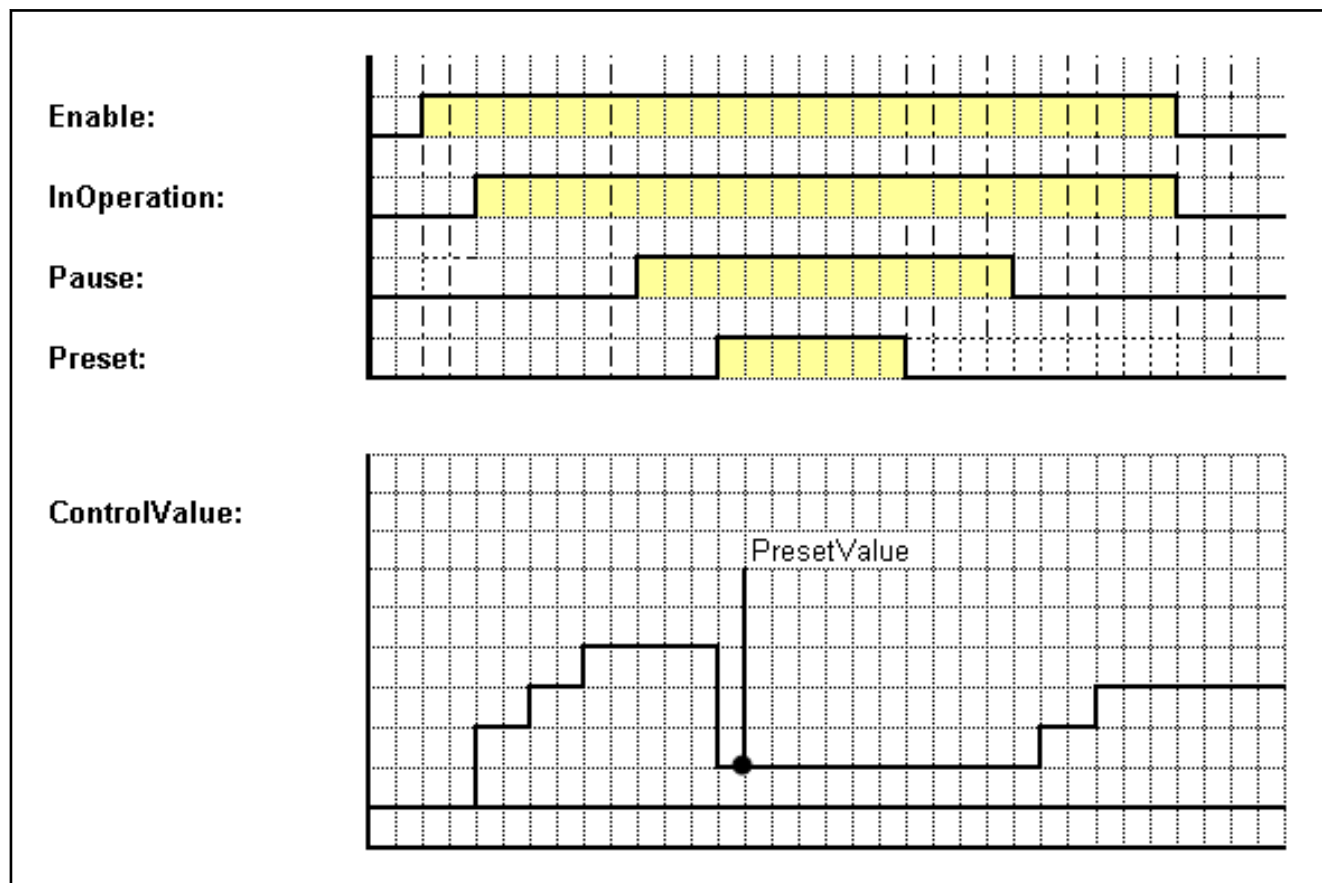


Fig. 5-32: Timing diagram of the MB_RegisterControllerType02 function block with the functions "Pause" and "Preset"

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 must provide a 24V signal with a rise time in 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-0-0694 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

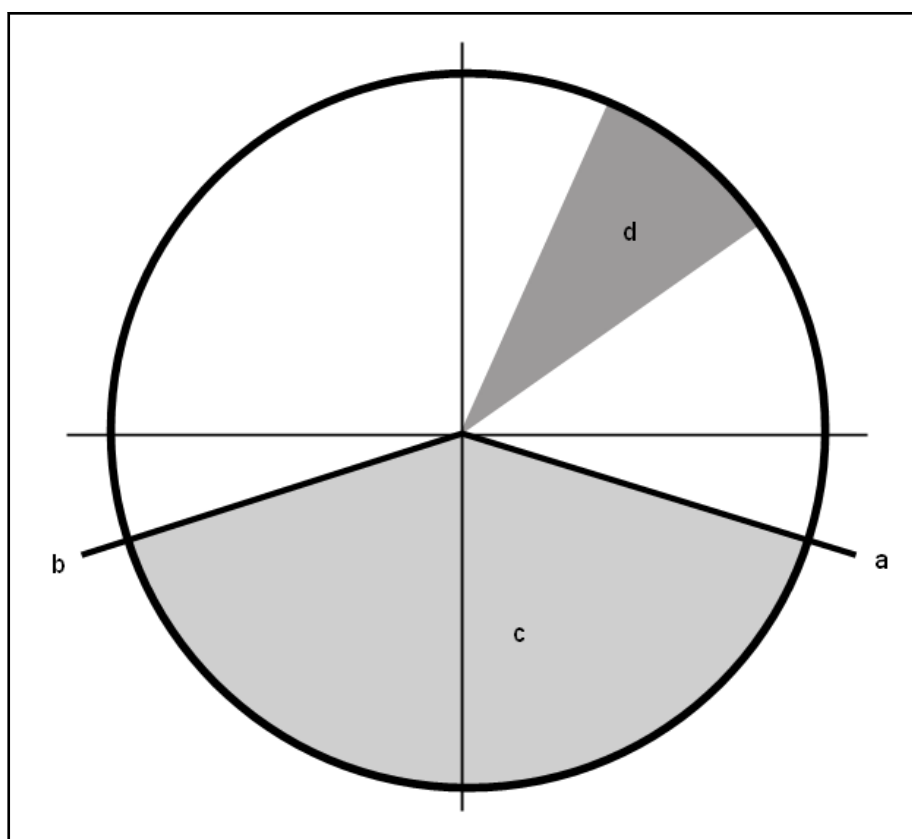
The following "ControlledValueIDN" values are integrated:

- P-0-0061, A-0-2730

ML_TechRegi.library/MX_TechRegi.lib

	• 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 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" 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.

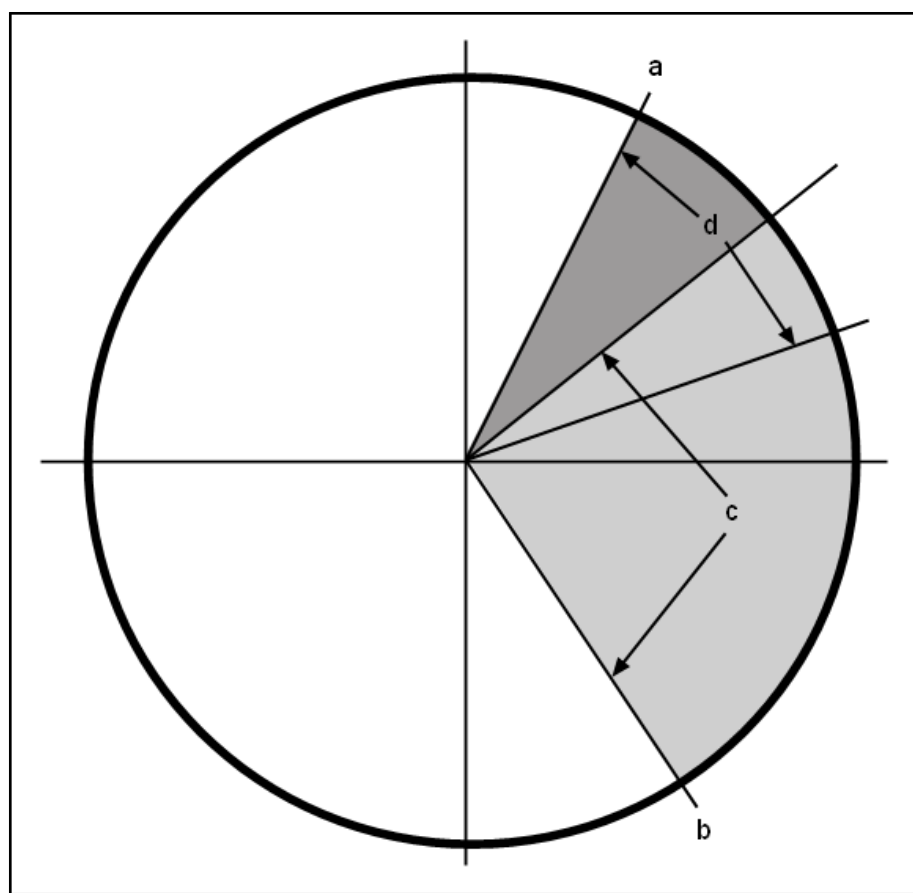
ML_TechRegi.library/MX_TechRegi.lib



- a:** Start correction window
- b:** End correction window
- c:** Correction Window
- d:** Expectation window

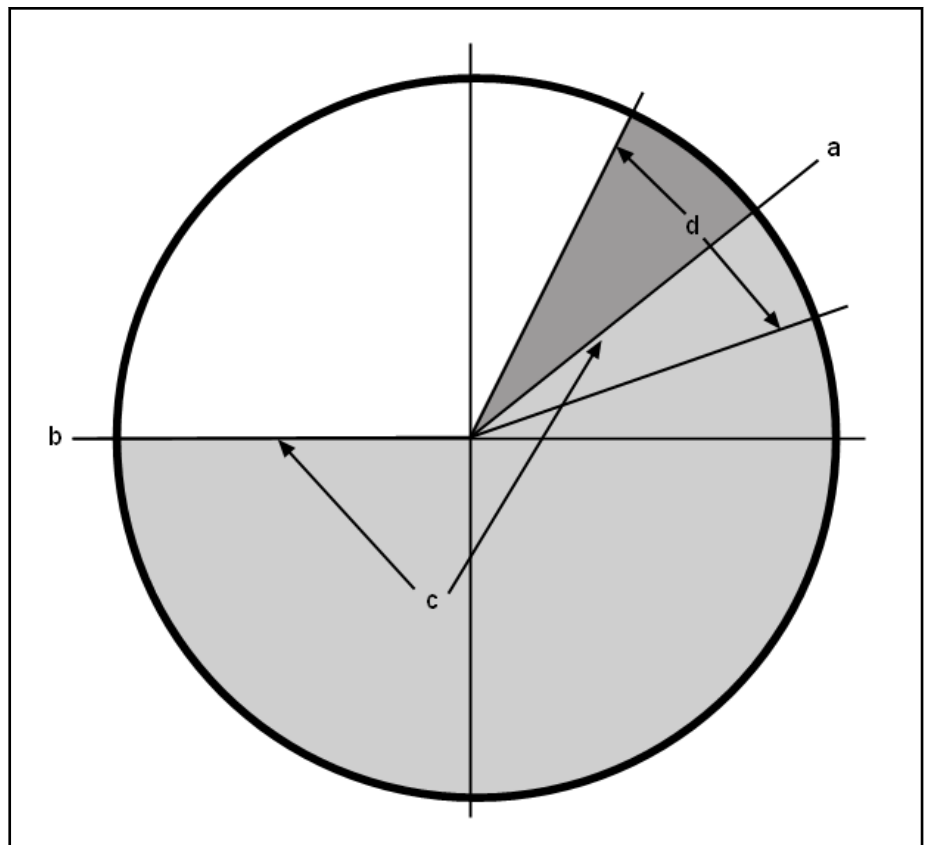
Fig. 5-34: Correction Window Absolute

ML_TechRegi.library/MX_TechRegi.lib



- 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

IndraMotion 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.

ML_TechRegi.library/MX_TechRegi.lib

If a **real axis with interpolation in the control** is used or a **virtual axis** is used as controlled 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

IndraMotion 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

Troubleshooting The function block outputs the error messages of the following function blocks used.

- MB_RegiMeasuringType01 (see [Error codes of the MB_RegiMeasuring-Type01 function block, page 306](#))
- Error codes of the IL_PIType02, RIL_LoopControl function block
Library RIL_LoopControl (refer to the documentation "Rexroth IndraWorks 13VRS Basic Libraries IndraLogic 2G", DOK-IL*2G*-BAS-LIB**V13-LIRS-EN-P, R911336285)
- MB_RegiRegulateType02 (see [Error codes of the MB_RegiRegulate-Type02 function block, page 328](#))
- MB_RegiSetpointLockType01 (see [Error codes of the MB_RegiSetpointLockType01 function block, page 346](#))

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

ML_TechRegi.library/MX_TechRegi.lib

Target system	Library
IndraMotion MLC 13VRS	ML_TechRegi
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-45: Reference table of the MB_RegisterControllerType04

Interface description

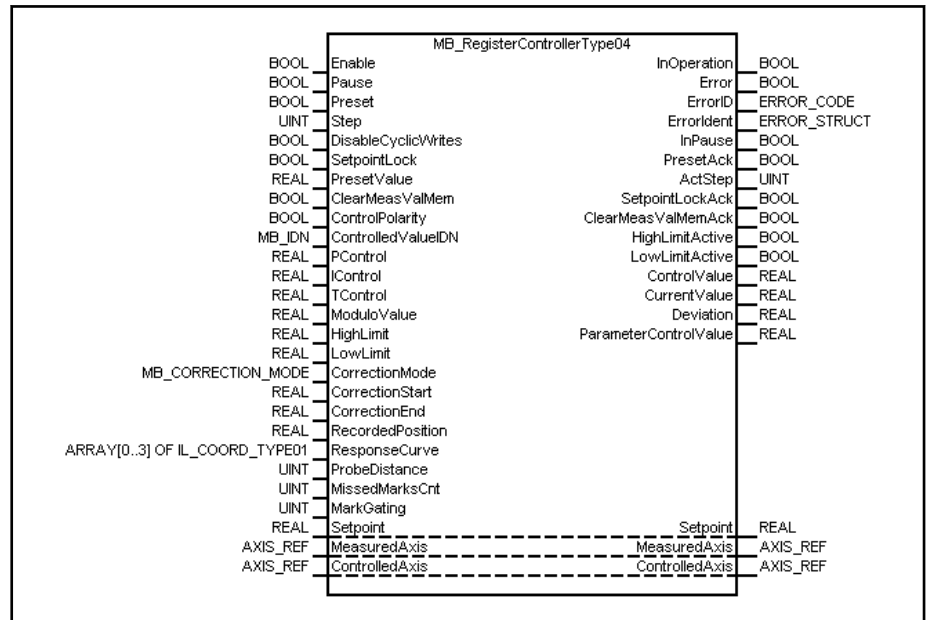


Fig. 5-37: MB_RegisterControllerType04 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Setpoint	REAL	Command value. This value is compared to "RecordedPosition"
	MeasuredAxis	AXIS_REF	Reference to the measured axis
	ControlledAxis	AXIS_REF	Reference to the controlled axis
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

ML_TechRegi.library/MX_TechRegi.lib

I/O type	Name	Data type	Description
	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: 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
	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 343
	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 Troubleshooting, page 368
	ErrorIdent	ERROR_STRUCT	Detailed error description
	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

I/O type	Name	Data type	Description
	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

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
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

ML_TechRegi.library/MX_TechRegi.lib

Name	Type	Min. value	Max. value	Default value	Effective
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	n.def.	n.def.	0.0	Continuous
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.

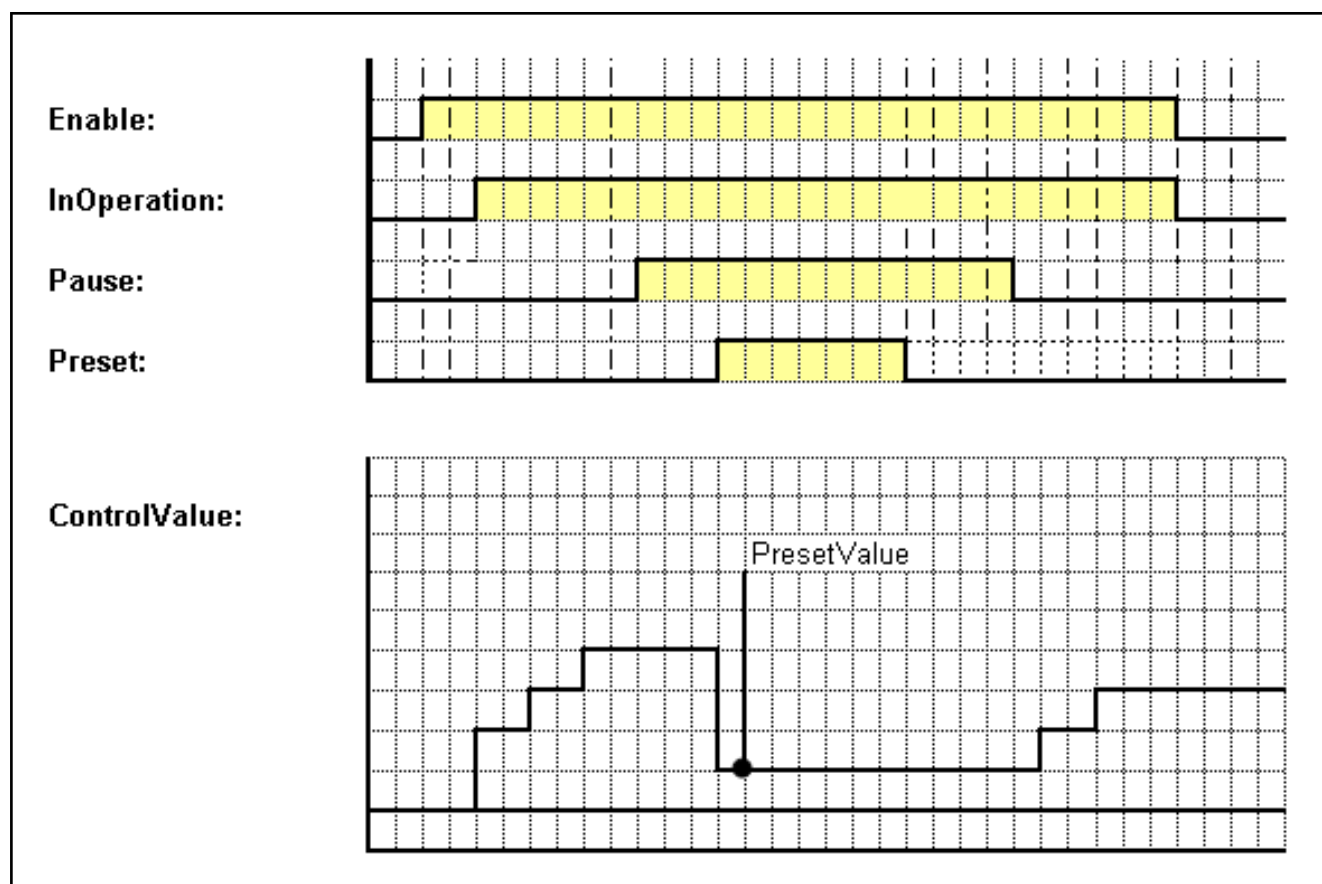


Fig. 5-38: Timing diagram of the MB_RegisterControllerType04 function block with the functions "Pause" and "Preset"

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 must provide a 24V signal with a rise time in 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.

ML_TechRegi.library/MX_TechRegi.lib

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-0-0694 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 "ControlValueIDN" 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 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" 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:

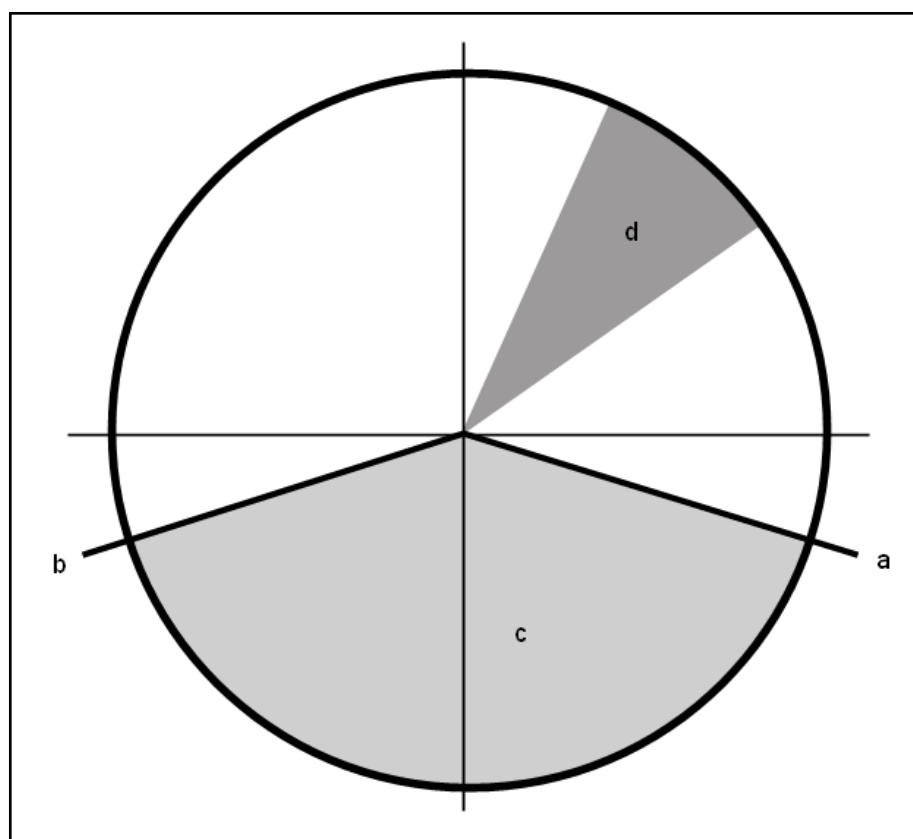
The "CorrectionMode" input sets the correction window either absolute or relative. If the "CorrectionMode" is set to CORR_WIN_DISABLED, the correction window is inactive.

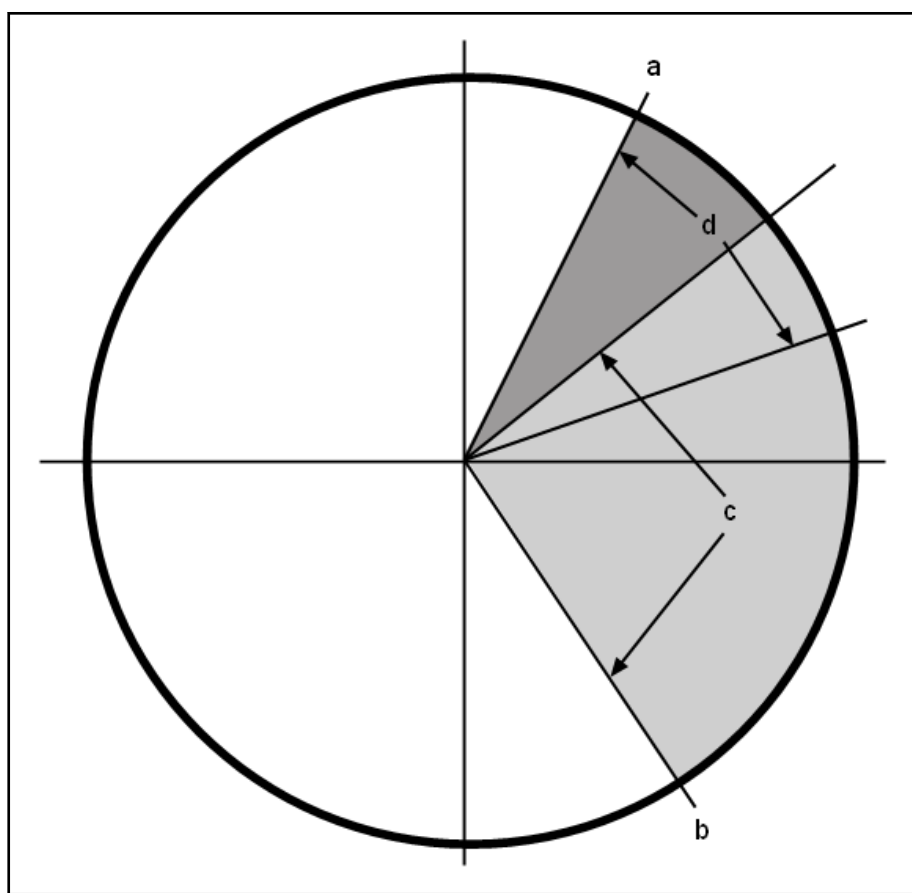
The "CorrectionStart" input defines the start of the correction window.

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.

ML_TechRegi.library/MX_TechRegi.lib

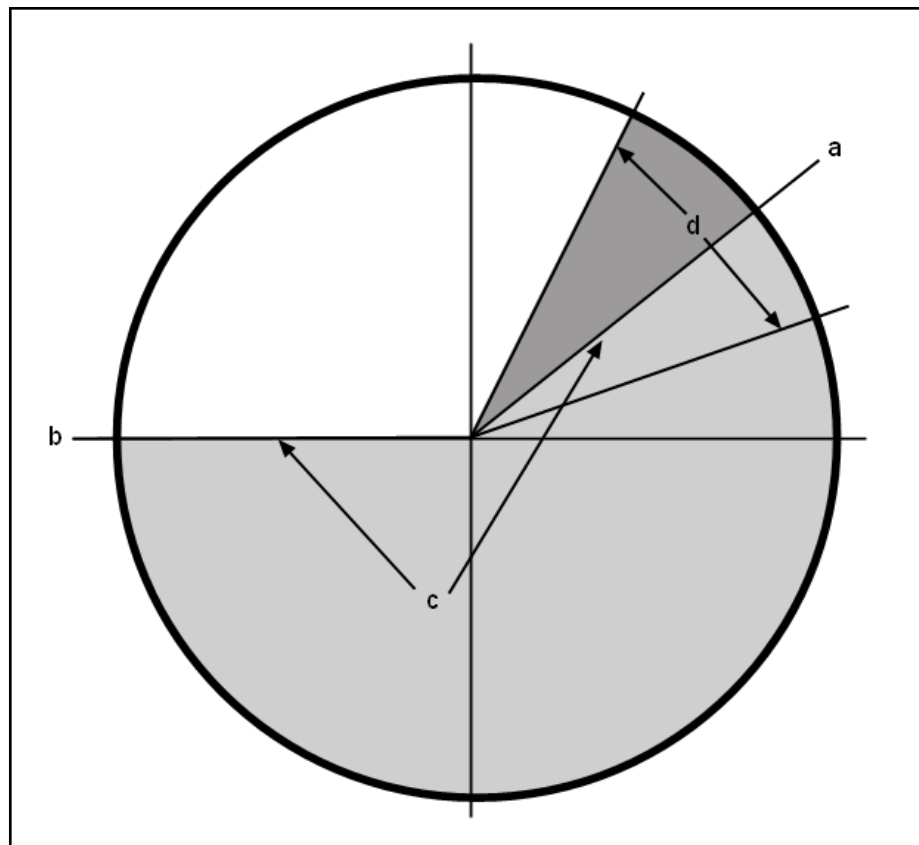
**a:** Start correction window**b:** End correction window**c:** Correction Window**d:** Expectation window*Fig. 5-40: Correction Window Absolute*



- 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

ML_TechRegi.library/MX_TechRegi.lib



- 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 (= controlled) 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.

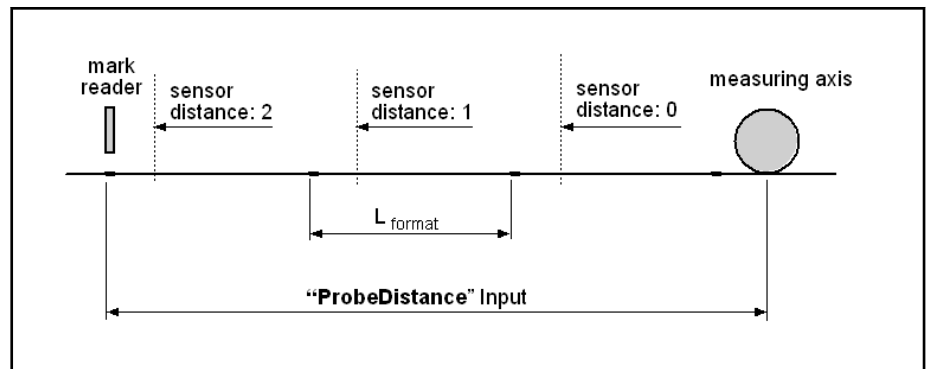


Fig. 5-43: Probe Distance

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.

DeadBand

The differential signal deviation from the filtered "RecordedPosition" input and the "Setpoint" input is scaled via the "ResponseCurve" input. A characteristic curve is used in order to pause the register controller within small register errors.

MarkGating

This feature allows the user to control the registration only every N-th mark by gating invalid marks. This can be used e.g. if the marks on the web were printed with a cliché where the printed image appears several times. In this situation, the mark distances between consecutive marks are not constant. For example, the mark distances between every N-th consecutive mark is constant. The register controller then only uses every N-th mark for the closed loop register control.

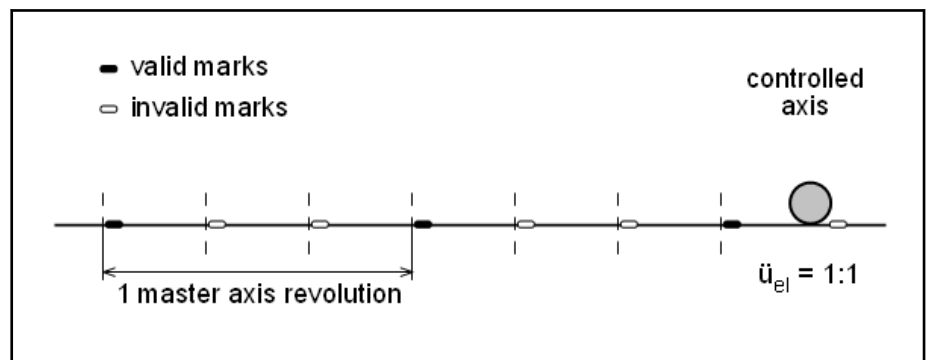


Fig. 5-44: Mark Gating

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.

ML_TechRegi.library/MX_TechRegi.lib

Required Parameterization **IndraMotion 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 or a **virtual axis** is used as controlled 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

IndraMotion 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

Troubleshooting

The function block outputs the error messages of the following function blocks used.

- MB_RegiMeasuringType03 (see [Error codes of the MB_RegiMeasuringType03 function block, page 313](#))
- Error codes of the IL_PIType02, RIL_LoopControl function block
Library RIL_LoopControl (refer to the documentation "Rexroth IndraWorks 13VRS Basic Libraries IndraLogic 2G", DOK-IL*2G*-BAS-LIB**V13-LIRS-EN-P, R911336285)
- MB_RegiRegulateType02 (see [Error codes of the MB_RegiRegulateType02 function block, page 328](#))
- MB_RegiSetpointLockType01 (see [Error codes of the MB_RegiSetpointLockType01 function block, page 346](#))

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

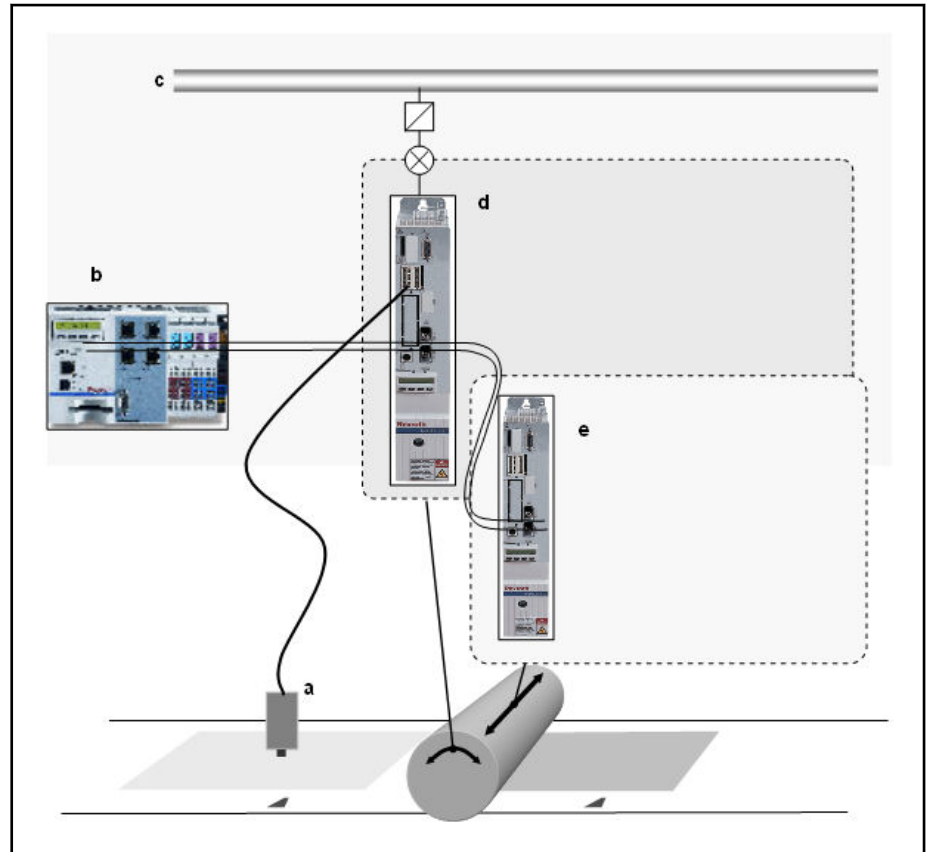
5.6.4 Configuring the Register Controller

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 posi-

tion. 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: IndraMotion MLC control (closed-loop)
c Master axis (virtual axis)
d Register control, touch probe function, phase synchronization

Fig. 5-45: Example of a register controller on an IndraMotion 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

ML_TechRegi.library/MX_TechRegi.lib

3. Marker failure enabled
4. Expected window enabled



For a description of the touch probe function block, refer to [chapter 3.9.3 "MB_InitTouchProbe"](#) on page 106.

Configuring the Touch Probe and RegisterControllerType04 Function Blocks

The following figure shows the connection of the MB_TouchProbeContinuous and MB_RegisterControllerType04 function blocks when triggering the touch probe function using a positive edge. As part of the MB_InitTouchProbe function block configuration, the marker failure and the expectation window are enabled. Thus, the counter for marker failures have to be connected as well.

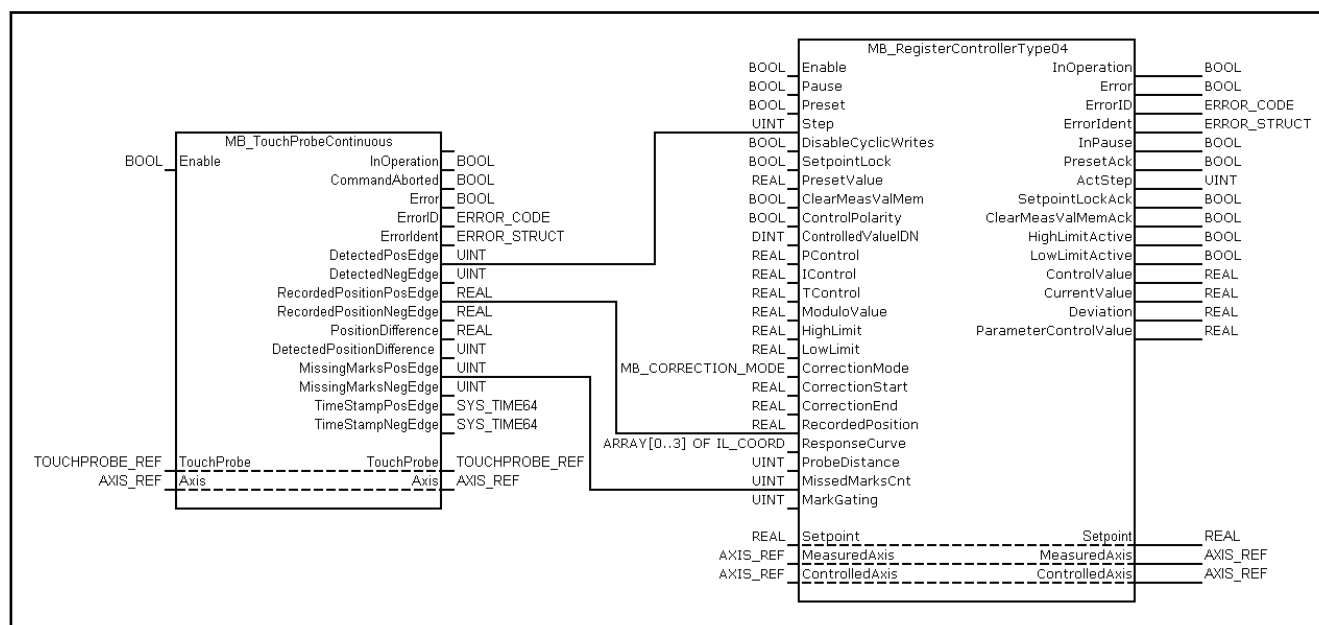


Fig. 5-46: MB_TouchProbeContinuous and MB_RegisterControllerType04 interface

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 IndraMotion MLD, select **IndraWorks** ► **Axis** ► **Scaling/Mechanical System** ► **Scaling/Units**.

- For the IndraMotion MLC, select **IndraWorks** ► **Motion** ► **Virtual Axes/ Real Axes** ► **Axis** ► **Scaling/Mechanical System** ► **Scaling/Units**.

The following figure is an example:

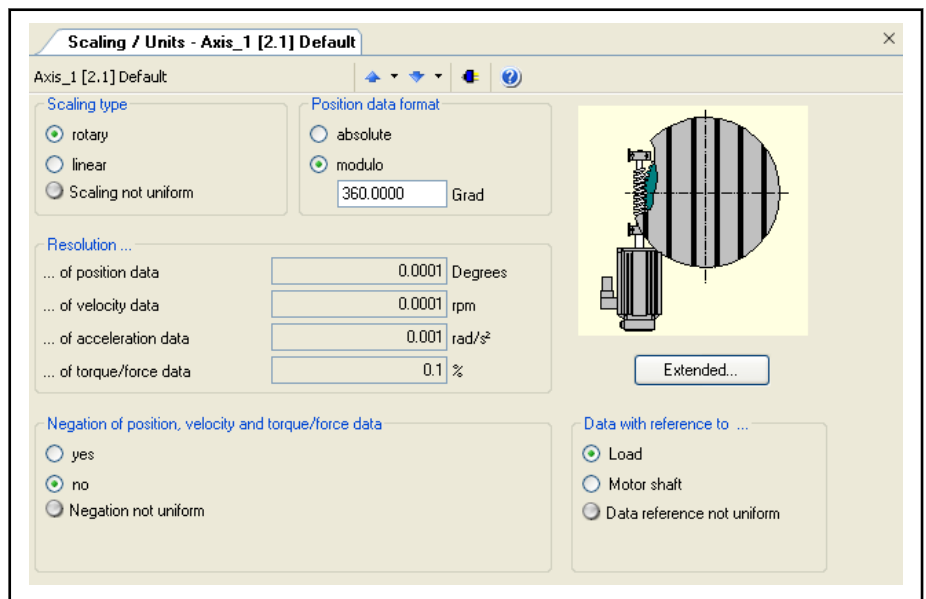


Fig. 5-47: Setting the axis scaling

Configuring the IndraMotion 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:

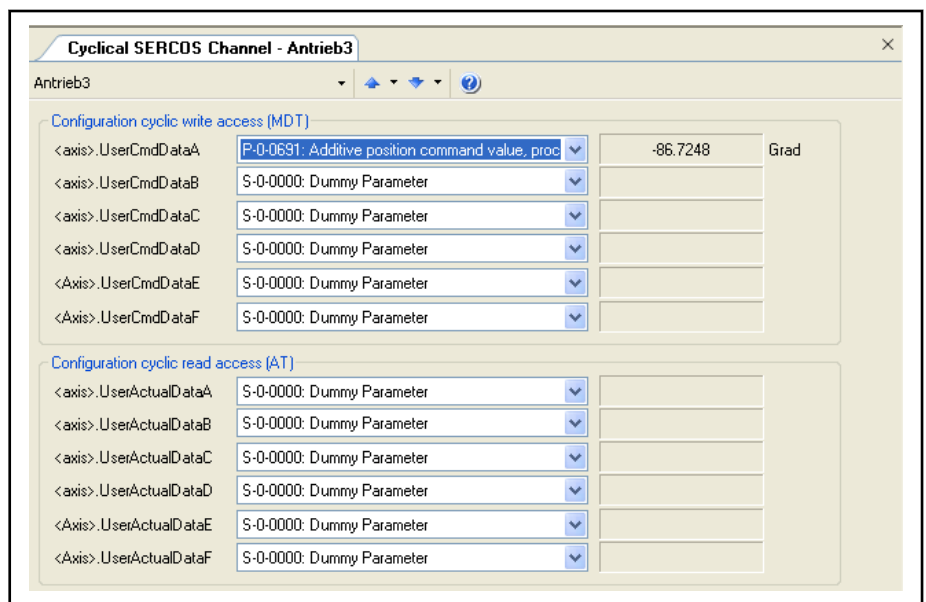


Fig. 5-48: MDT configuration of the controlled axis (IndraMotion MLC)

IndraMotion 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.

ML_TechRegi.library/MX_TechRegi.lib

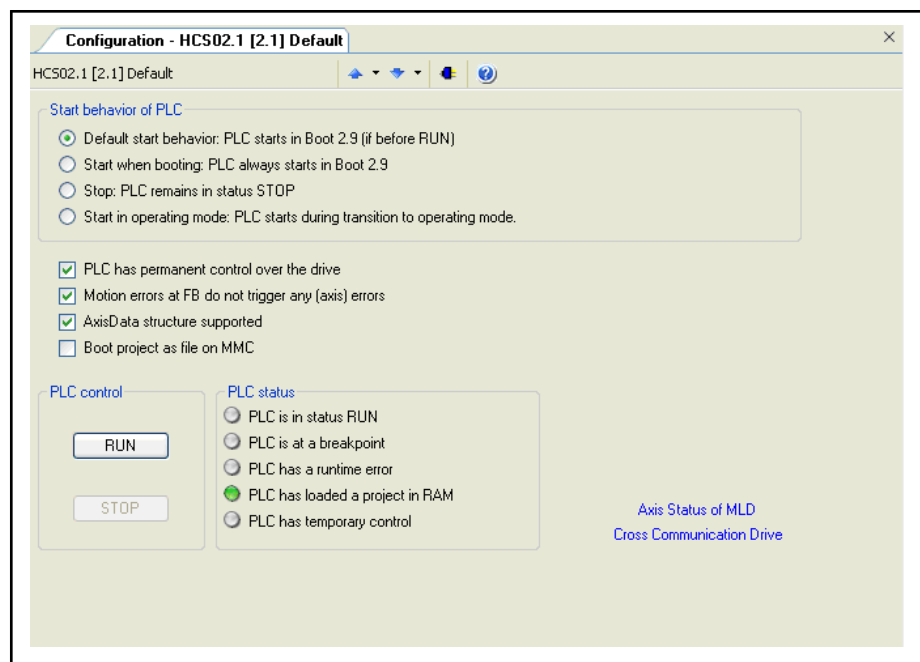


Fig. 5-49: IndraMotion MLD configuration

Then, select **IndraWorks** ► **MLD** ► **Configuration** ► **AxisData** and select the parameter used for the "ControlledValueIDN" input.

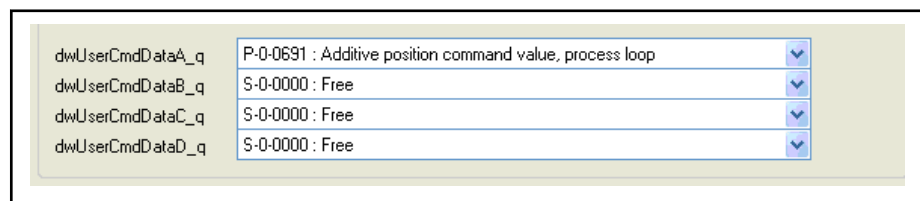


Fig. 5-50: AxisData configuration

5.6.5 Register Controller – Uses Cases

Punching in label printing

The following example describes a punching process on a label printing machine:

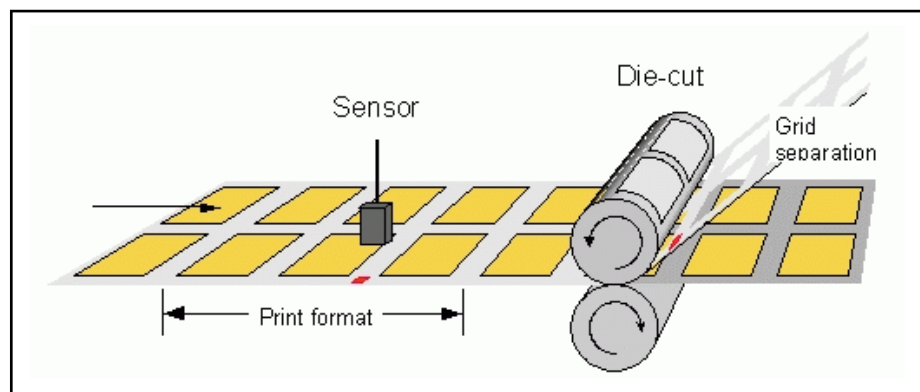


Fig. 5-51: Closed-loop register control of a punching device

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:

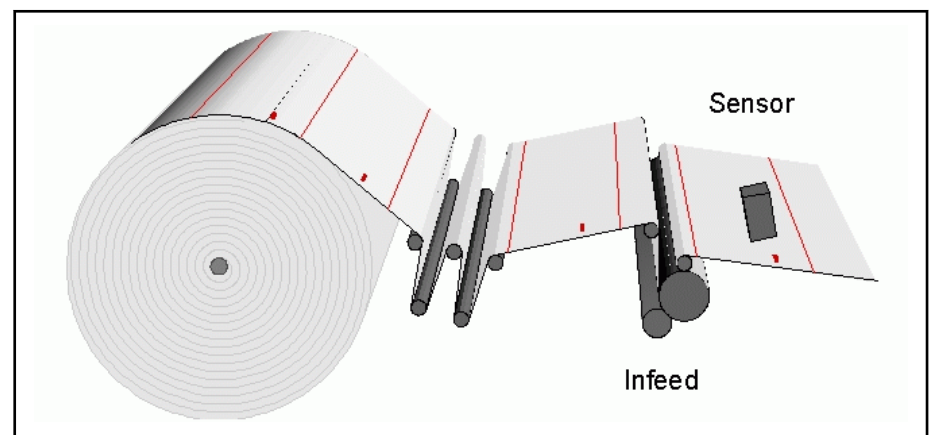


Fig. 5-52: Closed-loop insetter control of the material infeed

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:

ML_TechRegi.library/MX_TechRegi.lib

- 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 sideseal axis (foil infeed axis) of a flow wrapper application:

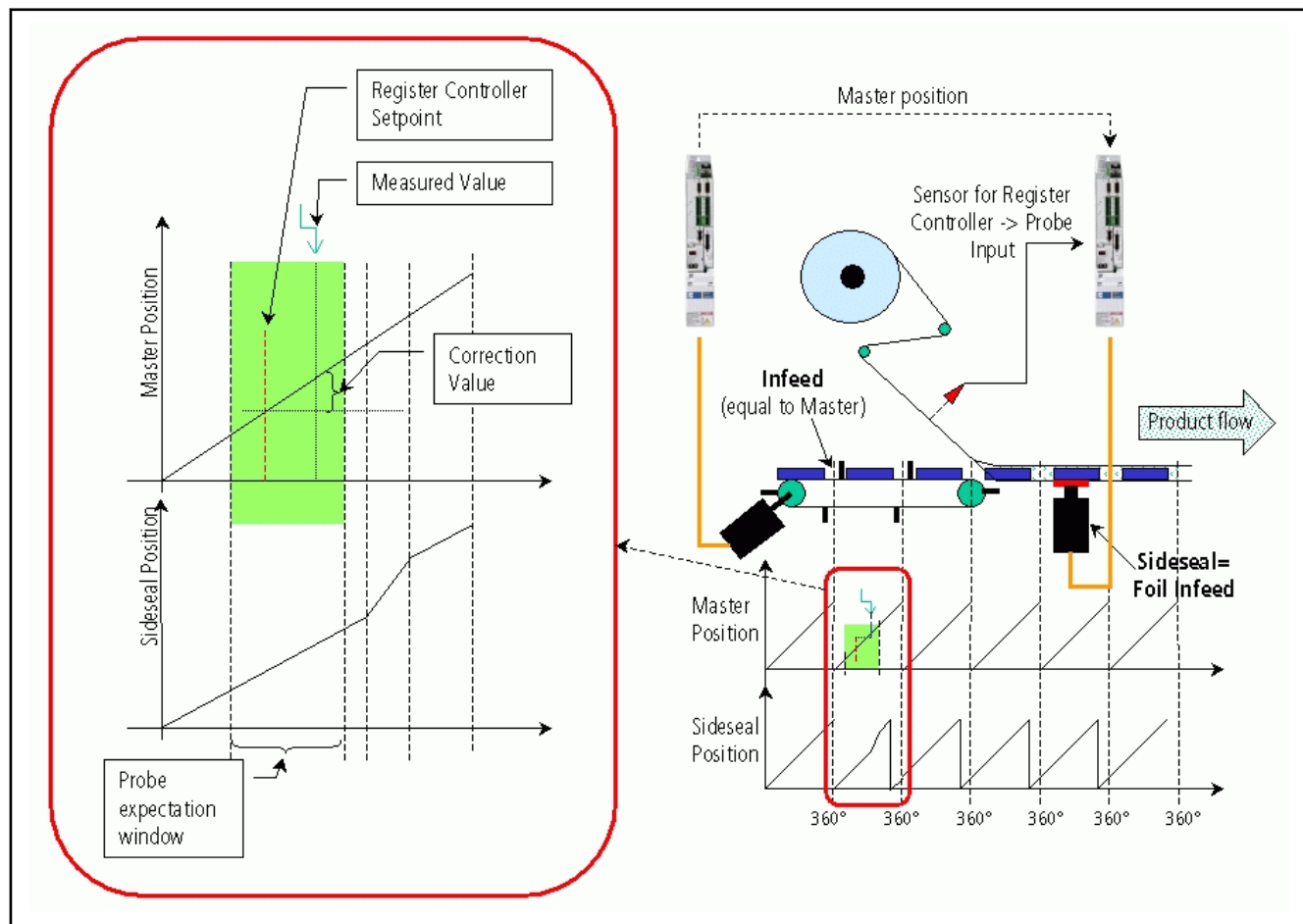


Fig. 5-53: Foil infeed axis of a flow wrapper application

In this application, the product is fed into the packaging machine with the help of an infeed belt. The foil infeed axis (sideseal 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 (IndraMotion MLC) and MX_TechRegi.lib (IndraMotion 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
IndraMotion MLC 13VRS	ML_TechRegi
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechRegi.lib

Tab. 5-49: Reference table of the MB_RegisterControllerSideType01

ML_TechRegi.library/MX_TechRegi.lib

Interface description

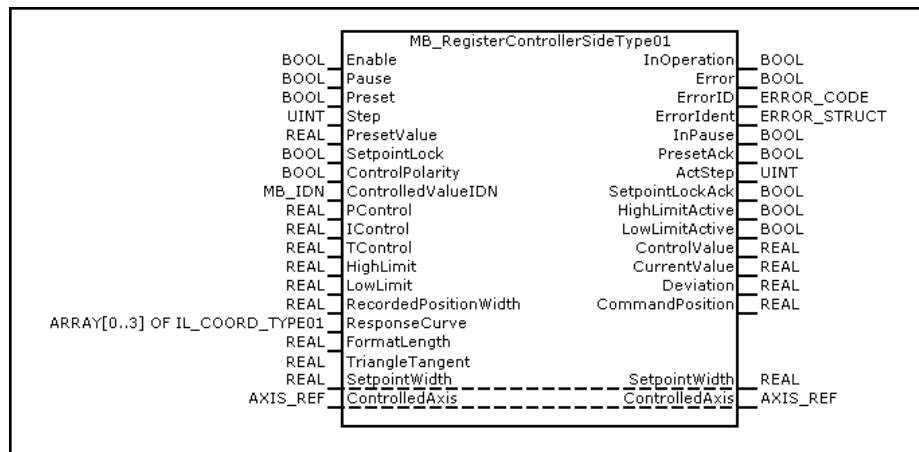


Fig. 5-54: MB_RegisterControllerSideType01 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	SetpointWidth	REAL	Setpoint width value (in technical units)
	ControlledAxis	AXIS_REF	Reference to the controlled axis
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)

I/O type	Name	Data type	Description
	ResponseCurve	ARRAY[0..3] OF IL_COORD_TYPE01	Characteristic curve
	FormatLength	REAL	Format length in mm
	TriangleTangent	REAL	Value of $\tan \alpha$ = (opposite leg length/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 380
	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

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

ML_TechRegi.library/MX_TechRegi.lib

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.

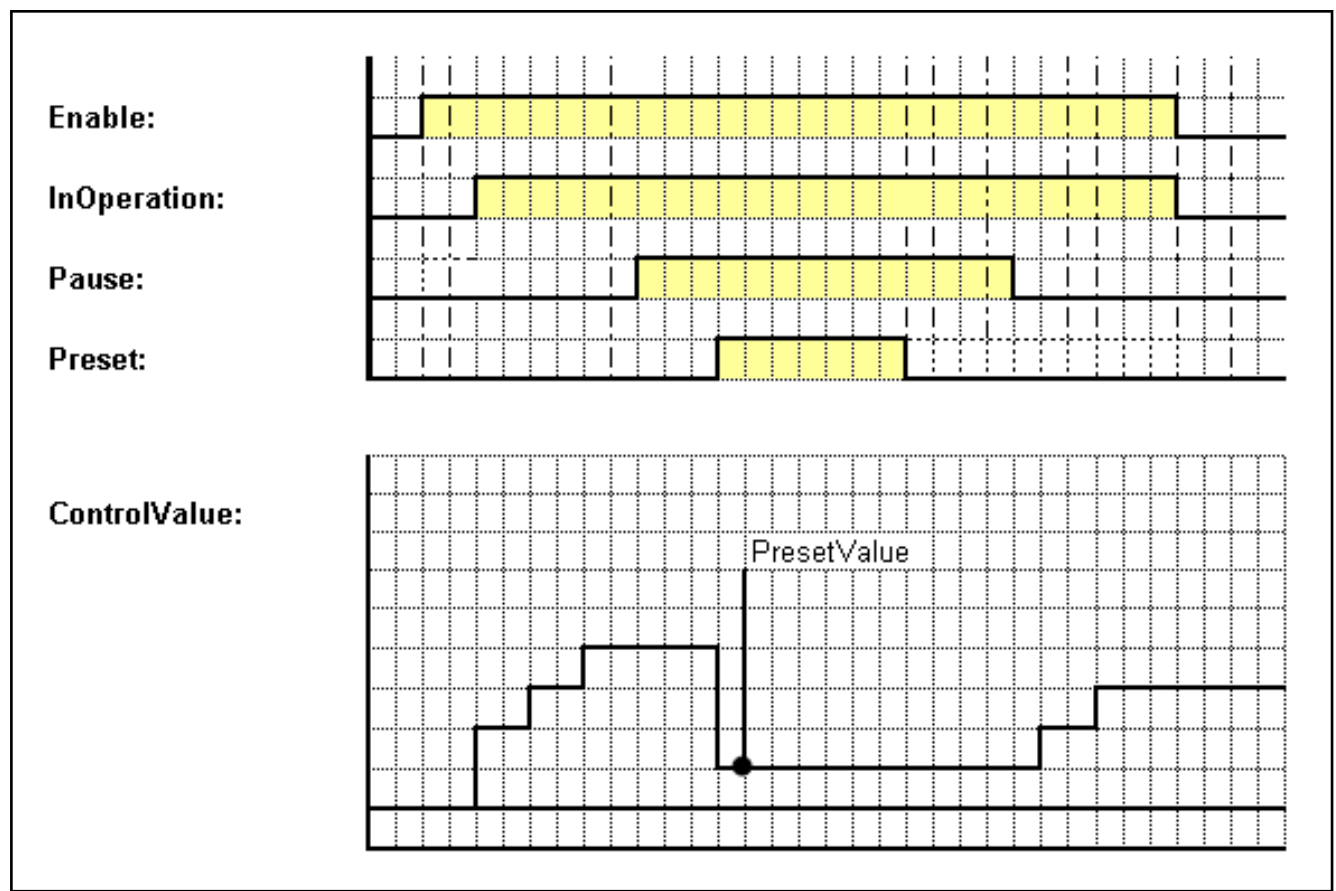


Fig. 5-55: Timing diagram of the MB_RegisterControllerSideType01 with the functions "Pause" and "Preset"

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.

ML_TechRegi.library/MX_TechRegi.lib

- DeadBand** The differential signal of the filtered "RecordedPositionWidth" input and of the "SetpointWidth" input are scaled via the "ResponseCurve" input. A characteristic curve pauses the register controller in small register errors.
- FormatLength** Format length (master axis revolutions, formats) in mm.
- TriangleTangent** The mark length between opposite leg and adjacent leg ($\tan \alpha$).

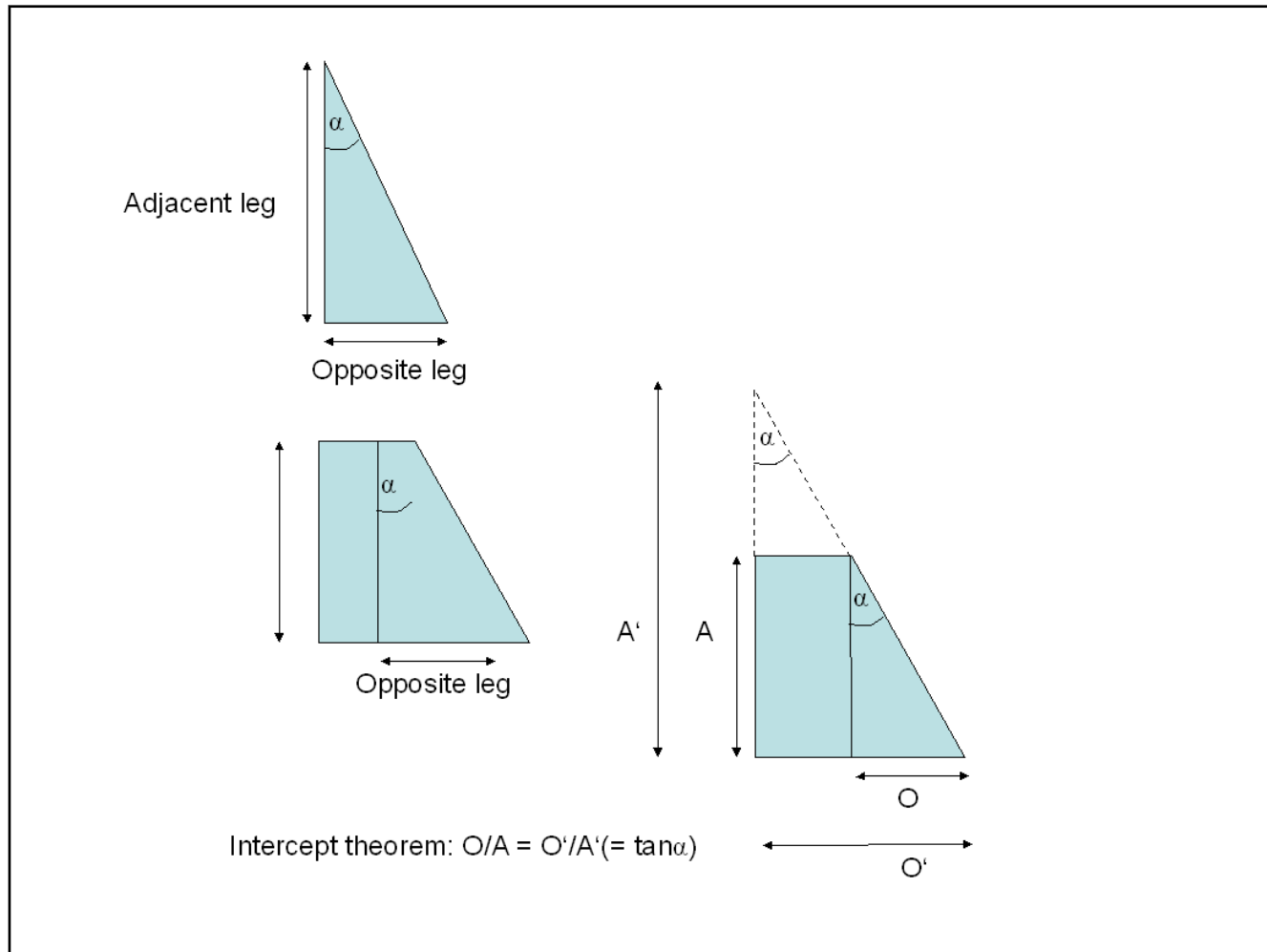


Fig. 5-56: Right triangle mark and conical frustum mark

- 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
- MB_RegiMeasuringType01 (see [Error codes of the MB_RegiMeasuring-Type01 function block, page 306](#))

- Error codes of the IL_PIType02, RIL_LoopControl function block
Library RIL_LoopControl (refer to the documentation "Rexroth IndraWorks 13VRS Basic Libraries IndraLogic 2G", DOK-IL*2G*-BAS-LIB**V13-LIRS-EN-P, R911336285)
- MB_RegiRegulateType01 (see [MB_RegiRegulateType01](#), page 317)
- MB_RegiSetpointLockType01 (see [Error codes of the MB_RegiSetpointLockType01 function block](#), page 346)

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
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"

ML_TechRegi.library/MX_TechRegi.lib

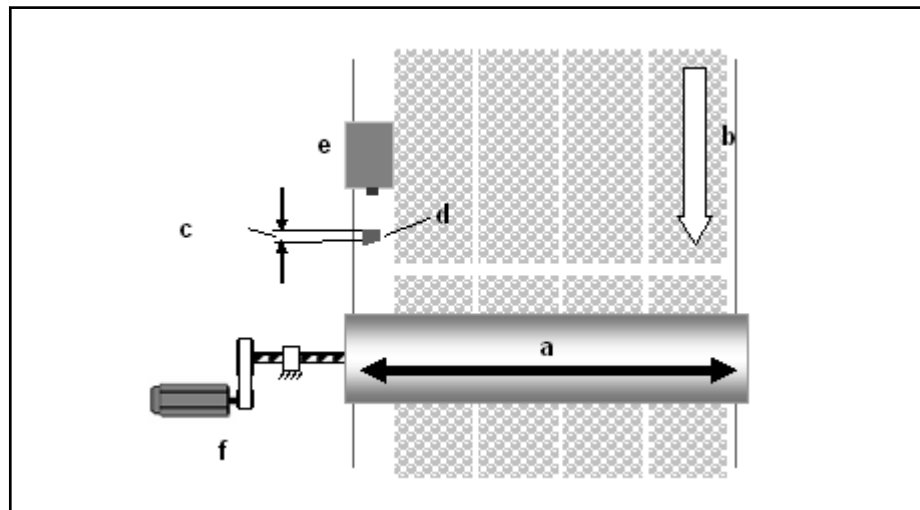
ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR (16#0001)	16#0C05	16#0003	Negative x-value in "Response-Curve"
ACCESS_ERROR (16#0004)	16#0815	16#0008	The controlled axis is an IndraMotion MLD-M slave axis, but the parameter P-0-1367, bit 6 is not TRUE

Tab. 5-52: Error codes of MB_RegisterControllerSideType01 function block

5.7.3 Configuring the Side Register

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.

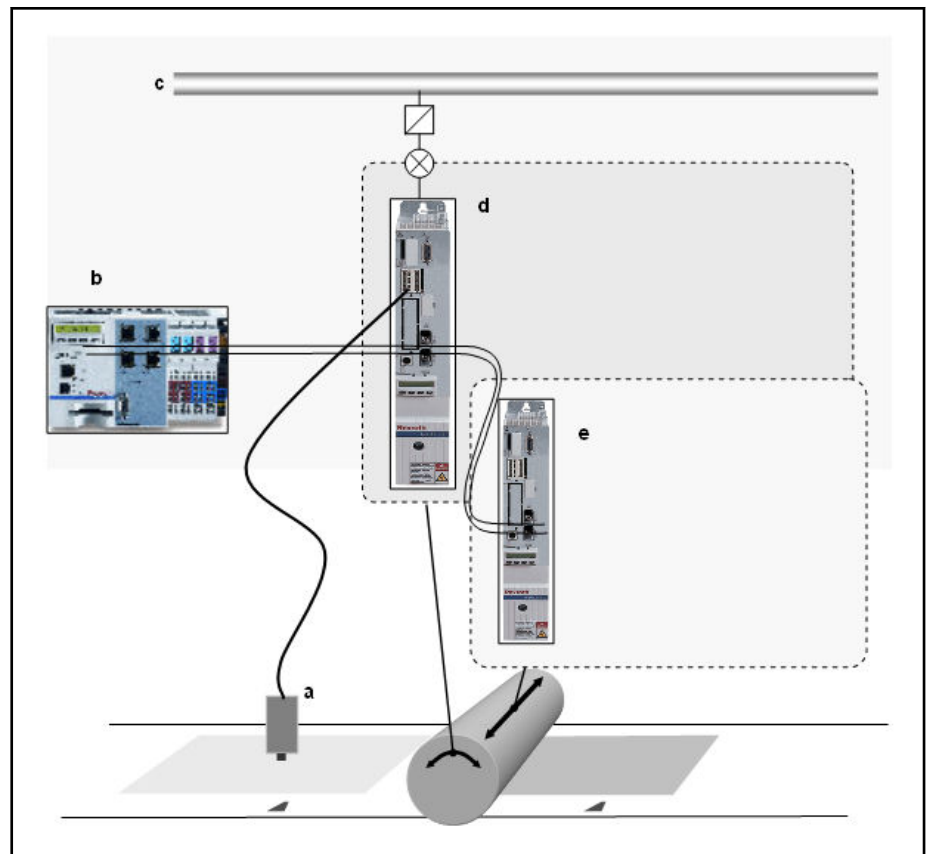


- a: Side register
- b: Web direction
- c: Measured value
- d: Registration mark
- e: Sensor
- f: Servo motor

Fig. 5-57: Side Register Configuration

Using Side Register Control

The following picture shows a possible configuration of a side register controller for an IndraMotion MLC system.

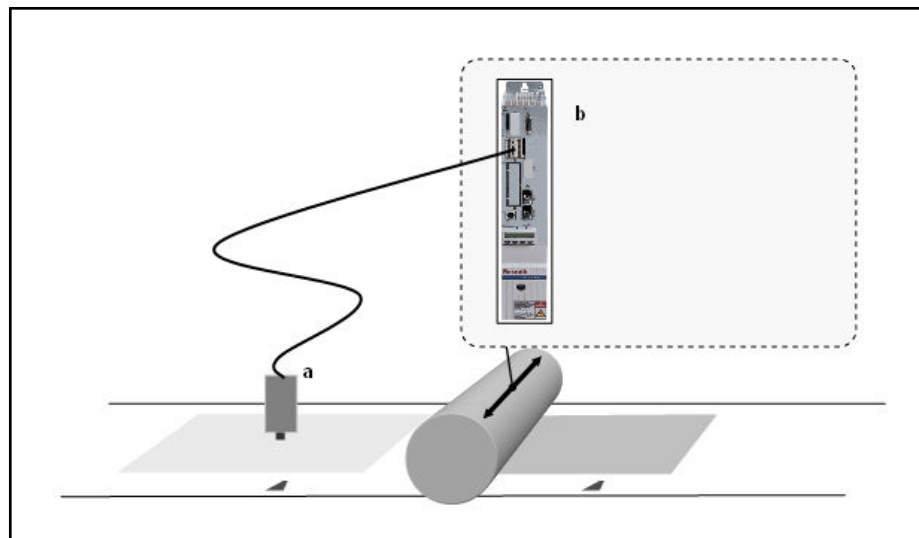


- a:** Sensor
- b:** IndraMotion MLC
- c:** Master axis
- d:** Register controller, touch probe function, phase synchronization
- e:** Side register controller, absolute positioning

Fig. 5-58: Register controller and side register controller overview on the IndraMotion MLC

The following figure shows the configuration of a side register controller for an IndraMotion MLD system.

ML_TechRegi.library/MX_TechRegi.lib



- a: Sensor
b: IndraMotion MLD: probe function, side register controller, absolute positioning

Fig. 5-59: Side register controller overview on IndraMotion MLD



The axial axis has to be an absolute linear axis.

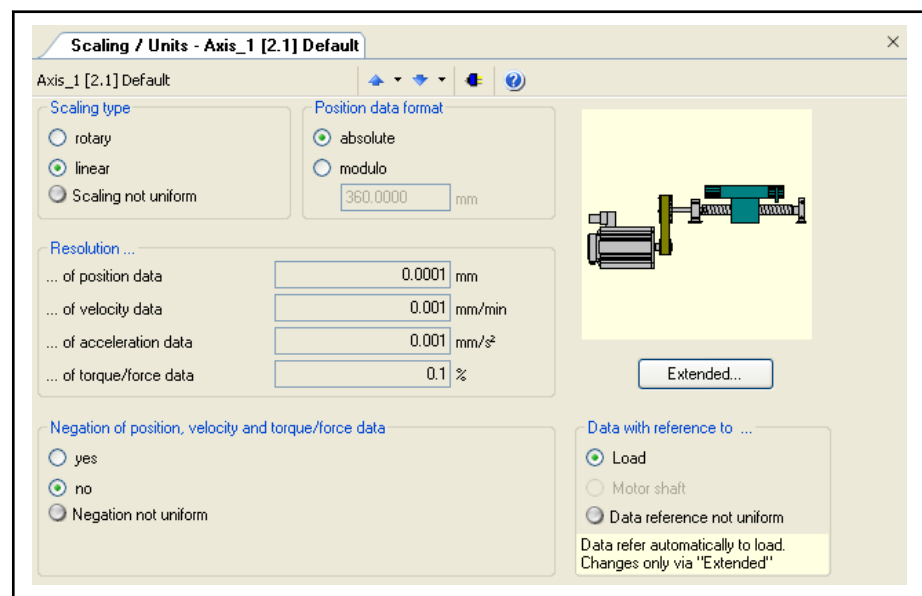


Fig. 5-60: Scaling of the controlled axis of the side register controller

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 demonstrates the MB_RegisterControllerSideType01 side register controller with the MB_TouchProbeContinuous function block and the AxisInterface function blocks.

1. Output "PositionDifference" to input the "RecordedPositionWidth"

2. Output "DetectedPositionDifference" to input "Step"

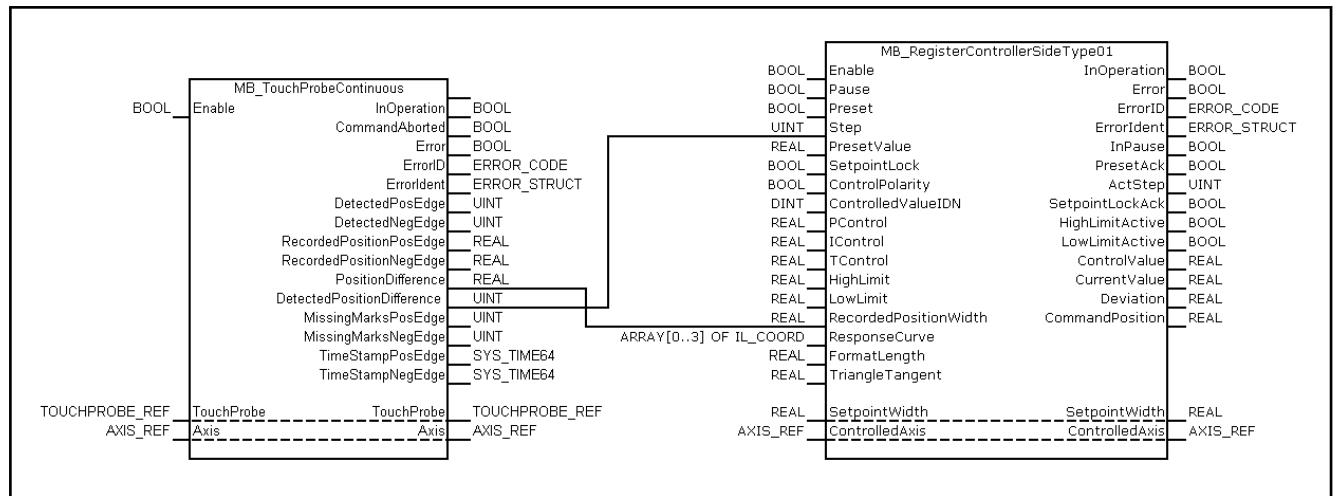


Fig. 5-61: Schematic representation of the function blocks MB_TouchProbeContinuous and MB_RegisterControllerSideType01

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 "Rexroth IndraMotion 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 *)

```


ML_TechRegi.library/MX_TechRegi.lib

```
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 The library ML_TechCrank.library contains functionalities for crank and bell-crank kinematics.

The function blocks of the ML_TechCrank

Function block	Description
Bell-crank kinematics (BellCrank)	
BellCrank\ Basics	
MB_BellCrankData, page 394	Calculates the y-positions and crank angles Phi at the top and bottom pole
MB_YvirtToYmech, page 411	Converts the virtual translatory position to the mechanical position
BellCrank\ CamCalculation	
MB_CamTableBellCrank, page 397	Calculates a cam for transformation for a bell-crank drive
MB_CamTableBellCrankSuperimposed, page 400	Superimposes a user cam with the transformation can for a superimposed cam.
BellCrank\ Trafos	
MB_PhiToYvirt, page 405	Converts the crank angle Phi to a mechanical and virtual translatory position
MB_YvirtToPhi, page 407	Converts the virtual translatory position to the crank angle Phi
MB_YvirtToPhiPoly5, page 409	Converts the virtual translatory position to the crank angle Phi using a substitute polynomial 4th order
MB_YVirtToWorkRange, page 396	Returns the working area of crank 1

ML_TechCrank.library

Function block	Description
Crank kinematics (Crank)	
MB_CRANK, page 418	The structure is responsible for internal data exchange between function blocks and functions.
Crank\ Crank_Cam	
MB_CamTableCrank, page 418	Calculates the cam for transforming a translatory virtual master axis position into rotary values (crank angle)
MB_CamTableCrankSuperimposed, page 422	Superimposes a user cam with the transformation can for a superimposed cam.
Crank\ Crank_Trafos	
MB_MasterToPhi, page 427	Outputs crank angles calculated using the master axis position and the superimposed cam
MB_PhiToXvirt, page 428	Converts crank angle (Phi) into virtual translatory position values (Xvirt) to be used by the PLC program
MB_XvirtToXmech, page 429	Converts virtual translatory position values (Xvirt) to mechanic translatory position values (Xmech)

Tab. 6-1: ML_TechCrank.library

6.2 Functions and function blocks for the bell crank kinematics (BellCrank)

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.

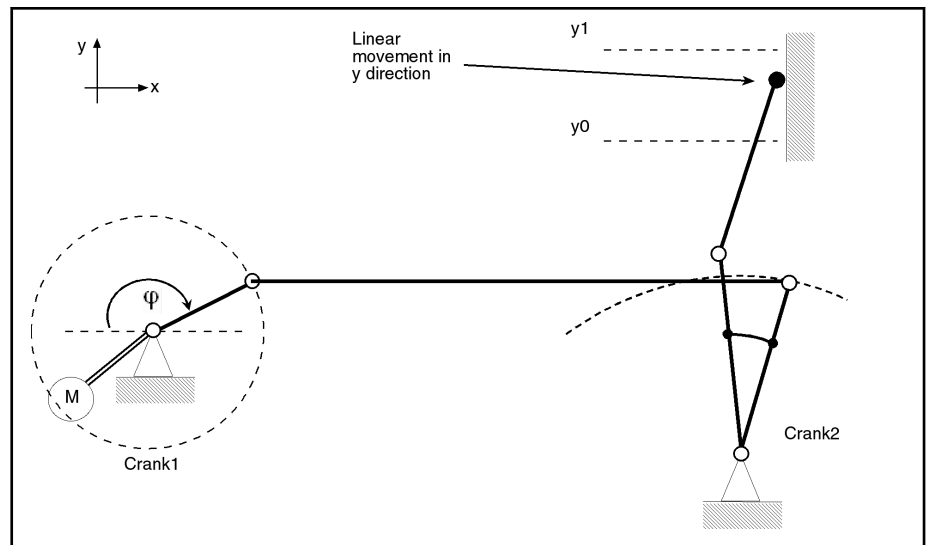


Fig. 6-1: Bell crank kinematics

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.

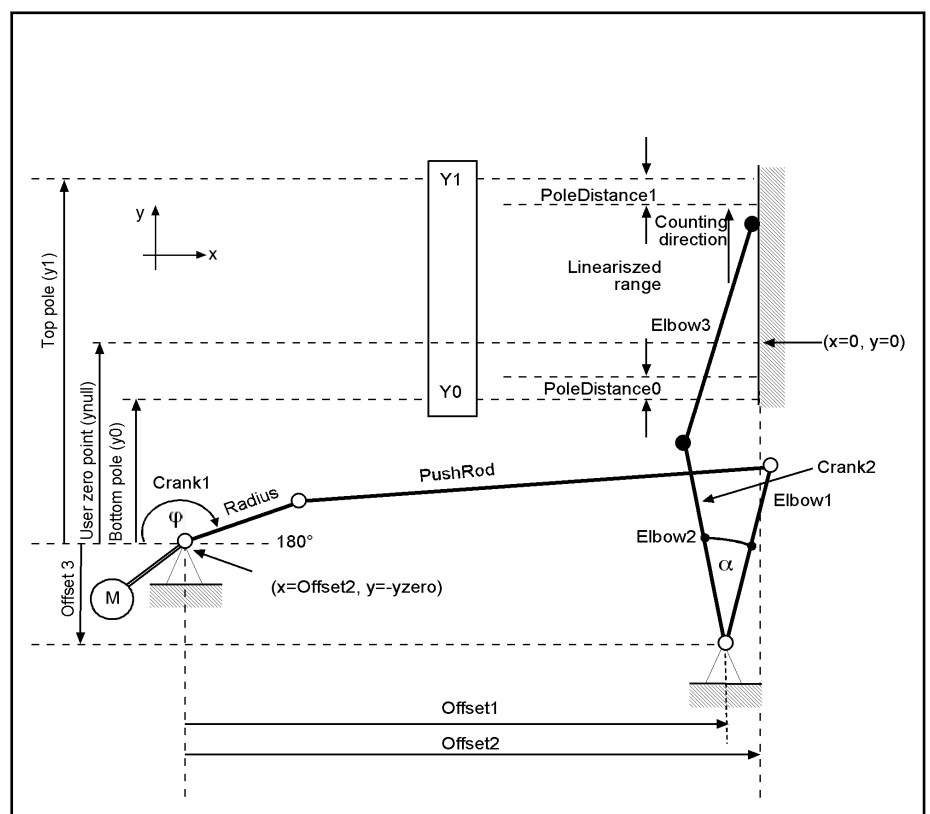


Fig. 6-2: Relation between crank and translatory coordinates at the bell-crank kinematics

Counting Direction

Basically, the following counting directions are applicable to the arrangement shown in [fig. 6-2 "Relation between crank and translatory coordinates at the bell-crank kinematics"](#) on page 391:

ML_TechCrank.library

- 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 travel range (S_v) is defined as distance between top (Y_1) and bottom (Y_0) pole.

