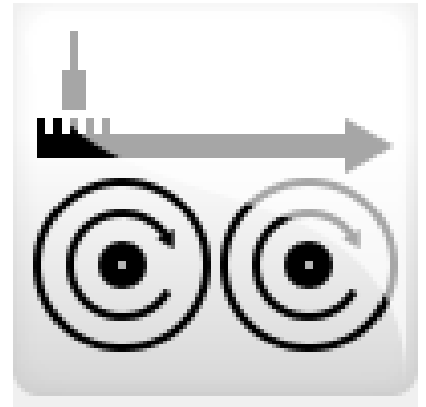


Technology module



Sync And Correction -----

Reference manual

EN

Contents

1	About this document	4
1.1	Document history	6
1.2	Conventions used	7
1.3	Layout of the safety instructions	8
2	Safety instructions	9
3	Overview of the technology module functions	11
3.1	Functional description	11
3.2	Functional overview of the TM	12
3.3	Important notes on how to operate the technology module	13
4	Application ranges	15
5	L_TS2P_SyncAndCorrectionState function block	16
5.1	Inputs	17
5.2	Outputs	19
5.3	Parameter	21
6	State machine	26
7	Signal flow diagrams	28
7.1	Structure of the signal flow	30
7.2	Structure of the access points	32
8	Description of the basic functions	33
8.1	Manual jog (jogging)	34
8.2	Homing	36
8.3	QSP application	39
8.4	Halt	40
8.5	Absolute positioning	41
8.6	Relative positioning	43
8.7	Continuous travel with a defined velocity	46
8.8	Aborting and completing a positioning process	49
8.9	Override for speed	51

Contents

9	Description of the technology functions	53
9.1	Position synchronism	53
9.1.1	Position synchronism without basic speed	53
9.1.2	Position synchronism with basic speed	56
9.2	Position coupling	58
9.2.1	Path-based coupling	59
9.2.2	Profile generator-based coupling	63
9.3	Position trimming	66
9.4	Position offset via profile generator	68
9.5	Limit setpoints to allowed travel range	70
9.6	Touch probe-based master value correction	71
9.6.1	Touch probe connection	73
9.6.2	Correction window and limitation of the master value correction	76
9.6.3	Mark window master value correction	78
9.6.4	Mark stack master value correction	81
9.6.5	Gearbox factor correction master value correction	83
9.7	Touch probe-based tool correction	86
9.7.1	Parameterize the touch probe sensor	86
9.7.2	Mark stack touch probe sensor	87
9.7.3	Tool correction window	88
9.7.4	Tool correction mark window	89
	Index	90
	Your opinion is important to us	91

1 About this document

This documentation ...

- contains detailed information on the functionalities of the "Sync And Correction" technology module;
- is part of the "Controller-based Automation" manual collection. It consists of the following sets of documentation:

Documentation type	Subject
Product catalogue	Controller-based Automation (system overview, sample topologies) Lenze controllers (product information, technical data)
System manuals	Visualisation (system overview/sample topologies)
Communication manuals Online helps	Bus systems • Controller-based Automation EtherCAT® • Controller-based Automation CANopen® • Controller-based Automation PROFIBUS® • Controller-based Automation PROFINET®
Reference manuals Online helps	Lenze controllers: • Controller 3200 C • Controller c300 • Controller p300 • Controller p500
Software manuals Online helps	Lenze engineering tools: • »PLC Designer« (programming) • »Engineer« (parameterisation, configuration, diagnostics) • »VisiWinNET® Smart« (visualisation) • »Backup & Restore« (backup, restore, update)

Further sets of technical documentation with regard to Lenze products

Further information on Lenze products which can be used in conjunction with Controller-based Automation can be found in the following sets of documentation:

Design / configuration / technical data	
☐	Product catalogues • Controller-based Automation • Controllers • Inverter Drives/Servo Drives
Mounting and wiring	
☒	Mounting instructions • Controllers • Communication cards (MC-xxx) • I/O system 1000 (EPM-Sxxx) • Inverter Drives/Servo Drives • Communication modules
☐	Hardware manuals • Inverter Drives/Servo Drives
Parameterisation / configuration / commissioning	
☐	Online help/reference manuals • Controllers • Inverter Drives/Servo Drives • I/O system 1000 (EPM-Sxxx)
☐	Online help/communication manuals • Bus systems • Communication modules
Sample applications and templates	
☐	Online help / software and reference manuals • i700 application sample • Application Samples 8400/9400 • FAST Application Template Lenze/PackML • FAST technology modules

Symbols:

-  Printed documentation
- ☐ PDF file / online help in the Lenze engineering tool



Tip!

Current sets of documentation and software updates with regard to Lenze products can be found in the Download area at:

www.lenze.com

Target group

This documentation is intended for all persons who plan, program and commission a Lenze automation system on the basis of the Lenze FAST Application Software.

1 About this document

1.1 Document history

1.1 Document history

Version			Description
4.0	04/2021	TD06	Error corrections & supplements
3.0	06/2020	TD06	Error corrections & supplements
2.0	12/2019	TD06	Error corrections & supplements
1.0	07/2019	TD29	First edition

1 About this document

1.2 Conventions used

This documentation uses the following conventions to distinguish between different types of information:

Type of information	Writing	Examples/notes
Numeric notation		
Decimal separator	Point	The decimal point is always used. For example: 1234.56
Text		
Program name	» «	»PLC Designer« ...
Variable identifiers	<i>Italics</i>	By setting <i>bEnable</i> to TRUE...
Function blocks	Bold	The L_MC1P_AxisBasicControl function block ...
Function libraries		The L_TT1P_TechnologyModules function library ...
Source code	Font "Courier new"	... dwNumerator := 1; dwDenominator := 1; ...
Symbols		
Page reference	 7)	Reference to further information: Page number in PDF file.

Variable identifiers

The conventions used by Lenze for the variable names of Lenze system blocks, function blocks, and functions are based on the "Hungarian Notation". This notation makes it possible to identify the most important properties (e.g. the data type) of the corresponding variable by means of its name, e.g. xAxisEnabled.

1 About this document

1.3 Layout of the safety instructions

DANGER!

Indicates an extremely hazardous situation. Failure to comply with this instruction will result in severe irreparable injury and even death.

WARNING!

Indicates an extremely hazardous situation. Failure to comply with this instruction may result in severe irreparable injury and even death.

CAUTION!

Indicates a hazardous situation. Failure to comply with this instruction may result in slight to medium injury.

NOTICE

Indicates a material hazard. Failure to comply with this instruction may result in material damage.

Application notes

Pictograph	Signal word	Meaning
	Tip!	Useful tip for easy handling
		Reference to another document

2 Safety instructions

Please observe the safety instructions in this documentation when you want to commission an automation system or a plant with a Lenze Controller.



The device documentation contains safety instructions which must be observed!

Read the documentation supplied with the components of the automation system carefully before you start commissioning the Controller and the connected devices.

DANGER!

High electrical voltage

Injury to persons caused by dangerous electrical voltage

Possible consequences

Death or severe injuries

Protective measures

Switch off the voltage supply before working on the components of the automation system.

After switching off the voltage supply, do not touch live device parts and power terminals immediately because capacitors may be charged.

Observe the corresponding information plates on the device.

DANGER!

Injury to persons

Risk of injury is caused by ...

- unpredictable motor movements (e.g. unintended direction of rotation, too high velocities or jerky movement);
- impermissible operating states during the parameterisation while there is an active online connection to the device.

Possible consequences

Death or severe injuries

Protective measures

- If required, provide systems with installed inverters with additional monitoring and protective devices according to the safety regulations valid in each case (e.g. law on technical equipment, regulations for the prevention of accidents).
- During commissioning, maintain an adequate safety distance to the motor or the machine parts driven by the motor.

NOTICE

Damage or destruction of machine parts

Damage or destruction of machine parts can be caused by ...

- Short circuit or static discharges (ESD);
- unpredictable motor movements (e.g. unintended direction of rotation, too high velocities or jerky movement);
- impermissible operating states during the parameterisation while there is an active online connection to the device.

Protective measures

- Always switch off the voltage supply before working on the components of the automation system.
 - Do not touch electronic components and contacts unless ESD measures were taken beforehand.
 - If required, provide systems with installed inverters with additional monitoring and protective devices according to the safety regulations valid in each case (e.g. law on technical equipment, regulations for the prevention of accidents).
-

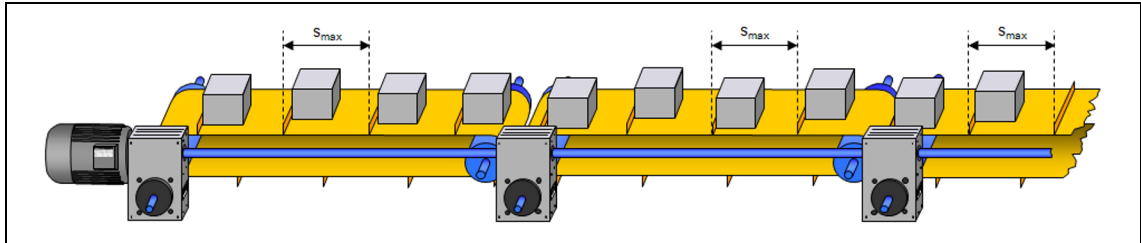
3 Overview of the technology module functions

3.1 Functional description

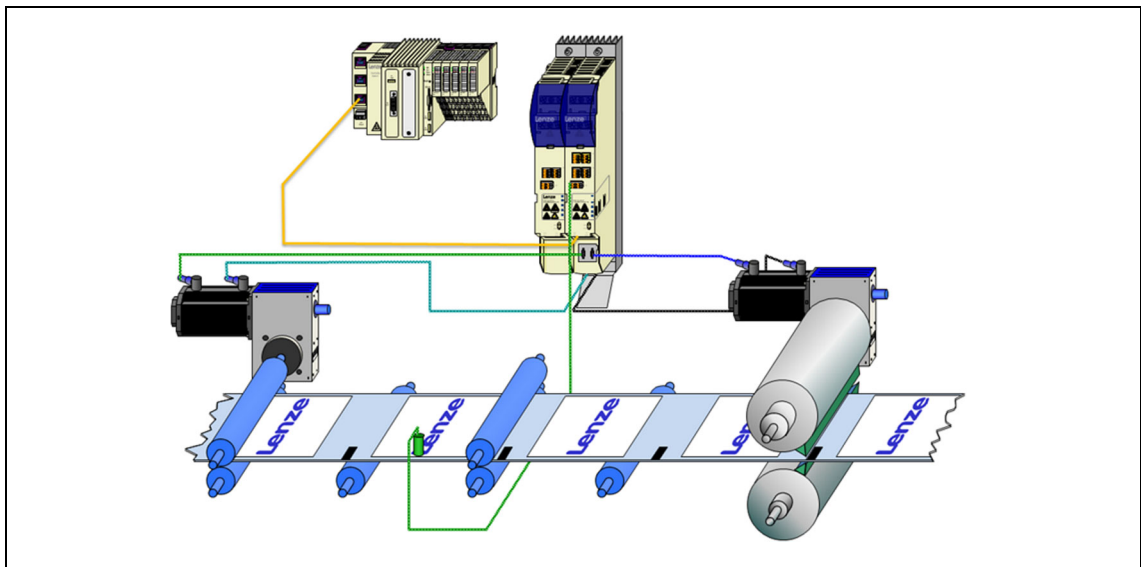
3 Overview of the technology module functions

3.1 Functional description

This application provides standardized functions to synchronize axes.



[3-1] Typical mechanics of the technology module in Sync operation



[3-2] Typical mechanics of the technology module in SyncAndCorrection operation

- A slave axis follows the master axis in position synchronism. This means that a cycle of the master will initiate a cycle of the slave. The synchronism is implemented without a position jump.
- The slave axis can be declutched at a certain position and then clutched in again without a step change.
- An offset between the master axis and slave axis can be set via trimming – as with manual jog – or as an absolute offset value. The offset setpoint is superimposed on the main setpoint via a profile generation.
- As a higher-level control loop, the integrated optional master value correction (register controller) controls the master position relative to a mark detected on the material. This serves to compensate process-related material mark position deviations with regard to the master position.
- The integrated optional tool correction corrects tool deviations resulting from slip or an imprecise feed constant.

3.2 Functional overview of the TM

Functionality	Description
Basic Motion Base (V2):	
Manual jog (jogging) (📖 34)	Traverse axis in manual operation
Homing (📖 36)	Reference the axis
OSP application (📖 39)	Stop motion via an application quick stop deceleration ramp (AxisRef)
Halt (📖 40)	Stop motion via a stop deceleration ramp
Relative positioning (📖 43)	Move axis to a target position relative to the current position
Continuous travel with a defined velocity (📖 46)	Execute continuous travel
Aborting and completing a positioning process (📖 49)	End the positioning
Override for speed (📖 51)	Specify overrides for speed
Sync And Correction State (V2):	
Position synchronism (📖 53)	Establish synchronism between a master axis and a drive axis (slave) • without basic speed • with basic speed
Position coupling (📖 58)	Synchronize setpoint position of the drive (slave axis) to the position setpoint of the master value (master axis) • Path-based coupling • Profile generator-based coupling
Position trimming (📖 66)	Adjust position of the slave axis in relation to the master axis
Position offset via profile generator (📖 68)	Specify position offset via a profile generator
Limit setpoints to allowed travel range (📖 70)	Block profile generation in the unauthorized direction.
Touch probe-based master value correction (📖 71)	Correct master position relative to a mark detected on the material (touch probe)
Touch probe-based tool correction (📖 86)	Homing "on the fly" for position followers

Furthermore the technology module can be used for status and error messages and for resetting an error.

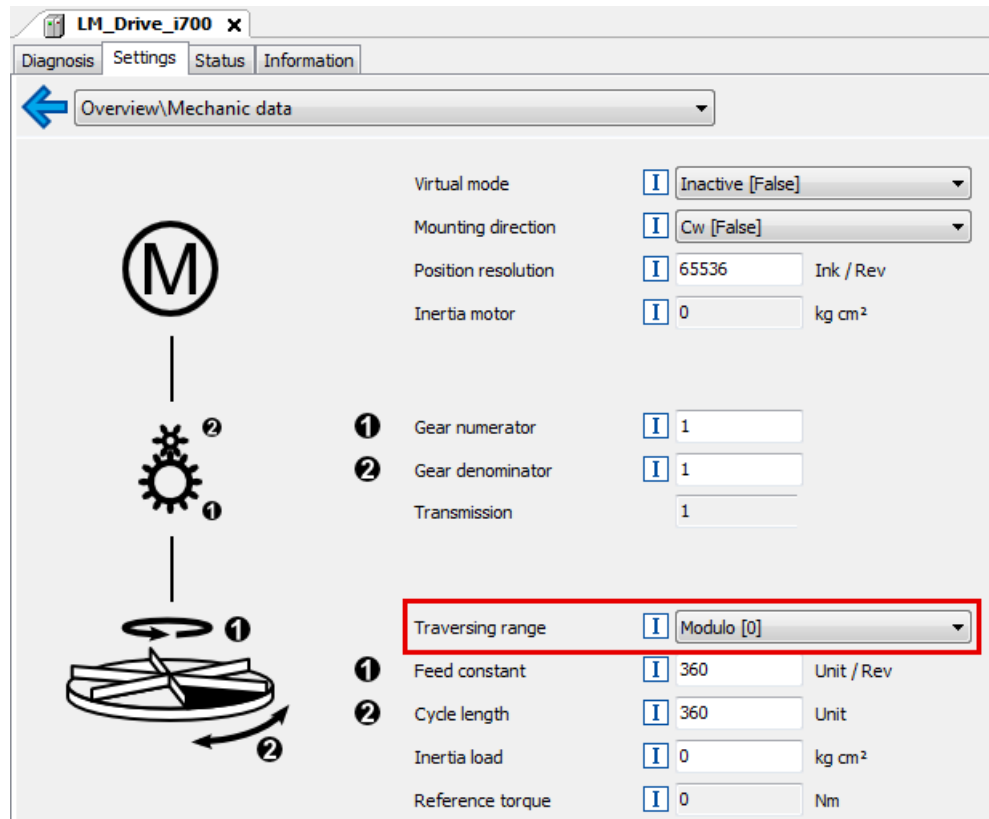
3 Overview of the technology module functions

3.3 Important notes on how to operate the technology module

3.3 Important notes on how to operate the technology module

The technology module supports "modulo" and "limited" axes. It must be ensured that both axes are always parameterized with the same traversing range.

Go to the »PLC Designer« and set the following parameters for each axis under the **Settings** tab:



Start of the axes

The start of the axes after the controller has been inhibited depends on the requested movement type.

A distinction is made between the movement type with level-activated function and the movement type with edge-activated function.

- The movement type with level-activated function is automatically activated when the controller is enabled, as long as no higher priority function is active. This movement type includes
 - [Manual jog \(jogging\)](#) (📖 34)
 - [QSP application](#) (📖 39)
- The movement type with edge-activated function must be activated with a FALSE↗TRUE edge after the controller has been enabled. This movement type includes
 - [Homing](#) (📖 36)



Example for a movement type with level-activated function: manual jog (jogging)

1. In the inhibited axis state ($xOperationEnabled = FALSE$), $xJogPos$ is set to TRUE.
 - $xEnableOperation = FALSE$ (axis is inhibited.)
==> "STANDBY" state ($xOperationEnabled = FALSE$)
 - $xJogPos = TRUE$ (manual jog is to be executed.)
2. Enable axis.
 - $xEnableOperation = TRUE$
==> "JOGGING" state ($xOperationEnabled = TRUE$)



Example for a movement type with edge-activated function: homing

1. In the inhibited axis state ($xOperationEnabled = FALSE$), $xHomeExecute$ is set to TRUE.
 - $xEnableOperation = FALSE$ (axis is inhibited.)
==> "STANDBY" state ($xEnableOperation = FALSE$)
 - $xHomeExecute = TRUE$ (homing is to be executed.)
2. Enable axis.
 - $xEnableOperation = TRUE$
==> "STANDBY" state (output $xOperationEnabled = TRUE$)
3. Execute homing.
 - $xHomeExecute = FALSE↗TRUE$
==> "HOMING" state

4 Application ranges

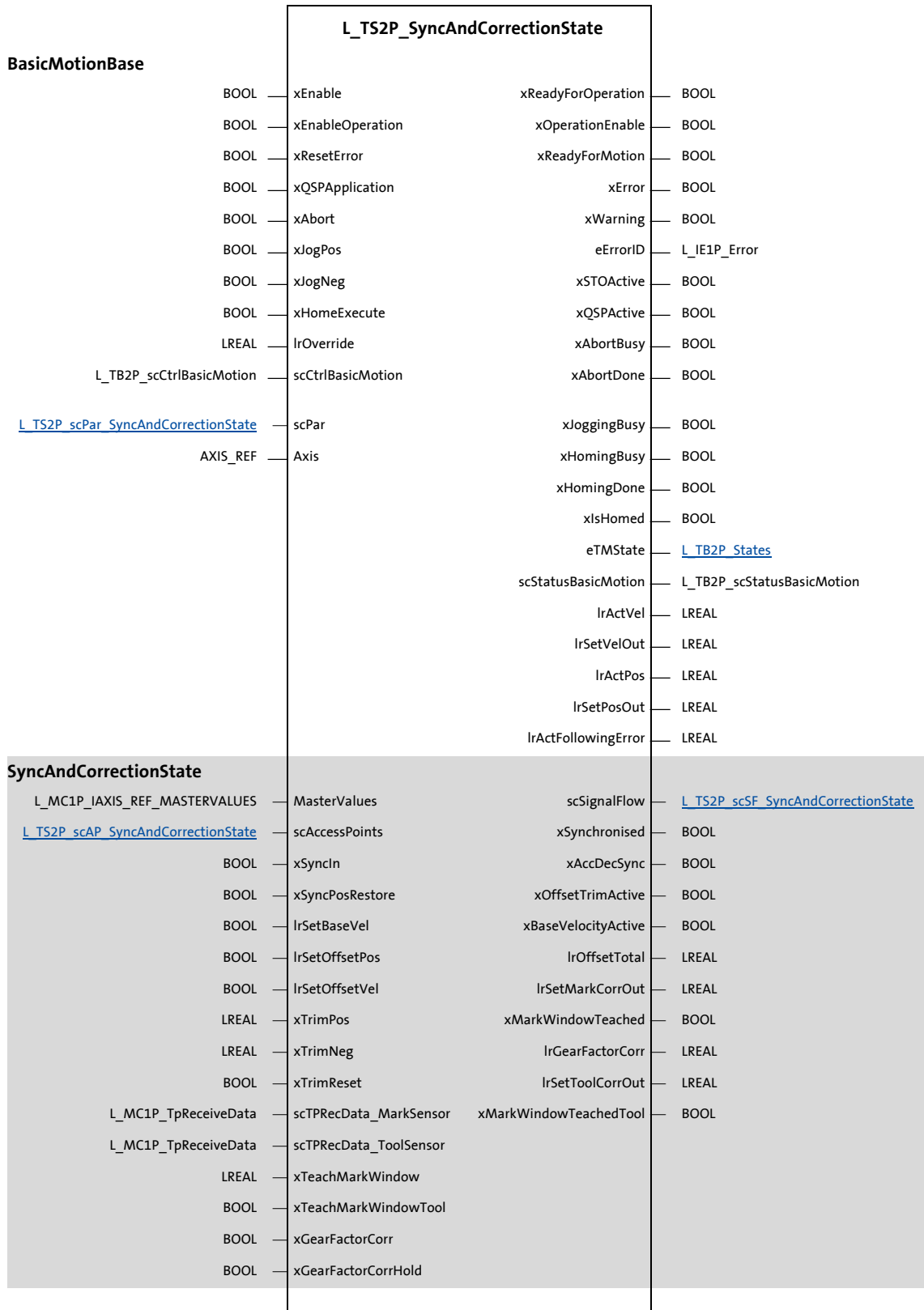
Typical applications of the technology module:

- Printing units
- Perforating machines
- Insetters
- Vibrating drives
- Line drives
- Labelling machines

5 L_TS2P_SyncAndCorrectionState function block

5 L_TS2P_SyncAndCorrectionState function block

The figure shows the affiliation of the inputs and outputs of the function block.



5.1

Inputs

Designator	Data type	Description
xEnable	BOOL	Execution of the function block
		TRUE The function block is executed.
		FALSE The function block is not executed.
xEnableOperation	BOOL	TRUE The axis is switched to ready for operation. Controller enable is issued.
		FALSE The axis is not ready for operation.
xResetError	BOOL	The input is edge-controlled and evaluates the rising edge.
		FALSE Reset axis error or software error.
		TRUE The axis error or software error is reset.
xQSPApplication	BOOL	The input is level-controlled and evaluates the HIGH level.
		TRUE Active motion is aborted and axis is brought to a standstill via the QSP application deceleration ramp (39). - The state changes to "STOP". - The technology module remains in the "STOP" state as long as <i>xQSPApplication</i> = TRUE.
xAbort	BOOL	TRUE Active motion is aborted and axis is brought to a standstill via the active profile deceleration. The state changes to "STOP". The deceleration ramp can be aborted by activating a new motion (e.g. <i>xMoveVelExecute</i>)
		FALSE The active motion is not aborted.
xJogPos	BOOL	The input is level-controlled and evaluates the HIGH level.
		TRUE Traverse axis in positive direction (manual jog). If <i>xJogNeg</i> is also TRUE, the InvalidInputJogPosJogNeg error message is output via the <i>eErrorID</i> output and the axis stops.
xJogNeg	BOOL	The input is level-controlled and evaluates the HIGH level.
		TRUE Traverse axis in negative direction (manual jog). If <i>xJogPos</i> is also TRUE, the InvalidInputJogPosJogNeg error message is output via the <i>eErrorID</i> output and the axis stops.
xHomeExecute	BOOL	The input is edge-controlled and evaluates the rising edge.
		FALSE Start homing.
		TRUE The function is aborted via the <i>scCtrlBasicMotion.xHalt</i> structure input.
lrOverride	LREAL	Using the <i>lrOverride</i> factor, the specified setpoint speed is multiplied (e.g. an override value of 0.5 generates half the setpoint speed) - Value range: 0 ... 1.0 - Initial value: 1.0 Note: The acceptance of the value is independent of the status in <i>xContinuousUpdate</i>
scCtrlBasicMotion L_TB2P_scCtrlBasicMotion		Structure of the status data You can find information on status data here: "Basic Motion Base V2" reference manual
scPar L_TS2P_scPar_SyncAndCorrectionState		The parameter structure contains the parameters of the technology module.
Axis AXIS_REF		Reference to the axis
MasterValues L_MC1P_IAXIS_REF_MASTervalues		Connection for the master values (master axis) The following types of axes can be connected to this input: - AXIS_REF, - AXIS_REF_i900 - AXIS_REF_ENC_i900_Drive, - AXIS_REF_VIRTUAL - AXIS_REF_VIRTUALM_i900

Designator	Data type	Description
scAccessPoints L_TS2P_scAP_SyncAndCorrectionState		Structure of the access points The access points can be used to influence the signal flow or technology module functions.
xSyncIn	BOOL	Synchronize slave axis to the master values of the master axis. ► Position coupling (58) • Initial value: FALSE The input is level-controlled and evaluates the HIGH level.
		TRUE Execute synchronization with position coupling.
xSyncPosRestore	BOOL	The input is edge-controlled and evaluates the rising edge.
		FALSE An offset built up via relative coupling (<i>scPar.xSyncAsynchronous</i> = TRUE) is compensated. TRUE
lrSetBaseVel	LREAL	Basic velocity If the clutch is open, the axis rotates at the speed specified here. • Initial value: • Unit: units/s
lrSetOffsetPos	LREAL	Position offset to the master axis • Initial value: 0 • Unit: units
lrSetOffsetVel	LREAL	Velocity offset to the master axis • Initial value: 0 • Unit: units/s
xTrimPos	BOOL	Trimming in positive direction. • Initial value: FALSE
		TRUE If eTrimMode = 0 (trimming per velocity offset), the input is level-controlled and evaluates the high level.
		FALSE If eTrimMode = 1 (trimming per position offset), the input is edge-controlled and evaluates the rising edge. TRUE
xTrimNeg	BOOL	Trimming in negative direction. • Initial value: FALSE
		TRUE If eTrimMode = 0 (trimming per velocity offset), the input is level-controlled and evaluates the high level.
		FALSE If eTrimMode = 1 (trimming per position offset), the input is edge-controlled and evaluates the rising edge. TRUE
xTrimReset	BOOL	The input is edge-controlled and evaluates the rising edge.
		FALSE Reset position offset TRUE The position offset resulting from the trimming is moved to zero via the profile generator.
scTPRecDataMarkSensor L_MC1P_TpReceiveData		Touch probe structure input for the master value correction
scTPRecData_ToolSensor L_MC1P_TpReceiveData		Touch probe structure input for the tool correction
xTeachMarkWindow	BOOL	TRUE Teach mark window of the master value correction
xTeachMarkWindowTool	BOOL	TRUE Teach mark window of the tool correction
xGearFactorCorr	BOOL	TRUE Activation of the gearbox factor correction
xGearFactorCorrHold	BOOL	TRUE Maintain current gearbox factor correction value

5.2

Outputs

Designator	Data type	Description
xReadyForOperation	BOOL	The axis is in correct working order. The DC-bus voltage is available. "Safe Torque Off" (STO) is not active.
xOperationEnabled	BOOL	The axis is enabled. The brake is released.
xReadyForMotion	BOOL	The axis is enabled and in correct working order. The brake is released.
xError	BOOL	TRUE There is an error in the technology module.
xWarning	BOOL	TRUE There is a warning in the technology module.
eErrorID	L_IE1P_Error	ID of the error or warning message if <i>xError</i> = TRUE or <i>xWarning</i> = TRUE. "FAST technology modules" reference manual: Here you can find information on error or warning messages.
xSTOActive	BOOL	TRUE "Safe Torque Off" (STO) was triggered and is active. The axis is free of torque, the controller is inhibited.
xQSPActive	BOOL	TRUE The <i>xQSPApplication</i> input was set. The technology module is in the "STOP" state; the axis is brought to a standstill. The output is reset as soon as the axis is at standstill.
xAbortBusy	BOOL	TRUE The activated function is aborted via the <i>xAbsort</i> input.
xAbortDone	BOOL	TRUE The activated function has been aborted via the <i>xAbsort</i> input.
xJoggingBusy	BOOL	TRUE The Manual jog (jogging) (📖 34) function is active. The technology module is in the "JOGGING" state.
xHomingBusy	BOOL	TRUE The Homing (📖 36) is executed. The technology module is in the "HOMING" state.
xHomingDone	BOOL	TRUE Homing has been successfully completed.
xIsHomed	BOOL	TRUE The axis has been referenced ("Reference known"). <i>xIsHomed</i> may only be evaluated when the <i>scStatusBasicMotion.xAxisAvailable</i> status is reported.
eTMState	L_TB2P_States	Current state of the technology module
scStatusBasicMotion	L_TB2P_scStatusBasicMotion	Structure of the status data You can find information on status data here: "Basic Motion Base V2" reference manual
IrActVel	LREAL	Current speed • Unit: units/s
IrSetVelOut	LREAL	Setpoint speed • Unit: units/s
IrActPos	LREAL	Current position • Unit: unit
IrSetPosOut	LREAL	Setpoint position • Unit: unit
IrActFollowingError	LREAL	Current following error • Unit: unit
scSignalflow	L_TS2P_scSF_SyncAndCorrectionState	Structure of the signal flow