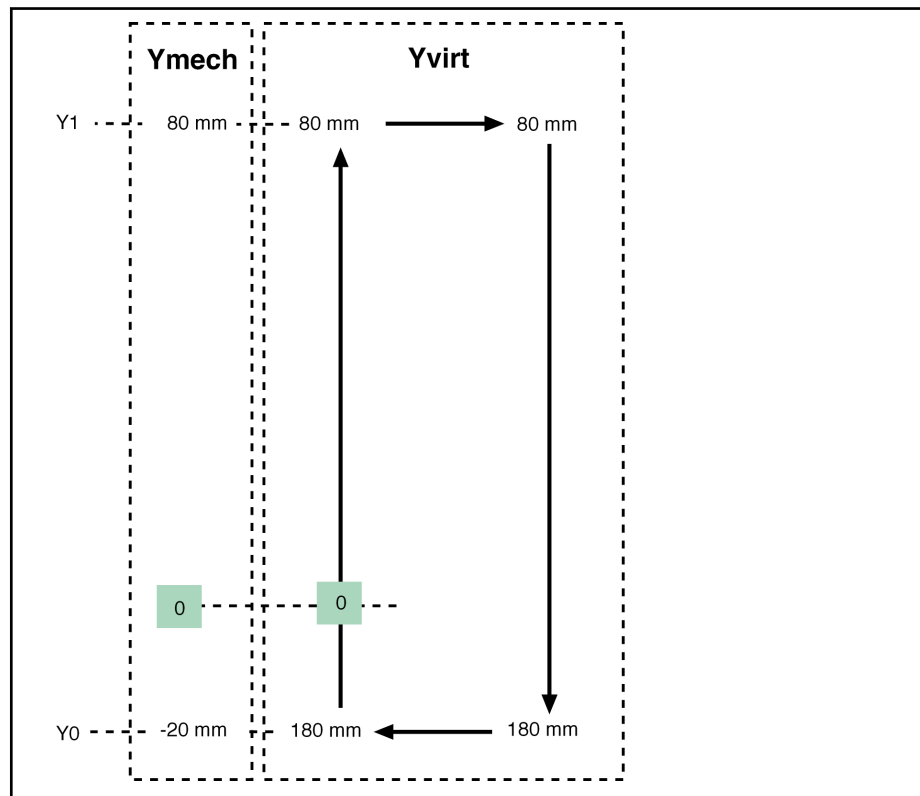


Fig. 6-3: Relation between Y_{virt} and Y_{mech}

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 IndraMotion MLD, the virtual axis does not support any synchronous operation modes. That is why this use case is only reasonable to a limited extent.

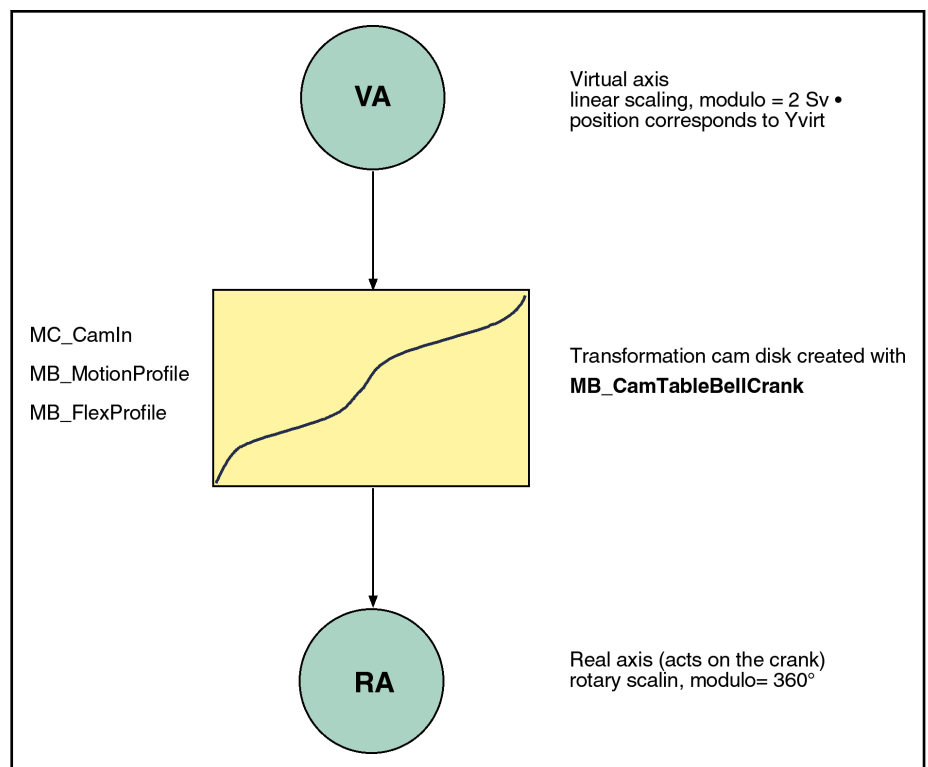


Fig. 6-4: Mode of operation of the transformation cam with a translatable virtual axis

- 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 IndraMotion MLD, as the bell-crank drive can be coupled to a master axis. For the target system IndraMotion MLC this use case example has a lower significance. However, an additional virtual axis is not required.

ML_TechCrank.library

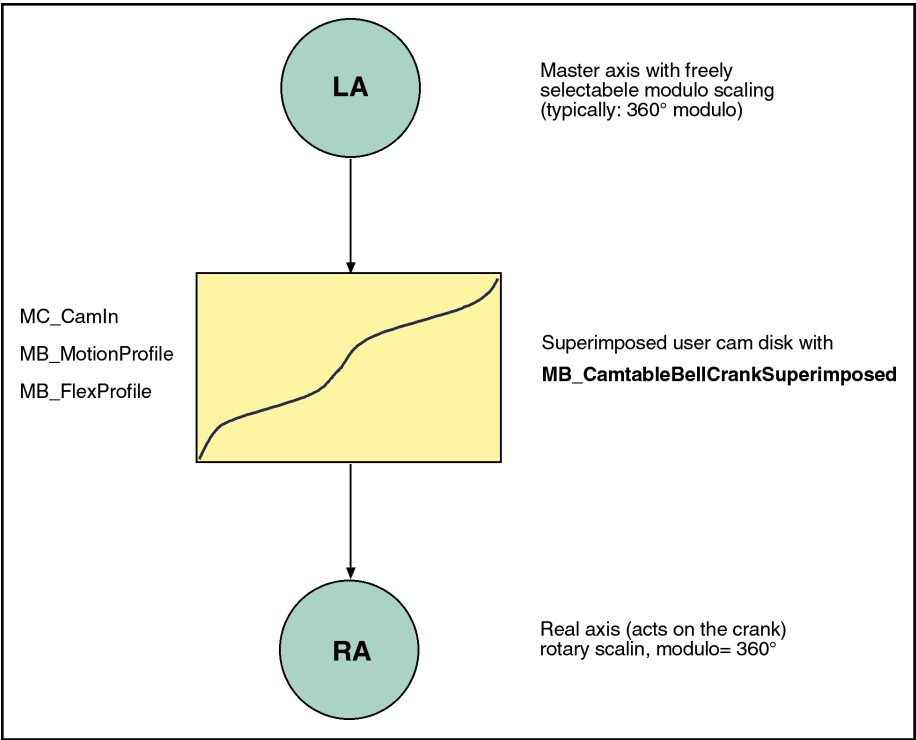


Fig. 6-5: Mode of operation of the superimposed to any master axis

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 range 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 ranges 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	Library
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion 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

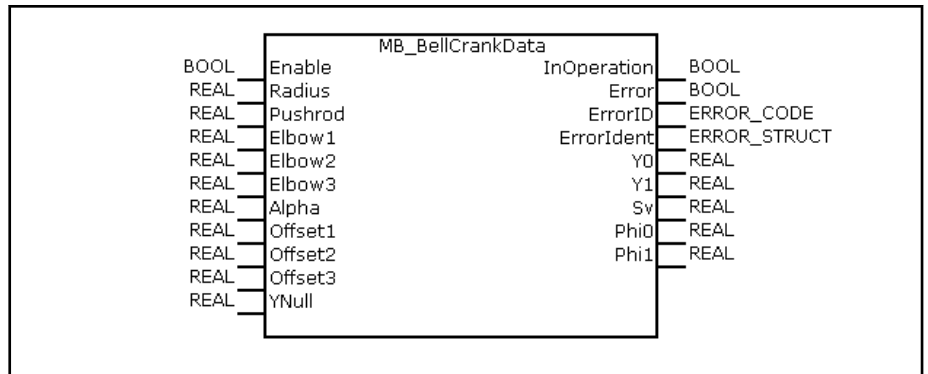


Fig. 6-6: MB_BellCrankData function block

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 ErrorIdent)
	ErrorID	ERROR_CODE	Error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 395
	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

Troubleshooting The function block generates the following error messages in Additional1/ Additional2 of the table "F_RELATED_TABLE", 16#0170:

ML_TechCrank.library

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_YVirtToWorkRange

Functional Description The function MB_YVirtToWorkRange provides the working areas(1 or 2) of crank 1 as return value resulting from the virtual translatory position.

In working area 1 the mechanical position of the y-axis moves in positive direction if crank 1 rotates clockwise.

In working area 2 the mechanical position of the y-axis moves in negative direction if crank 1 rotates clockwise. Internally, this function is used by other function blocks.

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion MLD with MPx08 and MLD with MPx17 with MA function package	MX_Packaging_07.lib

Tab. 6-5: Reference table of the MB_YVirtToWorkRange

Interface description

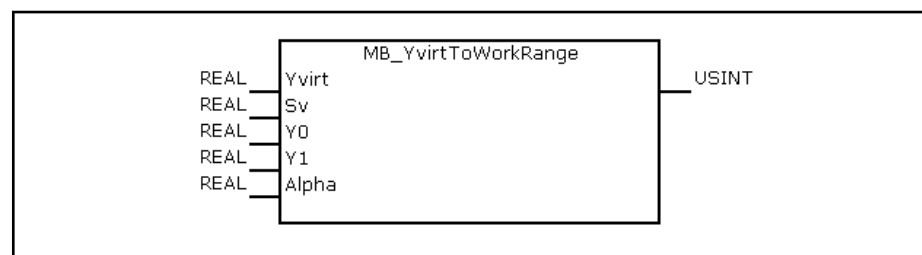


Fig. 6-7: MB_YVirtToWorkRange function

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 the first and the second elbow [°]
Return value		USINT	Provides the working area (1 or 2) of crank 1.

Tab. 6-6: Interface variables of the MB_YVirtToWorkRange function

Troubleshooting The function does not generate any errors.

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 and 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".



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
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion MLD with MPx08 and MLD with MPx17 without MA function package	MX_Packaging_07.lib

Tab. 6-7: Reference table of the MB_CamTableBellCrank

ML_TechCrank.library

Interface description

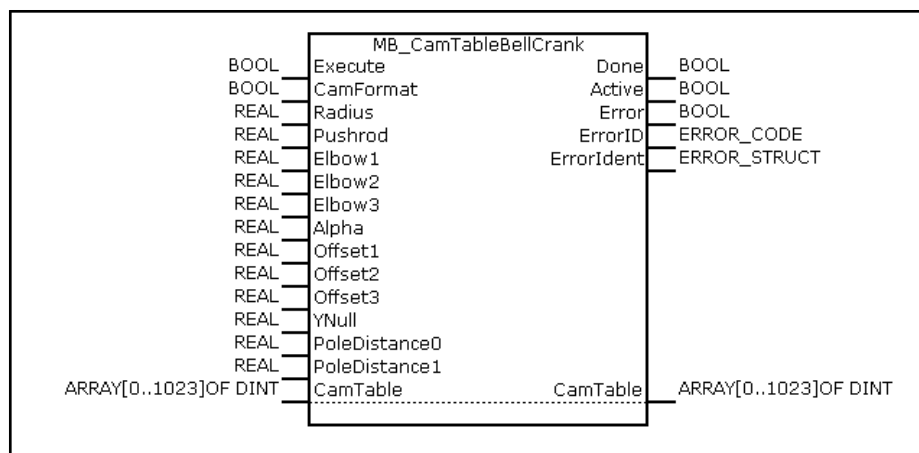


Fig. 6-8: MB_CamTableBellCrank function block

I/O type	Name	Data type	Description
VAR_IN_OUT	CamTable	ARRAY [0..1023] OF DINT	Array with the calculated transformation cam data
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]
	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

I/O type	Name	Data type	Description
VAR_OUTPUT	Done	BOOL	Calculation complete -> cam table valid
	Active	BOOL	Function block is processed
	Error	BOOL	Error (see ErrorID and ErrorIdent)
	ErrorID	ERROR_CODE	Error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 400

Tab. 6-8: 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 translatable and modulo-scaled.

The modulo value is to be set as follows: Modulo value = $2 \cdot \text{travel range (Sv)}$. For this, the travel range (Sv) is defined as distance between top (Y1) and bottom (Y0) pole.

The travel range (Sv) can be determined using the MB_BellCrankData function block

- The crank drive has to be scaled rotary, modulo 360°

Use case of MB_CamTableBell-Crank

Using the following sequence, the translatable 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 translatable position resulting from the crank angle. The resulting translatable position "Yvirt" can be determined from the crank angle (Phi) using the MB_PhiToYvirt function block. 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 C_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

ML_TechCrank.library



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 > 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. 6-9: 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 397,.

The principle of superimposition is shown in [fig. 6-9 "Principle of the cam superimposition" on page 401](#). 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). Thus, the table value 100% corresponds to movement $2 \cdot \text{travel range} = \text{modulo value of the virtual translatory axis}$.

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 translatable axis moves forwards and backwards with reversal of direction of crank 1 (as shown in [fig. 6-9 "Principle of the cam superimposition" on page 401](#))

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.

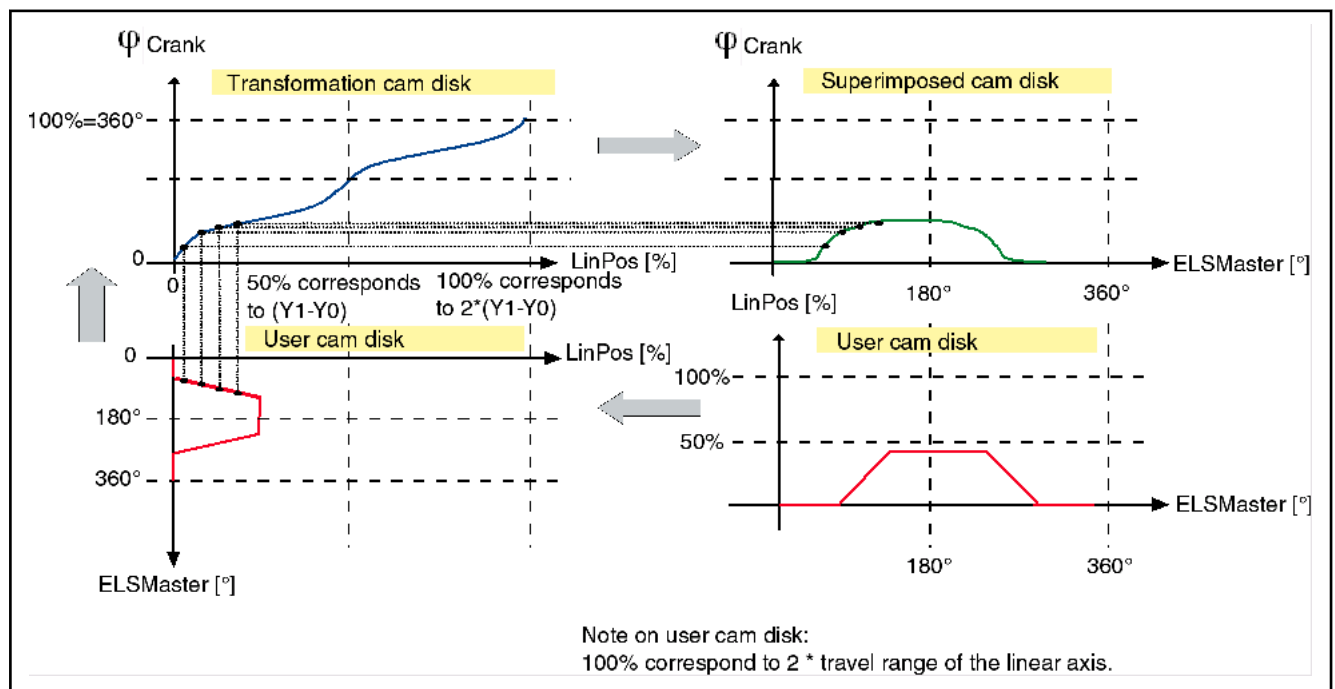


Fig. 6-9: Principle of the cam superimposition



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 cyclotime.

Troubleshoot the error F5113002 as follows

- Reduce Motion cycle time (if possible)
- Reduce master axis velocity (if possible)
- Increase PoleDistance0 and PoleDistance1 inputs

ML_TechCrank.library

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion MLD with MPx08 and MLD with MPx17 with MA function package	MX_Packaging_07.lib

Tab. 6-10: Reference table of the MB_CamTableBellCrankSuperimposed

Interface description

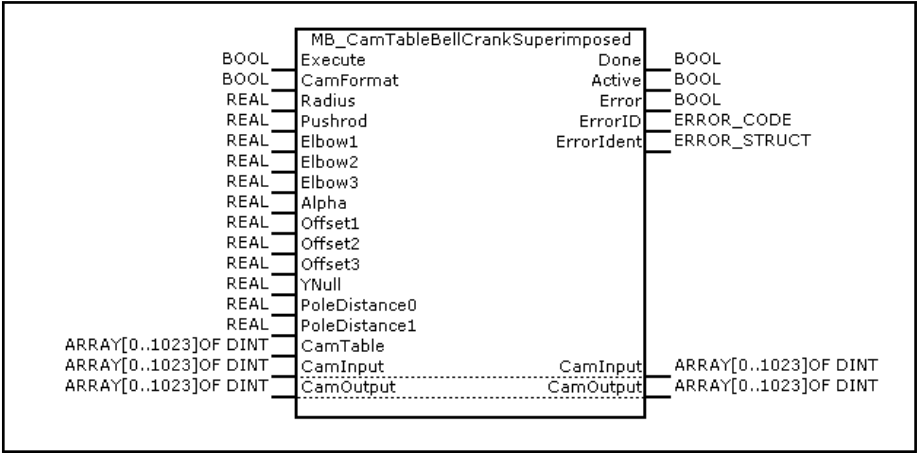


Fig. 6-10: Function block MB_CamTableBellCrankSuperimposed

I/O type	Name	Type	Comment
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

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 ErrorIdent)
	ErrorID	ERROR_CODE	Error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 405

Tab. 6-11: 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 • travel range (Sv). For this, the travel range (Sv) is defined as distance between top (Y1) and bottom (Y0) pole.

The traversing range (Sv) can be determined using the [MB_BellCrank-Data, page 394](#), function block

- 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 > 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. 6-12: 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	Library
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion MLD with MPx07/08/17 with MA function package	MX_Packaging_07.lib

Tab. 6-13: Reference table of the MB_PhiToYvirt

Interface description

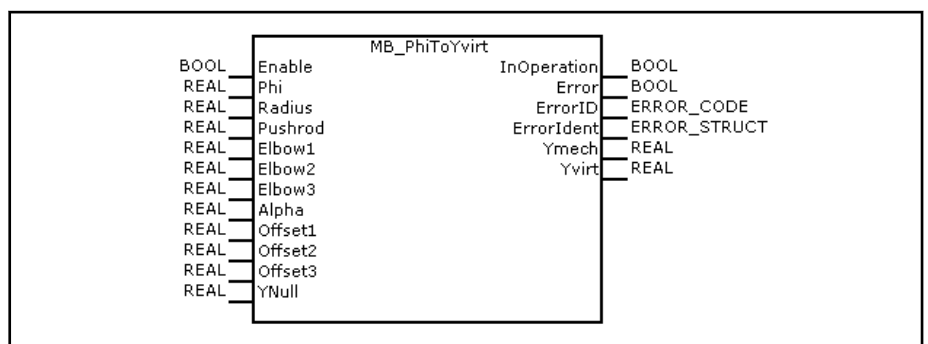


Fig. 6-11: Function block MB_PhiToYvirt

ML_TechCrank.library

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 ErrorIdent)
	ErrorID	ERROR_CODE	Error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 406
	Ymech	REAL	Mechanical translatory position [mm]
	Yvirt	REAL	Virtual translatory position [mm]

Tab. 6-14: Interface variables of the MB_PhiToYvirt 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#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. 6-15: 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	Library
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion MLD with MPx08 and MLD with MPx17 with MA function package	MX_Packaging_07.lib

Tab. 6-16: Reference table of the MB_YvirtToPhi

Interface description

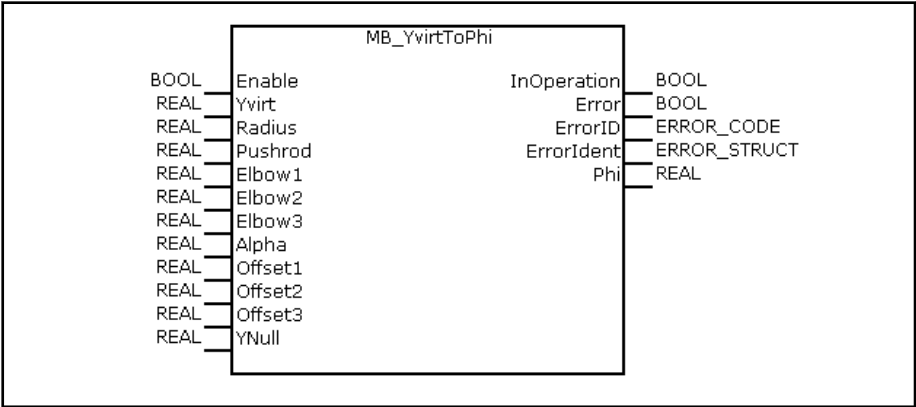


Fig. 6-12: MB_YvirtToPhi function block

ML_TechCrank.library

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 ErrorIdent)
	ErrorID	ERROR_CODE	Error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 408
	Phi	REAL	Crank angle in [°]

Tab. 6-17: Interface variables of the MB_YvirtToPhi 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#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. 6-18: 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	Library
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion MLD with MPx08 and MLD with MPx17 with MA function package	MX_Packaging_07.lib

Tab. 6-19: Reference table of the MB_YvirtToPhiPoly5

Interface description

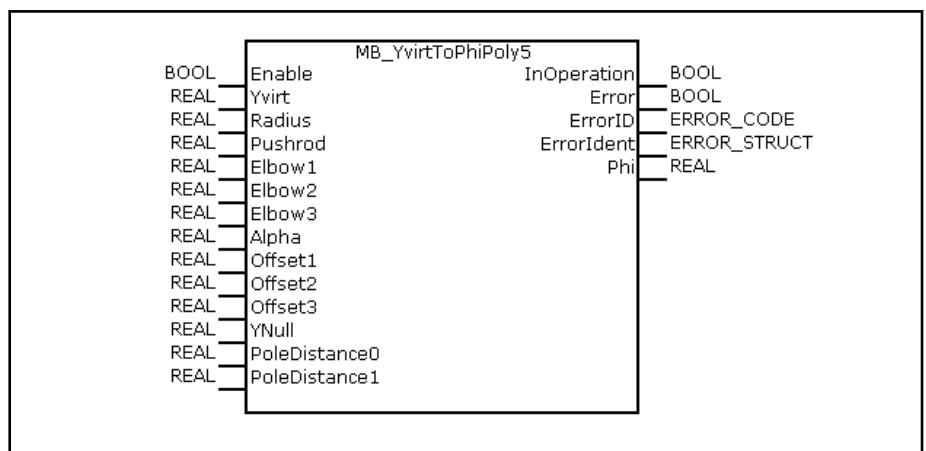


Fig. 6-13: MB_YvirtToPhiPoly5 function block

ML_TechCrank.library

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 ErrorIdent)
	ErrorID	ERROR_CODE	Error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 410
	Phi	REAL	Crank angle in [°]

Tab. 6-20: Interface variables of the MB_YvirtToPhiPoly5 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#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. 6-21: Error codes of the MB_YvirtToPhiPoly5 function block

6.2.10 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	Library
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion MLD with MPx07/08/17 with MA function package	MX_Packaging_07.lib

Tab. 6-22: Reference table of the MB_YvirtToYmech

Interface description

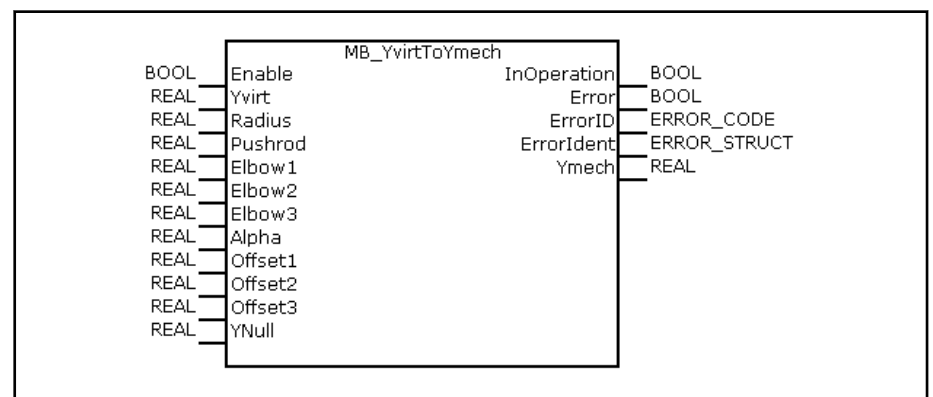


Fig. 6-14: MB_YvirtToYmech function block

ML_TechCrank.library

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 ErrorIdent)
	ErrorID	ERROR_CODE	Error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 412
	Ymech	REAL	Mechanical translatory position (Ymech) [mm]

Tab. 6-23: Interface variables of the MB_YvirtToYmech

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#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. 6-24: Error codes of the MB_YvirtToYmech function block

6.3 Functions and function blocks for the crank kinematics (Crank)

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.

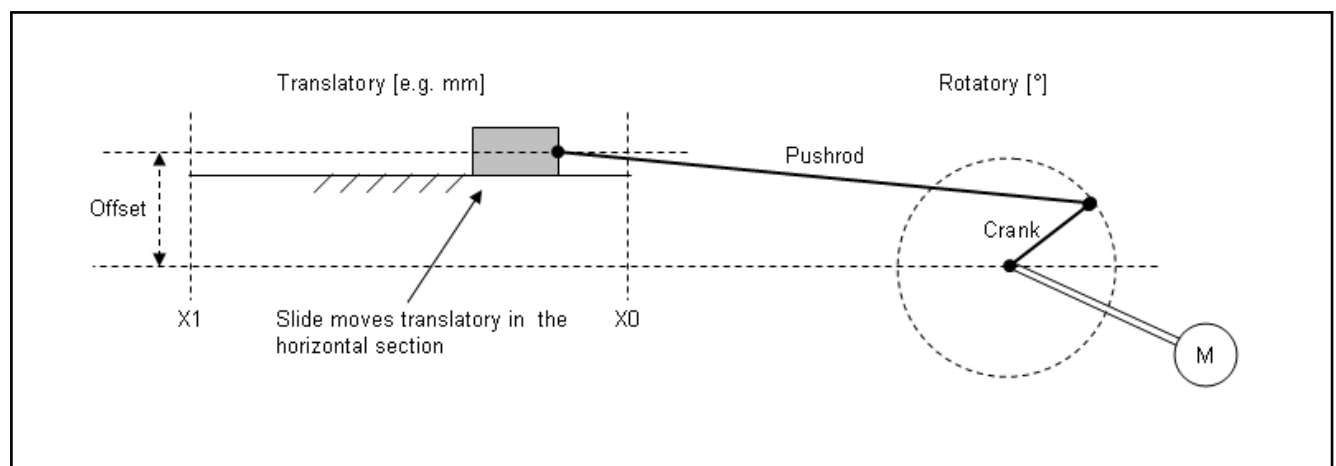


Fig. 6-15: Single-axis crank kinematics with offset crank axis

The following functions and function blocks are supported:

Function	Description
MB_MasterToPhi	Outputs crank angles calculated using the master axis position and the superimposed cam

Tab. 6-25: Crank kinematics function

ML_TechCrank.library

Function block	Description
MB_CamTableCrank	Calculates the cam for transforming a translatory virtual master axis position into rotary values (crank angle)
MB_CamTableCrankSuperimposed	Superimposes a user cam with the transformation cam for a superimposed cam.
MB_PhiToXvirt	Converts crank angle (Phi) into virtual translatory position values (Xvirt) to be used by the PLC program
MB_XvirtToXmech	Converts virtual translatory position values (Xvirt) to mechanic translatory position values (Xmech)

Tab. 6-26: Crank kinematics function blocks

Typical use cases are:

Use case 1: 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 the MB_CamTableCrank function block 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 [chapter 6.3.4 "MB_CamTableCrank" on page 418](#).

Use case 2: 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 the MB_CamTableSuperimposed function block, and the resulting cam is transferred to the drive by the PLC program. For detailed information, see [chapter 6.3.5 "MB_CamTableCrankSuperimposed" on page 422](#).

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. Therefore, the MB_PhiToXvirt function block is used. For detailed information, see [chapter 6.3.7 "MB_PhiToXvirt" on page 428](#).

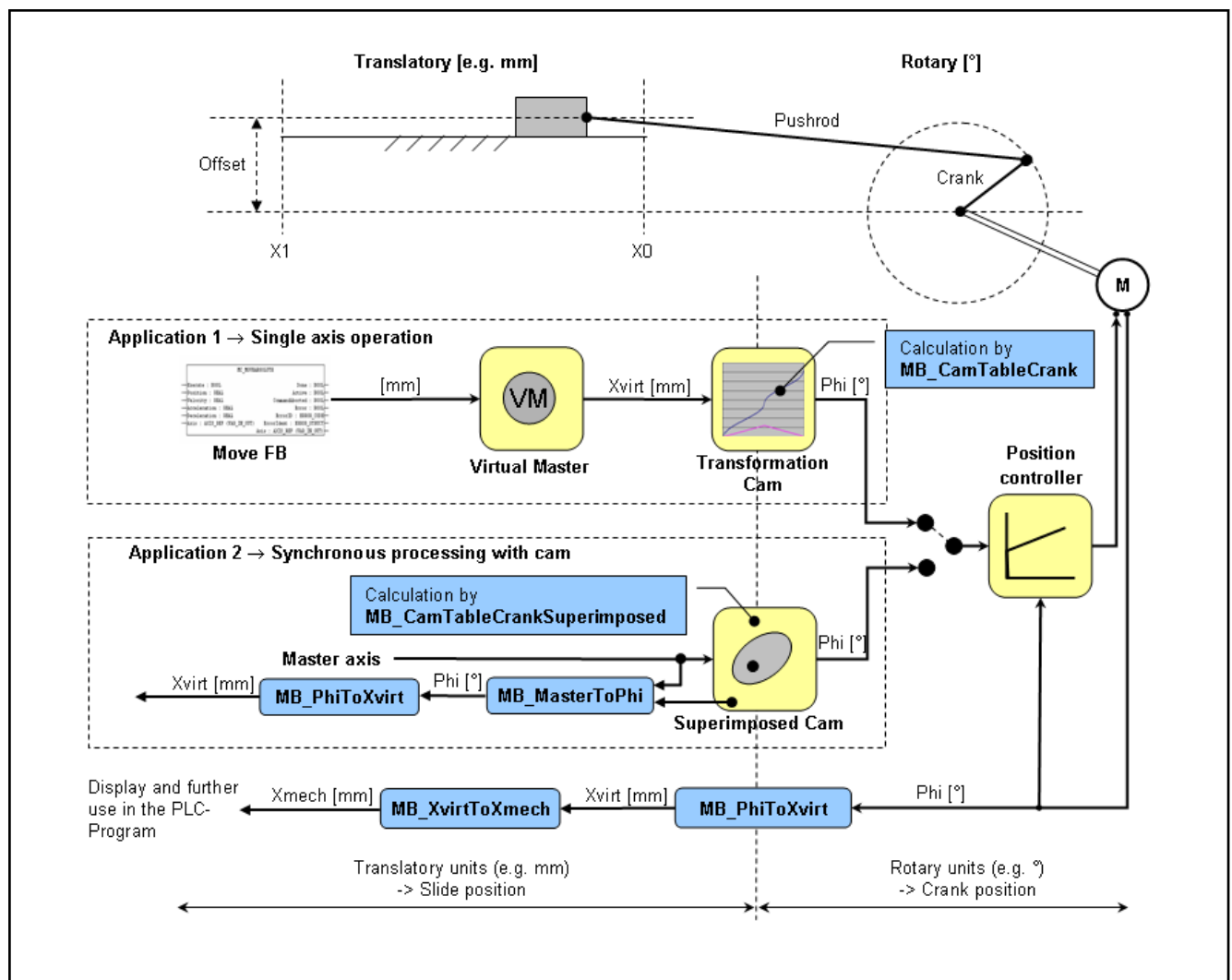


Fig. 6-16: Schematic representation of the crank kinematics technology function blocks

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.

ML_TechCrank.library

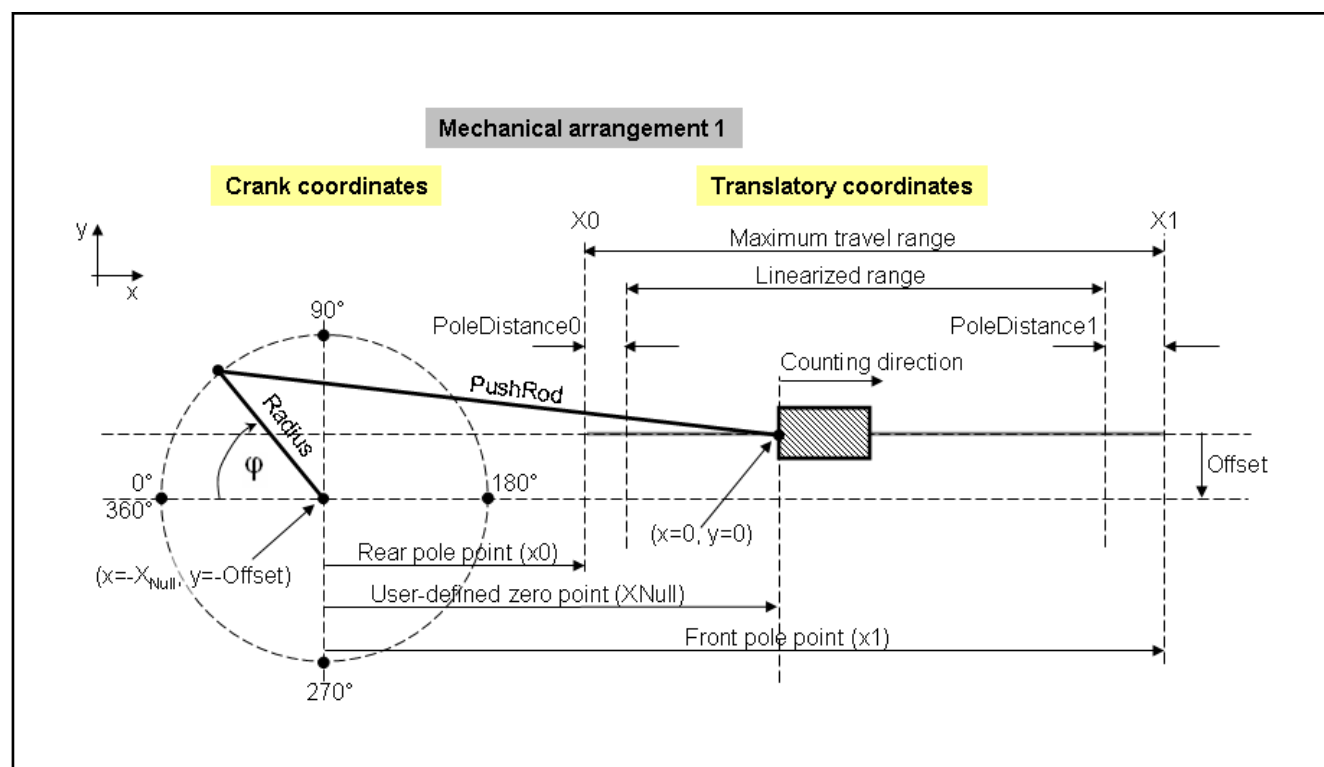


Fig. 6-17: Mechanical arrangement 1: Relation of crank and translatory coordinates in the crank kinematics

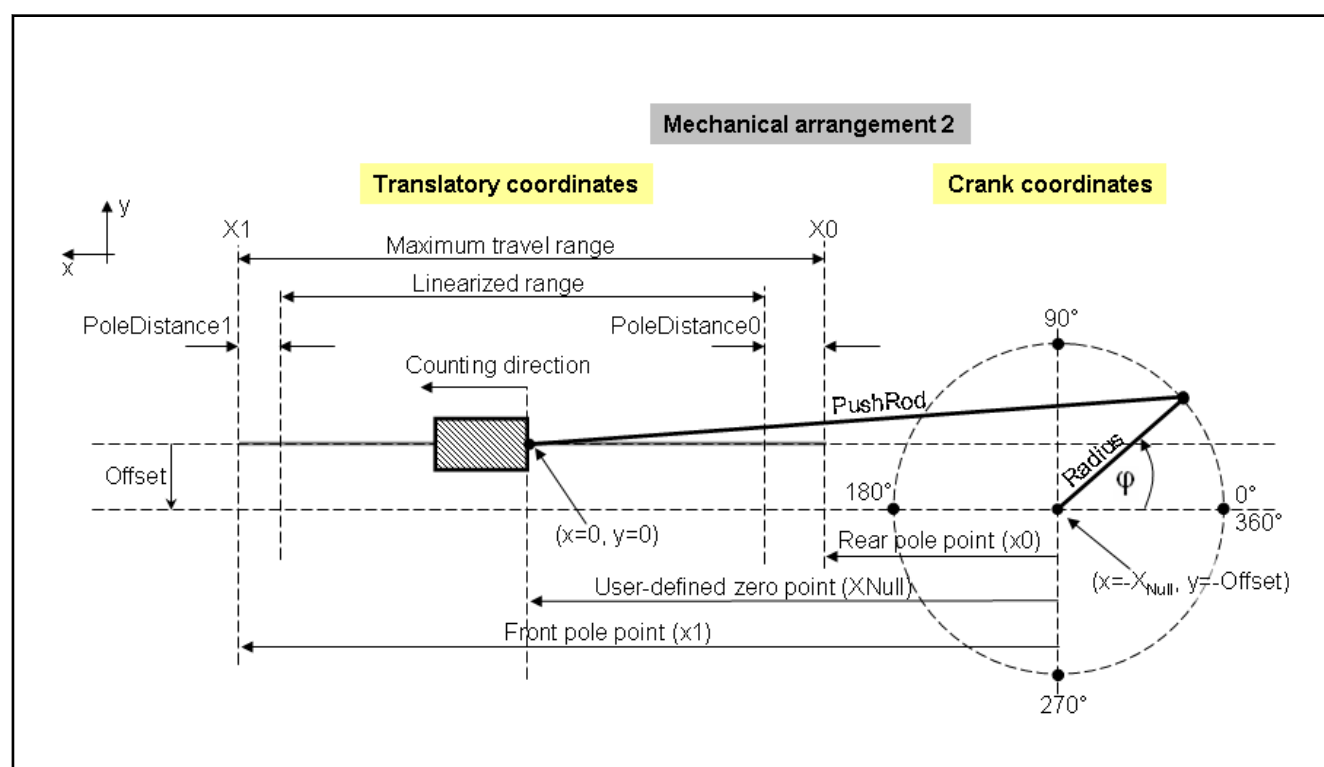


Fig. 6-18: Mechanical arrangement 2: Relation of crank and translatory coordinates in the crank kinematics

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 travel 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: Variable 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 416 and [fig. 6-18 "Mechanical arrangement 2: Relation of crank and translatory coordinates in the crank kinematics"](#) on page 416, :

- 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:

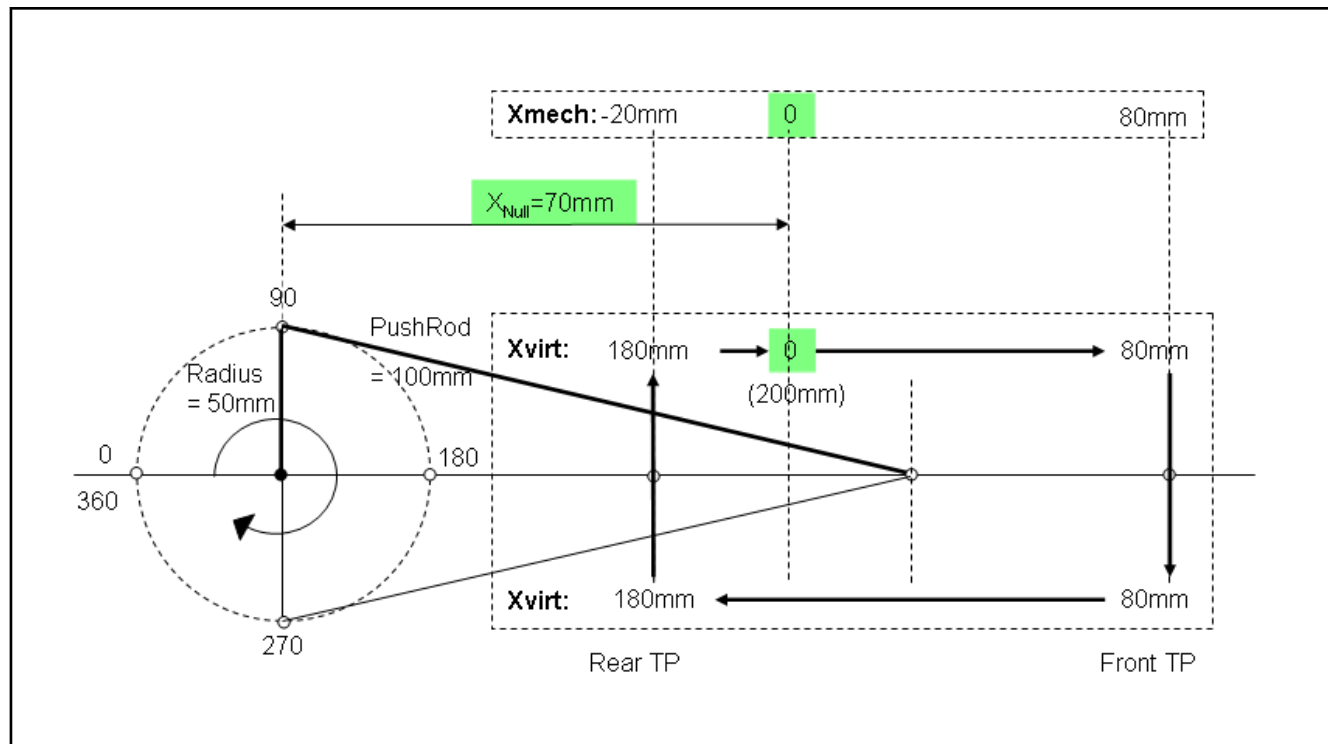
$$\text{Travelrange} = (\text{FrontPolePoint} - \text{RearPolePoint})$$

$$\text{Travelrange} = \sqrt{(\text{Radius} + \text{PushRod})^2 - \text{Offset}^2} - \sqrt{(\text{Radius} - \text{PushRod})^2 - \text{Offset}^2}$$

Fig. 6-19: Travel range equation for the crank kinematic

The following figure shows Xvirt and Xmech in an example:

ML_TechCrank.library

Fig. 6-20: Counting direction of the slide position (X_{virt} and X_{mech})

6.3.3 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 418 and [MB_CamTableCrankSuperimposed](#), page 422 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

```

6.3.4 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 413.

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).



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
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion MLD with MPx07/08/17 with MA function package	MX_Packaging_07.lib

Tab. 6-28: Reference table of the MB_CamTableCrank

Interface description

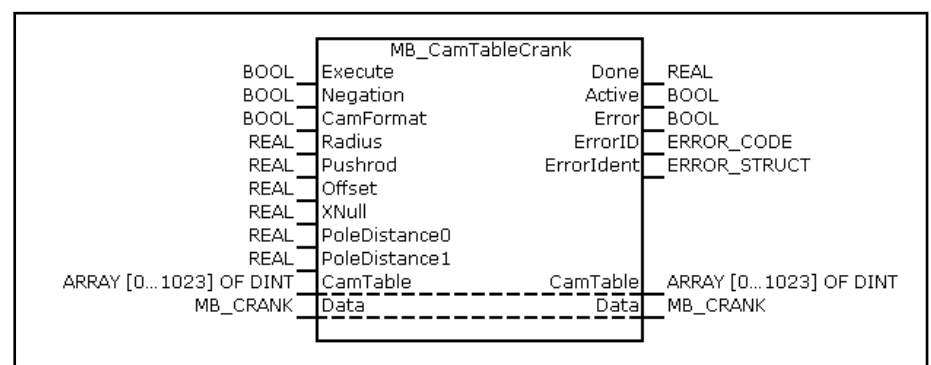


Fig. 6-21: MB_CamTableCrank function block

ML_TechCrank.library

I/O type	Name	Data type	Explanation
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 418
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 ErrorIdent)
	ErrorID	ERROR_CODE	Error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 421

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 travel range

Tab. 6-30: Boundary conditions

In order to be able to travel the translatory slide in single-axis mode, the following configuration has to be conducted in IndraWorks:

- Translatory scaling of the virtual axis with a modulo value of (2 x travel range).

- Rotary scaling of the crank drive (real axis) with a modulo value of 360°.



In order 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_PhiToX-virt, page 428](#).
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:

ML_TechCrank.library

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: Xnull > SQRT((Pushrod radius)^2-off-set^2)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0003	Xnull 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 > travel range
CALCULATION_ERROR (16#0007)	16#1001	16#0005	Internal calculation error: $(X_{\text{mech}}+X_{\text{zero}})^2+\text{offset}^2 < 0$
CALCULATION_ERROR (16#0007)	16#1001	16#0006	Internal calculation error: $(\text{Radius}^2+(X_{\text{mech}}+X_{\text{zero}})^2+\text{offset}^2+\text{Pushrod}^2)/2*\text{radius}*SQRT(rVar1) > 1$
CALCULATION_ERROR (16#0007)	16#1001	16#0007	Internal calculation error: $X_{\text{mech}}+X_{\text{zero}} = 0$

Tab. 6-31: Error codes of the MB_CamTableCrank function block

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 418](#).

The superimposition principle is shown in [Principle of the cam superimposition, page 423](#). 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 movement (2 x travel 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 translatable axis takes place, during which the crank changes direction continually (see figure below).

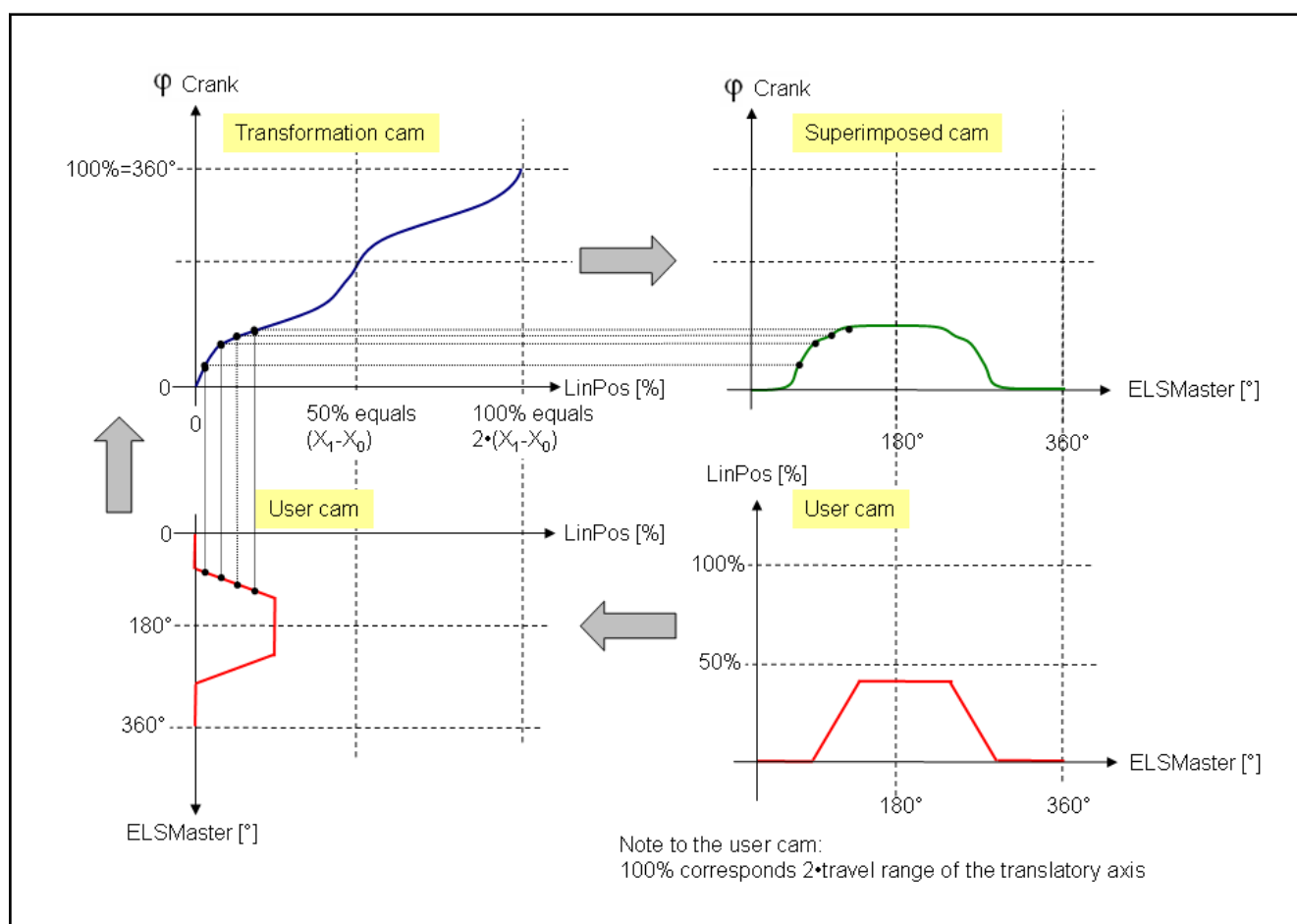


Fig. 6-22: Principle of the cam superimposition



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
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion MLD with MPx07/08/17 with MA function package	MX_Packaging_07.lib

Tab. 6-32: Reference table of the MB_CamTableCrankSuperimposed

ML_TechCrank.library

Interface description

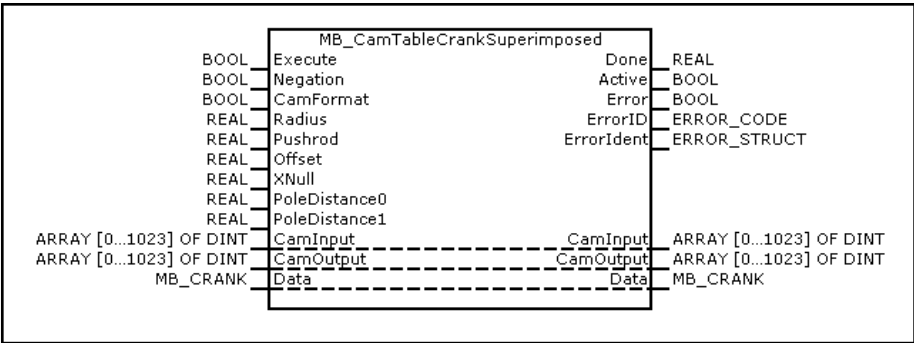


Fig. 6-23: MB_CamTableCrankSuperimposed function block

I/O type	Name	Data type	Description
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 418
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
	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 ErrorIdent)
	ErrorID	ERROR_CODE	Error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 426

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 travel range

Tab. 6-34: Boundary conditions

ML_TechCrank.library

In order to be able to travel the translatory slide in single-axis mode, the following configuration has to be conducted in IndraWorks:

- Translatory scaling of the virtual axis with a modulo value of (2 x travel 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°.



In order 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_CamTable-CrankSuperimposed. It was calculated with the superimposition of a user-defined cam in a transformation cam (see [Principle of the cam superimposition, page 423](#)).
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: Pushrod > radius + ABS(offset)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0002	Xnull is too small. The following condition has to be fulfilled: Xnull > SQRT((Pushrod radius)^2 - offset^2)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0003	Xnull too large. The following condition has to be fulfilled: Xzero < SQRT((Pushrod radius)^2 - offset^2) + 2*traversing range
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0004	Radius <= 0

ErrorID	Additional1	Additional2	Description
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 > travel range
CALCULATION_ERROR (16#0007)	16#1001	16#0005	Internal calculation error: $(X_{mech} + X_{zero})^2 + offset^2 < 0$
CALCULATION_ERROR (16#0007)	16#1001	16#0006	Internal calculation error: $(Radius^2 + (X_{mech} + X_{zero})^2 + offset^2 + Pushrod^2) / 2 * radius * SQRT(rVar1) > 1$
CALCULATION_ERROR (16#0007)	16#1001	16#0007	Internal calculation error: $X_{mech} + X_{zero} = 0$

Tab. 6-35: Error codes of the MB_CamTableCrankSuperimposed function block

6.3.6 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 has to be calculated before retrieving this function (by [MB_CamTableCrankSuperimposed](#), page 422).

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	Library
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion MLD with MPx07/08/17 with MA function package	MX_Packaging_07.lib

Tab. 6-36: Reference table of the MB_MasterToPhi

Interface description

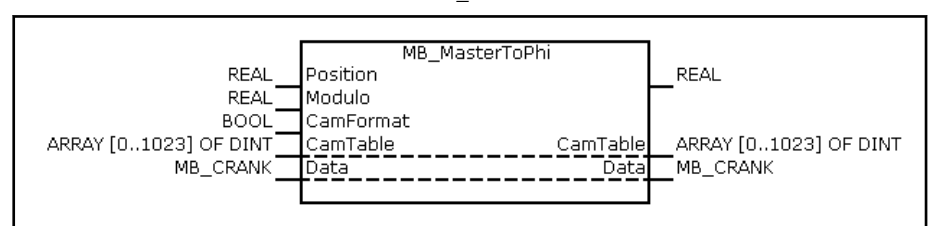


Fig. 6-24: Function MB_MasterToPhi

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 internally calculated data of the crank kinematics. The data is calculated in this function block and transferred to other function blocks MB_CRANK , page 418
VAR_INPUT	Position	REAL	Master axis position

ML_TechCrank.library

I/O type	Name	Data type	Description
	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)

Tab. 6-37: Interface variables of the MB_MasterToPhi function

6.3.7 MB_PhiToXvirt

Brief description The MB_PhiToXvirt function block converts crank angle values (Phi) to virtual translatory position values (Xvirt).

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion MLD with MPx08 and MLD with MPx17 with MA function package	MX_Packaging_07.lib

Tab. 6-38: Reference table of the MB_PhiToXvirt

Interface description

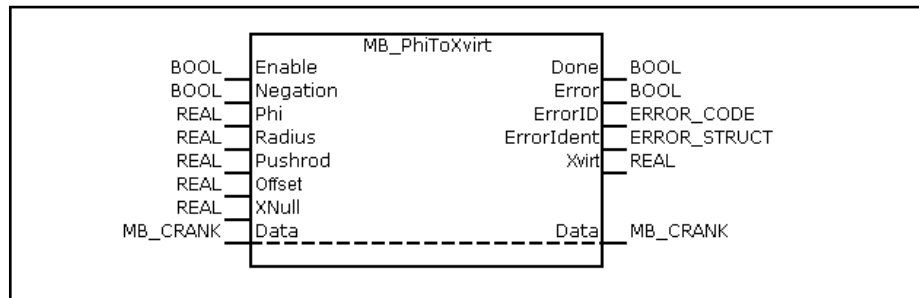


Fig. 6-25: MB_PhiToXvirt function block

I/O type	Name	Data type	Description
VAR_IN_OUT	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 418
VAR_INPUT	Enable	BOOL	Calculation of Xvirt in every cycle, while Enable=TRUE
	Negation	BOOL	Inverts the direction of counting of a 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 ErrorIdent)
	ErrorID	ERROR_CODE	Error description

I/O type	Name	Data type	Description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 429
	Xvirt	REAL	Virtual translatory position (Xvirt)

Tab. 6-39: Interface variables of the MB_PhiToXvirt 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	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 too large. The following condition has to be fulfilled: $\text{Xzero} < \text{SQRT}((\text{Pushrod radius})^2 - \text{offset}^2) + 2 * \text{traversing range}$
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. 6-40: Error codes of the MB_PhiToXvirt function block

6.3.8 MB_XvirtToXmech

Brief description The MB_XvirtToXmech function block converts the virtual translatory position (Yvirt) to the mechanical translatory position (Ymech) of the carriage. Xmech values can be used for display purposes (e.g. on an HMI device).

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechCrank
IndraMotion 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

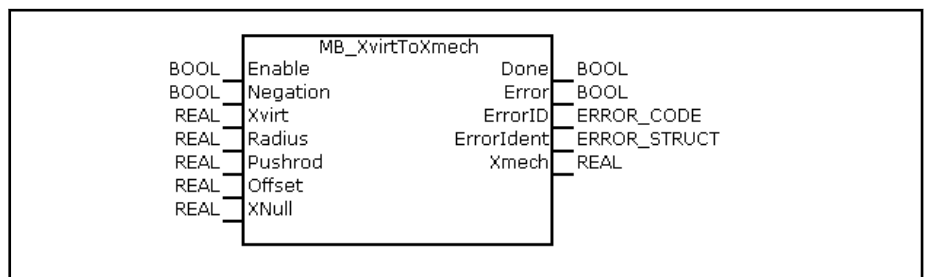


Fig. 6-26: MB_XvirtToXmech function block

ML_TechCrank.library

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 direction of counting of a 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]
VAR_OUTPUT	Done	BOOL	Output value "Phi" valid
	Error	BOOL	Error (see ErrorID and ErrorIdent)
	ErrorID	ERROR_CODE	Error description
	ErrorIdent	ERROR_STRUCT	Detailed error description Troubleshooting, page 430
	Xmech	REAL	Mechanical translatory position (Ymech) of the carriage

Tab. 6-42: Interface variables of the MB_XvirtToXmech 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: Pushrod > radius + ABS(offset)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0002	Xnull is too small. The following condition has to be fulfilled: Xnull > SQRT((Pushrod radius)^2-offset^2)
INPUT_RANGE_ERROR (16#0006)	16#1000	16#0003	Xnull too large. The following condition has to be fulfilled: Xzero < SQRT((Pushrod radius)^2-offset^2)+2*traversing 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 that are **subject to license**.

This documentation describes the inputs and outputs of the individual function blocks and provides notes on their use.

The function blocks of the ML_TechTensionAdvanced

Function block	Function block
MB_TensionControlLoadCellType02, page 431	Multi-axis tension controller function block with dynamic precontrol for up to eight axes
MB_TensionControlLoadCellMatParType02, page 445	Multi-axis tension controller function block with simple parameterization and dynamic precontrol for up to eight axes
MB_TensionDecouplingNetworkType01, page 461	Decoupling network function block for multi-axis tension controller



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).

7.2 MB_TensionControlLoadCellType02

Brief description

Depending on the measured value of a load cell, the tension controller controls the tension of up to eight axes to the desired command value.

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	ML_TechTensionAdvanced

Tab. 7-1: Reference table of the MB_TensionControlLoadCellType02



The use of the function block is subject to license (for details, refer to [Introduction and overview, page 431](#)).



IndraMotion MLC 12VRS or higher additionally supports slave axes with interpolation on the control.

ML_TechTensionAdvanced.library

Interface description

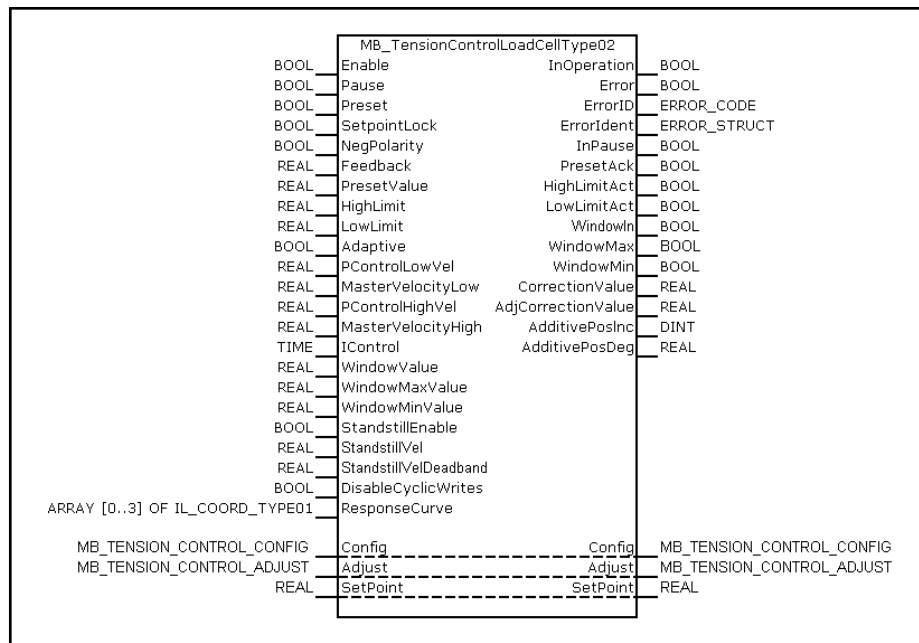


Fig. 7-1: MB_TensionControlLoadCellType02 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Config	MB_TENSION_CONTROL_CONFIG	Configuration of the tension controller (axes, distribution of tension) MB_TENSION_CONTROL_CONFIG, page 459
	Adjust	MB_TENSION_CONTROL_ADJUST	Structure for the slave axis settings at runtime MB_TENSION_CONTROL_ADJUST, page 460
	Setpoint	REAL	Tension command value This is generally set by the user but can also be changed by the function block (see "SetpointLock" on page 441)
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

ML_TechTensionAdvanced.library

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 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 from RIL_LoopControl.library

ML_TechTensionAdvanced.library

I/O type	Name	Data type	Description
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 444)
	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"
	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)

Tab. 7-2: 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
DisableCyclicWrites	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. 7-3: 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

ML_TechTensionAdvanced.library

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).

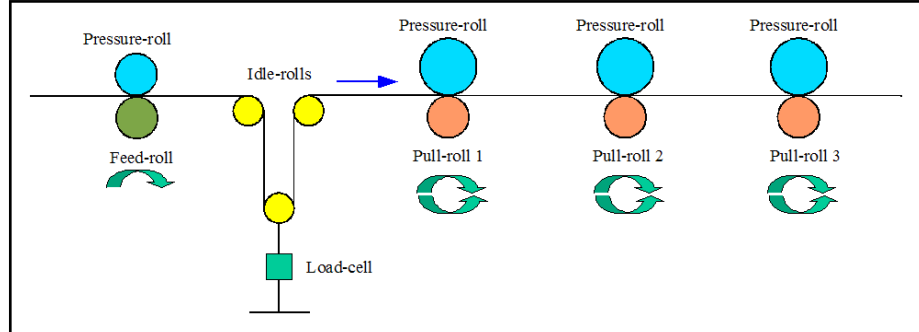


Fig. 7-2: Multi-axis tension controller

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.

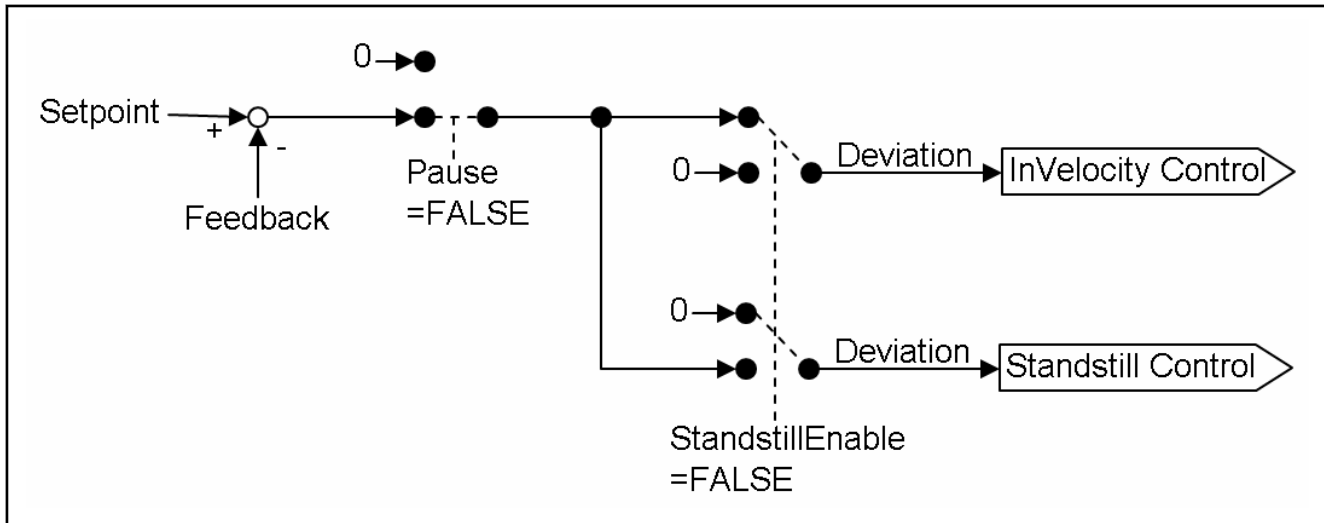


Fig. 7-3: Flowchart of the tension controller with branch for the control at standstill/in velocity

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.

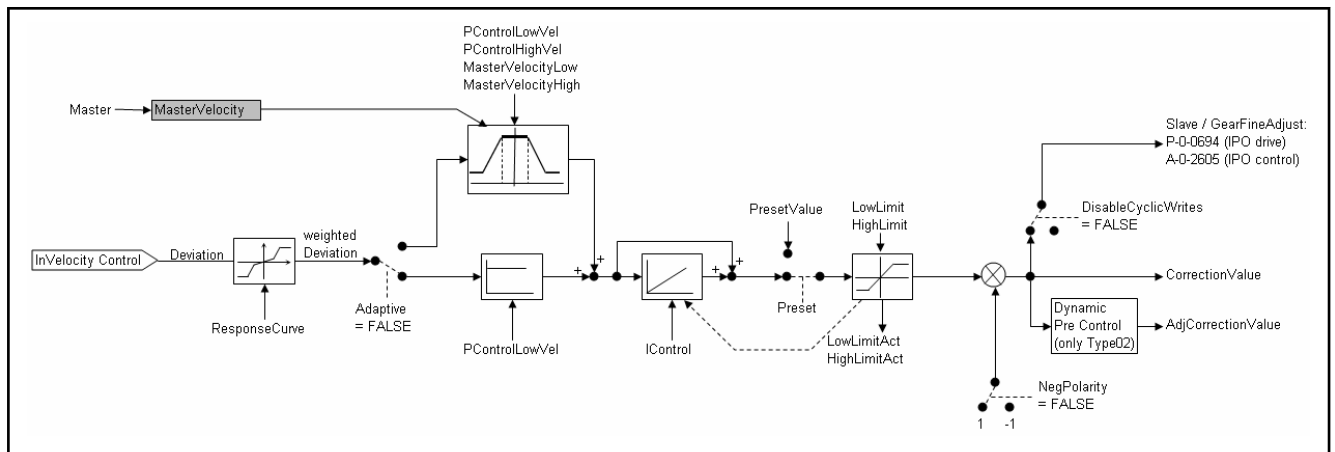


Fig. 7-4: Flowchart of the tension controller for the "InVelocity Control" branch

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).

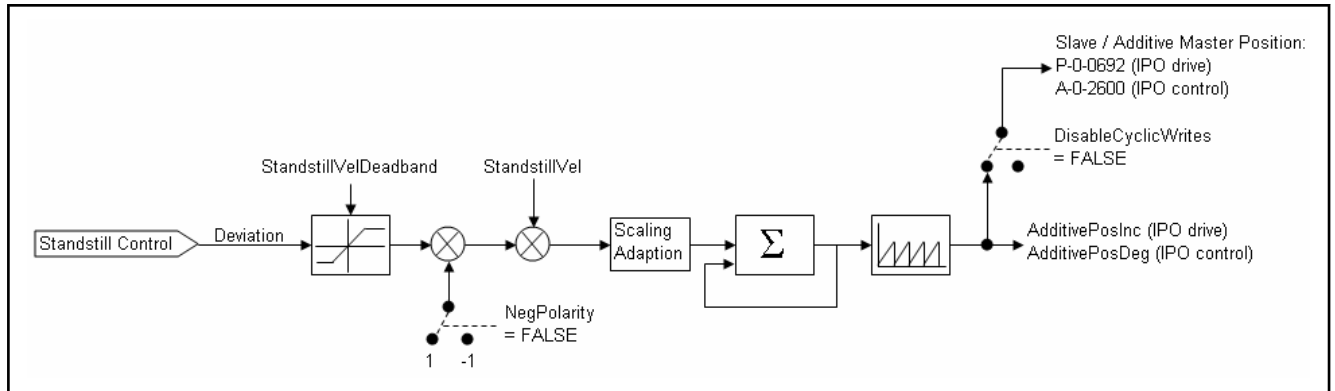


Fig. 7-5: Flowchart of the tension controller for the closed-loop control at standstill (Standstill Control)

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.

ML_TechTensionAdvanced.library



- 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.

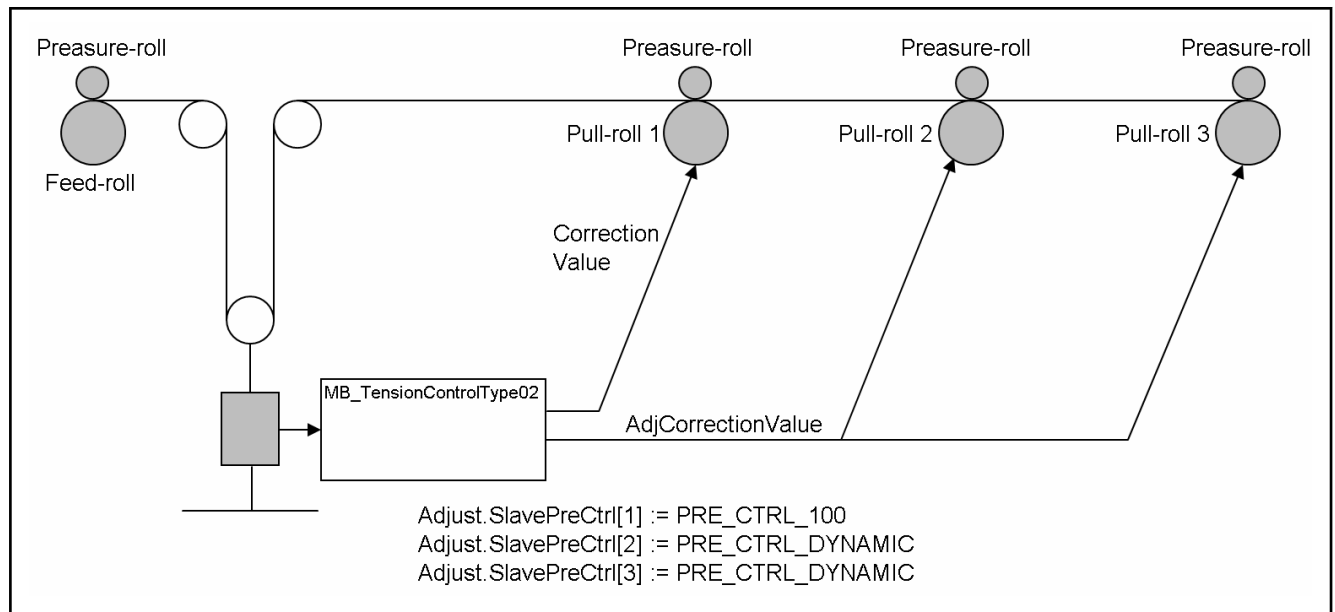


Fig. 7-6: MB_TensionControlType02: Example cascade with three draw rollers

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" input 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. This is achieved with a characteristic curve with up to four data points. If the input is not assigned, a 1-to-1 characteristic curve is used as default value. That means that it is not scaled.

ML_TechTensionAdvanced.library

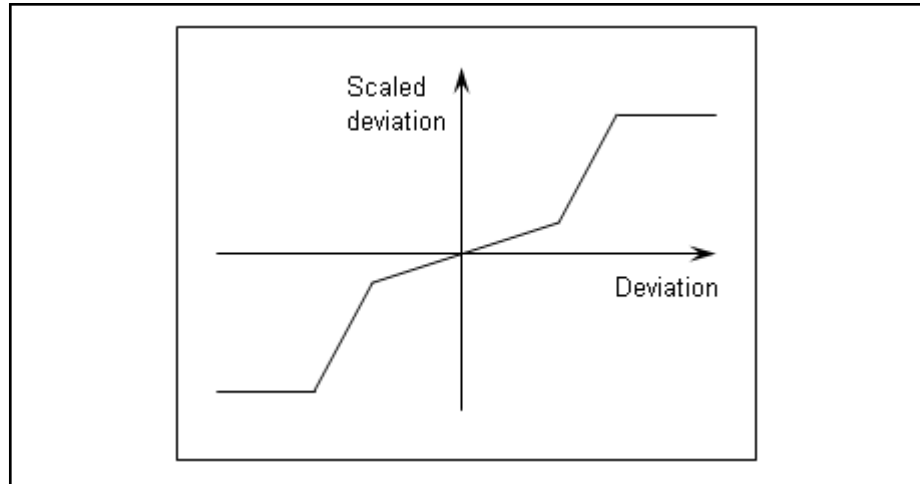


Fig. 7-7: Scaling the control deviation using the "ResponseCurve" input

The function can for example 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 set the controller parameters first using the system stability and then, the steady state is optimized.

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:

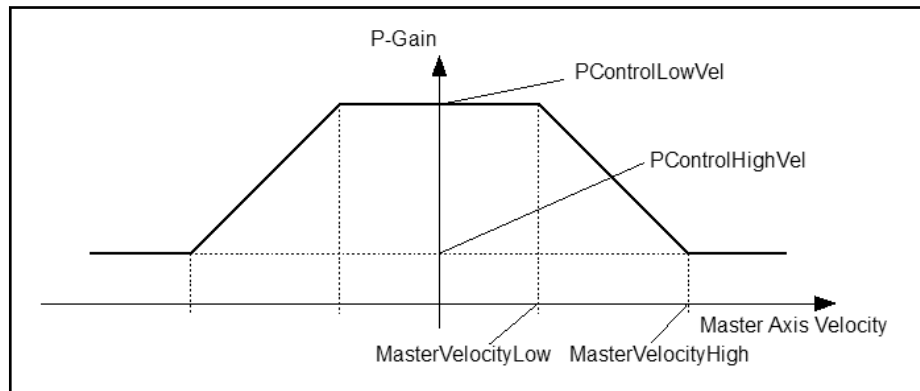


Fig. 7-8: Characteristic curve for the adaptive P-gain

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. 7-9: 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:

ML_TechTensionAdvanced.library

$$Feedback > \left(SetPoint + |SetPoint| \cdot \frac{WindowMaxValue - 100}{100} \right)$$

Fig. 7-10: 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. 7-11: 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 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.

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. 7-12: 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:

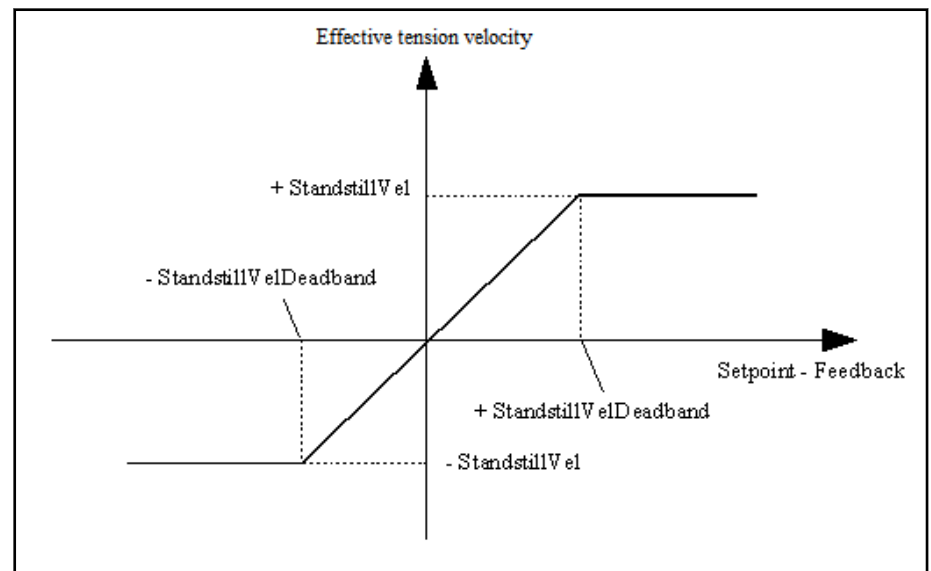


Fig. 7-13: Modified signum function to limit the added "StandstillVel"

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 "StandstillVel". 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
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. 7-4: Error codes of the MB_TensionControlLoadCellType02 function block

7.3 MB_TensionControlLoadCellMatParType02

Brief description Depending on the function block, 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_TensionControlLoadCellType0, this variant offers the following benefits.

- Controller parameterization by machine and material data

ML_TechTensionAdvanced.library

- 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
IndraMotion MLC 13VRS	ML_TechTensionAdvanced

Tab. 7-5: MB_TensionControlLoadCellMatParType02 reference table



The use of the function block is subject to license (for details, refer to [Introduction and overview, page 431](#)).

Interface description

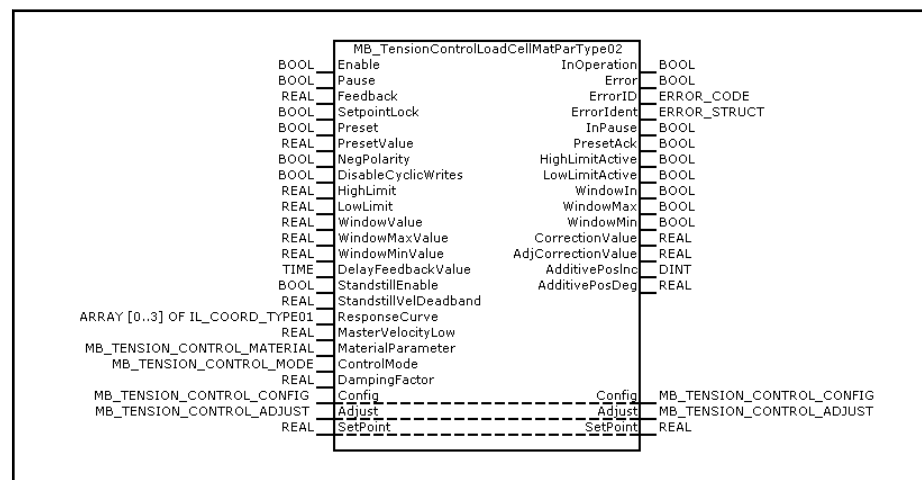


Fig. 7-14: MB_TensionControlLoadCellMatParType02 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Config	MB_TENSION_CONTROL_CONFIG	Configuration of the tension controller (axes, distribution of tension) MB_TENSION_CONTROL_CONFIG on page 459
	Adjust	MB_TENSION_CONTROL_ADJUST	Structure for the slave axis settings at runtime MB_TENSION_CONTROL_ADJUST on page 460
	Setpoint	REAL	Tension command value [N]*. This is generally set by the user but can also be changed by the function block (see SetpointLock on page 452)
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]*

ML_TechTensionAdvanced.library

I/O type	Name	Data type	Description
	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) [%]
	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 "InVelocity Control" branch IL_COORD_TYPE01 from RIL_LoopControl.library

ML_TechTensionAdvanced.library

I/O type	Name	Data type	Description
	MasterVelocityLow	REAL	Master axis velocity below which the controller adaption 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 on page 300
	ControlMode	MB_TENSION_CONTROL_MODE	Selection of the control loop response. Optimized for command value or disturbance response MB_TENSION_CONTROL_MODE on page 301
	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_STRUCTURE	For a detailed error description according to the error table (see Error codes on page 458).
	InPause	BOOL	"Pause" is active
	PresetAck	BOOL	If this output is active, the "Preset" input was analyzed and the "CorrectionValue" output applied the "PresetValue" 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 "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 [%]
	AdjCorrectionValue	REAL	Control variable of the tension controller with dynamic precontrol [%]

I/O type	Name	Data type	Description
	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) [°]

*

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. 7-6: 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"
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

ML_TechTensionAdvanced.library

Name	Type	Min. value	Max. value	Default value	Effective
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. 7-7: 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.

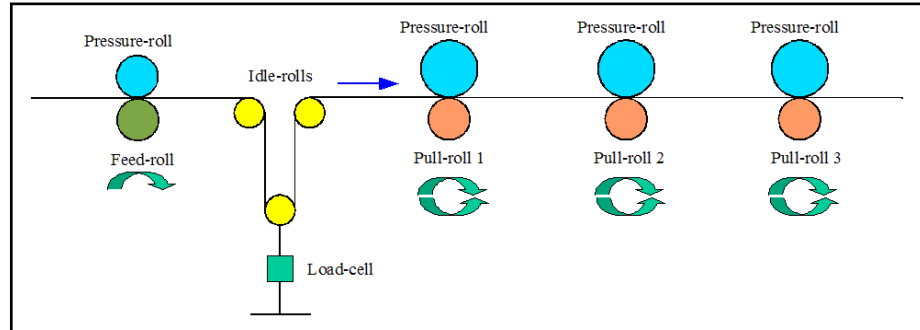


Fig. 7-15: Multi-axis tension controller

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.

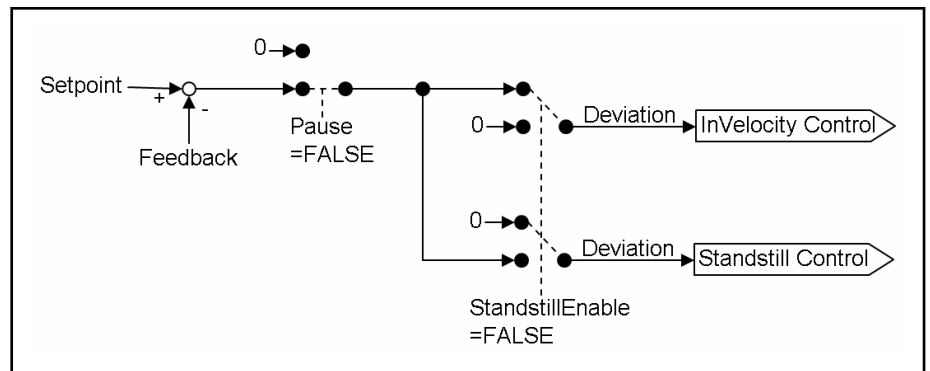


Fig. 7-16: Flowchart of the tension controller with branch for the control at standstill/in velocity

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).

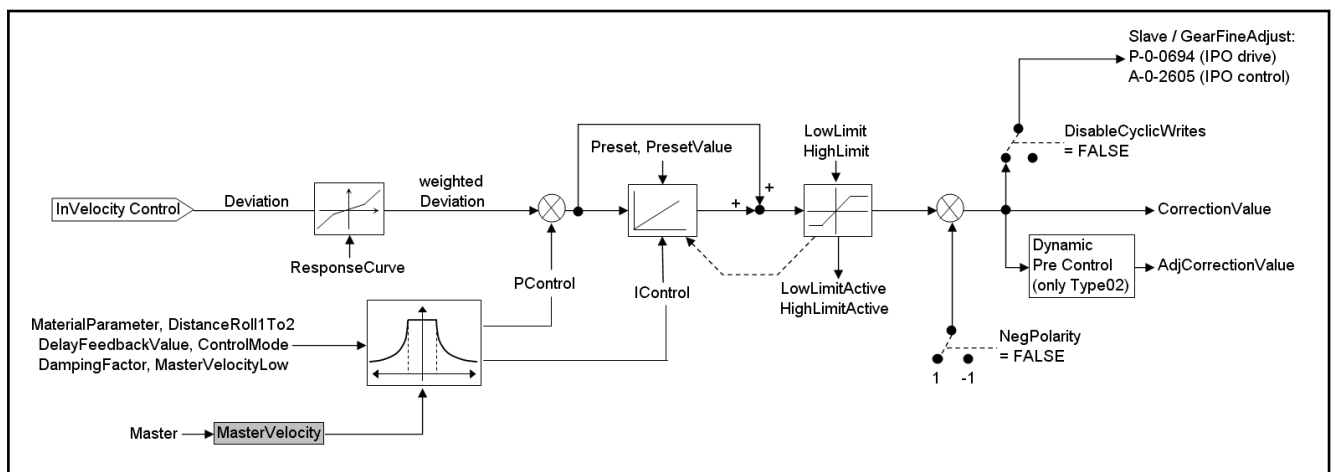


Fig. 7-17: Flowchart of the tension controller for the (InVelocity Control) branch

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.

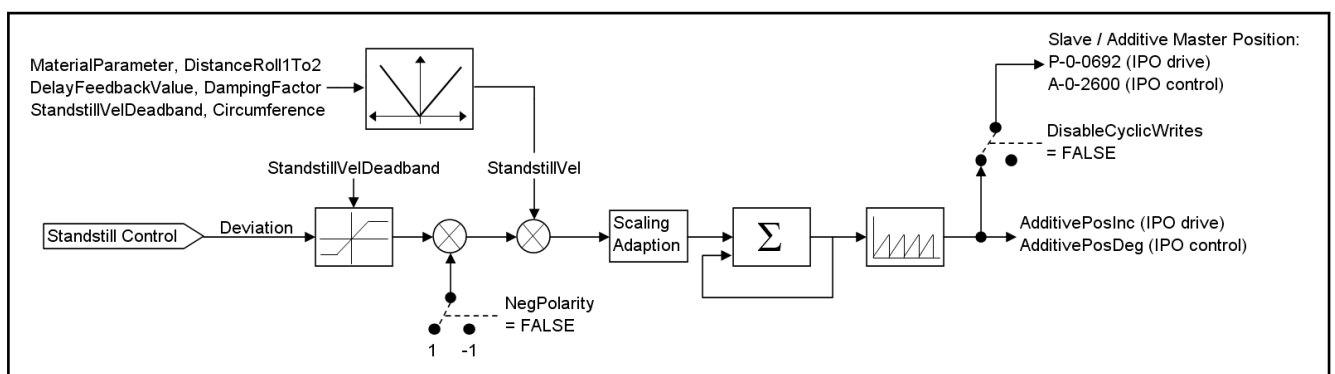


Fig. 7-18: Flowchart of the tension controller for the closed-loop control at standstill (Standstill Control)

To ensure user-friendliness, the flowcharts do not contain the "WindowValue", "WindowMaxValue" or "WindowMinValue" limitations and their status displays.

ML_TechTensionAdvanced.library

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 section Gear ratio fine adjustment, page 458). 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. 7-19: 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. 7-20: 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:

ML_TechTensionAdvanced.library

$$Feedback < \left(SetPoint + |SetPoint| \cdot \frac{WindowMinValue - 100}{100} \right)$$

Fig. 7-21: 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. This is achieved with a characteristic curve with up to four data points. If the input is not assigned, a 1-to-1 characteristic curve is used as default value. That means that it is not scaled.

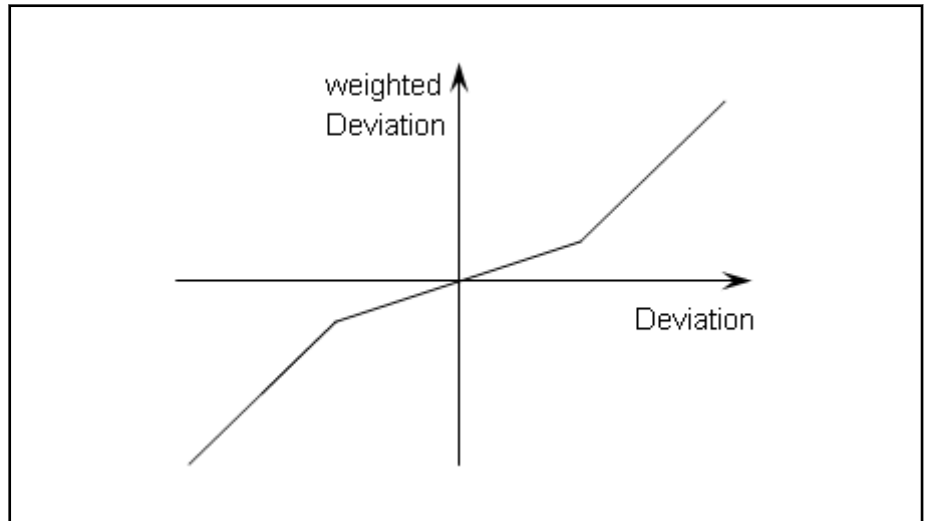


Fig. 7-22: Scaling the control deviation using the "ResponseCurve" input

The function can for example 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).

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.

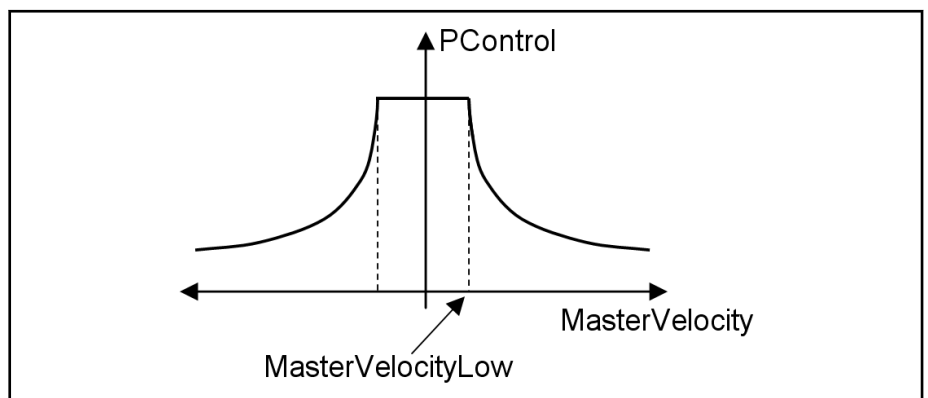


Fig. 7-23: Dependency of the controller gain from the web velocity

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. 7-24: 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 "Distance-Roll1to2" 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 "Slave-PreCtrl" 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.

ML_TechTensionAdvanced.library

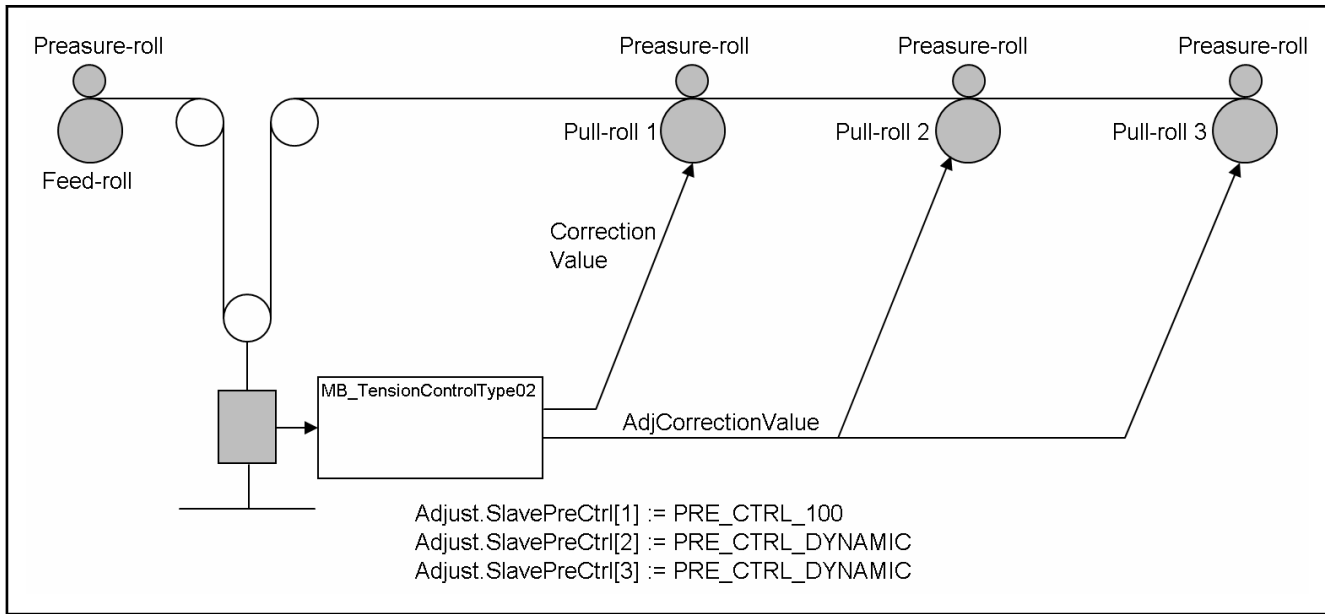


Fig. 7-25: *MB_TensionControlLoadCellMatParType02 example cascade with 3 draw rollers*

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. 7-26: *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.Circumfence 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 IndraMotion MLD

Tab. 7-8: Error codes of the MB_TensionControlLoadCellMatParType02 function block

7.4 MB_TENSION_CONTROL_CONFIG

The MB_TENSION_CONTROL_CONFIG structure is used to configure the tension controller (axes, tension distribution).

ML_TechTensionAdvanced.library

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. 7-9: MB_TENSION_CONTROL_CONFIG structural description

7.5 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 460.

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. 7-10: Structural description of MB_TENSION_CONTROL_ADJUST

7.6 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 460.

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. 7-11: Structure of enumeration type MB_TENSION_PRE_CTRL

7.7 MB_TensionDecouplingNetworkType01

Brief description The tension controller controls the tension in a web section to the desired command value ([MB_TensionControlLoadCellType02, page 431](#) / [MB_TensionControlLoadCellMatParType02, page 445](#)) 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	Library
IndraMotion MLC 13VRS	ML_TechTensionAdvanced

Tab. 7-12: Reference table of the MB_TensionDecouplingNetworkType01



The use of the function block is subject to license (for details, refer to [Introduction and overview, page 431](#)).

Interface description

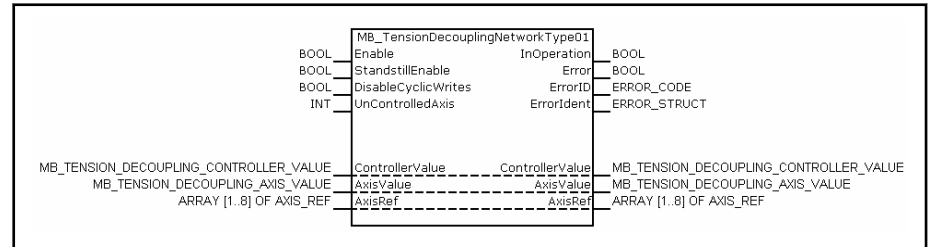


Fig. 7-27: MB_TensionDecouplingNetworkType01 function block

I/O type	Name	Data type	Description
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 468
	AxisValue	MB_TENSION_DECOUPLING_AXIS_VALUE	Structure of the axis inputs (CorrectionValue, AdditivePosInc and AdditivePosDeg) MB_TENSION_DECOUPLING_AXIS_VALUE, page 469
	AxisRef	ARRAY [1..8] OF AXIS_REF	Control axes of the system (without "UnControlledAxis")
VAR_INPUT	Enable	BOOL	Enabling the function block (once, edge-controlled)
	StandstillEnable	BOOL	Closed-loop standstill control enabled if the input is set

ML_TechTensionAdvanced.library

I/O type	Name	Data type	Description
	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
	ErrorID	ERROR_CODE	Diagnostic description in the event of an error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics Troubleshooting, page 468

Tab. 7-13: 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. 7-14: 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 analyzed 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 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.

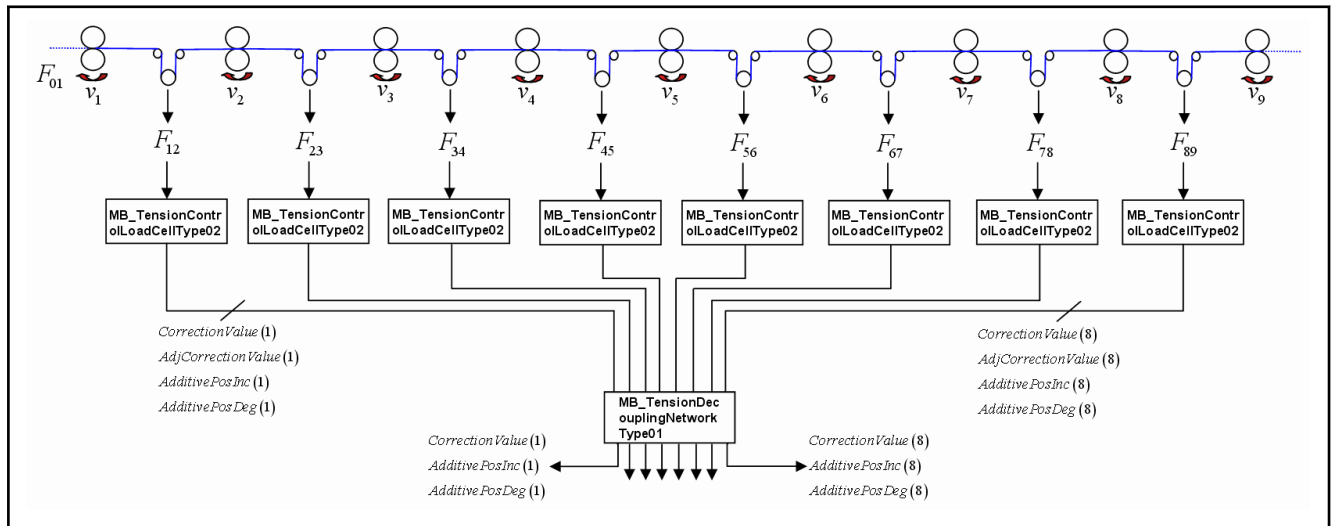


Fig. 7-28: General diagram of the MB_TensionDecouplingNetworkType01 function block

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.

ML_TechTensionAdvanced.library

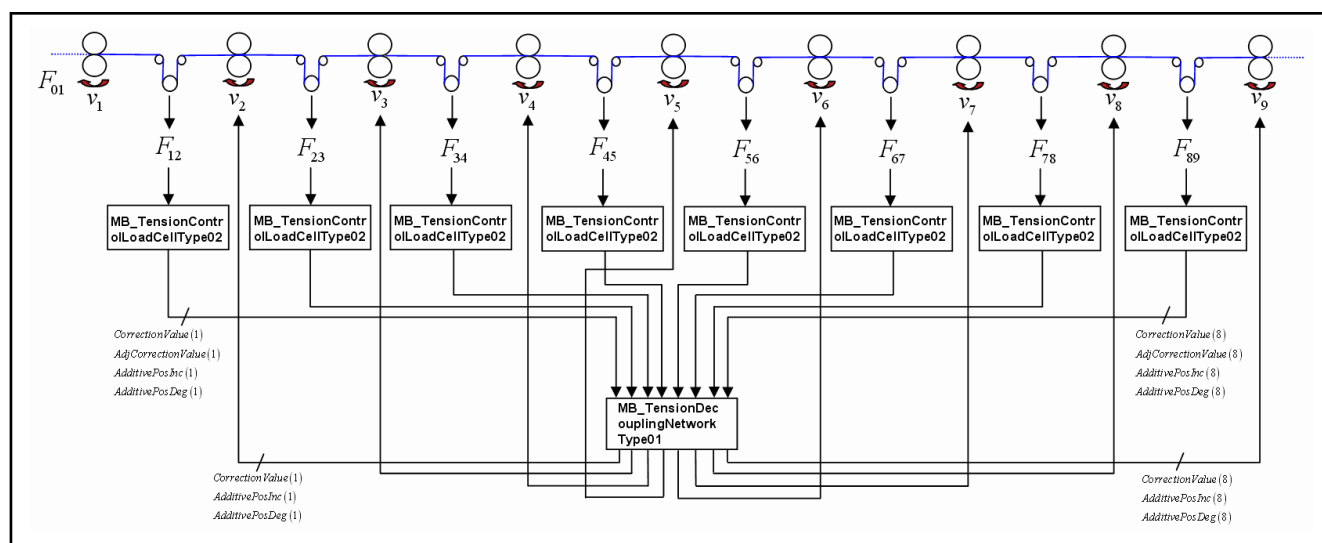


Fig. 7-29: Decoupling web tensions by precontrol exclusively in web direction (leading nips are not required for the closed-loop control and decoupling)

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.

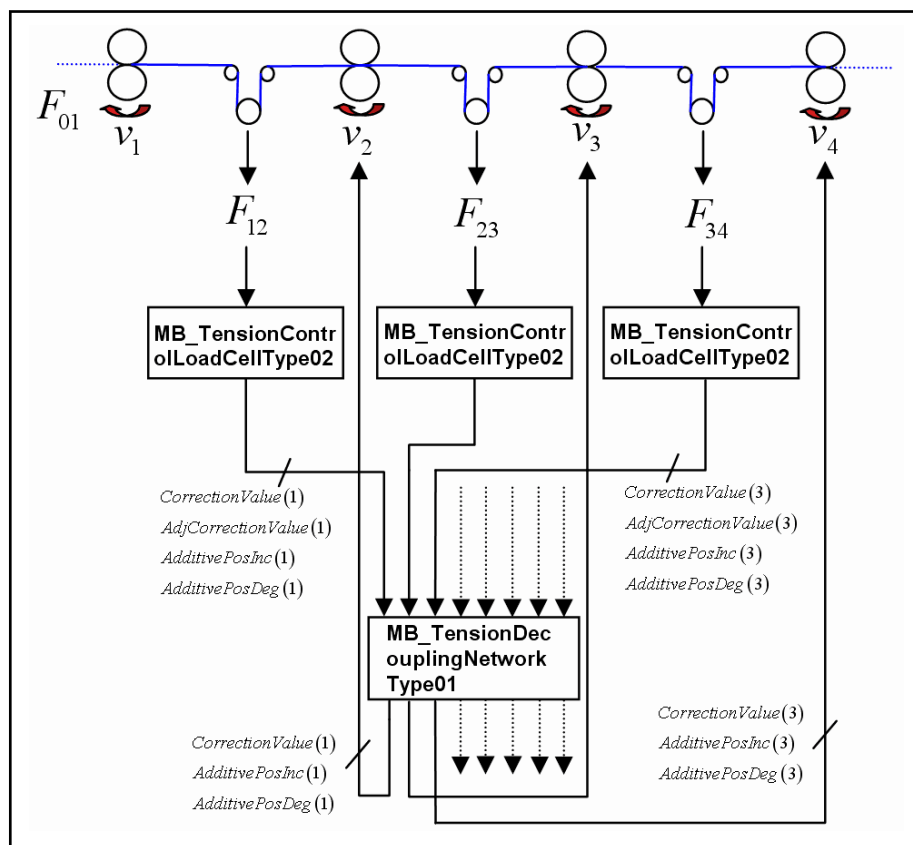


Fig. 7-30: Decoupling web tensions by precontrol exclusively in web direction (representation for three web sections to be controlled)

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).

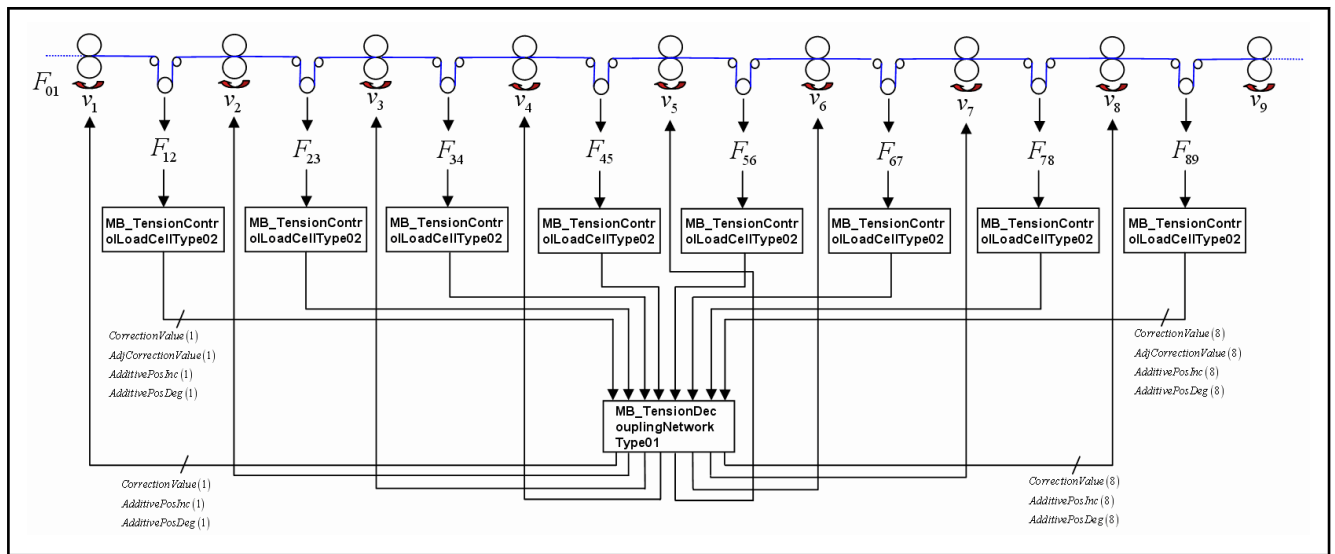


Fig. 7-31: Decoupling of web tensions by precontrol exclusively opposite web direction (following nips are not required for the closed-loop control and decoupling)

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.

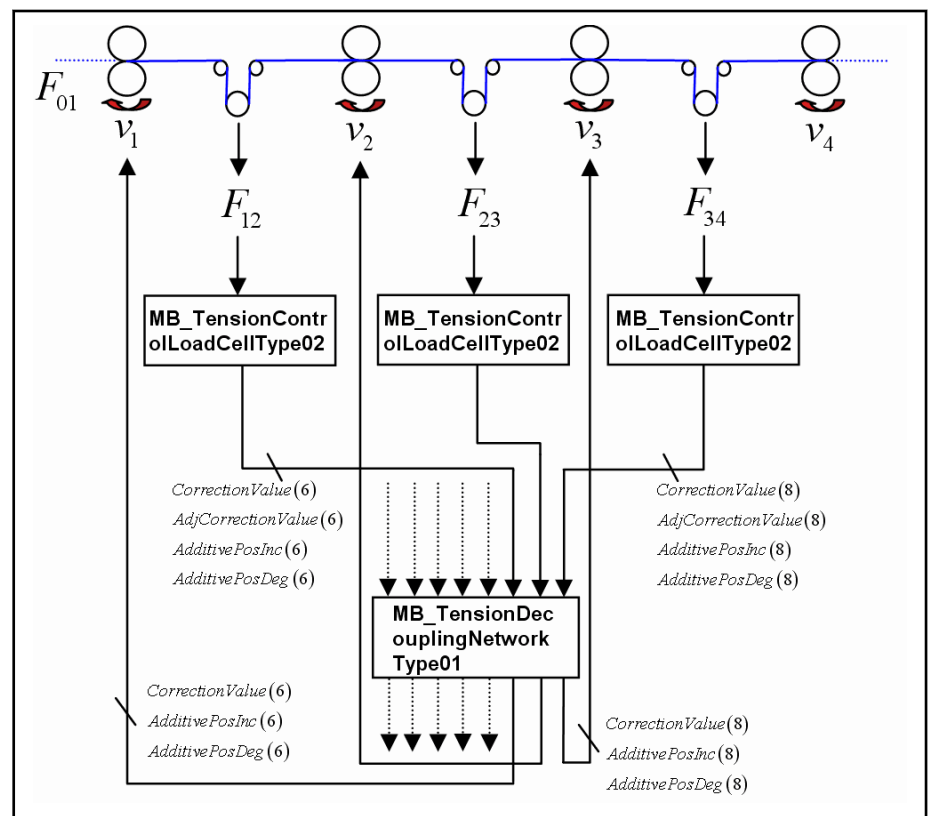


Fig. 7-32: Decoupling web tensions by precontrol exclusively opposite web direction (representation for three web sections to be controlled)

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).

ML_TechTensionAdvanced.library

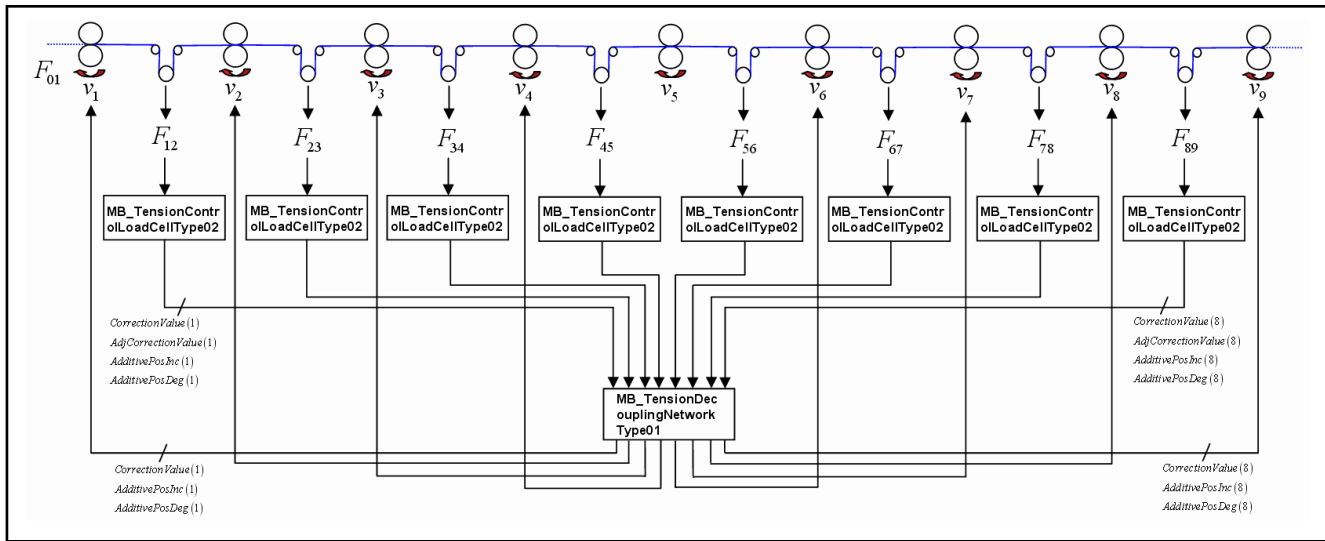


Fig. 7-33: Decoupling of web tensions by precontrol in and opposite web direction (centered nips in figure "v₅" are not required for the closed-loop control and decoupling)

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.

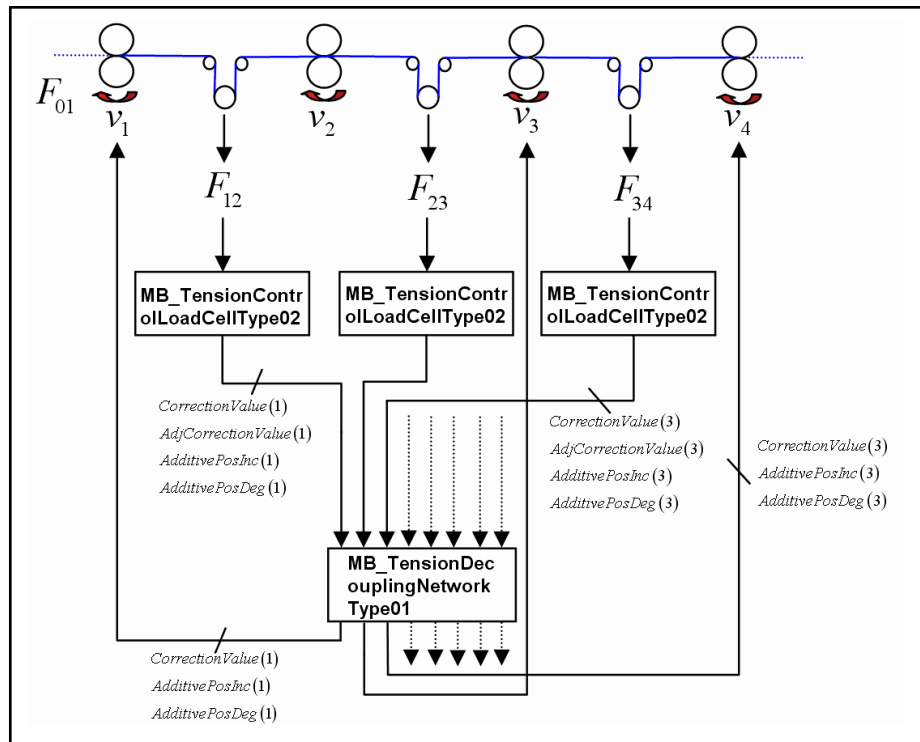


Fig. 7-34: Decoupling web tensions by precontrol in and opposite web direction (representation for three web sections to be controlled)

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.

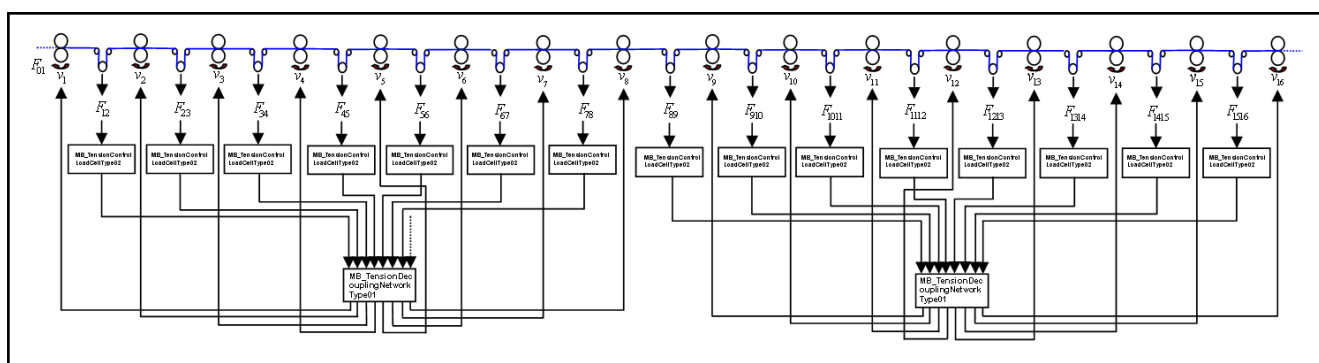


Fig. 7-35: Expansion of the decoupling network to 15 web sections to be controlled at an additional request to a minimum control effort

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 sections1/2) 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 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.

Enable

The "Enable" input activates the function block.



When activating the function block, all controller control values and axis positions are evaluated to avoid position jumps at the axes.

Consequently, this "Enable" input is only to be activated after all web tension controller 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.

ML_TechTensionAdvanced.library

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).

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 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. 7-15: Error codes of the MB_TensionDecouplingNetworkType01

7.8 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

Name	Type	Min. value	Max. value	Description
ControllerCorrectionValue [7]	REAL			CorrectionValue of the seventh web tension controller
ControllerCorrectionValue [8]	REAL			CorrectionValue of the eighth 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. 7-16: Structural description of the MB_TENSION_DECOUPLING_CONTROLLER_VALUE

7.9 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.

ML_TechTensionAdvanced.library

Name	Type	Min. value	Max. value	Description
AxisCorrectionValue [1]	REAL			CorrectionValue of the first axis
AxisCorrectionValue [2]	REAL			CorrectionValue of the second axis
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. 7-17: Structural description of the MB_TENSION_DECOUPLING_AXIS_VALUE

8 RMB_TechWinder.library/MX_TechWinder.lib

8.1 Introduction and overview

Different function blocks are combined in the libraries RMB_TechWinder.compiled (IndraMotion MLC) and MX_TechWinder.lib (IndraMotion MLD) summarizing different functionalities that provide functionalities for winding and unwinding fabric webs (winders).

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 and
- a 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 503	Function block for winders with dancer contains for example: MB_DancerControlType03, page 475 MB_DiameterCalculatorType03, page 489 MB_DiameterMeasurementType01, page 498
MB_WinderDancerCtrlType01, page 503	Extended function block for winders with dancer contains for example: MB_DancerControlType04, page 481 MB_DiameterCalculatorType03, page 489 MB_DiameterMeasurementType01, page 498 1. Standardizing the dancer control variable from diameter calculator 2. Additional diagnostic outputs for the dancer control.
MB_WinderTensionCtrlType02, page 522	The function block for winders with closed-loop tension control or open-loop tension control (without sensor) contains among others: MB_DiameterCalculatorType03, page 489 MB_DiameterMeasurementType01, page 498
MB_WinderTensionCtrlSpeedType01, page 542	The function block for winders with closed-loop tension control and speed correction contains among others: MB_DiameterCalculatorType03, page 489 MB_DiameterMeasurementType01, page 498

Tab. 8-1: Already connected winder function blocks

RMB_TechWinder.library/MX_TechWinder.lib



The preconnected function blocks combine all functions of the individual function blocks and are responsible for the correct connection of the individual function blocks.

Function blocks for individual functionalities

Function block	Description
MB_DancerControlType03 , page 475	Function block to calculate a closed-loop dancer position control for a tension-controlled axis with dancer or winding axis
MB_DancerControlType04 , page 481	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 from diameter calculator 2. Additional diagnostic outputs for the dancer control.
MB_DiameterCalculatorType03 , page 489	Function block to calculate the current diameter for winding axes with velocity adaptation
MB_DiameterMeasurementType01 , page 498	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 561	Function block to calculate the remaining length, the remaining time...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 571	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).

Tab. 8-3: Function blocks for additional functionalities

Library dependencies

IndraMotion MLC control -
RMB_TechWinder

The required libraries are automatically included.

IndraMotion MLD control -
MX_TechWinder

The following libraries are required:

- MX_Technology0x (MPX06 drive firmware) or
- RIL_LoopControl

Function packages for IndraMotion
MLD

For IndraMotion MLD, the drive function package MA "IndraMotion 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.

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, over an analog channel, transferred to the control and processed in the process controller.

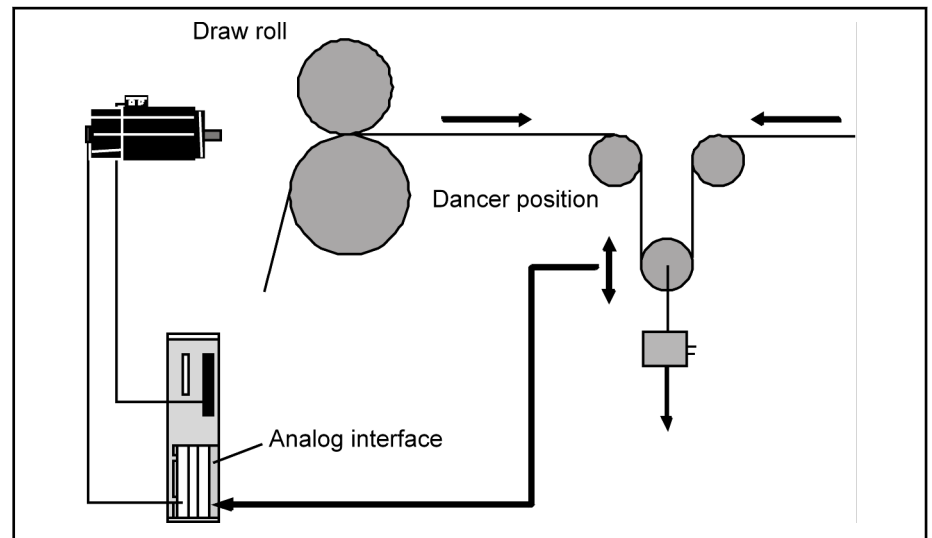


Fig. 8-1: Principle of a closed-loop dancer position control

The dancer position controller is implemented in the function block MB_DancerControlType03 and designed as PID controller (MB_DancerControlType03, page 475).

By setting the controller parameters, the individual controller functions (P value: Proportional part, I value: integral part, 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

RMB_TechWinder.library/MX_TechWinder.lib

No.	P-value	I-value	D-value	Permitted	Note
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.

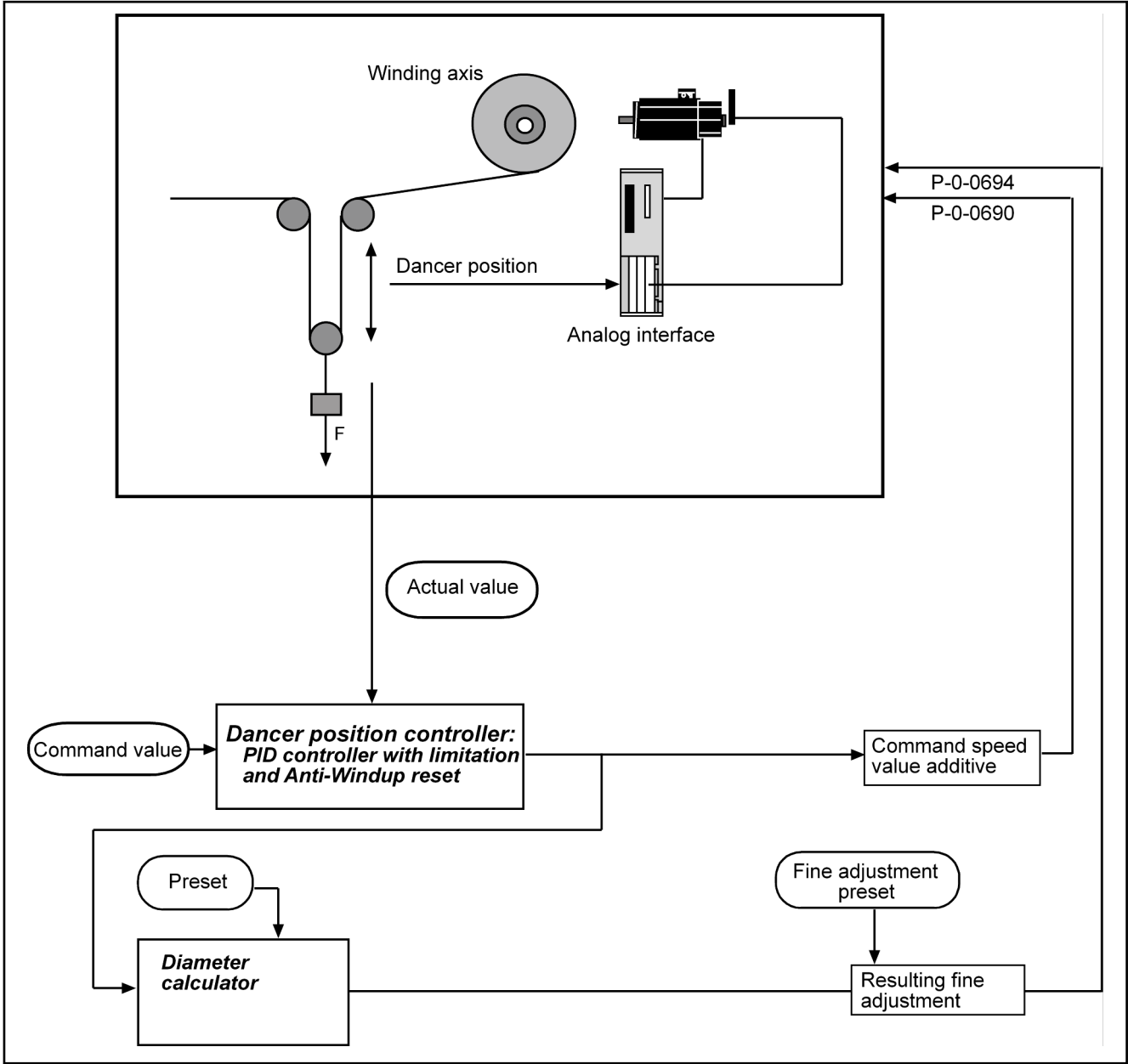


Fig. 8-2: Functional diagram of the dancer position controller

8.2.2 MB_DancerControlType03

Brief description The function block calculates a closed-loop dancer position control for a draw roller with a dancer or winding axis with a dancer on a cyclic basis. 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 "Velocity command value additive, process controller" (interpolation in the drive) or the parameter A-0-2615 "ELS, velocity command value additive, 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
IndraMotion MLC 13VRS	RMB_TechWinder.compiled-library
IndraMotion MLD with MPx07/08/17VRS with MA function package	MX_TechWinder.lib

Tab. 8-5: Reference table of the MB_DancerControlType03

Interface description

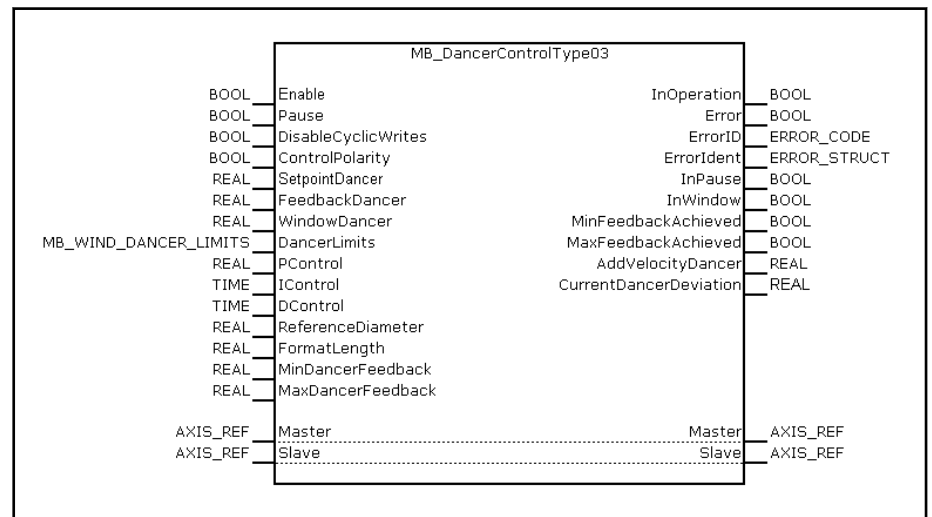


Fig. 8-3: Function block MB_DancerControlType03

I/O type	Name	Type	Comment
VAR_IN_OUT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to slave axis (tension-controlled axis)
VAR_INPUT	Enable	BOOL	Processing of function block 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 edge of "Enable")
	SetpointDancer	REAL	Command value (setpoint) for the closed-loop dancer control

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	FeedbackDancer	REAL	Actual value (feedback) for the dancer position
	WindowDancer	REAL	Windows for monitoring 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
	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 the event of an error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics
	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"

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	AddVelocityDancer	REAL	Speed offset for the winding drive or draw roll drive [rpm]
	CurrentDancerDeviation	REAL	Current control deviation of the dancer controller

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

MB_WIND_DANCER_LIMITS

RMB_TechWinder.library/MX_TechWinder.lib

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 "RepeatLength")
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 "RepeatLength")
VelocityHighLimit	REAL	0.0	n.def	0.0	Continuous	Speed of the master for HighLimitControl [rpm]

Tab. 8-8: Interface description of the data structure of the MB_WIND_DANCER_LIMITS

The data structure MB_WIND_DANCER_LIMITS summarized in the table describes the limiters of the controller control variable shown in the diagram.

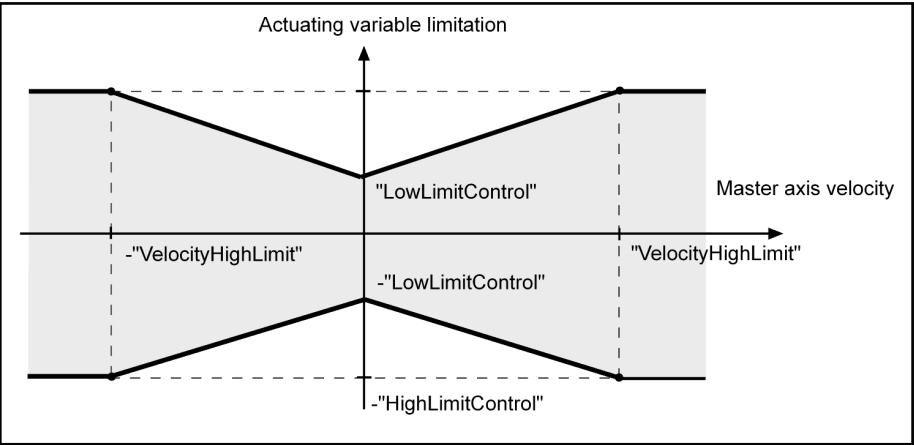


Fig. 8-4: Control variable limitation of the dancer position controller

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 ("format length" / π * "ReferenceDiameter"). This means that at higher velocity, the limitation of the additive velocity of the dancer position controller is automatically higher than at 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 the selected signals of the function block.

Signal context

Signal input	Signal output	Description
Enable	InOperation	Commissioning the function block
	Error	Ready for operation state has been interrupted
Pause	InPause	Stop internal controller process

Tab. 8-9:

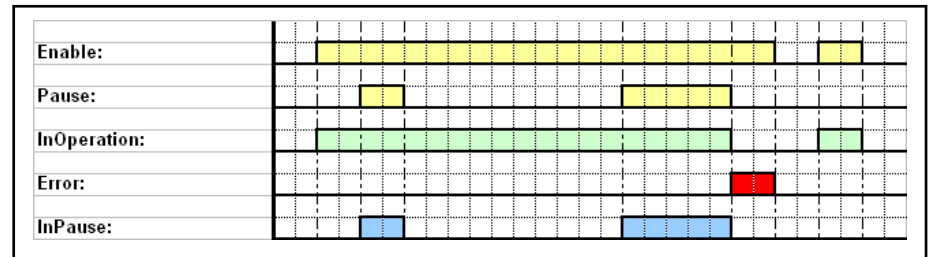


Fig. 8-5: Signal-time diagram for selected inputs and outputs of the function block MB_DancerControlType03

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 controller" (external connection) for a closed-loop dancer position control. The current control deviation "CurrentDancerDeviation" is available for diagnostic purposes. 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 pull roll "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 derivative element (derivative 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 acts upon the speed command value by addition. Thus, the winding

RMB_TechWinder.library/MX_TechWinder.lib

axis rotates faster or slower compared to 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 means of the fine adjustment of the gear. The tension-controlled axis is approximately adjusted by means of the electronic gear. Thus, the dancer position controller should only control small deviations.

Direction of rotation of the tension or winding axis, control direction of the closed-loop dancer control

The direction of rotation of the tension axis or winding axis can be set 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 the 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 numbers of the function block.

The error codes from the "F_RELATED table (16#0170)" are used. Additionally, the error codes of the function block IL_PIDType01 from the RIL_Loop-Control library (refer to the documentation "Rexroth IndraWorks 13VRS Basic Libraries") as well as the MB_GetCyclicParameterHandle (see [chapter 3.11.5 "MB_GetCyclicParameterHandle" on page 135](#)) 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. 8-10: Error codes of the MB_DancerControlType03 function block

Required hardware

- IndraMotion MLC hardware CML65, CML45, CML25
- IndraDrive

RMB_TechWinder.library/MX_TechWinder.lib

Required firmware	- IndraMotion MLC firmware FWA-CMLXX*-MLC-13VRS XX - Device variant, e.g. CML65 - IndraDrive drive firmware
Required software	IndraWorks13VRS for IndraMotion MLC or IndraMotion MLD
Special features for the IndraMotion 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 controller") has to be entered in 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 may have been written to the cyclic sercos channel of the respective drive. The target variable is the control parameter A-0-2615 "ELS, velocity command value additive, process controller".
Special features for the IndraMotion MLD	"Interpolation in the drive" has to be activated in the drive for the dancer controller axis when using the IndraMotion MLD-S control with a superordinate IndraMotion MLC control (Real axes → Corresp. axis → Properties). This setting does not exist when using the IndraMotion 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 controller") 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 must be parameterized (P-0-1367 "PLC configuration" Bit 6 is set or dialog PLC configuration)

8.2.3 MB_DancerControlType04

Brief description	The function block calculates a closed-loop dancer position control for a draw roller with a dancer or winding axis with a dancer on a cyclic basis. 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 "CurrentDancerDeviation" of the dancer position controller is output. The "AddVelocityDancer" output of the function block affects either the parameter P-0-0690 "Velocity command value additive, process controller" (interpolation in the drive) or the parameter A-0-2615 "ELS, velocity command value additive, 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 controlled variable limit standardization or the adaption of the P-gain is executed via an adjustable minimum reference diameter (e.g. minimum sleeve diameter) and does not depend on the format length set in the diameter calculator. Furthermore, diagnostic inputs are available to limit the controlled variable.

Assignment: Target system/library

Target system	Library
IndraMotion MLC 13VRS	RMB_TechWinder.compiled-library

Tab. 8-11: Reference table of the MB_DancerControlType04

RMB_TechWinder.library/MX_TechWinder.lib

Interface description

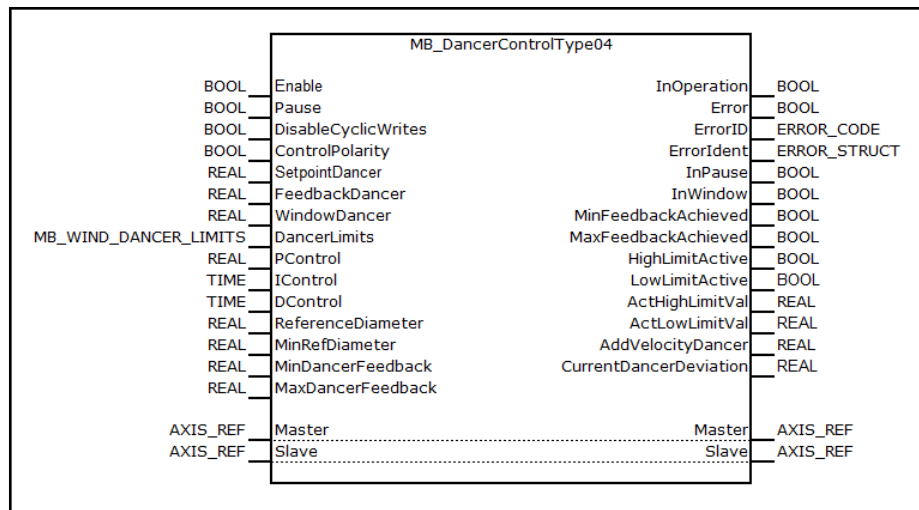


Fig. 8-6: Function block MB_DancerControlType04

I/O type	Name	Type	Comment
VAR_IN_OUT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to slave axis (tension-controlled axis)
VAR_INPUT	Enable	BOOL	Processing of function block 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 edge of "Enable")
	SetpointDancer	REAL	Command value (setpoint) for the closed-loop dancer control
	FeedbackDancer	REAL	Actual value (feedback) for the dancer position
	WindowDancer	REAL	Windows for monitoring 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
	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

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	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 ReferenceDiameter <i>For winders with dancer:</i> - Smallest possible sleeve diameter
	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 the event of an error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics
	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"
	HighLimitActive	BOOL	Speed offset limit "HighLimit " is active
	LowLimitActive	BOOL	Speed offset limit "LowLimit " is active
	ActHighLimitVal	REAL	Current value for upper limit of speed offset
	ActLowLimitVal	REAL	Current value for lower limit of speed offset
	AddVelocityDancer	REAL	Speed offset for the winding drive or draw roll drive [rpm]
	CurrentDancerDeviation	REAL	Current control deviation of the dancer controller

Tab. 8-12: 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.

RMB_TechWinder.library/MX_TechWinder.lib

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"
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"
MinDancerFeed-back	REAL	0.0	MaxDancerFeed-back	0.0	Continuous
MaxDancerFeed-back	REAL	MinDancerFeed-back	n.def	0.0	Continuous

Tab. 8-13: Min./max. and default values of the MB_DancerControlType03 inputs

MB_WIND_DANCER_LIMITS

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. 8-14: Interface description of the data structure of the MB_WIND_DANCER_LIMITS

The data structure MB_WIND_DANCER_LIMITS summarized in the table describes the limiters of the controller control variable shown in the diagram.

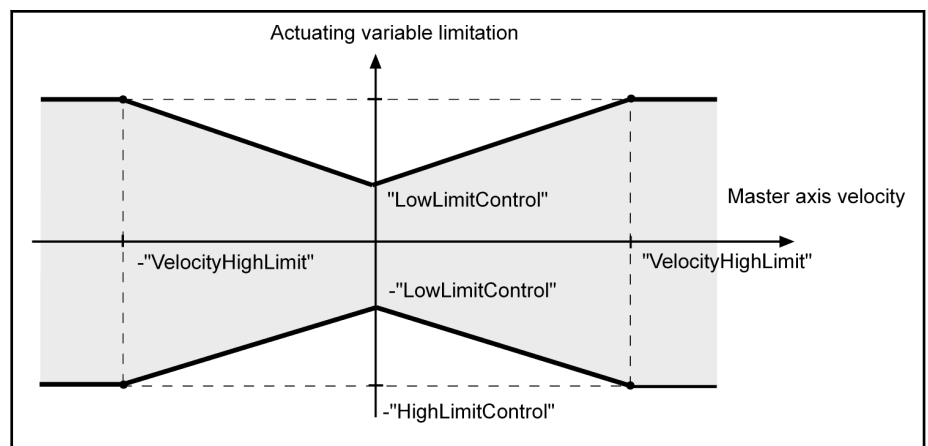


Fig. 8-7: Control variable limitation of the dancer position controller

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 ("Min-RefDiameter" / "ReferenceDiameter"). This means that at higher velocity, the limitation of the additive velocity of the dancer position controller is automatically higher than at 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 controlled variable of the dancer position controller is limited, it is displayed via the binary outputs "HighlimitActive" or "LowLimitActive". The binary outputs "ActHighlimitVal" and "ActLowLimitVal" display the current values for the limit.

Signal-time diagram

The following signal-time diagram represents the selected signals of the function block.

Signal context

Signal input	Signal output	Description
Enable	InOperation	Commissioning the function block
	Error	Ready for operation state has been interrupted
Pause	InPause	Stop internal controller process

Tab. 8-15:

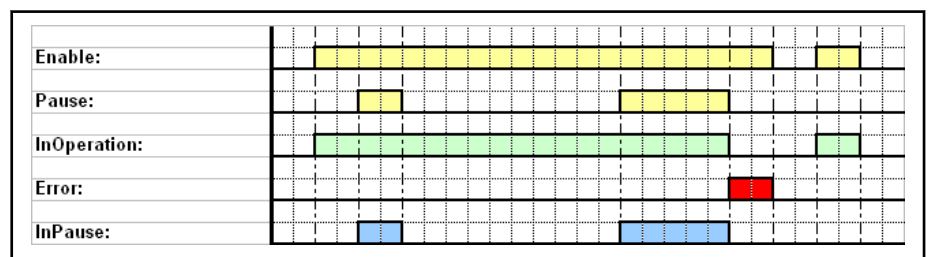


Fig. 8-8: Signal-time diagram for selected inputs and outputs of the function block MB_DancerControlType03

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 "CurrentDancerDeviation" of the

RMB_TechWinder.library/MX_TechWinder.lib

dancer position controller. The output "AddVelocityDancer" of the function block acts on the parameter P-0-0690 "Velocity command value, additive, process controller". The current control deviation "CurrentDancerDeviation" is available for diagnostic purposes. 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 pull roll "ReferenceDiameter" and "MinRefDiameter" have 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) and "MinRefDiameter", the smallest possible sleeve 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 have been 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 acts upon the speed command value by addition. Thus, the winding axis rotates faster or slower compared to the master axis. The position of the dancer is thereby controlled.

Under normal conditions (e.g. no high speed command value jumps), the rough setting of the peripheral velocity of a winder axis is selected via the gear fine adjustment. The rough setting of a tension-controlled axis is selected via the electronic gear. Thus, the dancer position controller should only control small deviations.

Direction of rotation of the tension or winding axis, control direction of the closed-loop dancer control

The direction of rotation of the tension axis or winding axis can be set 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 the 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 numbers of the function block.

The error codes from the "F_RELATED table (16#0170)" are used. Additionally, the error codes of the function block IL_PIDType01 from the RIL_Loop-Control library (refer to the documentation "Rexroth IndraWorks 13VRS Basic Libraries") as well as the MB_GetCyclicParameterHandle (see [chapter 3.11.5 "MB_GetCyclicParameterHandle" on page 135](#)) 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. 8-16: Error codes of the MB_DancerControlType03 function block

Required hardware	- IndraMotion MLC hardware CML65, CML45, CML25 - IndraDrive
Required firmware	- IndraMotion MLC firmware FWA-CMLXX*-MLC-13VRS XX - Device variant, e.g. CML65 - IndraDrive drive firmware
Required software	IndraWorks 13VRS for the IndraMotion MLC
Special features for the IndraMotion 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 controller") has to be entered in 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 may have been written to the cyclic sercos channel of the respective drive. The target variable is the control parameter A-0-2615 "ELS, velocity command value additive, process controller" .

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 roll drive was set. That means that the parameters S-0-0100 "Velocity controller proportional gain" and S-0-0101 "Velocity controller integral time" have to be set according to the application.

RMB_TechWinder.library/MX_TechWinder.lib

Commanding 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" operation 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 also [chapter 8 "RMB_TechWinder.library/MX_TechWinder.lib" on page 471](#)). 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 will be displayed by the "MinFeedbackAchieved" and "MaxFeedbackAchieved" outputs. These values should be within the limit positions of the dancer.

The process variable window ("WindowDancer") is defined. The monitoring process of the dancer command position is parameterized in this window. If the actual dancer position value differs from the position command value ("SetpointDancer"), related to the command value, by more than the specified percentage, the "InWindow" binary output is reset ("InWindow": = FALSE). This allows the superordinate PLC to react to the process accordingly.

Exiting the parameterized window has initially no effect on the function of the dancer position controller. If, for example, the dancer command value "SetpointDancer" is 500.0 and the window size has been 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_DancerCtrlType03:

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_DancerCtrlType04:

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 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. 8-17: Controller parameterization according to Ziegler and Nichols

Finally, the inputs of the MB_WIND_DANCER_LIMITS structure are assigned. The operating range 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.

After having enabled 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 Function blocks for the diameter calculation

8.3.1 Introduction and overview

The MB_DiameterCalculatorType03 function block provides a versatile function block to carry out diameter calculations. It is used in all diameter calculators described in this manual. Therefore, it is described first. The following explains the function block in detail.

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 up for each winding axis. Cyclic output variables are the current winding diameter "CurrentDiameter", 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 "Fine adjustment, gear ratio, process controller" (interpolation in the drive) or the parameter A-0-2605 "ELS, fine adjustment of the gear ratio" (interpolation on the control). This parameter is entered in the respective cyclic sercos channel. Parameterization is carried out in the parameterization mode.

RMB_TechWinder.library/MX_TechWinder.lib

Assignment: Target system/library

Target system	Library
IndraMotion MLC 13VRS	RMB_TechWinder.compiled-library
IndraMotion MLD with MPx07/08/17VRS with MA function package	MX_TechWinder.lib

Tab. 8-18: Reference table of the MB_DiameterCalculatorType03

Interface description

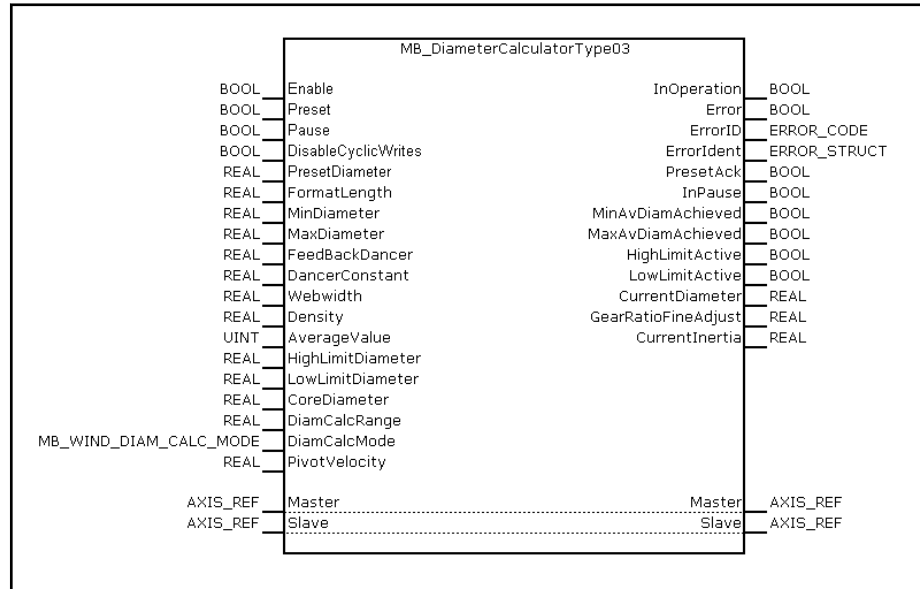


Fig. 8-9: Function block MB_DiameterCalculatorType03

I/O type	Name	Type	Comment
VAR_IN_OUT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to slave axis (tension-controlled axis)
VAR_INPUT	Enable	BOOL	Processing of function block enabled (level-controlled)
	Preset	BOOL	Preset of the current winding diameter
	Pause	BOOL	Diameter calculator 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	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

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	FeedbackDancer	REAL	Actual dancer position value (only for the "Winder with dancer" functionality)!
	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 structure of the dancer (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 averaged. The input is limited to a maximum of 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 0x0000: CALC_DIAM_MANUAL Diameter calculation depending on CalcRange 0x0001: FIRST_REV_AUTO_CALC Diameter calculation is executed after every 5 degrees for the first revolution. Thereafter, it depends on the CalcRange
	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 the event of an error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics
	PresetAck	BOOL	Preset diameter was applied (is set as long as "Preset" is pending)
	InPause	BOOL	Diameter calculator paused
	MinAvDiamAchieved	BOOL	Minimum diameter reached (based on the averaged diameter)
	MaxAvDiamAchieved	BOOL	Maximum diameter reached (based on the averaged diameter)

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	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)
	CurrentDiameter	REAL	Current winding diameter
	GearRatioFineAdjust	REAL	Fine gear adjustment for speed tracing of the drive [%]
	CurrentInertia	REAL	Current total moment of inertia (based on the motor side [kg/m ²])

Tab. 8-19: 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. 8-20: Min./max. and default values of the MB_DiameterCalculatorType03

MB_WIND_DIAM_CALC_MODE

Name	Value	Description
CALC_DIAM_MANUAL	0	Diameter calculation depending on CalcRange
FIRST_REV_AUTO_CALC	1	The diameter is calculated for the 1st revolution after 3 degrees, 6 degrees, 12 degrees, 24 degrees... of the load revolution. Diameter calculation depending on CalcRange.

Tab. 8-21: Elements of the MB_WIND_DIAM_CALC_MODE enumeration type

Signal-time diagram The following signal-time diagram describes the inputs and outputs of the function block.

Signal context

Signal input	Signal output	Description
Enable	InOperation	Commissioning the function block
	Error	Ready for operation state has been interrupted
Pause	InPause	Stop internal calculation
Preset	PresetAck	The diameter calculator is started with a positive edge at the "Preset" input. The value at the "PresetDiameter" input is used as starting value
PresetDiameter		

Tab. 8-22:

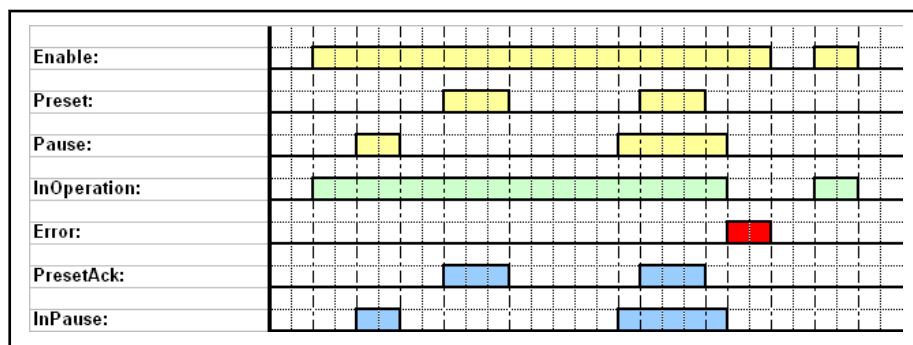


Fig. 8-10: Signal-time diagram for selected inputs and outputs of the MB_DiameterCalculatorType03 function block

Functional description Preset of the diameter

Since the calculated diameter directly affects the winding axis velocity via the resulting fine adjustment, a meaningful 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 winder, the value should be slightly lower and for an unwinder, it should be slightly higher for an unwinder. This allows an easier startup in both cases.

With each new roll, a preset of the winding diameter should be performed once at the start. This is executed via the "Preset" input. The diameter is set to the value of the "PresetDiameter" input. Setting the diameter with the help of "Preset" is always possible without setting the "Pause" input.

After having switched 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 in order to preset the respective variables with meaningful values. If this is not done, no meaningful value can be calculated for the current winding diameter. This can

RMB_TechWinder.library/MX_TechWinder.lib

result in 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 "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 system has been switched on in order to obtain a meaningful 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. When the control is switched on again, a preset can be performed again with this value.

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).

$$d \sim \frac{n_{Ref}}{n_{Winkel}} * F$$

Fig. 8-11: 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 set accordingly.

The calculated winding diameter directly affects the resulting fine adjustment of the gear. Thus, the velocity of the winder is adjusted according to the diameter.

Average value generation

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 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 a maximum of 30. If a higher value is entered, "AverageValue" is limited to 30. The higher the number selected, the more precise is the calculated diameter. However, the accuracy of the diameter is reduced by increasing values for "AverageValue", as this diameter is adjusted due to the "filtering". Thus, the value for "AverageValue" should be selected appropriately. With smaller winding diameters, the variation in the diameter values calculated are higher. Therefore, the value of the filter should not be too high. Otherwise, the delay of the diameter calculated, caused by the averaging, would result in a distinct "lag" 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

With the help of the inputs "Webwidth" and "Density", the moment of inertia of the winder is calculated internally in the diameter calculator and displayed at the "CurrentInertia" output. With the help of the moment of inertia, which 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 the IndraLogic if required.

Preparation for splice

Finally, the inputs "MinDiameter" and "MaxDiameter" can be set. The variables specified here 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 for the 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 numbers of the function block. The error codes from the "F_RELATED table (16#0170)" are used. Additionally, the error codes of the MB_GetCyclicParameterHandle function block (see [chapter 3.11.5 "MB_GetCyclicParameterHandle" on page 135](#)) 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. 8-23: Error codes of the MB_DiameterCalculatorType03 function block

Required hardware

- IndraMotion MLC hardware CML65, CML45, CML25

RMB_TechWinder.library/MX_TechWinder.lib

	• IndraDrive
Required firmware	• IndraMotion MLC firmware FWA-CMLXX*-MLC-13VRS XX - Device variant, e.g. CML65 • IndraDrive drive firmware
Required software	• IndraWorks 13VRS for IndraMotion MLC or IndraMotion MLD
Special features for the IndraMotion 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 should 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 controller") has to be entered in 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 may have been 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 IndraMotion MLD	Interpolation in the drive has to be activated in the drive for the winding axis when using the IndraMotion MLD-S control with a superordinate IndraMotion MLC control (Real axes → Corresp. axis → Properties). This setting does not exist when using the IndraMotion 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 ("Fine adjustment gear ratio, process controller") has to be entered in 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 must 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 should 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, it can be modified using the following PLC code lines: Declaration: <pre>prMaStandstillWdw: POINTER TO REAL; (* for StandstillWindow master MLD *) rTestMaStandstillWdw:REAL := 0.5; (* for StandstillWindow master MLD *)</pre>

Implementation:

```
prMaStandstillWdw := ADR(FunctionBlockInstanceNameDiameterCalculator.rMaStandstillWindow);
```