Designator	Data type	Description	
xSynchronised	BOOL	TRUE	The axis is coupled with respect to the master axis. The technology module is in the "SYNCHRONIZED" state.
xAccDecSync	BOOL	TRUE	The synchronization function is active. The axis is synchronized or desynchronized (clutch opens or closes). The technology module is in the "ACC_DEC_SYNC" state.
xOffsetTrimActive	BOOL	TRUE	An offset function (position offset, velocity offset) and/or the trimming function is active.
xBaseVelocityActive	BOOL	TRUE	The clutch is open and the axis moves at the speed set at the <i>IrSetBaseVel</i> input. The technology module is in the "BASEVEL" state.
IrOffsetTotal	LREAL	Total position offset	
IrSetMarkCorrOut	LREAL	Current mark error limited to the maximum value	
xMarkWindowTeached	BOOL	TRUE	Mark window of the master value correction taught
IrGearFactorCorr	LREAL	Gearbox factor correction value	
IrSetToolCorrOut	LREAL	Current position error of the tool correction limited to the maximum value	
xMarkWindowTeachedTool	BOOL	TRUE	Mark window of the tool correction taught

5.3

Parameter

L_TS2P_scPar_SyncAndCorrectionState

Designator	Data type	Description
IrVel	LREAL	Speed for motions activated via the scCtrlBasicMotion structure. You can find information on status data here: "Basic Motion Base" reference manual. - Unit: units/s - Initial value: 3600
IrAcc	LREAL	Acceleration for motions activated via the scCtrlBasicMotion structure. Specification of the maximum velocity variation which is to be used for acceleration. - Unit: units/s² - Initial value: 36000
IrDec	LREAL	Deceleration for motions activated via the scCtrlBasicMotion structure. Specification of the maximum speed variation which is to be used for deceleration to standstill. - Unit: units/s² - Initial value: 100
IrJerk	LREAL	Jerk for motions activated via the scCtrlBasicMotion structure. - Unit: units/s³ - Initial value: 0
eDirection	MC_DIRECTION	Traversing direction: <i>mcCurrentDirection</i>: Retain current direction (only in case of modulo axis) <i>mcPositiveDirection</i>: Positive direction <i>mcNegativeDirection</i>: Negative direction <i>mcShortestWay</i>: Shortest distance (only in case of modulo axis) - Initial value: <i>mcPositiveDirection</i>
scHomeExtTP	MC_TRIGGER_REF	Transfer of an external touch probe event: - Only relevant for "external encoder" touch probe configuration. - For describing the MC_TRIGGER_REF structure, see the MC_TouchProbe function block.
dwGearingNumerator	DWORD	Gearbox numerator Parameter changes are accepted immediately. - Unit: - - Initial value: 1
dwGearingDenominator	DWORD	Gearbox factor denominator Parameter changes are accepted immediately. - Unit: - - Initial value: 1
IrSyncAcc	LREAL	Acceleration for the clutch-in/declutch process at <i>eSyncMode</i> = 0 (ProfGenAbsolute) or 1 (ProfGenRelative). Specification of the speed variation which is to be used for maximum acceleration. - Unit: units/s² - Initial value: 1000
IrSyncDec	LREAL	Deceleration for the clutch-in/declutch process at <i>eSyncMode</i> = 0 (ProfGenAbsolute) or 1 (ProfGenRelative). Specification of the speed variation which is to be used for maximum deceleration. - Unit: units/s² - Initial value: 1000
IrSyncJerk	LREAL	Jerk for the clutch-in/declutch process at <i>eSyncMode</i> = 0 (ProfGenAbsolute) or 1 (ProfGenRelative). Specification of the maximum acceleration/deceleration variation which is to be used for the clutch-in/declutch process. - Unit: units/s³ - Initial value: 1 000 000

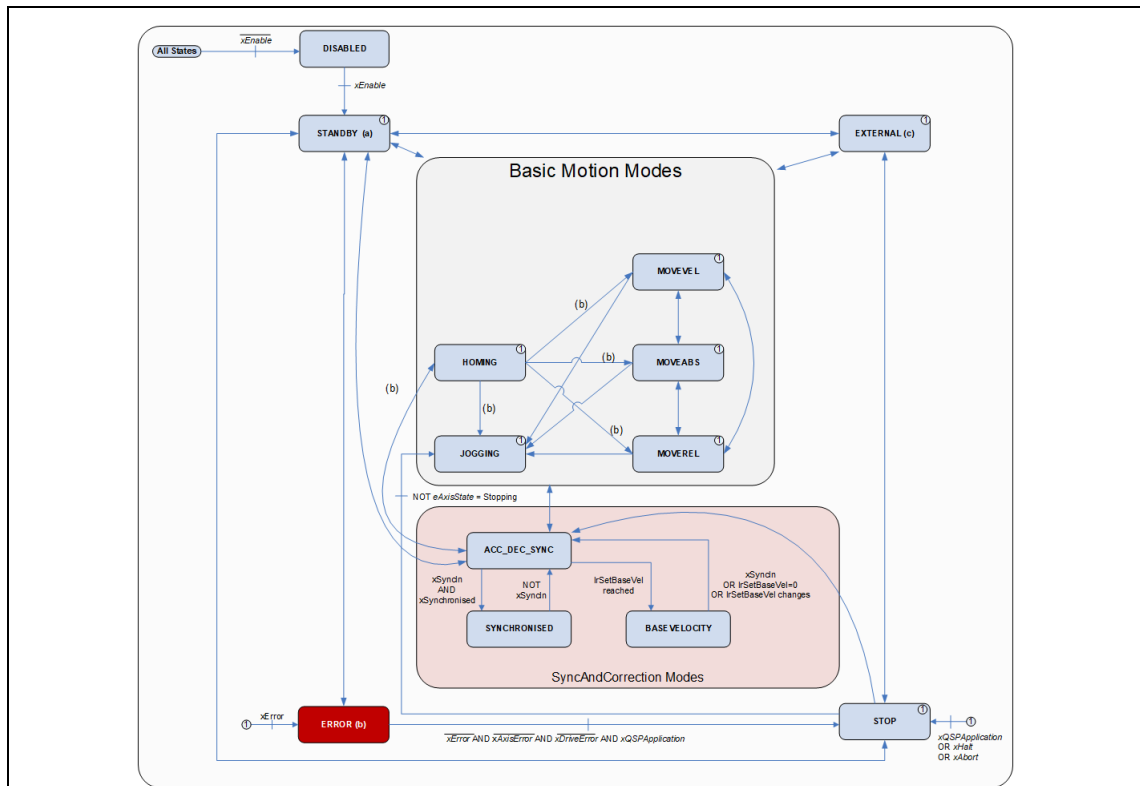
Designator	Data type	Description
eAllowedDirection L_TB2P_AllowedDirection		Allowed direction of rotation of the effective overall setpoint (master speed + offset speed + clutch-in speed) for the synchronous function. • Initial value: positive and negative directions (0)
	0	Positive and negative direction of rotation The slave axis may engage in both directions. In the case of offset or trim, it is allowed to move in both directions.
	1	Only positive direction of rotation The slave axis may only engage in positive direction. An offset may only be moved in positive direction. A negative offset is only moved to the extent that the speed of the total conductance does not become less than '0'.
	2	Only positive direction of rotation The slave axis may only engage in negative direction. An offset may only be moved in negative direction. A positive offset is only moved to the extent that the speed of the total conductance does not exceed '0'.
xLimitAllowedDirection	BOOL	Activate total setpoint limitation to the allowed range (eAllowedDirection).
	TRUE	The speed of the total setpoint is limited to the allowed range (eAllowedDirection). For example, a negative total speed setpoint is not possible with eAllowedDirection = 1 (only positive direction of rotation).
	FALSE	The speed of the total setpoint can be in a non-allowed range if the conductivity speed is negative.
IrToleranceAllowedDirection	LREAL	Maximum setpoint difference in a blocked direction (eAllowedDirection) before an error is triggered. This tolerance window is only effective if xLimitAllowedDirection = TRUE. • Unit: unit • Initial value: 5
IrSetOffsetTrimVel	LREAL	Speed for the offset and trimming functions Specification of the maximum speed for the additional profile of the offset and trimming functions. • Unit: units/s • Initial value: 10
IrSetOffsetTrimAcc	LREAL	Acceleration for the offset and trimming functions Specification of the speed variation relative to the master axis which is to be used for acceleration. The acceleration acting on the drive is the sum of the master and slave accelerations. • Unit: units/s ² • Initial value: 100
IrSetOffsetTrimDec	LREAL	Deceleration for the offset and trimming functions Specification of the speed variation relative to the master axis which is to be used for deceleration. The deceleration acting on the drive is the sum of the master and slave decelerations. • Unit: units/s ² • Initial value: 100
IrSetOffsetTrimJerk	LREAL	Jerk for the offset and trimming functions • Unit: units/s ³ • Initial value: 100 000
xLoadOffset	BOOL	Loading of the position offset to the master axis (via IrSetOffsetPos input). • Initial value: TRUE
	TRUE	The position offset is loaded cyclically. The profile generator is deactivated.
	FALSE	The position offset is moved via the profile generator.

Designator	Data type	Description
eTrimMode L_TS2P_TrimMode		Setting of the trimming mode. Trimming can be triggered via the inputs <i>xTrimPos</i> or <i>xTrimNeg</i> = TRUE. • Initial value: 1
	0	Velocity (trimming per velocity offset)
	1	Position (trimming per position offset <i>scPar.IrTrimDist</i>)
IrTrimDist	LREAL	Setting of the position offset to be trimmed. The position offset set here is trimmed in the positive/negative direction for each rising edge at the <i>xTrimPos</i> / <i>xTrimNeg</i> input. • Only effective if <i>eTrimMode</i> = 1 • Unit: unit • Initial value: 1
eSyncMode L_TB2P_SyncMode		Coupling mode • Initial value: 0 (ProfGenAbsolute)
	0	ProfGenAbsolute
	1	ProfGenRelative
	2	DistAbsolute
	3	DistRelative
IrSlaveSyncOutPos	LREAL	Declutch position • Unit: unit • Initial value: 0
IrSlaveSyncInDist	LREAL	Clutch-in distance if <i>eSyncMode</i> = 2 (DistAbsolute) or 3 (DistRelative) • Unit: unit • Initial value: 90
IrSlaveSyncOutDist	LREAL	Declutch distance if <i>eSyncMode</i> = 2 (DistAbsolute) or 3 (DistRelative) • Unit: unit • Initial value: 90
IrSyncVel	LREAL	Additional speed value if <i>eSyncMode</i> = 0 (ProfGenAbsolute) or 1 (ProfGenRelative). This parameter defines the allowed offset speed for the under- or oversynchronous clutch-in. • Unit: units/s • Initial value: 100
xCalcMarkErrorMaster	BOOL	Activation of the master value correction or the register control. • Initial value: FALSE
	TRUE	Master value correction activated
	FALSE	Master value correction deactivated
xCorrMarkErrorMaster	BOOL	Enabling to correct the calculated mark error. • Initial value: FALSE
	TRUE	Correction movement on
	FALSE	Correction movement off
IrSensorToolDistMaster	LREAL	Distance between the touch probe sensor and the position where the tool (e.g. cutting blade, printheads) engages the material. • Unit: unit • Initial value: 0
IrMarkDistMaster	LREAL	Mark distance (register length) in master value units • Unit: unit • Initial value: 360
IrMaxCorrPosMaster	LREAL	Maximum positive correction value per register cycle • Unit: unit • Initial value: 30
IrMaxCorrNegMaster	LREAL	Maximum negative correction value per register cycle • Unit: unit • Initial value: -30

Designator	Data type	Description
eSourceForCorrWindow L_TS2P_SourceForCorrWindow		Selection of the measuring system to which the correction window will be applied. • Initial value: 4 (tool position)
	1	Corrected master position
	2	Tool position
lrUpperCorrPosMaster LREAL		Upper limit value of the correction window for the compensating movement of the mark correction. • Unit: unit • Initial value: 180
lrLowerCorrPosMaster LREAL		Lower limit value of the correction window for the compensating movement of the mark correction. • Unit: unit • Initial value: 90
xEnableMarkWindowMaster BOOL		Activation of the mark window for the master value. • Initial value: FALSE
	TRUE	Mark window activated
	FALSE	Mark window deactivated
lrMarkWindowSizeMaster LREAL		Size of the mark window in master value units. The mark window is placed symmetrically around the expected mark position. • Unit: unit • Initial value: 90
lrSetOffsetMarkWindowMaster LREAL		Offset to shift the mark window in master value units. • Unit: unit • Initial value: 0
lrGearFactorCorrGainMaster LREAL		Gain factor of the gearbox factor correction. Caution: If the selected gain is too high, the setpoint speed may build up. • Unit: - • Initial value: 0.1
lrMaxGearFactorCorrMaster LREAL		Maximum deviation of the gearbox factor correction in master value units. • Unit: unit • Initial value: 10
dwMaxNumberVirtualMarksMaster DWORD		Maximum number of allowed mark failures If no marks are detected within the mark window, a virtual mark is generated at the optimum position. This happens as long as the number of marks set here is not exceeded. If the number of maximum allowed mark failures is exceeded, an error is triggered. • Unit: - • Initial value: 5
xCalcToolError BOOL		Tool correction mode • Initial value: FALSE
	TRUE	Synchronism with tool correction
	FALSE	Synchronism without tool correction
xCorrToolError BOOL		Enabling to correct the calculated tool error. • Initial value: FALSE
	TRUE	Correction movement activated
	FALSE	Correction movement deactivated
lrSensorToolDistTool LREAL		Distance between the tool sensor and the position where the tool (e.g. cutting blade, printheads) engages the material in drive axis units. • Unit: unit • Initial value: 0

Designator	Data type	Description
xEnableMarkWindowTool	BOOL	Activation of the mark window for the tool correction. • Initial value: FALSE
		TRUE Mark window activated
		FALSE Mark window deactivated
lrMaxCorrPosTool	LREAL	Maximum positive correction distance per tool revolution. • Unit: unit • Initial value: 30
lrMaxCorrNegTool	LREAL	Maximum negative correction distance per tool revolution. • Unit: unit • Initial value: -30
lrUpperCorrPosTool	LREAL	Upper limit value of the correction window for the compensating movement of the tool correction. • Unit: unit • Initial value: 180
lrLowerCorrPosTool	LREAL	Lower limit value of the correction window for the compensating movement of the tool correction. • Unit: unit • Initial value: 90
lrMarkWindowSizeTool	LREAL	Size of the mark window for the tool correction in drive axis units. The mark window is placed symmetrically around the expected mark position. • Unit: unit • Initial value: 90
lrSetOffsetMarkWindowTool	LREAL	Offset to shift the tool correction mark window in drive axis units. • Unit: unit • Initial value: 0
dwMaxNumberVirtual-MarksTool	DWORD	Maximum number of allowed mark failures for the tool correction. If no marks are detected within the mark window, a virtual mark is generated at the optimum position. This happens as long as the number of marks set here is not exceeded. If the number of maximum allowed mark failures is exceeded, an error is triggered. • Unit: - • Initial value: 5
xLoadRegPos	BOOL	TRUE The register position is loaded with the value applied at <i>scPar.lrLoadRegPos</i> .
lrLoadRegPos	LREAL	The register position is loaded with this value if <i>scPar.xLoadRegPos</i> = TRUE. • Unit: Master value unit • Initial value: 0

6 State machine



① A transition to the "ERROR" or "STOP" state is possible.

(a) In the "STANDBY" state, *xEnableOperation* has to be set to TRUE.

(b) In the "ERROR" state, *xResetError* has to be set to TRUE to acknowledge and reset the errors.

(c) The "EXTERNAL" state is set if a function generates setpoints for the connected axis outside the technology module, e.g. **MC_CamIn**, **MC_MoveAbsolute**.

(d) *eAxisState* = "STANDSTILL"

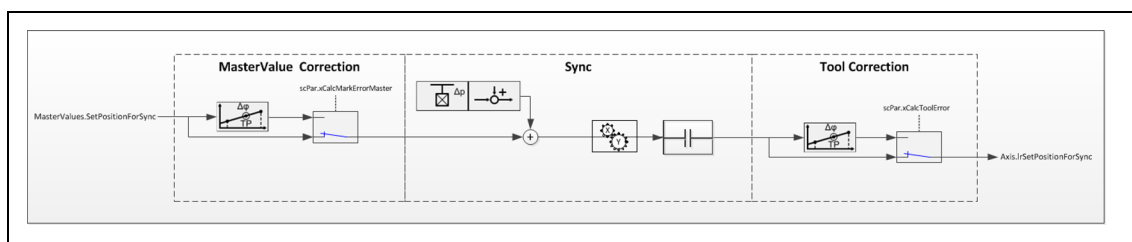
[6-1] State machine of the technology module

States of the eTMState output (L_TB2P_States)

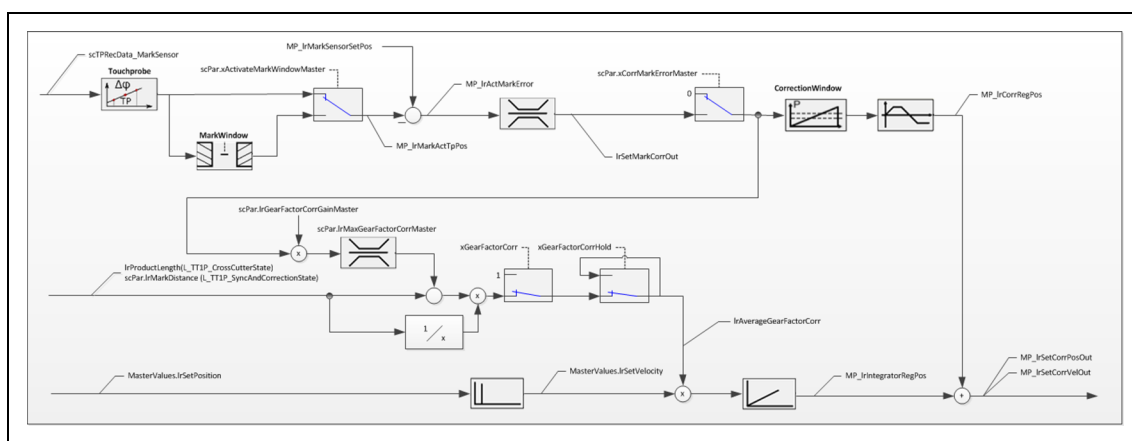
No.	L_TB2P_States	Description
-1	DISABLED	Technology module deactivated
0	STANDBY	Technology module is on standby
1	STOP	Stop/Halt active
2	HOMING	Homing active
3	JOGGING	Manual jog active
4	MOVEVEL	Continuous travel with a defined velocity active
5	MOVEABS	Absolute positioning active
6	MOVEREL	Relative positioning active
7	EXTERNAL	An external function generates setpoints for the connected axis
8	ERROR	Error state
30	ACC_DEC_SYNC	Synchronization function active
31	SYNCHRONIZED	Axes are synchronized
51	BASEVELOCITY	Drive rotates at basic speed
1000	SYSTEMFAULT	System fault

7

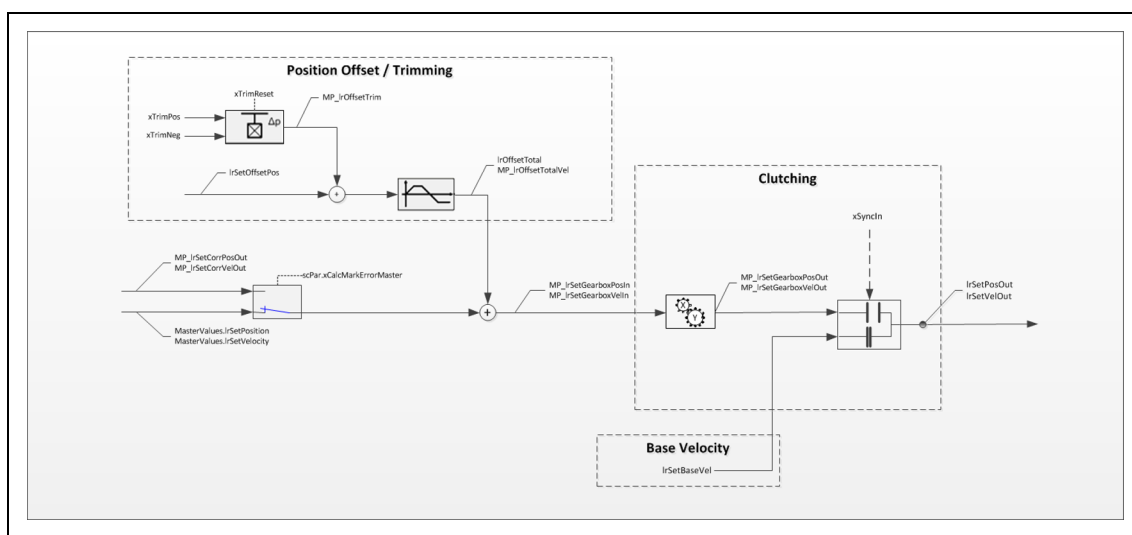
7



[7-1] Signal flow overview

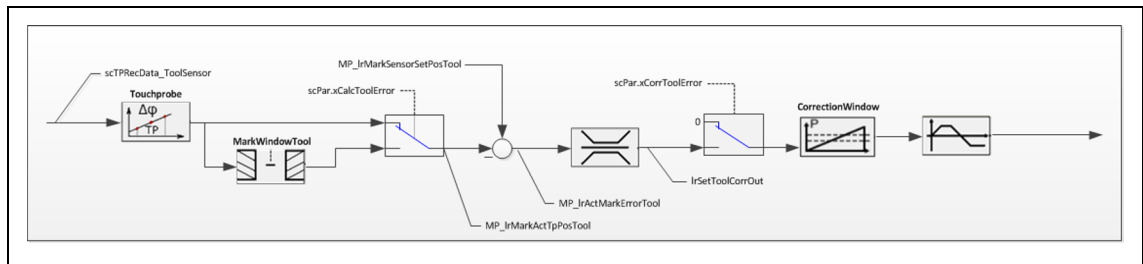


[7-2] Signal flow master value correction



[7-3] Signal flow synchronism (sync)

7 Signal flow diagrams



[7-4] Signal flow tool correction

7.1 Structure of the signal flow

L_TS2P_scSF_SyncAndCorrectionState

The contents of the structure are read-only and offer a practical diagnostics option within the signal flow.

► [Signal flow diagrams](#) (28)

Designator	Data type	Description
MP_lIrIntegratorRegPos	LREAL	Master speed, integrated
MP_lIrCorrRegPos	LREAL	Correction movement
MP_lIrMarkSensorSetPos	LREAL	Position of the mark sensor within the mark cycle length.
MP_lIrMarkActTpPos	LREAL	Measured mark position within the mark cycle length.
MP_xMarkReceived	BOOL	TRUE Mark occurred.
MP_lIrActMarkError	LREAL	Current unlimited mark error.
MP_lIrActMarkDist	LREAL	Distance between the last two occurring marks.
MP_xCorrActive	BOOL	TRUE The setpoint is within the master value correction window.
MP_xCorrLimited	BOOL	TRUE The correction of the master value is limited to the maximum or minimum correction value.
MP_xMarkWindowOpen	BOOL	TRUE The mark window of the master value correction is open.
MP_xMarkOutOfWindow	BOOL	TRUE A mark has occurred outside the mark window.
MP_xMarkNotDetected	BOOL	TRUE No mark has been detected in the mark window.
MP_lIrSetCorrPosOut	LREAL	Mark-corrected master position at the master value correction output.
MP_lIrSetCorrVelOut	LREAL	Mark-corrected master speed at the master value correction output.
MP_lIrSetMasterPos	LREAL	Setpoint position of the master value axis.
MP_lIrSetMasterVel	LREAL	Setpoint speed of the master value axis.
MP_lIrSetGearboxPosIn	LREAL	Input position of the measuring system adaptation.
MP_lIrSetGearboxVelIn	LREAL	Input speed of the measuring system adaptation.
MP_lIrSetGearboxPosOut	LREAL	Output position of the measuring system adaptation.
MP_lIrSetGearboxVelOut	LREAL	Output speed of the measuring system adaptation.
MP_lIrSetClutchPosIn	LREAL	Input position of the clutch.
MP_lIrOffset	LREAL	Position offset

Designator	Data type	Description	
MP_lOffsetTrim	LREAL	Position offset of the trimming function.	
MP_lOffsetTotalVel	LREAL	Speed of the total position offset.	
MP_lOffsetSyncPos	LREAL	Position offset generated by asynchronous coupling.	
MP_lrMarkSensorSetPosTool	LREAL	Position of the tool sensor within the slave cycle length.	
MP_lrMarkActTpPosTool	LREAL	Measured tool position within the slave cycle length.	
MP_lrActMarkErrorTool	LREAL	Unlimited tool error.	
MP_xMarkReceivedTool	BOOL	TRUE	The setpoint of the slave axis is within the tool correction window.
MP_xCorrActiveTool	BOOL	TRUE	The tool correction movement is limited.
MP_xCorrLimitedTool	BOOL	TRUE	The tool correction movement is limited.
MP_xMarkWindowOpenTool	BOOL	TRUE	The tool correction mark window is open.
MP_xMarkOutOfWindowTool	BOOL	TRUE	Mark outside the tool correction mark window.
MP_xMarkNotDetectedTool	BOOL	TRUE	No mark has occurred in the tool correction mark window.
MP_xGearFactorCorrLimited	BOOL	TRUE	The gearbox factor correction is limited to the maximum value from <i>scPar.lMaxGearFactorCorrMaster</i> .

7.2 Structure of the access points

L_TS2P_scAP_SyncAndCorrectionState

The access points (AP) can be used to influence signals. In the initial state, the access points do not have any effect.

Each access point acts as an alternative branch and is activated via an OR operation or a switch.

Designator	Data type	Description
AP_xLoadGearBoxPosOut	BOOL	TRUE Load gearbox factor output position as per <i>AP_IrLoadGearBoxPosOut</i> .
AP_IrLoadGearBoxPosOut	LREAL	Gearbox factor output position
AP_xLoadTrimOffset	BOOL	TRUE Load trimming offset as per <i>AP_IrLoadTrimOffset</i> .
AP_IrLoadTrimOffset	LREAL	Trimming offset
AP_xLoadOffsetSync	BOOL	TRUE Load coupling offset as per <i>AP_IrLoadOffsetSync</i> .
AP_IrLoadOffsetSync	LREAL	Coupling offset

8 Description of the basic functions

Overview of the basic functions

- ▶ [Manual jog \(jogging\)](#) (📖 34)
- ▶ [Homing](#) (📖 36)
- ▶ [QSP application](#) (📖 39)
- ▶ [Halt](#) (📖 40)
- ▶ [Absolute positioning](#) (📖 41)
- ▶ [Relative positioning](#) (📖 43)
- ▶ [Continuous travel with a defined velocity](#) (📖 46)
- ▶ [Aborting and completing a positioning process](#) (📖 49)
- ▶ [Override for speed](#) (📖 51)

8 Description of the basic functions

8.1 Manual jog (jogging)

8.1 Manual jog (jogging)

Prerequisite

- The axis is enabled (*xReadyForMotion* = TRUE).