```
prMaStandstillWdw^ := rTestMaStandstillWdw;
```

The value of this variable is not saved. Thus, it has to be written again when switching on the PLC.

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 controller proportional gain" and S-0-0101 "Velocity controller integral 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 and
- further specifications

Specify 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 on a cyclical basis, they have to be applied as input values at the inputs prior to activation of 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 diameter "LowLimitDiameter" and "HighLimitDiameter". With the help of 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 on a cyclical basis 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 with the aid of two inputs of the function block. "AverageValue" specifies the number of diameter values to be used for averaging. "DiamCalcRange" allows the frequency of the calculation to be influenced, which, in turn, influences the accuracy of the diameter value. Both variables can be manipulated and adjusted during operation since they are read and processed on a cyclical basis.

RMB_TechWinder.library/MX_TechWinder.lib

Further specifications

Depending on the respective application, the inputs "DancerConstant", "FeedBackDancer" (both only for a winder with dancer), "MinDiameter" and "MaxDiameter" are connected. 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" operation mode using the AxisInterface (recommended)
- Use the PLCopen "MC_Power", "MC_GearIn", "MC_Stop" function blocks to command the axis

8.4 Function block to detect an externally measured diameter

8.4.1 Introduction and overview

The MB_DiameterMeasurementType01 function block reads a measured diameter and adapts the velocity of the winding drive accordingly. In this case, it replaces the function block MB_DiameterCalculatorType03. It is used in the function blocks MB_WinderDancerCtrlType01 and MB_WinderTensionCtrlType02.

8.4.2 MB_DiameterMeasurementType01

Brief description

The function block calculates and reads 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 "Fine adjustment, gear ratio, process controller" (interpolation in the drive) or the parameter A-0-2605 "ELS, fine adjustment of the gear ratio" (interpolation on the control). This parameter is entered in the respective cyclic sercos channel. Parameterization is carried out in the parameterization mode.

Assignment: Target system/library

Target system	Library
IndraMotion MLC 13VRS	RMB_TechWinder.compiled-library
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechWinder.lib

Tab. 8-24: Reference table of the MB_DiameterMeasurementType01

Interface description

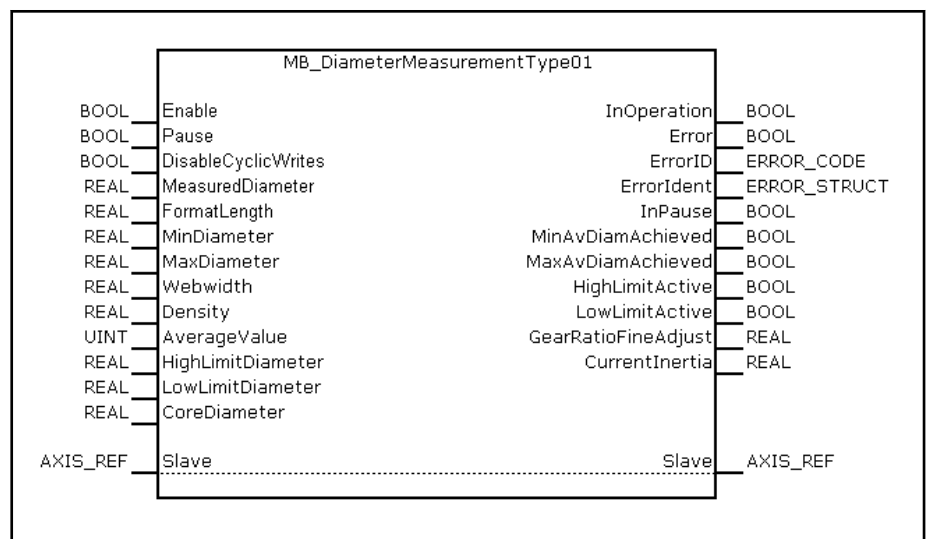


Fig. 8-12: MB_DiameterMeasurementType01 function block

I/O type	Name	Type	Comment
VAR_IN_OUT	Slave	AXIS_REF	Reference to slave axis
VAR_INPUT	Enable	BOOL	Processing of function block 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/dm³]
	AverageValue	UINT	Number of diameter values averaged. The input is limited to a maximum of 30
	HighLimitDiameter	REAL	Upper limit for the diameter [mm] The diameter is limited to this value
	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

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_ CODE	Diagnostic description in the event of an error
	ErrorIdent	ERROR_ STRUCT	Detailed diagnostics
	InPause	BOOL	Dancer controller pausing
	MinAvDiamAchieved	BOOL	Minimum diameter reached (based on the averaged diameter)
	MaxAvDiamAchieved	BOOL	Maximum diameter reached (based on 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 tracing of the drive [%]
	CurrentInertia	REAL	Current total moment of inertia (based on the motor side) [kg/m ²]

Tab. 8-25: 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
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. 8-26: Min./max. and default values of the MB_DiameterMeasurement-Type01 inputs

Signal-time diagram

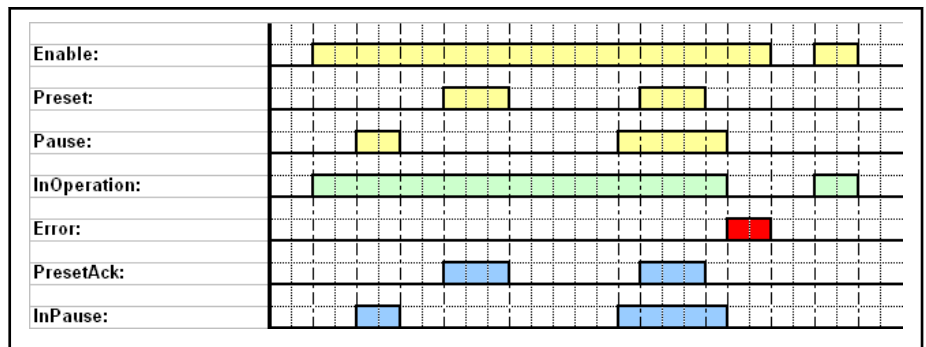


Fig. 8-13: Signal-time diagram for selected inputs and outputs of the function block MB_DiameterMeasurementType01

Functional description

The function block MB_DiameterMeasurementType01 reads a measured diameter value (e.g. by means of 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 and MB_WinderTensionCtrlType02.



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 numbers of the function block. The error codes from the "F_RELATED table (16#0170)" are used. Additionally, the error codes of the MB_GetCyclicParameterHandle function block (see [chapter 3.11.5 "MB_GetCyclicParameterHandle" on page 135](#)) 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
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. 8-27: Error codes of the MB_DiameterMeasurementType01 function block

Required hardware

- IndraMotion MLC hardware CML65, CML45, CML25

RMB_TechWinder.library/MX_TechWinder.lib

	• IndraDrive
Required firmware	• IndraMotion MLC firmware FWA-CMLXX*-MLC-13VRS XX - Device variant, e.g. CML65 • IndraDrive drive firmware
Required software	• IndraWorks 13VRS for IndraMotion MLC or IndraMotion MLD
Special features for the IndraMotion 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 should 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 controller") has to be entered in 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 may have been 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 IndraMotion MLD	Interpolation in the drive has to be activated for the winding axis when using the IndraMotion MLD-S control with a superordinate IndraMotion MLC control (Real axes → Corresp. axis → Properties). This setting does not exist when using the IndraMotion 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 ("Fine adjustment gear ratio, process controller") has to be entered in the cyclic channel between the control and 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 must 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").

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.

RMB_TechWinder.library/MX_TechWinder.lib

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.

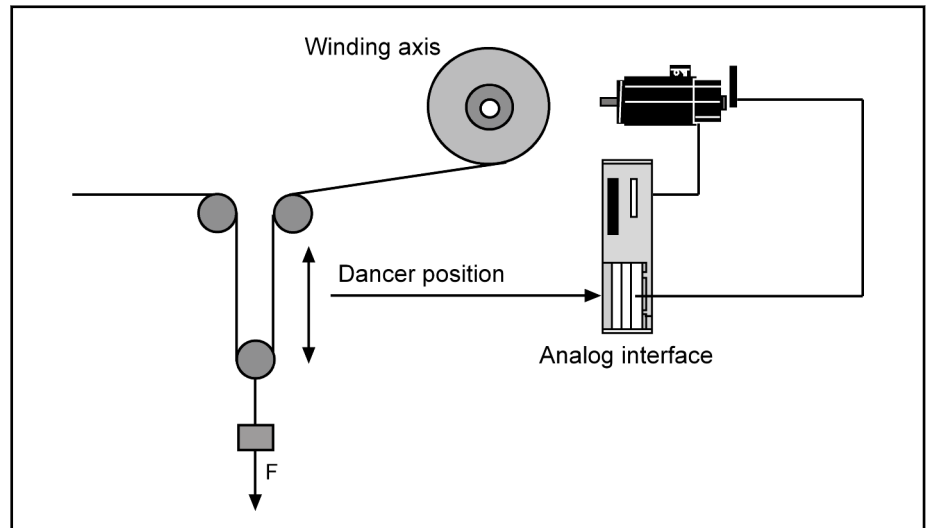


Fig. 8-14: Principle of the diameter calculator with dancer position controller

8.5.2 MB_WinderDancerCtrlType01

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	RMB_TechWinder.compiled-library
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechWinder.lib

Tab. 8-28: Reference table of the MB_WinderDancerCtrlType01

RMB_TechWinder.library/MX_TechWinder.lib

Interface description

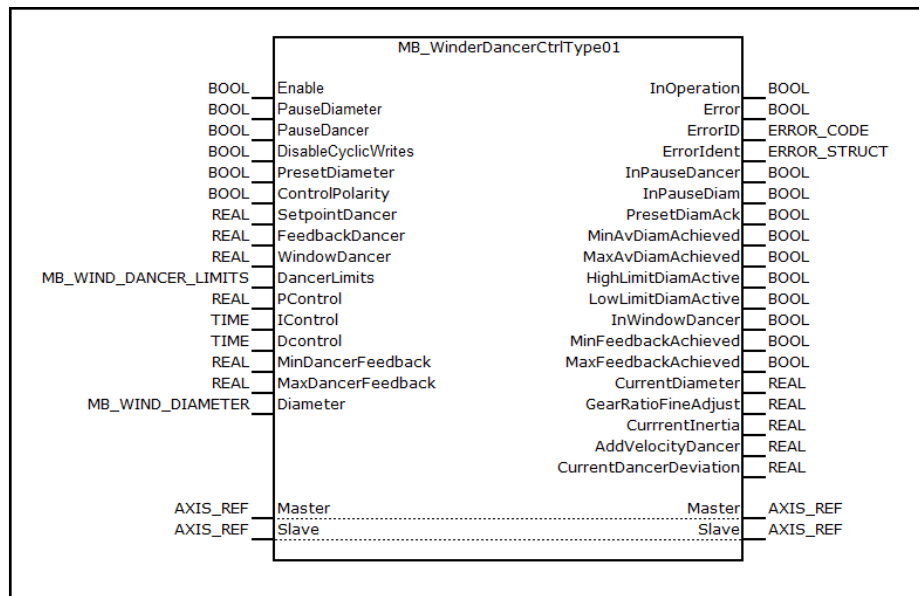


Fig. 8-15: MB_WinderDancerCtrlType01 function block

I/O type	Name	Type	Comment
VAR_IN_OUT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to the slave axis (winding axis)
VAR_INPUT	Enable	BOOL	Processing of function block 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 edge of "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
	PControl	REAL	Value for the Kp-gain of the closed-loop dancer control

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	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
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in the event of an error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics
	InPauseDancer	BOOL	Dancer controller pausing
	InPauseDiam	BOOL	Diameter calculator paused
	PresetDiamAck	BOOL	Preset diameter was applied (is set provided "Preset" is pending)
	MinAvDiamAchieved	BOOL	Minimum diameter reached (based on the averaged diameter)
	MaxAvDiamAchieved	BOOL	Maximum diameter reached (based on 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)
	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 tracing of the drive [%]
	CurrentInertia	REAL	Current moment of inertia of the winding material + mechanics (based on the motor side) [kg m ²]
	AddVelocityDancer	REAL	Speed offset for the winding drive [rpm]
	CurrentDancerDeviation	REAL	Current control deviation of the dancer controller

Tab. 8-29: Interface variables of the MB_WinderDancerCtrlType01 function block

RMB_TechWinder.library/MX_TechWinder.lib

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
DisableCyclic-Writes	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
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. 8-30: Min./max. and default values of the inputs MB_WinderDancerCtrl-Type01

MB_WIND_DIAMETER

Name	Type	Min. value	Max. value	Default value	Effective	Description
PresetVal	REAL	> 0.0	n. def	100.0	Rising edge at "Preset"	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

RMB_TechWinder.library/MX_TechWinder.lib

Name	Type	Min. value	Max. value	Default value	Effective	Description
Density	REAL	0.0	n. def	1.0	Rising edge at "Enable"	Material density [kg/dm³] 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 m²] based on the motor side Is required for the resulting moment of inertia
DancerConstant	REAL	n. def	n. def	0.0	Continuous	Displacement of the dancer, based on the unit of dancer adjustment [mm/-] Is only required with a winder with dancer
MeasuredVal	REAL	0.0	n. def	0.0	Continuous	Measured diameter [mm]
PivotVelocity	REAL	n. def	n. def	0.0	Continuous	Pivot velocity (for relative motion of the winding axis) [mm/s]
DiamCalcMode	MB_WIND_DIAM_CALC_MODE				Rising edge at "Enable"	Diameter calculation mode CALC_DIAM_MANUAL diameter calculation depending on CalcRange FIRST_REV_AUTO_CALC Diameter calculation is performed every 5 degrees for the first revolution and thereafter depending on the CalcRange
DiamSource	BOOL				Rising edge at "Enable"	Diameter determination: FALSE: Diameter is calculated TRUE: Diameter is measured (MeasuredVal)

Tab. 8-31: Interface description of the MB_WIND_DIAMETER data structure

MB_WIND_DANCER_LIMITS

RMB_TechWinder.library/MX_TechWinder.lib

Name	Type	Min. value	Max. value	Default value	Effective	Description
LowLimitControl	REAL	0.0	n. def	0.0	Continuous	Lower limit of the controller output (at zero speed), bipolar [rpm]
HighLimitControl	REAL	0.0	n. def	0.0	Continuous	Upper limit of the controller output (at VelocityHighLimit speed), bipolar [rpm]
VelocityHighLimit	REAL	0.0	n. def	0.0	Continuous	Speed of the master for HighLimitControl [rpm]

Tab. 8-32: Interface description of the data structure of the MB_WIND_DANCER_LIMITS

Functional Description

Diameter calculation

The diameter calculation provides the currently calculated diameter over the function block MB_DiameterCalculatorType03 retrieved on a cyclical basis. The additive fine adjustment of the winding axis is set according to the diameter. The axis is moved faster or slower compared to the reference axis to achieve a constant tension of the web of fabric.

A relative motion of the dancer roll is taken into account in the diameter calculation with the help of the "DancerConstant" input. This dancer constant has to be determined once for this purpose.

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 have a significant influence on 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. When the control is switched on again, a preset can be performed again with this value.

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 in order to achieve a resulting fine adjustment as accurate as possible.



If the preset is not performed, malfunctions can occur. Thus, the preset should be performed 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 having removed 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 specified as a control deviation on an internal PID-controller. The control variable of the controller acts upon the speed command value by addition. Thus, the winding axis rotates faster or slower compared to the master axis. This controls the tension of the dancer (see chapter [chapter 8.6 "Diameter calculator with open-loop tension control or closed-loop tension control" on page 521](#)).

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

The direction of rotation of the tension axis and winding axis can be set with the help of the drive parameter P-0-0-108 "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 the 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.

RMB_TechWinder.library/MX_TechWinder.lib

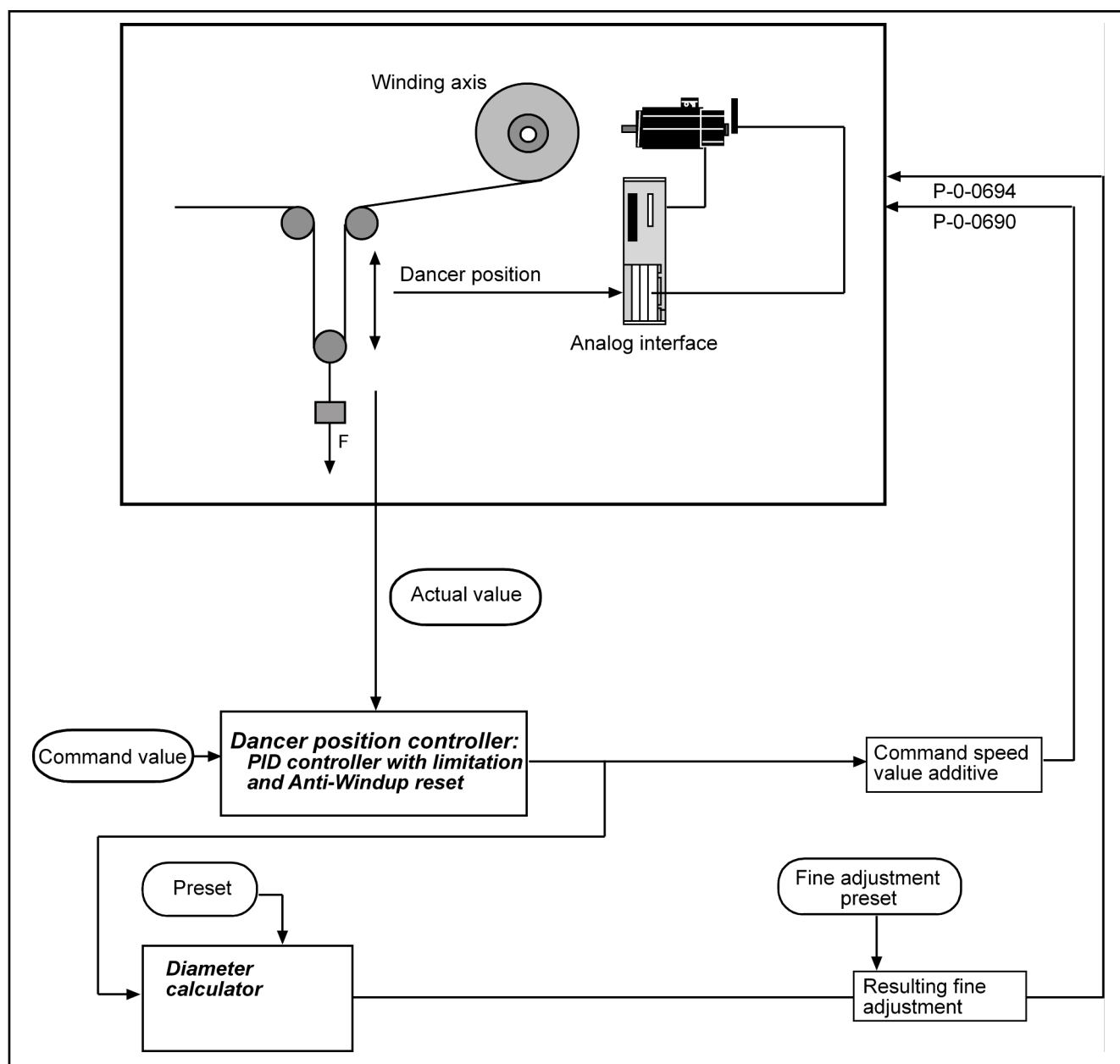


Fig. 8-16: Functional diagram of the diameter calculator with dancer

Troubleshooting

The following table lists and describes the error numbers of the function block. The error codes from the "F_RELATED" table (16#0170) are used. Additionally, the error codes of the function blocks MB_DancerControlType03 (see [chapter 8.2.2 "MB_DancerControlType03" on page 475](#)), MB_DiameterCalculatorType03 (see [chapter 8.3.2 "MB_DiameterCalculatorType03" on page 489](#)), MB_DiameterMeasurementType01 (see [chapter 8.4.2 "MB_DiameterMeasurementType01" on page 498](#)), IL_PIDType01 from the RIL_LoopControl library (refer to the documentation "Rexroth IndraWorks 13VRS Basic Libraries IndraLogic 2G") as well as MB_GetCyclicParameterHandle (see [chapter 3.11.5 "MB_GetCyclicParameterHandle" on page 135](#)) can be output.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1190	16#0020	Error in state machine

Tab. 8-33: Error codes of the MB_WinderDancerCtrlType01 function block

Required hardware	- IndraMotion MLC hardware CML65, CML45, CML25 - IndraDrive
Required firmware	- IndraMotion MLC firmware FWA-CMLXX*-MLC-13VRS XX - Device variant, e.g. CML65 - IndraDrive drive firmware
Required software	IndraWorks 13VRS for IndraMotion MLC or IndraMotion MLD
Required parameterization for the IndraMotion MLC control	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 should 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	Write the parameters P-0-0694 ("Gear ratio fine adjustment, process controller") and P-0-0690 ("Additive velocity command value, process controller") to the cyclic Sercos channel of the respective 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" .
Required parameterization for the IndraMotion MLD control	Interpolation in the drive has to be activated in the drive for the winding axis when using the IndraMotion MLD-S control with a superordinate IndraMotion MLC control (Real axes ▶ Corresp. axis ▶ Properties). This setting does not exist when using the IndraMotion MLD-M control. The scaling type has to be identical for master axis and slave axis. Rotary and preference scaling are recommended. Write the parameters P-0-0694 ("Gear ratio fine adjustment, process controller") and P-0-0690 ("Additive velocity command value, process controller") to the cyclic Sercos channel of the respective 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 must 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 should 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!

RMB_TechWinder.library/MX_TechWinder.lib

8.5.3 MB_WinderDancerCtrlType02

Assignment: Target system/library

Target system	Library
IndraMotion MLC 13VRS	RMB_TechWinder.compiled-library

Tab. 8-34: Reference table of the MB_WinderDancerCtrlType01

Interface description

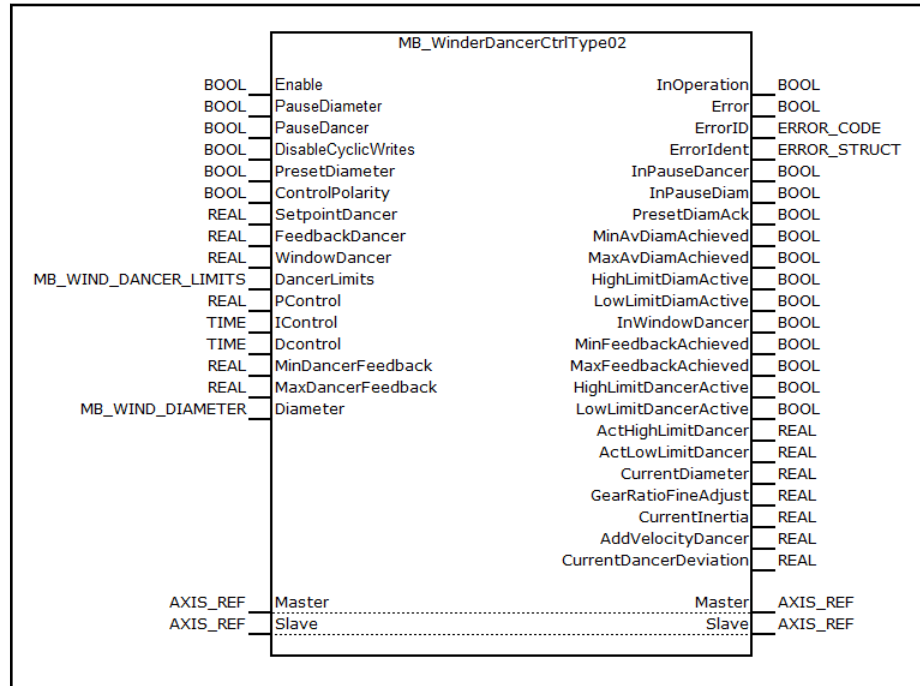


Fig. 8-17: MB_WinderDancerCtrlType02 function block

I/O type	Name	Type	Comment
VAR_IN_OUT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to the slave axis (winding axis)
VAR_INPUT	Enable	BOOL	Processing of function block 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 edge of "Enable")
	SetpointDancer	REAL	Command value (setpoint) for the closed-loop dancer control
	FeedbackDancer	REAL	Actual value (feedback) for the dancer position

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	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
	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
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in the event of an error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics
	InPauseDancer	BOOL	Dancer controller pausing
	InPauseDiam	BOOL	Diameter calculator paused
	PresetDiamAck	BOOL	Preset diameter was applied (is set provided "Preset" is pending)
	MinAvDiamAchieved	BOOL	Minimum diameter reached (based on the averaged diameter)
	MaxAvDiamAchieved	BOOL	Maximum diameter reached (based on 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)
	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"

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	HighLimitDancerActive	BOOL	Limit of speed offset "HighLimit " is active
	LowLimitDancerActive	BOOL	Limit of speed offset "LowLimit" is active
	ActHighLimitDancer	REAL	Current value for upper limit of the speed offset
	ActLowLimitDancer	REAL	Current value for lower limit of the speed offset
	CurrentDiameter	REAL	Current winding diameter
	GearRatioFineAdjust	REAL	Fine gear adjustment for speed tracing of the drive [%]
	CurrentInertia	REAL	Current moment of inertia of the winding material + mechanics (based on the motor side) [kg m ²]
	AddVelocityDancer	REAL	Speed offset for the winding drive [rpm]
	CurrentDancerDevia- tion	REAL	Current control deviation of the dancer controller

Tab. 8-35: 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
DisableCyclic- Writes	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
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. 8-36: Min./max. and default values of the MB_WinderDancerCtrlType01 inputs

MB_WIND_DIAMETER

RMB_TechWinder.library/MX_TechWinder.lib

Name	Type	Min. value	Max. value	Default value	Effective	Description
PresetVal	REAL	> 0.0	n.def	100.0	Rising edge at "Preset"	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] (One 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	REAL	0.0	n.def	0.0	Continuous	Displacement of the dancer, based on the unit of dancer adjustment [mm/-] Is only required with 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]

RMB_TechWinder.library/MX_TechWinder.lib

Name	Type	Min. value	Max. value	Default value	Effective	Description
DiamCalcMode	MB_WIND_DIAM_CALC_MODE				Rising edge at "Enable"	Diameter calculation mode CALC_DIAM_MANUAL diameter calculation depending on CalcRange FIRST_REV_AUTO_CALC Diameter calculation is performed every 5 degrees for the first revolution and thereafter depending on the CalcRange
DiamSource	BOOL				Rising edge at "Enable"	<i>Diameter determination:</i> - FALSE: Diameter is calculated - TRUE: Diameter is measured (MeasuredVal)
MinRefDiameterDancer	REAL	0.0	n.def	100.0	Rising edge at "Enable"	Reference diameter for winder with dancer for Kp standardization and controller variable limit of the dancer, e.g. smallest possible sleeve diameter Is only required with a winder with dancer

Tab. 8-37: Interface description of the MB_WIND_DIAMETER data structure

MB_WIND_DANCER_LIMITS

Name	Type	Min. value	Max. value	Default value	Effective	Description
LowLimitControl	REAL	0.0	n.def	0.0	Continuous	Lower limit of the controller output (at zero speed), bipolar [rpm]
HighLimitControl	REAL	0.0	n.def	0.0	Continuous	Upper limit of the controller output (at VelocityHighLimit speed), bipolar [rpm]
VelocityHighLimit	REAL	0.0	n.def	0.0	Continuous	Speed of the master for HighLimitControl [rpm]

Tab. 8-38: Interface description of the data structure of the MB_WIND_DANCER_LIMITS

Functional description Diameter calculation

The diameter calculation provides the currently calculated diameter over the function block MB_DiameterCalculatorType03 retrieved on a cyclical basis. The additive fine adjustment of the winding axis is set according to the diameter. The axis is moved faster or slower compared to the reference axis to achieve a constant tension of the web of fabric.

A relative motion of the dancer roll is taken into account in the diameter calculation with the help of the "DancerConstant" input. This dancer constant has to be determined once for this purpose.

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 have a significant influence on 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. When the control is switched on again, a pre-set can be performed again with this value.

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 in order to achieve a resulting fine adjustment as accurate as possible.



If the preset is not performed, malfunctions can occur. Thus, the preset should be performed 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 and "Pause" can be disabled after having removed 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 specified as a control deviation on an internal PID-controller. The control variable of the controller acts upon the speed command value by addition. Thus, the winding axis rotates faster or slower compared to the master axis. This controls the tension of the dancer (see chapter [chapter 8.6 "Diameter calculator with open-loop tension control or closed-loop tension control" on page 521](#)).

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

RMB_TechWinder.library/MX_TechWinder.lib

The direction of rotation of the tension axis or winding axis can be set 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 the 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.

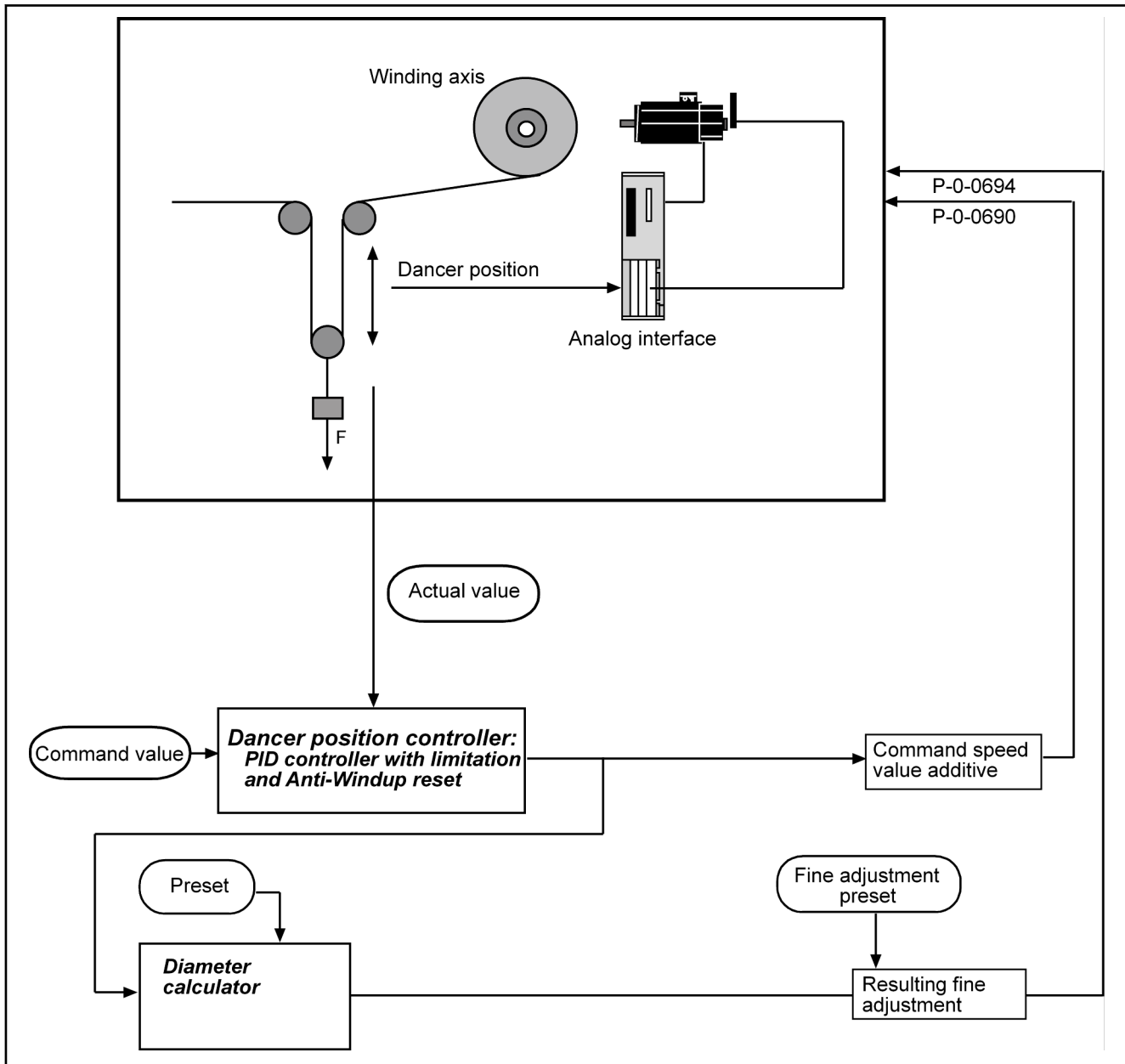


Fig. 8-18: Functional diagram of the diameter calculator with dancer

Error handling

The following table lists and describes the error numbers of the function block. The error codes from the "F_RELATED" table (16#0170)" are used. Additionally, the error codes of the function blocks MB_DancerControlType03 (see [chapter 8.2.2 "MB_DancerControlType03" on page 475](#)), MB_DiameterCalculatorType03 (see [chapter 8.3.2 "MB_DiameterCalculatorType03" on](#)

page 489), MB_DiameterMeasurementType01 (see chapter 8.4.2 "MB_DiameterMeasurementType01" on page 498), IL_PIDType01 from the RIL_LoopControl library (refer to the documentation "Rexroth IndraWorks 13VRS Basic Libraries IndraLogic 2G") as well as MB_GetCyclicParameterHandle (see chapter 3.11.5 "MB_GetCyclicParameterHandle" on page 135) can be output.

ErrorID	Additional1	Additional2	Text
STATE_MACHINE_ERROR	16#1190	16#0020	Error in state machine

Tab. 8-39: Error codes of the MB_WinderDancerCtrlType01 function block

Required hardware	- IndraMotion MLC hardware CML65, CML45, CML25 - IndraDrive
Required firmware	- IndraMotion MLC firmware FWA-CMLXX*-MLC-13VRS XX - Device variant, e.g. CML65 - IndraDrive drive firmware
Required software	IndraWorks 13VRS for IndraMotion MLC
Required parameterization for the IndraMotion MLC control	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 should 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	Write the parameters P-0-0694 ("Gear ratio fine adjustment, process controller") and P-0-0690 ("Additive velocity command value, process controller") to the cyclic sercos channel of the respective drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks.
Interpolation in the control	No parameter may have been 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".

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 controller proportional gain" and S-0-0101 "Velocity controller integral time" have to be set according to the application. <i>This section describes the commissioning of a diameter calculator with dancer. For this purpose, the following steps are reasonable:</i> - Parameterization of the dancer position controller - Parameterization for the diameter calculator - Initial startup of the winding axis
Commanding winding axis	The correct operation mode of the drive and the drive enable of the drive have to be set for the winding axis. <i>There are two ways to set the operation mode and controller enabled:</i> Set the "ModeSyncVel" operation mode using the AxisInterface (recommended)

RMB_TechWinder.library/MX_TechWinder.lib

- 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 in the chapter [8.2 Function Block for Closed-Loop Dancer Position Control](#), page 473.

Parameterization for the diameter calculator

The individual inputs of the diameter calculator MB_DiameterCalculator-Type03 are enabled as described before. A number of 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 have been 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 having activated 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 tension control or 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.

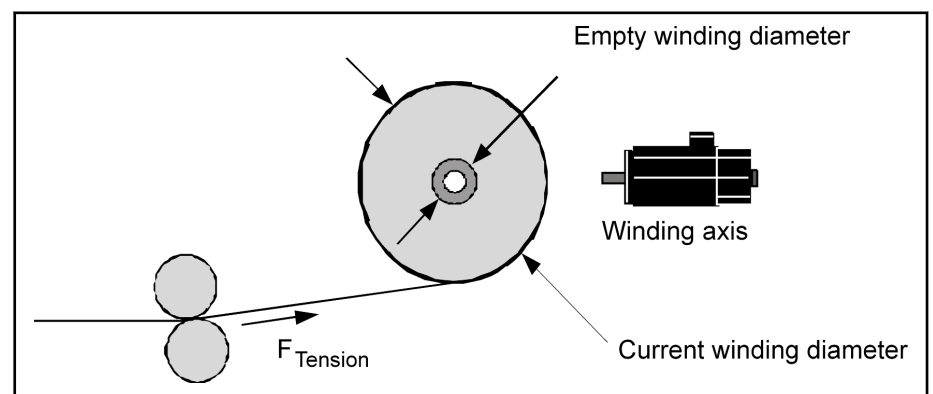


Fig. 8-19: Diameter calculator with open-loop tension control

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 re-

RMB_TechWinder.library/MX_TechWinder.lib

quired 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 with the help of 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.

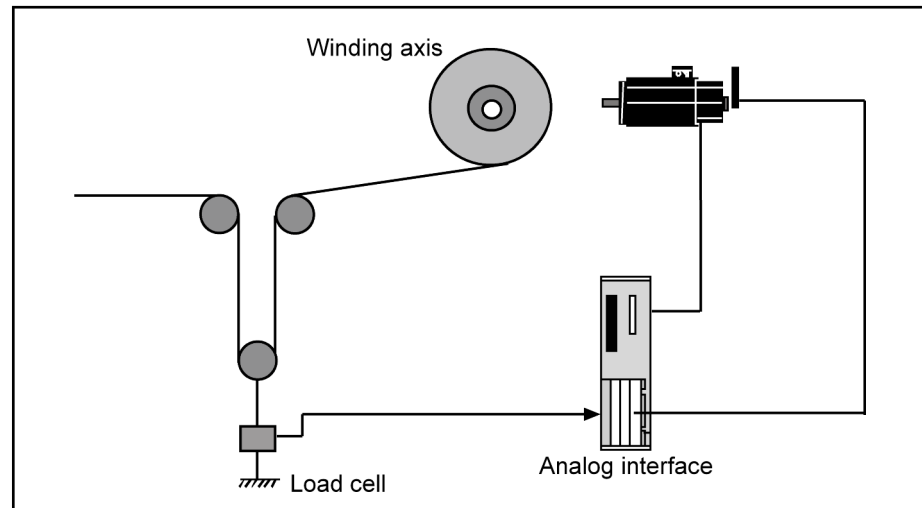


Fig. 8-20: Diameter calculator with closed-loop tension control

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 adjust, process controller" P-0-0690 "Additive velocity command value, process controller" and S-0-0092 "Bipolar torque/force limit value". These parameters have to be stored in the cyclic data container.

At the interpolation in the drive, the 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 and force limit value" apply. Only the parameter S-0-0092 has to be stored in the cyclic data container.

RMB_TechWinder.library/MX_TechWinder.lib

Assignment target system/library

Target system	Library
IndraMotion MLC 13VRS	RMB_TechWinder
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechWinder.lib

Tab. 8-40: Reference table of the MB_WinderTensionCtrlType02

Interface description

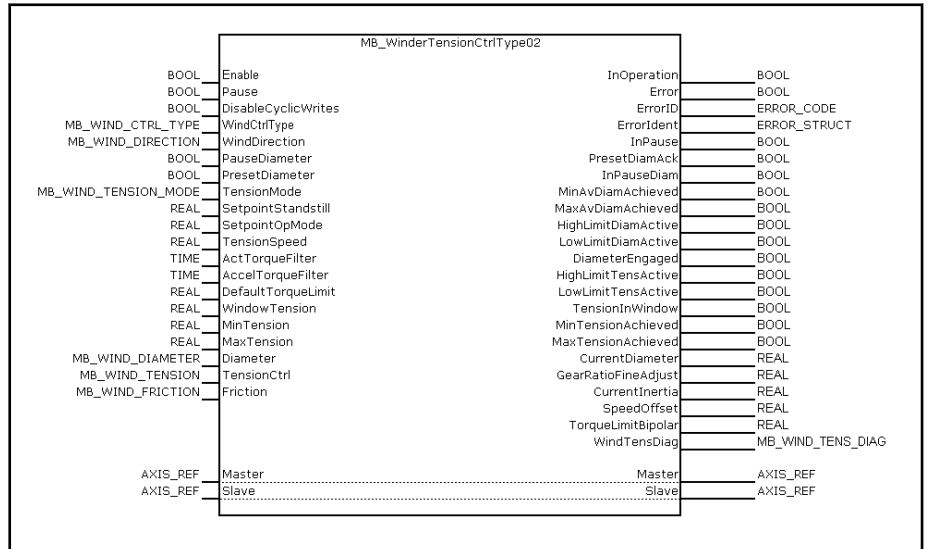


Fig. 8-21: MB_WinderTensionCtrlType02 function block

I/O type	Name	Type	Comment
VAR_IN_OUT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to the slave axis (winding axis)
VAR_INPUT	Enable	BOOL	Processing of function block 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	Winding structure Open-loop tension control (without sensor) OPEN_LOOP (16#0000) Closed-loop tension control (load cell) CLOSED_LOOP (16#0001)
	WindDirection	MB_WIND_DIRECTION	Winding direction Rewinder REWIND (16#0000) Unwinder UNWIND (16#0001)
	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)

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	TensionMode	MB_WIND_TENSION_MODE	Mode to tension the material web Tensioning at constant speed CONST_SPEED Tensioning at constant web velocity CONST_WEB_VEL
	SetpointStandstill	REAL	Specification of the standstill tension [N] (Switching between standstill tension and operating tension by means of the standstill message from the drive).
	SetpointOpMode	REAL	Specification of the operating tension [N] (Switching between standstill tension and operating tension by means of the standstill message from 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 the display with 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
	TensionCtrl	MB_WIND_TENSION	Structure for tension controller
	Friction	MB_WIND_FRICTION	Structure for friction torque
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in the event of an error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics
	InPause	BOOL	Tension controller and diameter calculator pause
	PresetDiamAck	BOOL	Preset diameter was applied (is set as long as "Pre- set" is pending)
	InPauseDiam	BOOL	Diameter calculator pauses
	MinAvDiamAchieved	BOOL	Minimum diameter reached (based on the averaged diameter)
	MaxAvDiamAchieved	BOOL	Maximum diameter reached (based on the averaged diameter)
	HighLimitDiamActive	BOOL	Currently calculated diameter has reached the upper limit (diameter value is discarded)

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	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 is 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 (based on 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

Tab. 8-41: 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
DisableCyclicWrites	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
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

RMB_TechWinder.library/MX_TechWinder.lib

Name	Type	Min. value	Max. value	Default value	Effective
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_FRICTION				Continuous

Tab. 8-42: Min./max. and default values of the MB_WinderTensionCtrlType02 inputs

MB_WIND_CTRL_TYPE

Name	Value	Description
OPEN_LOOP	0	Winding type "open-loop", without sensor
CLOSED_LOOP	1	Winding type "closed-loop", with load cell

Tab. 8-43: Elements of the MB_WIND_CTRL_TYPE enumeration type

MB_WIND_DIRECTION

Name	Value	Description
REWIND	0	Rewinder
UNWIND	1	Unwinder

Tab. 8-44: Elements of the MB_WIND_DIRECTION enumeration type

MB_WIND_TENSION_MODE

Name	Value	Description
CONST_SPEED	0	Tensioning method: Constant tensioning speed
CONST_WEB_VEL	1	Tensioning method: Constant path velocity

Tab. 8-45: Elements of the MB_WIND_TENSION_MODE enumeration type

MB_WIND_DIAMETER

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)

RMB_TechWinder.library/MX_TechWinder.lib

Name	Type	Min. value	Max. value	Default value	Effective	Description
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]

RMB_TechWinder.library/MX_TechWinder.lib

Name	Type	Min. value	Max. value	Default value	Effective	Description
DiamCalcMode	MB_WIND_DIAM_CALC_MODE				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)

Tab. 8-46: Interface description of the MB_WIND_DIAMETER data structure

MB_WIND_TENSION

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	REAL	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
PreControl-Weighting	REAL	0.0	n. def	1.0	Continuous	Scaling of the command value addition
HighLimit	REAL	LowLimit	n. def	0.0	Continuous	Upper limit of the tension controller
LowLimit	REAL	n. def	HighLimit	0.0	Continuous	Lower limit of the tension controller

Tab. 8-47: Interface description of the MB_WIND_TENSION data structure

MB_WIND_FRICTION

Name	Type	Min. value	Max. value	Default value	Effective	Description
StandstillTorque	REAL	0.0	n. def	0.0	Continuous	Friction torque at standstill
MaxTorque	REAL	0.0	n. def	0.0	Continuous	Friction torque at S-0-0091, "Velocity limit value, bipolar"
MinTorque	REAL	0.0	n. def	0.0	Continuous	Minimum friction torque
MinTorqueVelocity	REAL	0.0	n. def	0.0	Continuous	Velocity at minimum friction torque

Tab. 8-48: Interface description of the MB_WIND_FRICTION data structure

MB_WIND_TENS_DIAG

Name	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 [%]

Tab. 8-49: Interface description of the MB_WIND_TENS_DIAG data structure

Signal-time diagram

The following signal-time diagram represents the selected signals of the function block. 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".

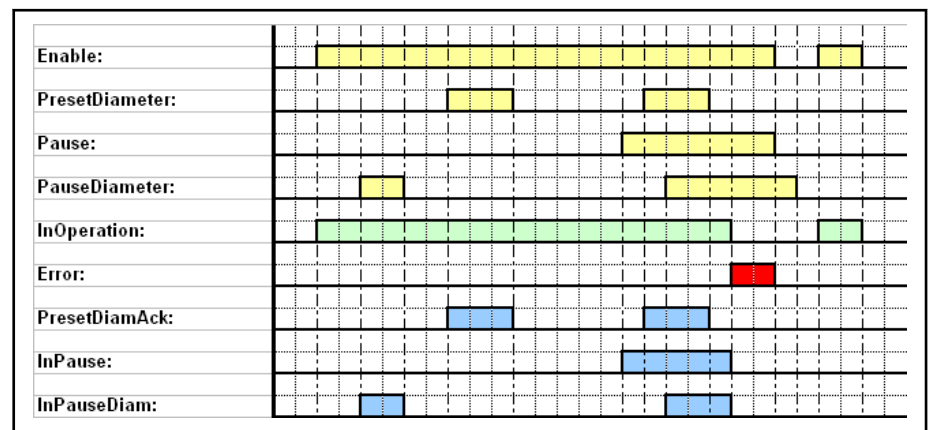


Fig. 8-22: Signal-time diagram for selected inputs and outputs of the function block MB_WinderTensionCtrlType02

Functional Description

Diameter calculator with open-loop tension control

The functional scope of the diameter calculator with open-loop tension control includes:

- Tensioning the untensioned winding material
- Speed-dependent friction compensation
- Compensation of acceleration torque
- Separate parameterization of the operating tension and standstill tension and
- Calculation of diameter-dependent variables
(using the MB_DiameterCalculatorType03 function block)

Functional diagram of the diameter calculator with open-loop tension control

The following functional diagram schematically explains the functions of the diameter calculator with open-loop tension control.

RMB_TechWinder.library/MX_TechWinder.lib

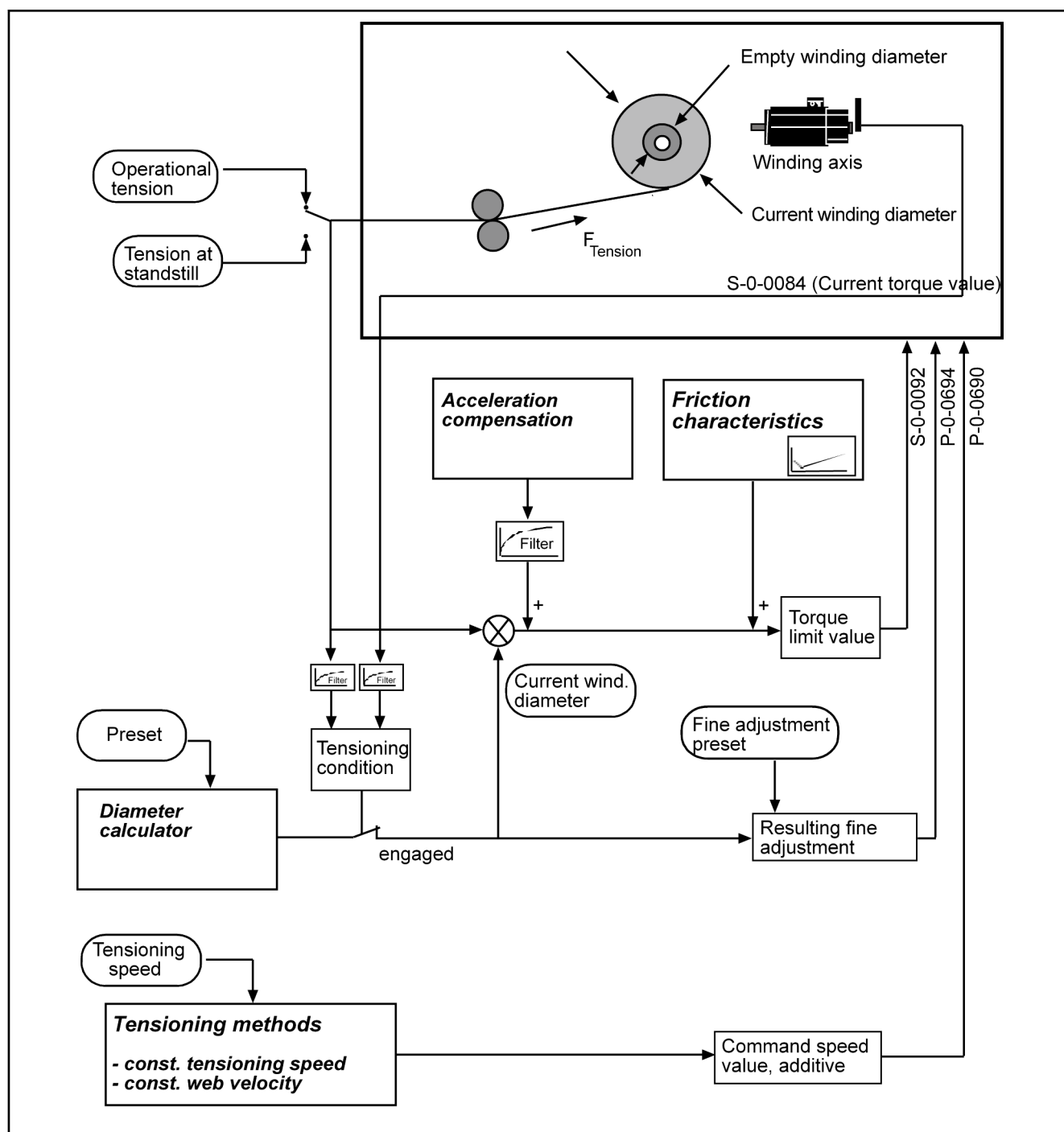


Fig. 8-23: Functional diagram of the winder with open-loop tension control

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
- Speed-dependent friction compensation
- Acceleration torque compensation,
- Separate parameterization of the operating tension and standstill tension

- PI-controller for tension (via IL_PIDType01) and
- Calculating diameter-dependent variables (via MB_DiameterCalculator-Type02)

Functional diagram of the diameter calculator with closed-loop tension control

The following functional diagram schematically explains the functions of the diameter calculator with closed-loop tension control.

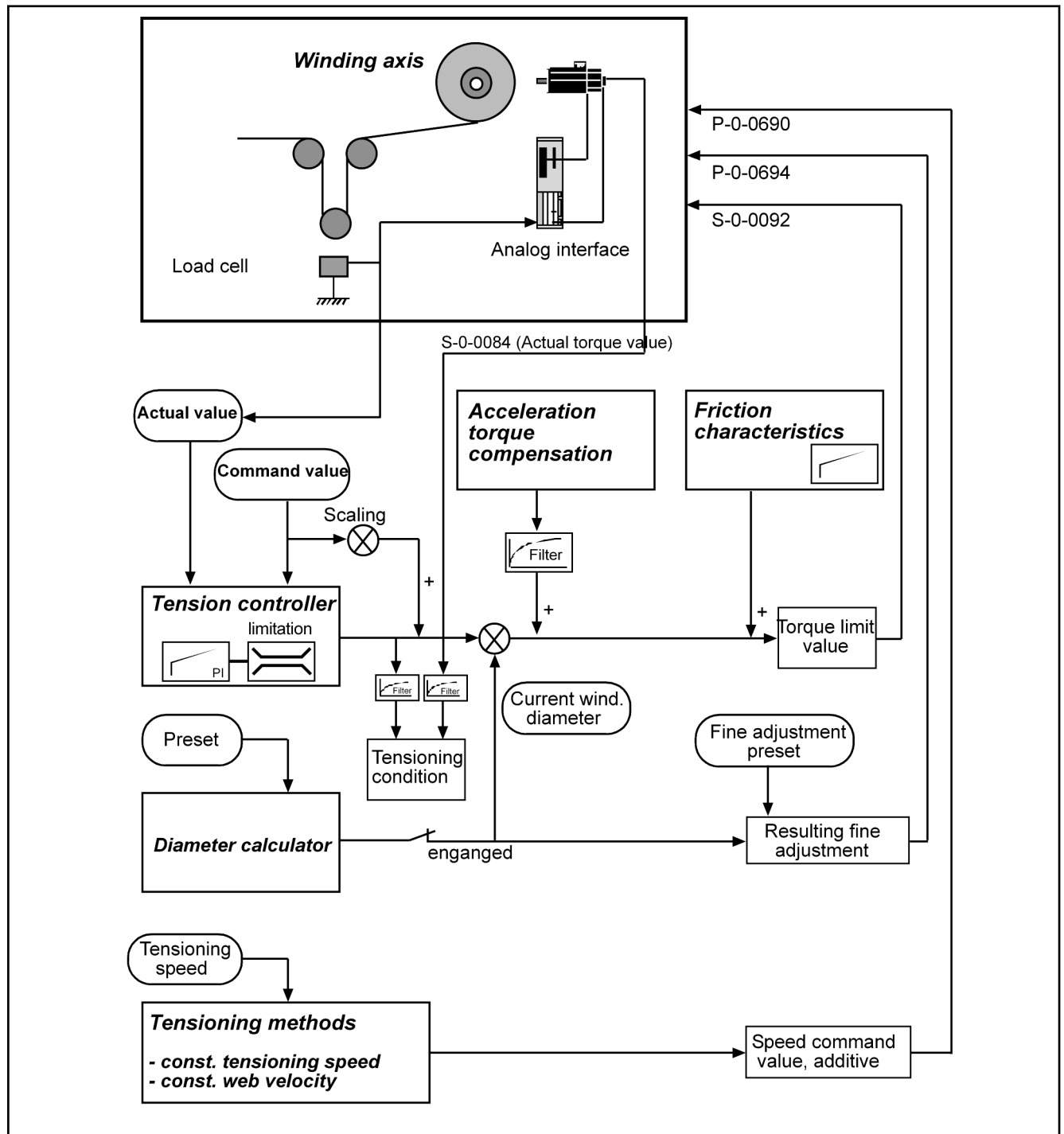


Fig. 8-24: Functional diagram of the winder with closed-loop tension control

RMB_TechWinder.library/MX_TechWinder.lib

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. An internal diameter calculator (function block MB_DiameterCalculatorType03, see [chapter 8.3.2 "MB_DiameterCalculator-Type03" on page 489](#)) is always incorporated in the function block.

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 read 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. When the control is switched on again, a preset can be performed again with this value.



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 be rotating. The winding material is tensioned if the torque command value of the drive exceeds a predefined 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 at constant speed and
- Tensioning at constant web velocity.

When tensioning the winding material at a constant speed, the tensioning speed is maintained on a constant level. Thus, the web velocity increases with an increasing winding diameter.

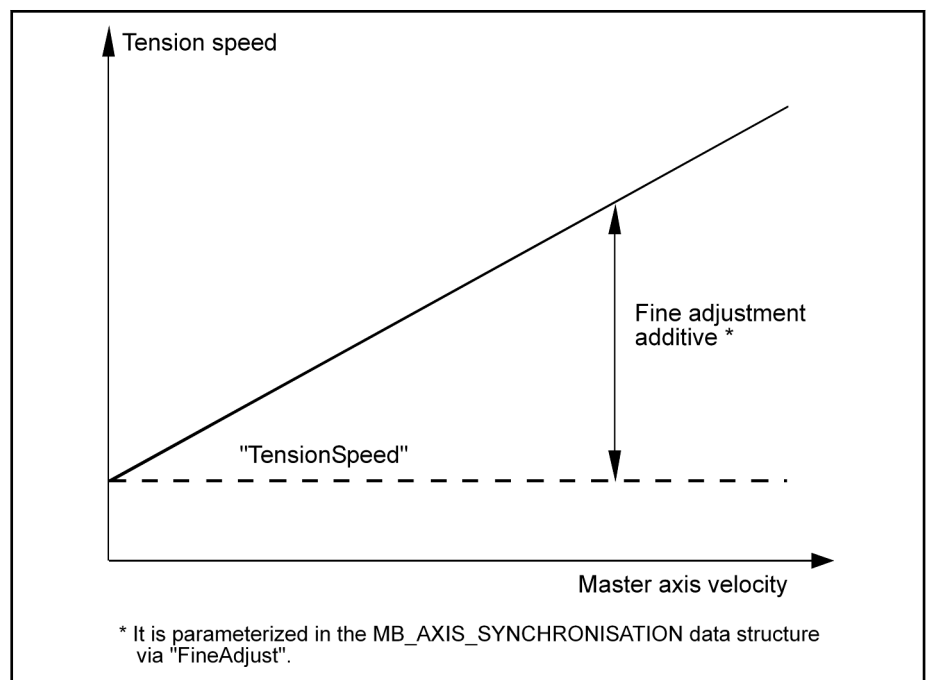


Fig. 8-25: Tensioning at constant tensioning speed

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 has to be transferred according to the function block.

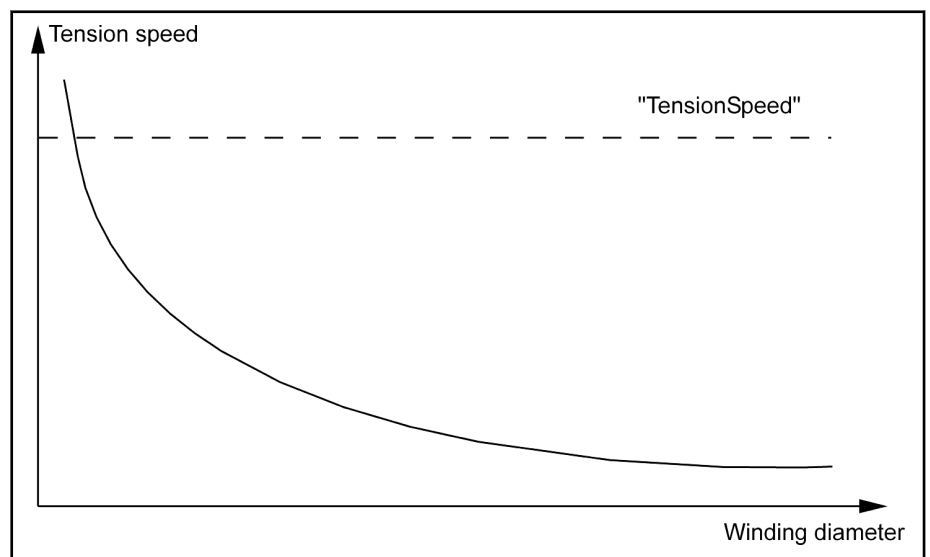


Fig. 8-26: Tensioning with constant web velocity.

Friction compensation

Friction torque compensation is performed using different inputs. 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" or A-0-0033 "Velocity limit value, negative"). This behavior also applies to negative speeds. The curve is mirrored at the y-

RMB_TechWinder.library/MX_TechWinder.lib

axis (friction torque). With this friction torque behavior, the effect of the friction can be counteracted by a compensation.

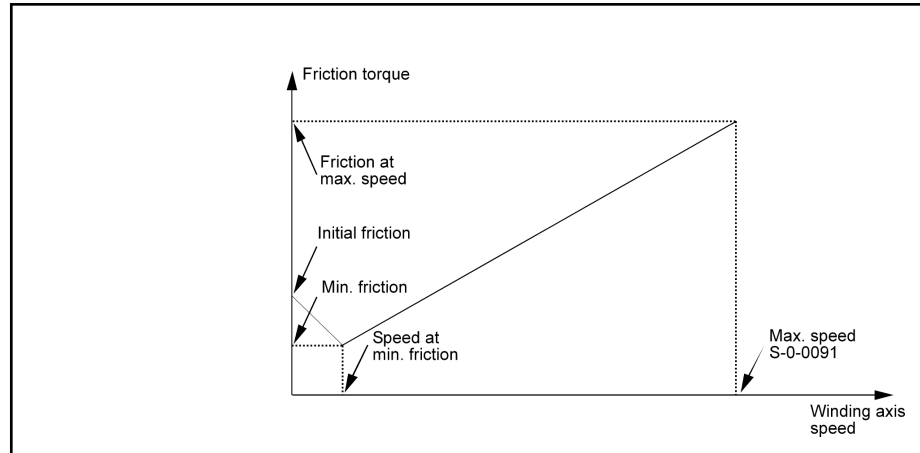


Fig. 8-27: Friction torque compensation

Acceleration torque compensation

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:

- With a rewinder, it should be slightly lower than or equal to the actual winding diameter and
- with 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 roll. 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. 20ms).

Additional leading and lagging of the winding axis

Additional leading or lagging via the fine adjustment of the gear (presetting fine adjustment "FineAdjust") can be set via the structure variable "qrAxisCtrl[AxisNumber].Admin.SyncMode.FineAdjust" of the MB_AXIS_SYNCHRONISATION data structure of the ML_TechInterface library. For a rewinder, a low positive value (e.g. 5%) should be set. A low negative value

Enabling the correction of the torque constant at synchronous motors

should be set for an unwinder. 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.

Troubleshooting

The following table lists and describes the error numbers of the function block. The error codes from the "F_RELATED table (16#0170)" are used. Additionally, the error codes of the function blocks MB_DiameterCalculatorType03 (see [chapter 8.3.2 "MB_DiameterCalculatorType03" on page 489](#)) and IL_PIDType01 from the RIL_LoopControl library (refer to the documentation "Rexroth IndraWorks 13VRS Basic Libraries IndraLogic 2G") as well as MB_GetCyclicParameterHandle (see [chapter 3.11.5 "MB_GetCyclicParameterHandle" on page 135](#)) can be output.

ErrorID	Additional1	Additional2	Description
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

RMB_TechWinder.library/MX_TechWinder.lib

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR	16#1121	16#05	Friction.StandstillTorque < 0.0
INPUT_RANGE_ERROR	16#1121	16#06	Friction.MaxTorque < 0.0
INPUT_RANGE_ERROR	16#1121	16#07	Friction.MinTorque < 0.0
INPUT_RANGE_ERROR	16#1121	16#08	Friction.MinVelocity < 0.0
INPUT_RANGE_ERROR	16#1121	16#09	WindowTension < 0.0
INPUT_RANGE_ERROR	16#1121	16#0A	MinTension < 0.0
INPUT_RANGE_ERROR	16#1121	16#0B	MaxTension < 0.0
INPUT_RANGE_ERROR	16#1121	16#0C	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. 8-50: Error codes of the MB_WinderTensionCtrlType02 function block

Required hardware

- IndraMotion MLC hardware CML65, CML45, CML25
- IndraDrive

Required firmware

- IndraMotion MLC firmware FWA-CMLXX*-MLC-13VRS
XX - Device variant, e.g. CML65
- IndraDrive drive firmware

Required software

- IndraWorks 13VRS for IndraMotion MLC or IndraMotion 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 should 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 controller")** and the **parameter P-0-0690 ("Additive velocity command value, process controller")** and **S-0-0092 "Bipolar torque / force limit value"** have to be entered in the cyclical Sercos channel of the corresponding drive ("UserCmd-

RMB_TechWinder.library/MX_TechWinder.lib

DataX"). 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"** and **S-0-0092 "Bipolar torque and force limit value"**.

Write the parameter **S-0-0092 "Bipolar torque and force limit value"** to the cyclic Sercos channel of the respective drive.

Special features for the MLD

Interpolation in the drive has to be activated in the drive for the winding axis when using the IndraMotion MLD-S control with a superordinate IndraMotion MLC control (Real axes **Corresp. axis ► Properties**). This setting does not exist when using the IndraMotion 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 controller")** and the parameter **P-0-0690 ("Additive velocity command value, process controller")** and **S-0-0092 "Bipolar torque / force limit value"** have to be entered in the cyclical Sercos channel of the corresponding drive ("UserCmd-DataX"). 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 must 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 should 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, it can be modified 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(FunctionBlockInstanceNameDiameterCalculator.rMaStandstillWindow);
```

```
prMaStandstillWdw^ := rTestMaStandstillWdw;
```

The value of this variable is not saved. Thus, it has to be written again when switching on the PLC.

RMB_TechWinder.library/MX_TechWinder.lib

8.6.3 Commissioning the diameter calculator with open-loop tension control 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 controller proportional gain" and S-0-0101 "Velocity controller integral 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 of the friction torque at standstill
4. Determining of the friction torque at maximum speed
5. Determining of the minimum friction torque
6. final settings and
7. initial startup of the winding axis

Commanding 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" operation 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: Unwind, 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

The direction of rotation of the winder can be set with the help of the drive parameter P-0-0-108 "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 the 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-0-0-055, A-0-0029, S-0-0043, S-0-0085).

The following figure shows the principal relations between rewinders and unwinders

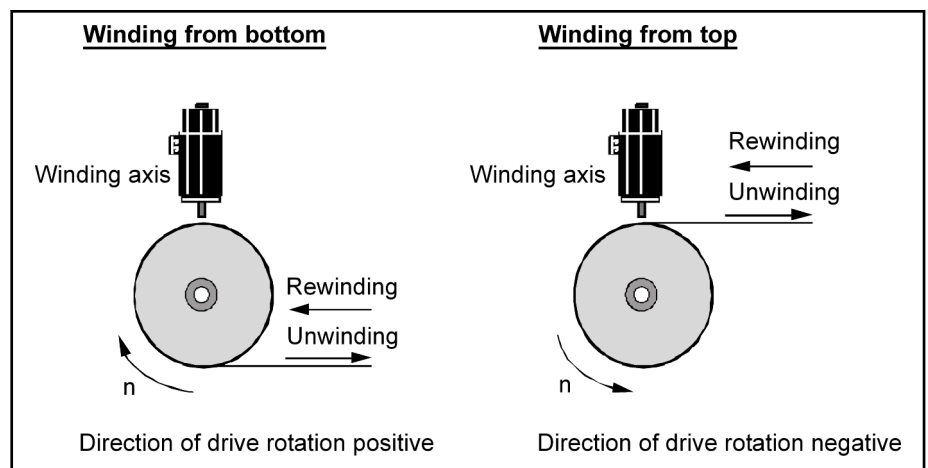


Fig. 8-28: Principles of rewinders and unwinders

Determining 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 "Peak torque/force 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. The axis enabled "MODE_VEL" is canceled (also by using the PLCopen function block "MC_Stop").

The parameter P-0-0109 "Peak torque/force limit" is reset to its previous value (e.g. 400%).

The friction torque setting at standstill is thus completed.

Determining 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. The operating mode "Velocity control" has to be selected for the winding axis (slave axis) (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". Enter the value for the parameter S-0-0091 "Bipolar velocity limit value" according to application and motor data.

3. In the next step, the value of the parameter P-0-0109 "Peak torque/force 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 with the help of the drive parameter S-0-0040 "Velocity feedback value" or the axis parameter A-0-0102 "Actual velocity". The last value

RMB_TechWinder.library/MX_TechWinder.lib

of P-0-0109 "Peak torque/force limit" corresponds to the maximum friction torque during maximum speed.

4. The axis enabled "MODE_VEL" is canceled (also by using the PLCopen function block "MC_Stop").

The parameter P-0-0109 "Peak torque/force limit" is reset to its previous value (e.g. 400%).

Setting the friction torque at maximum speed "MaxTorque" is thus completed.

Determining 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 "Actual velocity value" (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 with constant web velocity
Offset in fine adjustment	In the data structure MB_AXIS_SYNCHRONISATION of the ML_TechInterface, the fine adjustment can be preset via the variable "FineAdjust" with a value depending 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 %
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 "Actual torque/force 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. 8-51: 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 a number of 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 "Bipolar torque/force limit"). 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 Function Block for the Winder Diameter 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 operate 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).

RMB_TechWinder.library/MX_TechWinder.lib

This winder is mainly used for elastic, expandable material.
 The actual tension value is measured directly with the help of 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.

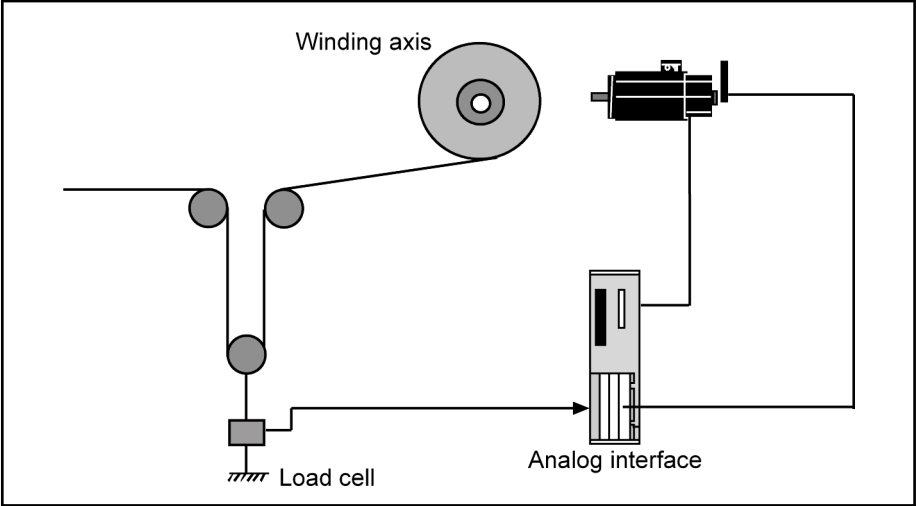


Fig. 8-29: Diameter calculator with closed-loop tension control

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 controller" P-0-0690 "Additive velocity command value, process controller", S-0-0081 "Additive torque/force command value" and S-0-0092 "Bipolar torque/force limit value".

Target system	Library
IndraMotion MLC 13VRS	RMB_TechWinder

Tab. 8-52: Reference table of the MB_WinderTensionCtrlSpeedType01

Interface description

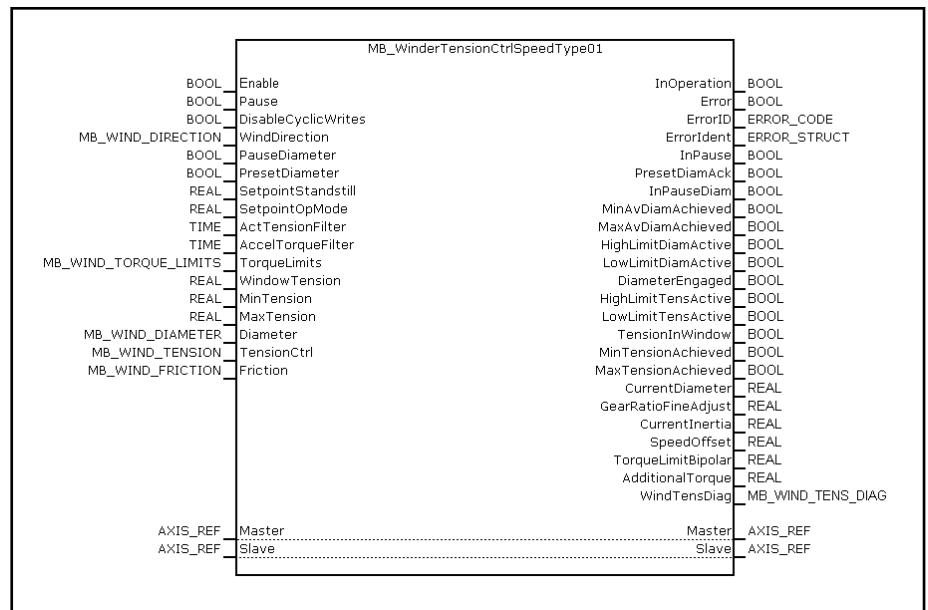


Fig. 8-30: MB_WinderTensionCtrlSpeedType01 function block

I/O type	Name	Type	Comment
VAR_IN_OUT	Master	AXIS_REF	Reference to master axis
	Slave	AXIS_REF	Reference to the slave axis (winding axis)
VAR_INPUT	Enable	BOOL	Processing of function block 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	Winding direction Rewinder REWIND (16#0000) Unwinder UNWIND (16#0001)
	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 by means of the standstill message from the drive).
	SetpointOpMode	REAL	Specification of the operating tension [N] (Switching between standstill tension and operating tension by means of the standstill message from the drive).
	ActTensionFilter	TIME	Filter time constant for actual tension control
	AccelTorqueFilter	TIME	Filter time constant for acceleration torque

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	TorqueLimits	MB_WIND_TORQUE_LIMITS	Structure for bipolar torque limit (S-0-0092)
	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
	TensionCtrl	MB_WIND_TENSION	Structure for tension controller
	Friction	MB_WIND_FRICTION	Structure for friction torque
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in the event of an error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics
	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 (relates to the averaged diameter)
	MaxAvDiamAck	BOOL	Maximum diameter reached (relates 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
	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 is 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 [%]

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Type	Comment
	CurrentInertia	REAL	Current total moment of inertia (based on 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	Diagnostic structure

Tab. 8-53: 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
DisableCyclicWrites	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
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_FRICTION				Continuous

Tab. 8-54: Min./max. and default values of the inputs MB_WinderTensionCtrlSpeedType01

MB_WIND_DIRECTION

RMB_TechWinder.library/MX_TechWinder.lib

Element	Value	Description
REWIND	0	Rewinder
UNWIND	1	Unwinder

Tab. 8-55: Elements of the MB_WIND_DIRECTION enumeration type

MB_WIND_TORQUE_LIMITS

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. 8-56: Interface description of the data structure of the MB_WIND_TORQUE_LIMITS

MB_WIND_DIAMETER

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]
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]
LowLimit	REAL	> 0.0	HighLimit	100.0	Continuous	Lower limit for the diameter [mm]
MinDiameter	REAL	> 0.0	n. def	100.0	Continuous	Minimum diameter [mm]
MaxDiameter	REAL	> 0.0	n. def	2000.0	Continuous	Maximum diameter [mm]

RMB_TechWinder.library/MX_TechWinder.lib

Name	Type	Min. value	Max. value	Default value	Effective	Description
MechanicInertia	REAL	0.0	n. def	0.0	Continuous	Moment of inertia of the mechanics [kg/m²] based on the motor side Is required for the resulting moment of inertia
DancerConstant	REAL	n. def	n. def	0.0	Continuous	Displacement of the dancer, based on the unit of dancer adjustment [mm/-] Is only required with a winder with dancer
MeasuredVal	REAL	0.0	n. def	0.0	Continuous	Measured diameter [mm]
PivotVelocity	REAL	n. def	n. def	0.0	Continuous	Pivot velocity (for relative motion of the winding axis) [mm/s]
DiamCalcMode	MB_WIND_DIAM_CALC_MODE				Rising edge at "Enable"	CALC_DIAM_MANUAL diameter calculation depending on CalcRange FIRST_REV_AUTO_CALC Diameter calculation is performed every 5 degrees for the first revolution and thereafter depending on the CalcRange
DiamSource	BOOL			False	Rising edge at "Enable"	Diameter determination: FALSE: Diameter is calculated TRUE: Diameter is measured (MeasuredVal)

Tab. 8-57: Interface description of the MB_WIND_DIAMETER data structure

MB_WIND_DIAM_CALC_MODE

Element	Value	Description
CALC_DIAM_MANUAL	0	Diameter calculation depending on CalcRange
FIRST_REV_AUTO_CALC	1	Diameter calculation is executed after every 3 degrees for the first revolution. Thereafter, it depends on the CalcRange

Tab. 8-58: Elements of the MB_WIND_DIAM_CALC_MODE enumeration type

MB_WIND_TENSION

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	REAL	0.0	n. def	0.0	Continuous	Tn integral action time of the tension controller

RMB_TechWinder.library/MX_TechWinder.lib

Name	Type	Min. value	Max. value	Default value	Effective	Description
FeedbackVal	REAL	0.0	n. def	0.0	Continuous	Actual value of the tension
PreControl-Weighting	REAL	0.0	n. def	1.0	Continuous	Scaling of the command value addition
HighLimit	REAL	LowLimit	n. def	0.0	Continuous	Upper limit of the tension controller
LowLimit	REAL	n. def	HighLimit	0.0	Continuous	Lower limit of the tension controller

Tab. 8-59: Interface description of the MB_WIND_TENSION data structure

MB_WIND_FRICTION

Name	Type	Min. value	Max. value	Default value	Effective	Description
StandstillTorque	REAL	0.0	n. def	0.0	Continuous	Friction torque at standstill
MaxTorque	REAL	0.0	n. def	0.0	Continuous	Friction torque at S-0-0091, Velocity limit value, bipolar
MinTorque	REAL	0.0	n. def	0.0	Continuous	Minimum friction torque
MinTorqueVelocity	REAL	0.0	n. def	0.0	Continuous	Velocity at minimum friction torque

Tab. 8-60: Interface description of the MB_WIND_FRICTION data structure

MB_WIND_TENS_DIAG

Structure 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 control value [N]

Tab. 8-61: Interface description of the MB_WIND_TENS_DIAG data structure

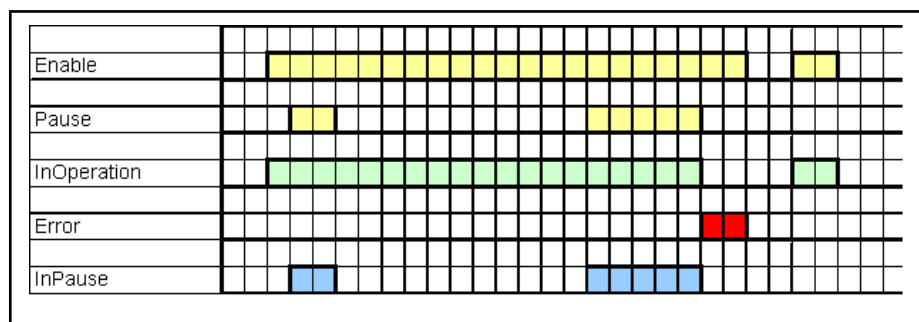
Signal-time diagram

Fig. 8-31: 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: parameter P-0-0694 "Gear ratio fine adjustment, process controller" P-0-0690 "Additive velocity command value, process controller", 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.

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.

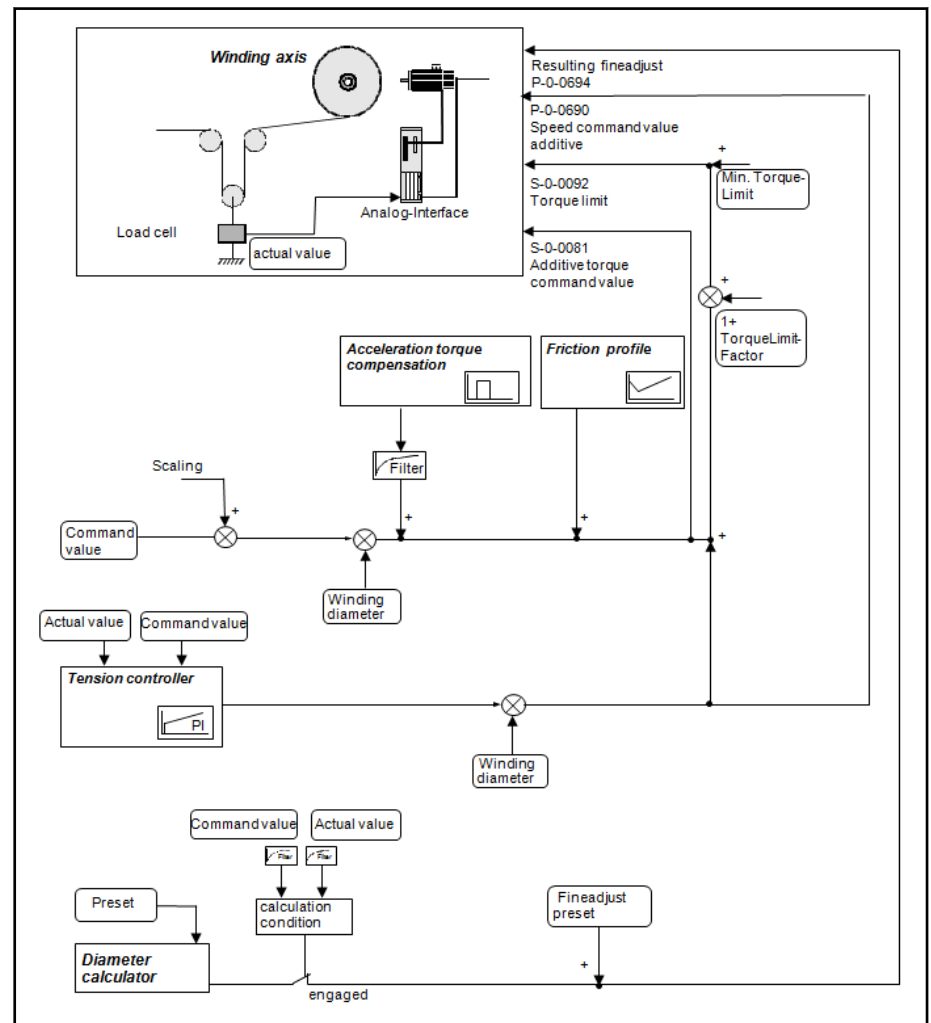


Fig. 8-32: Schematic representation of the control for a winding axis

Direction of rotation of the winding drive, rewinder/unwinder

RMB_TechWinder.library/MX_TechWinder.lib

With the enumeration of the "MB_WindDirection" 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. An internal diameter calculator (function block MB_DiameterCalculatorType03, see [chapter 8.3.2 "MB_DiameterCalculatorType03" on page 489](#)) is always incorporated in the function block.

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 read 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. When the control is switched on again, a pre-set can be performed again with this value.



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 additive speed offset. To avoid the formation of loops in the winding material, the master axis should not be rotating. 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

Friction torque compensation is performed using different inputs. 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" or 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.