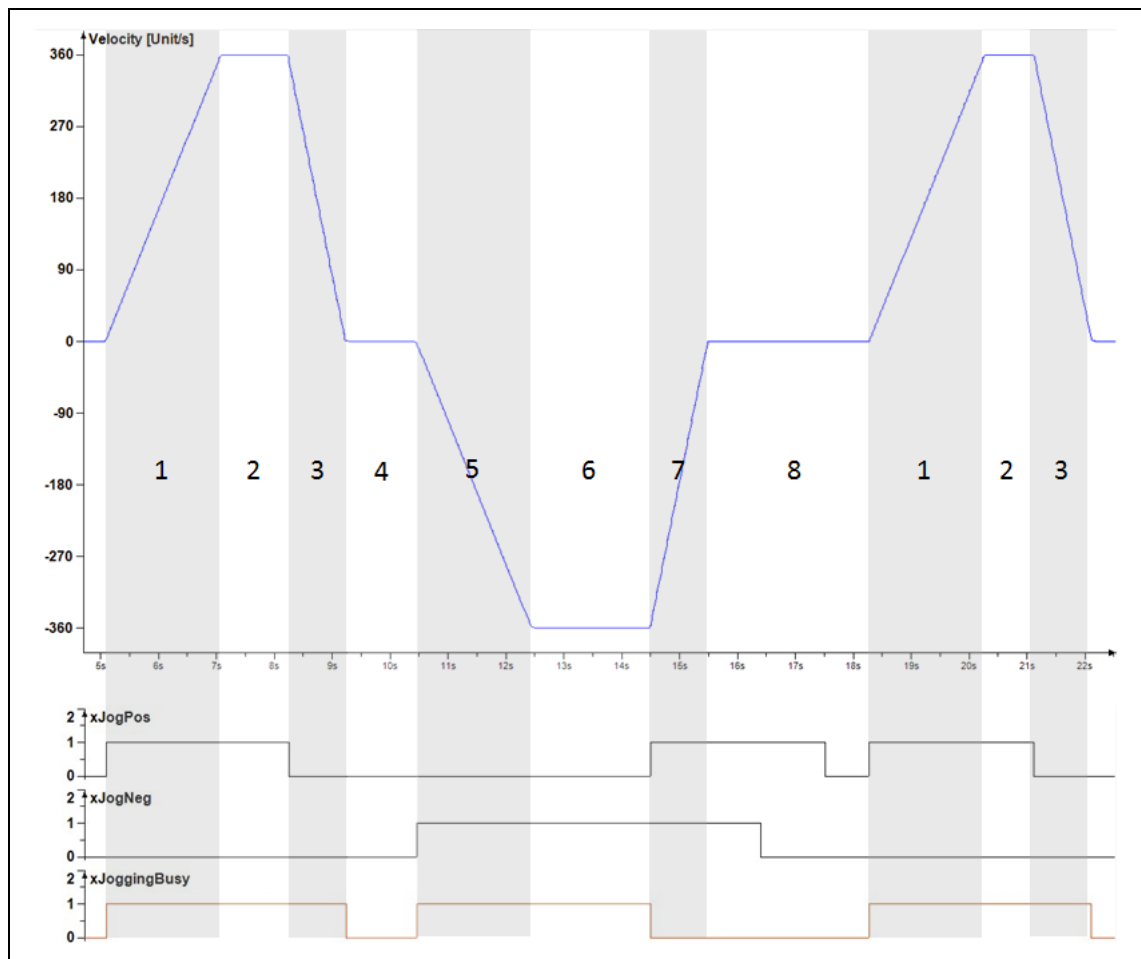
Execution

With the *xJogPos* input = TRUE, the axis is traversed in the positive direction and with the *xJogNeg* input = TRUE, it is traversed in the negative direction. The axis is traversed as long as *xJogPos* or *xJogNeg* remain set to TRUE.

The current travel command cannot be replaced by the other jog command. If *xJogPos* = TRUE and *xJogNegPos* = TRUE, the axis brakes into standstill and the *eTMState* status output changes to the "ERROR" state.

Only if both inputs have been reset, the technology module changes back to the "STANDBY" state.

Travel diagram



No	xJogPos	xJogNeg	xJogging Busy	eTMState	Info
1	FALSE → TRUE	FALSE	FALSE → TRUE	STANDBY → JOGGING	The axis accelerates to the positive manual speed 360 units/s with 180 unit/s².
2	TRUE	FALSE	TRUE	JOGGING	The axis travels constantly with 360 units/s.
3	FALSE → TRUE	FALSE	TRUE	JOGGING	The axis decelerates to a standstill with 360 unit/s².
4	FALSE	FALSE	TRUE → FALSE	JOGGING → STANDBY	The axis has reached a standstill.
5	FALSE	FALSE → TRUE	FALSE → TRUE	STANDBY → JOGGING	The axis accelerates to the negative manual speed -360 units/s with 180 unit/s².
6	FALSE	TRUE	TRUE	JOGGING	The axis travels constantly with -360 units/s.
7	TRUE	TRUE	TRUE → FALSE	ERROR	The axis decelerates to a standstill with 360 unit/s².
8	FALSE	FALSE	FALSE	ERROR → STANDBY	The axis is at a standstill and changes to "STANDBY" if both inputs xJogPos and xJogNeg have the value FALSE.

Parameters to be set

The parameters for manual jog can be found in the AXIS_REF in subindex 181 ... 184.

The parameter values can be changed during operation. They are accepted and become effective immediately when setting the inputs xJogPos = TRUE or xJogNeg = TRUE.

The manual jog parameters from the PLC program can also be written via the L_MC1P_SetBasicParameter block.

8.2**Homing****Prerequisites**

- The technology module is in "STANDBY" mode.
- The axis is enabled (*xReadyForMotion* = TRUE).

Execution

With a rising edge (FALSE→TRUE) at the *xHomeExecute* input, homing is started (output *xHomingBusy* = TRUE). The axis travels until the home position is reached.

When the reference switch has been reached, *xisHomed* switches to TRUE.

When the homing process has been completed successfully, *xHomingDone* changes to TRUE.

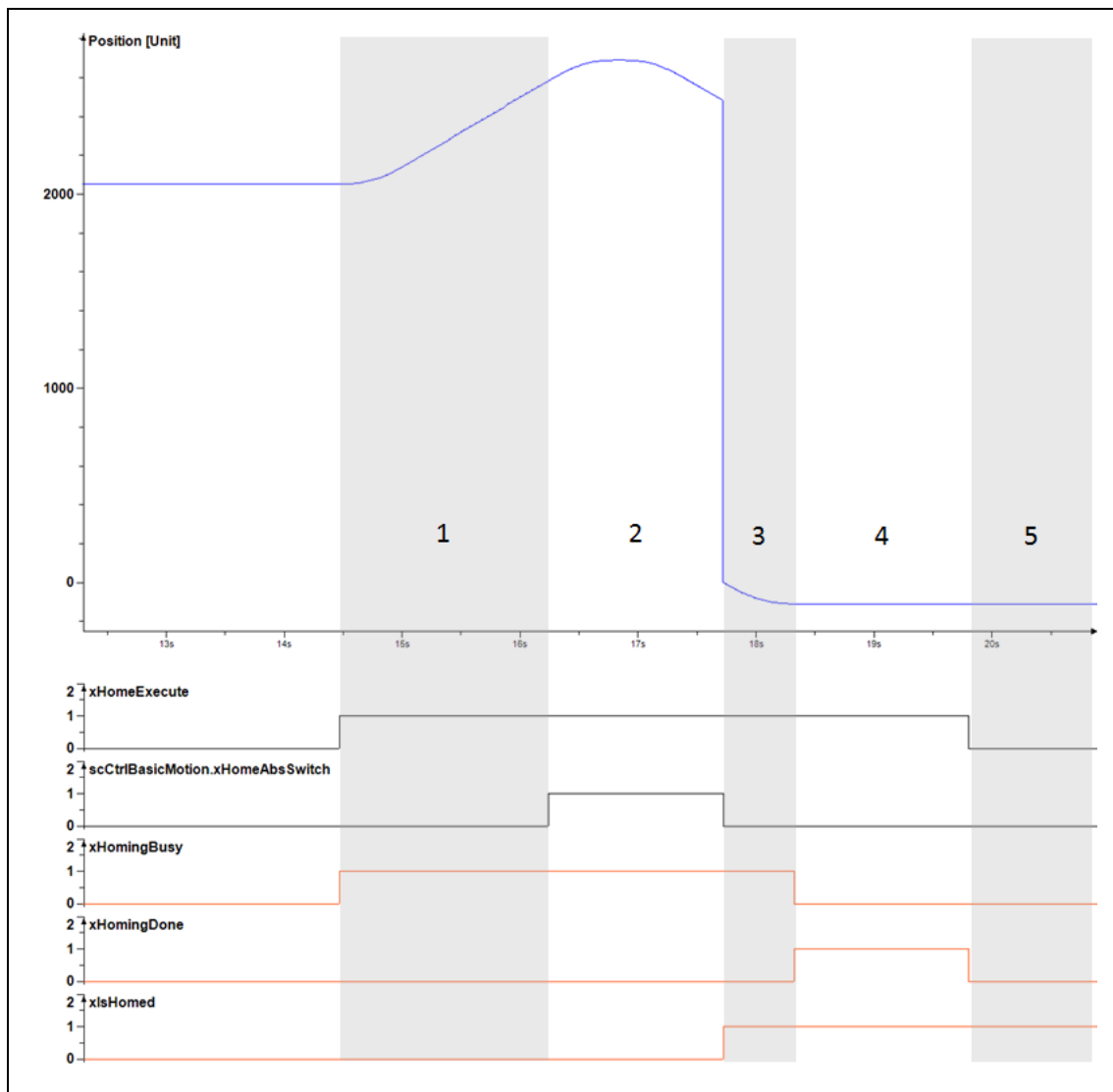
If *xHomeExecute* = FALSE, the state is changed to "STANDBY" and *xHomingDone* is set = FALSE.

Abort of the homing process

The function is aborted via the *xHalt* input.

Homing is not interrupted if *xHomeExecute* is set to = FALSE early.

Travel diagram



[8-1] Homing in the CwRpCwRn module with the home position "0"

No	xHome Execute	xHome AbsSwitch	xHoming Busy	xHoming Done	xIsHomed	eTMState	Info
1	FALSE → TRUE	FALSE	FALSE → TRUE	FALSE	FALSE	STANDBY → HOMING	The axis starts homing in positive direction.
2	TRUE	TRUE → FALSE	TRUE	FALSE	FALSE	HOMING	The axis changes the direction of rotation due to the positive edge of xHomeAbsSwitch.
3	TRUE	FALSE → TRUE	TRUE	FALSE	FALSE → TRUE	HOMING	The negative edge of xHomeAbsSwitch sets the position of the axis to the home position 0 units and the axis is brought to a standstill.
4	TRUE	FALSE	TRUE → FALSE	FALSE → TRUE	TRUE	HOMING	As soon as the axis has reached the standstill, xHomingBusy is set = FALSE and xHomingDone is set = TRUE. xHomingDone remains TRUE for as long as xHomeExecute is set = TRUE, but always at least for one clock cycle.
5	TRUE → FALSE	FALSE	FALSE	TRUE → FALSE	TRUE	HOMING → STANDBY	The axis is ready for operation and can be positioned in an absolute fashion.

Parameters to be set

The parameters for homing can be found in the `AXIS_REF` in subindex 70 ... 83.

The manual jog parameters from the PLC program can also be written via the `L_MC1P_SetBasicParameter` block.

8.3 QSP application

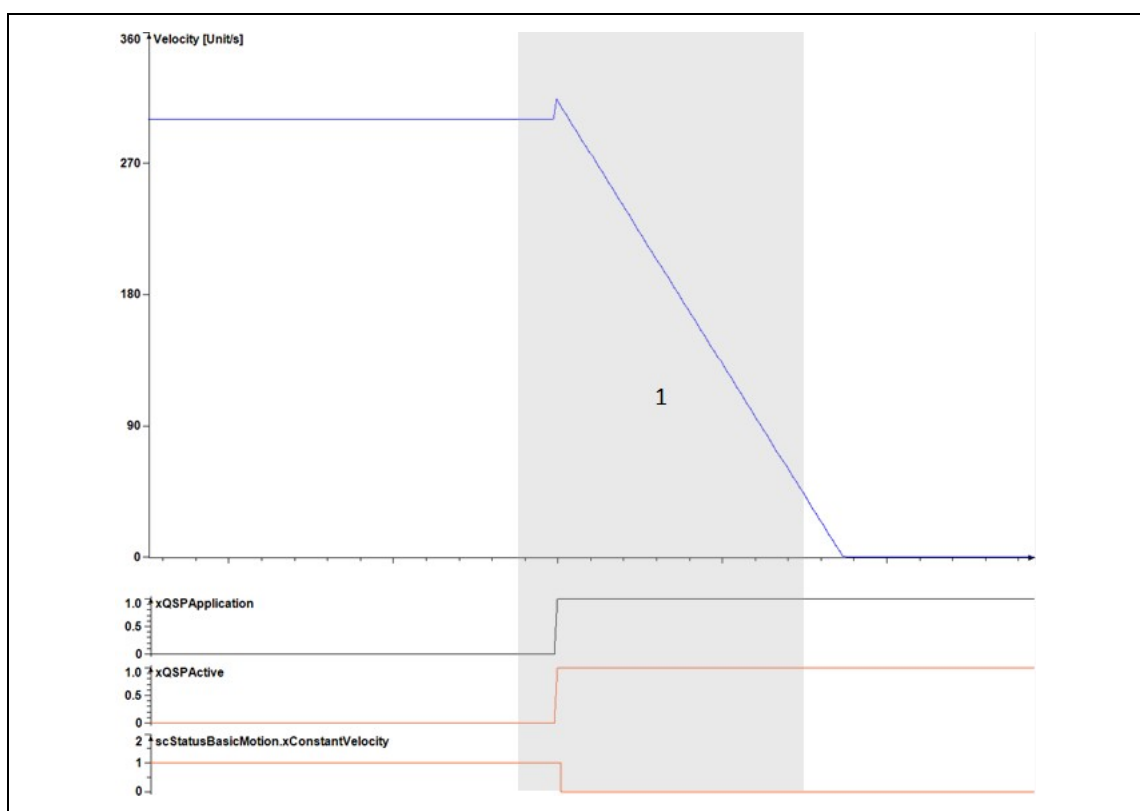
Prerequisite

- The axis is enabled ($xReadyForMotion = TRUE$).

Execution

With a rising edge ($FALSE \rightarrow TRUE$) at the $xQSPApplication$ input, the current movement of the axis is interrupted and the axis is brought to a standstill on the basis of its last current speed value on the QSP application deceleration ramp. The parameter for the QSP application deceleration ramp is set in the AXIS_REF.

Travel diagram



No	xQSP Application	xQSPActive	xConstant Velocity	eTMState	Info
1	$FALSE \rightarrow TRUE$	$FALSE \rightarrow TRUE$	$TRUE \rightarrow FALSE$	MOVEVEL → STOP	The speed setpoint of the axis is set to the last current speed and brought to a standstill via the QSP deceleration ramp.

Parameters to be set

The parameters for "QSP Application" can be found in the AXIS_REF in subindex 48 and 49.

The manual jog parameters from the PLC program can also be written via the **L_MC1P_SetBasicParameter** block.

8.4 Halt

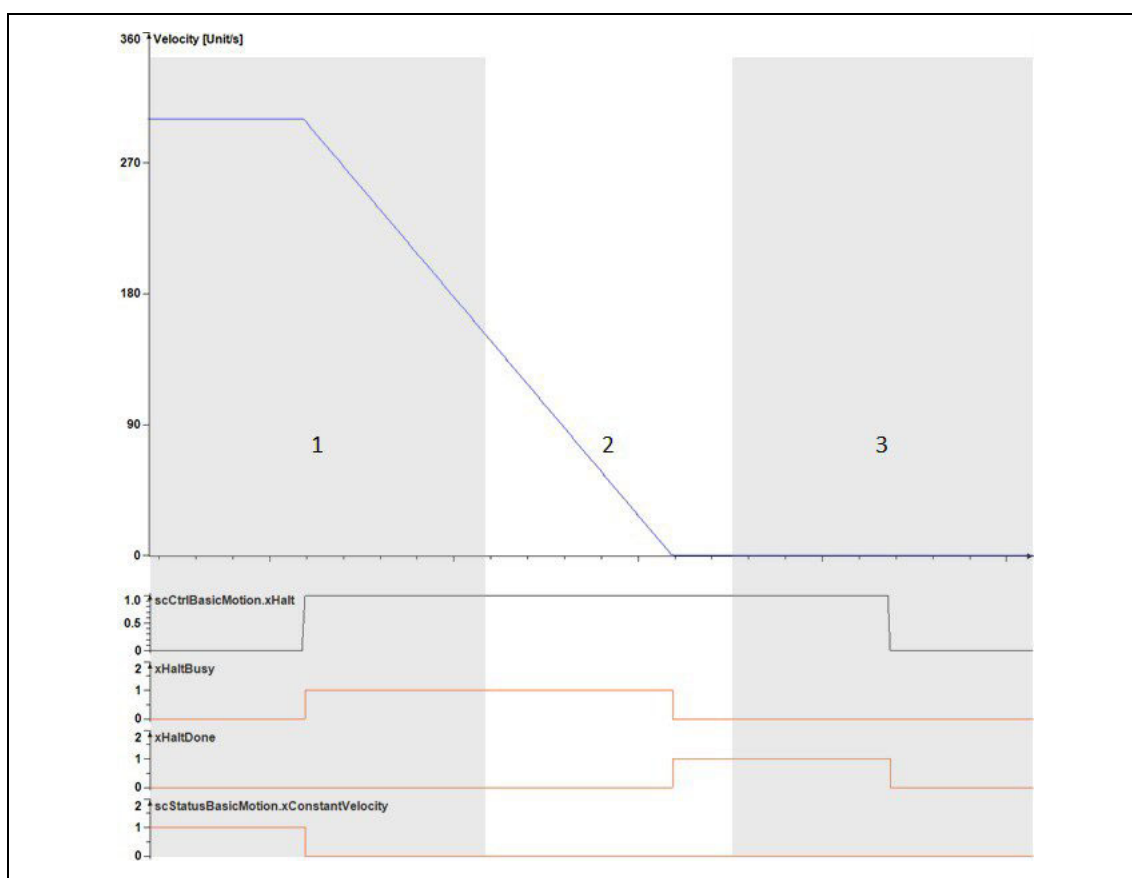
Prerequisite

- The axis is enabled ($xReadyForMotion = TRUE$).

Execution

With a rising edge ($FALSE \rightarrow TRUE$) at the `scCtrlBasicMotion.xHalt` structure input, the current movement of the axis is interrupted and the axis is brought to a standstill with a halt deceleration ramp.

Travel diagram



No	xHalt	xHaltBusy	xHaltDone	xConstant Velocity	eTMState	Info
1	$FALSE \rightarrow TRUE$	$FALSE \rightarrow TRUE$	$FALSE$	$TRUE \rightarrow FALSE$	$MOVEVEL \rightarrow STOP$	The axis decelerates with the halt deceleration ramp: $xHaltBusy = TRUE$ $xConstantVelocity = FALSE$.
2	$TRUE$	$TRUE \rightarrow FALSE$	$FALSE \rightarrow TRUE$	$FALSE$	$STOP$	The axis is brought to a standstill: $xHaltBusy = FALSE$ $xHaltDone = TRUE$
3	$TRUE \rightarrow FALSE$	$FALSE \rightarrow TRUE$	$TRUE \rightarrow FALSE$	$FALSE$	$STOP \rightarrow STANDBY$	$xHalt = FALSE$ $xHaltDone = TRUE \rightarrow FALSE$.

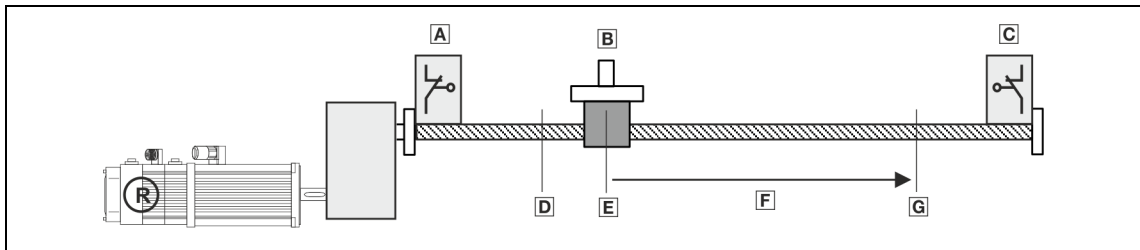
Parameters to be set

The parameter for the halt can be found in the `AXIS_REF` in subindex 186 and 187.

8 Description of the basic functions

8.5 Absolute positioning

8.5 Absolute positioning



[8-2] Example: Absolute positioning of a linear axis

Pos.	Description
A	Negative hardware limit switch
B	Load (e.g. slide)
C	Positive hardware limit switch
D	Zero position of the measuring system
E	Current actual position
F	Distance to be travelled
G	Target position (<i>lrPos_Dist</i> input)

With absolute positioning, the target position in the *lrPos_Dist* input (G) is defined as a unique absolute position, relating to the zero position of the measuring system (D). The distance to be travelled (F) results from the distance between the current actual position (E) and the absolute target position (G).

Prerequisites

- The axis is enabled (*xReadyForMotion* = TRUE).
- The axis is referenced (*xisHomed* = TRUE).

Execution

With a rising edge (FALSE→TRUE) at the *xMoveAbsExecute* input, a travel to the absolute target position in the *lrSetPos_Dist* input is started on the basis of the parameters for the motion profile (see below). The technology module changes to the "MOVEABS" state.

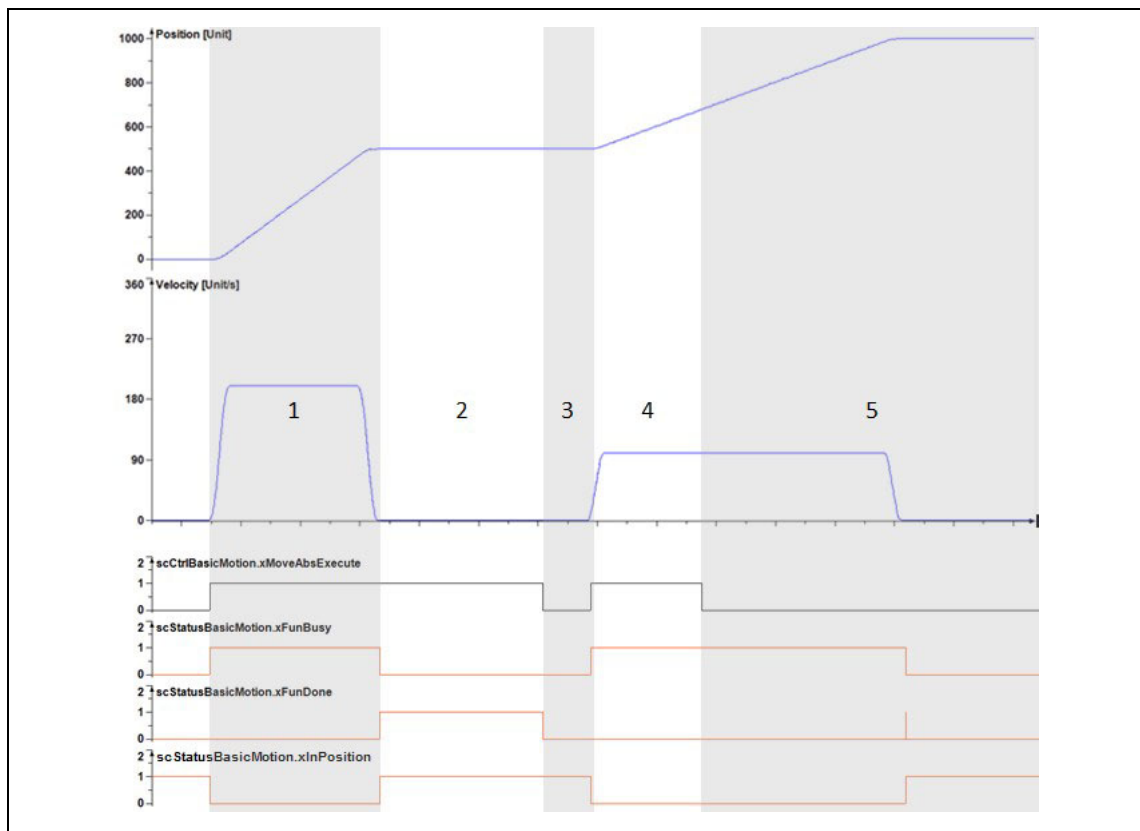
In order to reach the target position, the axis is brought to a standstill via the *lrDec* parameter. The axis reaches a standstill exactly in the target position.

After reaching the target position, the state changes back to STANDBY (*xMoveAbsExecute* = FALSE).

During the travel process, the absolute target position in the *lrPos_Dist* input can be changed, which requires triggering the Execute input for positioning again.

Positioning can be interrupted via *xAbort*, *xHalt* or *xQSPApplication*. Via *xFinishPositioning*, positioning can be continued again and completed.

Travel diagram



No	xMoveAbsExecute	xFuncBusy	xFuncDone	xInPosition	eTMState	Info
1	FALSE ↗ TRUE	FALSE ↗ TRUE	FALSE	TRUE ↘ FALSE	STANDBY → MOVEABS	The axis starts the positioning run to the absolute target position 500 units.
2	TRUE	TRUE ↘ FALSE	FALSE ↗ TRUE	FALSE ↗ TRUE	MOVEABS → STANDBY	When the target position has been reached, <i>xInPosition</i> is set to TRUE.
3	TRUE ↘ FALSE	FALSE	TRUE ↘ FALSE	TRUE	STANDBY	When <i>xMoveAbsExecute</i> is set to FALSE, the <i>xFuncDone</i> status signal changes from TRUE ↘ FALSE.
4	FALSE ↗ TRUE	FALSE ↗ TRUE	FALSE	TRUE ↘ FALSE	STANDBY → MOVEABS	The axis starts the positioning run to the absolute target position 1000 units.
5	TRUE ↘ FALSE	TRUE ↘ FALSE	FALSE ↗ TRUE ↘ FALSE	FALSE ↗ TRUE	MOVEABS → STANDBY	After the target position has been reached, <i>xFuncDone</i> remains TRUE for exactly one cycle, because <i>xMoveAbsExecute</i> = FALSE has been requested during the positioning process.

Parameters to be set

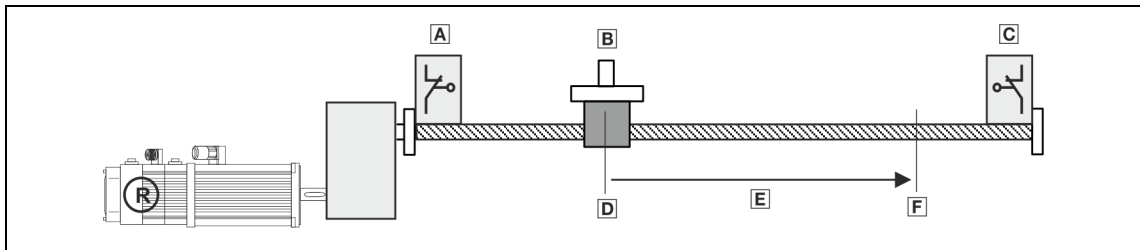
- *IrVel*, initial value: 3600 [units/s]
- *IrAcc*, initial value: 36000 [units/s²]
- *IrDcc*, initial value: 36000 [units/s²]
- *IrJerk*, initial value: 0 [units/s³]

Further parameters for absolute positioning can be found in the *AXIS_REF* in subindex 130 ... 137.

8 Description of the basic functions

8.6 Relative positioning

8.6 Relative positioning



[8-3] Example: Relative positioning of a linear axis

Pos.	Description
A	Negative hardware limit switch
B	Load (e.g. slide)
C	Positive hardware limit switch
D	Current actual position
E	Distance to be travelled (= <i>lrPos_Dist</i> input)
F	Target position (input current actual position + <i>lrPos_Dist</i>)

With relative positioning, the target position defined in the *lrPos_Dist* input corresponds to the distance to be travelled ($F = E$), meaning that the drive exactly travels the distance defined from the current actual position (D).

Prerequisite

- The axis is enabled (*xReadyForMotion* = TRUE).

Execution

With a rising edge (FALSE→TRUE) at the *xMoveRelExecute* input, a travel to the relative target position (*lrPos_Dist* input) is started on the basis of the parameters for the motion profile (see below). The technology module changes to the "MOVEREL" state.

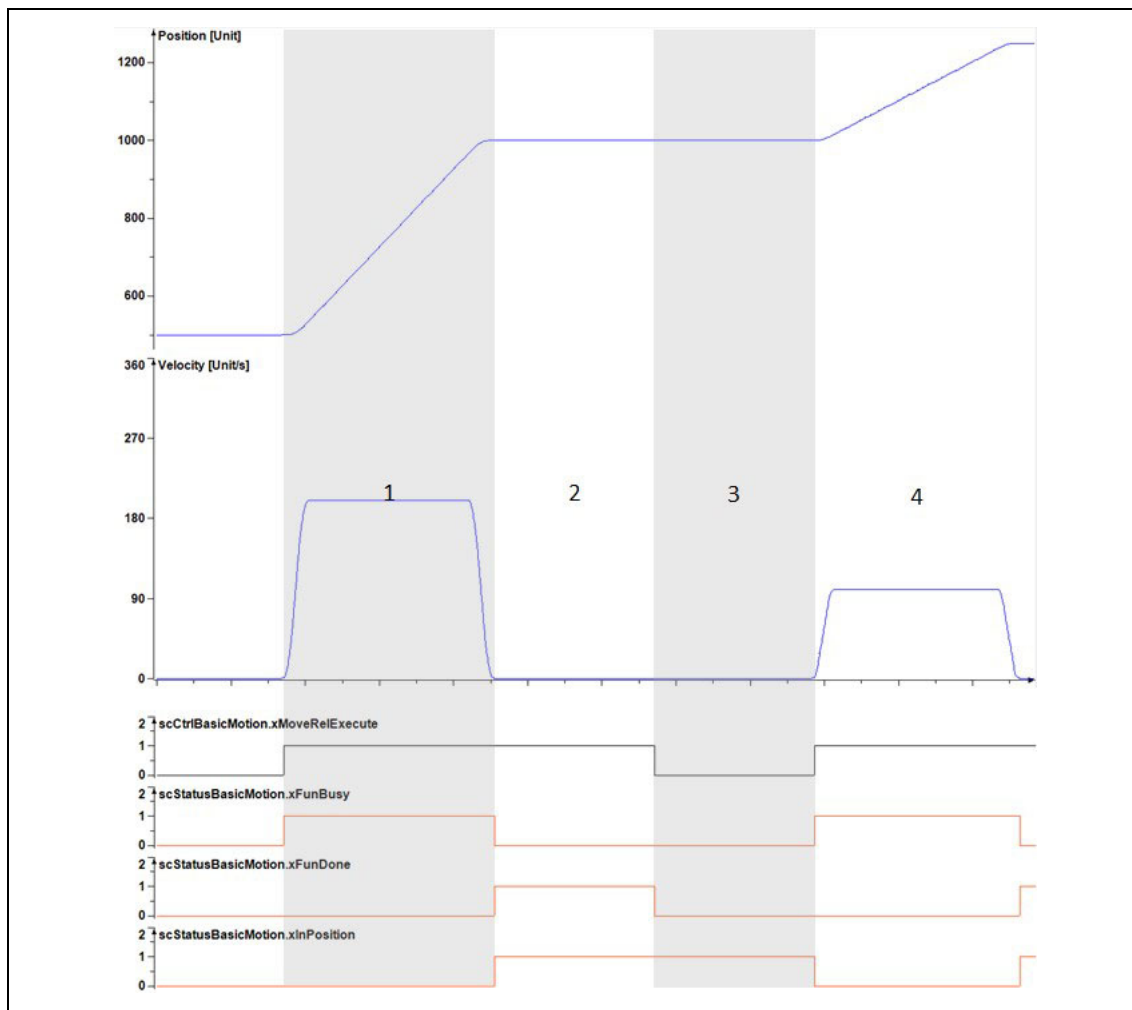
In order to reach the target position, the axis is brought to a standstill via the *lrDec* parameter. The axis reaches a standstill exactly in the target position.

After reaching the target position, the state changes back to STANDBY (*xMoveRelExecute* = FALSE).

During the travel process, the absolute target position in the *lrPos_Dist* input can be changed, which requires triggering the Execute input for positioning again.

Positioning can be interrupted via *xAbort*, *xHalt* or *xQSPApplication*. Via *xFinishPositioning*, positioning can be continued again and completed.

Travel diagram



No	xMoveRelExecute	xFuncBusy	xFuncDone	xInPosition	eTMState	Info
1	FALSE → TRUE	FALSE → TRUE	FALSE	FALSE	STANDBY → MOVEREL	The axis starts the positioning run to the relative target position 500 units.
2	TRUE	TRUE → FALSE	FALSE → TRUE	FALSE → TRUE	MOVEREL → STANDBY	As soon as the relative target position has been reached, <i>xInPosition</i> and <i>xFuncDone</i> are set to TRUE.
3	FALSE → TRUE	FALSE	TRUE → FALSE	TRUE	STANDBY	When <i>xMoveRelExecute</i> is set = FALSE, <i>xFuncDone</i> changes from TRUE → FALSE.
4	FALSE → TRUE	FALSE → TRUE → FALSE	TRUE → FALSE → TRUE	TRUE → FALSE → TRUE	STANDBY → MOVEREL → STANDBY	The axis starts the positioning run to the relative target position 250 units. After reaching the relative target position, <i>xInPosition</i> and <i>xFuncDone</i> are set to TRUE.

Parameters to be set

- *IrVel*, initial value: 3600 [units/s]
- *IrAcc*, initial value: 36000 [units/s²]
- *IrDcc*, initial value: 36000 [units/s²]
- *IrJerk*, initial value: 0 [units/s³]

Further parameters for relative positioning can be found in the AXIS_REF in subindex 130 ... 137.

8.7 Continuous travel with a defined velocity

Prerequisite

- The axis is enabled (*xReadyForMotion* = TRUE).

Execution

With a rising edge (FALSE→TRUE) at the *xMoveVelExecute* input, the continuous travel is started on the basis of the parameters for the motion profile (see below). The technology module changes to the "MOVEVEL" state.

The axis travels until a software limit position has been reached or a hardware limit switch has been approached:

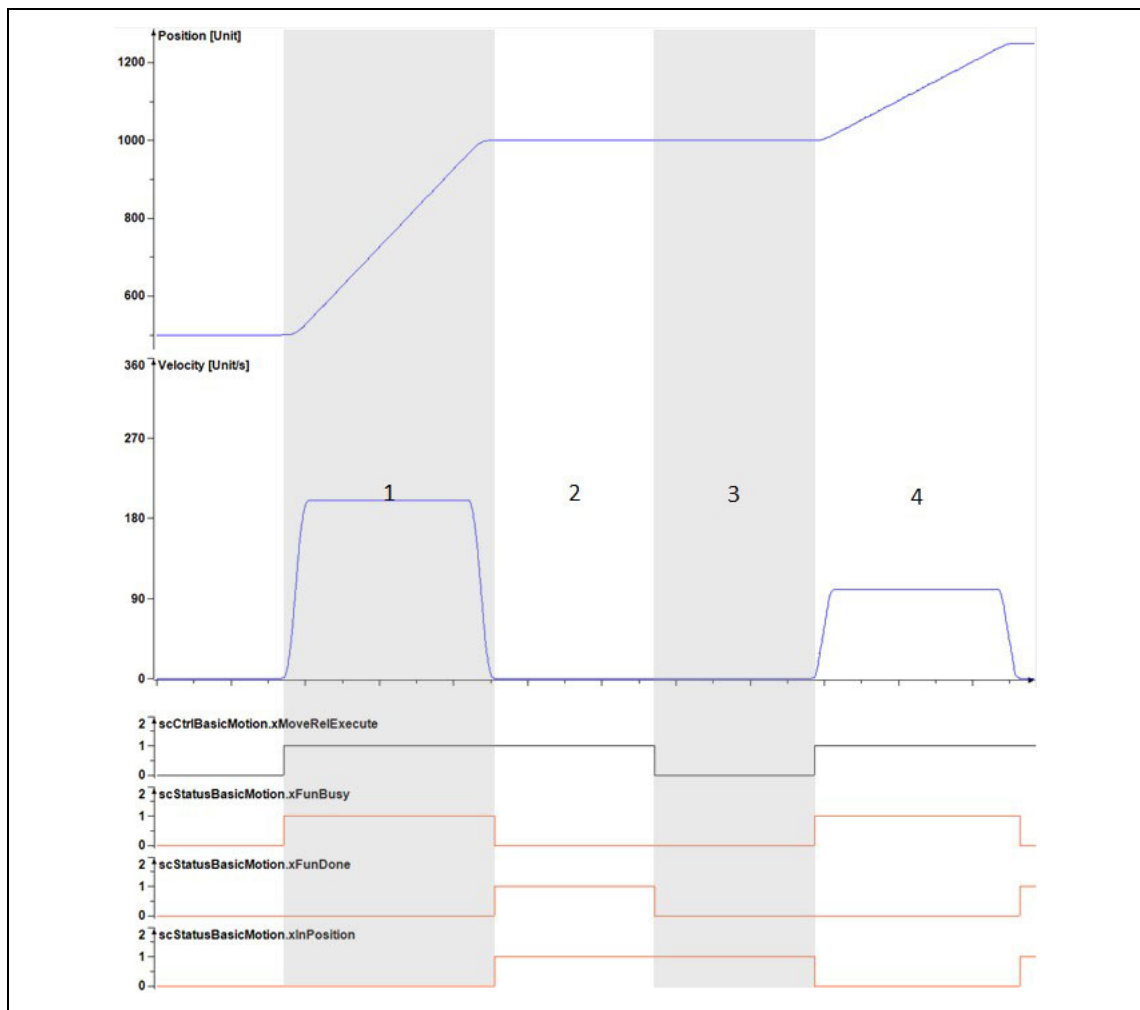
- Input *xDisableSWLimit* = FALSE
- Output *xSwLimitEnabled* = TRUE
- Output *xSwLimitSwitchPos/Neg* = TRUE
- Output *xHwLimitSwitchPos/Neg* = TRUE

When a software limit position has been reached, the axis is brought to a standstill via the *IrDec* parameter.

When a hardware limit switch has been reached, the axis is brought to a standstill via the QSP application deceleration ramp.

Then the state is changed back to "STANDBY".

Travel diagram



No	<code>xMoveVelExecute</code>	<code>xAbort</code>	<code>xFuncBusy</code>	<code>xFuncDone</code>	<code>xConstantVelocity</code>	<code>eTMState</code>	Info
1	FALSE ↗ TRUE ↘ FALSE	FALSE	FALSE ↗ TRUE	FALSE	FALSE	STANDBY → MOVEVEL	The axis accelerates to the target speed of 200 units/s.
2	FALSE	FALSE	TRUE	FALSE ↗ TRUE	FALSE ↗ TRUE	MOVEVEL	After reaching the target speed, <code>xFuncDone</code> changes from FALSE ↗ TRUE and <code>xConstantVelocity</code> changes from FALSE ↗ TRUE.
3	FALSE	FALSE ↗ TRUE	TRUE ↘ FALSE	TRUE ↘ FALSE ↗ TRUE	TRUE ↘ FALSE	MOVEVEL → STANDBY	The speed run is terminated by <code>xAbort = TRUE</code> and brought to a standstill with the profile deceleration. When standstill has been reached, <code>xFuncBusy</code> is set = FALSE.

Parameters to be set

- *IrVel*, initial value: 3600 [units/s]
- *IrAcc*, initial value: 36000 [units/s²]
- *IrDcc*, initial value: 36000 [units/s²]
- *IrJerk*, initial value: 0 [units/s³]

Further parameters for continuous travel can be found in the AXIS_REF in subindex 50 ... 53 and subindex 104 ... 105.

8.8 Aborting and completing a positioning process

Prerequisite

- The axis is enabled ($xReadyForMotion = TRUE$).

Execution

An active positioning process is aborted with a rising edge ($FALSE \rightarrow TRUE$) at the $xAbort$ input.

The axis travels to a standstill with the deceleration ramp $scPar.IrDec$.

- Exception: **TM SpeedControl** and **TM TablePositioning**

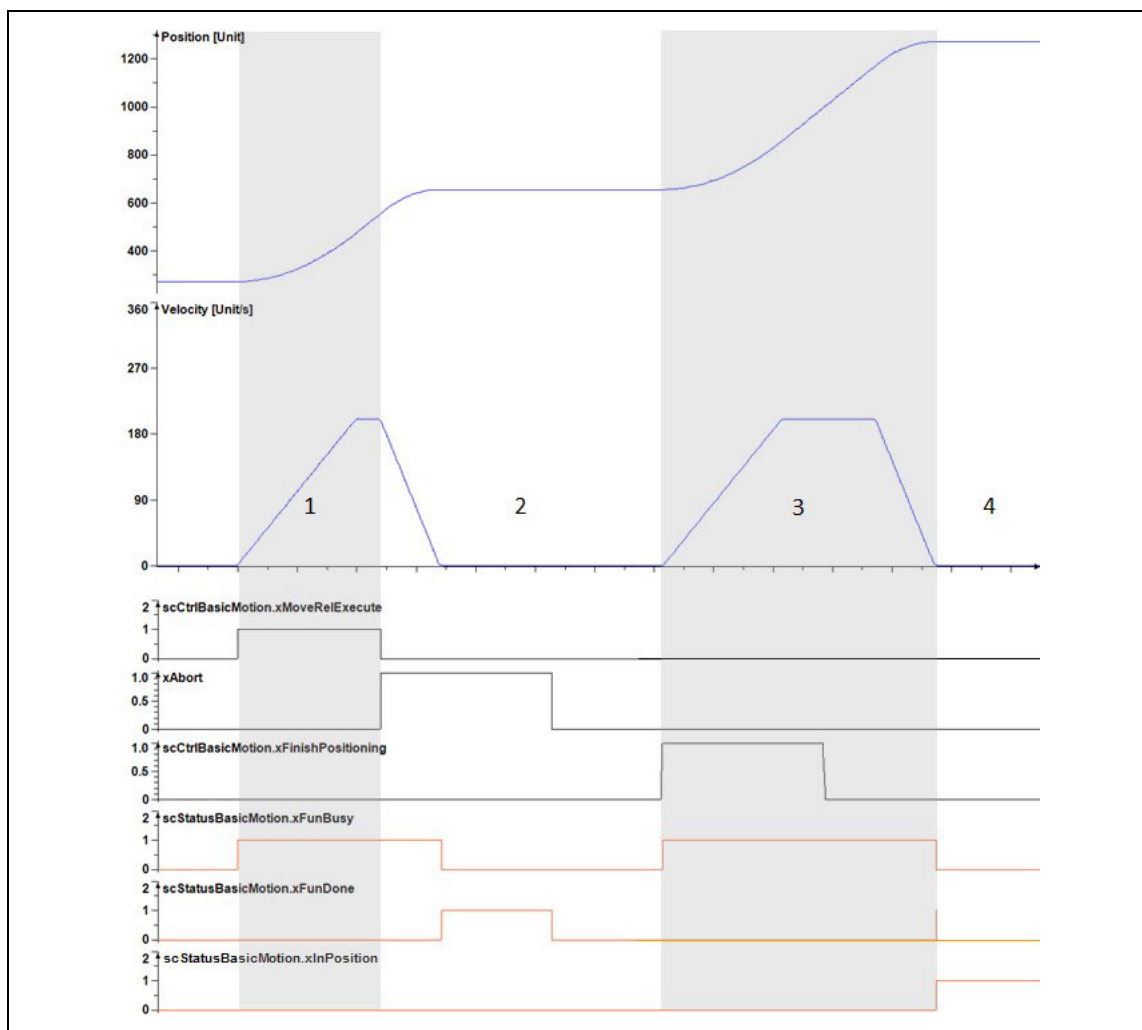
Both technology modules provide several profile data sets, one of which can be active.

If one of these profile data sets is active, the axis moves to standstill with the deceleration ramp of this active profile data set.

If no profile data set is active, the default deceleration ramp $scPar.IrDec$ is used.

The positioning process is completed with a rising edge ($FALSE \rightarrow TRUE$) at the $xFinishPositioning$ input.

Travel diagram



No	xMoveRel Execute	xAbort	xFinish Positioning	xInPosition	eTMState	Info
1	FALSE ↗ TRUE	FALSE	FALSE	FALSE	STANDBY→ MOVEREL	The axis starts the positioning process with the relative target position 1000 units.
2	TRUE↘ FALSE	TRUE↘ FALSE	FALSE	FALSE	MOVEREL→ STOP→ STANDBY	The positioning process is aborted with <i>xAbsort</i> = TRUE. The axis is brought to a standstill with the profile deceleration.
3	FALSE	FALSE ↗ TRUE	FALSE↗ TRUE↘ FALSE	FALSE	STANDBY→ MOVEREL	With <i>xFinishPositioning</i> = TRUE, the positioning process is continued.
4	FALSE	FALSE	FALSE	FALSE ↗ TRUE	MOVEREL→ STANDBY	As soon as the relative target position has been reached, <i>xInPosition</i> and <i>xFunDone</i> are set to TRUE.
	FALSE ↗ TRUE	FALSE↗ TRUE↘ FALSE	TRUE↘ FALSE↗ TRUE	TRUE↘ FALSE↗ TRUE	STANDBY→ MOVEREL→ STANDBY	The axis starts the positioning process with the relative target position 250 units. After reaching the relative target position, <i>sxInPosition</i> and <i>xFunDone</i> are set to TRUE.

8.9

Override for speed

Prerequisite

- The axis is enabled ($xReadyForMotion = TRUE$).

Execution

In order to be able to set overrides for the active profile data, the following inputs are available at the technology module:

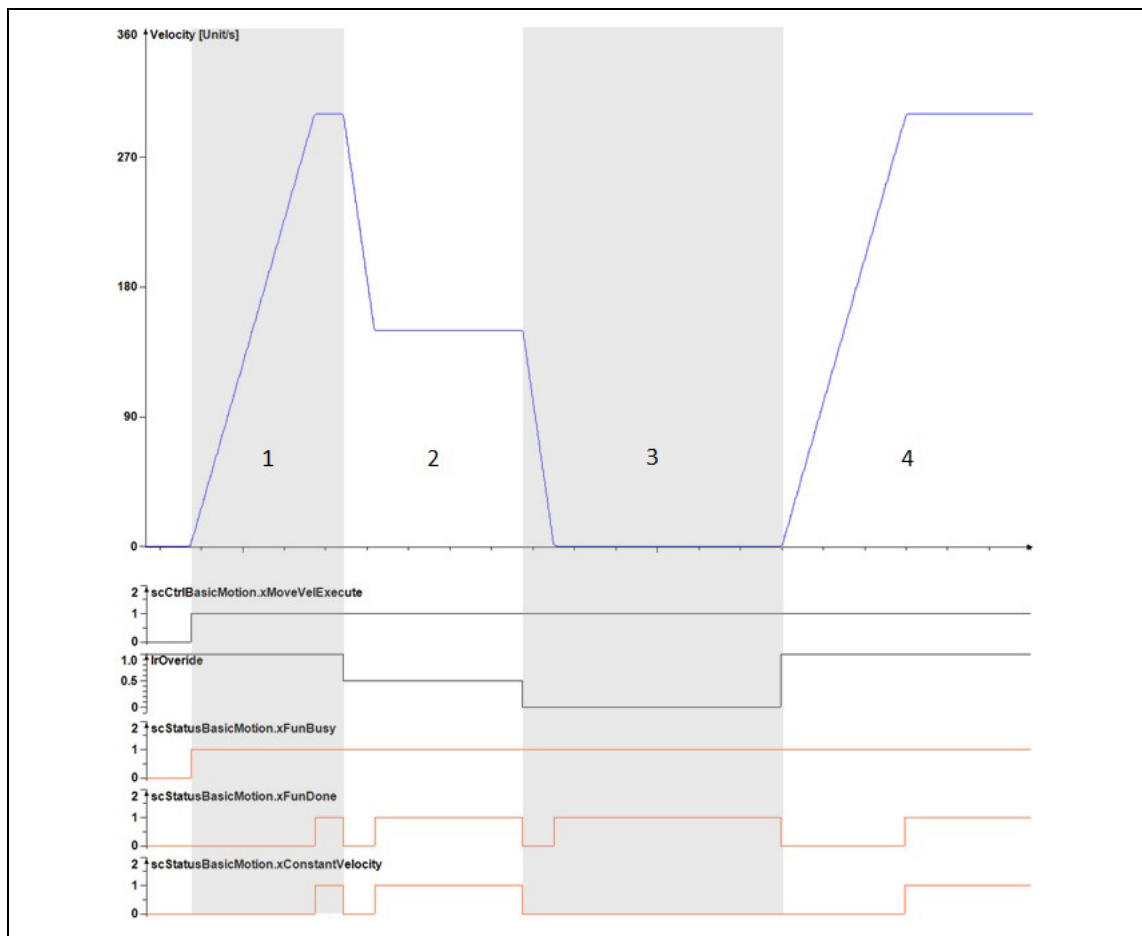
- *IrOverride*, initial value: 1.0
- *scCtrlBasicMotion.IrOverrideAcc*, initial value: 1.0 (100 %)
- *scCtrlBasicMotion.IrOverrideJerk*, initial value: 1.0 (100 %)

For profile-generating functions, separate overrides for speed, acceleration, deceleration and jerk can be set.

The values set act as a factor and are multiplied with the value active in each case.

Example

In this example, the use of the *IrOverride* input is shown.



8 Description of the basic functions

8.9 Override for speed

No	xMoveVel Execute	IrOverride	xFuncBusy	xFunDone	xConstant Velocity	eTMState	Info
1	FALSE ↗ TRUE	1	FALSE ↗ TRUE	FALSE ↗ TRUE	FALSE ↗ TRUE	STANDBY→ MOVEVEL	The axis accelerates to the target speed of 300 units/s.
2	TRUE	0.5	TRUE	TRUE↘ FALSE↗ TRUE	TRUE↘ FALSE↗ TRUE	MOVEVEL	The override value is set to 0.5 and the axis decelerates to 150 unit/s with the profile deceleration
3	TRUE	0	TRUE	TRUE↘ FALSE↗ TRUE	TRUE↘ FALSE	MOVEVEL	The override value is set to 0 and the axis is brought to a standstill to 0 unit/s with the profile deceleration
4	TRUE	1	TRUE	TRUE↘ FALSE↗ TRUE	FALSE ↗ TRUE	MOVEVEL	The override value is set to 1 and the axis accelerates to 300 unit/s with the profile deceleration

9 Description of the technology functions

9.1 Position synchronism

9.1.1 Position synchronism without basic speed

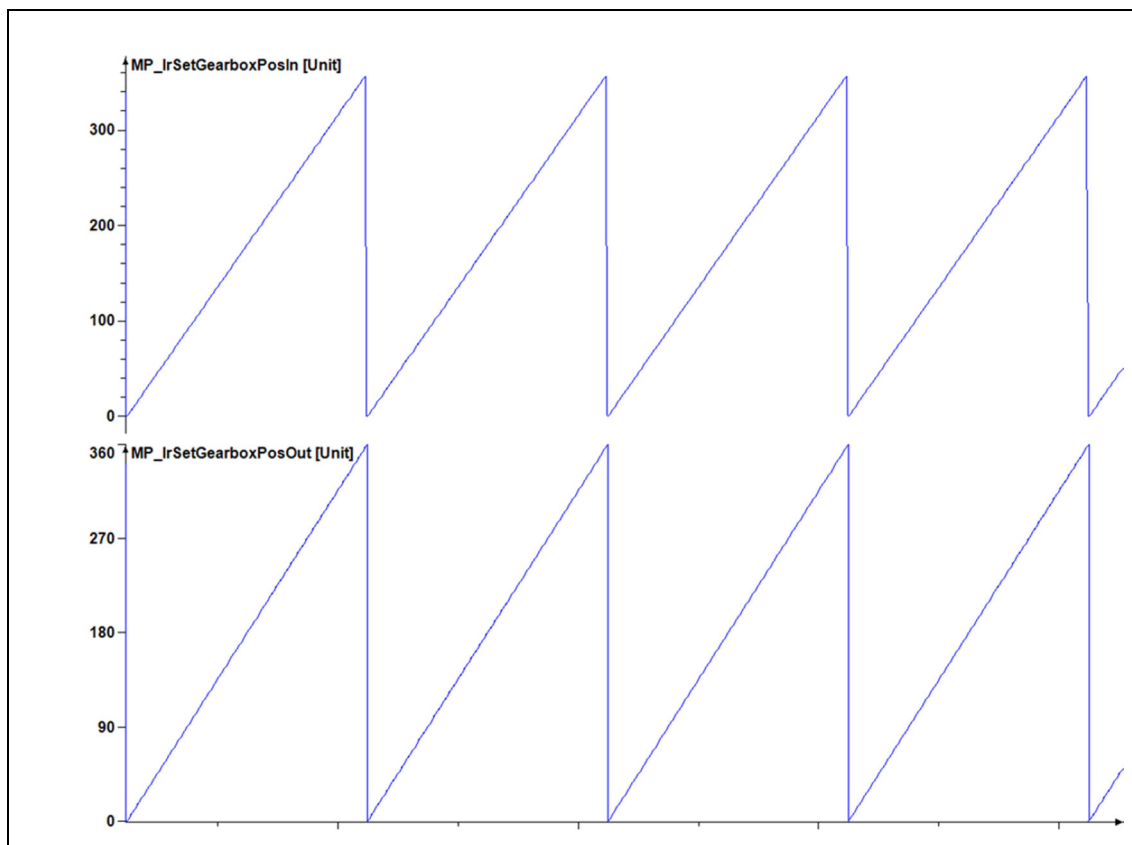
This function can be used to establish synchronism between a master axis and a drive axis (slave).

Preconditions

- The technology module is in an error-free state.
- The axis is enabled ($xReadyForMotion = TRUE$).

Type

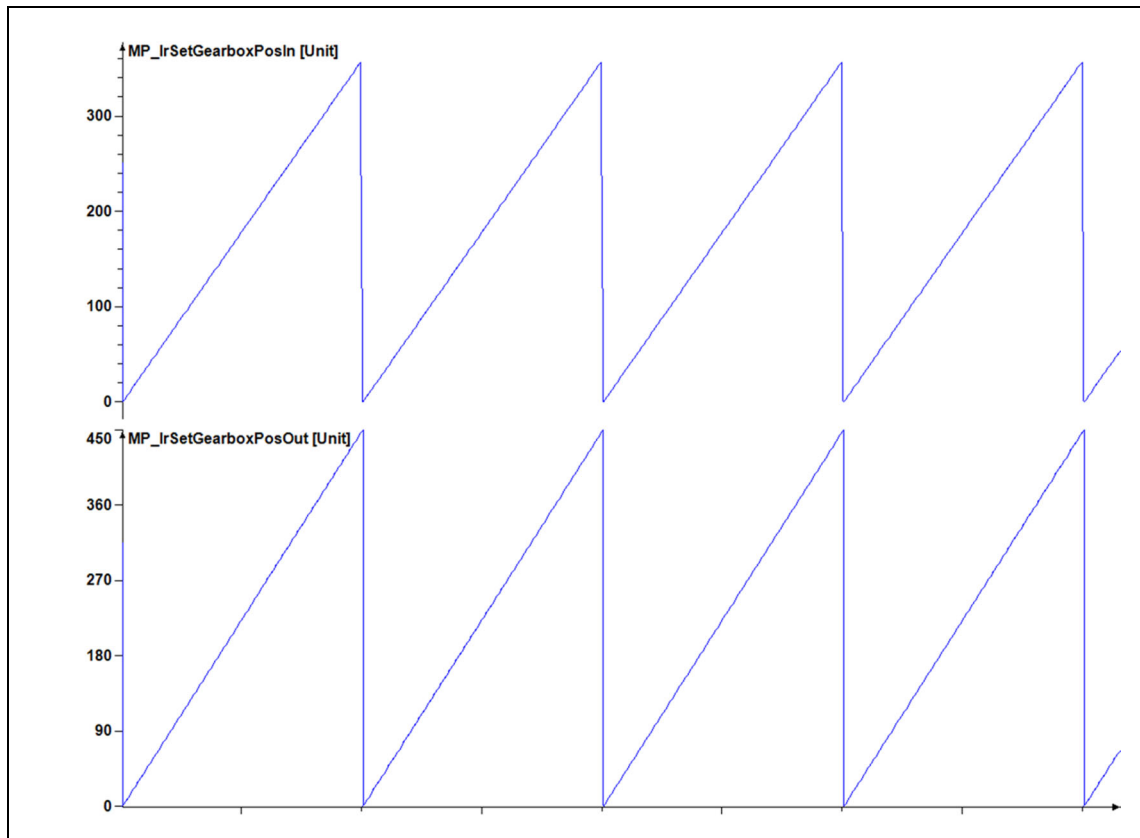
For the angular synchronism of master axis and drive axis, a setpoint position based on the master axis is calculated by means of cycle length coupling.