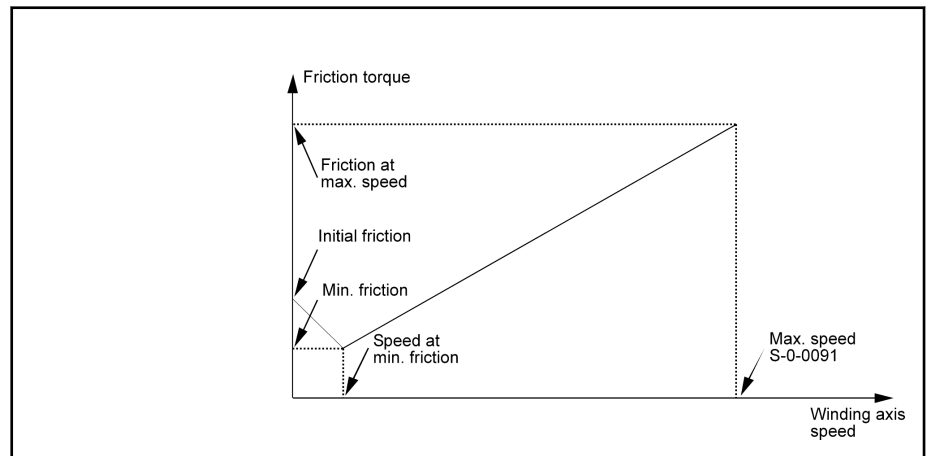


Fig. 8-33: Friction torque compensation

Acceleration torque compensation

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. 8-34: 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. 8-35: 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

RMB_TechWinder.library/MX_TechWinder.lib

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:

- With a rewinder, it should be slightly lower than or equal to the actual winding diameter
- With 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 roll. 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.

Troubleshooting

The following table lists and describes the error codes of the function block. The error codes from the "F_RELATED table (16#0170)" are used. Additionally, the error codes of the function blocks MB_DancerControlType03 (see [chapter 8.2.2 "MB_DancerControlType03" on page 475](#)), MB_DiameterCalculatorType03 (see [chapter 8.3.2 "MB_DiameterCalculatorType03" on page 489](#)), MB_DiameterMeasurementType01 (see [chapter 8.4.2 "MB_Diameter-MeasurementType01" on page 498](#)), IL_PIDType01 from the RIL_LoopControl library (refer to the documentation "Rexroth IndraWorks 13VRS Basic Libraries IndraLogic 2G") as well as MB_GetCyclicParameterHandle (see [chapter 3.11.5 "MB_GetCyclicParameterHandle" on page 135](#)) can be output.

ErrorID	Additional1	Additional2	Description
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"

RMB_TechWinder.library/MX_TechWinder.lib

ErrorID	Additional1	Additional2	Description
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	AxisRef.CntrlNo incorrect
RESOURCE_ERROR	16#001C	16#00xx	Incorrect axis type for axis xx

Tab. 8-62: Error codes of the MB_WinderTensionCtrlSpeedType01 function block

Special features of IndraMotion 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 should 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 controller"), the parameter P-0-0690 ("Additive velocity command value, process controller", S00092 "Bipolar torque/force limit value" and S-0-0081 "Additive velocity command value, process controller" 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", A-0-2615 "ELS, additive velocity command value, process controller", S-0-0092 "Bipolar torque and force limit value" and S-0-0081 "Additive torque/force command value".

Write the parameters S-0-0092 "Bipolar torque and force limit value" and S-0-0081 "Additive torque/force command value" to the cyclic Sercos channel of the respective drive.

The interpolation in the drive must be enabled for the winding axis (Real axis **Corresp. axis** ► **Properties**).

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 controller" and parameter P-0-0690 "Additive velocity command value, process controller" and S-0-0092 "Torque/force limit value bipolar" and S-0-0081 "Torque/force command value, additive" have to be entered in the cyclic Sercos channel of the corresponding drive ("UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks.

An actual existing mechanical gear ratio from the drive axis to the winding axis has to be entered in the parameters S-0-0121 ("Input revolutions of load gear") and S-0-0122 "Output revolutions of load gear". The parameters A-0-0032 "Velocity limit value, positive" and A-0-0033 "Velocity limit value, negative" should be set according to the maximum speed of the winding drive (this value can be read out in the parameter S-0-0091 "Velocity limit value, bipolar").

The standstill window of the reference axis should 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!

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 controller proportional gain" and S-0-0101 "Velocity controller integral 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 of the friction torque at standstill
4. Determining of the friction torque at maximum speed
5. Determining of the minimum friction torque
6. final settings and
7. initial startup of the winding axis

Commanding winding axis *There are two ways to set the operation mode and controller enabled:*

- Set the "ModeSyncVel" operation mode using the AxisInterface (recommended)
- Use the PLCopen "MC_Power", "MC_GearIn", "MC_Stop" function blocks to command the axis

Determining 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 direction of rotation of winding axis

The direction of rotation of the winder can be set with the help of the drive parameter P-0-0-108 "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 the 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-0-0-055, A-0-0029, S-0-0043, S-0-0085).

The following figure shows the principal relations between rewinders and unwinders

RMB_TechWinder.library/MX_TechWinder.lib

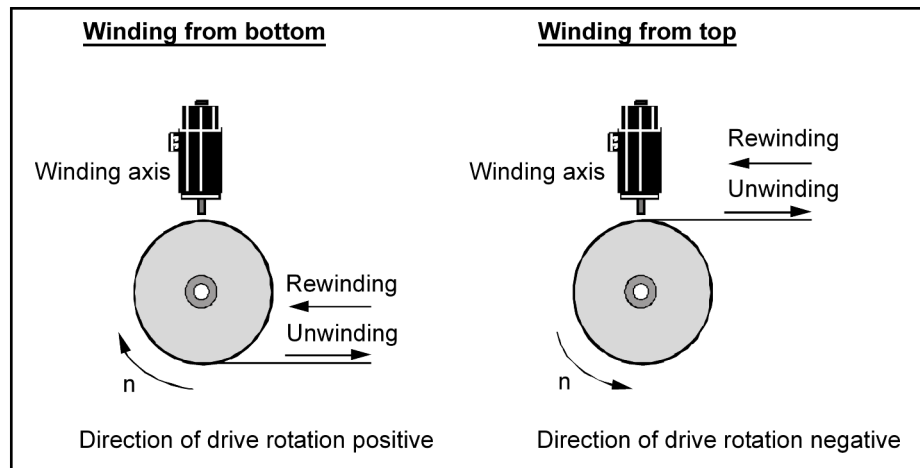


Fig. 8-36: Principles of rewinders and unwinders

Determining 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 the parameter P-0-0109 "Peak torque/force 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. The axis enabled "MODE_VEL" is canceled (also by using the PLCopen function block "MC_Stop").

The parameter P-0-0109 "Peak torque/force limit" is reset to its previous value (e.g. 400%).

The friction torque setting at standstill is thus completed.

Determining 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. The operating mode "Velocity control" has to be selected for the winding axis (slave axis) (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 "Peak torque/force limit" has to be entered according to the application and motor data.

3. In the next step, the value of the parameter P-0-0109 "Peak torque/force 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 with the help of the drive parameter S-0-0040 "Velocity feedback value" or the axis parameter A-0-0102 "Actual velocity". The last value

of P-0-0109 "Peak torque/force limit" corresponds to the maximum friction torque during maximum speed.

4. The axis enabled "MODE_VEL" is canceled (also by using the PLCopen function block "MC_Stop").

The parameter P-0-0109 "Peak torque/force limit" is reset to its previous value (e.g. 400%).

Setting the friction torque at maximum speed "MaxTorque" is thus completed.

Determining 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 "Actual velocity value" (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
Offset in fine adjustment	In the data structure MB_AXIS_SYNCHRONISATION of the ML_TechInterface, the fine adjustment can be preset via the variable "FineAdjust" with a value depending 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 %
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. 8-63: 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 a number of revolutions, the "DiameterEngaged" output has to be set. It

RMB_TechWinder.library/MX_TechWinder.lib

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" from the data structure MB_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. 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" 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").

8.8 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):

Designation	Type	Description
MB_CalcInertiaLimitType02	FB	Calculates the maximum moment of inertia of the winding material on the winder
MB_UnwindMaterialType02	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	FB	It calculates a winding taper characteristic curve for a winder

RMB_TechWinder.library/MX_TechWinder.lib

Designation	Type	Description
MB_WindSpeedControlAdaption-Type01	FB	It calculates for a winder a Kp adaptation for the speed controller
MB_TraverseControlType01	FB	Controls a translatory traversing axis for the winding material smaller than the width of the winder (e.g. wire)

Tab. 8-64: 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" can be used for a speed controller adaptation for example (adaptation of the parameter S-0-0100 "Velocity controller 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
IndraMotion MLC 13VRS	RMB_TechWinder.compiled-library
IndraMotion MLD with MPx07/08/17 with MA function package	MX_TechWinder.lib

Tab. 8-65: Reference table of the MB_CalcInertiaLimitType02

Interface description

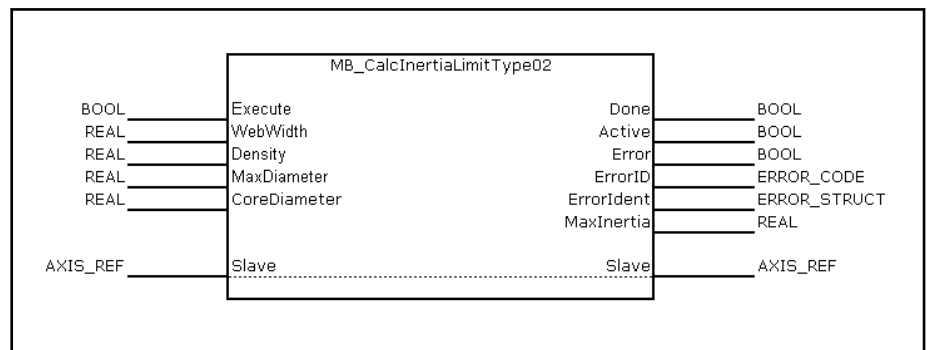


Fig. 8-37: MB_CalcInertiaLimitType02 function block

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/dm³]
	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

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Data type	Description
	Error	BOOL	Error occurred
	ErrorID	ERROR_CODE	Error code
	ErrorIdent	ERROR_STRUCT	Advanced error display
	MaxInertia	REAL	Maximum moment of inertia [kg/m ²] based on the motor side
VAR_IN_OUT	Slave	AXIS_REF	Axis structure of the winding axis

Tab. 8-66: 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. 8-67: 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-controlled 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.

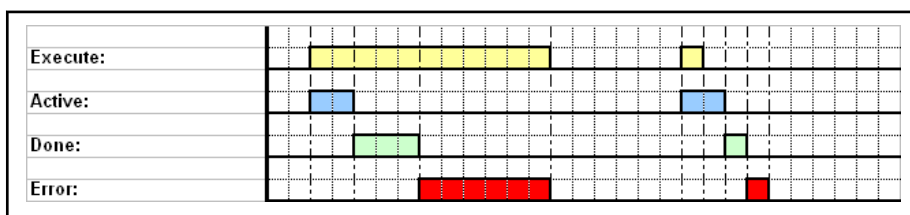


Fig. 8-38: Signal-time diagram for selected inputs and outputs of the MB_CalcInertiaLimitType02 function block

Functional description

The moment of inertia "MaxInertia" is calculated for the specified maximum diameter "MaxDiameter" using the core diameter of the winder "CoreDiameter", the web width "WebWidth" and the web density "Density".

Error handling

The following table lists and describes the error numbers of the function block. 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. 8-68: Error codes of the MB_CalcInertiaLimitType02 function block

Required hardware	- IndraMotion MLC hardware CML65, CML45, CML25 - IndraDrive
Required firmware	- IndraMotion MLC firmware FWA-CMLXX*-MLC-13VRS XX - Device variant, e.g. CML65 - IndraDrive drive firmware
Required software	IndraWorks 13VRS for IndraMotion MLC or IndraMotion MLD
Required parameterization	An actually existing mechanical gear ratio from the drive axis to the winding axis has to be entered into 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 acts as 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
IndraMotion MLC 13VRS	RMB_TechWinder.compiled-library

Tab. 8-69: Reference table of the MB_UnwindMaterialType02

RMB_TechWinder.library/MX_TechWinder.lib

Interface description

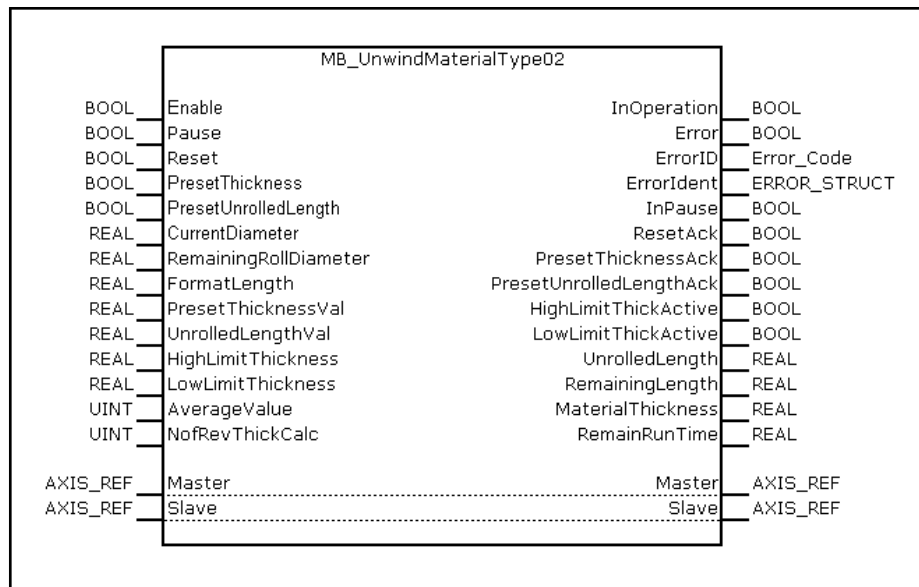


Fig. 8-39: MB_UnwindMaterialType02 function block

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
	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 which is averaged. The input is limited to a maximum of 30
	NofRevThickCalc	UINT	Number of revolutions which is accomplished after the calculation of the material thickness
VAR_OUTPUT	InOperation	BOOL	Function block is working
	Error	BOOL	Error occurred
	ErrorID	ERROR_CODE	Error code
	ErrorIdent	ERROR_STRUCT	Advanced error display
	InPause	BOOL	Material calculation paused

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Data type	Description
	ResetAck	BOOL	Reset completed
	PresetThicknessAck	BOOL	Material thickness preset
	PresetUnrolledLength-Ack	BOOL	Unrolled length preset
	HighLimitThickA-chieved	BOOL	The currently calculated material thickness has achieved the upper limit (value will be discarded)
	LowLimitThickA-chieved	BOOL	The currently calculated material thickness has 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 (based on the remaining roll diameter) [s]
VAR_IN_OUT	Master	AXIS_REF	Axis structure of the reference axis
	Slave	AXIS_REF	Axis structure of the winding axis

Tab. 8-70: 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:

RMB_TechWinder.library/MX_TechWinder.lib

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. 8-71: Min./max. and default values of the MB_UnwindMatrrialType02 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.

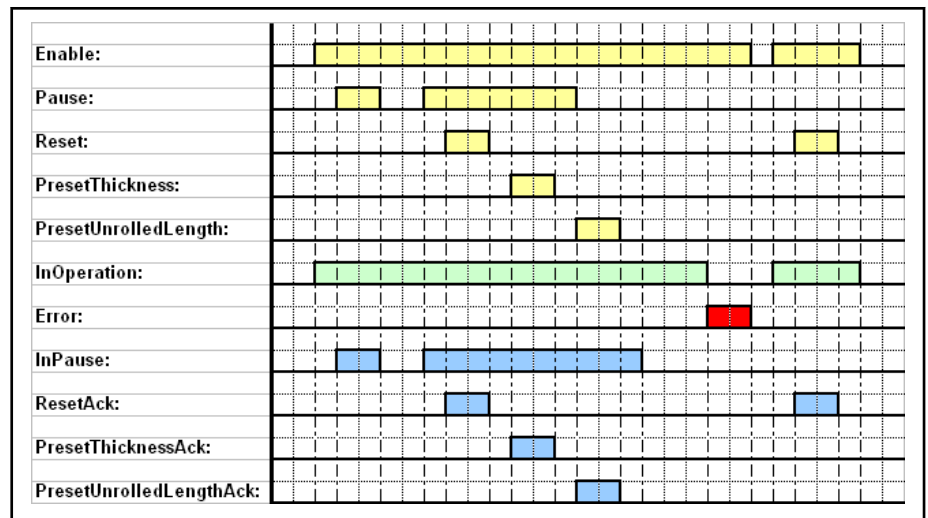


Fig. 8-40: Signal-time diagram for selected inputs and outputs of the MB_UnwindMaterialType02 function block

Functional description

Calculating the material variables

The remaining length of the "RestLength" web is calculated with the help of the current diameter. This means that variations in the current diameter also have to be observed while calculating 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 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 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.

Averaging the material thickness

The calculated material thickness can be averaged by setting the "AverageValue" input. The number of material thickness values used for averaging 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 a maximum of 30. If a higher value is entered, "AverageValue" is set to 30. However,

RMB_TechWinder.library/MX_TechWinder.lib

er, no error message 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 numbers of the function block. The error codes from the "F_RELATED" table (16#0170) are used.

ErrorID	Additional1	Additional2	Description
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. 8-72: Error codes of the MB_UnwindMaterialType02 function block

Required hardware

- IndraMotion MLC hardware CML65, CML45, CML25
- IndraDrive

Required firmware

- IndraMotion MLC firmware FWA-CMLXX*-MLC-13VRS
XX - Device variant, e.g. CML65
- IndraDrive drive firmware

Required software

- IndraWorks 13VRS for IndraMotion MLC or IndraMotion 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
IndraMotion MLC 13VRS	RMB_TechWinder.compiled-library

Tab. 8-73: Reference table of the MB_WindTaperProfileType01

Interface description

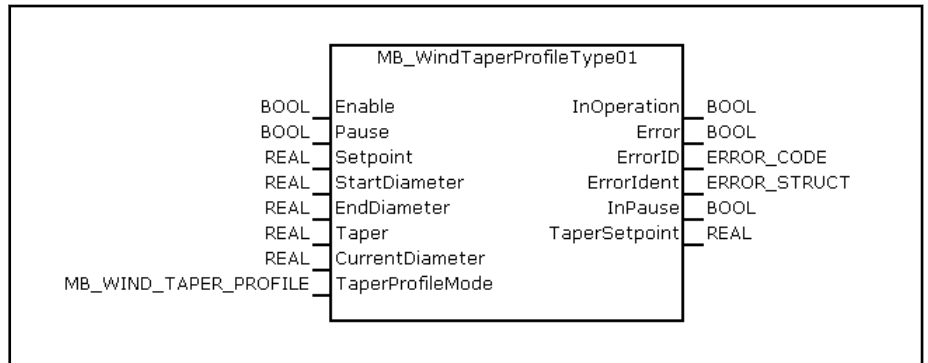


Fig. 8-41: MB_WindTaperProfileType01 function block

I/O type	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Processing of function block 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 winder (e.g. of the MB_WinderTensionCtrlType02 FB)
VAR_OUTPUT	TaperProfileMode	MB_WIND_TAPER_PROFILE	Selecting the winding taper profile
	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in the event of an error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Data type	Description
	InPause	BOOL	Winding taper pauses (tension command value is "frozen")
	TaperSetpoint	REAL	Command value considering the winding taper characteristic curve

Tab. 8-74: 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. 8-75: Min./max. and default values of the MB_WindTaperProfileType01 function block

MB_WIND_TAPER_PROFILE

Element	Value	Description
NO_PROFILE	0	No profile
PROFILE_LINEAR	1	Linear curve profile
PROFILE_HYPERBOLIC	2	Hyperbolic curve profile

Tab. 8-76: Elements of the MB_WIND_TAPER_PROFILE enumeration type

Signal-time diagram

Selected signals of the function block are displayed in the following signal-time diagram.

Signal context

Signal input	Signal output	Description
Enable	InOperation	Commissioning the function block
	Error	Ready for operation state has been interrupted
Pause	InPause	Stop internal controller process

Tab. 8-77:

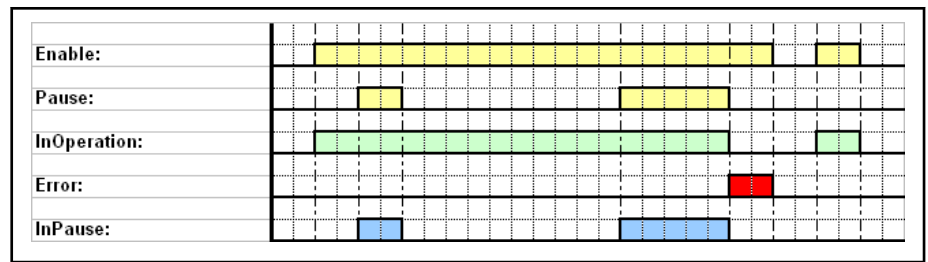


Fig. 8-42: Signal-time diagram for selected inputs and outputs of the MB_WindTaperProfileType01 function block

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. That means that the "SetpointOpMode" input of the winder function block MB_TensionCtrlType02 is interconnected with the "TaperSetpoint" output.

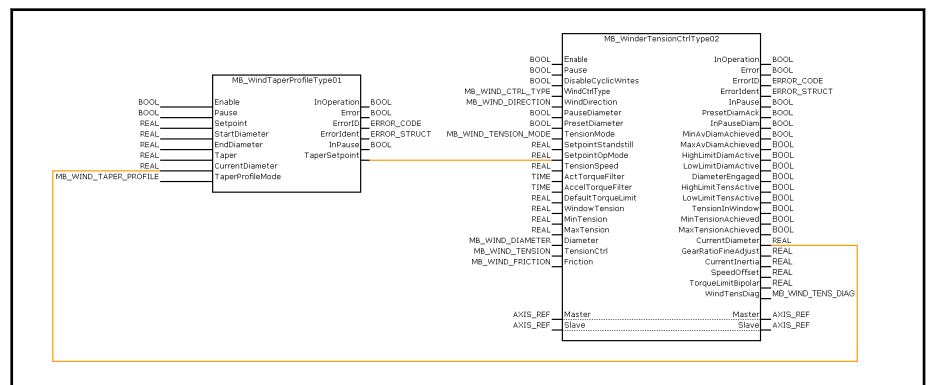


Fig. 8-43: MB_WindTaperProfileType01, interconnection

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:

RMB_TechWinder.library/MX_TechWinder.lib

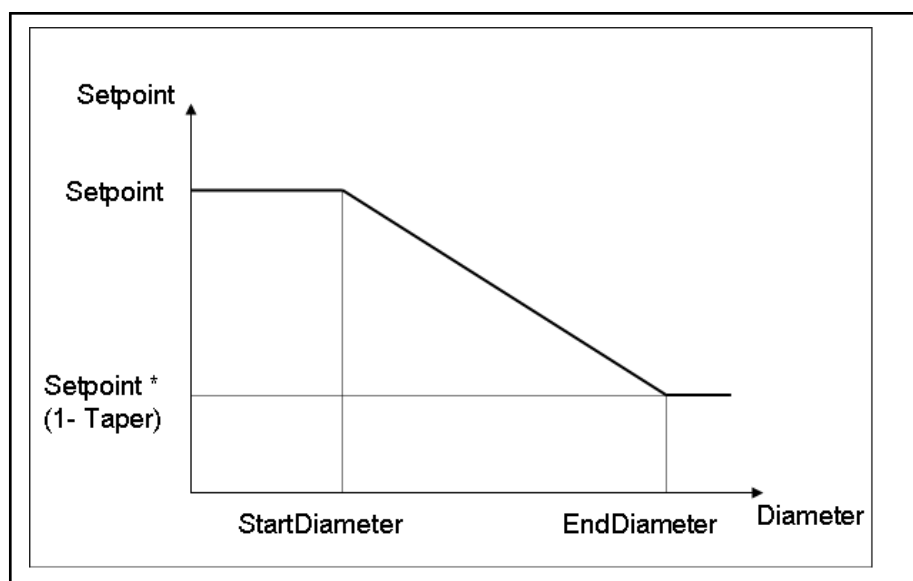


Fig. 8-44: MB_WindTaperProfileType01, linear winding taper characteristic curve
Hyperbolic winding taper characteristic curve:

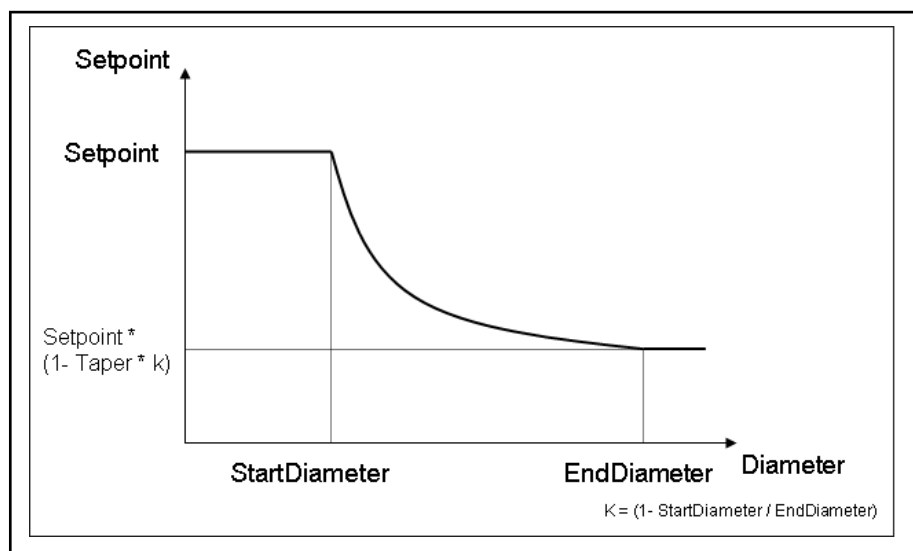


Fig. 8-45: MB_WindTaperProfileType01, hyperbolic winding taper characteristic curve

Error handling The following table lists and describes the error numbers of the function block. The error codes from the "F_RELATED" table (16#0170)" are used.

ErrorID	Additional1	Additional2	Description
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

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR	16#1151	16#0006	CurrentDiameter <= 0.0
INPUT_RANGE_ERROR	16#1151	16#0007	Incorrect TaperProfile mode

Tab. 8-78: Error codes of the MB_WindTaperProfileType01 function block

Required hardware	- IndraMotion MLC hardware CML65, CML45, CML25 - IndraDrive
Required firmware	- IndraMotion MLC firmware FWA-CMLXX*-MLC-13VRS XX - Device variant, e.g. CML65 - IndraDrive drive firmware
Required software	IndraWorks 13VRS for IndraMotion MLC
Special features for the IndraMotion MLC	No special parameterization required.
Special features for the IndraMotion MLD control	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
IndraMotion MLC 13VRS	RMB_TechWinder.compiled-library

Tab. 8-79: Reference table of the MB_WindSpeedControlAdaptionType01

Interface description

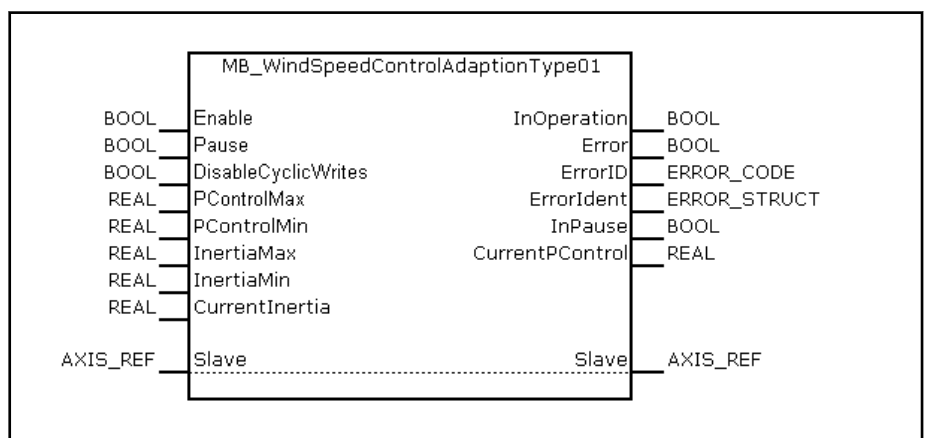


Fig. 8-46: MB_WindSpeedControlAdaptionType01 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	Slave	AXIS_REF	Axis structure of the winding axis
VAR_INPUT	Enable	BOOL	Processing of the function block 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)]

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Data type	Description
	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 is valid
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_ CODE	Diagnostic description in the event of an error
	ErrorIdent	ERROR_ STRUCT	Detailed diagnostics
	InPause	BOOL	Function block pauses (CurrentPControl is "frozen")
	CurrentPControl	REAL	Current Kp-value (speed controller) [Nm/(rad/s)]

Tab. 8-80: 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. 8-81: Min./max. and default values of the MB_WindSpeedControlAdaption-Type01 function blocks

Signal-time diagram Selected signals of the function block are displayed in the following signal-time diagram.

Signal context

Signal input	Signal output	Description
Enable	InOperation	Commissioning the function block
	Error	Ready for operation state has been interrupted
Pause	InPause	Stop internal controller process

Tab. 8-82:

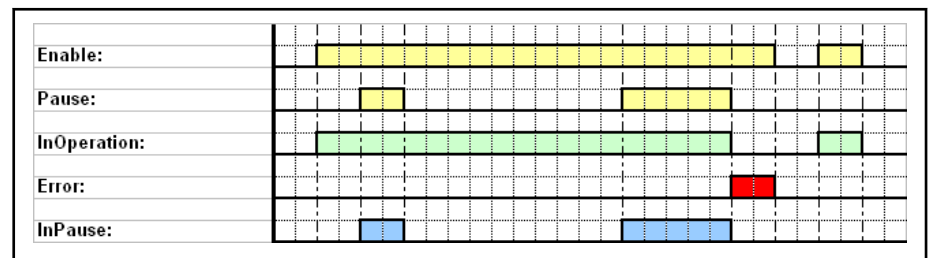


Fig. 8-47: Signal-time diagram for selected inputs and outputs of the MB_WindSpeedControlAdaptionType01 function block

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.

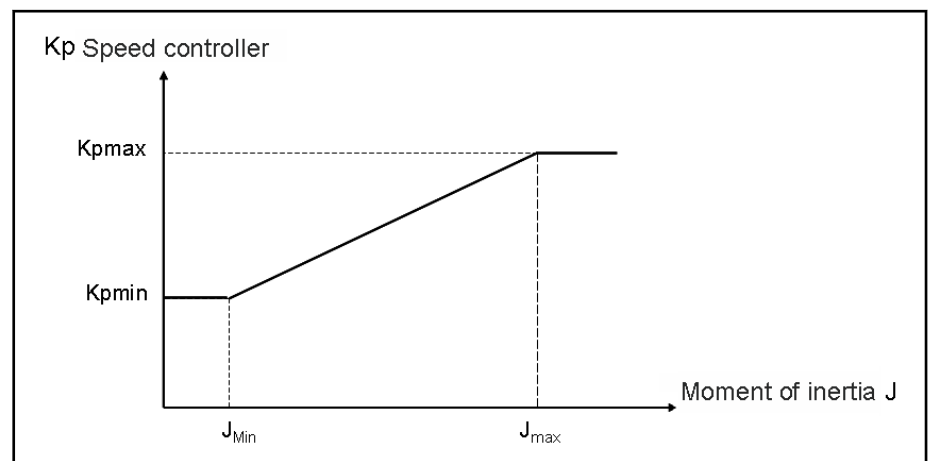


Fig. 8-48: MB_WindSpeedControlAdaptionType01

The current Kp-value is cyclically written to the drive parameter S-0-0100 "Velocity controller, 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 with 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 parameters.

RMB_TechWinder.library/MX_TechWinder.lib



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 controller, proportional gain" is configured cyclically, it has to be ensured that the respective cyclic channel is pre-initialized with a reasonable value after the control start. This applies until the function block was enabled.

The following table lists and describes the error numbers of the function block. The error codes from the "F_RELATED table (16#0170)" are used.

ErrorID	Additional1	Additional2	Description
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. 8-83: Error codes of the MB_WindSpeedControlAdaptionType01 function block

Required hardware	• IndraMotion MLC hardware CML65, CML45, CML25 • IndraDrive
Required firmware	• IndraMotion MLC firmware FWA-CMLXX*-MLC-13VRS XX - Device variant, e.g. CML65 • IndraDrive drive firmware
Required software	• IndraWorks 13VRS for IndraMotion MLC
Special features for the IndraMotion MLC	The Parameter S-0-0100 ("Velocity controller, proportional gain") has to be entered in the cyclic sercos channel of the corresponding drive "UserCmdDataX"). The real axes are parameterized in the parameterization mode in IndraWorks.



The initial values for parameters in the cyclic SERCOS channel are always 0. That means that when switching on control, the value for the parameter S-0-0100 "Velocity controller, 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 activating the control, e.g.

```
AxisData[drivename.AxisNo].dwUserCmdDataX_q.REAL_ := 10.0;
```

8.8.6 MB_TraverseControlType01

Brief description The block controls a translatory traverse axis that in combination with a winder (rotary axis) facilitates the laying of different material. The diameter of the

RMB_TechWinder.library/MX_TechWinder.lib

material is much smaller than the winder width. Consequently, several material windings can be laid out next to each other. It is differentiated between winding and unwinding.

During winding, the velocity of the traverse axis is coupled to the winder velocity and ensures a constant and closely spaced placing of the material.

During unwinding, several methods are available that can be selected depending on the given preconditions (e.g. available hardware/load cells).

Output variables of the function block are a target position and a positioning velocity for positioning block MC_MoveAbsolute or "Position" and "Velocity" of the operation mode "ModePosAbs" of the AxisInterface for a translatable axis.

Assignment: Target system/library

Target system	Library
MLC13VRS	RMB_TechWinder

Tab. 8-84: Reference table of the MB_TraverseControl function block

Interface description

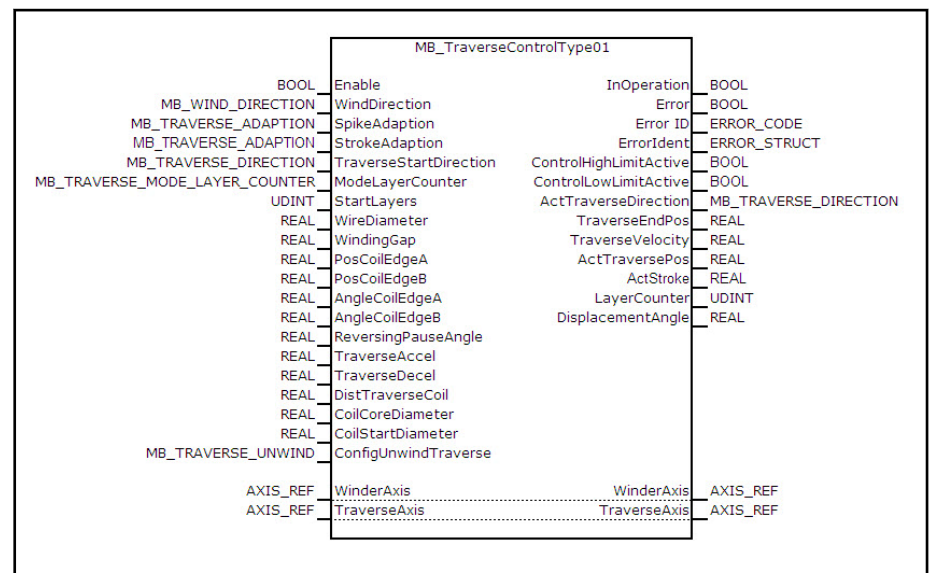


Fig. 8-49: MB_TraverseControlType01 function block

I/O type	Name	Data type	Description
VAR_IN_OUT	WinderAxis	AXIS_REF	Axis reference to the winder axis
	TraverseAxis	AXIS_REF	Axis reference to the traverse axis
VAR_INPUT	Enable	BOOL	Processing of function block enabled
	WindDirection	MB_WIND_DIRECTION	Differentiating between rewinding (REWIND) and unwinding (UNWIND).
	SpikeAdaption	MB_TRAVERSE_ADAPTION	Enabling and disabling of the velocity adaption to compensate for a missing path between an acceleration or a deceleration phase.

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Data type	Description
	StrokeAdaption	MB_TRAVERSE_ADAPTION	Enabling and disabling of the forward adaption to the edge of the coil. If enabled, the traverse axis traverses beyond the edge of the coil with the "ActStroke" value, to ensure that the wire or the material reaches the end of the coil. The traversing motion ends at the edge of the coil if the traverse axis is disabled.
	TraverseStart-Direction	MB_TRAVERSE_DIRECTION	Direction it is traversed to initially; i.e. in the direction of edge of coil A to edge of coil B (A_TO_B) or vice versa (B_TO_A).
	ModeLayer-Counter	MB_TRAVERSE_MODE_LAYER_COUNTER	<i>Select how the initial value has to be determined for the length counter in case of a positive edge at Enable:</i> - via "StartLayers", the starting diameter is calculated and the "CoilStartDiameter" input is not taken into consideration. - calculate via starting diameter ("CoilStartDiameter"), the input "StartLayers" is not taken into consideration.
	StartLayers	UDINT	Number of the already wound layer 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 individual windings (wires, etc.) of a layer. [% of WireDiameter]
	PosCoilEdgeA	REAL	Absolute position of edge of coil A. [mm]
	PosCoilEdgeB	REAL	Absolute position of edge of coil B. [mm]
	AngleCoilEdgeA	REAL	Angle of edge of coil against vertical. (0: Edge of coil straight, positive value: Edge of coil tilted outwards, i.e. coil becomes wider) [degree]
	AngleCoilEdgeB	REAL	Angle of edge of coil against vertical. (0: Edge of coil straight, positive value: Edge of coil tilted outwards, i.e. coil becomes wider) [degree]
	ReversingPauseAngle	REAL	Angle which the winder continues to rotate while the traverse axis remains at the edge of the coil. [Degree]
	TraverseAccel	REAL	Accelerates the traverse axis. Corresponds to the command value for MC_MoveAbsolute (or "ModePosAbs" operation mode Techinterface) of the traverse axis. [mm/s ²]
	TraverseDecel	REAL	Deceleration of traverse axis. Corresponds to the command value for MC_MoveAbsolute of the traverse axis. [mm/s ²]
	DistTraverseCoil	REAL	Distance from traverse axis and coil corresponds to the wire length that is positioned vertically in the air from the traverse axis to the coil. [mm]
	CoilCoreDiameter	REAL	Diameter of empty coil. [mm]

RMB_TechWinder.library/MX_TechWinder.lib

I/O type	Name	Data type	Description
	CoilStartDiameter	REAL	Diameter of coil at start (incl. wire). The diameter is only evaluated upon "CalculateStartLayers" at the "ModeLayerCounter" input. [mm]
	ConfigUnwindTraverse	MB_TRAVERSE_UNWIND	Configuration structure of the unwinding process
VAR_OUTPUT	InOperation	BOOL	Function block is operating
	Error	BOOL	Processing completed with error
	ErrorID	ERROR_CODE	Diagnostic description in the event of an error
	ErrorIdent	ERROR_STRUCT	Detailed diagnostics
	ControlHighLimitActive	BOOL	Output of PID controller has reached the upper limit during the unwinding process
	ControlLowLimitActive	BOOL	Output of PID controller has reached the lower limit during the unwinding process
	ActTraverseDirection	ENUM	Current traversing direction from the edge of coil A to the edge of coil B (A_TO_B) or vice versa (B_TO_A)
	TraverseEndPos	REAL	Target position (reversal point) of the traverse axis. Corresponds to the position command value of MC_MoveAbsolute of the traverse axis (including stroke and offset due to inclined coil edges). [mm]
	TraverseVelocity	REAL	Command velocity of the traverse axis. Corresponds to the velocity command value of MC_MoveAbsolute of the traverse axis. [mm/min]
	ActTraversePos	REAL	Current traverse axis position. [mm]
	ActStroke	REAL	Current value for the forward stroke (distance that is traveled beyond the edge of the coil). [mm]
	LayerCounter	UDINT	Number of layers on the coil. [-]
	DisplacementAngle	REAL	Angle displacement at the end of coil to the starting angle after a displaced position. [Degree]

Tab. 8-85: MB_TraverseControlType01 I/O interface

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"

RMB_TechWinder.library/MX_TechWinder.lib

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. 8-86: Minimum/maximum values and default values of the MB_TraverseControlType01 inputs

MB_TRAVERSE_ADAPTION

An adaptive variable or mode can be enabled or disabled using this enumeration (e.g. forward stroke adaption "StrokeAdaption", velocity adaption "SpokeAdaption")

Element	Value	Description
ADAPTION_OFF	0	Adaption is disabled
ADAPTION_ON	1	Adaption is enabled

Tab. 8-87: Elements of the enumeration type MB_TRAVERSE_ADAPTION

MB_TRAVERSE_DIRECTION

The traversing direction of the traverse axis is specified by this enumeration. It is possible to traverse from the edge of the coil A (PosCoilEdgeA) in the direction of the edge of the coil B (PosCoilEdgeB) and vice versa:

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. 8-88: Elements of the enumeration type MB_TRAVERSE_DIRECTION

MB_TRAVERSE_MODE_LAYER_COUNTER

Specifies how the initial value has to be determined for the length counter in case of a positive edge at Enable:

Element	Value	Description
INPUT_START_LAYERS	0	Read in initial value from the input for the layer counter.
CALCULATE_START_LAYERS	1	Calculate the initial value for the layer counter from the coil diameter.

Tab. 8-89: Elements of the enumeration type MB_TRAVERSE_MODE_LAYER_COUNTER

MB_TRAVERSE_UNWIND

By means of this structure, the required settings can be selected for the unwinding process at the traverse axis. The unwinding mode can be selected by using the "Mode" element. The "WireFeedDirection" input describes the direction from which the wire or the material wraps around the traverse axis. Using the "Control" structure, a PID controller is configured that controls the traversing velocity (depending on the unwinding mode). The "TraverseVel-MiddleFixed" input specifies the velocity at which the traverse axis travels to the center position in mode "MIDDLE_FIXED". The values of the load cells are read in using the "ActWireTension" and "ActWireTensionTraverse" inputs (depending on the unwinding mode). The values of the load cells can be filtered with a PT1 element, whose time constants are specified using the "FilterWireTensionTraverse" and "FilterWireTension" inputs.

RMB_TechWinder.library/MX_TechWinder.lib

Name	Type	Min. value	Max. value	Default value	Effective	Description
Mode	MB_TRAVERSE_UNWIND_MODE			MIDDLE_FIXED	Rising edge at "Enable"	Selecting the different unwinding modes.
WireFeed-Direction	MB_TRAVERSE_DIRECTION			A_TO_B	Continuous	Direction from which the wire or the material is routed around the traverse axis.
Control	MB_TRAVERSE_UNWIND_CONTROL				Continuous	Configuring the controller for the velocity control during unwinding
VelMiddle-Fixed	REAL	0.0	n.def.	50	Continuous	Velocity at which the traverse axis travels to the center position in the "MIDDLE_FIXED" mode. [mm/min]
ActWireTensionTraverse	REAL	0.0	n.def.	0.0	Continuous	Current measured value of the wire or material tension at the traverse axis, measured at a load cell in the traversing unit. [N]
ActWireTension	REAL	0.0	n.def.	0.0	Continuous	Current measured value of the wire or material tension at the "System", measured at a load cell in the traversing unit. [N]
FilterWireTensionTraverse	TIME	0 s	n.def.	0 s	Continuous	Time constant of the PT1 element to filter "ActWireTensionTraverse"
FilterWireTension	TIME	0 s	n.def.	0 s	Continuous	Time constant of the PT1 element to filter "ActWireTension"

Tab. 8-90: Min./max. and default values of the MB_TRAVERSE_UNWIND structure

MB_TRAVERSE_UNWIND_MODE

The different unwinding modes can be selected:

Element	Value	Description
MIDDLE_FIXED	0	Traverse axis is positioned in the center of the coil
WINDING_STEP	1	Traverse axis travels by the wire width or the material width (plus gap)
WIRE_TENSION_TRIM	2	Automatic adjustment of the traversing velocity depending on the force difference in the traverse axis and outside of the traverse axis. End position is the coil edge.
WINDING_STEP_TENSION_TRIM	3	Combination of WINDING_STEP and WIRE_TENSION_TRIM, i.e. traversed by the wire width or the material width (plus gap) and additive correction by means of the force difference

Tab. 8-91: Elements of the enumeration type MB_TRAVERSE_UNWIND_MODE

MB_TRAVERSE_UNWIND_CONTROL

Using this structure, the PID controller inputs are connected for the unwinding process. The controller output additively influences the velocity of the traverse 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
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. 8-92: Min./max. and default values of the MB_TRAVERSE_UNWIND_CONTROL structure

Signal-time diagram Signal-time diagram

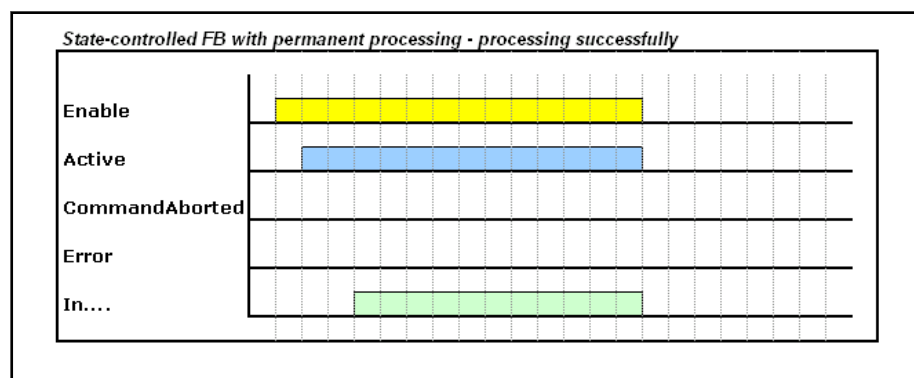


Fig. 8-50: Signal time diagram for MB_TraverseControlType01

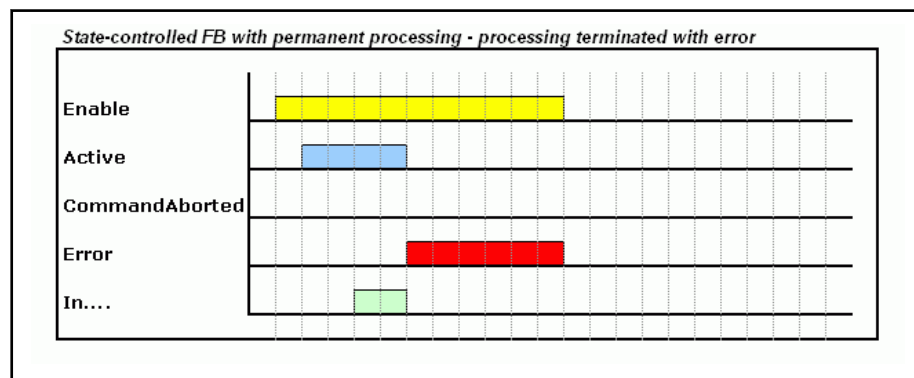


Fig. 8-51: Signal time diagram for MB_TraverseControlType01 in case an error occurred

Orientation guide:

["Functional description" on page 582](#)

["Winding functionality" on page 583](#)

["Unwinding functionality" on page 585](#)

RMB_TechWinder.library/MX_TechWinder.lib

["Changing between winding and unwinding:" on page 591](#)["Basic use cases" on page 591](#)

Functional description The MB_TraverseControl function block facilitates winding and unwinding of wires, threads/filaments etc. on coils together with the winder function block. A translatory traverse axis is always used.

Output variables of the function block are a target position and a positioning velocity for positioning block MC_MoveAbsolute or "Position" and "Velocity" of the operation mode "ModePosAbs" of the AxisInterface for a translatory axis.

Commanding a traverse axis

The correct operation mode of the drive and the drive enable of the drive have to be set for the traverse axis.

There are two ways:

- Set the "ModePosAbs" operation mode using the AxisInterface (recommended)
- Use the PLCopen "MC_Power", "MC_MoveAbsolute", "MC_Stop" function blocks to command the axis

Furthermore, the corresponding winder axis has to be commanded.

The following starting/stopping sequence is recommended for the traverse and winder axis:

Starting sequence:

1. Starting the winder axis operation mode
2. Starting the traverse axis operation mode
3. Positioning the traverse axis, if required
4. Starting the traverse axis function block
5. Starting the winder function block
6. Starting the operation mode master axis

Stopping sequence:

1. Stopping the master axis operation mode
2. Stopping the traverse axis function block
3. Stopping the traverse axis operation mode
4. Stopping the winder function block
5. Stopping the winder operation mode

Connecting to the commanding function block MC_MoveAbsolute or AxisInterface

The values for acceleration and deceleration of the commanding function block or the AxisInterface operation mode also have to be connected to the inputs of the traverse axis function block "TraverseAccel" and "TraverseDecel".

The outputs of the traverse function block "TraverseEndpos" and "TraverseVelocity" have to be connected to the inputs of the commanding function block or the AxisInterface "Position" and "Velocity".

Differences: Winding/Unwinding

It is differentiated for the traverse axis between winding and unwinding. The traverse axis is translatory scaled.

- During winding, the velocity of the traverse axis is coupled to the winder velocity (a winder axis) and ensures a constant and closely spaced placing of the material. It is alternated from the edge of coil A to edge of coil B.
- During unwinding, several methods are available that can be selected depending on the given preconditions (e.g. available hardware/load cells).

The difference between winding and unwinding is established using the "Wind-Direction" input.

Winding functionality

During the winding process, the traverse axis alternately travels from edge of coil A to edge of coil B. Upon each turn, it is automatically traversed by material strength and gap.

Velocity ("TraverseVelocity"):

The traverse axis velocity is coupled to the winder speed. The traverse axis traverses by one winder step per winder rotation. The winder step results from material strength and gap between the winding turns.

Starting direction ("TraverseStartDirection"):

Upon the start of the winding process, it is required that the user specifies the traversing direction. I.e., the user has to enter the direction (A_TO_B, from edge of coil A to edge of coil B) in "TraverseStartDirection".

The current direction (e.g. A_TO_B) and position is output.

Coil edges:

The coil edges have to be specified as absolute positions. "PosCoilEdgeA" has to have a smaller value than "PosCoilEdgeB". The end points at the coil form have to be specified for coils with inclined edges.

If the coil edges are not straight but inclined by an angle against the vertical, the angle has to be specified in "AngleCoilEdgeA" or "AngleCoilEdgeB". A positive angle corresponds to a broadening coil. The edge position and thus the target position of the traverse axis (corresponds to the target position of the corresponding commanding function block "MC_MoveAbsolute" or the "ModePosAbs" operation mode of the TechInterface) is automatically adjusted, depending on the winder position.

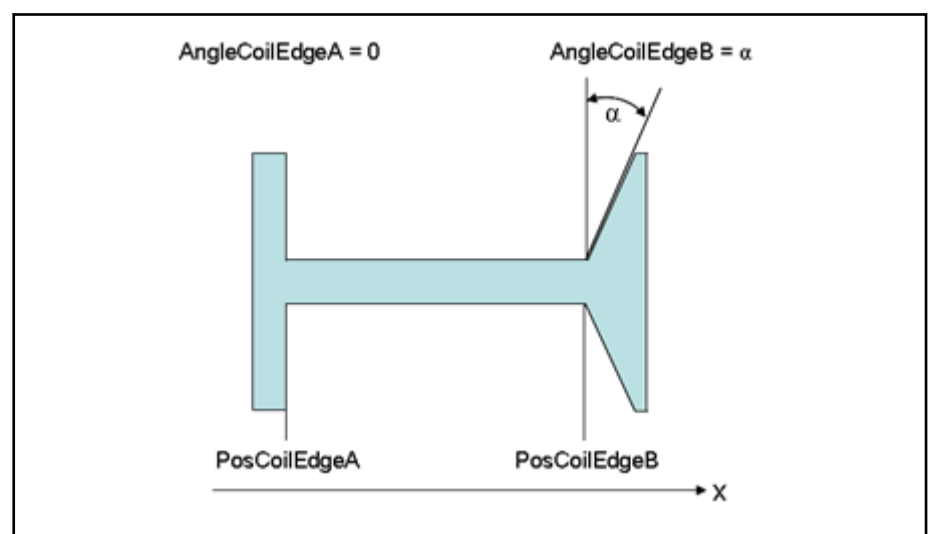


Fig. 8-52:

Forward stroke ("StrokeAdaption"):

RMB_TechWinder.library/MX_TechWinder.lib

Due to the distance between traverse axis and coil, the traverse axis has to lead the position at which the wire is wound on the coil. The larger the winding step (material width plus gap) and the distance of traverse axis to coil, the larger the forward stroke. Moreover, the current (variable) coil diameter is reduced.

To ensure that the wire reaches the end of the coil, it might be necessary to travel beyond the edge of the coil. Use "StrokeAdaption" to start or stop traveling beyond the edge of the coil.

If "StrokeAdaption" is active, the traverse axis traverses beyond the actual edge of coil by the forward stroke ("ActStroke"). I.e., the end position is calculated from the edge of coil A – forward stroke or edge of coil B + forward stroke. Please note that the wire can sporadically rub at the upper edge in case of straight coil edges.

If "StrokeAdaption" is not active, the traverse axis reverses at the edge of the coil.

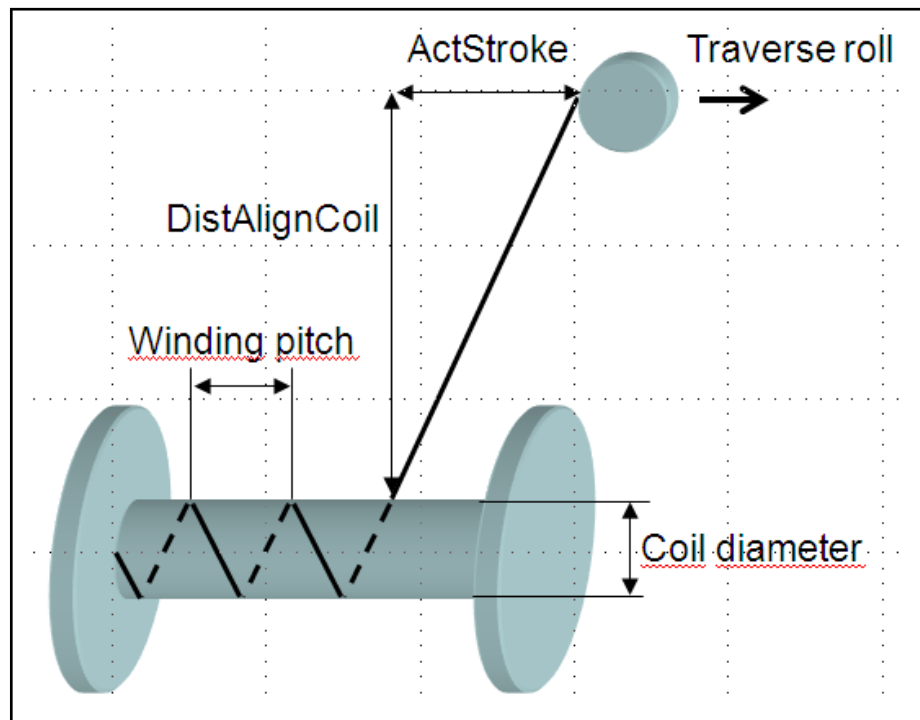


Fig. 8-53:

Layer counter ("LayerCounter"):

The number of wound layers is counted and output.

If an already filled coil should still be wound, the initial value has to be determined for the layer counter. There are two options how to select the mode layer counter ("ModeLayerCounter").

First option: The user specifies the number of layers ("StartLayers"). The "CoilStartDiameter" input is not evaluated but calculated from the number of layers.

Second option: The number of layers is calculated from the initial coil diameter ("CoilStartDiameter"). The "StartLayers" input is not evaluated.

Pause angle ("ReversingPauseAngle"):

If the traverse axis reaches its target position (edge of coil), it pauses until the winder has rotated around the pause angle. Pausing facilitates changing to the next (higher) level.

Velocity adaption ("SpikeAdaption"):

During accelerating and decelerating at the reverse point (edge of coil), i.e. at the start and end of each layer (reversing), the desired traversing distance (winder step) per winder rotation cannot be reached completely. If the deviation is too large, velocity peaks can compensate for the deviation that can be activated using "SpikeAdaption". If this function is enabled, the velocity of the traverse axis is increased during the first and the last winder rotation, to compensate for the loss. This can be required in case of thick wires that have to wound on coils with a comparatively small diameter.

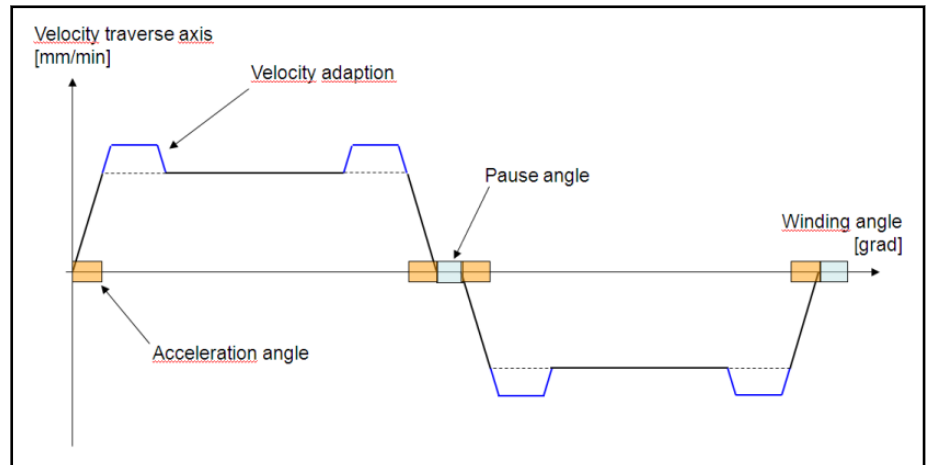


Fig. 8-54: Velocity curve with SpikeAdaption

Displacement angle ("DisplacementAngle"):

The wires of a layer are positioned in the grooves of the lower layer between two wires over the largest part of the coil diameter. However, wire crossings occur in each layer in which the winder height increases. To avoid an imbalance of the coil, the wire crossings should be distributed evenly over the coil circumference. The displacement angle is specified for checking purposes. The angle specifies the angle difference (rotation angle of winder axis) before the start of a new layer of one coil end to the start of the next layer of the other coil end. The angle should thus not be a divider of 180° . The displacement angle can be influenced with the gap between the turns and the pause angle.

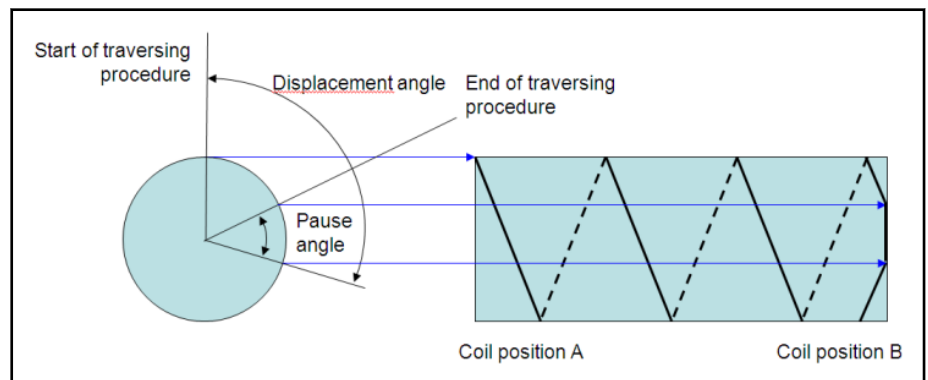


Fig. 8-55: Displacement angle

Unwinding functionality

Four different methods are available for unwinding. They can be selected using the "ConfigUnwindTraverse.Mode" input.

If the routing of the wire on the coil is unknown, jumps or changes of direction can be contained. The methods "WIRE_TENSION_TRIM", "WIRE_TEN-

RMB_TechWinder.library/MX_TechWinder.lib

SION_TRIM_DELTA" or "WINDING_STEP_TENSION_TRIM" are recommended as the velocity control compensates for such effects.

1. "MIDDLE_FIXED" traverse axis is always fixed in the center position
2. "WINDING_STEP" traverse axis is always traversed by the material strength (as for winding)
3. "WIRE_TENSION_TRIM" traverse axis is scaled and traversed using a PID controller (two load cells are required), depending on the difference between the actual traverse axis tension value and the actual tension value
4. "WINDING_STEP_TENSION_TRIM" Combination of method 2 and 3, traversing by material strength and tension difference
(Two load cells are required)

If no measuring equipment is available, the methods "MIDDLE_FIXED" and "WINDING_STEP" can be used. "MIDDLE_FIXED" as most simple option, "WINDING_STEP" if the coil to be unwound has been wound uniformly.

If the traverse axis position should be adjusted automatically and a uniformly wound coil cannot be assumed, the methods "WIRE_TENSION_TRIM" or "WINDING_STEP_TENSION_TRIM" have to be used. The tension in the traverse axis area is compared to the web tension outside the traverse axis. This web tension difference is scaled via an adjustable PID controller and is set to the initial velocity of the traverse axis "TraverseVelocity" using **value** and **sign**.

Calibrating the load cells:

The load cells have to be calibrated to show an identical value in case of a vertical wire layout from the traverse axis to the coil. The wire must be routed around the coil during operation, also if the load cell is outside the traverse axis.

Force measurement

The wire or the material is routed via a coil in which the tension is measured. The measured force also depends on to what extend the wire or material is wrapped around the coil, i.e., how large the angle of wrap is.

The values of the forces F_D and F_{D1} or F_{D2} are identical and constitute the tension of the wire.

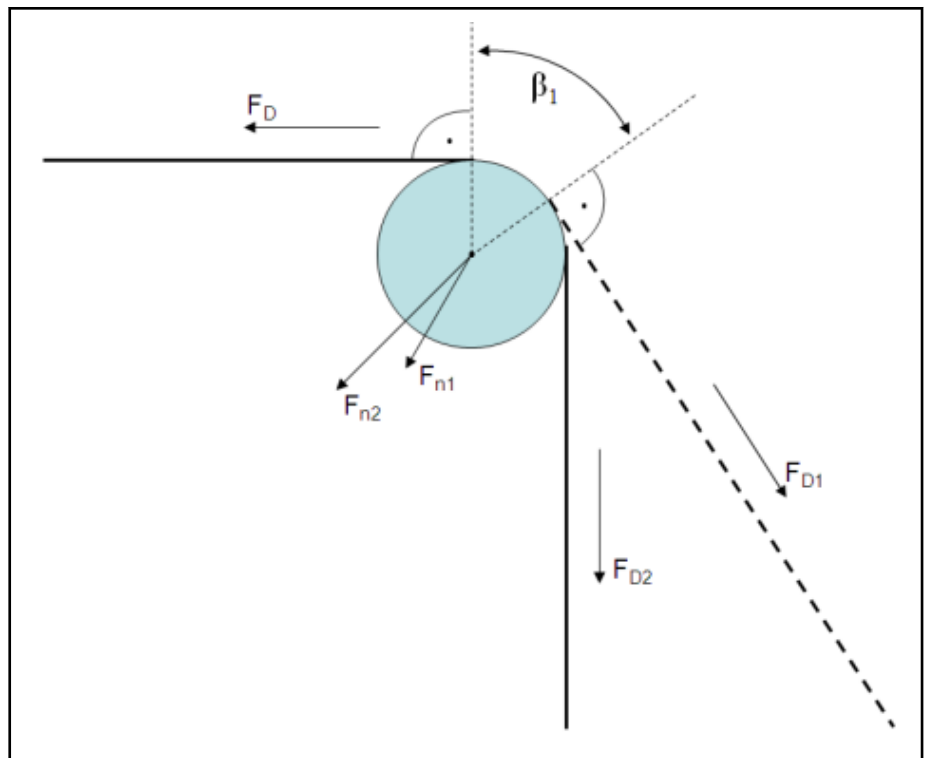


Fig. 8-56: Forces and angles of wrap

The force measured by the load cell F_{n1} (F_{n2} for β_2) depends on the angle of wrap β_1 . The angle of wrap β identifies the coil area in which the wire or the material has contact with the coil. In other words, the angle that both verticals form from the wire or the material through the center of the coil.

The following applies for the measured actual tension:

$$F_n = 2 * \sin(\beta/2) * F_{\text{max}}$$

Fig. 8-57: Interconnection between actual tension and angle of wrap

Direction of wire ("ConfigUnwindTraverse.WireFeedDirection"):

To ensure correct scaling of the velocity, the input "ConfigUnwindTraverse.WireFeedDirection" has to be specified. This input specifies from which direction the wire is routed around the traverse axis. If the, e.g., the wire is routed from the edge of coil A to the traverse axis and subsequently down to the coil, enter "A_TO_B".

RMB_TechWinder.library/MX_TechWinder.lib

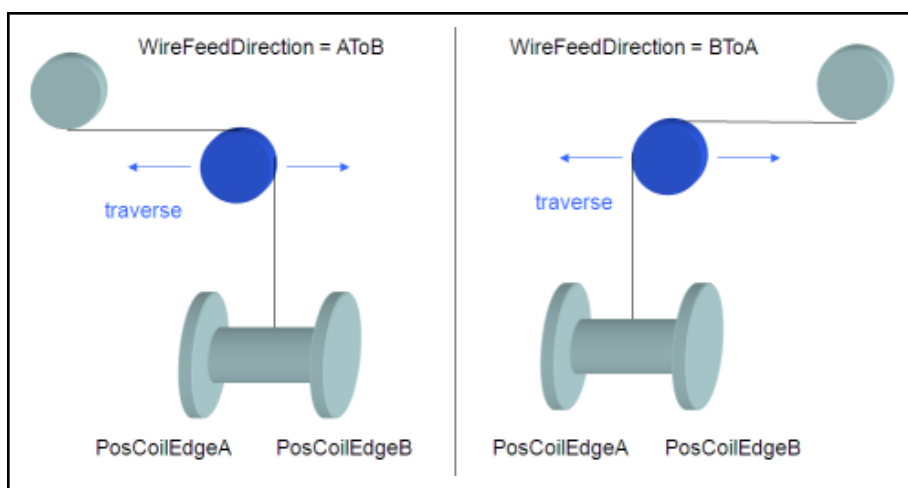


Fig. 8-58: WireFeedDirection

Layer counter ("LayerCounter"):

The layer counter is continued but the end of the layer cannot be determined exactly. Instead, the theoretically possible turns of a layer are calculated and the winder rotations are counted. If the calculated turns have been reached, a layer is assumed as completed and the layer counter is decremented.

The initial value for the layer counter is determined. There are two options how to select the mode layer counter ("ModeLayerCounter").

First option: The user specifies the number of layers ("StartLayers"). The "CoilStartDiameter" input is not evaluated but calculated from the number of layers.

Second option: The number of layers is calculated from the initial coil diameter ("CoilStartDiameter"). The "StartLayers" input is not evaluated.

Mode "MIDDLE_FIXED":

In case of this unwinding method, the traverse axis is positioned in the center of the coil. The position is automatically calculated from the coil edge positions and are traversed with the velocity specified with ""ConfigUnwindTraverse.Traverse VelMiddleFixed"". If the center position has been reached, the traverse axis stops.

Application:

Using this mode is only recommended in case of a relatively large distance between traverse axis and coil, relatively narrow coil and robust wires, as the wire comes into contact with the lower layer due to its inclined position at the coil ends and oscillates.

Direct switching to winding mode is not possible in this mode as the traverse axis position gets lost.

Mode "WINDING_STEP":

In case of this winding method, the winder travels with all its properties, just as in winding mode. That means, the traverse axis is moved by material strength and gap upon each winding.

Application:

This method is only recommended if the wire has been wound previously. That means, no jumps or changes of direction are expected or it can be assumed that the coil to be unwound has been wound uniformly.

This method can be implemented with the least effort.

Mode "WIRE_TENSION_TRIM":

In the mode "WIRE_TENSION_TRIM", the velocity of the traverse axis is controlled to ensure a vertical wire layout to the traverse axis of the coil.

Force measurement:

Two forces are required ("ConfigUnwindTraverse.ActWireTension" and "ConfigUnwindTraverse.ActWireTensionTraverse"). They are filtered via a PT1 element whose filter time constants can be set via "ConfigUnwindTraverse.FilterWireTension" or "ConfigUnwindTraverse.FiliterWireTensionTraverse" (e.g. 20ms).

Velocity Control:

Apply the forces to the controller. The force outside the traverse axis is the command value, the force inside the traverse axis is the actual value. As both forces are identical in case of a vertical traverse axis position with respect to the coil, (cf. "Calibrating the load cells"), the velocity of the traverse axis is controlled to ensure a vertical traverse axis position.

The controller output constitutes the command velocity of the traverse axis in both cases (this corresponds to the command velocity of the corresponding commanding function block "ModePosAbs" of the TechInterface).

The control parameters can be selected via "ConfigUnwindTraverse.Control". The P-gain amplifies the control difference. It has to be selected to ensure a quick reaction of the traverse axis to jumps in the wire position. The I-gain ensure a complete smoothing of the control difference. The D-gain reacts to changes in the lead of the control difference.

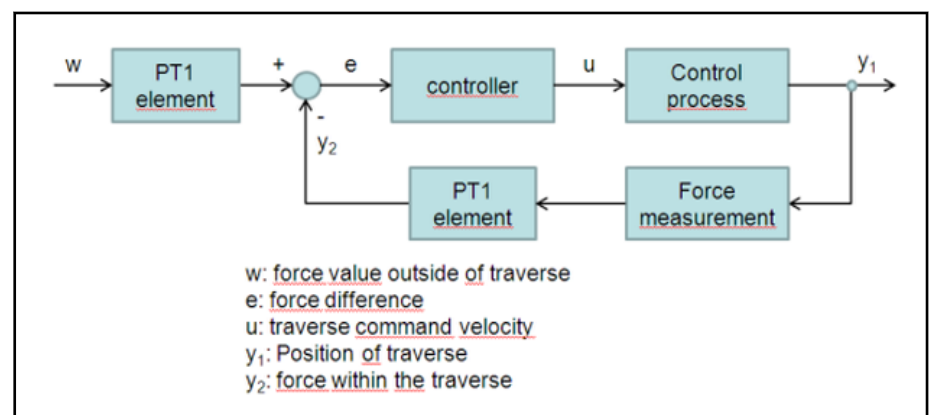


Fig. 8-59: Velocity control with force difference

Only one P-gain can be used in the first step.

The P-gain should be set to a high value to prevent oscillation of the traverse axis at standstill. As the control difference is low in comparison, a large value (e.g. 1000) has to be selected to facilitate a fast traverse axis reaction.

$$(K_p = 0.5 \cdot k_{p,\text{critical}})$$

Target position ("TraverseEndPos"):

The target position of the traverse axis (corresponds to the target position of the corresponding function block MC_MoveAbsolute or the operation mode "ModePosAbs" of the TechInterfaces) is the respective coil edge (does not necessarily have to be reached). Depending on the calculated command velocity, the direction and thus the target position (edge of could) is changed. I.e. in case of a sign reversal, the "TraverseEndPos" is changed.

Mode "WINDING_STEP_TENSION_TRIM":

This winding mode is a combination of "WINDING_STEP" and "WIRE_TENSION_TRIM". The traverse axis is traveled by one winding step per winder

RMB_TechWinder.library/MX_TechWinder.lib

rotation. A velocity control is superimposed that is used to adjust irregularities in the specified winding.

Velocity Control:

The "basic velocity" of the traverse axis is calculated from the winder speed and the winding step (material width plus gap). No velocity compensation by velocity peaks is provided.

A velocity control is superimposed as is the case in the setting "WireTension-Trim". The control parameters can be selected via "ConfigUnwindTraverse.Control". A P-gain value has to be selected to prevent that the superimposed velocity does not overwrite the "basic velocity" in case of minor deviations from the vertical position but has a decisive effect in case of larger deviations. Furthermore, the control parameters especially the P-gain value have to be adjusted to the winder velocity during runtime by the user. As the "basic velocity" of the traverse axis increases upon an increasing winder speed and thus, the ratio of "basic velocity" and superimposed controlled velocity changes.

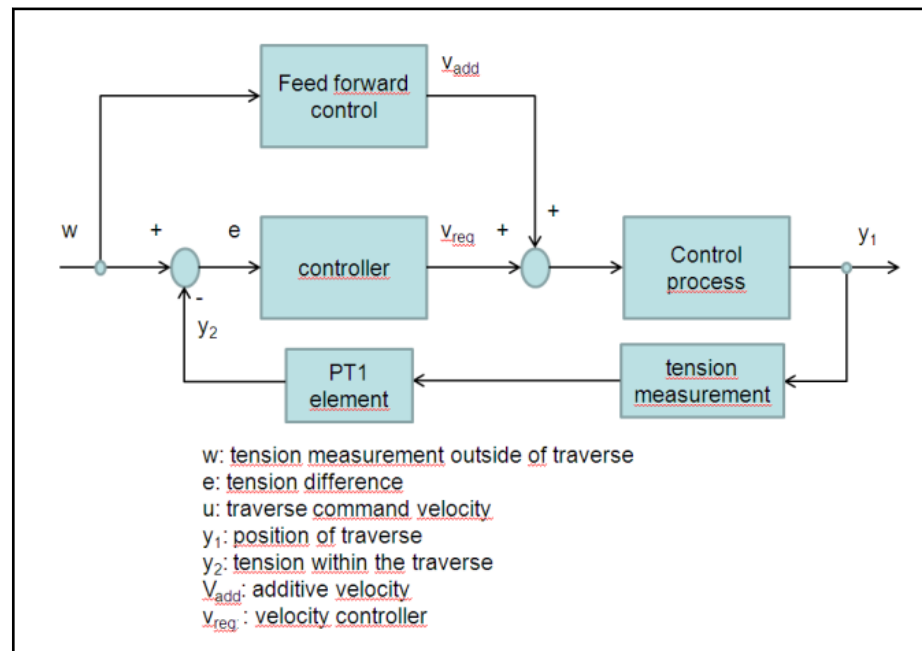


Fig. 8-60: Velocity control with force difference and feed forward

The winding method "WindingStepTensionTrim" is a combination of a control and a superimposed traverse axis velocity control. The determined velocities are added and the sum is reported at the traverse axis function block output. The controller settings have to be set to avoid overwriting of the feed forward but to still react in case of larger deviations. A P-controller can be used in the first step. Possible method to determine the P-gain:

The material velocity " v_{Mat} " is specified. The traverse axis velocity " $v_{traverse}$ " can be calculated from the material velocity in case of a uniform installation of the material using the following formulas:

$$n_{Wind} = \frac{v_{Mat}}{d_{coil} * \pi} \quad \text{and} \quad v_{traverse} = WS * v_{Wind}$$

A traverse axis velocity of " $v_{traverse}$ " = 31 mm/min is calculated for a material velocity " v_{Mat} " = 1 m/s, a coil diameter " d_{coil} " = $80 * 10^{-3}$ m and a winding step (material width + gap = ws = 0.13 mm).

Using the formula "[Functional description](#)" on page 582, a force difference depending on the angle of wrap. For example, if an angle of wrap is changed by 5 degrees (against 90°), a force and thus a control deviation of 0.6 (force at 90° minus force at 85°) results from this modification.

If the maximum angle of deflection has to be equal to the five degrees, the proportional controller gain has to set to a value to facilitate a traverse axis velocity equal to zero. The controlled variable has to be equated with the velocity determined by feed forward. Result: The traversing velocity v_{trav} has to be divided by the control deviation which results in a P-gain of approximately 51.7.

However, this calculation depends on the variable winder speed. Selecting the five degrees as limit value for the changed angle of wrap depends on the application and serves only as an example in this chapter.

Due to the feed forward, the velocity curve is smoother than for "WireTensionTrim". However, the reaction to jumps etc. in the wire position is slower due to the considerably lower P-value and the resulting lower traversing velocity.

Target position ("TraverseEndPos"):

The target position of the traverse axis (corresponds to the target position of the corresponding function block MC_MoveAbsolute or the operation mode "ModePosAbs" of the TechInterfaces) is the respective coil edge (does not necessarily have to be reached).

The direction and thus the target position of the traverse axis changed if either the edge of coil is reached and the pause angle has been waited for or if the superimposed velocity exceeds the "basic velocity" and the sum is negative.

Application:

This method is recommended in particular for unwinding of coils that have a uniformly distributed wire/material due to the implicit feed forward. Compared to the controlled unwinding with winder step, the settings used for winding (e.g. gap between windings) does not have to be known when using this method. Furthermore, the user can react promptly to irregularities in the placing of the material.

Changing between winding and unwinding:

It is possible to change between different modes during the operation of the function block.

When changing from winding to unwinding mode or vice versa, the direction of winding is reversed (e.g. from "A_TO_B" to "B_TO_A").

It is not recommended to change from the unwinding mode "MiddleFixed" to the winding mode, as no starting position is specified for the traverse axis. Disable the function block, position the traverse axis and subsequently enable the function block again.

Basic use cases

It is differentiated between two use cases:

- In the first use case, the winding material (e.g. wire) is wound or unwound from a coil. The coil is completely emptied or filled and subsequently, a new coil is installed. In this case, the traverse axis and the winder function block are stop to change the coil and are subsequently reinitialized and started. (Use case: Wire-wound coil manufacturing)
- In the second use case, the coil is not emptied or filled completely but a part is unwound or wound immediately (or vice versa). In this case, it is

RMB_TechWinder.library/MX_TechWinder.lib

directly changed from winding to unwinding mode (or vice versa) without stopping the function block.

(Application example for the second use case: Ingot saw)

Error handling The following table lists and describes the error codes of the function block. The error codes from the "F_RELATED table (16#0170)" are used. Additionally, the error codes of the function blocks [chapter 8.2.2 "MB_DancerControl-Type03" on page 475](#), [chapter 8.3.2 "MB_DiameterCalculatorType03" on page 489](#), [chapter 8.3.2 "MB_DiameterCalculatorType03" on page 489](#), IL_PIDType01 and IL_PT1Type01 can be output.

ErrorID	Additional1	Additional2	Description
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	SpikeAdapton 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
INPUT_RANGE_ERROR	16#1111	16#0013	ConfigUnwindTraverse.TraverseVelMiddleFixed < 0.0
INPUT_RANGE_ERROR	16#1111	16#0014	StrokeAdapton 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. 8-93: Error codes function block MB_TraverseControlType01

- Required hardware**
- IndraMotion MLC hardware CML65, CML45, CML25
 - IndraDrive

- Required firmware**
- IndraMotion MLC firmware FWA-CMLXX*-MLC-13VRS
XX - Device variant, e.g. CML65
 - IndraDrive drive firmware
- Required software**
- IndraWorks 13VRS for IndraMotion MLC
- Special features for the IndraMotion MLC**
- Select translatory scaling for the traverse axis.

8.9 Parameter overview

This section lists the parameters used in the documentation.

There are axis parameters (A-parameters) for the IndraMotion MLC control, product-specific parameters (P-parameters) and default parameters (S-parameters) for the respective drive.