[9-1] Synchronism with identical cycle (360/360)

MP_IrSetGearboxPosIn: Input position of the measuring system adaptation

MP_IrSetGearboxPosOut: Output position of the measuring system adaptation

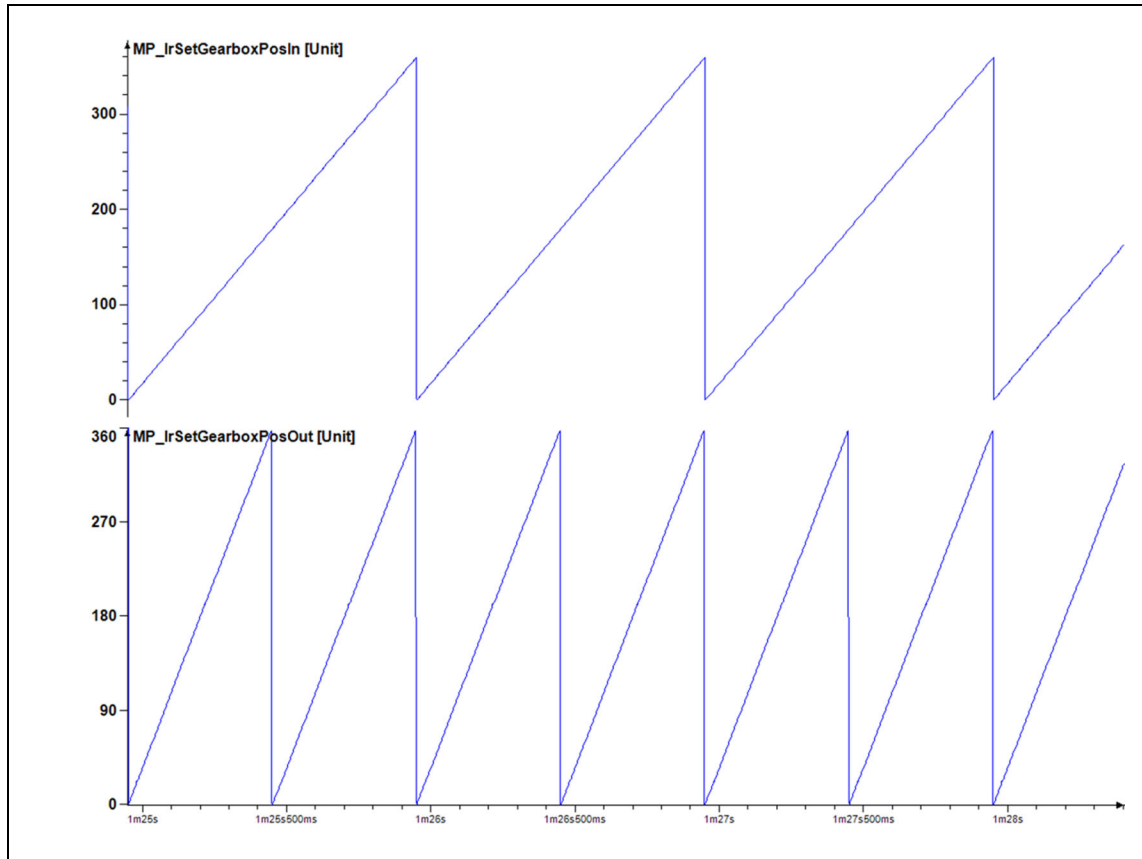


[9-2] Synchronism with different cycle lengths (450/360)

MP_IrSetGearboxPosIn: Input position of the measuring system adaptation

MP_IrSetGearboxPosOut: Output position of the measuring system adaptation

The transmission ratio is freely adjustable via a virtual gearbox factor (*scPar.dwGearingNumerator* and *scPar.dwGearingDenominator*). If, for example, the gearbox factor is set to 2:1 (*scPar.dwGearingNumerator* = 2, *scPar.dwGearingDenominator* = 1), the synchronous behavior will look like this:



[9-3] Synchronism with gearbox factor $\neq 1$ (here 2:1)

MP_IrSetGearboxPosIn: Input position of the measuring system adaptation

MP_IrSetGearboxPosOut: Output position of the measuring system adaptation

Parameters relevant for the function

The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- *scPar.dwGearingNumerator*, DWORD, initial value: 1
- *scPar.dwGearingDenominator*, DWORD, initial value: 1

9.1.2 Position synchronism with basic speed

This function can be used to establish synchronism with basic speed between a master axis and a drive axis (slave).

Preconditions

- The technology module is in an error-free state.
- The axis is enabled (*xReadyForMotion* = TRUE).
- *xSyncIn* input = FALSE
- Coupling mode *scPar.eSyncMode* = 0 (ProfGenAbsolute) or 1 (ProfGenRelative)

Type

The basic speed is active as soon as the *IrSetBaseVel* input takes on a non-zero value. Immediately after the controller has been enabled, the drive travels to the selected basic speed by means of S-rounding and the technology module changes to the "BASEVELOCITY" state.

From this state, the speed of the slave axis can be synchronized to the master speed of the master axis via the clutch function.

Clutch-in starts at any position when the *xSyncIn* input is set to TRUE.

When declutching by means of *xSyncIn* = FALSE, the drive decelerates at *scPar.IrSyncDec* to the basic speed *IrSetBaseVel*.

The *xBaseVelocityActive* output shows whether a basic speed is active.

Inputs relevant for the function

The inputs are located in the [L TS2P_SyncAndCorrectionState](#) (16) function block.

- *xSyncIn*
- *IrSetBaseVel*

Outputs relevant for the function

The outputs are located in the [L TS2P_SyncAndCorrectionState](#) (16) function block.

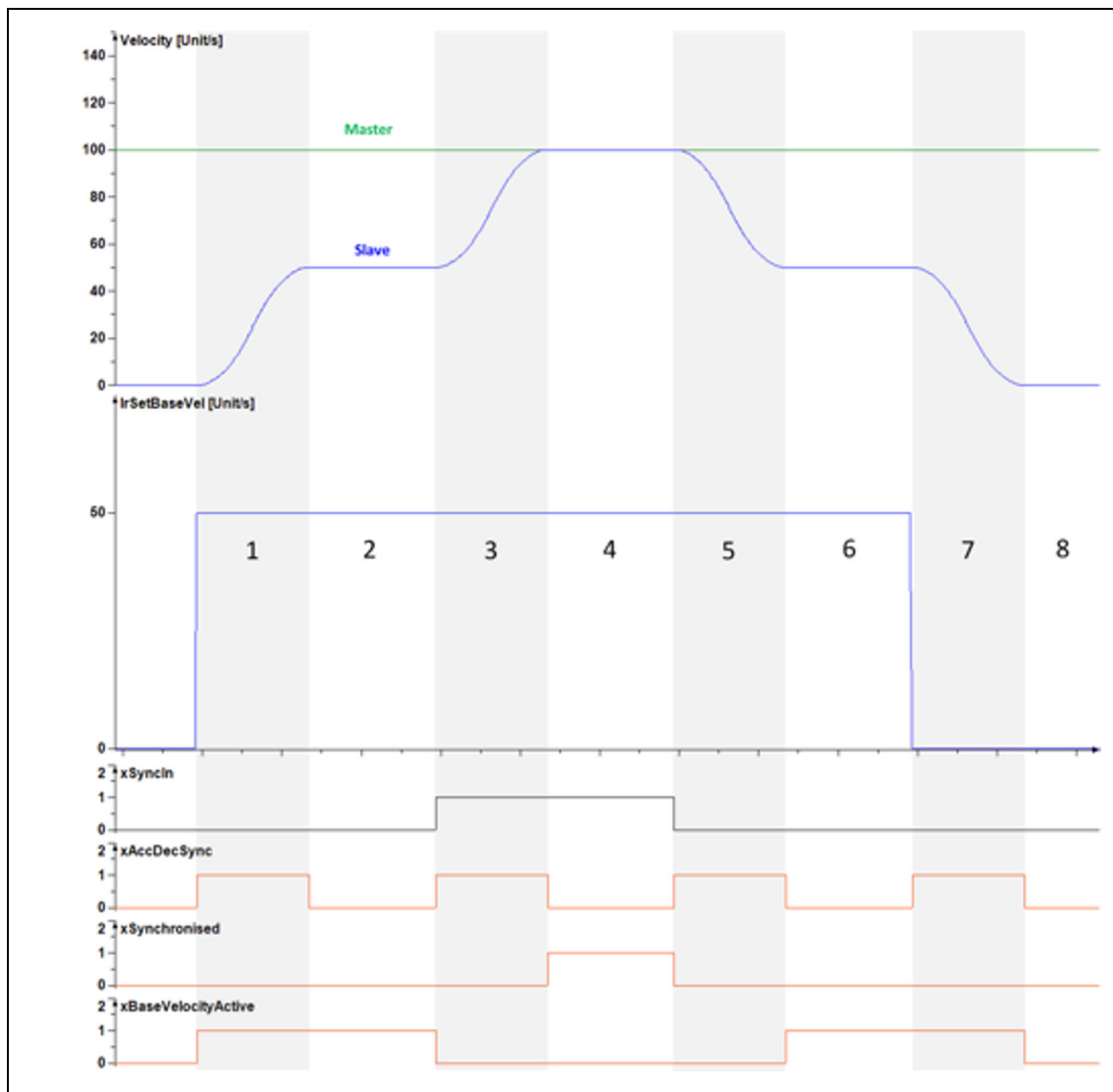
- *xBaseVelocityActive*

Parameters relevant for the function

The parameters are located in the [L TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- *scPar.IrSyncVel*, LREAL, initial value: 100 units/s
- *scPar.IrSyncAcc*, LREAL, initial value: 1 000 units/s²
- *scPar.IrSyncDec*, LREAL, initial value: 1 000 units/s²
- *scPar.IrSyncJerk*, LREAL, initial value: 1 000 000 units/s³

Example



[9-4] Clutch-in process (xSyncln = TRUE) with basic speed IrSetBaseVel = 50

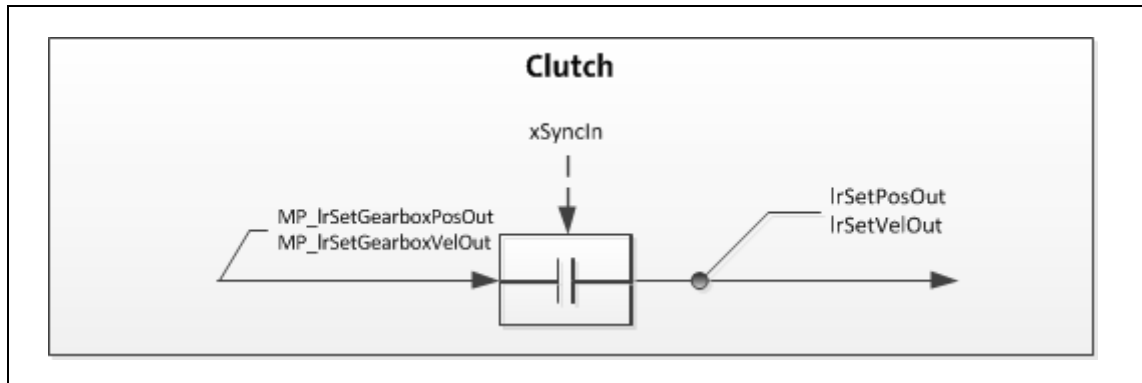
No	xSyncln	xAccDecSync	xSynchronised	xBaseVelocityActive	eTMState	Description
1	FALSE	FALSE → TRUE	FALSE	FALSE → TRUE	Standby → AccDecSync	The axis accelerates from standstill to a basic speed of 50 units/s.
2	FALSE	TRUE → FALSE	FALSE	TRUE	AccDecSync → BaseVel	The axis moves at a constant basic speed.
3	FALSE → TRUE	FALSE → TRUE	FALSE	TRUE → FALSE	AccDecSync → BaseVel	The axis synchronizes to the master speed.
4	TRUE → FALSE	TRUE → FALSE	FALSE → TRUE	FALSE	AccDecSync → Synchroni	The axis moves synchronously to the master axis.
5	TRUE	FALSE → TRUE	TRUE → FALSE	FALSE	Synchronised → AccDecSync	The coupling to the master axis is released and the axis decelerates to the basic speed.
6	FALSE	TRUE → FALSE	FALSE	FALSE → TRUE	AccDecSync → BaseVe	The axis moves at a constant basic speed.
7	FALSE	FALSE → TRUE	FALSE	TRUE	TRUE	The axis decelerates to standstill.
8	FALSE	TRUE → FALSE	FALSE	TRUE → FALSE	TRUE → FALSE	Standstill

9.2 Position coupling

This function synchronizes the setpoint position of the drive (slave axis) to the position setpoint of the master value (master axis, *scSignalflow.MP_IrSetGearboxPosOut*). The synchronization or coupling process is without a step change.

Coupling variant [Path-based coupling](#) (63)

scPar.eSyncMode = 2 (DistAbsolute) or 3 (DistRelative)

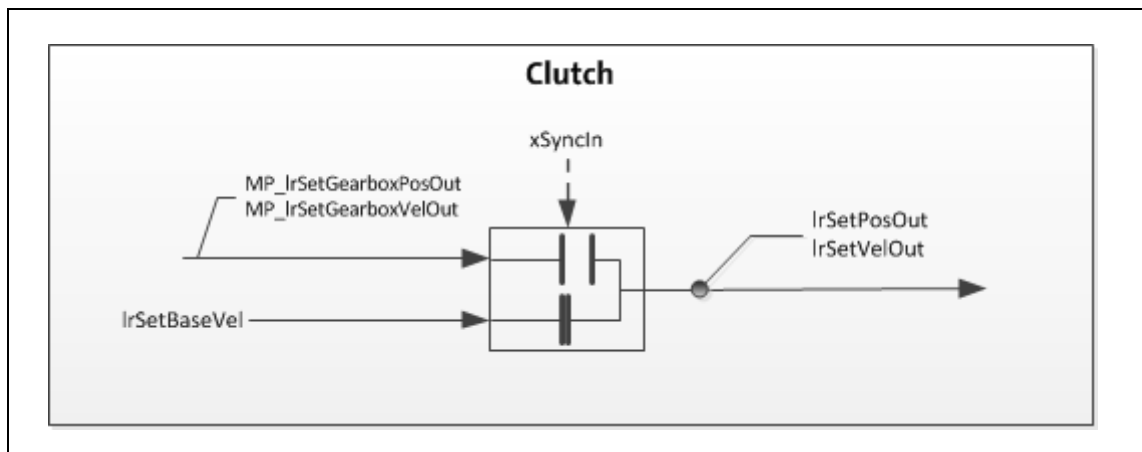


[9-5] Partial signal flow coupling for the coupling mode *scPar.eSyncMode* = 2 (DistAbsolute) or 3 (DistRelative)

Coupling variant [Profile generator-based coupling](#) (63)

scPar.eSyncMode = 0 (ProfGenAbsolute) or 1 (ProfGenRelative)

This coupling type includes the "Coupling to a basic speed" function.



[9-6] Partial signal flow coupling for the coupling mode *scPar.eSyncMode* = 0 (ProfGenAbsolute) or 1 (ProfGenRelative)

9.2.1 Path-based coupling

Preconditions

- The technology module is in an error-free state.
- The axis is enabled (*xReadyForMotion* = TRUE).
- Coupling mode *scPar.eSyncMode* = 2 (DistAbsolute)
- *xSyncIn* input = TRUE

Clutching in

The clutch-in path is specified via the *scPar.lrSlaveSyncInDist* parameter. The clutch-in process begins when the master position is at a distance of *scPar.lrSlaveSyncInDist* in front of the current setpoint position of the drive axis. If the *scPar.lrSlaveSyncInDist* parameter is set to its initial value of 90 units, the clutch-in process is complete after 90 path units.

NOTICE

In case of "Limited" axis setting:

- The drive axis is clutched in if the master position has reached the distance of *scPar.lrSlaveSyncInDist* in front of the current setpoint position of the drive axis.

Declutching

The declutch path is specified via the *scPar.lrSlaveSyncOutDist* parameter. The declutch process begins when the drive is at a distance of *scPar.lrSlaveSyncOutDist* from the 0x5041:035 declutch position. If the *scPar.lrSlaveSyncOutDist* parameter is set to its initial value of 90 units, the declutch process is complete after 90 path units.

NOTICE

In case of "Limited" axis setting:

- So that the drive axis is declutched at the *scPar.lrSlaveSyncOutPos* declutch position, the *scPar.lrSlaveSyncOutDist* distance must be defined in advance.

Asynchronous (relative) clutch-in

scPar.eSyncMode = 1 (ProfGenRelative) or 3 (DistRelative)

- If the master value is at standstill, the drive axis clutches in directly (abruptly) at its current position.
- If the master value moves, the drive axis coupling process starts immediately. (As with a speed coupling)

After the coupling process, the offset between the position of the drive axis and the position of the master axis is constant (*scSignal-Flow.MP_lrPositionOffsetClutch*). This offset can be eliminated by setting the *xSyncPosRestore* input to TRUE. The parameters of the offset function then make it possible to "reduce" the offset to 0.

Inputs relevant for the function

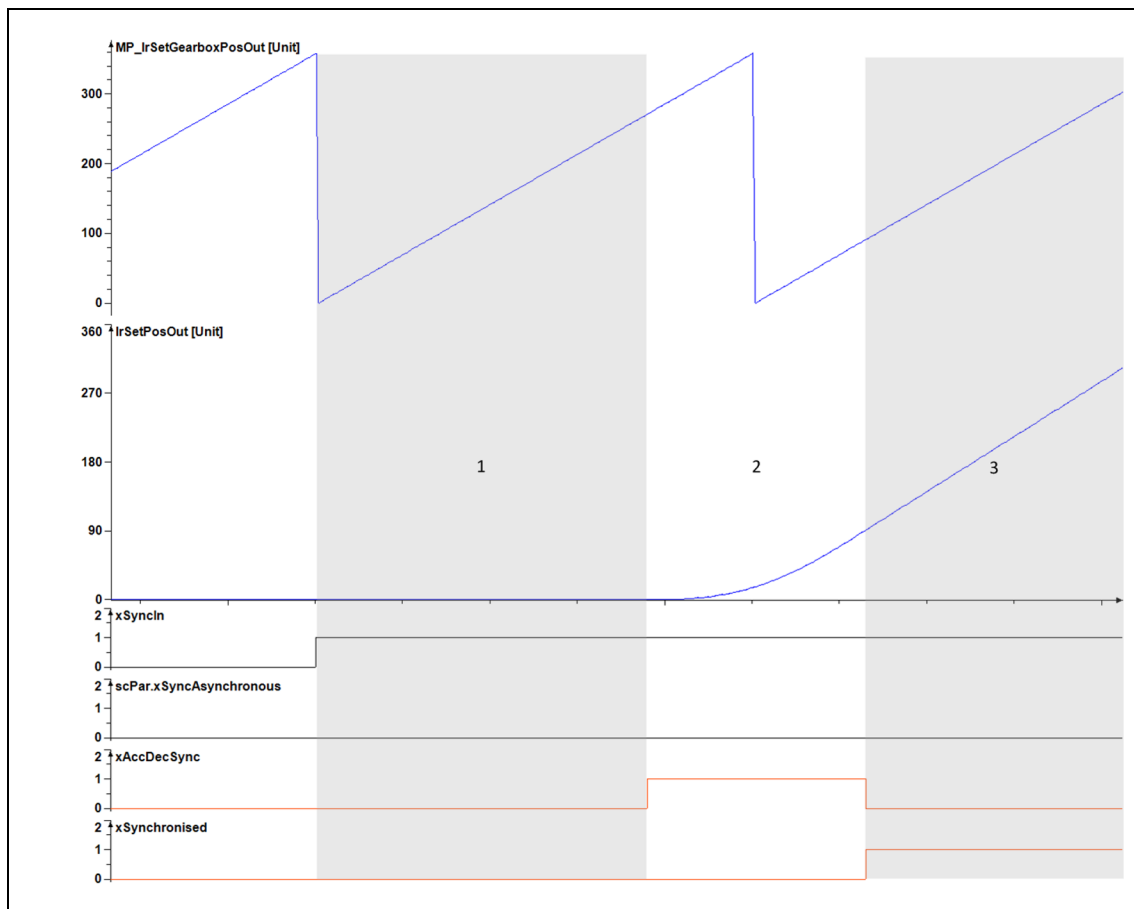
The inputs are located in the [L_TS2P_SyncAndCorrectionState](#) (16) function block.

- *xSyncIn*
- *xSyncPosRestore*

Parameters relevant for the function

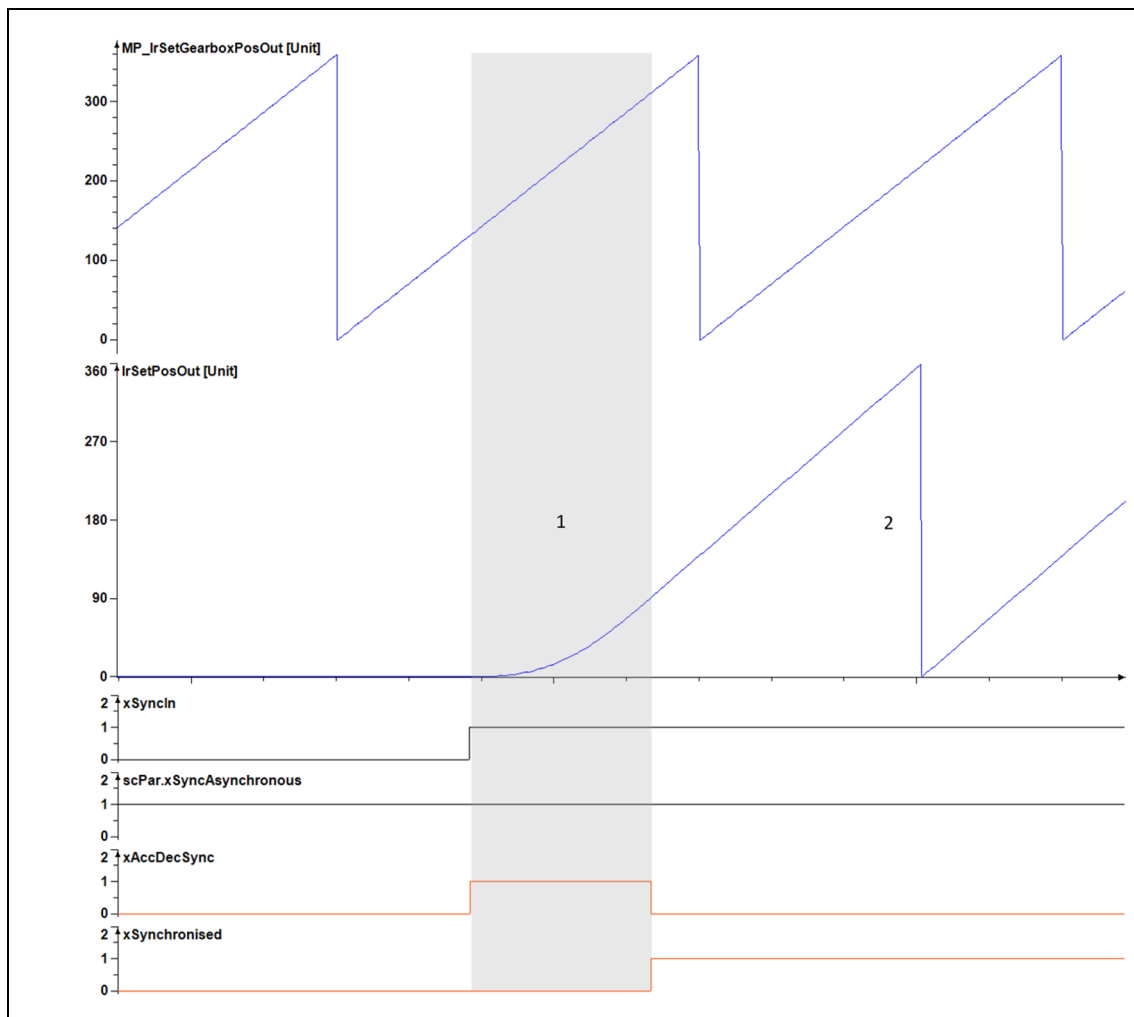
The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- *scPar.lrSlaveSyncInDist*, LREAL, initial value: 90 units
- *scPar.lrSlaveSyncOutDist*, LREAL, initial value: 90 units
- *scPar.lrSlaveSyncOutPos*, LREAL, initial value: 0
- *scPar.eSyncMode*, L_TB2P_SyncMode, initial value: 0 (ProfGenAbsolute)

Example 1: Synchronous clutch-in[9-7] Synchronous clutch-in in coupling mode `scPar.eSyncMode = 5` (Ramp_Dist)

No	<code>xSyncln</code>	<code>xAccDecSync</code>	<code>xSynchronised</code>	<code>eTMState</code>	Description
1	FALSE → TRUE	FALSE	FALSE	STANDBY → ACC_DEC_SYNC	The clutch-in process is started. The axis remains at standstill until the master value is in the right position.
2	TRUE	FALSE → TRUE	FALSE	ACC_DEC_SYNC	The coupling to the master value is established and the axis accelerates until it has reached the master speed.
3	TRUE	TRUE → FALSE	TRUE	ACC_DEC_SYNC → SYNCHRONIZED	The axis is synchronous with the master axis and follows the master value.

Example 2: Asynchronous clutch-in



[9-8] Asynchronous clutch-in in coupling mode `scPar.eSyncMode = 5 (Ramp_Dist)`

No	xSyncIn	xAccDecSync	xSynchronised	eTMState	Description
1	FALSE → TRUE	FALSE → TRUE	FALSE	STANDBY → ACC_DEC_SYNC	The clutch-in process is started. The coupling to the master value is established and the axis accelerates until it has reached the master speed.
2	TRUE	TRUE → FALSE	TRUE	ACC_DEC_SYNC → SYNCHRONIZE	The velocity of the axis is synchronized to the master axis and follows the master value. A position offset has built up between the master axis and the slave axis due to the asynchronous coupling.

9.2.2 Profile generator-based coupling

This clutch-in and declutch variant is independent of a master value movement. This means that the drive axis can also be synchronized if the master value is stationary.

Preconditions

- The technology module is in an error-free state.
- The axis is enabled (*xReadyForMotion* = TRUE).
- Coupling mode *scPar.eSyncMode* = 0 (ProfGenAbsolute)
- *xSyncIn* input = TRUE

Clutching in

The drive axis clutches in from its current position to the master position via the profile generator (see parameters below). For modulo axes, the movement is executed within the slave cycle.

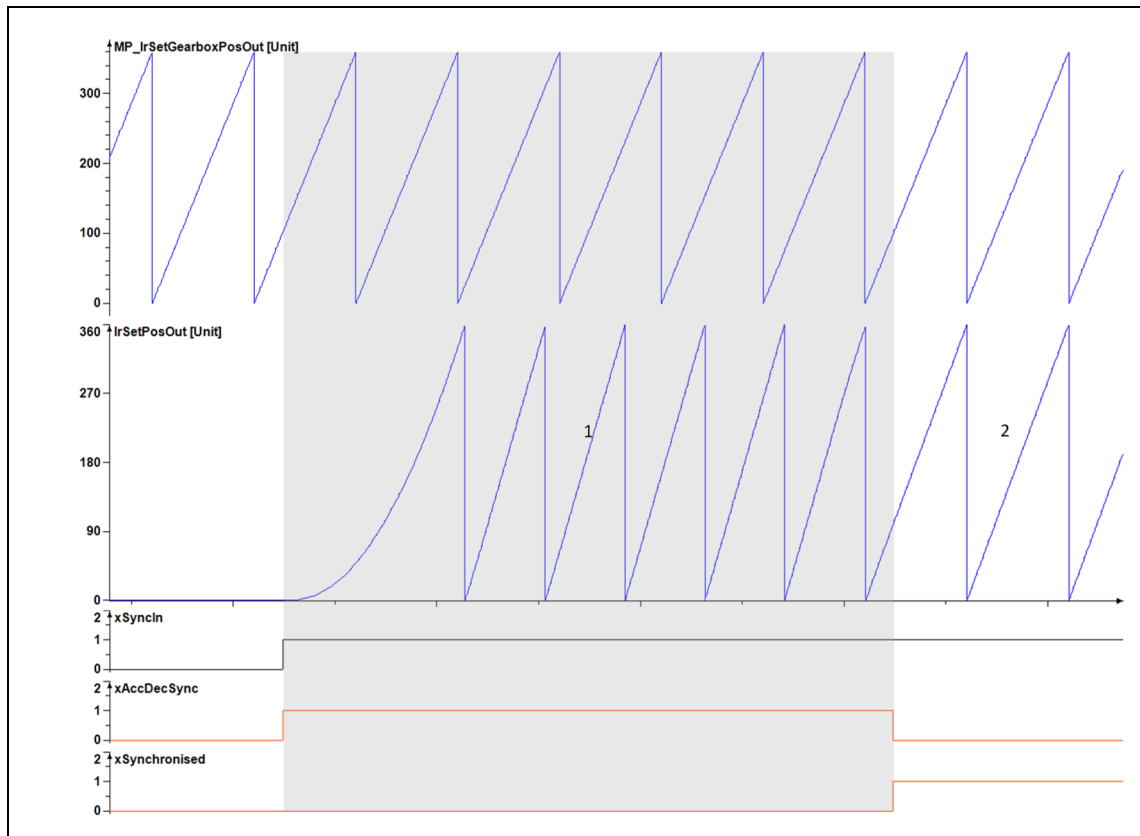
The resulting speed of the drive axis in the clutch-in phase results from the sum of the speed of the master axis and the allowed offset speed (*scPar.lrSyncVel* parameter). In this way a coupling procedure is always carried out sub-synchronously when the clutch-in speed is set (*lrSyncVel* = 0). In this case the slave axis can reach the maximum master speed.

If the clutch-in speed is set higher than '0', clutch-in takes place over the shortest clutch-in time. This means that clutching in can take place both undersynchronously and oversynchronously. The respective behaviour depends on the start conditions at the moment of the clutching in request.

The acceleration of the slave axis in the clutch-in phase also results from the sum of the acceleration of the master value and the acceleration and deceleration of the coupling (parameters *scPar.lrSyncAcc*, *scPar.lrSyncDec*).

The *eAllowedDirection* parameter specifies the direction in which the clutching in procedure may be started.

Allowed clutch-in direction <i>eAllowedDirection</i>	Master speed	Clutch-in possible
1: only positive direction of rotation	positive	Yes
2: only negative direction of rotation	positive	No
0: positive and negative direction of rotation	positive	Yes
1: only positive direction of rotation	negative	No
2: only negative direction of rotation	negative	Yes
0: positive and negative direction of rotation	negative	Yes

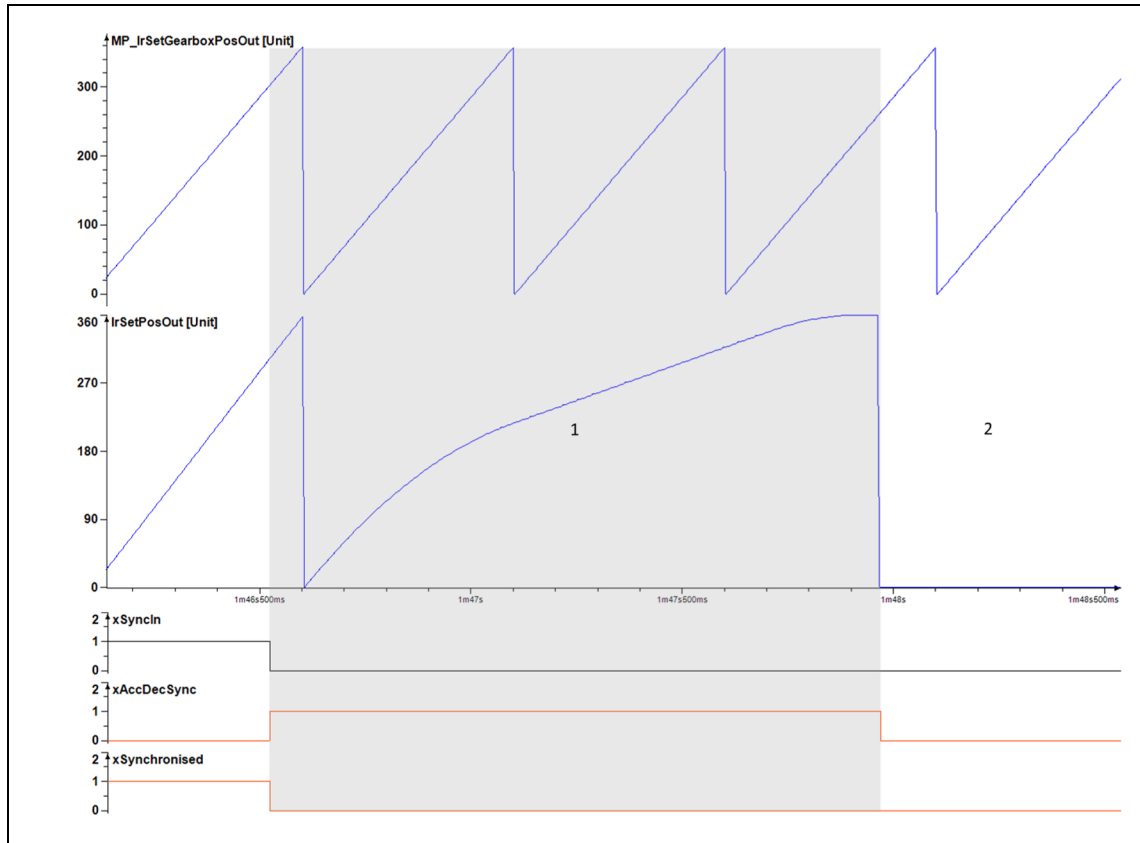


[9-9] Clutch-in in coupling mode `scPar.eSyncMode = 0` (ProfGenAbsolute)

No	xSyncln	xAccDecSync	xSynchronised	eTMState	Description
1	FALSE → TRUE	FALSE → TRUE	FALSE	STANDBY → ACC_DEC_SYNC	The clutch-in process is started. The coupling to the master value is established and the axis accelerates until it has reached the master speed.
2	TRUE	TRUE → FALSE	TRUE	ACC_DEC_SYNC → SYNCHRONIZED	The axis is synchronous with the master axis and follows the master value.

Declutching

The profile-controlled declutching brings the drive axis from its current position to a standstill with the parameters *scPar.lfSyncAcc*, *scPar.lfSyncDec* and *scPar.lfSyncJerk*. The *scPar.lfSlaveSyncOutPos* parameter defines the stopping position of the drive axis.



[9-10] Synchronous clutch-in in coupling mode *scPar.eSyncMode* = 5 (Ramp_Dist)

No	xSyncIn	xAccDecSync	xSynchronised	eTMState	Description
1	TRUE⇒FALSE	FALSE⇒TRUE	TRUE⇒FALSE	SYNCHRONIZED → ACC_DEC_SYNC	The declutch process is active and the axis decelerates to the declutch position.
2	FALSE	TRUE⇒FALSE	FALSE	ACC_DEC_SYNC → STANDBY	The axis is in the declutch position.

Parameters relevant for the function

The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- *scPar.eAllowedDirection*, *L_TB2P_AllowedDirection*, initial value: 0 (both)
- *scPar.lfSyncVel*, LREAL, initial value: 100 units/s
- *scPar.lfSyncAcc*, LREAL, initial value: 1 000 units/s²
- *scPar.lfSyncDec*, LREAL, initial value: 1 000 units/s²
- *scPar.lfSyncJerk*, LREAL, initial value: 1 000 000 units/s³

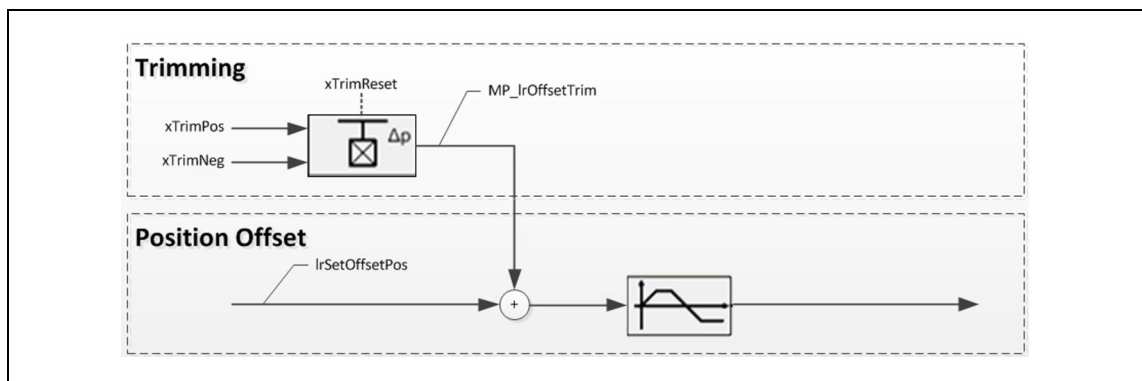
9.3 Position trimming

For trimming the position of the slave axis is adjusted in relation to the master axis.

Preconditions

- The technology module is in an error-free state.
- The axis is enabled ($xReadyForMotion = TRUE$).

Type



[9-11] Signal flow trimming

Trimming per position offset ($scPar.eTrimMode = 1$)

The trimming process corresponds to a relative positioning. With each positive edge at the $xTrimPos$ input, the position to be trimmed is increased by the value of $scPar.IrTrimDist$. With each positive edge at the $xTrimNeg$ input, the position to be trimmed is decreased by the value of $scPar.IrTrimDist$.

Trimming per velocity offset ($scPar.eTrimMode = 0$)

- As long as the $xTrimPos$ input is set to TRUE, the trimming is in the positive direction.
- As long as the $xTrimNeg$ input is set to TRUE, the trimming is in the negative direction.
- If both inputs are TRUE, the trimming is stopped.

The acceleration and velocity of the trimming are superimposed on those of the master axis. The results for the axis to be trimmed are as follows:

Resulting speed of $v_{AxisRes} = v_{MasterAxis} + IrSetOffsetTrimVel$

Resulting acceleration of $a_{AxisRes} = a_{MasterAxis} + IrSetOffsetTrimAcc$

Inputs relevant for the function

The inputs are located in the [L_TS2P_SyncAndCorrectionState](#) (16) function block.

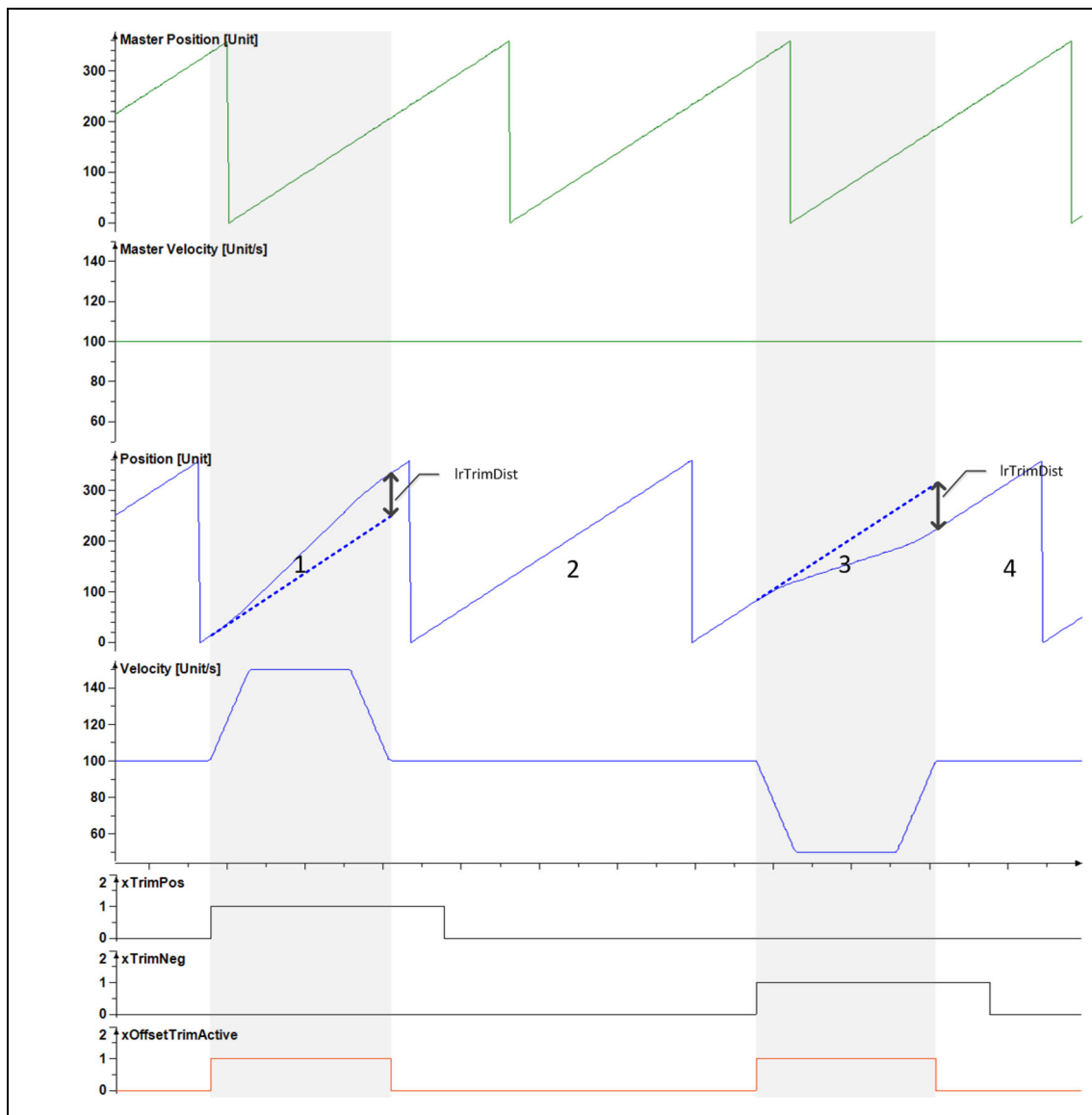
- $xTrimPos$, $xTrimNeg$

Parameters relevant for the function

The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- $scPar.IrTrimDist$, LREAL, initial value: 1 unit
- $scPar.IrSetOffsetTrimVel$, LREAL, initial value: 1 unit
- $scPar.IrSetOffsetTrimAcc$, LREAL, initial value: 1 unit

Example



[9-12] Trimming in the positive and negative directions by 90 units at a maximum trimming speed of 50 units/s in each case

No	xTrimPos	xTrimNeg	xOffsetTrimActive	eTMState	Description
1	FALSE→TRUE	FALSE	FALSE→TRUE	SYNCHRONIZED	Position trimming in the positive direction by IrTrim-Dist = 90 units
2	TRUE→FALSE	FALSE	TRUE→FALSE	SYNCHRONIZED	Synchronous travel
3	FALSE	FALSE→TRUE	FALSE→TRUE	SYNCHRONIZED	Position trimming in the negative direction by IrTrim-Dist = -90 units
4	FALSE	TRUE→FALSE	TRUE→FALSE	SYNCHRONIZED	Synchronous travel

9.4 Position offset via profile generator

This function can be used to specify a position offset via a profile generator.

Preconditions

- The technology module is in an error-free state.

Type

The position offset is transferred to the drive axis via a profile generator. The offset is specified via the *IrSetOffsetPos* input.

The *scPar.eAllowedDirection* parameter is used to define the direction in which the drive may rotate:

- in the positive and negative directions of rotation (0)
- only in the positive direction of rotation (1)
- only in the negative direction of rotation (2)

The basic conditions to calculate the profile are determined via the following parameters:

- *scPar.IrSetOffsetTrimVel*
- *scPar.IrSetOffsetTrimAcc*
- *scPar.IrSetOffsetTrimDec*
- *scPar.IrSetOffsetTrimJerk*

Inputs relevant for the function

The inputs are located in the [L_TS2P_SyncAndCorrectionState](#) (16) function block.

- *IrSetOffsetPos*

Outputs relevant for the function

The outputs are located in the [L_TS2P_SyncAndCorrectionState](#) (16) function block.

- *IrOffsetTotal*

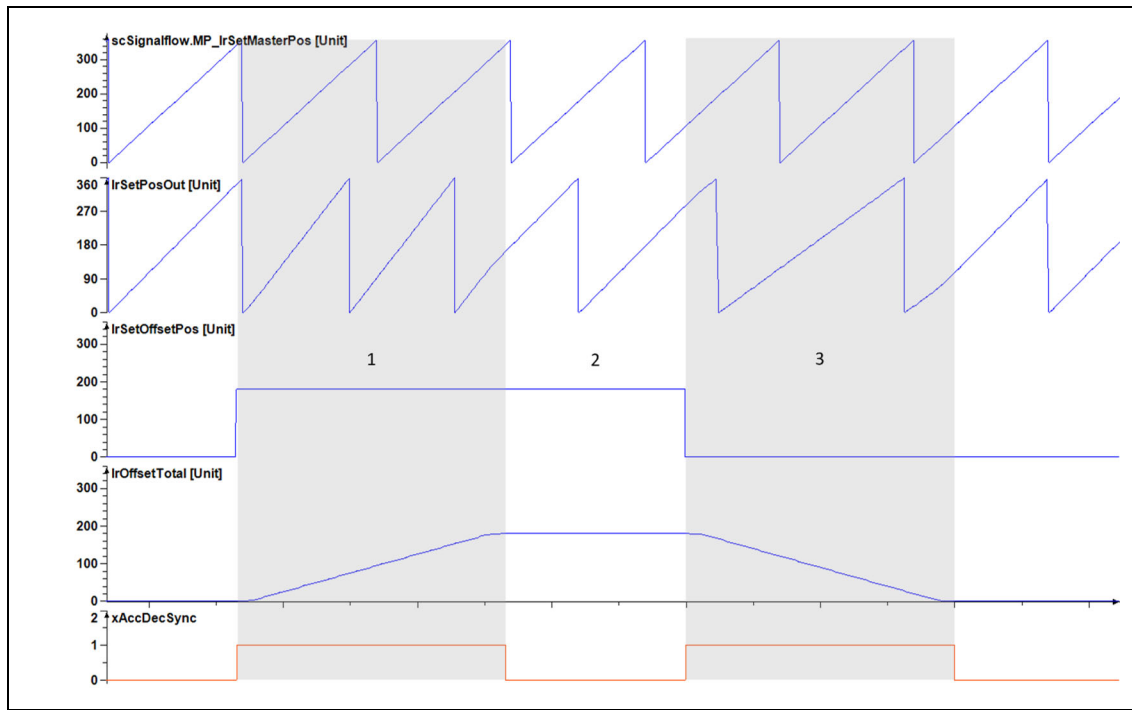
Parameters relevant for the function

The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- *scPar.eAllowedDirection*, initial value: 0
- *scPar.IrTrimDist*, LREAL, initial value: 1 unit
- *scPar.IrSetOffsetTrimVel*, LREAL, initial value: 10 units/s
- *scPar.IrSetOffsetTrimAcc*, LREAL, initial value: 100 units/s²
- *scPar.IrSetOffsetTrimDcc*, LREAL, initial value: 100 units/s²
- *scPar.IrSetOffsetTrimJerk*, LREAL, initial value: 100 000 units/s³

Example:

If, for example, the master value axis is operated in the positive direction, `scPar.eAllowedDirection = 1` (only in the positive direction of rotation) can be used to prevent the drive axis from rotating in the negative direction.



[9-13] Position offset

No.	IrSetOffsetPos	IrOffsetTotal	xOffsetTrimActive	Description
1	0 units -> 200 units	0 units	FALSE → TRUE	The offset is corrected to the IrSetOffsetPos target value (200 units) using the offset and trimming profile parameters via a profile generator.
2	200 units	200 units	TRUE → FALSE	The offset is corrected.
3	200 units -> 0 units	200 units	FALSE → TRUE	The offset is corrected to the IrSetOffsetPos target value (0 units) using the offset and trimming profile parameters via a profile generator.

9.5 Limit setpoints to allowed travel range

In some applications, the tool axis must not move in a certain direction. In this case, the `eAllowedDirection` parameter can be used to block profile generation in the unauthorized direction. The setting prevents an impermissible position offset or engagement.

WARNING!

Injury to persons and damage or destruction of machine parts

Damage or destruction of machine parts can be caused by unpredictable motor movements.

Setting the `eAllowedDirection` parameter does not prevent the total setpoint of the slave axis from acting in an inhibited direction. This could be the case, for example, if the master speed or a speed offset acts in the blocked direction. These setpoint components are not actively limited by this function.

Possible consequences

Death or severe injuries

Protective measures

Check and adjust limits and direction of action for the master speed or speed offset.

To limit the total setpoint of a tool axis to the allowed range, the parameter `xLimitAllowedDirection` = TRUE must be set. In this way, only setpoints that act in the allowed direction are transferred to the slave axis.

The parameter `IrToleranceAllowedDirection` can be used to specify a tolerance range. If the setpoint moves in a blocked direction, the setpoint of the tool axis is "frozen" from that moment.

If the total setpoint then changes in the blocked direction and the tolerance range is violated, the axis changes to the "FailQSP" state.

If this does not happen and the setpoint leaves the tolerance window in the allowed direction, it is passed back to the tool axis.

Parameters relevant for the function

- `eAllowedDirection`, initial value: 0 (positive and negative rotating direction)
- `xLimitAllowedDirection`, initial value: FALSE
- `IrToleranceAllowedDirection`, initial value: 5 units

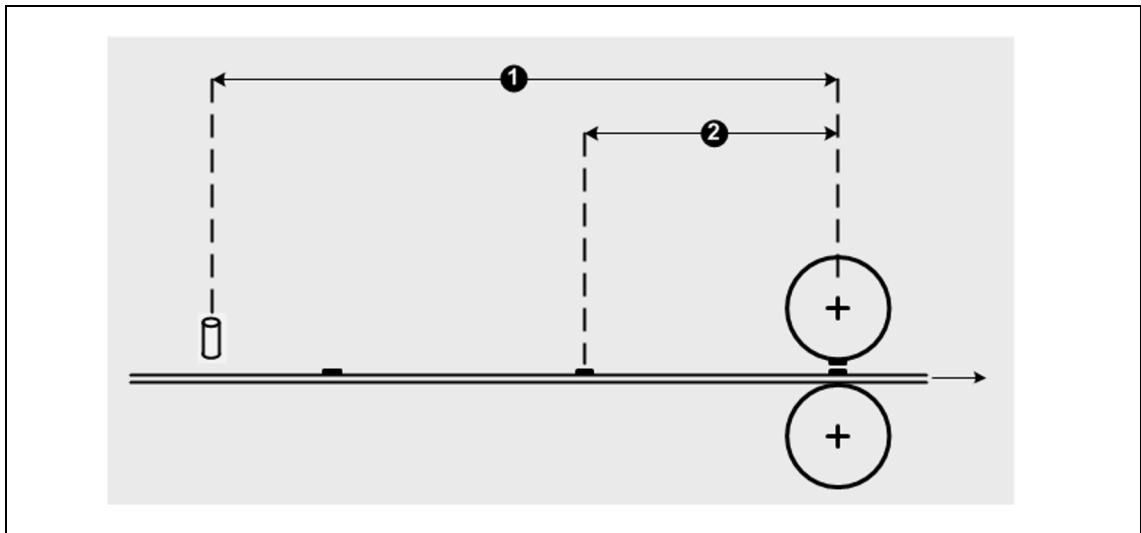
9.6 Touch probe-based master value correction

As a higher-level control loop, the master value correction controls the master position relative to a mark detected on the material (touch probe). This serves to compensate process-related mark shifts on the material with regard to the master position.

The *scPar.IrSensorToolDistMaster* parameter is used to specify the distance between the mark sensor and the position where the tool (e.g. cutting blade, printheads) engages the material. The determination of this parameter also determines the position of the *scSignalFlow.MP_IrMarkSensorSetPos* sensor within the register cycle.

The length of the register cycle is specified in the *scPar.IrMarkDistMaster* parameter. It corresponds to the distance of the marks.

If a mark (touch probe) occurs during the process, the detected register position is compared to the sensor position. The difference between these two positions as per the signal flow variable (measuring points) *MP_IrMarkSensorSetPos* and *MP_IrMarkActTpPos* forms the position error that is output at *IrSetMarkCorrOut*.



[9-14] Master value correction parameterization

(1): *scPar.IrSensorToolDistMaster*

(2): *scPar.IrMarkDistMaster*

Inputs relevant for the function

The inputs are located in the [L_TS2P_SyncAndCorrectionState](#) (16) function block.

- *scTPRecData_MarkSensor*

Outputs relevant for the function

The outputs are located in the [L_TS2P_SyncAndCorrectionState](#) (16) function block.

- *IrSetMarkCorrOut*

Parameters relevant for the function

The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- *scPar.IrSensorToolDistMaster*
- *scPar.IrMarkDistMaster*

Measuring points relevant for the function

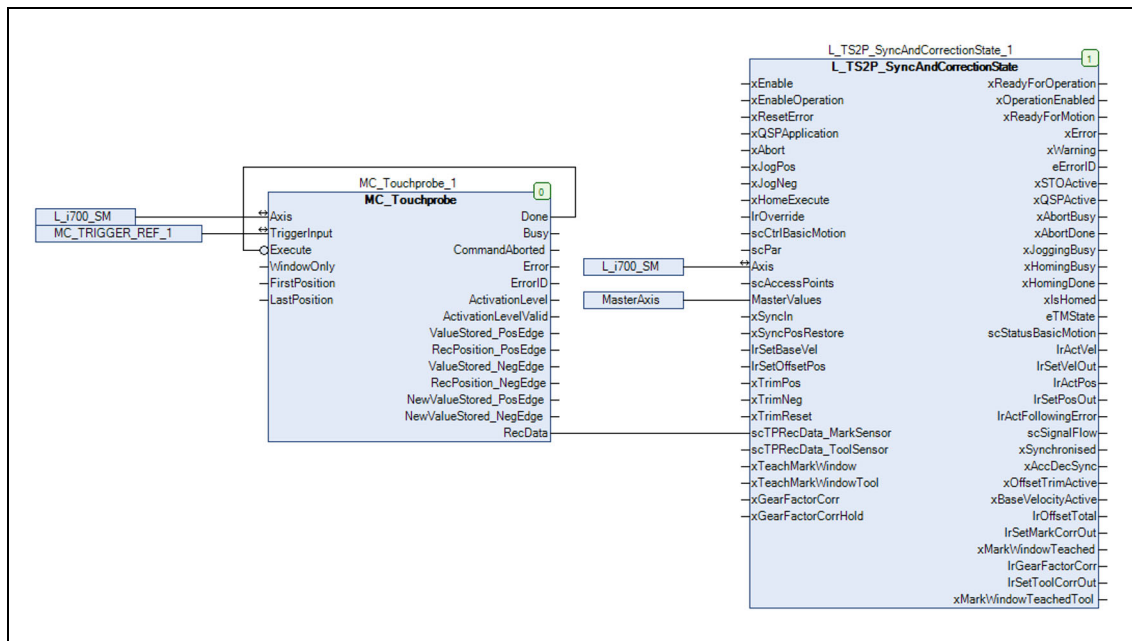
The measuring points are located in the [L TS2P_scSF_SyncAndCorrectionState](#) (30) structure.

- *MP_IrIntegratorRegPos*
- *MP_IrCorrRegPos*
- *MP_IrMarkSensorSetPos*
- *MP_IrMarkActTpPos*
- *MP_xMarkReceived*
- *MP_IrActMarkError*
- *MP_IrActMarkDist*

9.6.1 Touch probe connection

In order to be able to use the touch probe functionality, an instance of the **MC_Touchprobe** function block must be used in the application. The user can use the parameters of this function block to select the touch probe source that should be used for the mark sensor.