For the IndraMotion MLC control, there are only product-specific parameters (P-parameters) and default parameters (S-parameters).

Parameters	Designation
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
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-0-2798	Polarity of the master axis position

Tab. 8-94: IndraMotion MLC axis parameters (A-parameters)

Parameters	Designation
S-0-0040	Actual velocity value
S-0-0043	Velocity polarity parameter
S-0-0055	Position polarities
S-0-0080	Torque/force command value
S-0-0084	Actual torque/force value
S-0-0085	Torque/force polarity parameter
S-0-0091	Bipolar velocity limit value
S-0-0092	Torque/force limit value, bipolar
S-0-0100	Velocity controller, proportional gain

RMB_TechWinder.library/MX_TechWinder.lib

Parameters	Designation
S-0-0121	Load gear input revolutions
S-0-0122	Load gear output revolutions

Tab. 8-95: Default parameters (S-parameters)

Parameters	Designation
P-0-0108	Master drive polarity
P-0-0665	Axis controller configuration
P-0-0690	Additive velocity command value
P-0-0694	Gear ratio fine adjustment, process controller

Tab. 8-96: Product-specific parameters (P-parameters)

Parameters	Designation
P-0-1623	CCD: Configuration master command values
P-0-1624	CCD: Configuration actual slave values
P-0-1625	CCD: Configuration list slave command values
P-0-1626	CCD: Configuration list actual slave values
P-0-1730	CCD: MDT real-time container 1, master
P-0-1731-39	CCD: MDT real-time container 1, slave n (n = 1-9)
P-0-1740	CCD: MDT real-time container 2, master
P-0-1741-49	CCD: MDT real-time container 2, slave n (n = 1-9)
P-0-1750	CCD: MDT real-time container 3, master
P-0-1751-59	CCD: MDT real-time container 3, slave n (n = 1-9)
P-0-1760	CCD: MDT real-time container 4, master
P-0-1761-69	CCD: MDT real-time container 4, slave n (n = 1-9)
P-0-1770	CCD: AT real-time container 1, master
P-0-1771-79	CCD: AT real-time container 1, slave n (n = 1-9)
P-0-1780	CCD: AT real-time container 2, master
P-0-1781-89	CCD: AT real-time container 2, slave n (n = 1-9)
P-0-1790	CCD: AT real-time container 3, master
P-0-1791-99	CCD: AT real-time container 3, slave n (n = 1-9)
P-0-1800	CCD: AT real-time container 4, master
P-0-1801-09	CCD: AT real-time container 4, slave n (n = 1-9)

Tab. 8-97: Product-specific parameters for the cyclic channel of the IndraMotion MLD control

9 RMB_TechCrossCutCrossSeal.library

9.1 Introduction and overview

The "RMB_TechCrossCutCrossSeal" library contains function blocks for cross sealer and cross cutter applications.

RMB_TechCrosscutCrossseal function blocks

Function block	Description
MB_CrossCutSealType01, page 595	calculates, loads and activates a movement profile for cross sealer or cross cutter applications
MB_CrossCutterCalcType04, page 609	Provides at its output all required data 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.

Tab. 9-1: RMB_TechCrossCutCrossSeal.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 609](#) for calculation purposes.

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
IndraMotion MLC 13VRS	RMB_TechCrosscutCrossseal
IndraMotion MLD with MPx07/08/17 with MA function package	MX_Technology07.lib

Tab. 9-2: Reference table of the MB_CrossCutSealType01

RMB_TechCrossCutCrossSeal.library

Interface description

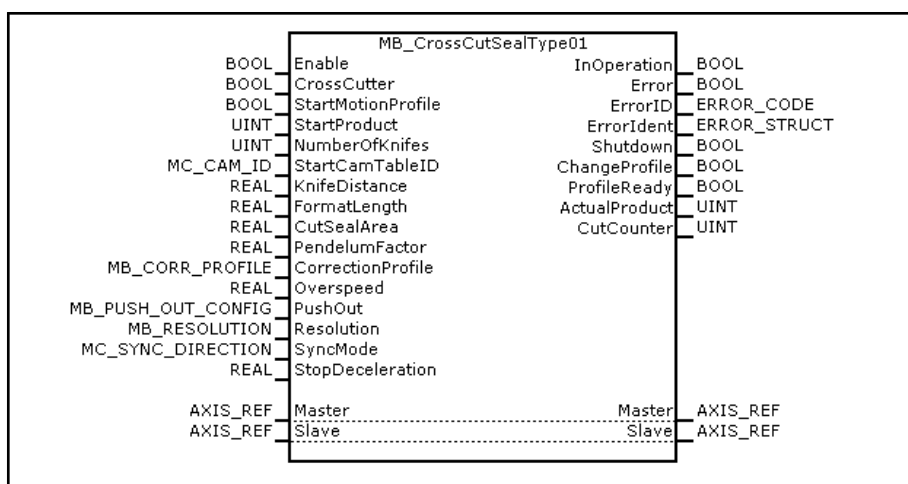


Fig. 9-1: MB_CrossSealType01 function block

I/O type	Name	Type	Comment
VAR_IN_OUT	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
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
	StartMotionProfile	BOOL	The activation of a new MotionProfile is blocked as long as this input is set to FALSE. "StartMotionProfile" = TRUE enables the new MotionProfile if the output "ProfileReady" is TRUE

RMB_TechCrossCutCrossSeal.library

I/O type	Name	Type	Comment
	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 MC_CAM_ID from RIL_CommonTypes
	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 [°]
	PendelumFactor	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 619

RMB_TechCrossCutCrossSeal.library

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	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 MB_PUSH_OUT_CONFIG, page 621
	Resolution	MB_RESOLUTION	The resolution of the cam calculation is specified by determining the number of data points in three stages. MB_RESOLUTION, page 620
	SyncMode	MC_SYNC_DIRECTION	Synchronizes the direction (Shortest_Way, Catch_Up, Slow_Down) MC_SYNC_DIRECTION from RIL_CommonTypes.library
	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 the event of an error
	ErrorIdent	ERROR_STRUCT	See Troubleshooting, page 608
	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

RMB_TechCrossCutCrossSeal.library

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° → 0°) of the slave axis. At a rising edge at "Enable", the "CutCounter" is reset to zero

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° 0 mm	< 360° < KnifeDistance	10° 10mm/in	Continuous*
PendelumFactor	REAL	0.0%	100%	0.0%	Continuous*
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"

RMB_TechCrossCutCrossSeal.library

Name	Type	Min. value	Max. value	Default value	Effective
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, page 607](#)).

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 limits 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.

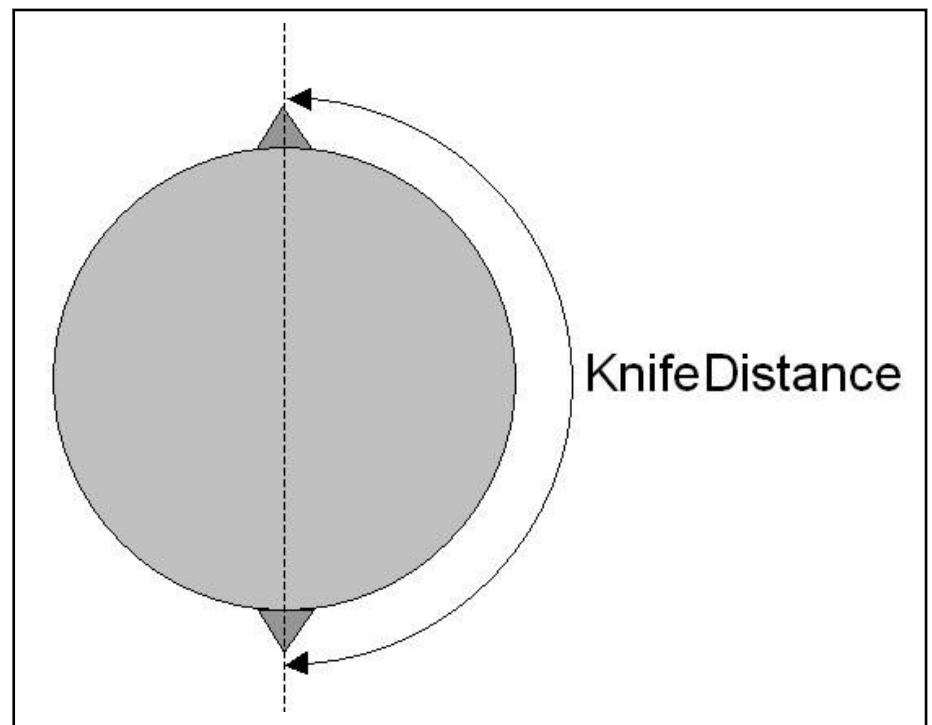


Fig. 9-2: Definition of the input "KnifeDistance"



For scaling:

The translatory 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 translatory 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:

RMB_TechCrossCutCrossSeal.library

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:

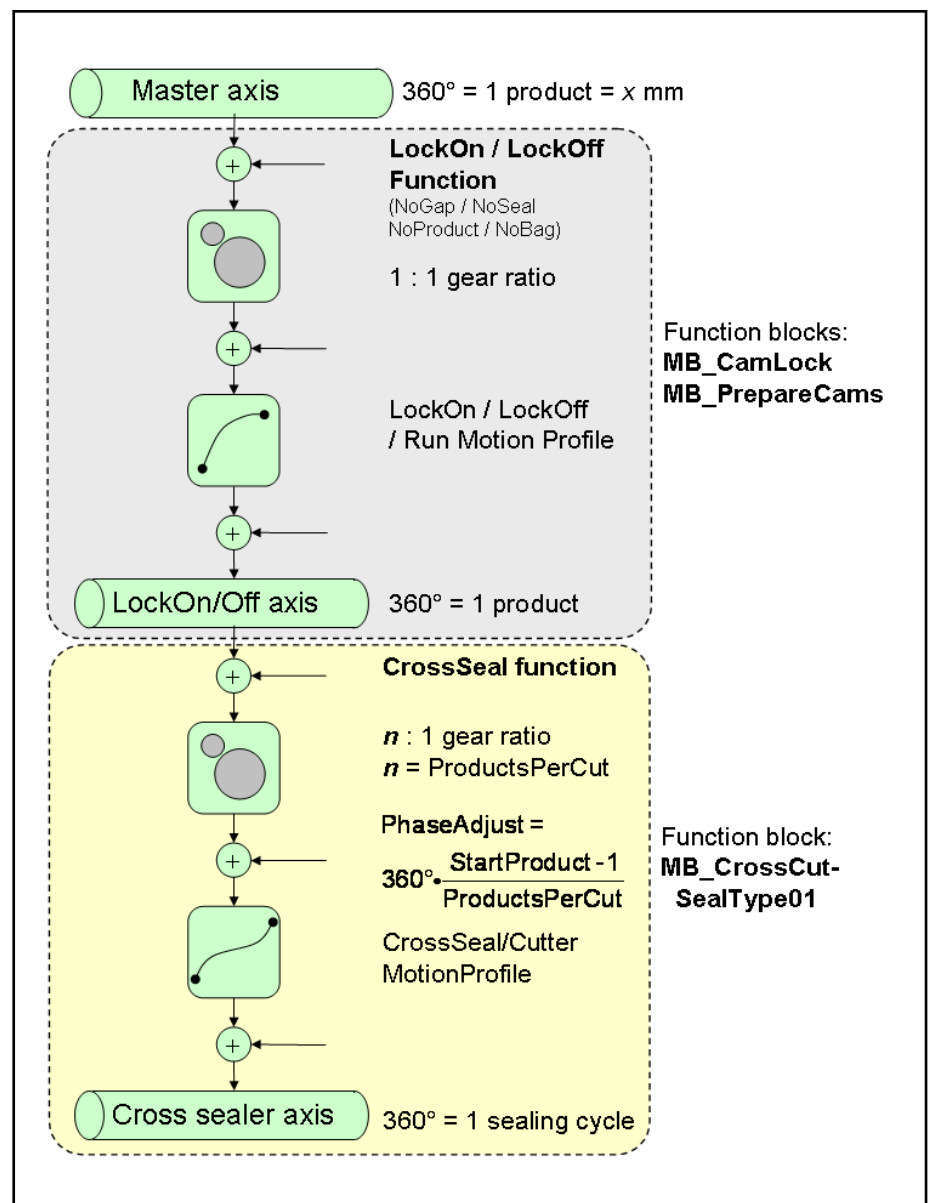


Fig. 9-3: Master axis structure for CrossSealer applications with lock on/lock off

The following figure shows an example of a signal-time diagram of the master axis and the CrossSealer axis for one product per package

RMB_TechCrossCutCrossSeal.library

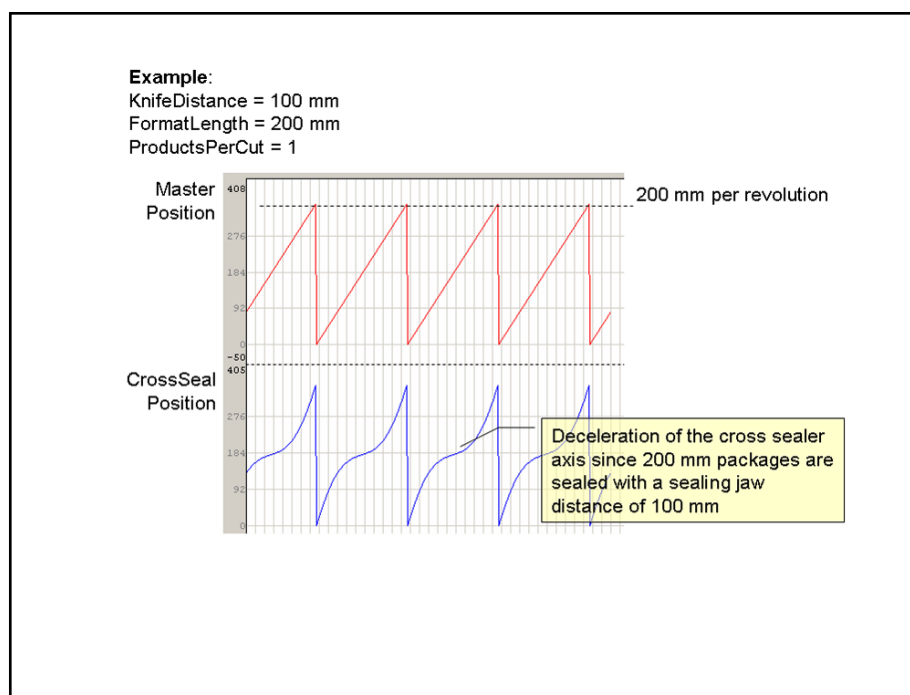


Fig. 9-4: Example of a signal-time diagram with one product per sealing process

The following figure shows an example of a signal-time diagram of the master axis and the CrossSealer axis for 3 products per package:

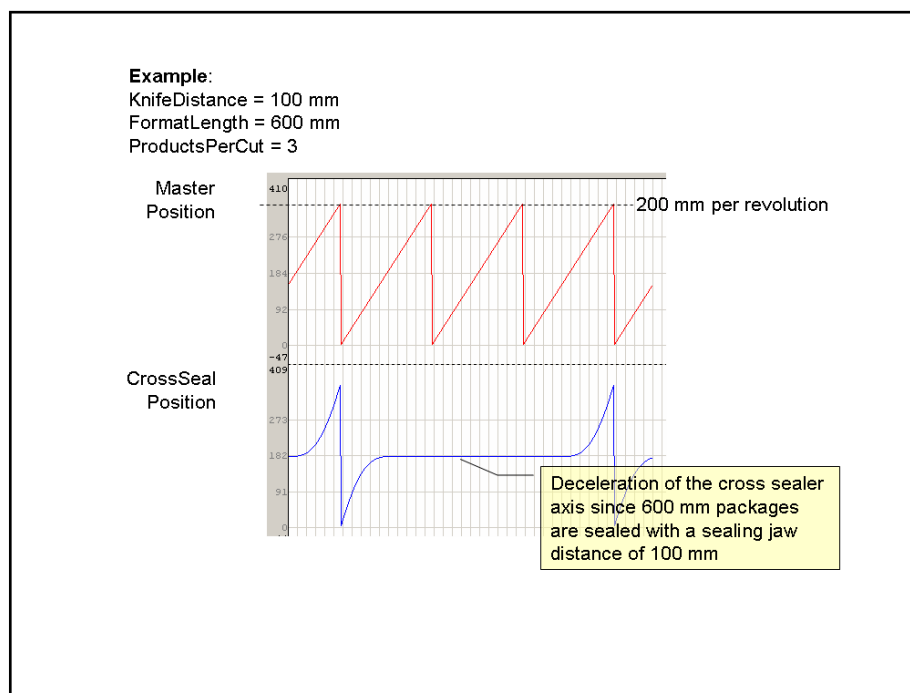


Fig. 9-5: Example of a signal-time diagram with 3 products per sealing process

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:

RMB_TechCrossCutCrossSeal.library

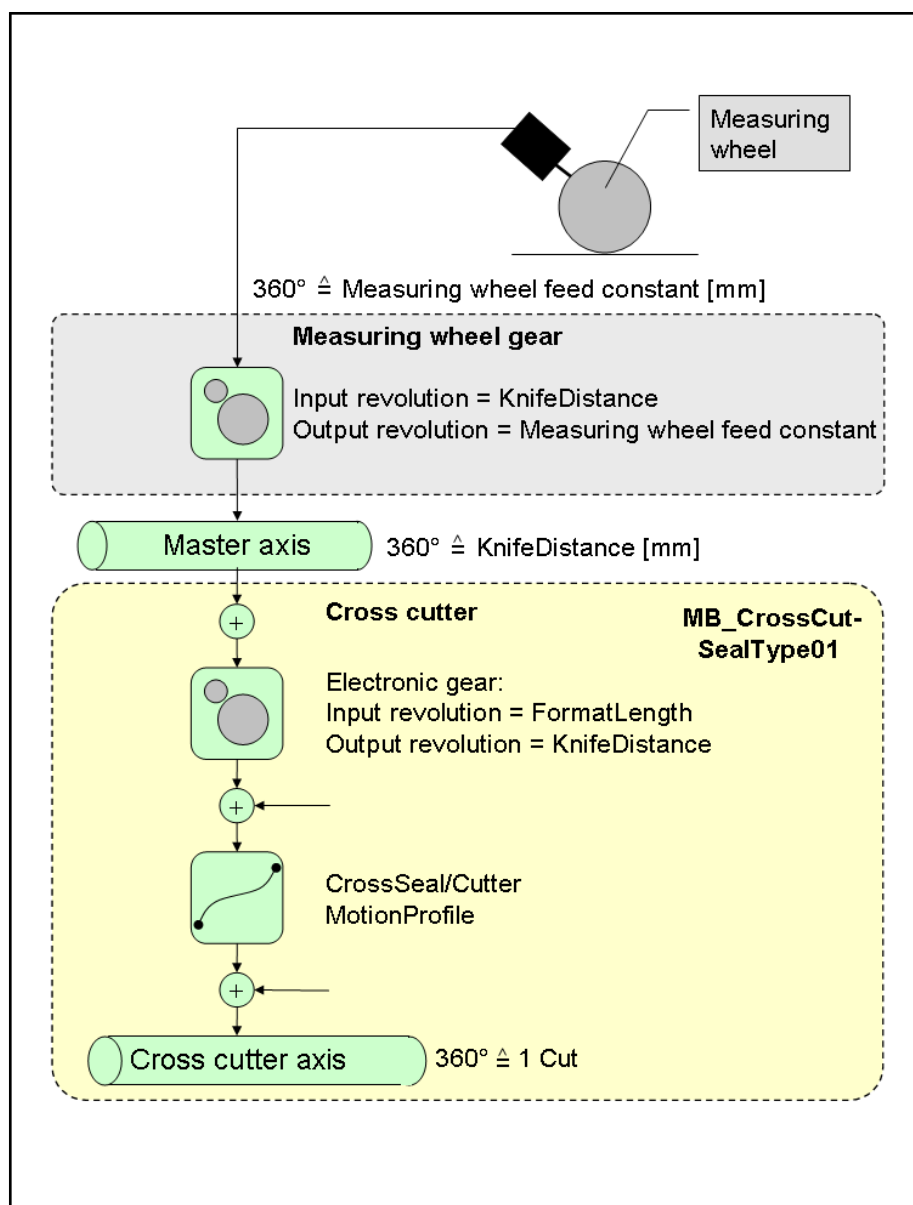


Fig. 9-6: Master axis structure for the CrossCutter application

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):

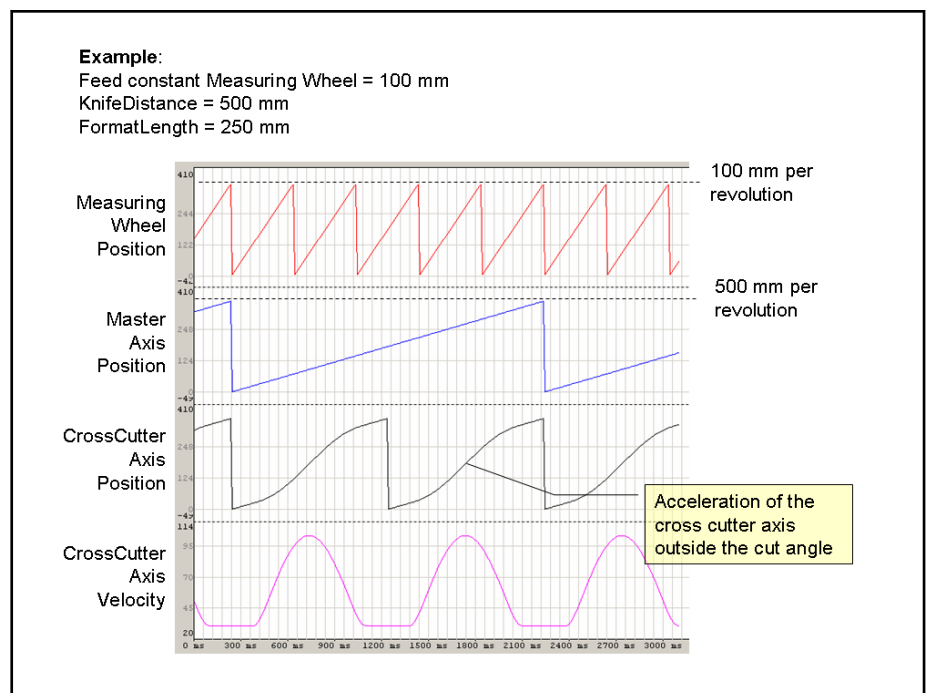


Fig. 9-7: Example of a signal-time diagram for a format length = 0.5 * knife distance

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 * knife distance):

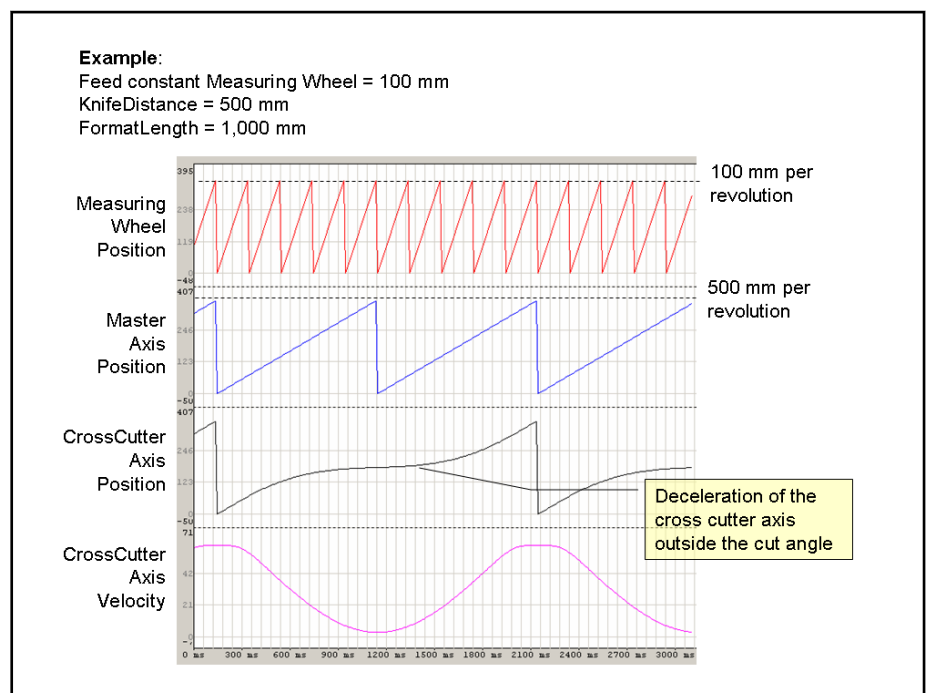


Fig. 9-8: Example of a signal-time diagram for a format length = 02 * knife distance

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

RMB_TechCrossCutCrossSeal.library



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 $n_{input} = 10$

Mechanical gear at the motor, output revolutions $n_{output} = 1$

Number of knives per circumference $n_{knife} = 4$

Calculation of the required gear setting:

$$\text{Mechanical gear} = \frac{\text{Input revolutions}}{\text{Output revolutions}} =$$

$$= \frac{n_{input}}{n_{output} \cdot n_{knife}} = \frac{10}{1 \cdot 4} = \frac{10}{4}$$

- 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

RMB_TechCrossCutCrossSeal.library

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 "PendulumFactor" 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	Library
IndraMotion MLC 13VRS	RMB_TechCrosscutCrossseal
IndraMotion MLD with MPx07/08/17 with MA function package	MX_Technology07.lib

Tab. 9-6: Reference table of the MB_CrossCutterCalcType04

Interface description

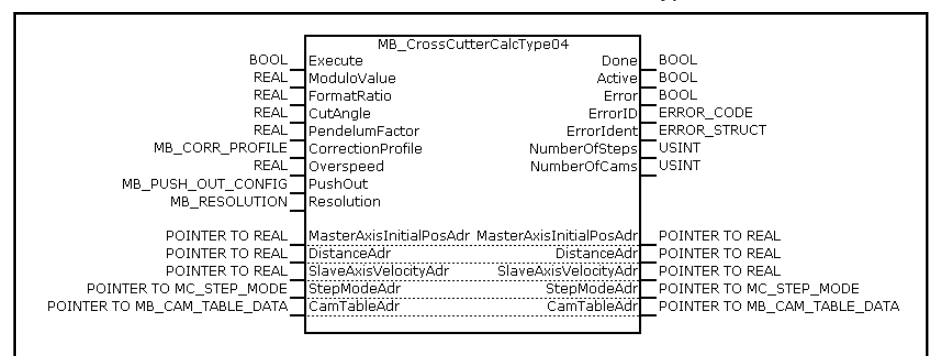


Fig. 9-9: MB_CrossCutterCalcType04 function block

RMB_TechCrossCutCrossSeal.library

I/O type	Name	Type	Comment
VAR_IN_OUT	MasterAxisInitialPosAdr	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
	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. MC_STEP_MODE from RIL_CommonTypes.library
	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 618
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 [°]
	PendulumFactor	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 619
	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_PROFILE_OVERSPEED option has to be selected. Otherwise, "Overspeed" is deactivated and thus ineffective

RMB_TechCrossCutCrossSeal.library

I/O type	Name	Type	Comment
	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. MB_PUSH_OUT_CONFIG, page 621
	Resolution	MB_RESOLUTION	The resolution of the cam calculation is specified by determining the number of data points in three stages. MB_RESOLUTION, page 620
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
	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 Troubleshooting, page 616
	NumberOfSteps	USINT	Number of generated and to be downloaded Motion-Profile segments
	NumberOfCams	USINT	Number of generated and to be downloaded cams

Tab. 9-7: Interface variables of the 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"
PendulumFactor	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"

RMB_TechCrossCutCrossSeal.library

Name	Type	Min. value	Max. value	Default value	Effective
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

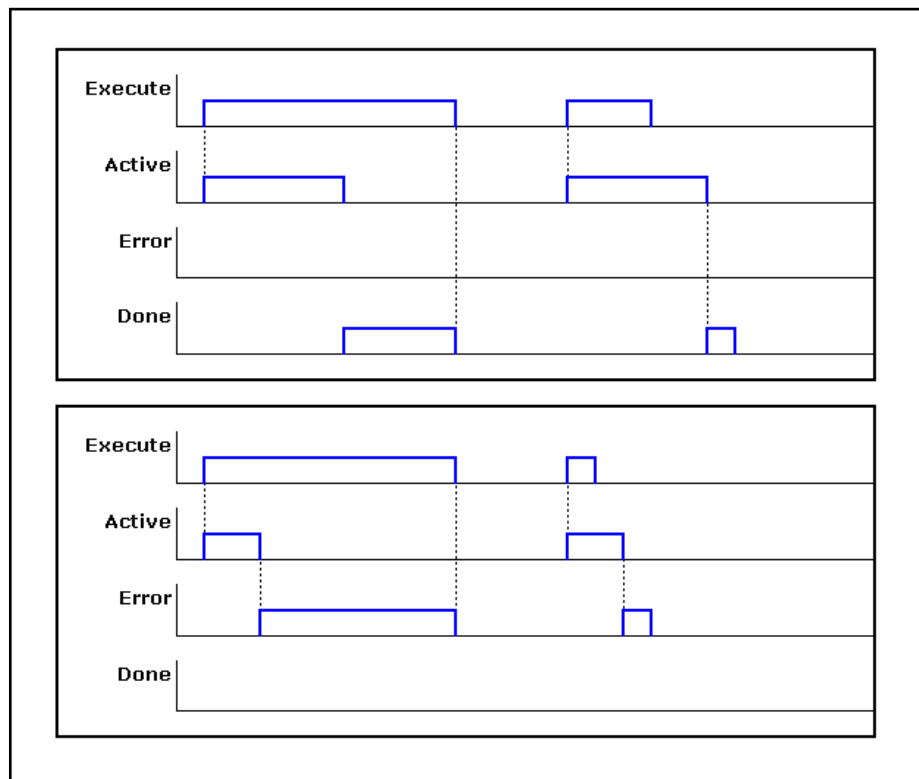


Fig. 9-10: Signal-time diagram of the MB_CrossCutterCalcType04 function block

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 Cross-Cutter 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

RMB_TechCrossCutCrossSeal.library

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 "FormatRatio" 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.

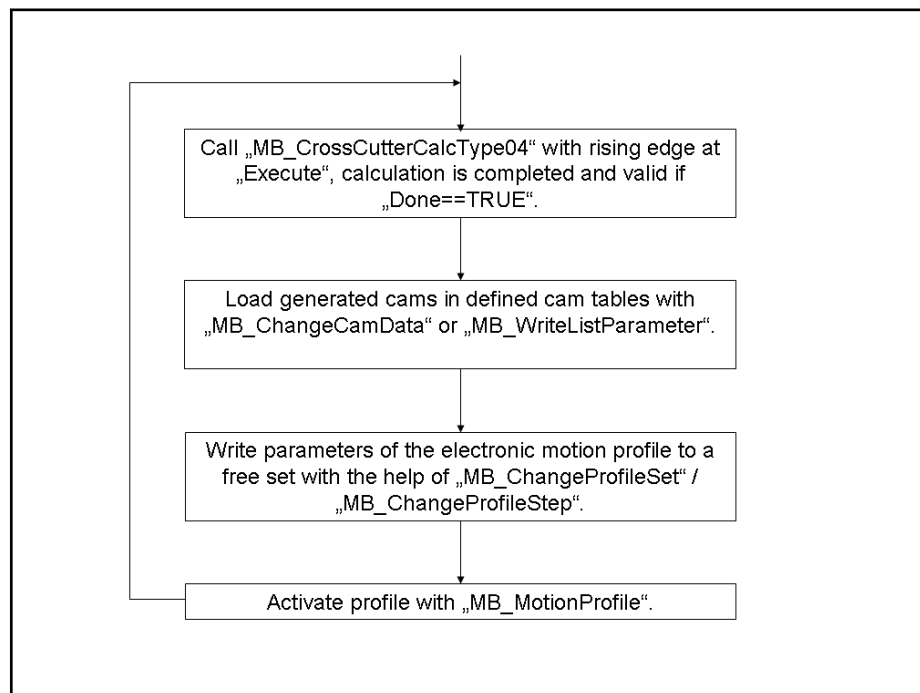


Fig. 9-11: Sequence diagram for using MB_CrossCutterCalcType04

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.

RMB_TechCrossCutCrossSeal.library

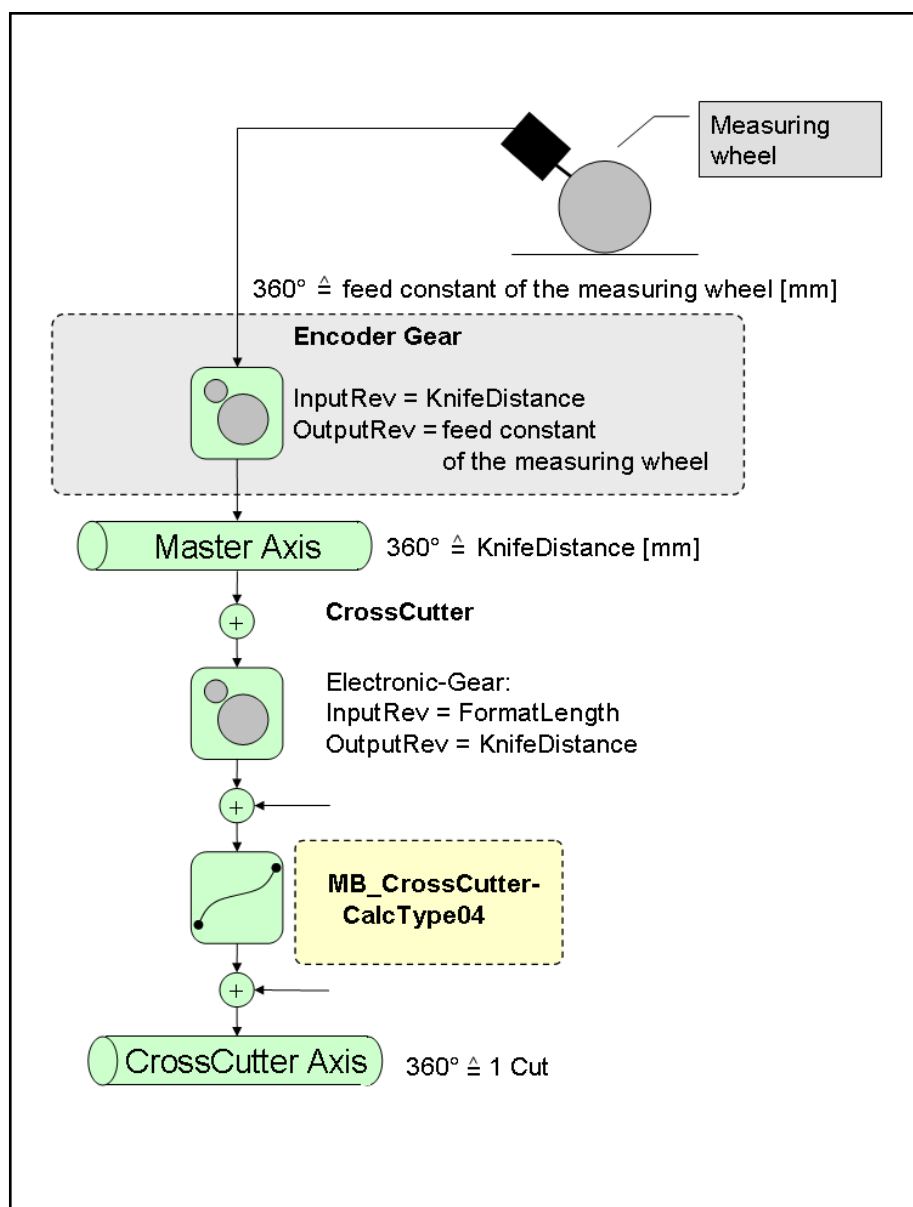


Fig. 9-12: Master axis structure when operating a CrossCutter

Troubleshooting

The function block uses the error table **F_RELATED_TABLE**, 16#061x. 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 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"

ErrorID	Additional1	Additional2	Description
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 function block MB_CrossCutterCalcType04

Special features of IndraMotion MLC

With regard to the IndraMotion 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 IndraMotion 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, it is to be ensured that the respective cam is loaded at its assigned position (see "CamTableID" element in the "Interface variables of the MB_CAM_TABLE_DATA structure" table (see [MB_CAM_TABLE_DATA, page 618](#)))

- A small format ratio (< 1) should be selected carefully since the CrossCutter 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

RMB_TechCrossCutCrossSeal.library

Format ratio	Ratio of the slave axis velocity to the master axis velocity
1	1
0,	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_CAM_TABLE_DATA

Brief Description The MB_CAM_TABLE_DATA structure is used by the MB_CrossCutterCalc-Type04 function block. It 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.

Interface Description The following table lists the interface variables of the function block.

Structure 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. 9-11: Interface variables of the MB_CAM_TABLE_DATA structure

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

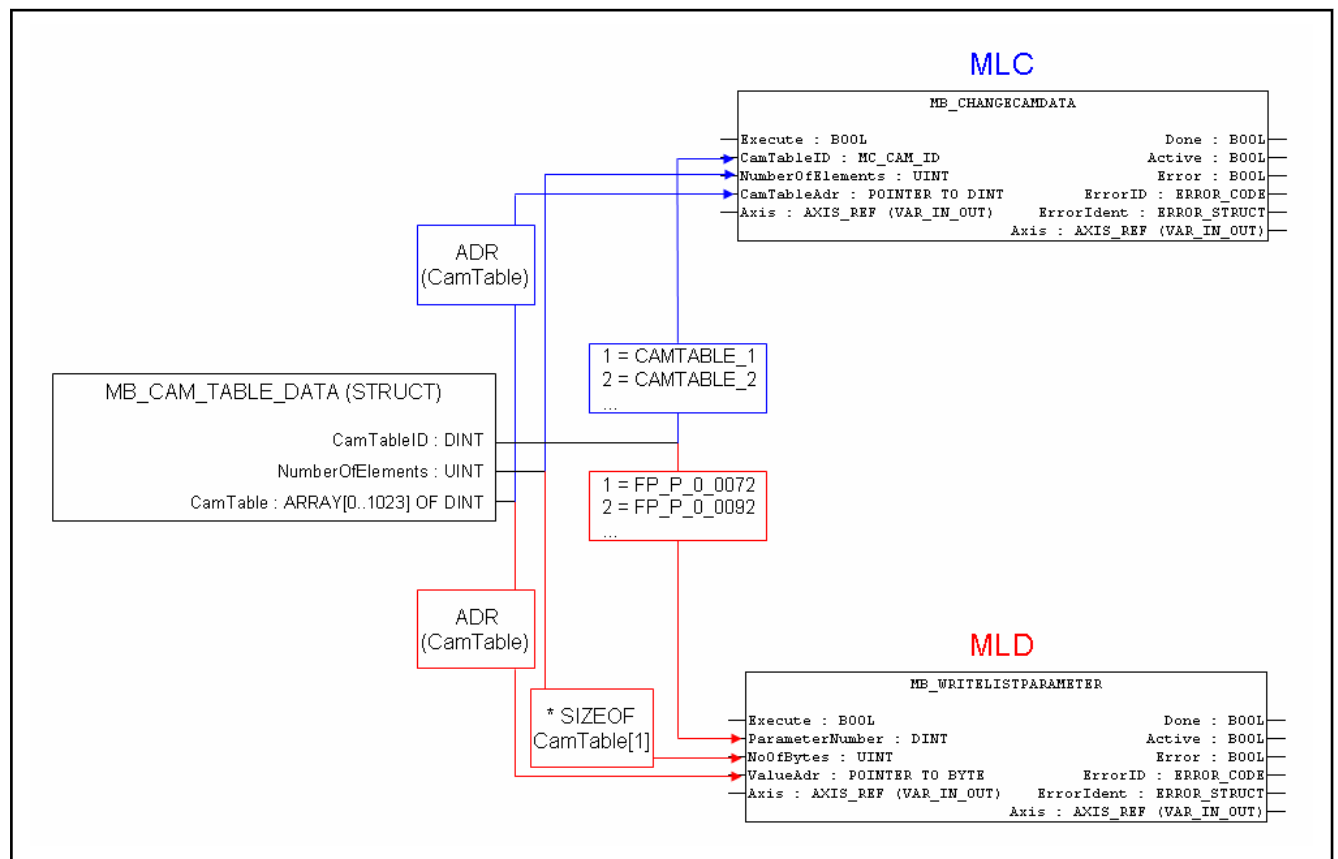
Tab. 9-12: Min./max. and default values of the MB_CAM_TABLE_DATA structure

Functional Description In the structure element "CamTableID", the number of the cam table is saved in the format required by the MB_ChangeCamData, MB_ChangeProfileStep and MB_ChangeProfileSet function blocks.

[Assigning CamTableID to parameter numbers, page 618](#), shows the mappings between "CamTableID" and the corresponding parameter number in the MLC and MLD.

CamTableID	Parameter number IndraMotion MLC	Parameter number IndraMotion 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

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.



9.5 MB_CORR_PROFILE

Brief Description	The enumeration MB_CORR_PROFILE is used by the MB_CrossCutterCalc-Type04 function block to define a correction motion in the cutting area.

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

RMB_TechCrossCutCrossSeal.library

Element	Value	Description
CORR_PROFILE_COS_COMP	2	Correction of the CrossCutter axis velocity The velocity is changed in a way that the CrossCut-ter axis rotates in material direction (the x compo-nent of the velocity vector) at a constant velocity. The speed of the CrossCutter axis has to be adap-ted depending on the current position in the cut
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 621
CORR_PROFILE_COS_COMP_AND_PUSH_OUT	4	Combination of the corrections "CORR_PRO-FILE_COS_COMP" and "CORR_PRO-FILE_PUSH_OUT". Parameterization via the "Push-Out" input of the MB_CrossCutterCalcType04 func-tion block. MB_PUSH_OUT_CONFIG, page 621

Tab. 9-14: Elements of the enumeration type MB_CORR_PROFILE

9.6 MB_RESOLUTION

Brief Description The enumeration MB_RESOLUTION is used by the MB_CrossCutterCalc-Type04 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 seg-ment + compensating segment + cut-ting segment)
RESOLUTION_LOW	1	Low resolution, short calculation time re-quired (faster by approx. factor 3 in con-trast 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. 9-15: 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 calcula-tion time of the function block can be reduced selecting the "RESOLU-TION_LOW" option. However, it is 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 ("RESOLU-TION_HIGH") it 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.

9.7 MB_PUSH_OUT_CONFIG

Brief Description The MB_PUSH_OUT_CONFIG structure is used by the MB_CrossCutter-CalcType04 function block. This structure can parameterize a defined Push-Out at the end of the cut.

Interface Description The following table lists the interface variables of the function block.

Structure 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. 9-16: Interface variables of the MB_PUSH_OUT_CONFIG structure

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. 9-17: 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 velocity 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.

RMB_TechCrossCutCrossSeal.library

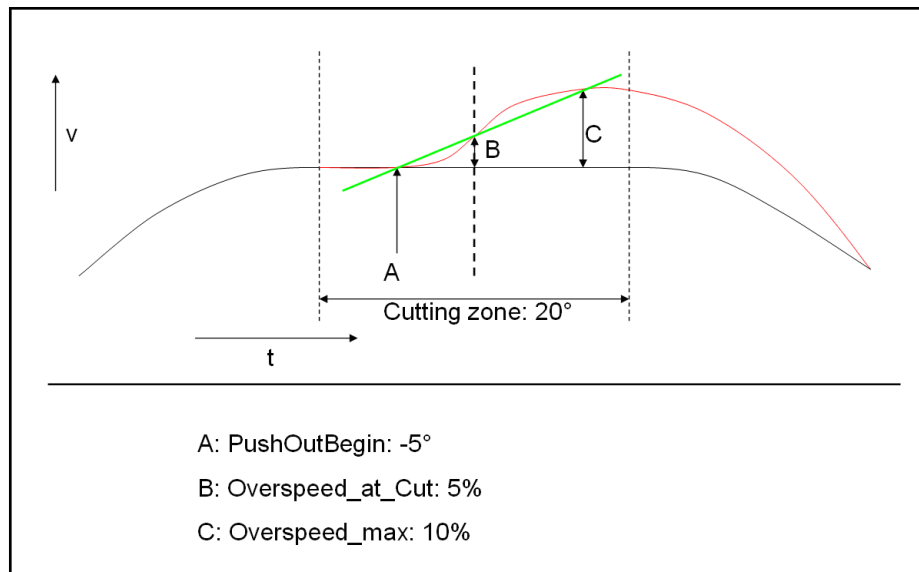


Fig. 9-14: Definition of PushOut.PushOutBegin < 0

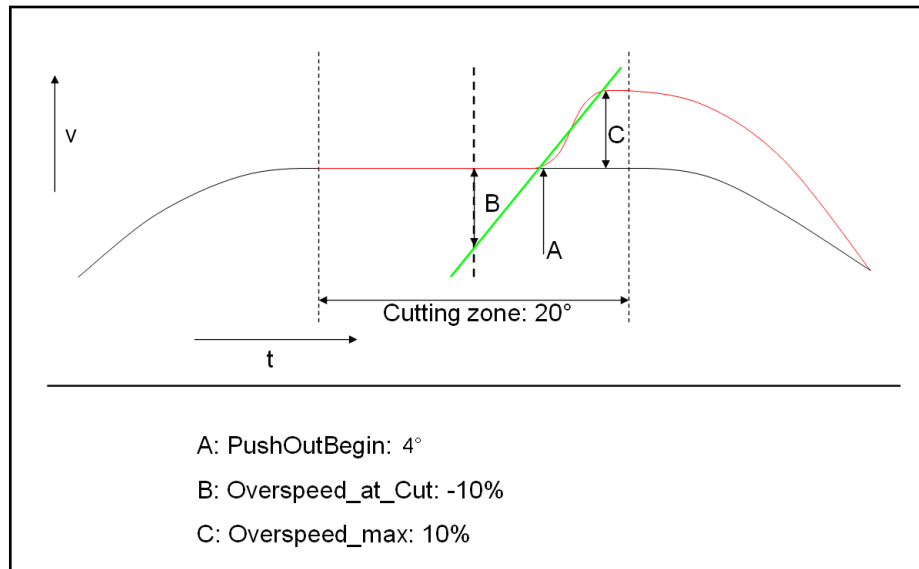


Fig. 9-15: Definition of PushOut.PushOutBegin > 0

10 RMB_TechCam.library

10.1 Introduction and overview

The function blocks of the RMB_TechCam.library are available for the different applications such as feed motion, CrossCutter, slotter to create cam tables.

The function blocks MB_FeedMovementCalcType01, MB_FeedMovementCalcType02 and MB_CoordinatePairCamType01 operate with the enumeration MB_CAMPROF, ML_TechBase.library and the structure MB_CAMTABLE, ML_TechBase.library.

The function blocks MB_SlotterCalcType01 und MB_SlotterCalcType02 operate with the enumeration "MB_CAMPROF_SLOTTER" and the same structure MB_CAMTABLE, ML_TechBase.library.

Furthermore, the cam tables of the mentioned function blocks can be loaded to a cam in the control or drive using the MB_LoadCamData, ML_TechBase.library.

RMB_TechCam function blocks

Function block	Description
MB_CoordinatePairCamType01, page 623	Calculates a cam table with up to 1,024 data point elements determined with regard to the selected coordinate pair type
MB_FeedMovementCalcType01, page 631	Calculates a cam table with up to 1,024 data point elements for a feed motion
MB_FeedMovementCalcType02, page 637	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 648	Calculates a cam table with up to 1,024 data point elements for a slotter with one knife on the slotter roll
MB_SlotterCalcType02, page 660	Calculates a cam table with up to 1,024 data point elements for a slotter with two knives on the slotter roll

Tab. 10-1: RMB_TechCam.library

10.2 MB_CoordinatePairCamType01

Brief description

The function block calculates a cam table with up to 1,024 data point elements that are determined depending on the selected coordinate pair type.

Assignment: Target system/library

IndraMotion MLC 13VRS	RMB_TechCam

Tab. 10-2: Reference table of the MB_CoordinatePairCamType01

Interface description

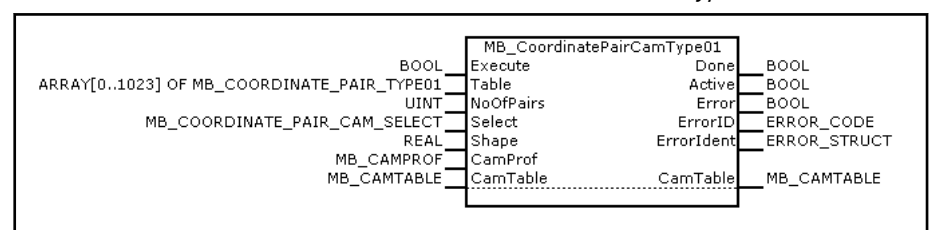


Fig. 10-1: MB_CoordinatePairCamType01 function block

RMB_TechCam.library

I/O type	Name	Type	Comment
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements
VAR_INPUT	Execute	BOOL	Processing of the function block enabled (once, edge-controlled)
	Table	ARRAY[0..1023] OF MB_COORDINATE_PAIR_TYPE01	List of coordinate pairs MB_COORDINATE_PAIR_TYPE01, page 630
	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 630
	Shape	REAL	Factor for jerk limitation
	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 Error handling, page 630)

Tab. 10-3: Interface variables of the 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 45](#), 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.

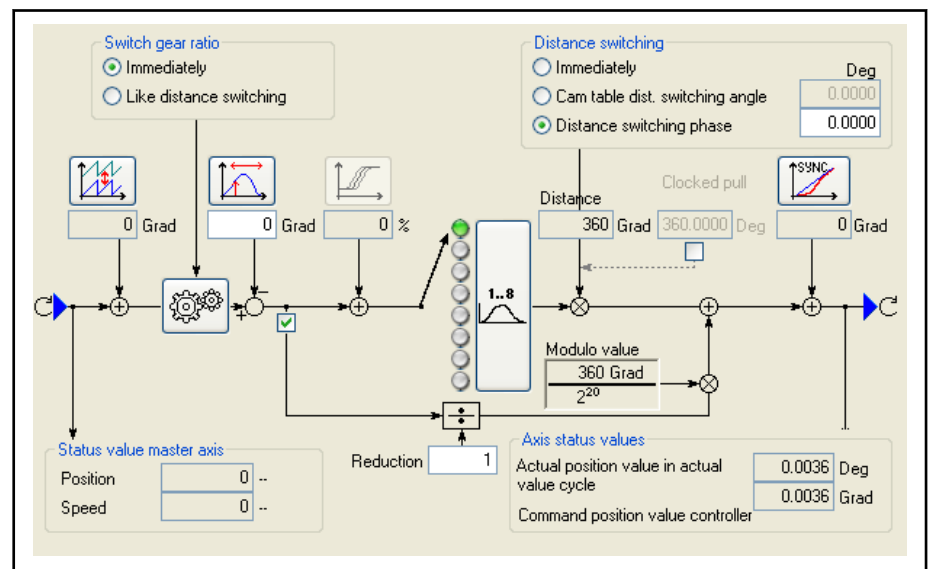


Fig. 10-2: Enabling the linear path

The structure [MB_CAMTABLE](#), [page 46](#), 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

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 is 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.

RMB_TechCam.library

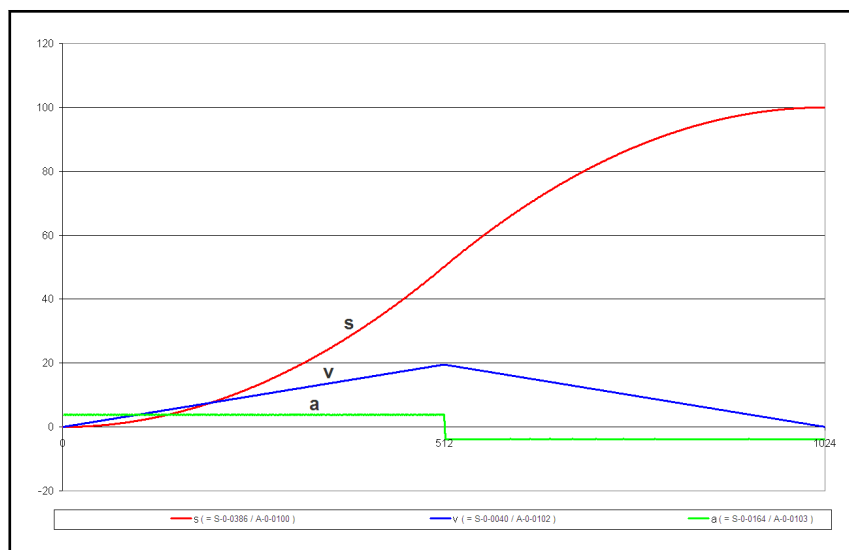


Fig. 10-3: MB_CoordinatePairCamType01, MotionProfile COORDI-
NATE_PAIR_POS_VEL_TYPE01

- 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.

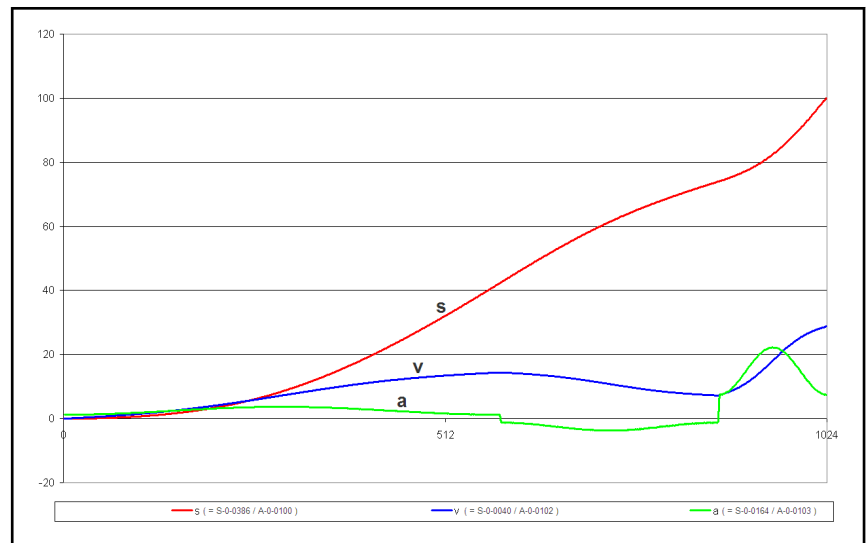


Fig. 10-4: MB_CoordinatePairCamType01, MotionProfile COORDI-
NATE_PAIR_POS_VEL_TYPE02

- 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.

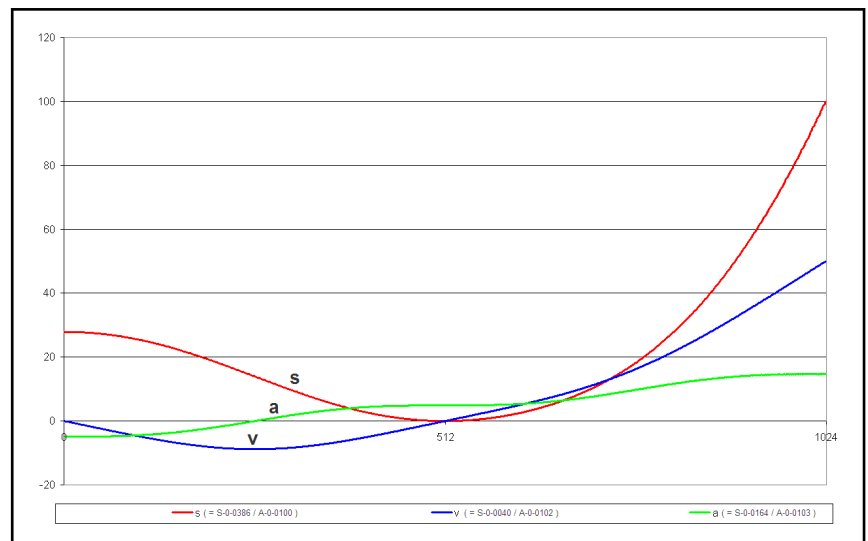
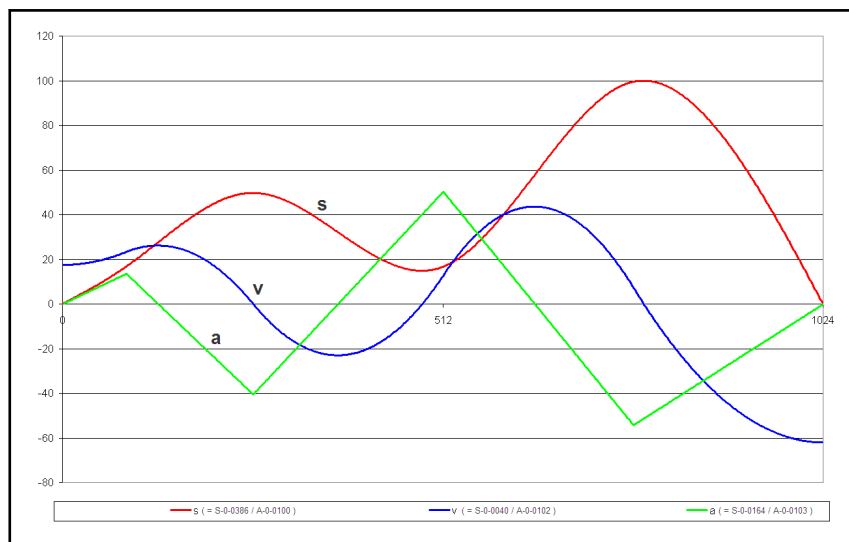


Fig. 10-5: MB_CoordinatePairCamType01, MotionProfile COORDI-
NATE_PAIR_POS_ACC

- Select: COORDINATE_PAIR_POS_POS_TYPE01

RMB_TechCam.library

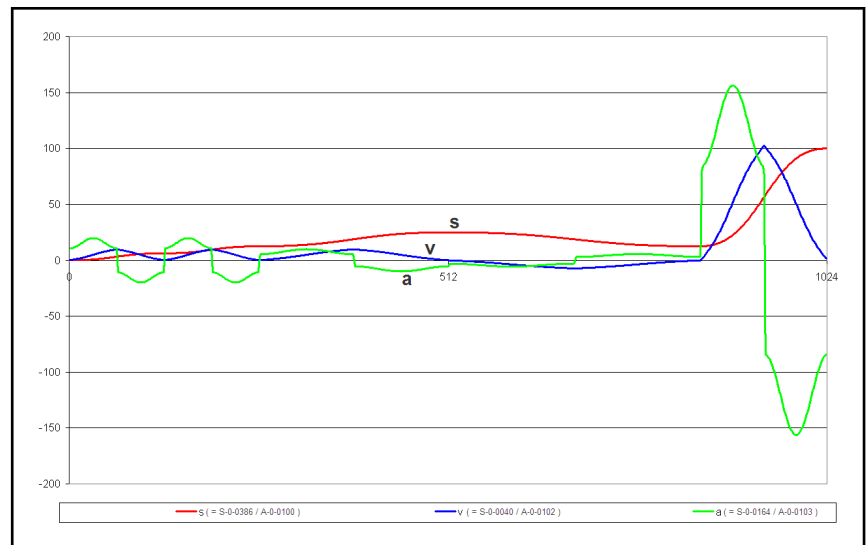
- To calculate the cam table, at least two coordinate pairs have to be given.
- It is traveled through the individual value pairs without rest. The target positions are connected with a spline polynomial 3rd order.



Coordinate pairs: (0;0),(30;10),(90;30),(180;10),(270;60),
(360;0)

Fig. 10-6: MB_CoordinatePairCamType01, MotionProfile COORDI-
NATE_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

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](#), page 43, 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_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

RMB_TechCam.library

- 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)

Error handling The function block uses the F_RELATED_TABLE, 16#0170 error table. 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
CALCULATION_ERROR	16#0000000C	16#00000000	Division by Zero
INPUT_RANGE_ERROR	16#00002200	16#00000001	"NumberOfElements" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#00002270	16#00000001	"NoOfPairs" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#00002270	16#00000002	"Shape" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#00002270	16#00000003	Input element of "Table" is < minimum or > maximum
INPUT_INVALID_ERROR	16#00002201	16#00000001	Address of the "CamTable" array is zero
INPUT_INVALID_ERROR	16#00002271	16#00000001	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_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. 10-6: MB_COORDINATE_PAIR_TYPE01 structure with min., default,- and max. value

10.4 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

Element	Value	Description
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. 10-7: Enumeration type MB_COORDINATE_PAIR_CAM_SELECT

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

IndraMotion MLC 13VRS	RMB_TechCam

Tab. 10-8: Reference table of the MB_FeedMovementCalcType01

Interface description

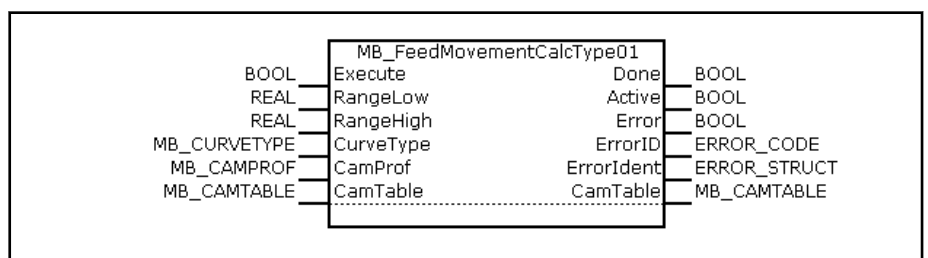


Fig. 10-8: MB_FeedMovementCalcType01 function block

I/O type	Name	Type	Comment
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements
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 MotionProfile
	CurveType	MB_CURVETYPE	Curve type selection MB_CURVETYPE, page 637
	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 Error handling, page 636

Tab. 10-9: Interface variables of the MB_FeedMovementCalcType01 function block

RMB_TechCam.library

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"
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_CAMPF			CAM_PROF_STANDARD	Rising edge at "Execute"

Tab. 10-10: Min./max. and default values of the MB_FeedMovementCalcType01 function block

The enumeration [MB_CAMPF](#), page 45, selects between different cam profiles.



When selecting the cam profile "CAMF_STANDARD", the linear path has to be disabled since it is a profile without gear reduction. The linear path for the profiles "CAMF_ADDITIVE" and "CAMF_ADDITIVE_SEGMENTABLE" has to be ticked and the reduction has to be set to "1", since they are profiles with gear reduction.

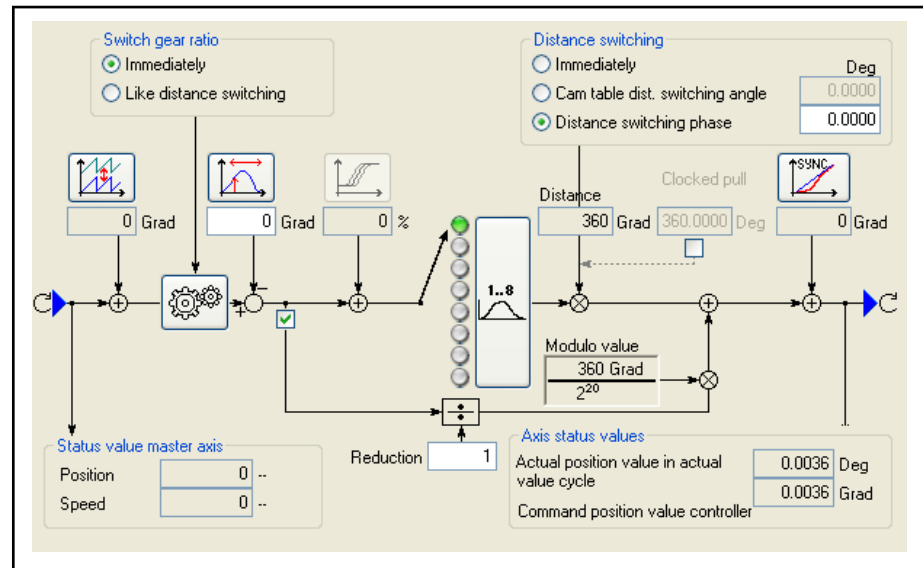


Fig. 10-9: Enabling the linear path

The structure [MB_CAMTABLE](#), page 46, 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

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:

- Quadratic parabola
- Simple sine curve
- Polynomial 5th order
- Inclined sine curve
- Modified acceleration trapezoid
- Modified sine curve

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, page 43](#), function block.

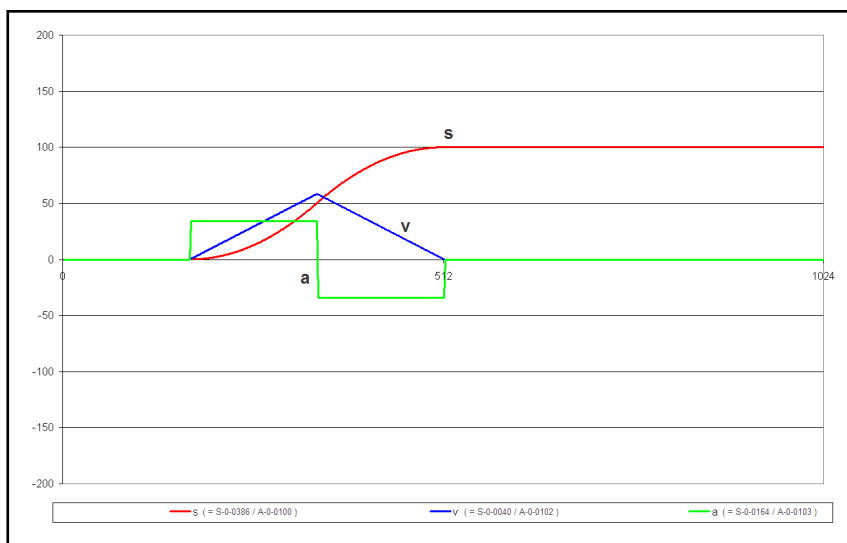
If an error occurs while processing the function block, this is signaled by the "Error" output. The elements of the "CamTable" structure are not updated in case of error.

Cam characteristics

The cam table of the MB_FeedMovementCalcType01 provides the following characteristics:

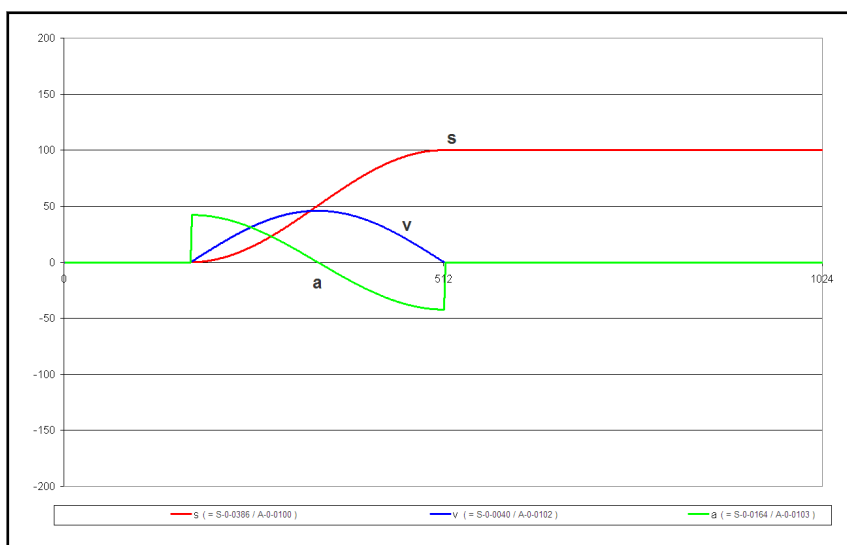
- The cam table generated by the function block describes a rest-in-rest motion
- The algorithm to calculate the cam data points depends on a difference (RangeHigh-RangeLow) of 1,026 cycles max, i.e. calls of the instance to avoid a considerable effect on the PLC cycle time
- The cam profile is provided 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)

RMB_TechCam.library



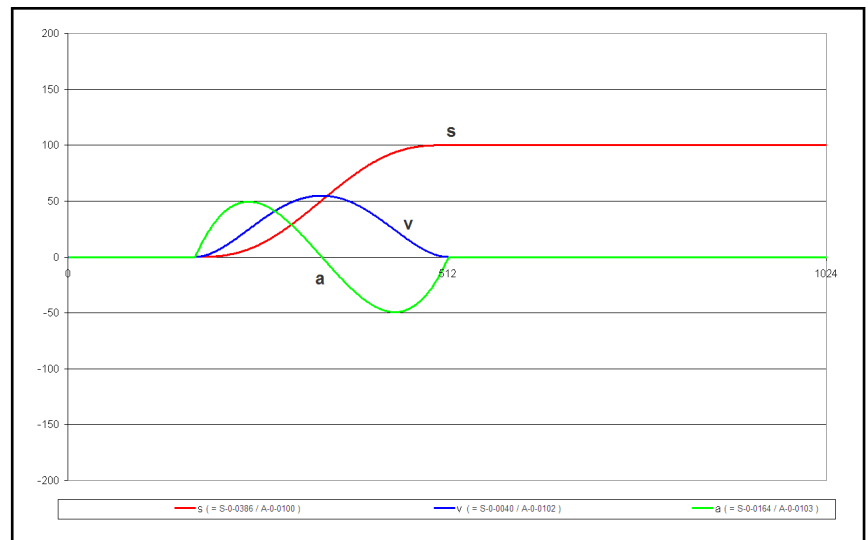
- CurveType: CURVETYPE_SIMPLE_SINUSLINE

- The MotionProfile characterized by the cam is constant up to the acceleration (s, v, a) and shows a jump in the jerk (j)



- CurveType: CURVETYPE_POLY_5

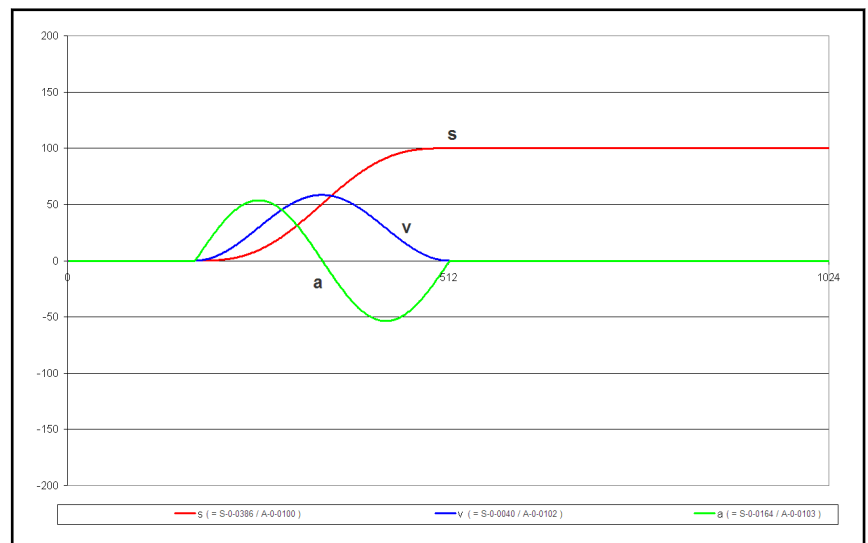
- The MotionProfile characterized by the cam is constant up to the jerk (s, v, a, j)



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)

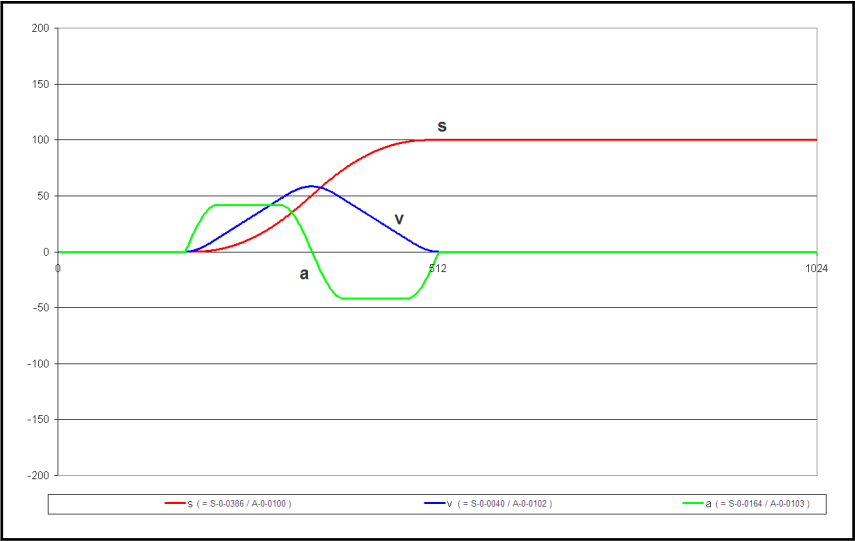


RangeLow = 60; RangeHigh = 180

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)

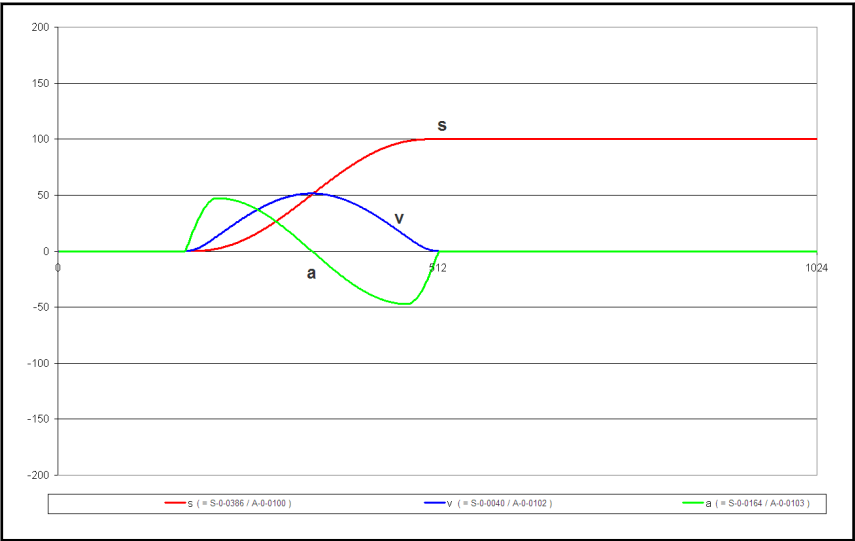
RMB_TechCam.library



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 F_RELATED_TABLE, 16#0170 error table. 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
CALCULATION_ERROR	16#0000000C	16#00000000	Division by Zero
INPUT_RANGE_ERROR	16#00002200	16#00000001	"NumberOfElements" input is < minimum or > maximum

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR	16#00002220	16#00000001	"RangeLow" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#00002220	16#00000002	"RangeHigh" input is < minimum or > maximum
INPUT_INVALID_ERROR	16#00002201	16#00000001	Address of the "CamTable" array is zero
INPUT_INVALID_ERROR	16#00002221	16#00000001	"RangeLow" input is $\geq$ RangeHigh

Tab. 10-11: Error codes of the MB_FeedMovementCalcType01 function block

10.6 MB_CURVETYPE

The "MB_CURVETYPE" enumeration selects between different compensation motions from rest-to-rest.

Element	Value	Description
CURVETYPE_QUADRATIC_PARABLE	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. 10-12: Elements of the enumeration type "MB_CURVETYPE"

10.7 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

IndraMotion MLC 13VRS	RMB_TechCam

Tab. 10-13: Reference table of the MB_FeedMovementCalcType02

Interface description

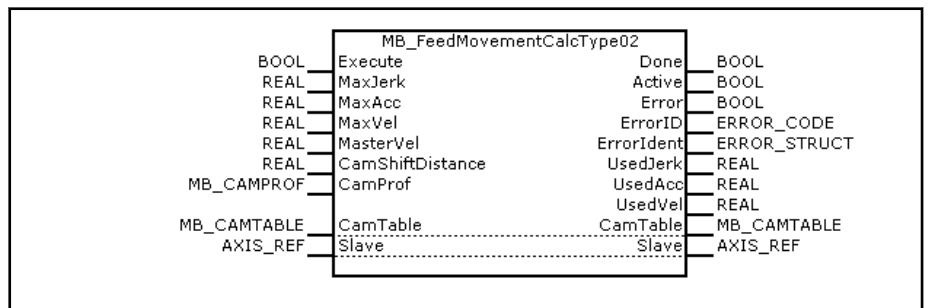


Fig. 10-16: MB_FeedMovementCalcType02 function block

I/O type	Name	Type	Comment
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements
	Slave	AXIS_REF	Reference to the slave axis

RMB_TechCam.library

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
	CamShaftDistance	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 Error handling, page 642
	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 [%]

Tab. 10-14: 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"
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. 10-15: Min./max. and default values of the MB_FeedMovementCalcType02 function block

The enumeration [MB_CAMPROF](#), [page 45](#), 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.

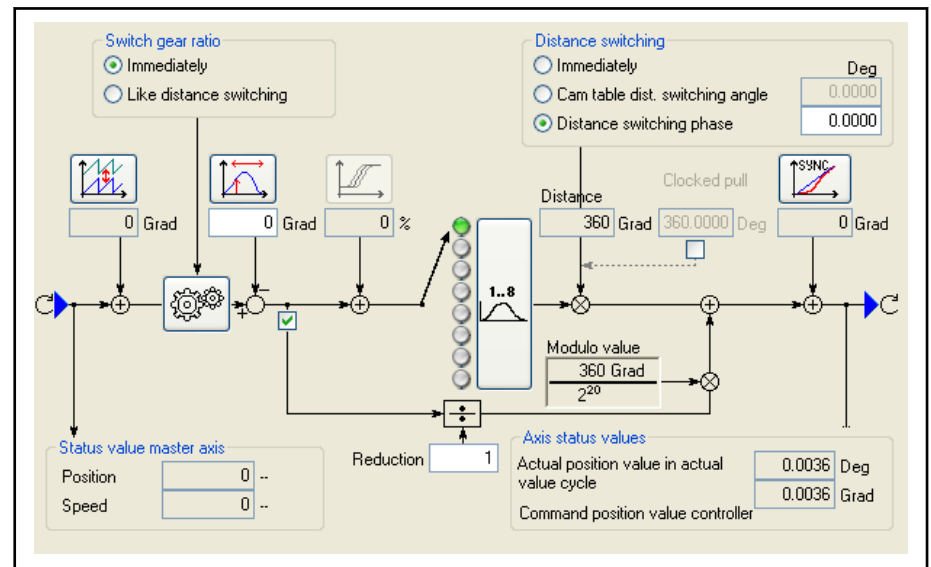


Fig. 10-17: Enabling the linear path

The structure [MB_CamTABLE](#), [page 46](#), 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

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 calculate 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.

Note:

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 to the control or drive using the [MB_LoadCamData](#), page 43, 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

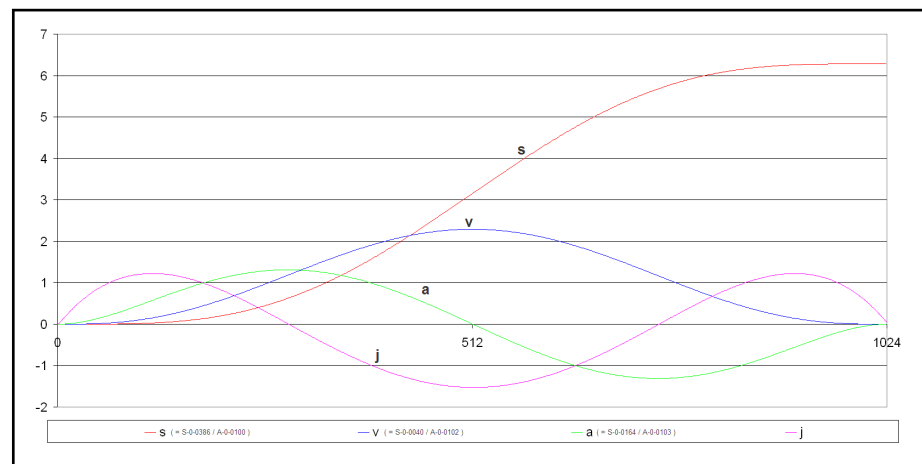
Overview on possible calculation procedures

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.

Note:

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.

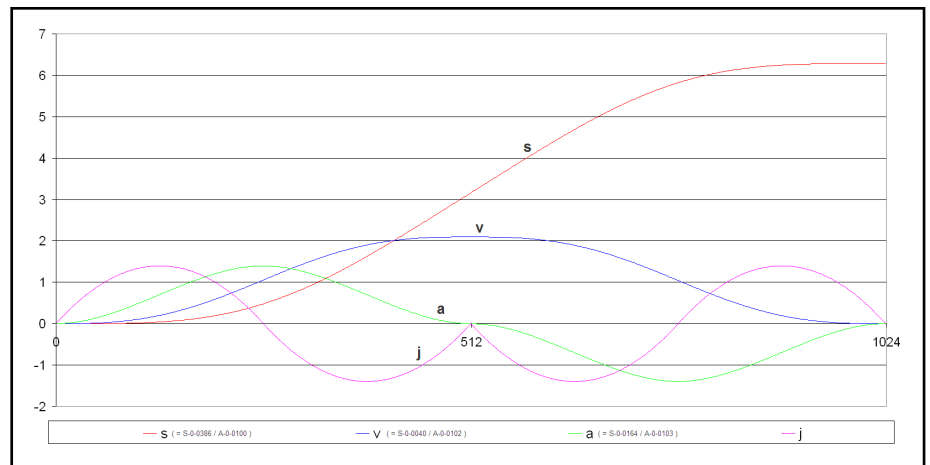
Variant 1: Polynomial 7th order



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)]

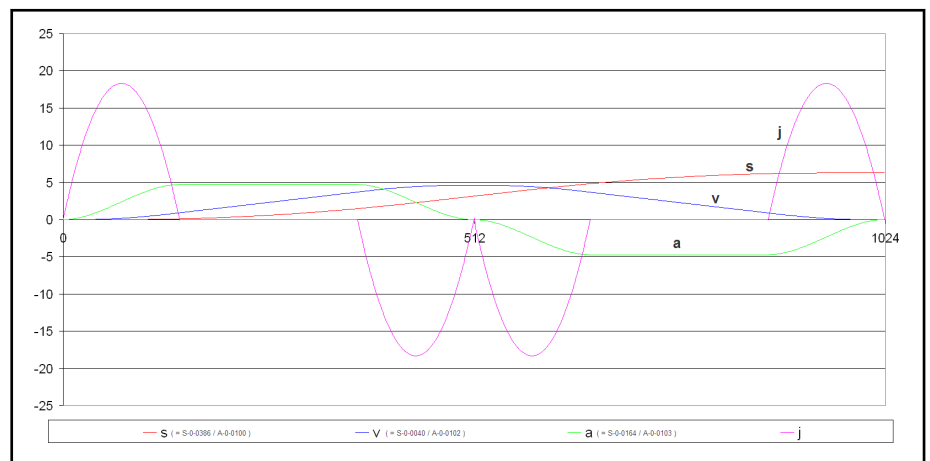
Variant 2: Velocity and acceleration limit values are not reached (variant is rarely used in practice)



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)

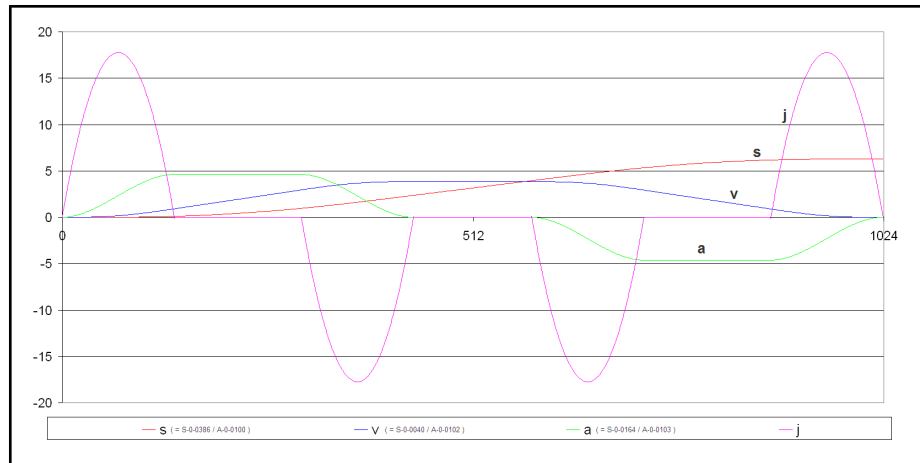


MaxJerk = 20 rad/s³; MaxAcc = 5 rad/s²; MaxVel = 5 rad/s; MasterVel= 22 rpm; modulo value = 2*Pi rad
Calculation from parabola parts

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)

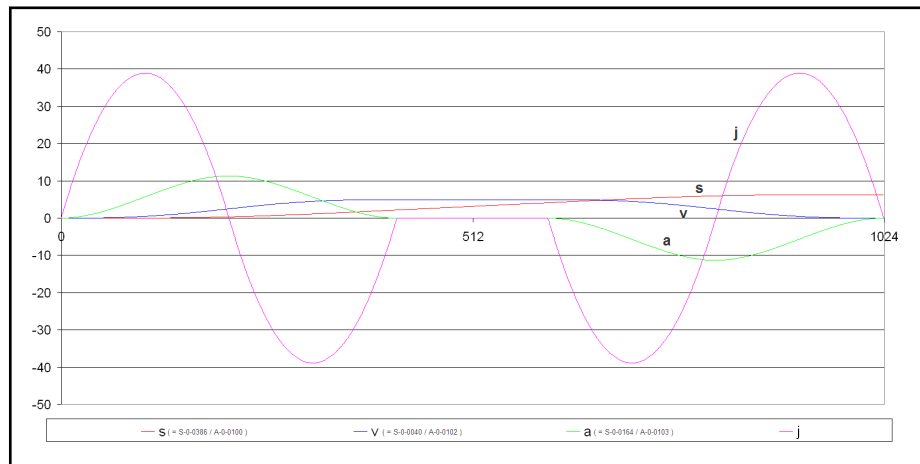
RMB_TechCam.library



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#00000006	16#00000000	Invalid function block state
INPUT_RANGE_ERROR	16#00002200	16#00000001	"NumberOfElements" input is < minimum or > maximum
INPUT_RANGE_ERROR	16#00002230	16#00000001	"MaxJerk" input is < minimum
INPUT_RANGE_ERROR	16#00002230	16#00000002	"MaxAcc" input is < minimum
INPUT_RANGE_ERROR	16#00002230	16#00000003	"MaxVel" input is < minimum
INPUT_RANGE_ERROR	16#00002230	16#00000004	"MasterVel" input is < minimum

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR	16#00002230	16#00000005	"CamShaftDistance" input is < minimum
INPUT_INVALID_ERROR	16#00002201	16#00000001	Address of the "CamTable" array is zero
INPUT_INVALID_ERROR	16#00002231	16#00000001	"CamShaftDistance" unequal to the modulo value of the axis reference
INPUT_INVALID_ERROR	16#00002231	16#00000002	Input values for jerk, acceleration or velocity are higher than preset in the axis parameters
CALCULATION_ERROR	16#00002232	16#00000001	Velocity of the master axis is too high or the maximum jerk, the maximum acceleration or the maximum velocity is too low.

Tab. 10-16: Error codes of the MB_FeedMovementCalcType02 function block

10.8 Slotter: Introduction and definitions

A slotter is a rotating roll with one or more knives attached to it which produce slots in the corrugated cardboard sheet in a corrugated cardboard printing machine. Thus, the knives can cut the subsequent bottom and cover flaps in a corrugated cardboard box.

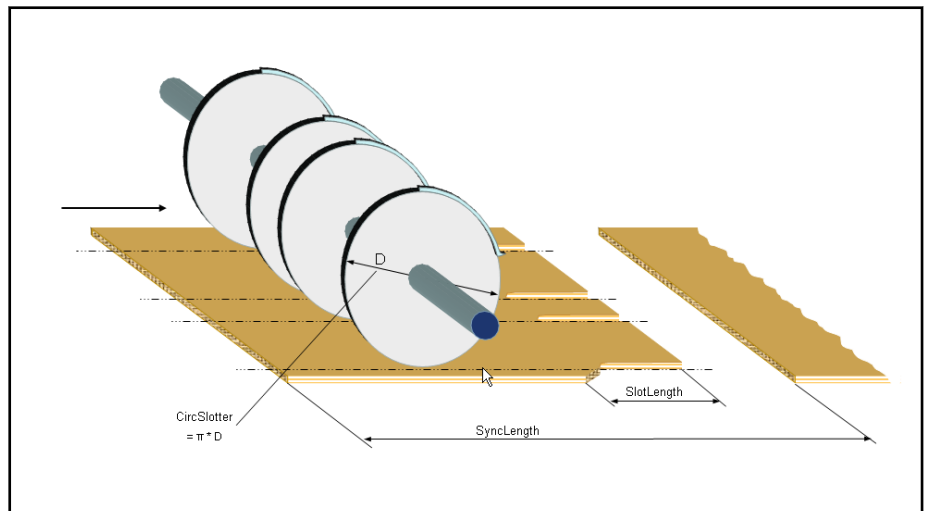


Fig. 10-23: Graphical representation of a slotter

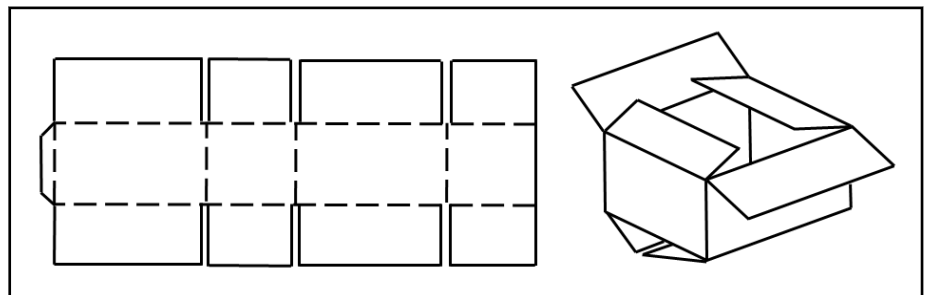


Fig. 10-24: Exemplary representation of a slotted box

Multiple slotters can be positioned in a processing line to increase the number of slits per sheet or to make slits longer than the knife of one of the slotter

RMB_TechCam.library

rolls. Typical are single slotters (slotter roll with one knife) and double slotters (slotter roll with two knives)

Definitions and specifications required to calculate slotter cams when using the function blocks MB_SlotterCalcType01 and MB_SlotterCalcType02 are described in the following.

All definitions and references described assume that the shaft equipped with knives is positioned above the conveying level of the box.

Direction of rotation, direction of motion and zero marks

The correct differentiation between direction of rotation and direction of motion is essential for the definition and the parameterization.

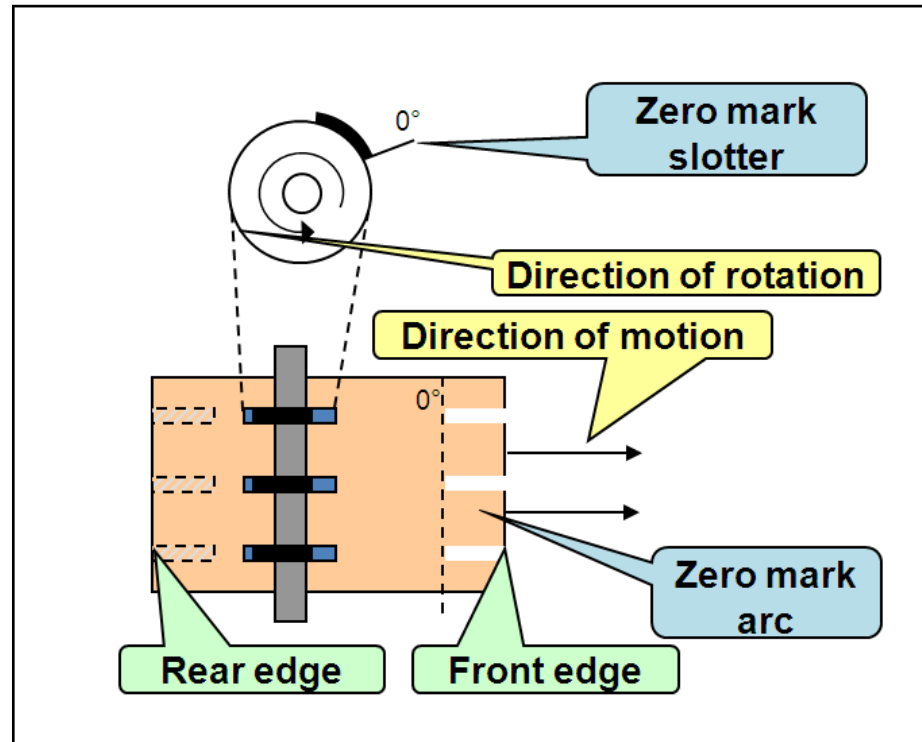


Fig. 10-25: Distinguishing direction of rotation and direction of motion

Direction of rotation

The direction of rotation refers to the direction of motion of the slotter roll. The direction of motion always refers to the direction of motion of the sheet to be slit. In this case, it is always counted from the front edge of the sheet - the edge the slotter roll passes first - towards the rear edge of the sheet. The following figure shows an example of the correct numbering of three points on the slotter roll. Thus, point 1 is the first and point 3 the last point in the direction of rotation.

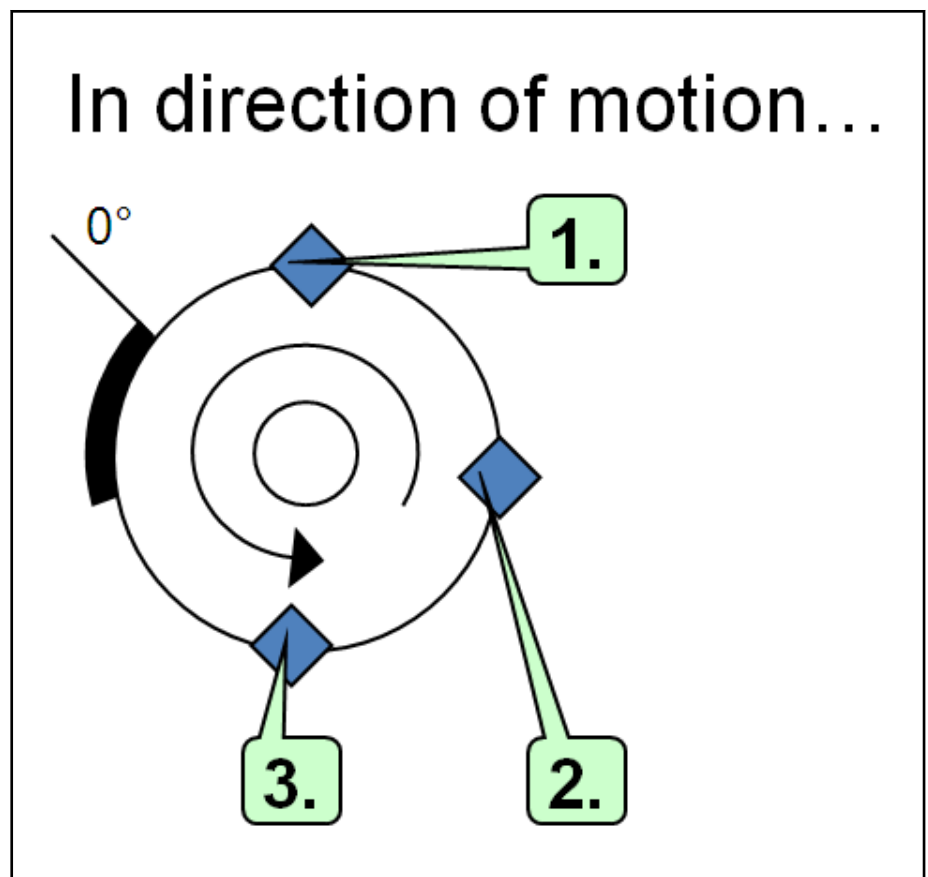


Fig. 10-26: Counting mode for specifications in the direction of rotation

Direction of motion

Furthermore, [fig. 10-25 "Distinguishing direction of rotation and direction of motion" on page 644](#) shows that the direction of motion always refers to the motion of the sheet to be slit. In this case, it is always counted from the front edge of the sheet - the edge the slotter roll passes first - towards the rear edge of the sheet. Refer to the following figure for an example on the counting mode in the direction of motion: Thus, point 1 is the first and point 3 the last point in the direction of motion.

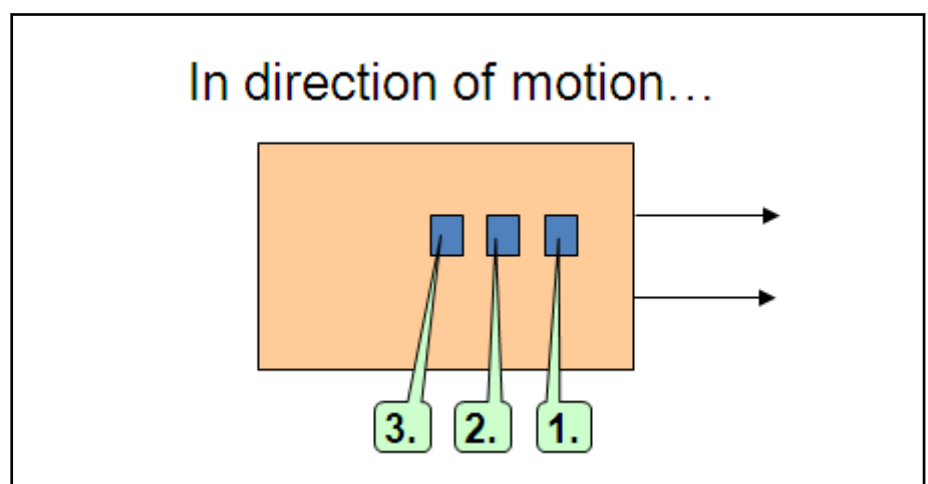


Fig. 10-27: Counting mode for specifications in the direction of motion

Slot numbers

The slotter function blocks documented here can make two slits per sheet.

RMB_TechCam.library

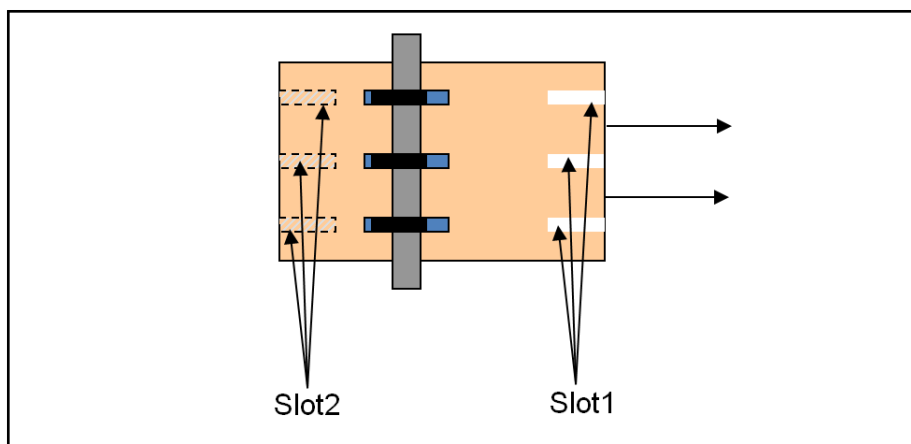
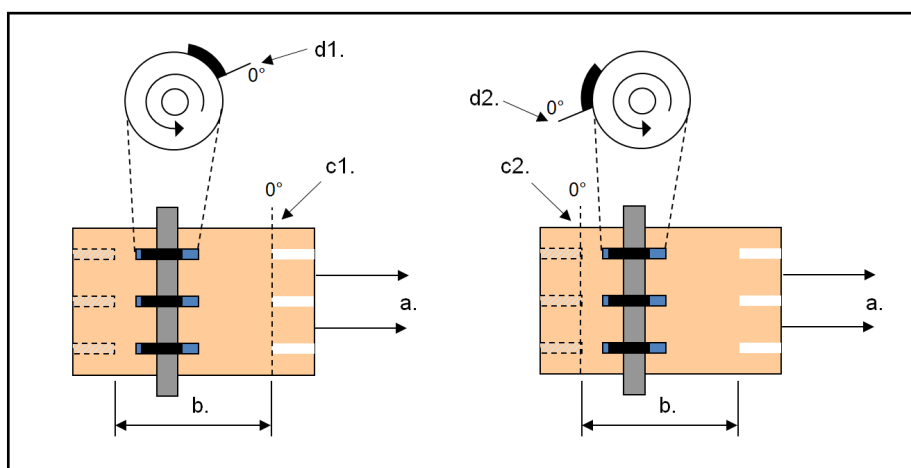


Fig. 10-28: Numbering of slots if there are multiple slits in one sheet

The first slot in the direction of motion is always referred to as "Slot 1" and the second slot as "Slot 2".

Box height and position of the zero marks

The position of the zero marks of the slotter roll defines the relation between the slotter roll and the sheet being slit. It corresponds to the 0° absolute position of the slotter axis.



- a. Direction of motion
- b. Box height ("BoxHeight")
- c1.,c2. Possible position of the zero mark of the sheet
- d1.,d2. Possible position of the zero mark of the slotter

Fig. 10-29: Determining the zero mark on the slotter roll and in the sheet to be slit

The position of the zero mark on the sheet defines an edge of the box height ("BoxHeight"). The box height is between "Slot1" and "Slot2".

On the slotter roll, the zero mark has to be always at a knife. Therefore, for slotters with one knife, there are two possible positions for the zero mark and for slotters with two knives on the cutting roll, there are four possible positions.

By definition, the zero mark on the sheet to be slit is always at the subsequent box height at Slot 1 or Slot 2. In [fig. 10-29 "Determining the zero mark on the slotter roll and in the sheet to be slit" on page 646](#), the two options for positioning the zero mark for a slotter with one knife are shown.

For the slotter function blocks, always specify the required zero mark position on the sheet for the calculation ("CamProf" input). Thus, it is defined whether the slotter roll slits "Slot1" or "Slot2".

Knife numbers The slotter function block MB_SlotterCalcType02 can be used to calculate the cams for slotters with two knives on the slotter roll. The zero mark of the slotter roll, which has to be always 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 roll from the position of the zero mark is knife 2, the other one is knife 1. Thus, by this definition, knife 1 always cuts Slot 1 and Knife 2 always cuts Slot 2.

The following figure below shows the numbering of the knives for all four possible positions of the zero mark for the cutting roll.

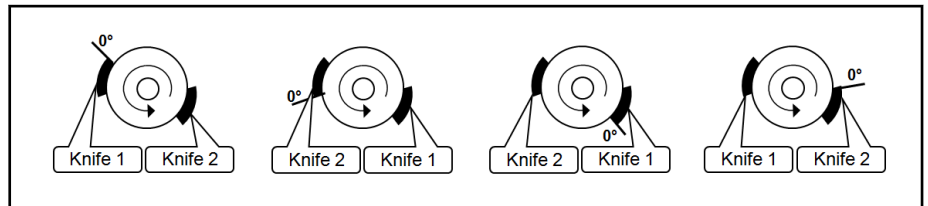


Fig. 10-30: Determining the knife numbers for slotters with two knives on the slotter roll

Knife distance The slotter function block MB_SlotterCalcType02 can be used to calculate the cams for slotters with two knives on the slotter roll. In this case, the knives do not have to be attached on the slotter roll symmetrically, but can have any distance from one another.

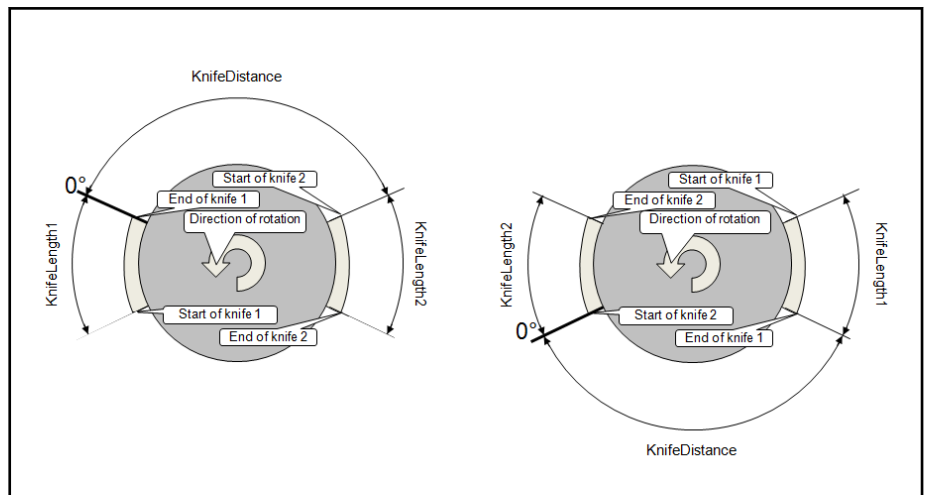


Fig. 10-31: Defining the knife distance for slotters with two knives on the slotter roll

As shown in [fig. 10-31 "Defining the knife distance for slotters with two knives on the slotter roll" on page 647](#), the knife distance is always measured from the end of knife 1 to the beginning 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.

Multi slotter If several consecutive slotters create an excessively long slot together, these slotters are called multi slotters. The following definitions and specifications applying to multi slotters are graphically shown in [fig. 10-32 "Definitions and specifications for multi slotters" on page 648](#).

In contrast to the definition option for a zero mark when using single slotters, the zero mark of the individual slotters for multi slotters is defined in a group. As shown, 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 beginning.



The function block cannot take care of incorrect entries at this point. The user is responsible to correctly parameterize the zero marks for the individual slotters.

The box height (BoxHeight) is 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.

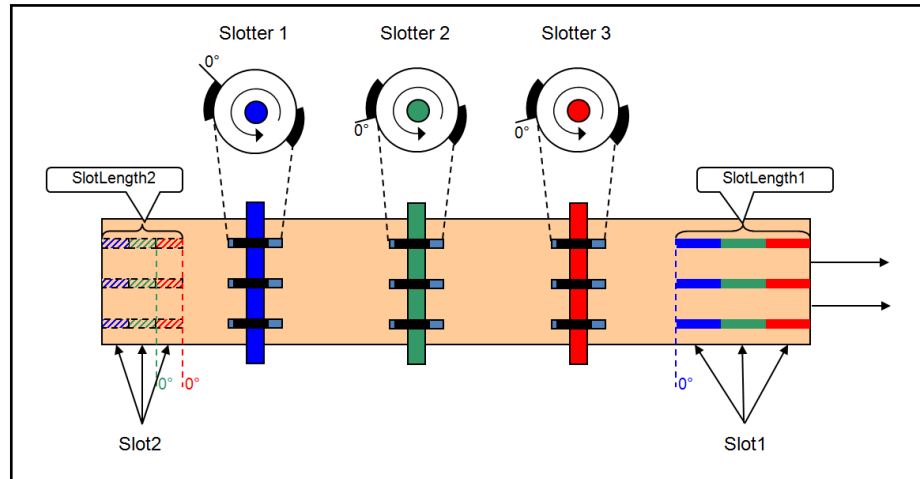


Fig. 10-32: Definitions and specifications for multi slotters

- The slotter touching the sheet first is slotter 1; the second is slotter 2 ... and the last slotter is slotter n.
- To ensure that the cams are calculated correctly, the function block with the different input variables has to be called n times, or n own instances of the function block have to be applied.
- The slots cut by the multi slotters are always specified with their complete length ("SlotLength1" or "SlotLength2"). The length of an individual partial slots (SlotLength/n) may never be applied at a function block.
- The individual slotters use "SlotLength1" and "SlotLength2" equally. Each of the n slotter cuts $1/n$ free of the SlotLength of Slot1 and Slot2.
- The knives of the individual slotters have to be of equal length. For slotters with several knives, the knives cutting at one slot have to have identical lengths.

10.9 MB_SlotterCalcType01

Brief description The function block MB_SlotterCalcType01 calculates a cam with 1024 data point elements for an individual slotter (single slotter) with one knife on the knife cylinder or for a slotter belonging to a multi slotter with one knife each on the knife cylinder.



See [Slotter: Introduction and definitions, page 643](#), for basic information on the function block described below.

The function block MB_SlotterCalcType01 has the following functionalities:

- One slot per sheet (single slotter). One slot at the beginning or end of the sheet (corresponds to MSV_CamTable14 by Rexroth SYNAX200).
- One slot per sheet in "Skip feed" mode with a sheet across several machine processes.

- One long slot at the beginning or end of the sheet (multi slotter). Two or more slotter cylinders work on one slot.
- One long slot per sheet in "Skip feed" mode with a sheet across several machine processes.
- Two slots per sheet (single slotter). One slot at the beginning of the sheet and one at the end.
- Two slots per sheet in "Skip feed" mode with a sheet across several machine processes.
- Two long slots per sheet (multi slotter). One long slot at the beginning of the sheet and one long slot at the end.
- Two long slots per sheet in "Skip feed" mode with a sheet across several machine processes.
- Slotting with slotter discs whose circumference (cutting circle/"CircSlotter") is smaller than the machine process ("SyncLength").

The function block always calculates one cam for one slotter. With n slotters in a multi slotter, the function block has to be called n times or n separate instances of the function block have to be generated. Define via the function block input "SlotterToCalculate" for which slotter the cam is to be calculated.

Assignment: Target system/library

IndraMotion MLC 13VRS	RMB_TechCam

Tab. 10-17: Reference table of MB_SlotterCalcType01

Interface description

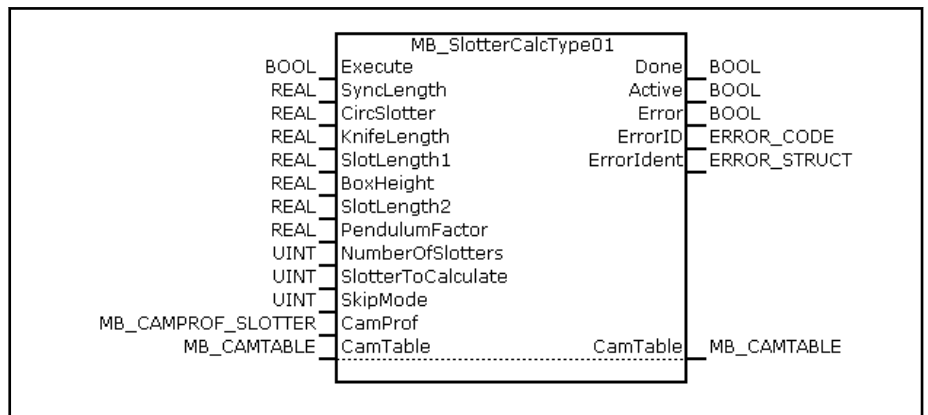


Fig. 10-33: Interface of the function block: MB_SlotterCalcType01

	Name	Type	Comment
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements
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 cutting circle [u]*
	KnifeLength	REAL	Knife length [u]*
	SlotLength1	REAL	Entire length of the first slot in the sheet in the direction of motion [u]*

RMB_TechCam.library

	Name	Type	Comment
	BoxHeight	REAL	Distance between the two slots in the sheet ("SlotLength1" + "BoxHeight" + "SlotLength2" = sheet length) [u]*
	SlotLength2	REAL	Entire 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 area) 0%: No pendular motion (standstill at large format length) 100%: Maximum pendular motion (maximum reverse rotation to the point where the knife cuts the cardboard backwards)
	NumberOfSlotters	UINT	Number of slotters in a multi slotter that cut the two slots at the beginning and end of the sheet together [-]**
	SlotterToCalculate	UINT	Number of the slotter the cam is to be calculated for when the function block is called [-]**
	SkipMode	UINT	Number of machine cycles per sheet -1 [-]**
	CamProf	MB_CAMPROF_SLOTTER	Cam profile selection and position of the zero mark of the slotter cylinder MB_CAMPROF_SLOTTER, page 671
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 659

* The unit [u] means "unit". This can be any unit such as [mm] or [inch]. Ensure that the selected unit is always used.

** The unit [-] refers to a variable without a unit.

Tab. 10-18: Interface variables of the MB_SlotterCalcType01 function block

Max./min. 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"

Name	Type	Min. value	Max. value	Default value	Effective
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"
SlotterToCalcu- late	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 MB_SlotterCalcType01 function block

RMB_TechCam.library

Signal-time diagram

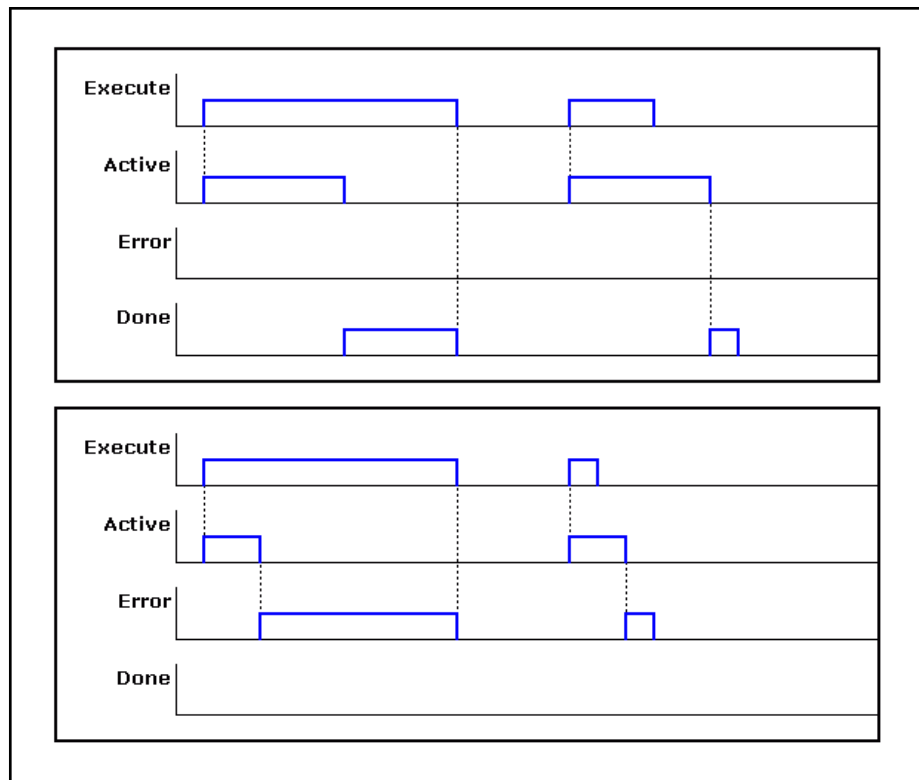


Fig. 10-34: Signal time diagram of the MB_SlotterCalcType01 function block

Cam characteristics

The cam table of the "MB_SlotterCalcType01" function block has the following characteristics:

- Load the cam profile for cyclic processing as cam parameter to the drive (P-0-0072, P-0-0092, P-0-0780 and P-0-0781) or to 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 can generate profiles both for indefinitely turning cams without gear reduction (P-0-0086 Bit4 = 0) and for superimposed profiles with gear reduction (P-0-0086 Bit4 = 1).
- The zero mark of the slotter cylinder can be positioned at the beginning and at the end of the knife in the direction of rotation. Also refer to [Box height and position of the zero marks, page 646](#).
- The number of calculation cycles of an instance of MB_SlotterCalcType01 is automatically adjusted to the current system load in order to avoid any significant influence on the PLC cycle time.
- With excessively long synchronous areas, the compensation motion can lead to a reverse rotation. The maximum reverse rotation of the slotting cutter can be specified.
- With cams cutting two slots per sheet, the parameterization of an electronic gear has to ensure that both revolutions are contained in one cam created by the function block. The function block calculates the synchronous areas in such a way that the slotter cylinder following the electronic gear revolves with material velocity during synchronous areas.
- In SkipFeed mode the product is assumed to span several machine cycles ("synchronous format lengths"). SkipMode n extends one operation rotation to n+1 machine cycles. Ensure that the cam created by the

function block is extended to the respective number of machine cycles by using an electronic gear. In SkipFeed mode, the function block calculates the synchronous areas in such a way that the slotter cylinder following the electronic gear revolves with material velocity during synchronous areas.

Properties in a multi slotter

Observe the following additional issues when operating multi slotters:

- Multi slotters cutting a long slot at the beginning or end of the sheet can be implemented with a joint cam ("NumberOfSlotters"=1) and offset zero marks on the sheet.
- The first slotter that the sheet reaches is slotter 1, the last is slotter n.
- To optimally use the slotters dynamically, slotter 1 always cuts a window into the sheet for the creation of slot 1. Slotters 2 to n then cut the slot. On the other side of the sheet (slot 2) slotter 1 cuts the area directly at the sheet edge and slotters 2 to n then increase the slot until it reaches "SlotLength2".
- The knives of the slotters have to have the same length.
- The individual slotters equally divide "SlotLength1" and "SlotLength2". Each slotter cuts 1/n of SlotLength of Slot1 and Slot2.
- The box height (BoxHeight) is situated 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 of the "SlotLength2" in the direction of the rear sheet edge.
- The entire length of slots 1 and 2 has to be specified at the inputs "SlotLength1" and "SlotLength2", not the length of the partial slot that the slotter to be calculated cuts.
- Define at input "NumberOfSlotters" how many slotters belong to the multi slotter and cut the two slots together.
- Define at input "SlotterToCalculate" for which of the "NumberOfSlotters" slotters the cam is to be calculated. The numbering of the slotters is carried out in the order they touch the sheet (slotter 1 touches the sheet first, slotter 2 second, etc.).
- As slotter 1 cuts a window into the material at Slot1, it has to be synchronous to the material velocity on its entire knife length. All remaining slotters are only synchronous on their respective partial slot length. For this reason, slotters 2 - n may be dynamically more favorable than slotter 1.

Functional description

It is required that the circumferential velocity of the knife cylinder remains synchronous to the velocity of the web in a synchronous area determined by the cutting process (determined by "SlotLength1" and "SlotLength2"). For this purpose, the function block uses "Execute" to once calculate a cam table with 1024 data point elements on the basis of the input variables after release for processing.

The figure below shows the meaning of the input variables.

RMB_TechCam.library

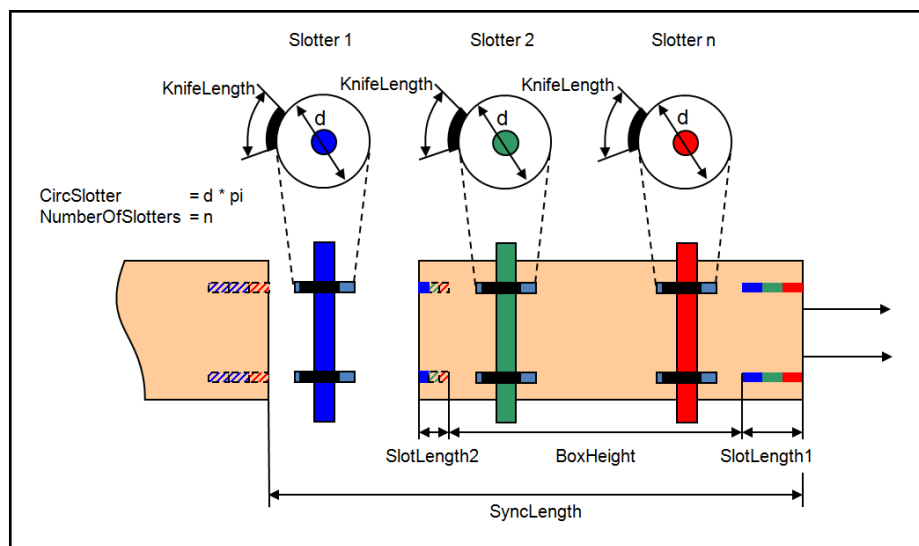


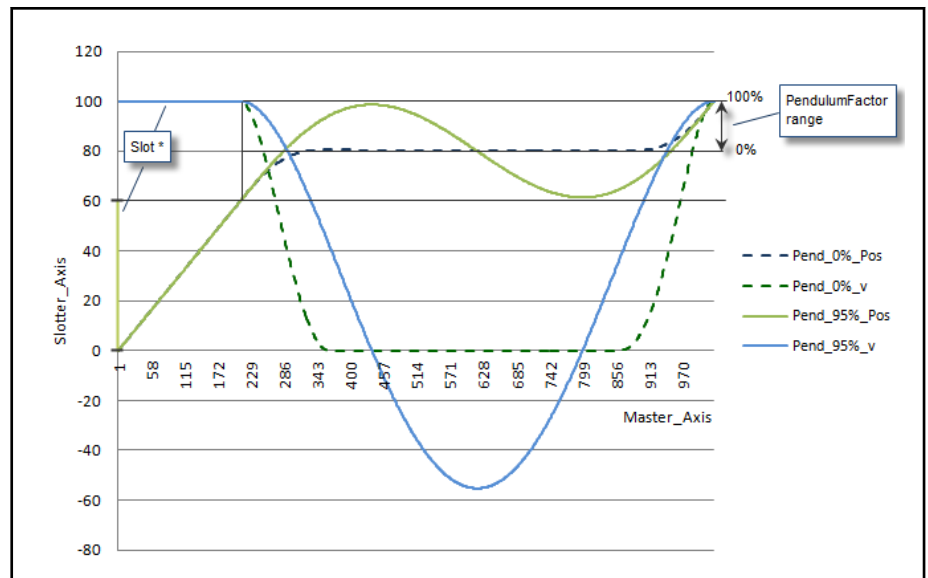
Fig. 10-35: Meaning of the input variables of the function block MB_SlotterCalc-Type01

In order to be able to cut formats which are longer or shorter than the circumference of the knife cylinder synchronously with only one cutting cylinder, 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 proportions between the cardboard lengths and slot lengths and the slotter disc circumferences are disadvantageous, there may be reverse rotations in the compensation motions. Those are desired as the dynamic strain of the motor due to peak accelerations becomes more favorable.

In extreme cases the knife may enter the cardboard contrary to the transport direction during the reverse rotation. To avoid this, the input "PendulumFactor" is provided. Use it to limit the maximum reverse rotation ($0\% < \text{"PendulumFactor"} < 100\%$) or to repress it completely ($0\% = \text{"PendulumFactor"}$). The compensation motion between the synchronous areas corresponds to $\pm 100\%$ ("PendulumFactor")



Pend_0%_Pos Cam table for "PendulumFactor" = 0%

Pend_0%_v Velocity for "PendulumFactor" = 0%

Pend_95%_Pos Cam table for "PendulumFactor" = 95%

Pend_95%_v Velocity for "PendulumFactor" = 95%

Slot* Synchronization speed during material entry

Fig. 10-36: Description of the input "PendulumFactor" using cam tables

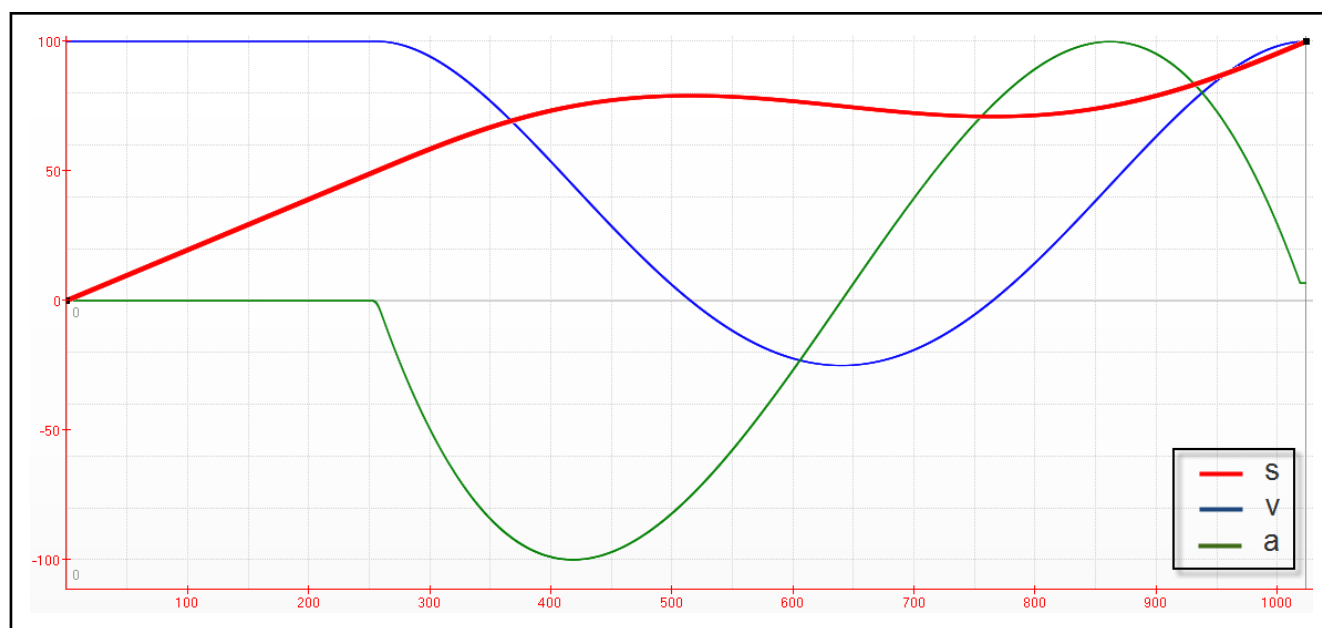
Function variants

Using various preassignments of the inputs, various function variants are possible for the function block MB_SlotterCalcType01:

Example 1

The first variant [MotionProfile 1 example, page 656](#), is suitable to cut formats with a slotter 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.

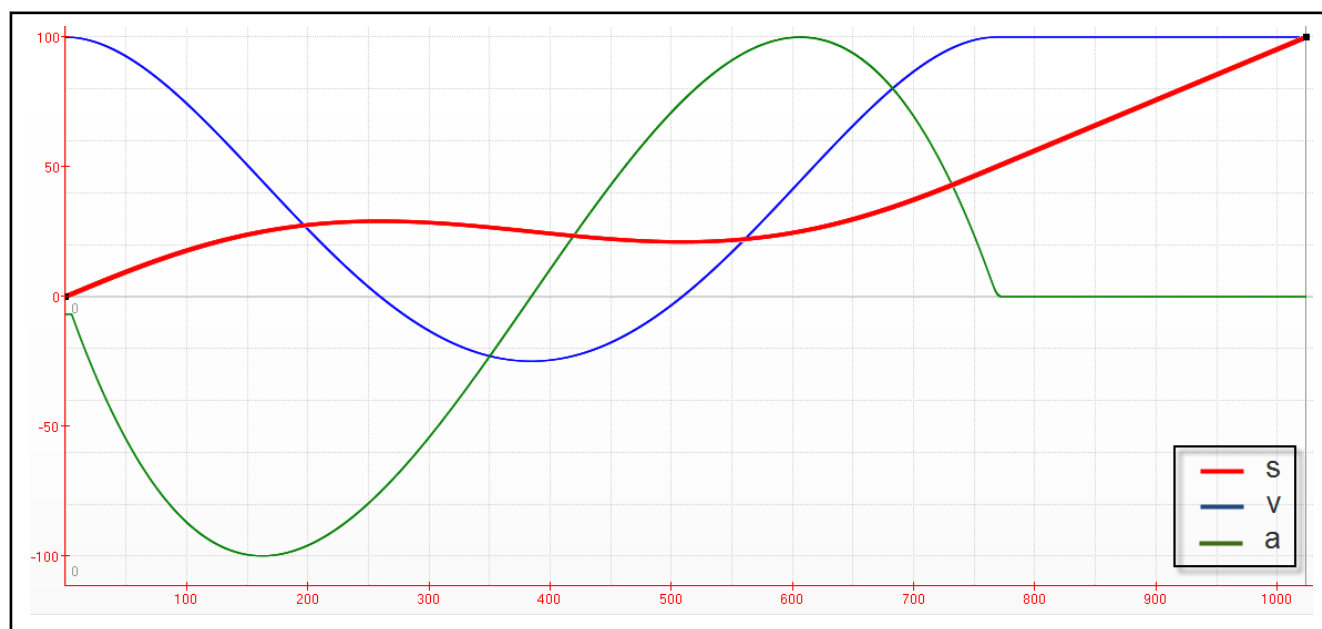
RMB_TechCam.library



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: MotionProfile 1 example

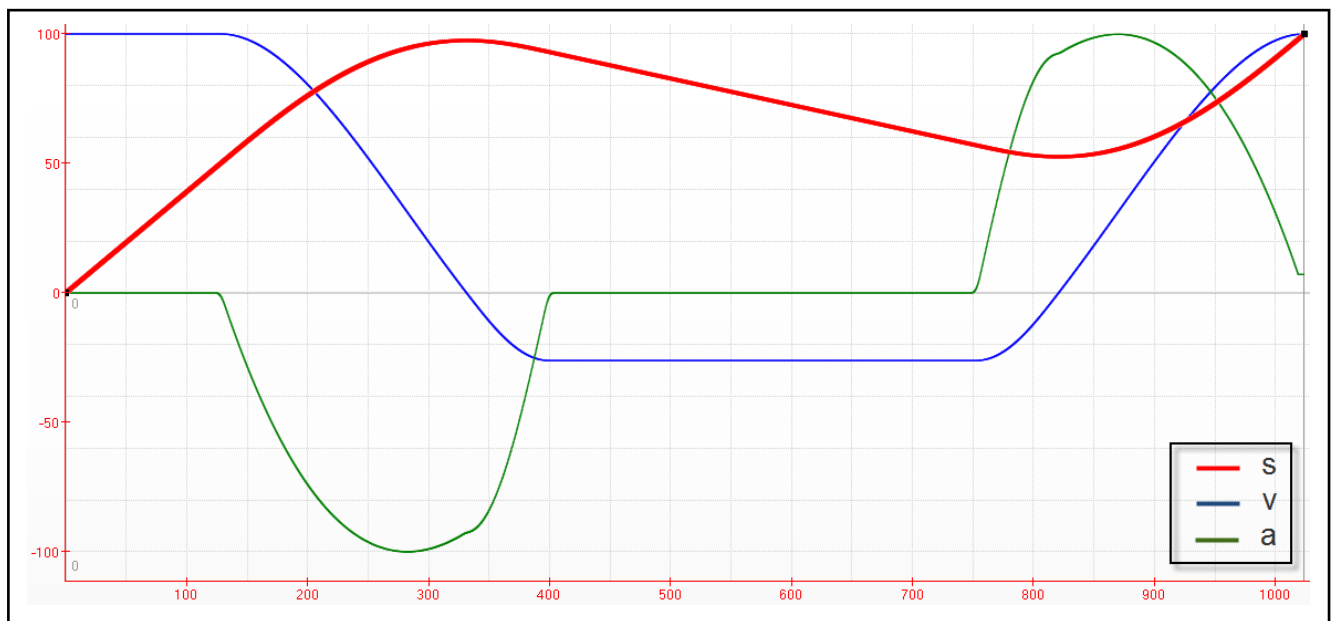
Example 2 In all single slotter variants "NumberOfSlotters" = 1 the zero mark of the slotter cylinder can be defined by specifying the input "CamProf" in front of or behind the knife. In the following variant [MotionProfile 2 example, page 656](#), the first variant [MotionProfile 1 example, page 656](#), was used again and with the input "CamProf" a compensation motion at the beginning and the synchronous motion at the end was selected.



Example 2: Zero mark behind the knife, compensation motion at the beginning (SyncLength = 1024; CircSlotter = 512; SlotLength1 = 256; SkipMode = 0; CamProf = 2)

Fig. 10-38: MotionProfile 2 example

Example 3 The third variant [MotionProfile 3 example, page 657](#), is suitable for processing excessively long formats, i.e. formats which are larger than the maximum synchronous format length (typically used in "SkipMode"). The slotting process is extended to several machine cycles, with the cutting area of the knife remaining 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 knife length ("SlotLength1") travels synchronously to the material, a compensation motion is carried out in the remaining motion until the n+1 cycle is completed. The following additionally shows the pendulum mode of operation. By specifying "PendulumFactor" = 90 the reverse rotation in the compensation area is limited 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; SlotLength1 = 256; PendulumFactor = 90; SkipMode = 1; CamProf = 0)

Fig. 10-39: MotionProfile 3 example

Example 4 The fourth variant [MotionProfile 4 example, page 658](#), shows a cam that creates a slot at the beginning and at the end of the sheet. If cams cut two slots per sheet, the slotter cylinder has to complete two revolutions while the sheet passes through it. To accomplish two revolutions per master axis revolution, both revolutions have to be contained in this one cam generated by the function block.

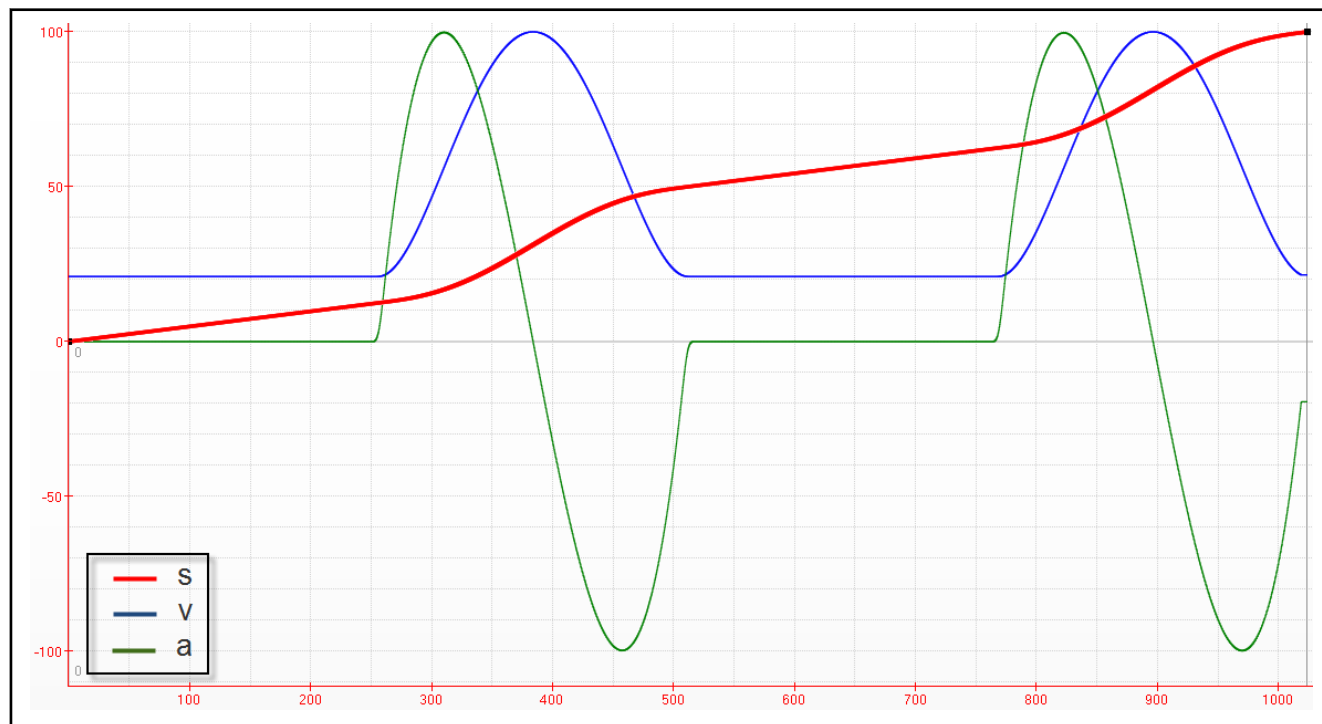


For format lengths that almost correspond to synchronous format, slot 2 of the first sheet and slot 1 of the following sheet are positioned directly behind each other. The slotter cylinder has to make an almost complete revolution in a very short time for the knife to enter the second sheet at the right time. This places high demands on the acceleration capability of the drives and motors used and is disadvantageous from a dynamic point of view.

Under these circumstances we do not recommend operating the function block with "SkipFeed" = 0.

RMB_TechCam.library

The cam is determined by two synchronous areas (determined by "SlotLength1" and "SlotLength2") and two compensation motions between the synchronous areas. During the synchronous areas the knife on the slotter cylinder is in the material and cuts the two slots.

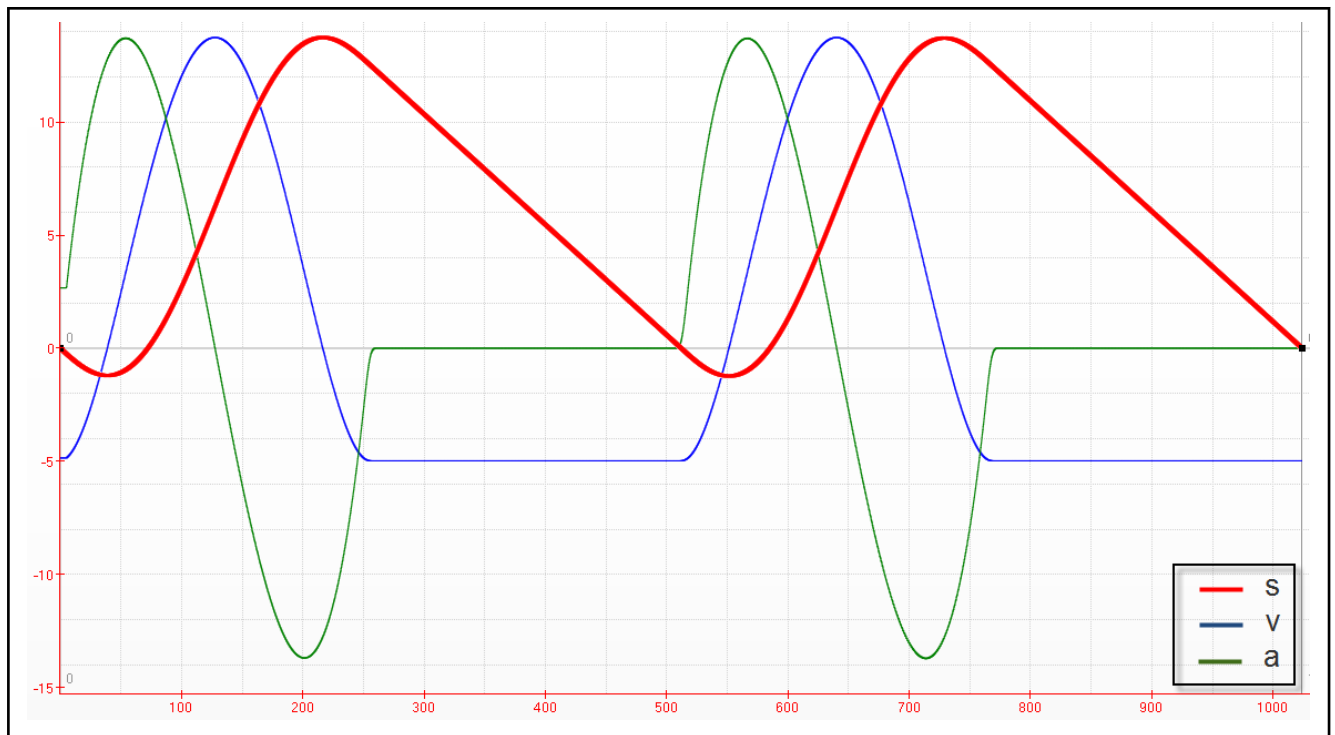


Example 4: Zero mark in front of the knife (SyncLength = 1024; CircSlotter = 1024; KnifeLength = 256; BoxHeight = 256; SlotLength1 = SlotLength2 = 256; SkipMode = 0; CamProf = 0)

Fig. 10-40: MotionProfile 4 example

Example 5

Based on the previous example 4 [MotionProfile 4 example, page 658](#), example 5 [MotionProfile 5 example, page 659](#), represents an additive curve. However, this time a superimposed profile with gear reduction and a compensation motion at the beginning and a synchronous motion at the end was selected via the input "CamProf".



Example 5: Zero mark behind the knife (SyncLength = 1024; CircSlotter = 1024; KnifeLength = 256; BoxHeight = 256; SlotLength1 = SlotLength2 = 256; SkipMode = 0; CamProf = 3)

Fig. 10-41: MotionProfile 5 example

The cam table to be calculated is saved in the structure "CamTable" if the function block was processed without errors. 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 F_RELATED_TABLE, 16#0170 error 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
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 less than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0002	"CircSlotter" is less than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0003	"KnifeLength" is less than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0004	"BoxHeight" is less than the minimum value

RMB_TechCam.library

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0005	"SlotLength1" is less than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0006	"SlotLength2" is less than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0007	"PendulumFactor" is less than the minimum or greater than the maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0008	"SkipMode" is less than the minimum or greater than the maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0009	"CamProf" is less than the minimum or greater than the maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0010	"NumberOfSlotters" is less than the minimum or greater than the maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0011	"SlotterToCalculate" is less than the minimum or greater than "NumberOfSlotters"
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0012	"CamTable.NumberOfElements" does not equal 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	The value of "CamProf" is invalid

Tab. 10-20: Error numbers caused by MB_SlotterCalcType01

10.10 MB_SlotterCalcType02

Brief description

The function block MB_SlotterCalcType02 calculates a cam with 1024 data point elements for an individual slotter (single knife slotter) with two knives on the knife cylinder or for a slotter belonging to a multi slotter with two knives each on the knife cylinder.

The function block MB_SlotterCalcType02 has the following functions:

- Two slots per sheet (single knife slotter). One slot at the beginning of the sheet and one at the end.
- Two slots per sheet in "Skip feed" mode with a sheet across several machine processes.
- Two long slots per sheet (multi slotter). One long slot at the beginning of the sheet and one long slot at the end.
- Two long slots per sheet in "Skip feed" mode with a sheet across several machine processes.

The function block always calculates only one cam for one slotter. For n slotters in a multi slotter, the function block has to be called n times. Define via

the function block input "SlotterToCalculate" for which slotter the cam is to be calculated.

Assignment: Target system/library

IndraMotion MLC 13VRS

RMB_TechCam

Tab. 10-21: Reference table of MB_SlotterCalcType02

Interface description of Type02

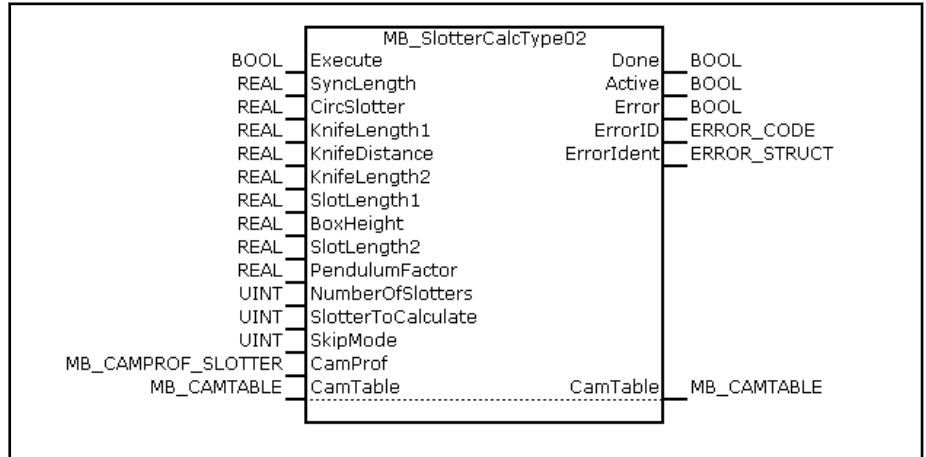


Fig. 10-42: Interface of the function block: MB_SlotterCalcType02

	Name	Type	Comment
VAR_IN_OUT	CamTable	MB_CAMTABLE	Structure with the calculated cam table elements and information on the number of elements
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 cutting circle [u]*
	KnifeLength1	REAL	Length of the first knife [u]*
	KnifeDistance	REAL	Distance between the end of knife 1 and the beginning of knife 2 [u]*
	KnifeLength2	REAL	Length of the second knife [u]*
	SlotLength1	REAL	Entire 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	Entire 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 with a large format length)

RMB_TechCam.library

	Name	Type	Comment
	NumberOfSlotters	UINT	Number of slotters in a multi slotter that cut the two slots at the beginning and end of the sheet together [-]
	SlotterToCalculate	UINT	Number of the slotter the cam is to be calculated for when the function block is called [-]
	SkipMode	UINT	Number of machine cycles per sheet -1 [-]
	CamProf	MB_CAMPROF_SLOTTER	Cam profile selection and position of the zero mark of the slotter cylinder MB_CAMPROF_SLOTTER, page 671
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 Error handling, page 670
	ErrorIdent	ERROR_STRUCT	If the "Error" output is set, the output contains detailed error information

* The unit [u] means "unit". This can be any unit such as [mm] or [inch]. Ensure that the selected unit is always used.

Tab. 10-22: Interface variables of the MB_SlotterCalcType02 function block

Max./min. 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 Slot-Length2 > 0

Name	Type	Min. value	Max. value	Default value	Effective
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 MB_SlotterCalcType02 function block

Signal-time diagram

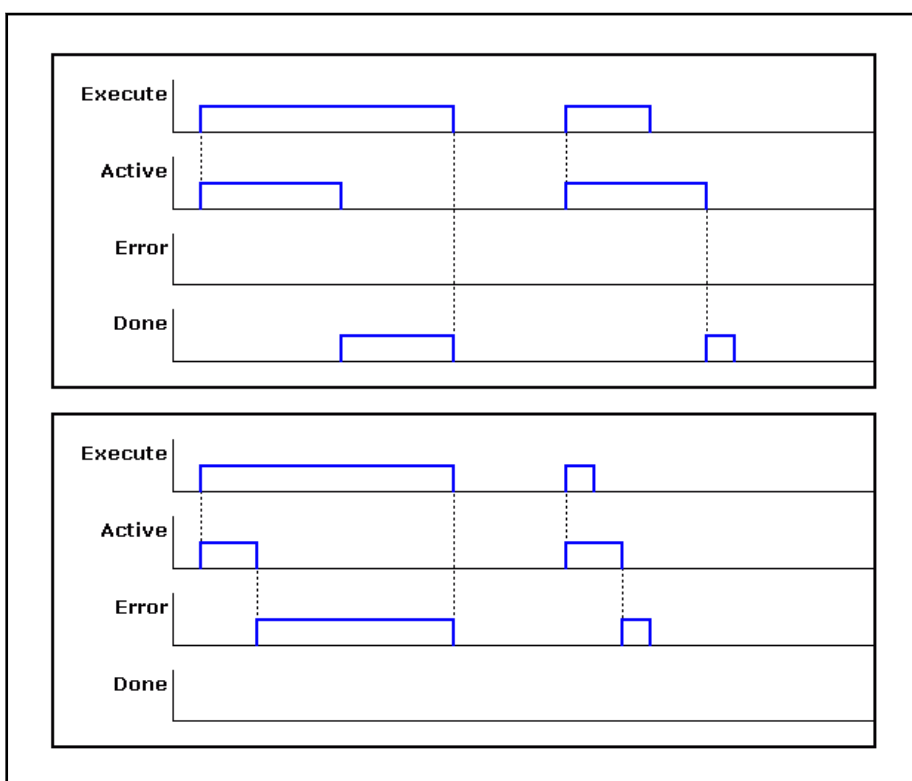


Fig. 10-43: Signal time diagram of the MB_SlotterCalcType02 function block

Cam characteristics

The cam table of the "MB_SlotterCalcType02" function block has the following characteristics:

- Load the cam profile for cyclic processing as cam parameter to the drive (P-0-0072, P-0-0092, P-0-0780 and P-0-0781) or to 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 can generate profiles both for indefinitely turning cams without gear reduction (P-0-0086 Bit4 = 0) and for superimposed profiles with gear reduction (P-0-0086 Bit4 = 1).

RMB_TechCam.library

- The zero mark of the slotter cylinder can be positioned at the beginning and at the end of the knife in the direction of rotation.
- 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 of the slotting cutter can be specified.
- In SkipFeed mode the product is assumed to span several machine cycles ("synchronous format lengths"). SkipMode n extends one operation rotation to n+1 machine cycles. Ensure that the cam created by the function block is extended to the respective number of machine cycles by using an electronic gear.

Properties in a multi slotter

Observe the following additional issues when operating multi slotters:

- The first slotter that the sheet reaches is slotter 1, the last is slotter n.
- Slotter 1 always starts the creation of slot 1 by cutting a window into the sheet. Slotters 2 to n then cut the slot. On the other side of the sheet (slot 2) slotter 1 cuts the area directly at the sheet edge and slotters 2 to n then increase the slot until it reaches "SlotLength2".
- The knives of the slotters have to have the same length.
- The individual slotters equally divide "SlotLength1" and "SlotLength2". Each slotter cuts 1/n of SlotLength of Slot1 and Slot2.
- The box height (BoxHeight) is situated 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 of the "SlotLength2" in the direction of the rear sheet edge.
- The entire length of slots 1 and 2 has to be specified at the inputs "SlotLength1" and "SlotLength2", not the length of the partial slot that the slotter to be calculated cuts.
- Define at input "NumberOfSlotters" how many slotters belong to the multi slotter and cut the two slots together.
- Define at input "SlotterToCalculate" for which of the "NumberOfSlotters" slotters the cam is to be calculated. The numbering of the slotters is carried out in the order they touch the sheet (slotter 1 touches the sheet first, slotter 2 second, etc.).
- As slotter 1 cuts a window into the material at Slot1, it has to be synchronous to the material velocity on its entire knife length. All remaining slotters are only synchronous on their respective partial slot length. For this reason, slotters 2 - n may be dynamically more favorable than slotter 1.

Functional description

The circumferential velocity of the knife cylinder has to remain synchronous to the web within a certain synchronous area determined by the cutting process (determined by "SlotLength1" and "SlotLength2"). On the basis of this requirement, the function block uses "Execute" to once calculate a cam table with 1024 data point elements on the basis of the input variables after release for processing.

The figure below shows the meaning of the input variables.

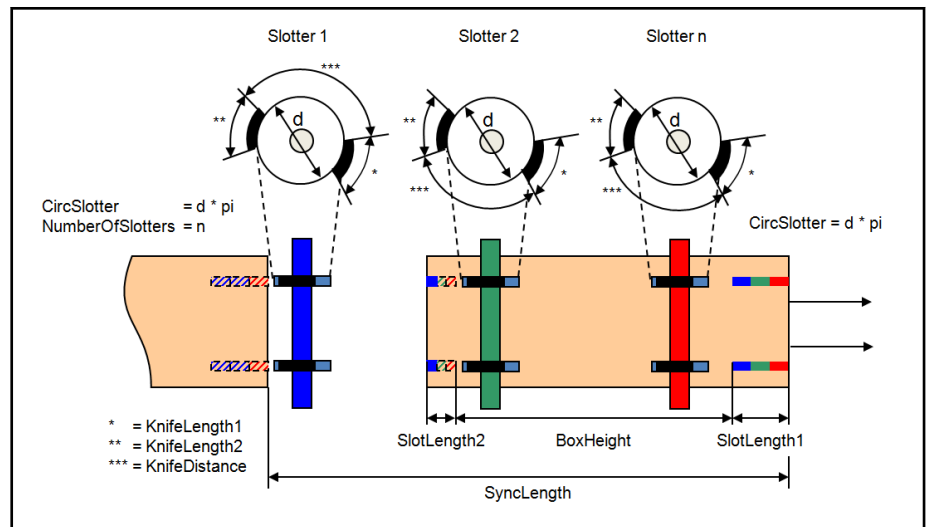


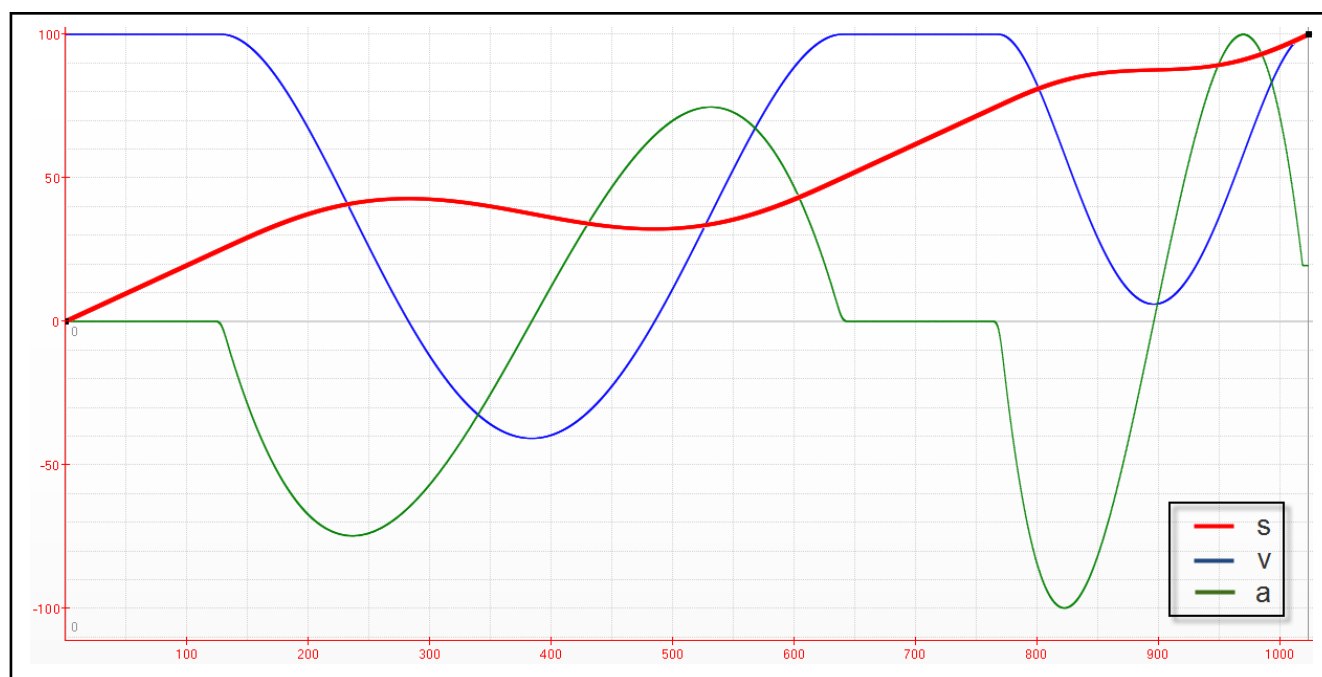
Fig. 10-44: Meaning of the input variables of the function block MB_SlotterCalcType02

Corresponding to the two synchronous areas specified, two compensation motions calculated by the function block have to be carried out. These compensation motions are characterized by a continuity up to the acceleration.

Using various preassignments of the inputs, various function variants are possible for the function block MB_SlotterCalcType02:

The first variant [MotionProfile 1 example, page 666](#), is suitable to cut formats with a slotter whose cutting circle "CircSlotter" is smaller than the mechanical processing length of the machine "SyncLength". The effective knife length is adjusted to the material transport velocity in the tool gearing. The remaining disc circumference of the slotter without cutting knife is defined as the area for the compensation motions.

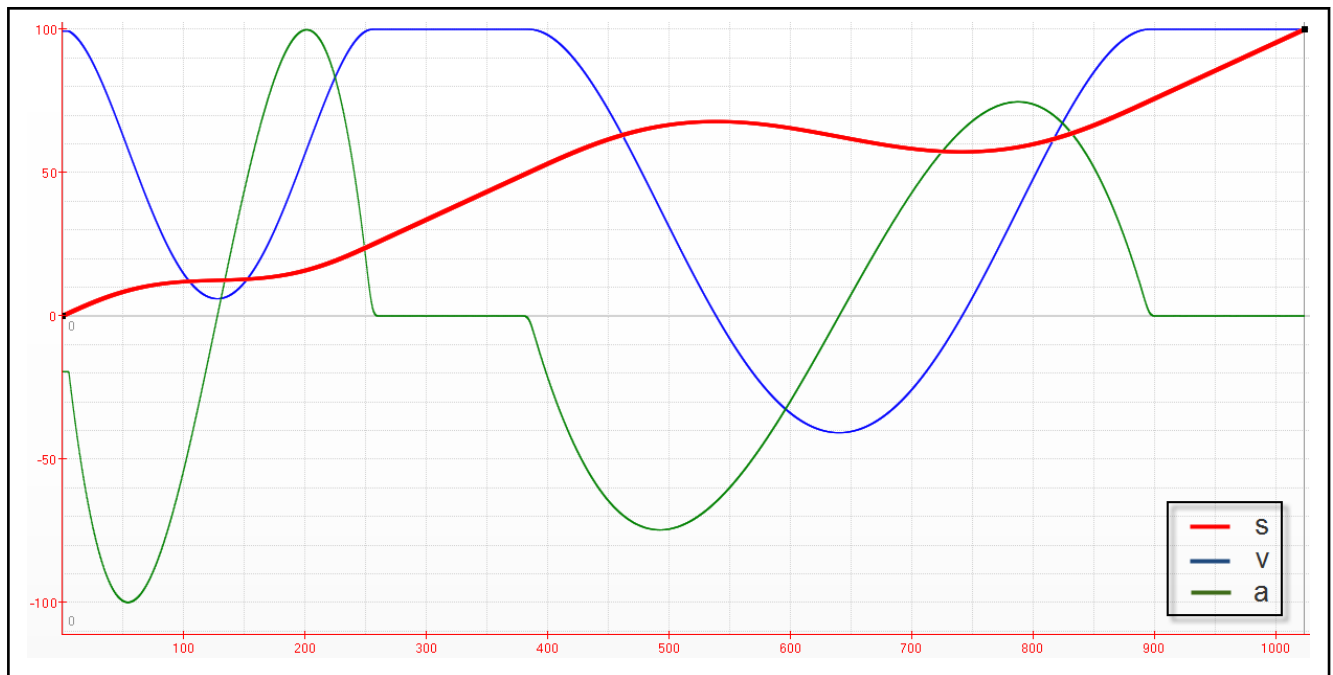
RMB_TechCam.library



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: MotionProfile 1 example

In all single knife slotter variants "NumberOfSlotters" = 1 the zero mark of the slotter cylinder can be defined by specifying the input "CamProf" at the end of the first knife or at the beginning of the second knife. In the following variant [MotionProfile 2 example, page 667](#), the first variant [MotionProfile 1 example, page 666](#), was used again and with the input "CamProf" a compensation motion at the beginning and the synchronous motion at the end was selected.