Touch probes are connected to the technology module via the *scTpRecData_MarkSensor* input structure. This input is connected to the *RecData* output of the **MC_Touchprobe** block:



[9-15] Touch probe connection interconnection example

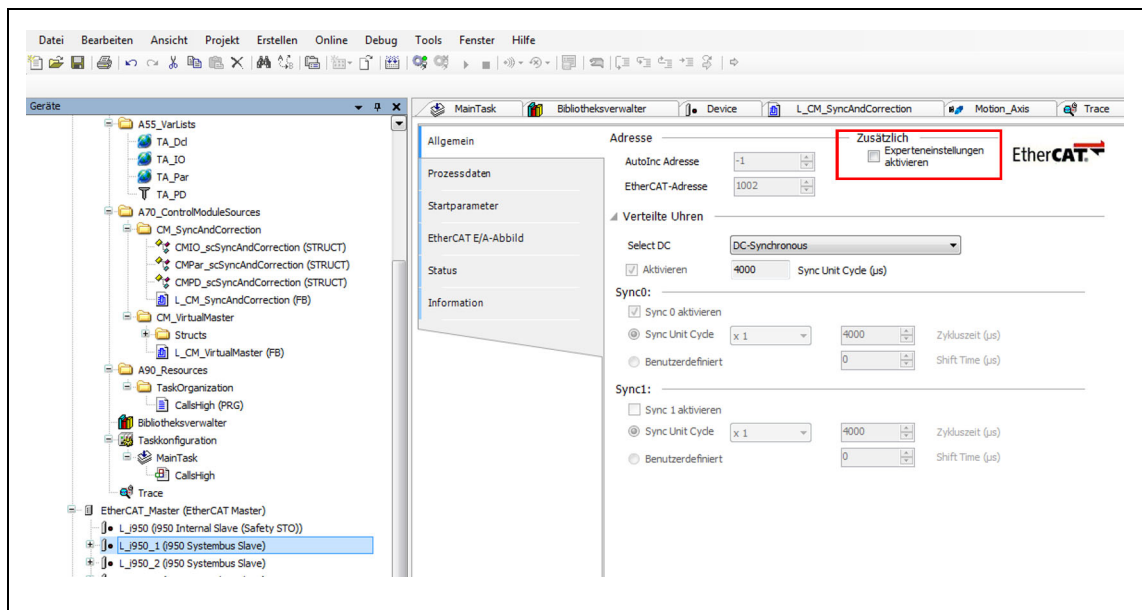
A digital slave axis input was used for the above example. The triggering mode of the **MC_Touchprobe** block is set to single measurement. As soon as a touch probe signal has been detected, a new touch probe measurement is requested via the inverted *MC_Touchprobe_1.Done* signal.

NOTICE

The touch probe evaluation of this technology module functions via a time stamp. If the technology module is used on a controller, the EtherCAT time stamp of the digital input used must be added to the EtherCAT mapping by the user. This step is not necessary for the i950 PLC device. Here all necessary data are already transferred to the application.

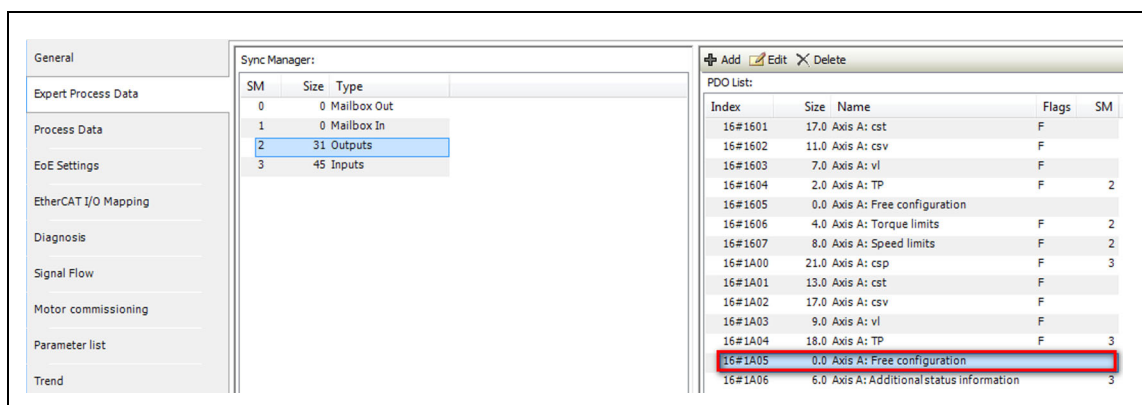
Procedure for adding a time stamp to the mapping

1. Activate expert settings.



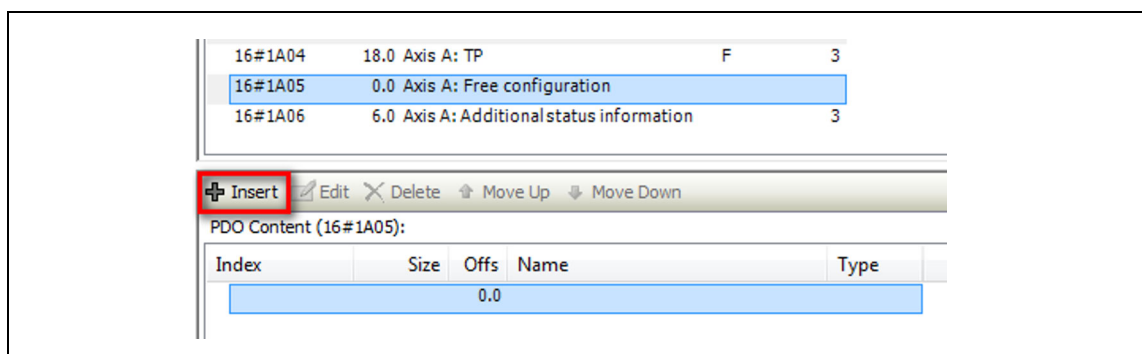
[9-16] Setting in the »PLC Designer« in the device tree in the interface of the respective EtherCAT slave

2. Select the mapping to be expanded.

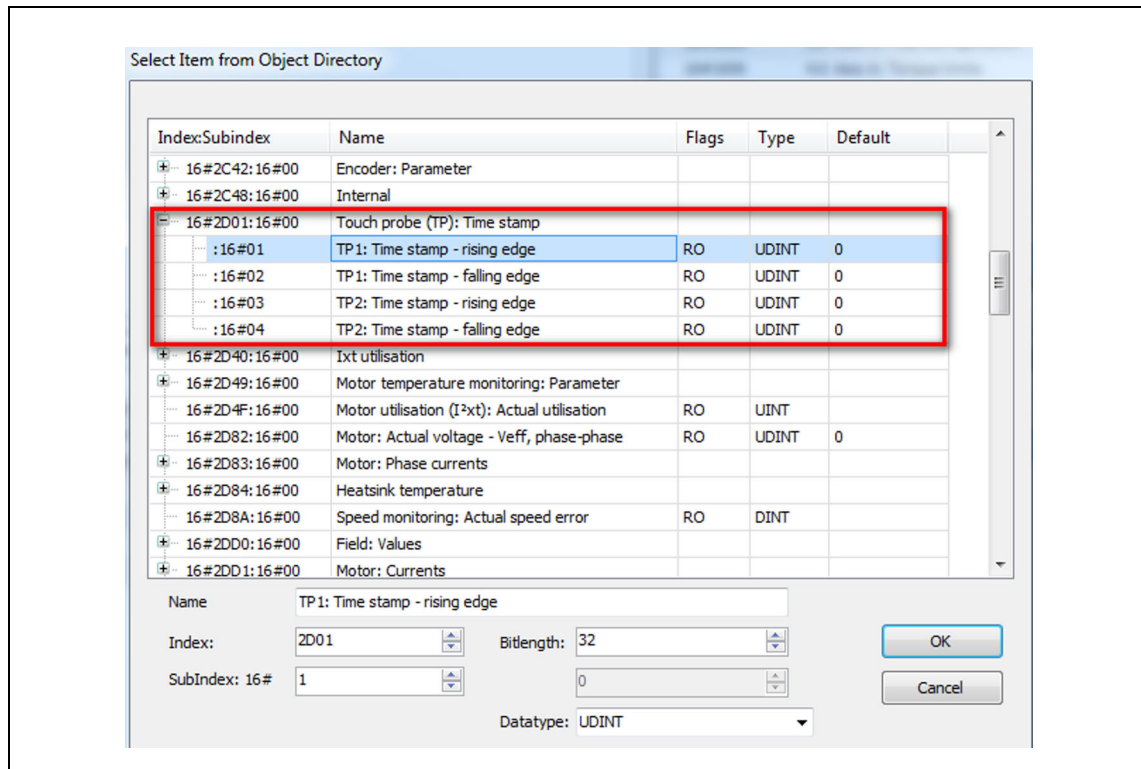


[9-17] Touch probe connection interconnection example

3. Add time stamp object to the free mapping.

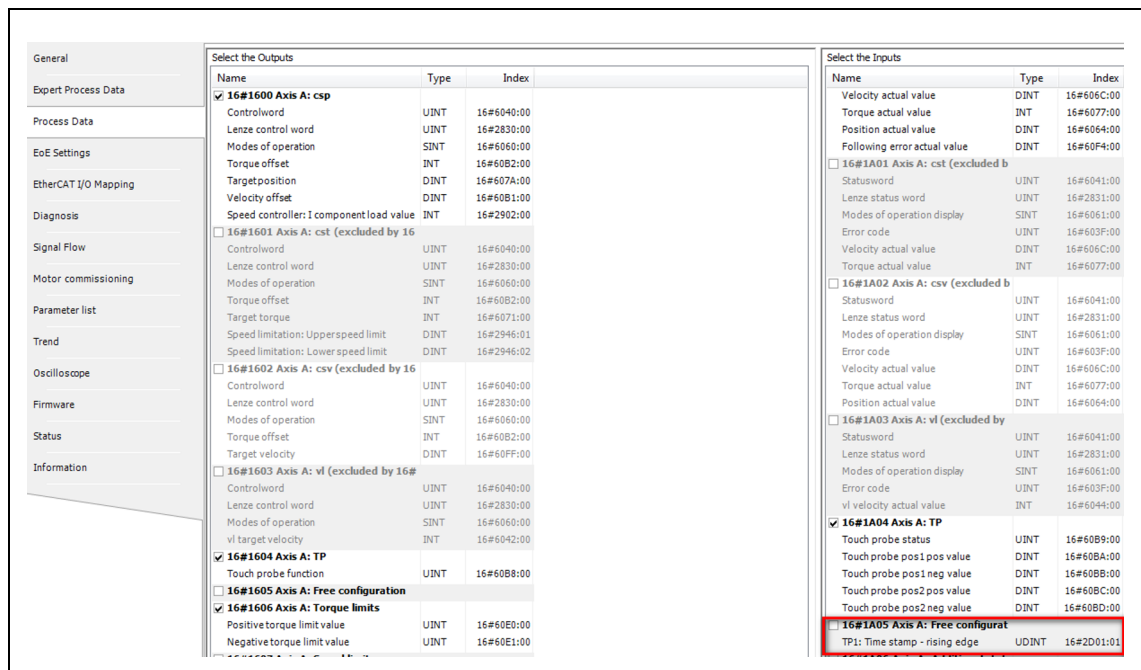


[9-18] Add objects 1



[9-19] Add objects 2

4. Add free mapping objects to the EtherCAT mapping.



[9-20] Touch probe connection interconnection example

9.6.2 Correction window and limitation of the master value correction

The position correction takes place within a correction window.

The position of this window is defined via the parameters *scPar.IrLowerCorrPosMaster* and *scPar.IrUpperCorrPosMaster*; the unit of both parameters is the unit of the measuring system used (e.g. master value in "mm", drive axis in "°").

The *scPar.eSourceForCorrWindow* parameter is used to specify whether the position of the correction window relates to the corrected master position or the position of the drive axis.

The determined position error can be limited to a maximum value in order to keep accelerations during the correction movements low, for instance. For the limitation, a parameter is available for the positive limit value (*scPar.IrMaxCorrPosMaster*) and the negative limit value (*scPar.IrMaxCorrNegMaster*) respectively. The *IrSetMarkCorrOut* output displays the limited position error.

The resulting position and speed of the corrected master value are available in the signal flow variables (measuring point) *MP_IrSetCorrPosOut* and *MP_IrSetCorrVelOut*.

Outputs relevant for the function

The outputs are located in the [L_TS2P_SyncAndCorrectionState](#) (16) function block.

- *IrSetMarkCorrOut*

Parameters relevant for the function

The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- *eSourceForCorrWindow*, initial value: Slave[2]
- *IrLowerCorrPosMaster*, initial value: 90 [unit]
- *IrUpperCorrPosMaster*, initial value: 180 [unit]
- *IrMaxCorrPosMaster*, initial value: 30 [unit]
- *IrMaxCorrNegMaster*, initial value: -30 [unit]

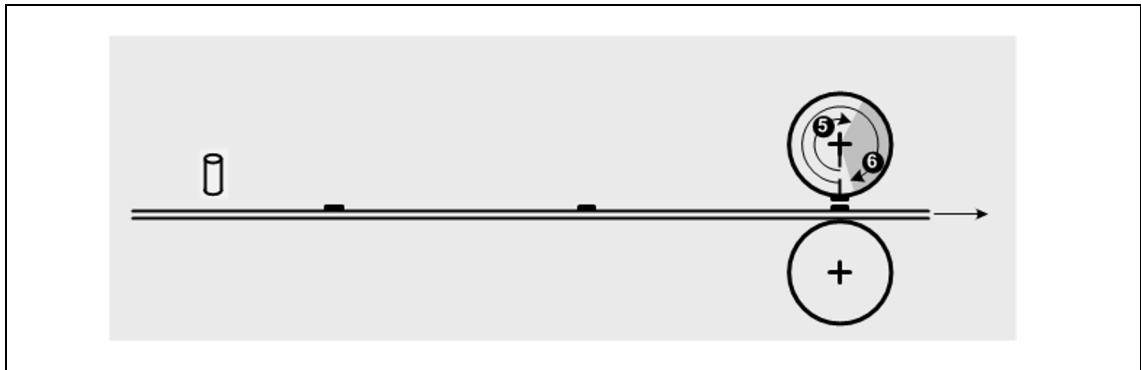
Measuring points relevant for the function

The measuring points are located in the [L_TS2P_scSF_SyncAndCorrectionState](#) (30) structure.

- *MP_xCorrActive*
- *MP_xCorrLimited*

Example: 1

scPar.eSourceForCorrWindow = Act. tool position (correction window in tool units)



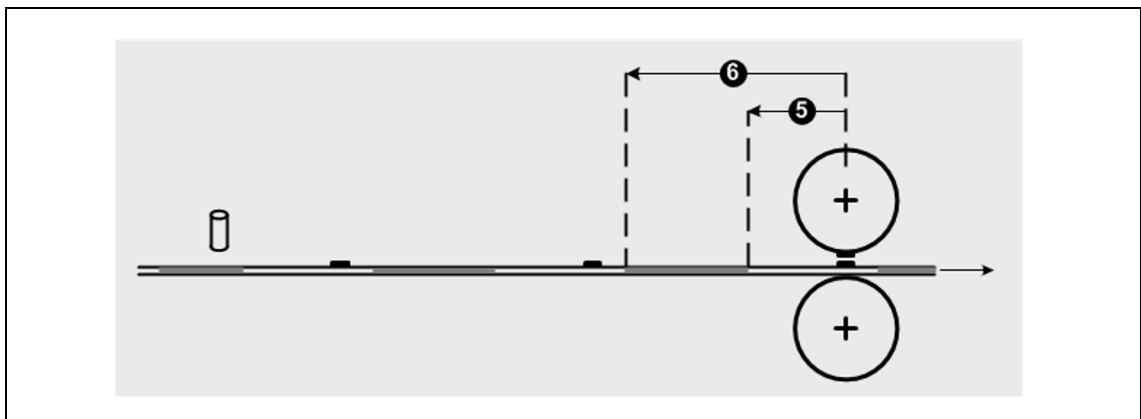
[9-21] Correction window relating to the position of the drive axis

(5): *scPar.lrlowerCorrPosMaster*

(6): *scPar.lrupperCorrPosMaster*

Example 2

scPar.eSourceForCorrWindow = Corr. master position (correction window in material units)



[9-22] Correction window relating to the position of the corrected master axis

(5): *scPar.lrlowerCorrPosMaster*

(6): *scPar.lrupperCorrPosMaster*

9.6.3 Mark window master value correction

In the case of print images, it can happen, for example, that not only the print marks themselves but also parts of the print image or other interfering signals are located in the scanning field of the mark sensor.

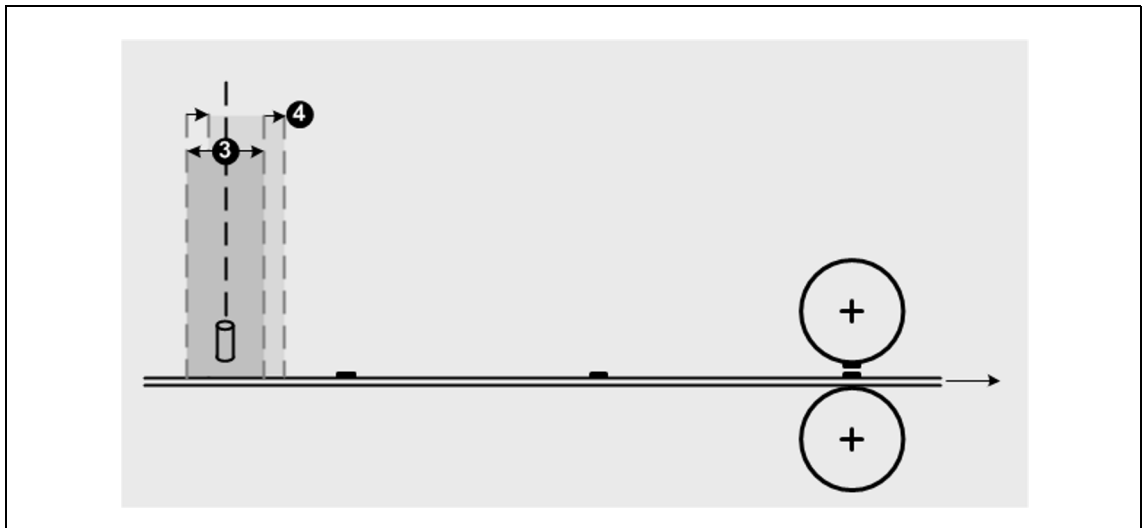
A mark window can be activated to prevent the mark sensor from being triggered unintentionally. When the *scPar.xEnableMarkWindowMaster* parameter = TRUE, the mark window function is active.

For the master value correction to function properly when the mark window is activated, the mark window has to be taught once. That can happen in the declutched state (master value is moving, drive axis is at standstill) as well as when the drive axis is in the clutched-in state (master value and drive axis are moving).

During teaching of the mark window, the *scPar.lrMarkWindowSizeMaster* mark window is placed symmetrically around the detected mark.

- If this happens when the drive axis is in the declutched state, the master position is set to the optimum position when the mark is detected.
- If the mark window is taught when the drive axis is in the clutched-in state, only the mark window is activated when the first mark occurs. The master position is not set and is corrected within the correction window via the master value correction.

The mark window can be shifted via *scPar.lrSetOffsetMarkWindowMaster*.



[9-23] Correction window relating to the position of the corrected master axis

(5): *scPar.lrMarkWindowSizeMaster*

(6): *scPar.lrSetOffsetMarkWindowMaster*

Inputs relevant for the function

The inputs are located in the [L_TS2P_SyncAndCorrectionState](#) (16) function block.

- *xTeachMarkWindow*

Outputs relevant for the function

The outputs are located in the [L_TS2P_SyncAndCorrectionState](#) (16) function block.

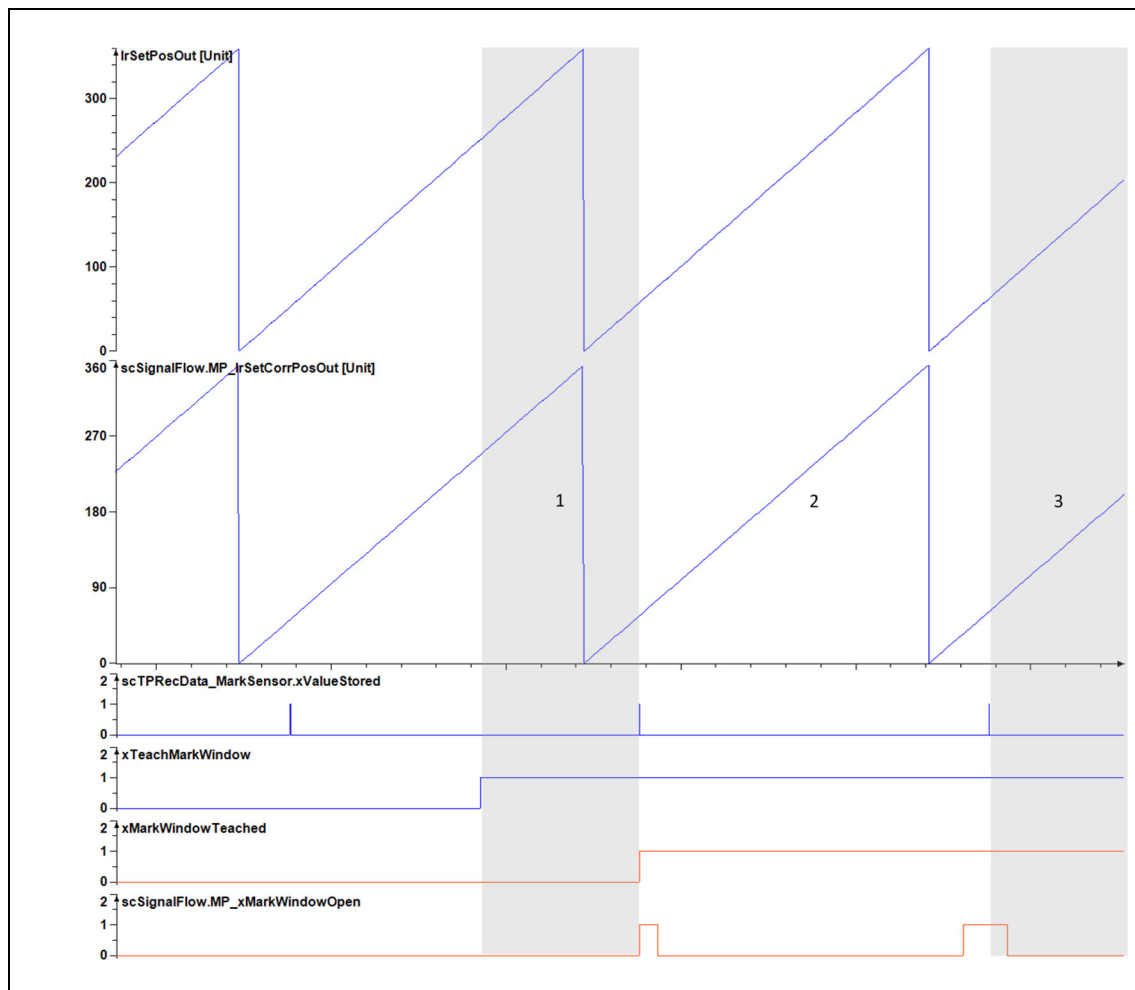
- *xMarkWindowTeached*

Parameters relevant for the function

The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

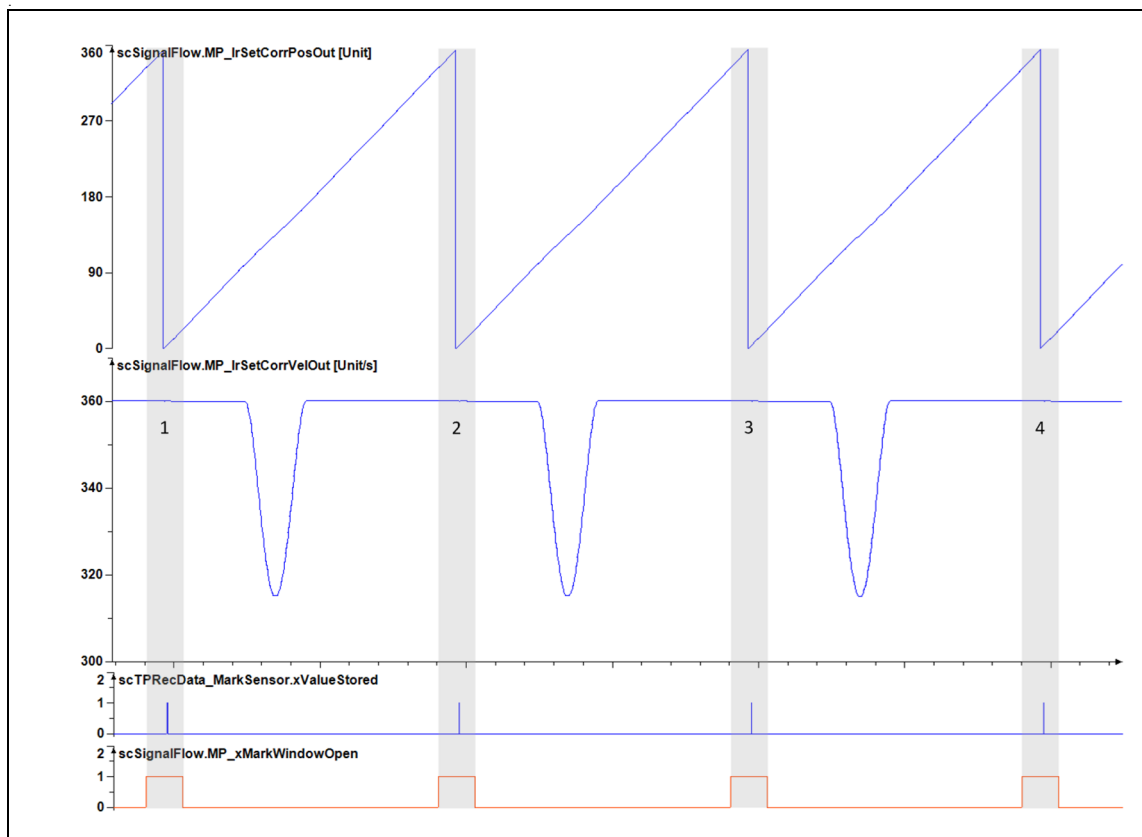
- *xEnableMarkWindowMaster*, BOOL, initial value: FALSE
- *IrMarkWindowSizeMaster*, LREAL, initial value: 90 [unit]
- *IrSetOffsetMarkWindowMaster*, LREAL, initial value: 0 [unit]

Signal characteristic



[9-24] Signal characteristic when the teaching function is executed (*xTeachMarkWindow* = TRUE)

No.	scTPRecData_MarkSensor.xValueStored	xTeachMarkWindow	xMarkWindow Teached	scSignal-Flow.MP_xMarkWindowOpen	Description
1	FALSE	FALSE → TRUE	FALSE	FALSE	Mark window teaching is started.
2	FALSE → TRUE	TRUE	FALSE → TRUE	FALSE → TRUE	The first mark after the teach process has started opens the mark window.
3	FALSE → TRUE	TRUE	TRUE	TRUE	The mark window is open. The next mark is detected within the mark window.



[9-25] Synchronous clutch-in in coupling mode `scPar.eSyncMode = 5` (Ramp_Dist)

No	<code>scTPRecData_MarkSensor.xValueStored</code>	<code>scSignalFlow.MP_xMarkWindowOpen</code>	Description
1	FALSE → TRUE → FALSE	FALSE → TRUE	The first mark occurs within the mark window.
2	FALSE → TRUE → FALSE	FALSE → TRUE	The second mark occurs within the mark window.
3	FALSE → TRUE → FALSE	FALSE → TRUE	The third mark occurs within the mark window.
4	FALSE → TRUE → FALSE	FALSE → TRUE	The fourth mark occurs within the mark window.

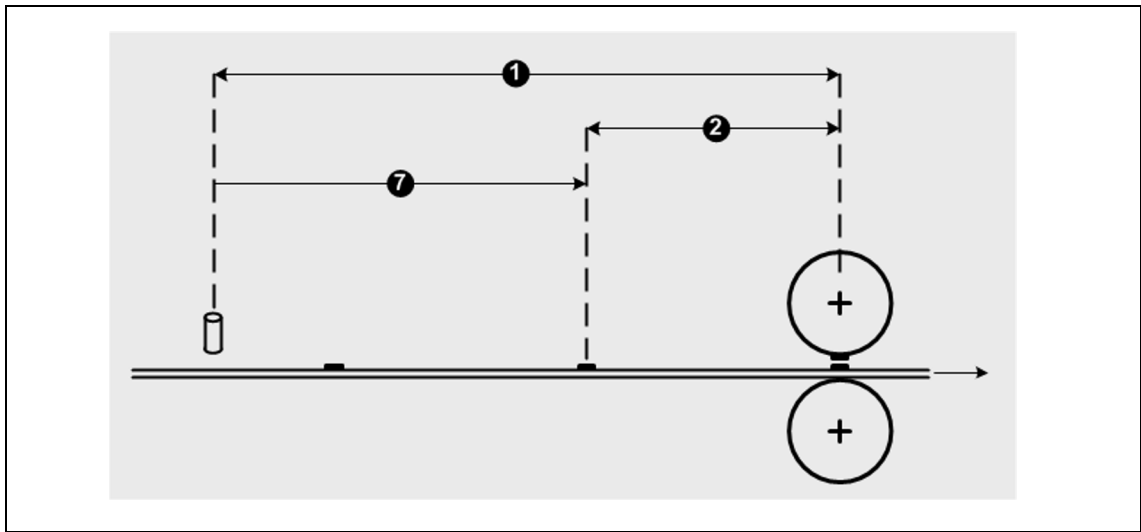
9.6.4 Mark stack master value correction

The mark stack is always required if, due to the design, the sensor cannot be mounted within one cycle in front of the tool (e.g. cutting blade, printheads); i.e. $scPar.IrSensorToolDistMaster > scPar.IrMarkDistMaster$.