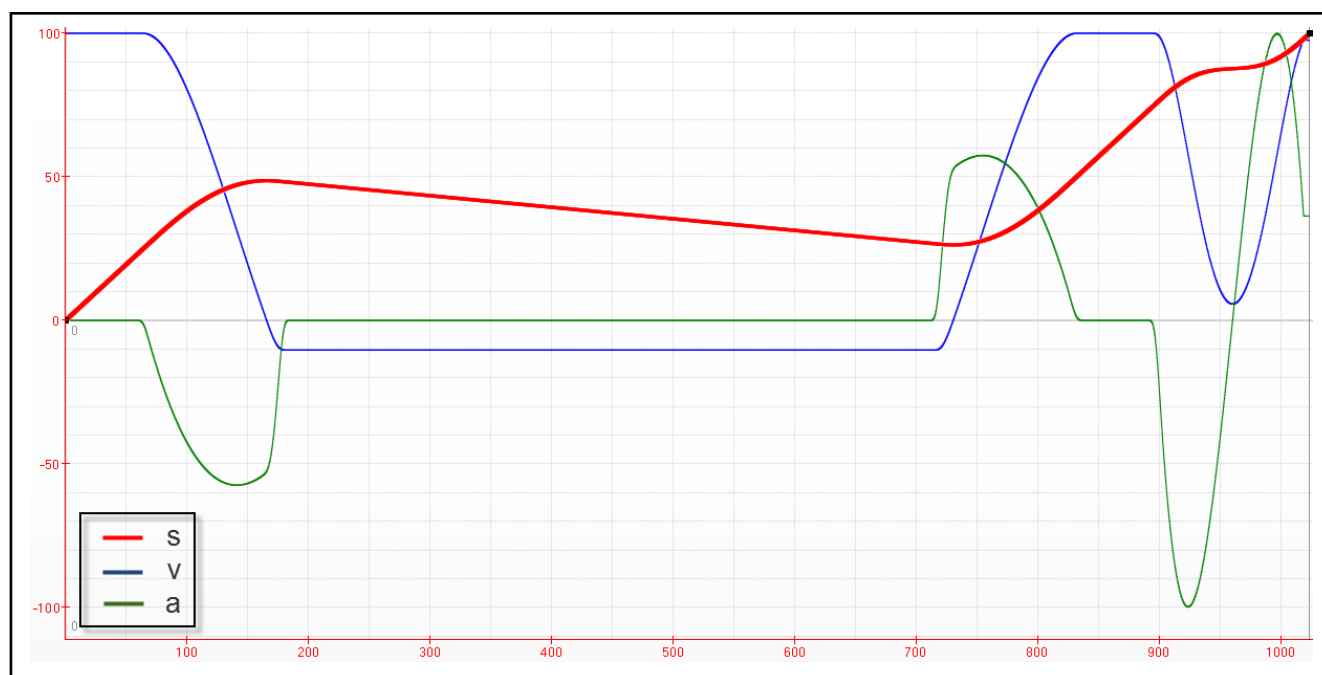


Example 2: Zero mark behind the knife, compensation motion at the beginning (SyncLength = 1024; CircSlotter = 512; SlotLength1 = SlotLength2 = 128; BoxHeight = 256; KnifeLength1 = KnifeLength2 = KnifeDistance = 128; SkipMode = 0; CamProf = 2)

Fig. 10-46: MotionProfile 2 example

The third variant [MotionProfile 3 example, page 668](#), is suitable for processing excessively long formats, i.e. formats which are larger than the maximum synchronous format length. The slotting process is extended to several machine cycles, with the cutting areas of the knives remaining 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 knife lengths travel synchronously to the material, a compensation motion is carried out in the remaining motion until the n+1 cycle is completed. The following additionally shows the pendulum mode of operation. By specifying "PendulumFactor" = 90, the reverse rotation in the compensation area is limited to $37.5\% + 12.5\% \cdot 0.9 = 48.75\%$.

RMB_TechCam.library



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)

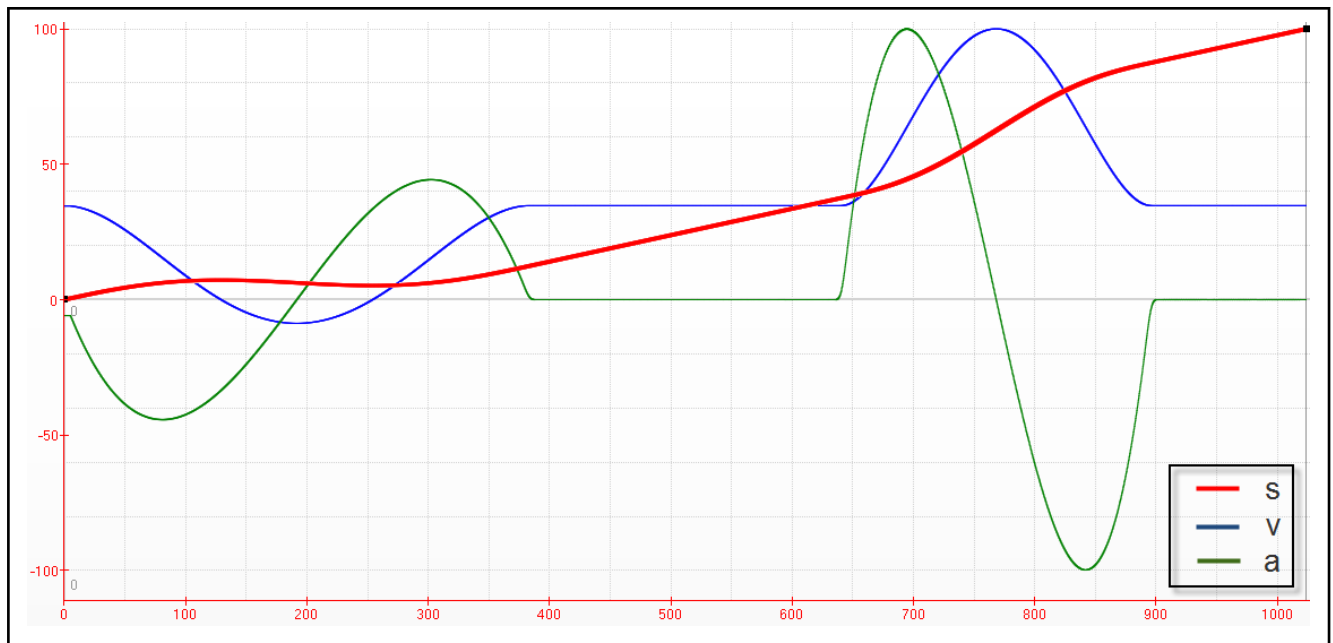
Fig. 10-47: MotionProfile 3 example

The fourth variant shows cams that create a long slot at the beginning and at the end of the sheet (multi slotter).



Contrary to the possibility to define the zero mark individually in single knife slotters, the zero marks of the individual slotters are determined according to [Definitions and specifications for multi slotters, page 648](#), for multi slotters. The zero mark of slotter 1 (NumberOfSlotter = 1) is always at the end of the knife in the direction of rotation (CamProf = 2, 3), and the zero marks of slotters 2 to n (NumberOfSlotter = 2,...,n) are at the beginning of the knife (CamProf = 0, 1).

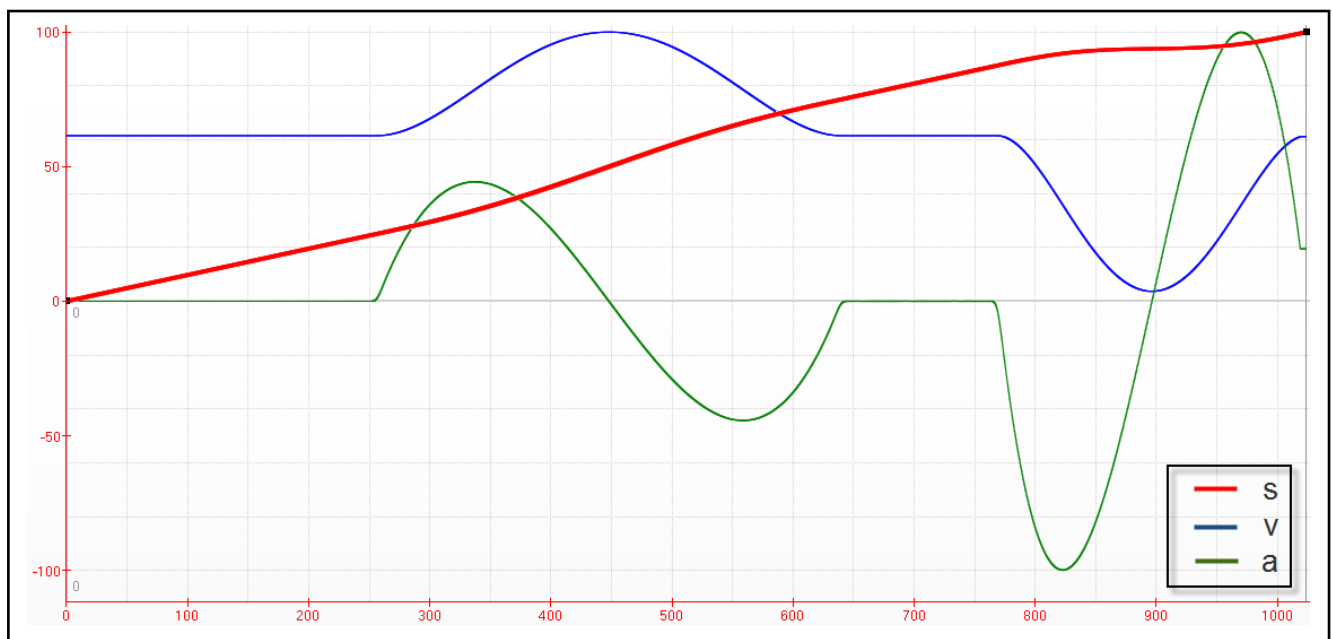
According to the position of the zero mark, the cam of the first slotter begins with a compensation motion and ends with a synchronous area.



Example 4: First slotter in a multi slotter (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: MotionProfile 4 example

According to the position of the zero mark, the cam of the second slotter begins with a synchronous area and ends with a compensation motion.



Example 5: Second slotter in a multi slotter (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: MotionProfile 5 example

RMB_TechCam.library

The cam table to be calculated is saved in the structure "CamTable" if the function block was processed without errors. 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#170. It can generate the following error messages in Additional1 and 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 less than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0002	"CircSlotter" is less than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0004	"BoxHeight" is less than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0005	"SlotLength1" is less than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0006	"SlotLength2" is less than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0007	"PendulumFactor" is less than the minimum or greater than the maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0008	"SkipMode" is less than the minimum or greater than the maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0009	"CamProf" is less than the minimum or greater than the maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0010	"NumberOfSlotters" is less than the minimum or greater than the maximum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0011	"SlotterToCalculate" is less than the minimum or greater than "NumberOfSlotters"
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0012	"CamTable.NumberOfElements" does not equal 1024
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0013	"KnifeLength1" is less than the minimum
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0014	"KnifeDistance" is less than the minimum value
INPUT_RANGE_ERROR, 16#0006	16#2250	16#0015	"KnifeLength2" is less than the minimum
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0004	"SlotLength1" + "BoxHeight" + "SlotLength2" is greater than "SyncLength" * ("SkipMode" + 1)

ErrorID	Additional1	Additional2	Description
INPUT_INVALID_ERROR, 16#0001	16#2251	16#0005	The value of "CamProf" is invalid
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 caused by 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, the RIL_ParameterChannel.library allows acyclic communication via the cyclic channel of a field bus connection.

11.2 IL_ParameterChannel

Brief description The function block allows an acyclic communication via the cyclic data channel. The data is exchanged via a function block on the control as well as via a corresponding function block in the field bus master.

Assignment target system/library

Target system	Library
From MLC 10VRS onwards	RIL_ParameterChannel

Tab. 11-1: Reference table of the IL_ParameterChannel function block

Interface description

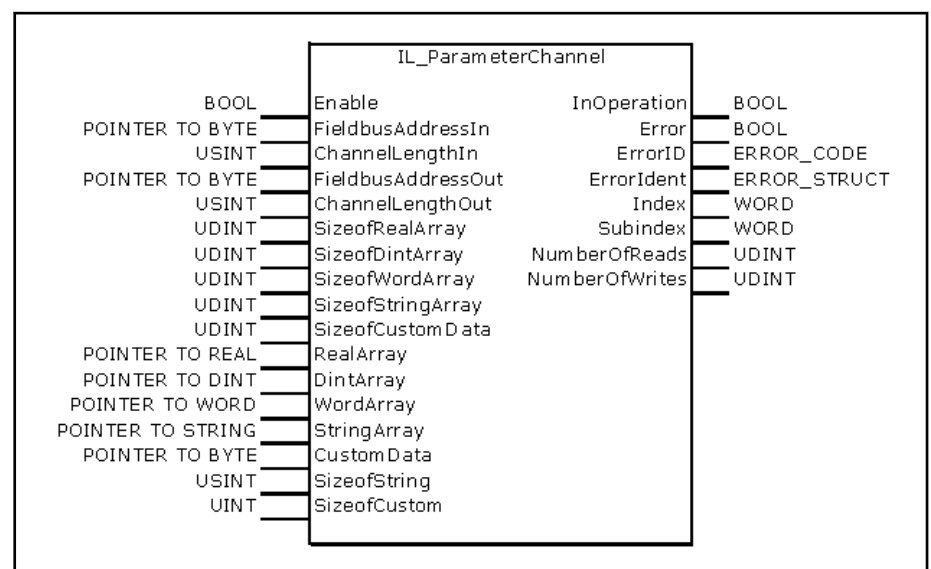


Fig. 11-1: IL_ParameterChannel function block

	Name	Data type	Description
VAR_INPUT	Enable	BOOL	Function block processing 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.

RIL_ParameterChannel.library

	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 have 4845 elements
	DintArray	POINTER TO DINT	A pointer to an array of DINT values. The array can have 4845 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 specified when defining 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 Troubleshooting, page 678
	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. and default values and
applying the inputs

Name	Type	Min. value	Max. value	Default value	Effective
Enable	BOOL			FALSE	Continuous
FieldbusAddressIn	POINTER TO BYTE	0x00000000	0xFFFFFFFF	0x00000000	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	0x00000000	0xFFFFFFFF	0x00000000	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	0x00000000	0xFFFFFFFF	0x00000000	Rising edge at "Enable"
DintArray	POINTER TO DINT	0x00000000	0xFFFFFFFF	0x00000000	Rising edge at "Enable"
WordArray	POINTER TO WORD	0x00000000	0xFFFFFFFF	0x00000000	Rising edge at "Enable"
StringArray	POINTER TO STRING	0x00000000	0xFFFFFFFF	0x00000000	Rising edge at "Enable"
CustomData	POINTER TO BYTE	0x00000000	0xFFFFFFFF	0x00000000	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. and 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. There are 5 different 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

RIL_ParameterChannel.library

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. The 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 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 Custom data can be used to consistently transfer a higher number of data. The Custom data 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.



In case of Custom data, a data set has to be copied completely. Thus, the time required for copying is significantly important at runtime of the function block. Measurements have shown the following maximum runtime for copying and editing the protocol:

- 256 bytes ca. 12 µs,
- 4 KBytes ca. 20 µs and
- 8 KBytes ca. 40 µs are required on an L65

This has to be considered for the task configuration.



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.

For Custom data of 4 KB, the runtime can increase from 20 µs up to 30ms!



The created pointers and lengths may not change, when the "Enable" input is TRUE.

This has to be especially considered during an online change if the sizes of the created arrays change.

If the pointers or lengths change at runtime, an error is 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
0x0000	0x1FFF		Reserved
0x2000	0x2FFF	Drive # (1-64)	Index = 0x2000 + S-parameter (only parameter set 0 is possible, no EIDN)
0x3000	0x3FFF	Drive # (1-64)	Index = 0x3000 + P-parameter (only parameter set 0 is possible, no EIDN)
0x4000	0x4FFF	Axis # (1-64)	Index = 0x4000 + A-parameter (only parameter set is 0 possible, no EIDN)
0x5000	0x5AEC		Free
0x5AED	0x5AED	0x01 – 0xFF	Index = 0x5AED; Subindex = Custom data array element; (elements 1 - 255)
0x5AEE	0x5AF7	0x01 – 0xFF	Index = (0x5AEE + [(WORD array element - 1) / 255]; subindex = (WORD array element) – [(index - 0x5AEE) * 255]; (elements 1 - 2295)
0x5AF8	0x5AFF	0x01 – 0xFF	Index = (0x5AF8 + [(STRING array element - 1) / 255]; subindex = (STRING array element) – [(index - 0x5AF8) * 255]; (elements 1 - 2040)
0x5B00	0x5CEB	0x00 – 0xFF	Free
0x5CEC	0x5CFF	0x01 – 0xFF	Index = (0x5CEC + [(REAL array element - 1) / 255]; subindex = (REAL array element) – [(index - 0x5CEC) * 255]; (elements 1 - 4845)
0x5D00	0x5D13	0x01 – 0xFF	Index = (0x5D00 + [(DINT array element - 1) / 255]; subindex = (DINT array element) – [(index - 0x5D00) * 255]; (elements 1 - 4845)
0x5D14	0x5FFF	0x00 – 0xFF	Free
0x6000	0x6FFF	0x00	Index = 0x6000 + C-parameter (only parameter set 0 is possible, no EIDN)

RIL_ParameterChannel.library

0x7000	0x7FFF	Kinematics # (1-16)	Index = 0x7000 + K-parameter (only parameter set 0 is possible, no EIDN)
0x8000	0x8FFF	Probe # (1-99)	Index = 0x(000 + M parameter (only parameter set 0 is possible, no EIDN)
0x9000	0x9FFF	PLS # (1-4)	Index = 0x9000 + N-parameter (only parameter set 0 is possible, no EIDN)
0xA000	0xAFFF	Osci # (1-16)	Index = 0xA000 + O-parameter (only parameter set 0 is possible, no EIDN)
0xB000	0xFFFF		Free

Tab. 11-4: Assigned address ranges for index/subindex

Troubleshooting

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
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000001	"FieldbusAddressIn" is either not assigned or is provided with 0.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000002	"ChannelLengthIn" is either not assigned or is provided with an invalid value (valid values are in the range 6 <= "ChannelLengthIn" <= 18).
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000003	"FieldbusAddressOut" is either not assigned or is provided with 0.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000004	"ChannelLengthOut" is either not assigned or provided with an invalid value (valid values are in the range 6 <= "ChannelLengthOut" <= 18).
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000005	"SizeofRealArray" is provided with an invalid value (valid values are in the range 0 <= "SizeofRealArray" <= 19380).
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000006	"SizeofDintArray" is provided with an invalid value (valid values are in the range 0 <= "SizeofDintArray" <= 19380).
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000007	"SizeofWordArray" is provided with an invalid value (valid values are in the range 0 <= "SizeofWordArray" <= 4590).
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000008	"SizeofStringArray" is provided with an invalid value (valid values are in the range 0 <= "SizeofStringArray" <= 522240).
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000009	"SizeofCustomData" exceeds the value allowed (2088960).

ErrorID	Additional1	Additional2	Description
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#0000000A	"SizeofRealArray" is provided with an invalid value. "SizeofRealArray" has to be divisible by 4 without remainder.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#0000000B	"SizeofRealArray" is provided with an invalid value. "SizeofDintArray" has to be divisible by 4 without remainder.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#0000000C	"SizeofWordArray" is provided with an invalid value. "SizeofWordArray" has to be divisible by 2 without remainder.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#0000000D	"SizeofStringArray" or "SizeofString" is provided with an invalid value. "SizeofStringArray" has to be divisible by "SizeofString" + 1 without remainder.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#0000000E	"SizeofCustomData" or "SizeofCustom" is provided with an invalid value. "SizeofCustomData" has to be divisible by "SizeofCustom" without remainder.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#0000000F	"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#00000006	16#0000000x	An invalid state has been detected in the function block. The function block must be restarted.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000010	The pointer in "FieldbusAddressIn" or the length "ChannelLengthIn" changed.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000011	The pointer in "FieldbusAddressOut" or the length "ChannelLengthOut" changed.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000012	The pointer in "RealArray" or the length "SizeofRealArray" changed.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000013	The pointer in "DintArray" or the length "SizeofDintArray" changed.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000014	The pointer in "WordArray" or the length "SizeofWordArray" changed.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000015	The pointer in "StringArray", the length "SizeofStringArray" or the length "SizeofString" changed.
IN-PUT_RANGE_ERROR (16#0001)	16#00000002	16#00000016	The pointer in "CustomArray", the length "SizeofCustomArray" or the length "SizeofCustom" changed.

Tab. 11-5: Error codes of the IL_ParameterChannel function block

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.

RIL_ParameterChannel.library

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
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:

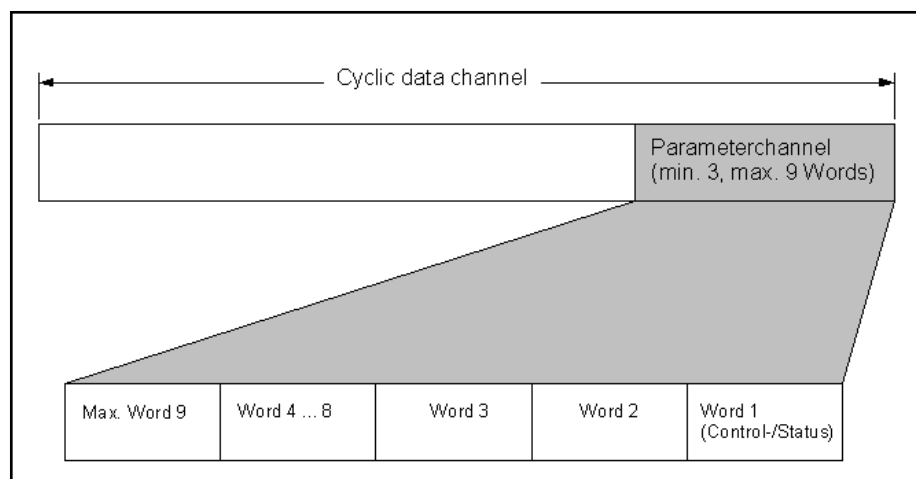


Fig. 11-2: Data content of the field bus in the parameter channel

Control/Status Word

The control word is assigned as follows:

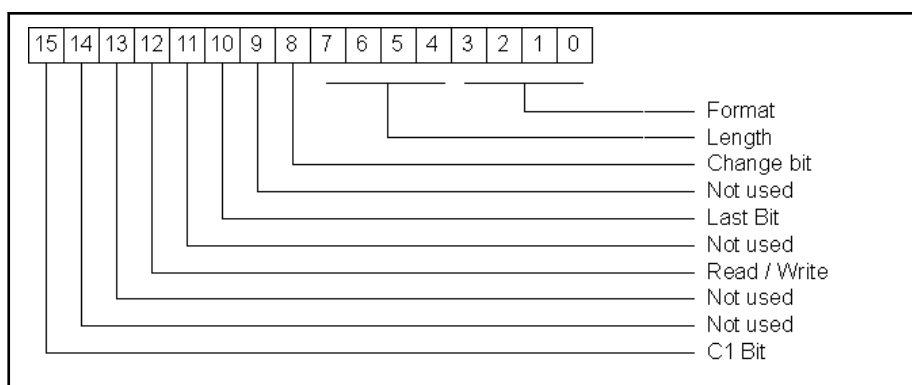


Fig. 11-3: Assignment of the control word in the parameter channel

The status word is assigned as follows:

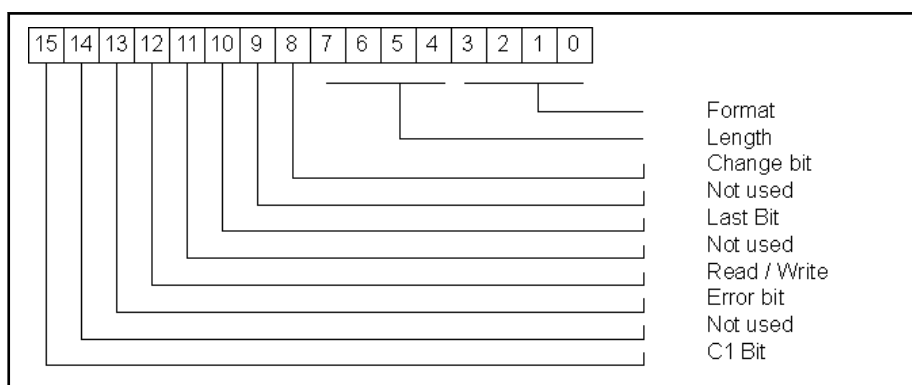


Fig. 11-4: Assignment of the status word in the parameter channel

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.

RIL_ParameterChannel.library

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:

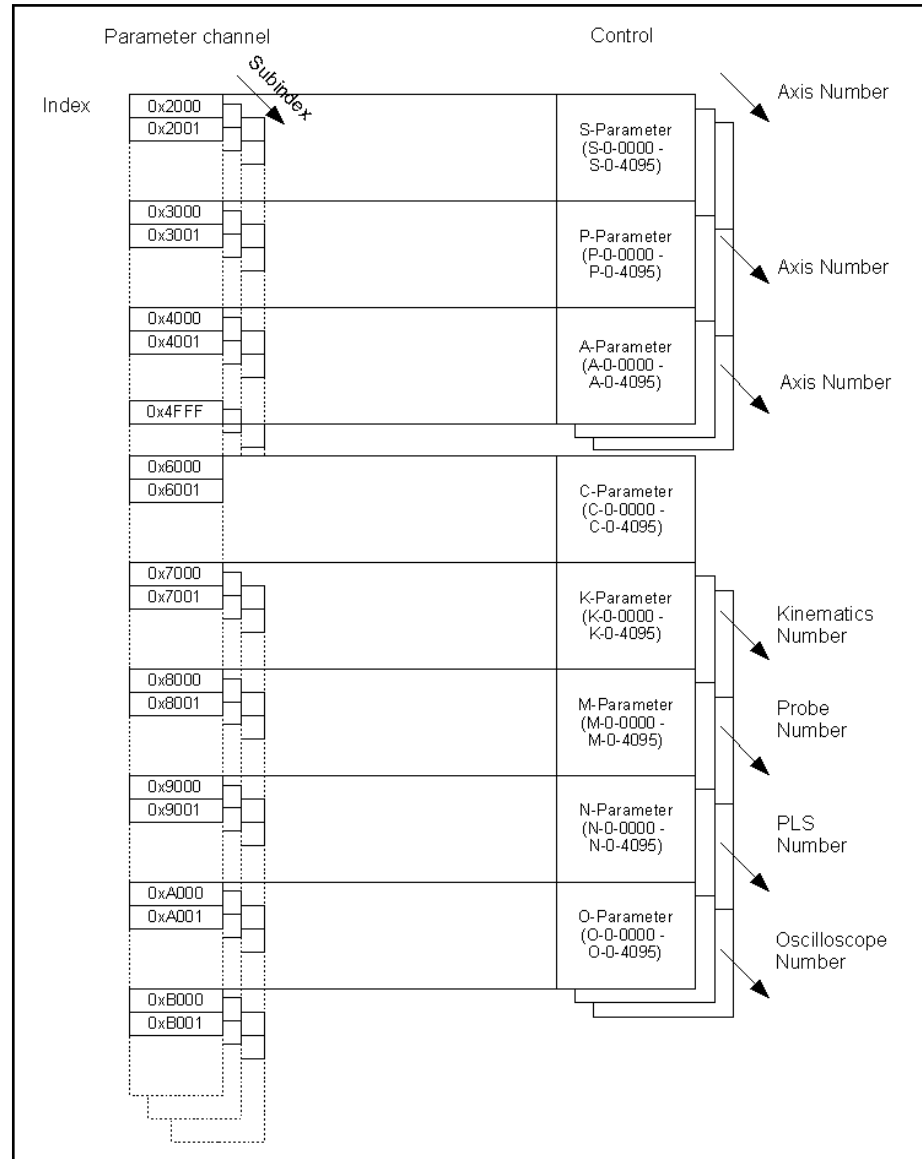


Fig. 11-5: Assignment of index/subindex to the parameters

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, $0x4000 + 100 = 0x4064$ 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.

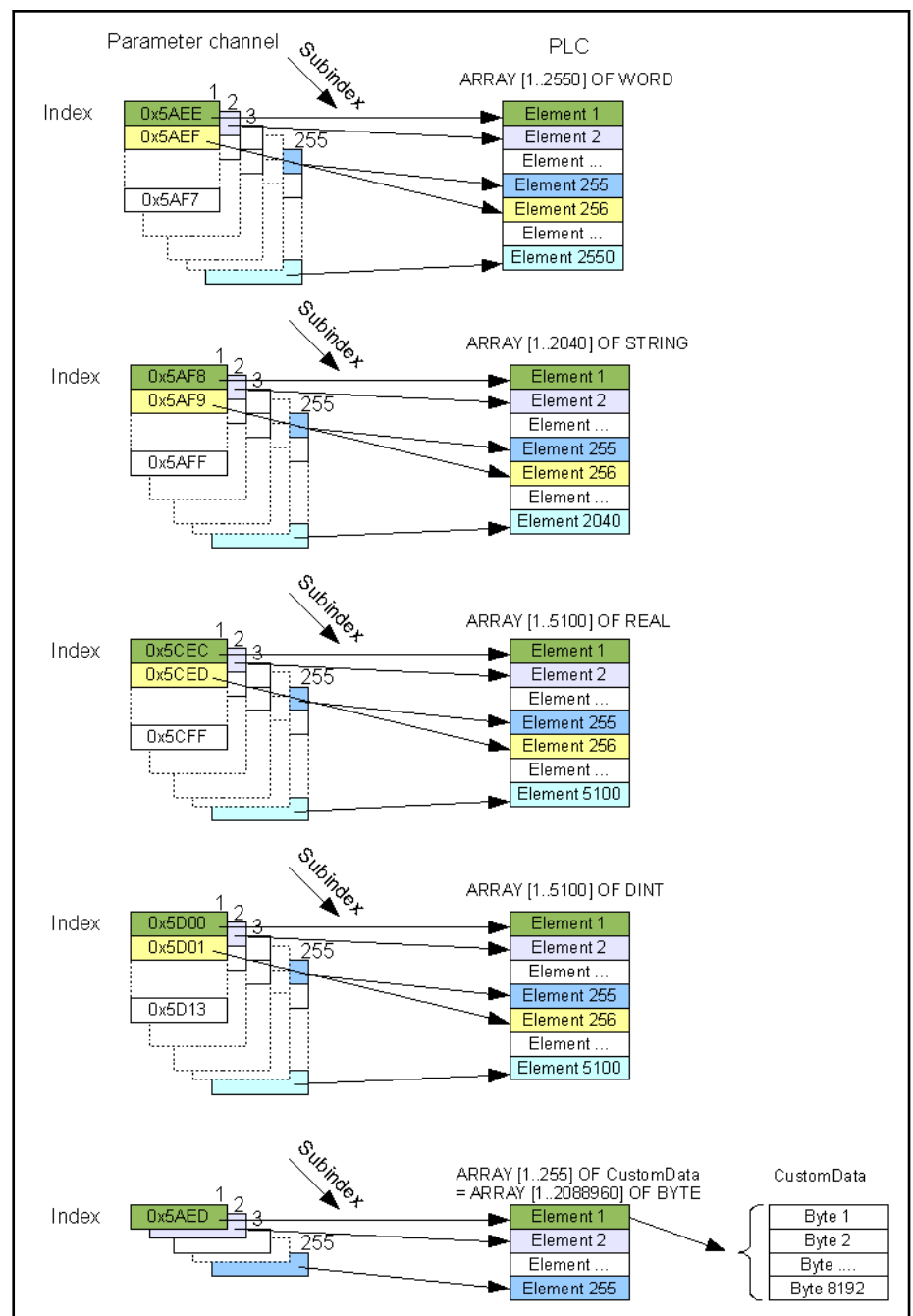


Fig. 11-6: Assignment of index/subindex to PLC variables

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 0x5CEC and the element number is 257.

The following results for the index:

RIL_ParameterChannel.library

$\text{Index} = 0x5CEC + (257 - 1) / 255 = 0x5CEC + (256 / 255) = 0x5CEC + 1 = 0x5CED$

The following results for the subindex:

$\text{Subindex} = 257 - [(0x5CED - 0x5CEC) * 255] = 257 - [1 * 255] = 2$

To convert the index/subindex to the element number, the following formula applies:

Element number = [(index - starting index) * 255] + subindex

Example:

Which element is queried by the index 0x5D02 and by the subindex 3?

The index is in the range from 0x5D00 to 0x5D13 and thus, it is an element in the DINT array. The element number is:

$\text{Element number} = [(0x5D02 - 0x5D00) * 255] + 3 = 2 * 255 + 3 = 513$

The same mapping rules as above apply to the CUSTOM data. Since only one index is used, the mapping can be simplified respectively.

Index = start index = 0x5AED

Subindex = 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 0x5AF9 is written on the index and 5 on the subindex. On which element was thus written?

0x5AF9 is in the range from 0x5AF8 to 0x5AFF. Thus, it is a string.

The element number results from:

$\text{Element number} = (0x5AF9 - 0x5AF8) * 255 + 5 = 0x1 * 255 + 5 = 260$

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: 1000 000x 0000 1111b. 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 1000 000x 0000 1111b.

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 [Troubleshooting, page 678](#)).

The error code 0x1F00 is 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 0x5AF8 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

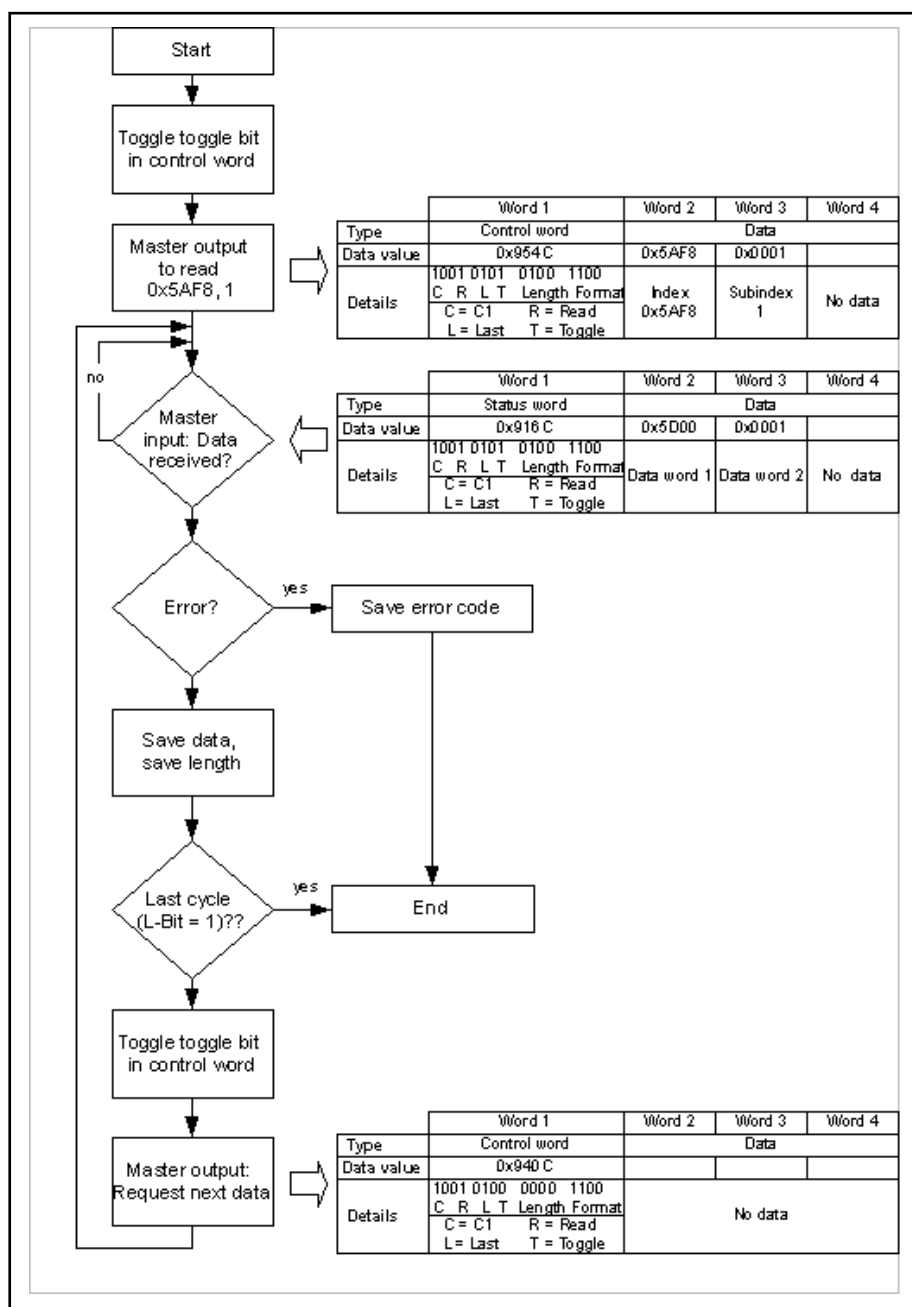


Fig. 11-7: Example on how to read a PLC STRING variable

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 0x4002 and the subindex 22 have to be selected

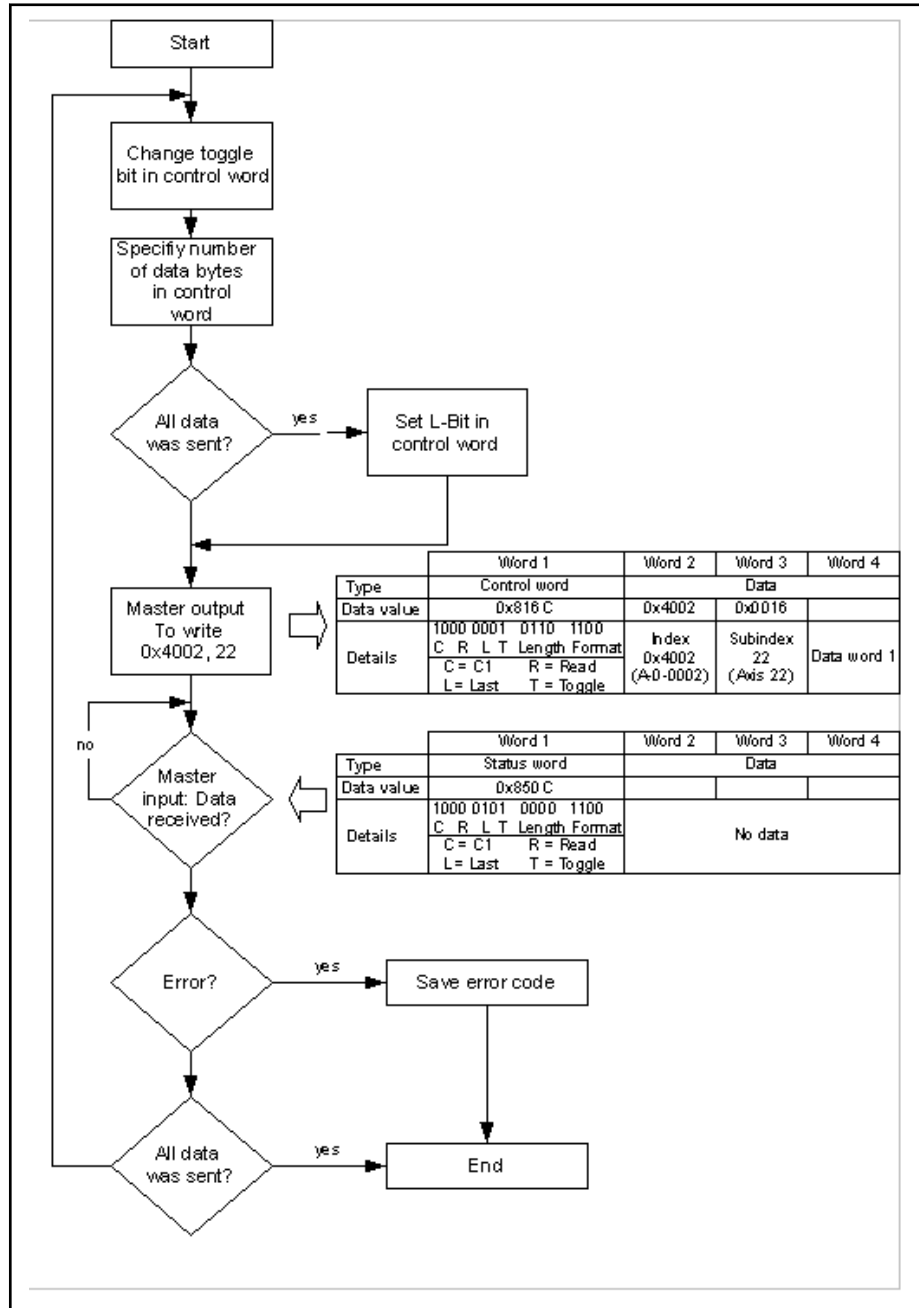


Fig. 11-8: Example on how to write an axis parameter

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.

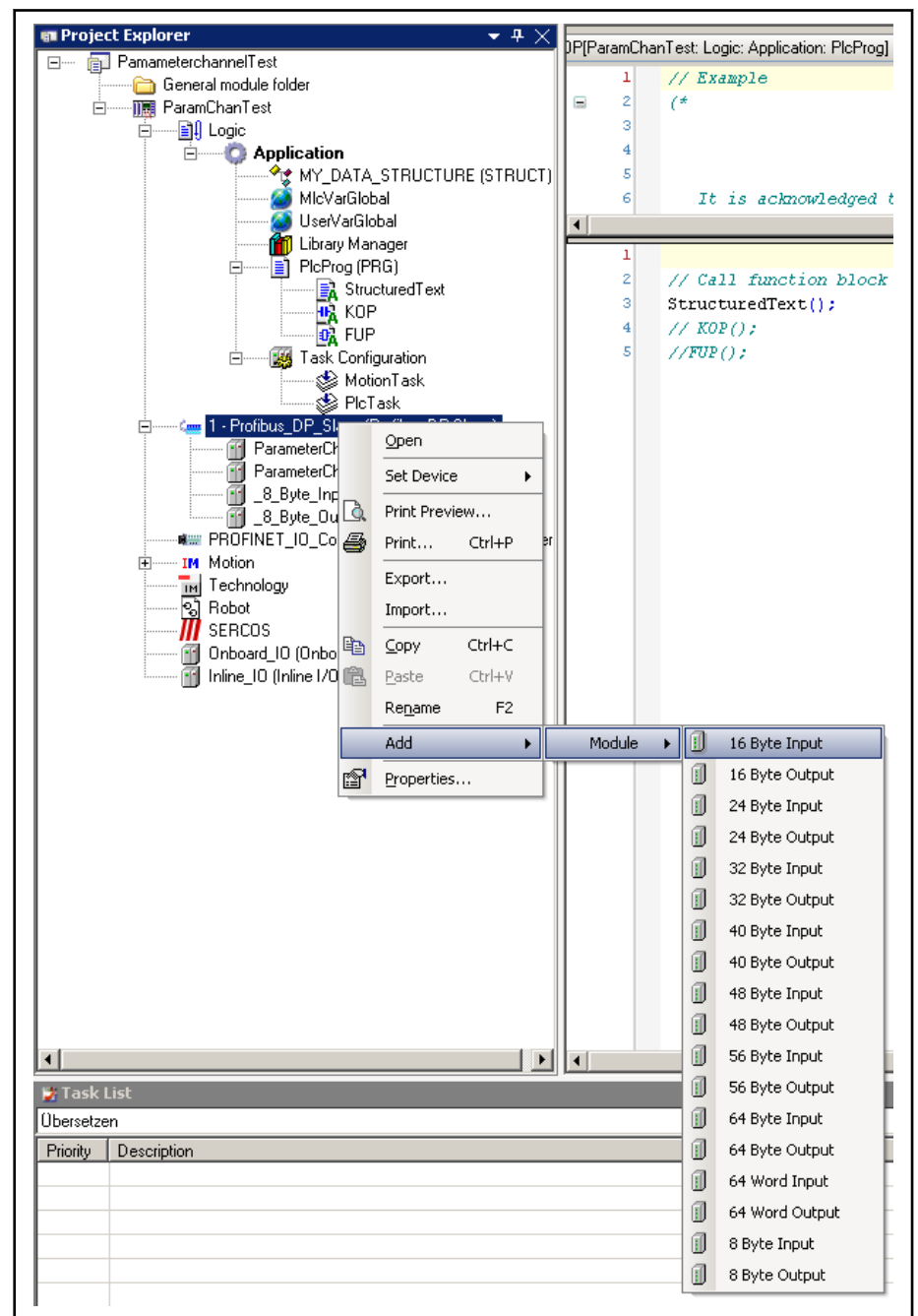


Fig. 11-9: Profibus modules used as parameter channel

RIL_ParameterChannel.library

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").

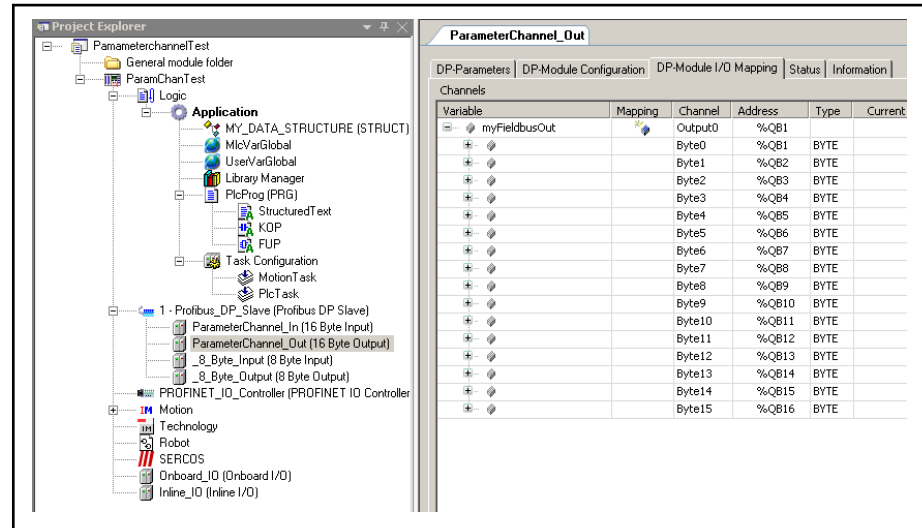


Fig. 11-10: Name and address of a Profibus module

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

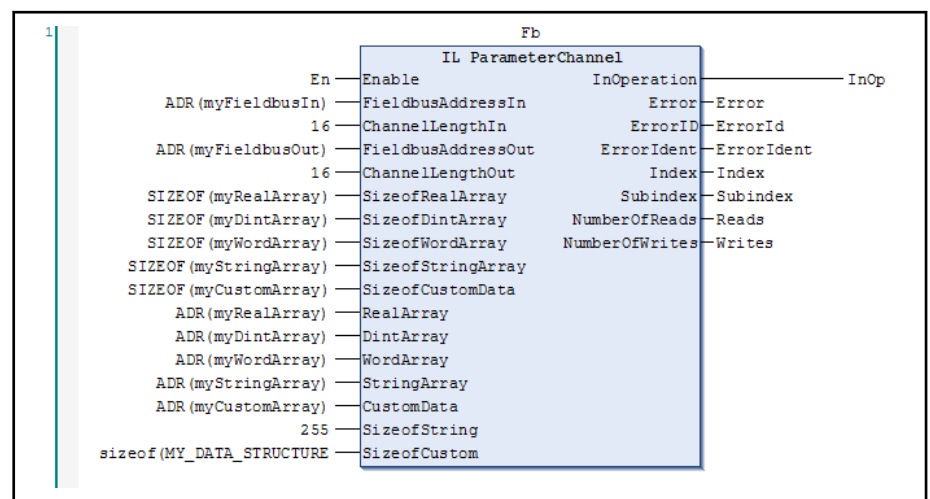


Fig. 11-12: Use of the IL_ParameterChannel function block in FBD

RIL_ParameterChannel.library

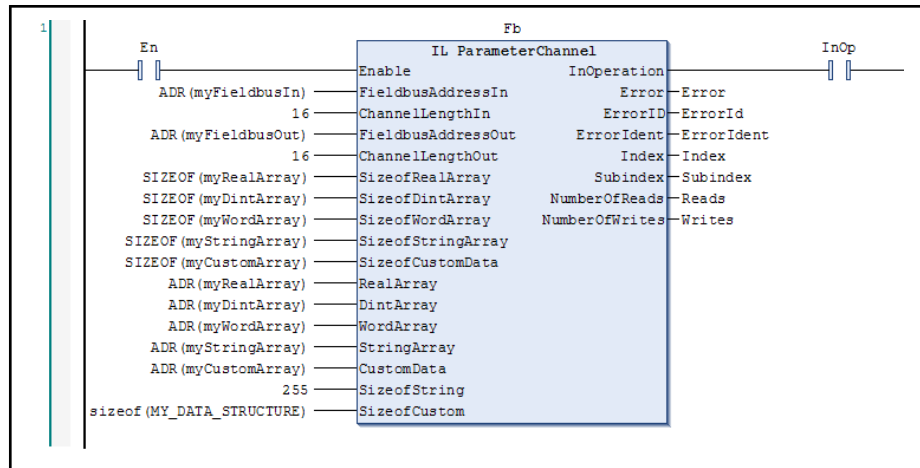


Fig. 11-13: Use of the IL_ParameterChannel function block in LD

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 IndraMotion MLC/IndraLogic 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 IndraMotion MLC/IndraLogic 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.

IndraMotion MLC/IndraLogic 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 IndraMotion MLC/IndraLogic 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" 0x5AEE – 0x5AF7 addressed the registers in VisualMotion. Now, the WORD array is accessed in the PLC
- "Index" 0x5CEC – 0x5CFF addressed the program floats in VisualMotion. Now, the REAL array is accessed in the PLC
- "Index" 0x5D00 – 0x5D13 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
0x0000	0x1FFF		Reserved
0x2000	0x2FFF	Drive # (1-64)	Index = 0x2000 + S-parameter
0x3000	0x3FFF	Drive # (1-64)	Index = 0x3000 + P-parameter
0x4000	0x4FFF	Axis # (1-64)	Index = 0x4000 + A-parameter
0x5000	0x5AEC		Free
0x5AED	0x5AED	0x01 – 0xFF	Index = 0x5AEDSubindex = Custom data array element (element 1 - 255)
0x5AEE	0x5AF7	0x01 – 0xFF	Index = (0x5AEE + [(WORD array index - 1) / 255] subindex = (WORD array index) – [(index - 0x5AEE) * 255] (index 1 - 2295)
0x5AF8	0x5BEC	0x01 – 0xFF	Index = (0x5AF8 + [(STRING array index - 1) / 255] subindex = (STRING array index) – [(index - 0x5AF8) * 255] (index 1 - 244)
0x5BED	0x5CEB	0x00 – 0xFF	Free
0x5CEC	0x5CFF	0x01 – 0xFF	Index = (0x5CEC + [(REAL array index - 1) / 255] subindex = (REAL array index) – [(index - 0x5CEC) * 255] (index 1 - 4845)
0x5D00	0x5D13	0x01 – 0xFF	Index = (0x5CEC + [(DINT array index - 1) / 255] subindex = (DINT array index) – [(index - 0x5CEC) * 255] (index 1 - 4845)
0x5D14	0x5FFF	0x00 – 0xFF	Free
0x6000	0x6FFF	0x00	Index = 0x6000 + C-parameter
0x7000	0x7FFF	Kinematics # (1-16)	Index = 0x7000 + K-parameter
0x8000	0x8FFF	Probe # (1-99)	Index = 0x8000 + M-parameter

RIL_ParameterChannel.library

0x9000	0x9FFF	PLS # (1-4)	Index = 0x9000 + N-parameter
0xA000	0xAFFF	Osci # (1-16)	Index = 0xA000 + O-parameter
0xB000	0xFFFF		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 IndraMotion MLC/ IndraLogic XLC	ErrorCode VisualMotion	Description
16#8500	16#8500	Data too long (VisualMotion: >128 Byte; MLC > 64 KBytes)
--	16#8800	Error not available on IndraMotion MLC/IndraLogic 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 IndraMotion MLC/IndraLogic 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 IndraMotion MLC/IndraLogic 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 IndraMotion MLC/IndraLogic 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 IndraMotion MLC/IndraLogic 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 IndraMotion MLC/IndraLogic XLC and VisualMotion error codes in the parameter channel

12 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.....	17
Acceleration torque.....	534, 551
Acceleration torque compensation.....	534, 551
ANSI Z535.6-2006.....	10
Averaging	
Material thickness.....	565

B

Backing up and restoring drive parameters.....	35
Bell-crank kinematics.....	390

C

Calculating material variables.....	565
CamLock.....	156
Application example.....	156
CamLock function block	
MB_CamLock.....	161
MB_PrepareCams.....	159
Closed-loop dancer position control..	473, 479, 486
MB_WinderDancerCtrlType01.....	509
MB_WinderDancerCtrlType02.....	517
Commissioning	
Dancer position controller.....	487
Diameter calculator.....	497
Diameter calculator with closed-loop ten-	
sion control.....	538
Diameter calculator with dancer.....	519
Diameter calculator with load cell.....	538
Diameter calculator with open-loop ten-	
sion control.....	538
Diameter calculator without sensor.....	538
Compensation	
Acceleration torque.....	534, 551
Friction torque.....	533, 550
Configure the side register.....	382
Core diameter.....	560
Crank kinematics.....	413
Counting direction.....	417
Mechanical arrangement 1.....	415
Mechanical arrangement 2.....	416
Mechanical translatory position.....	417
Overview.....	413
Travel range equation.....	417
Use case 1: Crank kinematics.....	414
Use case 2: Synchronous mode with cam...	414
Variable and terms.....	417
Xmech.....	417
Xvirt.....	417
CrossCutter.....	604
CrossSealer.....	601
CurrentDancerDeviation.....	479, 486
Cycle-synchronous lock-on and lock-off	156
Cyclic data channels.....	129

D

Dancer position controller	
Commissioning.....	487
Control variable limitation.....	478, 485
Parameterization.....	488
Data structure	
MB_AXIS_SYNCHRONISATION.....	540, 557
Default values	
MB_CalcInertiaLimitType02.....	560
MB_DancerControlType03.....	477, 484
MB_DiameterCalculatorType03.....	492
MB_DiameterMeasurementType01.....	500
MB_UnwindMaterialType02.....	564
MB_WinderDancerCtrlType01.....	506, 514
MB_WinderTensionCtrlType02.....	526
Derivative time.....	479
Detect an externally measured diameter.....	498
Determine dancer constant	
MB_WinderDancerCtrlType01.....	508
MB_WinderDancerCtrlType02.....	516
Diameter calculation	
Average value generation.....	494
Calculating the moment of inertia.....	495
Commissioning.....	497
Preset of the diameter.....	493
Proportionality in diameter calculation.....	494
Splice.....	495
With dancer.....	488
Diameter calculation, winder.....	489
Diameter calculator	
Engaging condition.....	532, 550
MB_WinderDancerCtrlType01.....	508
MB_WinderDancerCtrlType02.....	516
Diameter calculator with closed-loop tension	
control.....	521, 530
Command winding axis.....	538, 555
Commissioning.....	538
Determine direction of rotation of winding	
axis.....	555
Determine friction torque at maximum	
speed.....	539, 556
Determine friction torque at standstill...	539, 556
Determine minimum friction torque.....	540, 557
Determine winder type.....	555
Diameter calculator with dancer.....	502
Command winding axis.....	519
Commissioning.....	519
Functional diagram.....	509, 518
Diameter calculator with open-loop tension	
control.....	529
Acceleration torque compensation.....	529, 530
Calculating diameter-dependent variables...	531
Command winding axis.....	538, 555
Commissioning.....	538

Index

Determine direction of rotation of winding axis.....	555
Determine friction torque at maximum speed.....	539, 556
Determine friction torque at standstill...	539, 556
Determine minimum friction torque.....	540, 557
Determine winder type.....	555
Friction compensation, speed-depend-ent.....	529, 530
Operating tension parameterization.....	529, 530
PI-controller for tension.....	531
Standstill tension parameterization.....	529, 530
Tension winding material.....	529, 530
Diameter, externally measured.....	498
Distance accumulator.....	168
Documentation structure.....	9
Double-linked list.....	142
Drive parameter settings.....	249
Drive parameters	
Back up and restore.....	35

E

Engaging condition.....	532, 550
Error codes	
IL_LinkedList.....	146
IL_ParameterChannel.....	679
IL_ParameterChannel, parameter channel..	680
IL_RegiQualityType01.....	316
IL_TempControlType01.....	83
IL_TempControlType02.....	92
MB_AbortTrigger.....	124
MB_AllocCyclicParameter.....	132
MB_AxisDistanceAccumulator.....	171
MB_BellCrankData.....	396
MB_CalcInertiaLimitType02.....	560
MB_CamLock.....	168
MB_CamTableBellCrank.....	400
MB_CamTableBellCrankSuperimposed.....	405
MB_CamTableCrank.....	422
MB_CamTableCrankSuperimposed.....	427
MB_ContinuousAdjustType01.....	29
MB_ContinuousAdjustType02.....	32
MB_CoordinatePairCamType01.....	630
MB_CrossCutSealType01.....	609
MB_CrossCutterCalcType04.....	617
MB_DancerControlType03.....	480, 486
MB_DiameterCalculatorType03.....	495
MB_DiameterMeasurementType01.....	501
MB_DistanceAccumulator.....	170
MB_FeedMovementCalcType01.....	637
MB_FeedMovementCalcType02.....	643
MB_FreeCyclicParameter.....	133
MB_GetCyclicChannelConfig.....	135
MB_GetCyclicParameterHandle.....	137
MB_GetLastCyclicParameterError.....	139
MB_GetSystemInfo.....	154
MB_IncrementalAdjustType01.....	35

MB_LoadCamData.....	45
MB_MagicBeltType01.....	205
MB_PhiToXvirt.....	429
MB_PhiToYvirt.....	407
MB_PilgrimStepCalcType01.....	270
MB_PositionMonitoring_GantryCant.....	221
MB_PositionMonitoring_PosCollision.....	224
MB_PrepareCams.....	161
MB_PrintLengthCorrCalcType01.....	275
MB_ReachVelocityType02.....	237
MB_RegiMeasuringType01.....	306
MB_RegiMeasuringType02.....	310
MB_RegiMeasuringType03.....	313
MB_RegiRegulateType01.....	321
MB_RegiRegulateType02.....	329
MB_RegiRegulateType03.....	335
MB_RegiRegulateType04.....	343
MB_RegiSetpointLockType01.....	346
MB_RegisterControllerSideType01.....	382
MB_RegisterControllerType02.....	356
MB_RegisterControllerType04.....	368
MB_SlotterCalcType01.....	660
MB_SlotterCalcType02.....	671
MB_SmartBeltType01.....	243
MB_TensionControlLoadCellMatParType01	300
MB_TensionControlLoadCellMatParType02	459
MB_TensionControlLoadCellType01.....	287
MB_TensionControlLoadCellType02.....	445
MB_TensionDecouplingNetworkType01.....	468
MB_TouchProbe.....	117
MB_TouchProbeContinuous.....	120
MB_UnwindMaterialType02.....	566
MB_WinderDancerCtrlType02.....	518
MB_WinderDancerCtrlType03.....	510
MB_WinderTensionCtrlSpeedType01.....	554
MB_WinderTensionCtrlType02.....	535, 536
MB_WindSpeedControlAdaptionType01.....	574
MB_WindTaperProfileType01.....	570
MB_WriteExpectWindow.....	123
MB_XvirtToXmech.....	430
MB_YvirtToPhi.....	409
MB_YvirtToPhiPoly5.....	411
MB_YvirtToYmech.....	413
MC_DigitalCamSwitch.....	65
ML_AxisBackup.....	37
ML_AxisRestore.....	40
ML_FlyingShear.....	178
ML_InterpolationLinear.....	48
ML_SagControlA.....	232
ML_SagControlC.....	228
ML_TwoPosCtrlType01.....	42
MX_FlyingShear.....	178

F

Filter

Actual torque value.....	534
Tension control value.....	552

Fine adjustment	
Resulting.....	493, 494
Firmware, required	
MB_CalcInertiaLimitType02.....	561
MB_DancerControlType03.....	481, 487
MB_DiameterCalculatorType03.....	496
MB_DiameterMeasurementType01.....	502
MB_TraverseControlType01.....	593
MB_UnwindMaterialType02.....	566
MB_WinderDancerCtrlType01.....	511
MB_WinderDancerCtrlType02.....	519
MB_WinderTensionCtrlType02.....	536
MB_WindSpeedControlAdaptionType01.....	574
MB_WindTaperProfileType01.....	571
ML_FlyingShear.....	178
Flying Shear.....	171
Flying splice.....	534, 552
Friction compensation.....	533, 550
Friction torque compensation.....	533, 550
Function	
MB_GetModuloType03.....	149
MB_GetModuloType04.....	150
Function block	
MB_CalcInertiaLimitType02.....	559
MB_DancerControlType03.....	475
MB_DancerControlType04.....	481
MB_DiameterCalculatorType03.....	489
MB_TraverseControlType01.....	574
MB_UnwindMaterialType02.....	561
MB_WinderDancerCtrlType01.....	503
MB_WinderDancerCtrlType02.....	512
MB_WinderTensionCtrlSpeedType01.....	542
MB_WinderTensionCtrlType02.....	522
MB_WindSpeedControlAdaptionType01.....	571
MB_WindTaperProfileType01.....	567
Function block for applications	
Diameter calculator with closed-loop ten-	
sion control.....	521
Diameter calculator with load cell.....	521
Diameter calculator with open-loop ten-	
sion control.....	521
Diameter calculator without sensor.....	521
Functional diagram	
Diameter calculator with closed-loop ten-	
sion control.....	531
Diameter calculator with load cell.....	531
Diameter calculator without sensor.....	529

G

Gear fine adjustment.....	489, 498, 534
Gear ratio.....	489, 498
Gear setting - calculation.....	614

H

Hardware, required	
MB_CalcInertiaLimitType02.....	561
MB_DancerControlType03.....	480, 487

MB_DiameterCalculatorType03.....	495
MB_DiameterMeasurementType01.....	501
MB_TraverseControlType01.....	592
MB_UnwindMaterialType02.....	566
MB_WinderDancerCtrlType01.....	511
MB_WinderDancerCtrlType02.....	519
MB_WinderTensionCtrlType02.....	536
MB_WindSpeedControlAdaptionType01.....	574
MB_WindTaperProfileType01.....	571
ML_FlyingShear.....	178
Hazard warnings.....	10
Helpdesk.....	693
Hotline.....	693

I

I/O assignment	
MB_CalcInertiaLimitType02.....	560
MB_UnwindMaterialType02.....	563
IL_LINKED_LIST_MAINTENANCE.....	146
IL_LINKED_LIST_MODE.....	146
IL_LinkedList.....	142
IL_ParameterChannel.....	673
Commission the function block.....	687
Control/status word.....	680
Error telegrams.....	684
Index and subindex.....	682
Initialization of the communication.....	684
Parameter channel in the cyclic data	
channel.....	680
Protocol, error handling and commissioning	680
IL_REGI_QUALITY.....	316
IL_RegiQualityType01.....	314
IL_TEMPCONTROL_CONTR_PARAM.....	93
IL_TEMPCONTROL_CONTROL_MODE.....	94
IL_TEMPCONTROL_EXTENDED_SETTINGS... ..	94
IL_TEMPCONTROL_LIMITS.....	95
IL_TEMPCONTROL_MODEL_PARAM.....	95
IL_TempControlType01.....	66
IL_TempControlType02	84
IndraLogic steps, required	
ML_FlyingShear.....	179
IndraWorks.....	481, 487, 536, 571, 574, 593
Inertia.....	489
Integral action time.....	479, 486
Interface.....	172
Introduction and definitions.....	643
Introduction to SmartBelt usage.....	244

J

Jogging variables.....	23
Calculate change rate.....	23

L

Lead time.....	486
library	
ML_TechInterface.lib.....	540, 557
Library	

Index

ML_TechBase.....	21	MB_DiameterCalculatorType03.....	489
ML_TechCrank.....	389	MB_DiameterMeasurementType01.....	498
ML_TechMotion.....	155	MB_DistanceAccumulator.....	169
ML_TechRegi.....	303	MB_FeedMovementCalcType01.....	631
ML_TechTensionAdvanced.....	431	MB_FeedMovementCalcType02.....	637
MX_TechRegi.lib.....	303	MB_FreeCyclicParameter.....	132
MX_TechWinder.....	471	MB_FS_RETAIN_DATA.....	178
RIL_ParameterChannel.....	673	MB_GetCyclicChannelConfig.....	134
RMB_TechCam.....	623	MB_GetCyclicParameterHandle.....	135
RMB_TechCrossCutCrossSeal.....	595	MB_GetLastCyclicParameterError.....	139
RMB_TechWinder.....	471	MB_GetModuloType01.....	147
Linked list.....	142	MB_GetModuloType02.....	148
Logic.....	142	MB_GetModuloType03.....	149
		MB_GetModuloType04.....	150
M		MB_GetSystemInfo.....	152
MagicBelt.....	182	MB_IncrementalAdjustType01.....	33
Material calculator		MB_InitTouchProbe.....	106
Averaging the material thickness.....	565	MB_LoadCamData.....	43
Preset of material variables.....	565	MB_MagicBeltType01.....	184
Reset.....	565	Commissioning Notes.....	205
Material counter.....	561	MB_MasterToPhi.....	427
Mathematical functions.....	46	MB_MBELT_FEEDTYPE.....	214
MB_AbortTrigger.....	123	MB_MBELT_GENERAL.....	206
MB_AllocCyclicParameter.....	130	MB_MBELT_INFEED.....	211
MB_AxisDistanceAccumulator.....	170	MB_MBELT_INFEEDSTEPS.....	214
MB_BellCrankData.....	394	MB_MBELT_MODEAUTO.....	210
MB_CalcInertiaLimitType02.....	559	MB_MBELT_MODEJOG.....	209
MB_CAM_TABLE_DATA.....	618	MB_MBELT_MODES.....	206
MB_CamLock.....	161	MB_MBELT_MODETUNE.....	209
MB_CamLock Parameterization.....	168	MB_MBELT_MOVE.....	217
MB_CAMPROF.....	45	MB_MBELT_OUTFEED.....	216
MB_CAMPROF_SLOTTER.....	671	MB_MBELT_STARTTYPE.....	211
MB_CAMTABLE.....	46	MB_MBELT_STATUS.....	218
MB_CamTableBellCrank.....	397	MB_MBELT_STOPTYPE.....	211
MB_CamTableBellCrankSuperimposed.....	400	MB_PhiToXvirt.....	428
MB_CamTableCrank.....	418	MB_PhiToYvirt.....	405
MB_CamTableCrankSuperimposed.....	422	MB_PilgrimStepCalcType01.....	260
MB_ContinuousAdjustType01.....	26	MB_POS_MON_GANTRY_CANT.....	222
MB_ContinuousAdjustType02.....	30	MB_POS_MON_POS_COLLISION.....	224
MB_COORDINATE_PAIR_CAM_SELECT.....	630	MB_PositionMonitoring_GantryCant.....	220
MB_COORDINATE_PAIR_TYPE01.....	630	MB_PositionMonitoring_PosCollision.....	222
MB_CoordinatePairCamType01.....	623	MB_PrepareCams.....	159
MB_CORR_PROFILE.....	619	MB_PrintLengthCorrCalcType01.....	271
MB_CORRECTION_MODE.....	343	MB_PROBE_EXECUTION_MODE.....	125
MB_CRANK.....	418	MB_PUSH_OUT_CONFIG.....	621
MB_CrossCutSealType01.....	595	MB_RAMPING_MODE.....	344
CrossCutter use case.....	604	MB_ReachVelocityType02.....	233
CrossSealer use case.....	601	Interface.....	234
MB_CrossCutterCalcType04.....	609	MB_ReadCyclicParameter.....	137
Special features of IndraMotion MLC.....	617	MB_ReadCyclicRealParameter.....	137
MB_CURVETYPE.....	637	MB_RegiMeasuringType01.....	304
MB_CYCLIC_CHANNEL_CONFIG.....	141	MB_RegiMeasuringType02.....	306
MB_CYCLIC_CHANNELS.....	141	MB_RegiMeasuringType03.....	310
MB_CYCLIC_PARAM_REF.....	140	MB_RegiRegulateType01.....	317
MB_CYCLIC_TYPES.....	140	MB_RegiRegulateType02.....	321
MB_DancerControlType03.....	475	MB_RegiRegulateType03.....	329
MB_DancerControlType04.....	481	MB_RegiRegulateType04.....	335
		MB_RegiSetpointLockType01.....	344

MB_RegisterControllerSideType01.....	375	Measuring function blocks for register controllers.....	304
MB_RegisterControllerType02.....	346	ML_AxisBackup.....	35
MB_RegisterControllerType04.....	356	ML_AxisRestore.....	37
MB_RESOLUTION.....	620	ML_BACKUP_OPTIONS.....	40
MB_SlotterCalcType01.....	648	ML_BACKUP_PARAMS.....	40
MB_SlotterCalcType02.....	660	ML_BASIC_CAMSWITCH_REF.....	65
MB_SMARTBELT_STATISTICS.....	243	ML_BASIC_TRACK_REF.....	66
MB_SmartBeltType01.....	237	ML_COORD_PAIR_TYPE01.....	49
Introduction to SmartBelt usage.....	244	ML_DegToInc.....	125
MB_SYSTEM_TYPE.....	154	ML_DegToRad.....	126
MB_TENSION_CONTROL_ADJUST.....	460	ML_FlyingShear.....	171
MB_TENSION_CONTROL_CONFIG.....	459	Special features of IndraMotion MLC.....	178
MB_TENSION_CONTROL_MATERIAL.....	300	ML_IncToDeg.....	126
MB_TENSION_CONTROL_MATERIAL_TYPE.....	300	ML_IncToRad.....	127
MB_TENSION_CONTROL_MODE.....	301	ML_InterpolationLinear.....	46
MB_TENSION_DECOUPLING_AXIS_VALUE..	469	ML_MaxValue.....	48
MB_TENSION_DECOUPLING_CONTROL- LER_VALUE.....	468	ML_RadToDeg.....	127
MB_TENSION_PRE_CTRL.....	460	ML_RadToInc.....	128
MB_TensionControlLoadCellMatParType01.....	287	ML_SagControlA.....	229
MB_TensionControlLoadCellMatParType02.....	445	ML_SagControlC.....	226
MB_TensionControlLoadCellType01.....	276	ML_TechBase.library.....	21
MB_TensionControlLoadCellType02.....	431	ML_TechCrank.library.....	389
MB_TensionDecouplingNetworkType01.....	461	ML_TechMotion.library.....	155
MB_TouchProbe.....	114	ML_TechRegi.library.....	303
MB_TouchProbeContinuous.....	117	ML_TechTensionAdvanced.library.....	431
MB_TraverseControlType01.....	574	ML_TwoPosCtrlType01.....	41
MB_UnwindMaterialType02.....	561, 565	Modulo value calculation.....	147
Preset of material variables.....	565	Moment of inertia.....	495
Reset the material calculator.....	565	Multi-axis tension controller	
MB_WIND_CTRL_TYPE.....	526	MB_TensionControlLoadCellType02.....	431
MB_WIND_DANCER_LIMITS... 478, 484, 507, 516		MX_FlyingShear.....	171
MB_WIND_DIAM_CALC_MODE.....	547	Components and parameterization.....	180
MB_WIND_DIAMETER..... 506, 514, 526, 546		MX_TechRegi.lib.....	303
MB_WIND_DIRECTION.....	526, 545	MX_TechWinder.lib.....	471
MB_WIND_FRICTION.....	528, 548	MX(L)_FlyingShear.....	172
MB_WIND_TENS_DIAG.....	528, 548		
MB_WIND_TENSION.....	528, 547	O	
MB_WIND_TENSION_MODE.....	526	Overview on parameters	
MB_WIND_TORQUE_LIMITS.....	546	MLC axis parameters.....	593
MB_WinderDancerCtrlType01..... 503, 512			
MB_WinderTensionCtrlSpeedType01.....	542	P	
MB_WinderTensionCtrlType02.....	522	Parameter overview.....	593
MB_WindSpeedControlAdaptionType01.....	571	Parameterization	
MB_WindTaperProfileType01.....	567	Dancer position controller.....	488
MB_WriteCyclicParameter.....	138	Friction torque.....	533, 550
MB_WriteCyclicRealParameter.....	138	Parameterization for the IndraMotion MLC control	
MB_WriteExpectWindow.....	120	MB_WinderDancerCtrlType01.....	511
MB_XvirtToXmech.....	429	MB_WinderDancerCtrlType02.....	519
MB_YvirtToPhi.....	407	Parameterization for the IndraMotion MLD control	
MB_YvirtToPhiPoly5.....	409	MB_WinderDancerCtrlType01.....	511
MB_YVirtToWorkRange.....	396	Parameterization for the MLD control	
MB_YvirtToYmech.....	411	MB_DancerControlType03.....	481
MC_CAMSWITCH_REF.....	65	Parameterization, required	
MC_DigitalCamSwitch.....	49	MB_CalcInertiaLimitType02.....	561
MC_OUTPUT_REF.....	65	MB_UnwindMaterialType02.....	566
MC_TRACK_REF.....	66	ML_FlyingShear.....	178

Index

PLC project in IndraLogic	495
Position monitoring.....	220
Preset	
Winding diameter.....	493
Preset value	
Rewinder.....	534, 552
Unwinder.....	534, 552
Programmable limit switch.....	49

R

RegiMeasuring	
Error handling.....	313
Functional description.....	312
Timing diagram.....	312
Register controller	
Example punching in label printing.....	372
Use case: closed-loop insetter control.....	373
Use case: flow wrapper.....	374
Register controller function block.....	346
Register quality function block.....	314
Register setpoint lock function block.....	344
Regulating function blocks for register controllers.....	317
RIL_ParameterChannel.library.....	673
RMB_TechCam.library.....	623
RMB_TechCrosscutCrossseal.library.....	595
RMB_TechWinder.library.....	471
RMB_TechWinder/MX_TechWinder	
Closed-loop dancer position control.....	473
Diameter calculation.....	489
Diameter calculator with closed-loop tension control.....	521
Diameter calculator with dancer.....	502
Diameter calculator with open-loop tension control.....	521
Function block for the winder diameter with closed-loop tension control and speed correction.....	541
Parameter overview.....	593
Supplementary function blocks.....	558

S

S curve, programming motion sequences.....	233
Safety alert symbol.....	10
Safety instructions.....	10
Sag controller.....	225
Service hotline.....	693
Side register controller function block.....	375
Signal words.....	10
Signal-time diagram	
MB_CalInertiaLimitType02.....	560
MB_DancerControlType03.....	478, 485
MB_DiameterCalculatorType03.....	493
MB_DiameterMeasurementType01.....	501
MB_UnwindMaterialType02.....	564
MB_WIND_TAPER_PROFILE.....	568
MB_WinderTensionCtrlType02.....	529

MB_WindSpeedControlAdaptionType01.....	572
SmartBelt	
Accumulating conveyor.....	245
Calculate the ProbeSetpoint.....	255
Calculation of the SmartBeltPitch.....	253
Common mistakes and misunderstandings	259
Degree of order.....	246
Drive parameter settings.....	249
Drive requirements.....	249
Friction.....	247
PLC programming.....	252
Project planning.....	248
Requirements for belt and machine.....	248
SmartBelt limitations and behavior.....	256
Tips and tricks.....	257

Software, required

MB_CalInertiaLimitType02.....	561
MB_DancerControlType03.....	481, 487
MB_DiameterCalculatorType03.....	496
MB_DiameterMeasurementType01.....	502
MB_TraverseControlType01.....	593
MB_UnwindMaterialType02.....	566
MB_WinderDancerCtrlType01.....	511
MB_WinderDancerCtrlType02.....	519
MB_WinderTensionCtrlType02.....	536
MB_WindSpeedControlAdaptionType01.....	574
MB_WindTaperProfileType01.....	571

Special features for the IndraMotion MLC

MB_DancerControlType03.....	481, 487
MB_DiameterCalculatorType03.....	496
MB_DiameterMeasurementType01.....	502
MB_TraverseControlType01.....	593
MB_WinderTensionCtrlType02.....	536
MB_WindSpeedControlAdaptionType01.....	574
MB_WindTaperProfileType01.....	571

Special features for the IndraMotion MLD

MB_DiameterCalculatorType03.....	496
MB_DiameterMeasurementType01.....	502
MB_WinderTensionCtrlType02.....	537

Special features of the IndraMotion MLD..... 19

Splice.....	534, 551
MB_WinderDancerCtrlType01.....	508
MB_WinderDancerCtrlType02.....	517
Support.....	693
Symbols.....	11
System information.....	152

T

Temperature controller.....	66
Tension.....	532, 550
Tension controller.....	275
Tensioning	
At constant speed.....	532
At constant web velocity.....	532
Winding material.....	532, 550
Tensioning speed.....	532
Tensioning with constant web velocity.....	533

Terms and abbreviations.....	17
Torque for winding drive	
Diameter calculation.....	532, 550
Touch probe	
Components.....	98
Error codes of the MB_InitTouchProbe.....	113
Overview.....	96
Parameterization.....	98
Trigonometric functions.....	125
Two-point controller.....	41
 U	
Unwinder.....	532, 538, 549
 V	
Validity of the documentation.....	9
 W	
Warnings.....	10
Winder.....	532, 549
Winding diameter.....	489, 532, 550
Winding drive.....	489

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

Tel. +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911336324