The function ensures that the detected mark is only evaluated when it is located in the cycle in front of the tool. Simply put, the mark sensor is moved virtually to the position exactly one cycle in front of the tool. The tool may be mounted a maximum of 64 cycles in front of the tool. The length of the mark stack is calculated as follows:

Mark stack length = $scPar.IrSensorToolDistMaster - scPar.IrMarkDistMaster$

The aim should always be to mount the mark sensor as close as possible to the tool. The farther away the mark sensor is mounted from the axis, the more changes in the material flow remain undetected, causing cutting inaccuracies.



[9-26] Systematic representation of the mark register

(1): $scPar.IrSensorToolDistMaster$

(2): $scPar.IrMarkDistMaster$

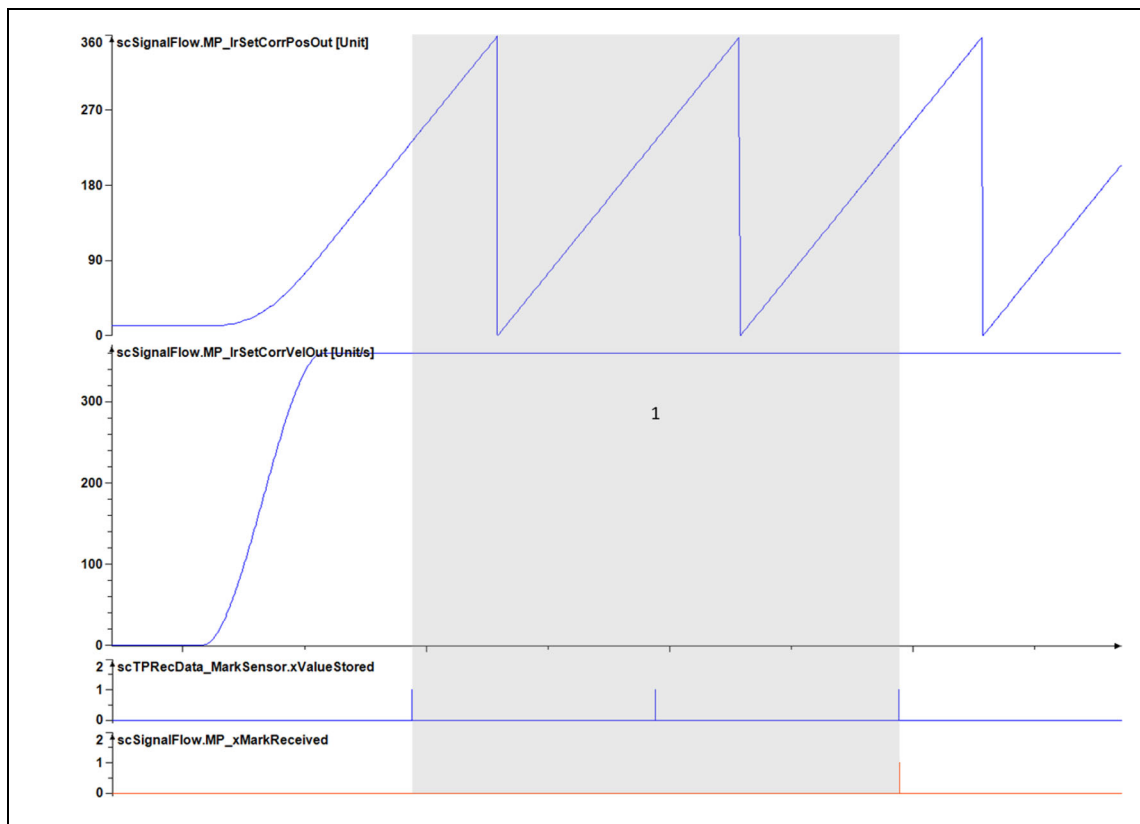
(7): Length of the mark stack

Parameters relevant for the function

The parameters are located in the [L TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- $scPar.IrSensorToolDistMaster$, LREAL, initial value: 0 unit
- $scPar.IrMarkDistMaster$, LREAL, initial value: 360 units

Signal characteristic



[9-27] Signal characteristic when the mark stack is used

No	scTPRecData_MarkSensor.xValueStored	scSignal-Flow.MP_xMarkReceived	Description
1	FALSE → TRUE	FALSE	The first mark has occurred and is saved in the mark stack.
2	FALSE → TRUE	FALSE → TRUE	The first mark has exited the mark stack. The length of the mark stack can be read from the distance covered between scSignalFlow.MP_IrSetCorrPosOut[1] and scSignal-Flow.MP_IrSetCorrPosOut[2].

9.6.5 Gearbox factor correction master value correction

Changing register properties (e.g. within a paper roll) lead to a changed real register length. The difference with respect to the parameterized register length (*scPar.IrSensorToolDistMaster*) leads in turn to corrections in always the same direction (positive or negative). This is inefficient and contributes to increased energy consumption and an increased mechanical load. In this case, the gearbox factor correction calculates the optimum speed setpoint of the master value.

The correction value is determined from the mean value of the detected position errors. This correction value affects the setpoint speed of the master value axis. Through adjustment of the gearbox factor the constant portion of the position errors is eliminated and the corrections are made equally in the positive and negative directions.

The gearbox factor correction is activated when the *xGearFactorCorr* input = TRUE.

The *scPar.IrGearFactorCorrGainMaster* parameter is used to set the gearbox factor correction gain factor, which determines how quickly a constant position error is corrected.

WARNING!

Injury to persons and damage or destruction of machine parts

There is a risk of injury from unpredictable motor movements, especially from a sudden increase in speed.

Possible consequences

Death or severe injuries

Protective measures

Do not set the gain factor in parameter *IrGearFactorCorrGainMaster* too high.

When the gearbox factor correction is activated, the current value of the corrected gearbox factor is output via the *MP_IrGearFactorCorr* signal flow variable.

The *scPar.IrMaxGearFactorCorrMaster* parameter is used to define the maximum gearbox factor correction value in master value axis units.

As soon as the optimum corrected gearbox factor has been detected, correction movements occur both in positive and negative direction. The detected gearbox factor value can be permanently accepted when the *xGearFactorCorrHold* input = TRUE. The value will only be changed when *xGearFactorCorrHold* is set to FALSE.

Inputs relevant for the function

The inputs are located in the [L_TS2P_SyncAndCorrectionState](#) (16) function block.

- *xGearFactorCorr*
- *xGearFactorCorrHold*

Outputs relevant for the function

The outputs are located in the [L_TS2P_SyncAndCorrectionState](#) (16) function block.

- *IrGearFactorCorr*

Parameters to be set

The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

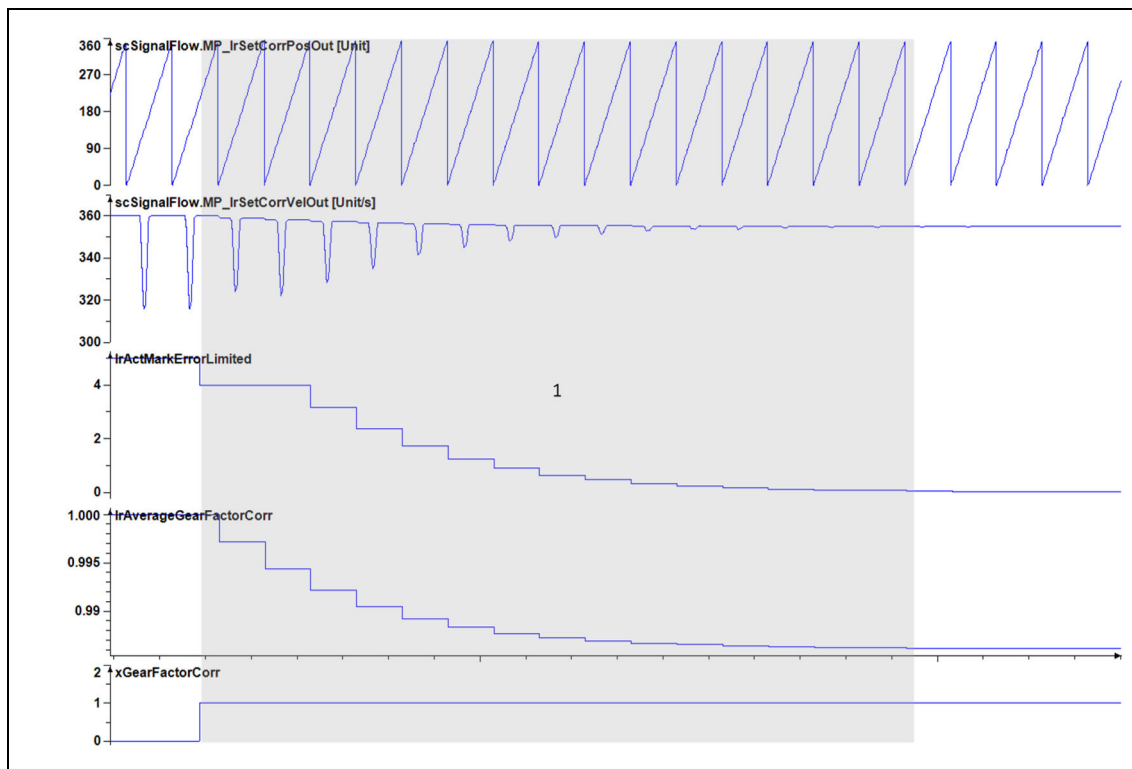
- *IrGearFactorCorrGainMaster*, LREAL, initial value: 0.1 [unit]
- *IrMaxGearFactorCorrMaster*, LREAL, initial value: 10 [unit]

Measuring points relevant for the function

The measuring points are located in the [L_TS2P_scSF_SyncAndCorrectionState](#) (30) structure.

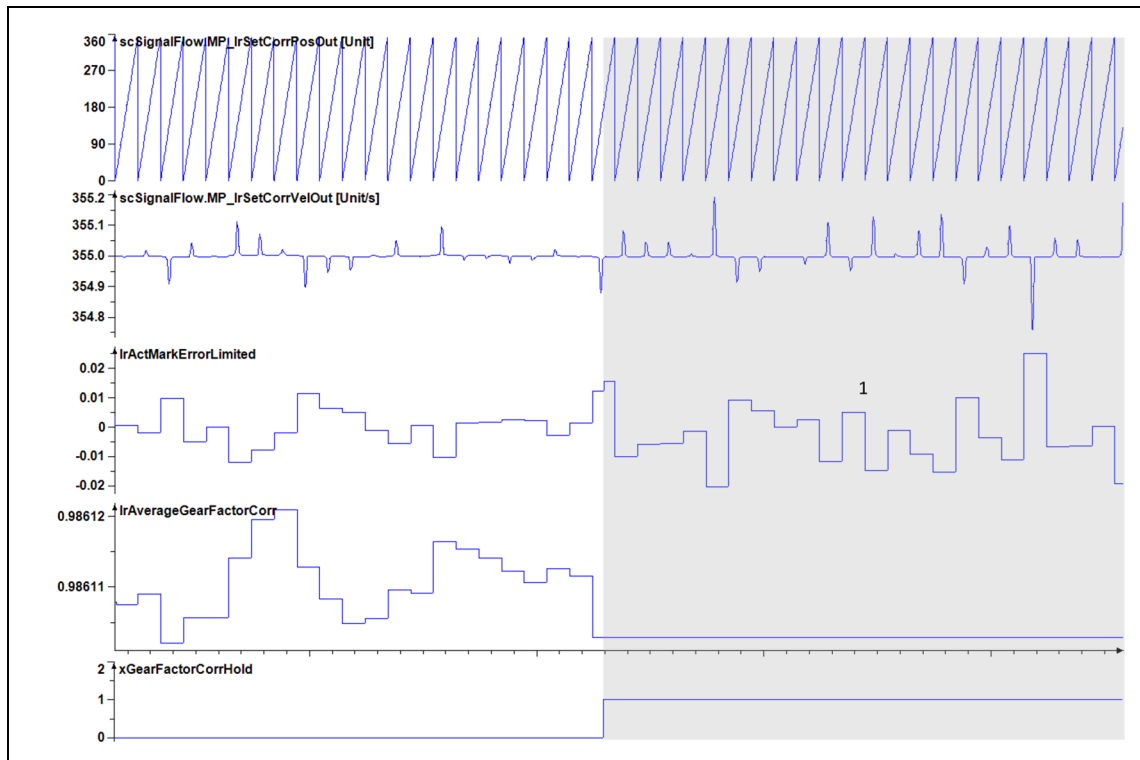
- `MP_xGearFactorCorrLimited`

Signal characteristic



[9-28] "Activate gearbox factor correction" signal characteristic

No	xGearFactorCorr	Description
1	FALSE → TRUE	The gearbox factor is corrected. At the end of the process, the mean mark error = 0 units.



[9-29] "Maintain optimized gearbox factor" signal characteristic

No	xGearFactorCorr	Description
1	FALSE → TRUE	The detected gearbox factor is maintained. There is no further gearbox factor calculation due to newly occurring mark errors.

9.7 Touch probe-based tool correction

This function makes "Homing on the fly" possible for position followers. This may be necessary e.g. if the tool drifts because the exact feed constant of the tool is not known or if, for process reasons, homing is not possible when the machine starts.

When the functionality is activated (*scPar.xCalcToolError* = TRUE), the expected home position is compared to the actual position when the tool sensor (touch probe) is triggered.

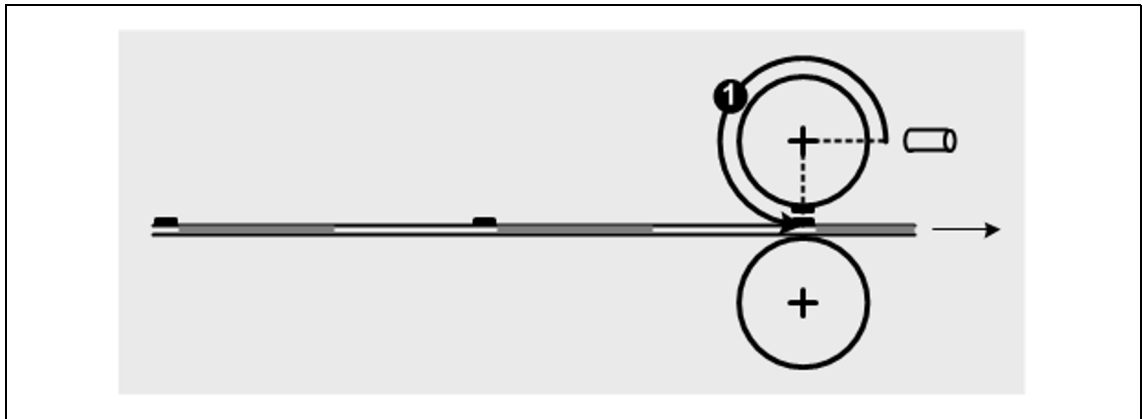
Parameters relevant for the function

The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- *scPar.xCalcToolError*, Bool, initial value: FALSE

9.7.1 Parameterize the touch probe sensor

The *scPar.lrSensorToolDistTool* parameter is used to set the distance between the touch probe sensor and the zero position of the tool (e.g. cutting blade, printheads) on the material.



[9-30] Tool correction sensor parameterization

(1): *scPar.lrSensorToolDistTool*

Parameters relevant for the function

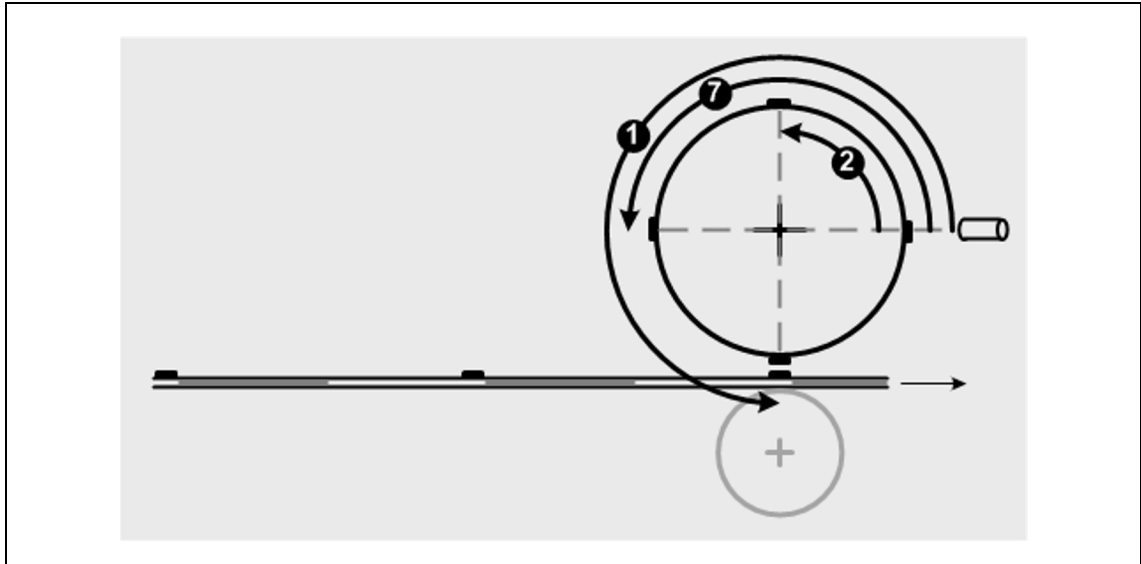
The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- *scPar.lrSensorToolDistTool*, LREAL, initial value: 0 unit

9.7.2 Mark stack touch probe sensor

The mark stack enables the mounting of the mark sensor further than one cycle from the home position (e.g. in the case of pocket-type conveyors or feed constant = $n \times \text{cycle length}$).

The following figure shows the use of a mark stack. Here, the distance between the mark sensor and the tool is greater than the set mark distance.



[9-31] Systematic representation of the mark stack for a tool with feed constant = $4 \times \text{cycle length}$

(1): *scPar.IrSensorToolDistTool*

(2): *0x500A:031 Cycle length*

(7): *Length of the mark stack*

The mark sensor should always be mounted as close as possible to the axis. The farther away the mark sensor is from the axis, the more changes in the material flow remain undetected, causing cutting inaccuracies.

Up to 64 mark signals can be managed in the mark stack. They are available for the system at the right time – delayed by the mark register length. This means that the tool can always be corrected at the right mark signal.

The distance between the touch probe sensor and the position where the tool engages the material is specified via the *scPar.IrSensorDistanceTool* parameter. The mark stack function is automatically activated when the *scPar.IrSensorDistanceTool* parameter $> 0x500A:031 \text{ cycle length}$.

After a mark has been detected, the correction of the value of the mark deviation will only be enabled when the position in the register cycle has covered the distance of the mark stack.

Parameters relevant for the function

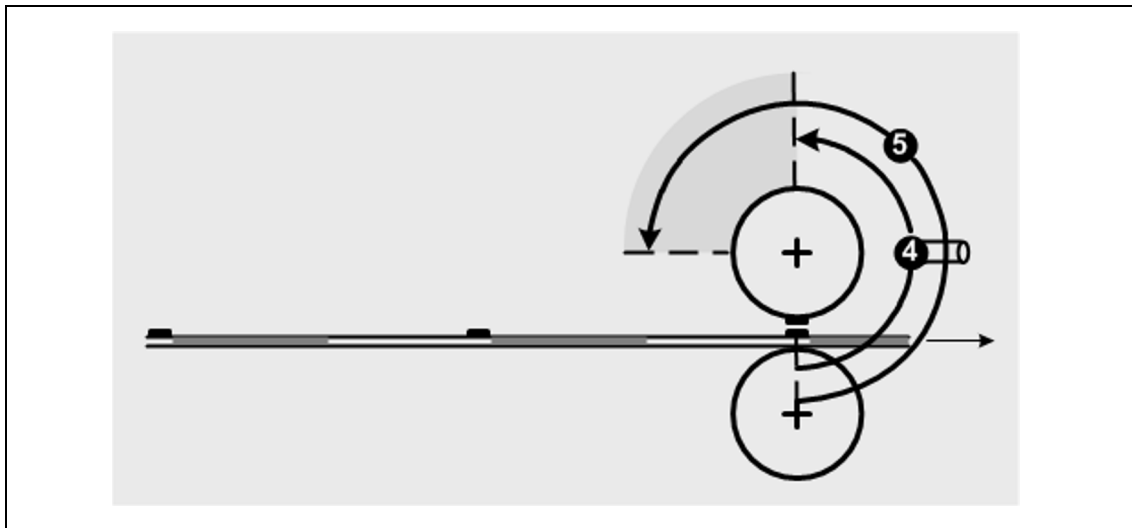
The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- *scPar.IrSensorToolDistTool*, LREAL, initial value: 0 unit

9.7.3 Tool correction window

The tool correction window (window for the tool correction compensating movement) defines the range within which the tool is corrected.

The *scPar.lrLowerCorrPosTool* and *scPar.lrUpperCorrPosTool* parameters are used to define the position of the correction window in the unit of the tool axis.



[9-32] Correction window

(4): *scPar.lrLowerCorrPosTool*

(5): *scPar.lrUpperCorrPosTool*

Parameters relevant for the function

The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- *scPar.lrLowerCorrPosTool*, LREAL, initial value: 90 units
- *scPar.lrUpperCorrPosTool*, LREAL, initial value: 180 units
-

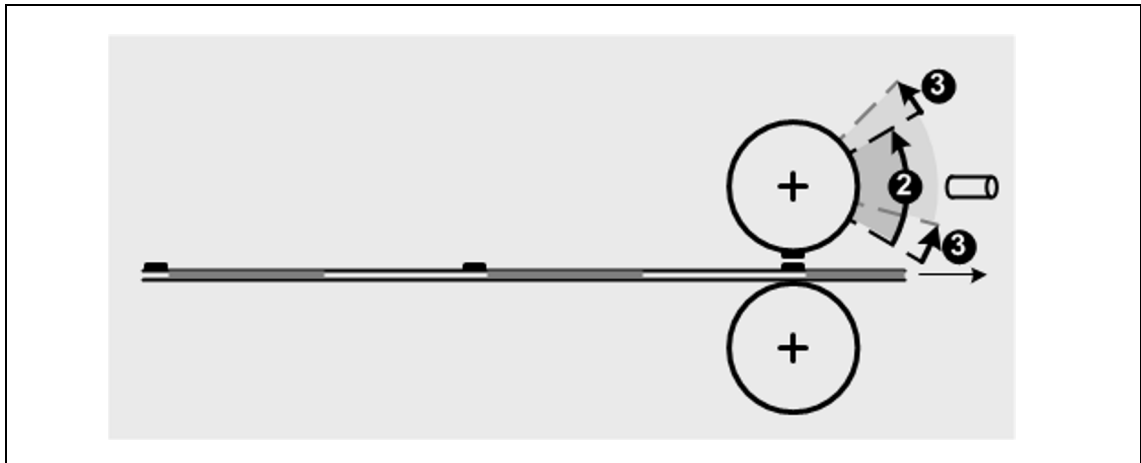
9.7.4 Tool correction mark window

When this function is activated (*scPar.xEnableMarkWindowTool* = TRUE), only touch probes located within the mark window are evaluated; when *scPar.xEnableMarkWindowTool* = FALSE, every touch probe is evaluated.

Shift teaching of the first mark and mark window

The first mark must be taught. To do so, the *xTeachMarkWindowTool* input must be set to TRUE. the *scPar.lrMarkWindowSizeTool* mark window is placed symmetrically around the detected mark when the next touch probe is detected.

The mark window can be shifted via *scPar.lrSetOffsetMarkWindowTool*.



[9-33] Tool correction mark window

(2): *scPar.lrMarkWindowSizeTool*

(3): *scPar.lrSetOffsetMarkWindowTool*

Mark failure detection

If no mark is detected within the mark window (mark failure; *scSF.MP_xMarkNotDetectedTool signal flow variable* = TRUE), this function creates a virtual mark at the optimum position. In this case, no position error is calculated so downstream functions can continue to be executed.

The *scPar.dwMaxNumberVirtualMarksTool* parameter is used to specify the maximum allowed number of mark failures. If the maximum number of undetected marks is reached, an error message is output.

Parameters relevant for the function

The parameters are located in the [L_TS2P_scPar_SyncAndCorrectionState](#) (21) structure.

- *scPar.xEnableMarkWindowTool*, BOOL, initial value: FALSE
- *scPar.lrMarkWindowSizeTool*, LREAL, initial value: 90 units
- *scPar.lrSetOffsetMarkWindowTool*, LREAL, initial value: 0 unit
- *scPar.dwMaxNumberVirtualMarksTool*, DWORD, initial value: 5

Symbols

"Sync & Correction (V2)" functional description [11](#)

"Sync & Correction" TM parameters [21](#)

A

Aborting a positioning process [49](#)

Absolute positioning [41](#)

Access points [32](#)

Application notes [8](#)

C

Competing a positioning process [49](#)

Continuous travel with a defined velocity [46](#)

Conventions used [7](#)

Correction window [76](#)

Coupling, path-based [59](#)

Coupling, profile generator-based [63](#)

D

Document history [6](#)

E

E-mail to Lenze [91](#)

eTMState [27](#)

F

Feedback to Lenze [91](#)

G

Gearbox factor correction [83](#)

H

Halt, [40](#)

Homing [36](#)

I

inputs [17](#)

L

L_TB2P_States [27](#)

L_TS2P_scAP_SyncAndCorrectionState [32](#)

L_TS2P_scPar_SyncAndCorrectionState [21](#)

L_TS2P_scPar_SyncAndCorrectionState parameter structure [21](#)

L_TS2P_scSF_SyncAndCorrectionState [30](#)

L_TS2P_SyncAndCorrectionState [16](#)

L_TS2P_SyncAndCorrectionState function block [16](#)

Layout of the safety instructions [8](#)

Limit setpoints to allowed travel range [70](#)

M

Manual jog (jogging) [34](#)

Mark failure detection [89](#)

Mark stack master value correction [81](#)

Mark stack touch probe sensor [87](#)

Mark window master value correction [78](#)

Master value correction [71](#), [78](#)

Master value correction limitation [76](#)

N

Notes on how to operate the technology module [13](#)

O

outputs [19](#)

Override for speed [51](#)

P

Position coupling [58](#)

Position offset [68](#)

Position synchronism [53](#)

Position synchronism with basic speed [56](#)

Position synchronism with cycle length coupling [53](#)

Q

QSP application [39](#)

R

Relative positioning [43](#)

S

Safety instructions [9](#)

Signal flow diagram [28](#)

Signal flow of the "Sync and Correction" technology module [30](#)

Speed override [51](#)

Start of the axes [14](#)

State machine [26](#)

States [26](#)

States of the eTMState output [27](#)

Structure of the access points

L_TS2P_scAP_SyncAndCorrectionState [32](#)

Structure of the access points

L_TW2P_scAP_SyncAndCorrectionState [32](#)

Structure of the signal flow L_TS2P_SyncAndCorrectionState [30](#)

T

Target group [5](#)

Technology module functions (overview) [12](#)

Tool correction [86](#)

Tool correction mark window [89](#)

Tool correction window [88](#)

Touch probe sensor [73](#), [86](#)

Touch probe-based master value correction [71](#)

Touch probe-based tool correction [86](#)

Trimming [66](#)

V

Variable identifiers [7](#)



Your opinion is important to us

We have compiled these instructions to the best of our knowledge with the objective of supporting you optimally when you handle our product.

Perhaps we have not succeeded in achieving this objective in every respect. If you have suggestions for improvement, please e-mail us to:

feedback-docu@lenze.com

Thank you very much for your support.

Your Lenze documentation team

