

Technology module



Cross Cutter -----

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	13
3.3	Important notes on how to operate the technology module	15
4	Application ranges	16
5	L_TS2P_CrossCutterState function block	17
5.1	inputs	18
5.2	outputs	20
5.3	Parameter	22
6	State machine	27
6.1	State machine of the technology module	27
6.2	State machine of the cutting process	29
7	Signal flow diagram	31
7.1	Main signal flow (incl. torque feedforward control)	31
7.1.1	Master value correction/register control partial signal flow	31
7.1.2	Cross cutter partial signal flow	32
7.1.3	Tool correction partial signal flow	32
7.2	Structure of the signal flow	33
7.3	Structure of the access points	36
8	Description of the basic functions	37
8.1	Manual jog (jogging)	38
8.2	Homing	40
8.3	QSP application	43
8.4	Halt	44
8.5	Absolute positioning	45
8.6	Relative positioning	47
8.7	Continuous travel with a defined velocity	50
8.8	Aborting and completing a positioning process	53
8.9	Override for speed	55
9	Description of the technology functions	57
9.1	Parameterisation of the sealing roller	57
9.2	Positioning to the parking position	58
9.3	"Length cutting" operating mode	60
9.3.1	Top cut	62
9.4	"Flow packer" operating mode	63
9.5	Positioning the cross cutter to static material	66
9.6	Clutching the cross cutter to moving material	68
9.7	Declutching the cross cutter	70
9.8	Positive opening operation / Emergency opening operation	71
9.9	Trimming and offset of the cutting position	73
9.9.1	Position trimming	74
9.9.2	Position offset via profile generator	77

Contents

9.10	Method of elimination	79
9.11	Increasing/decreasing the speed (overspeed)	81
9.12	Calculating the cutting length	82
9.13	Touch probe connection	82
9.14	"Length cutting with mark correction" operating mode	85
9.15	Partial cutting lengths in product length (print/cut format)	87
9.16	Mark window master value correction	88
9.17	Mark stack master value correction	91
9.18	Gearbox factor correction Master value correction	93
9.19	"Cutting on a mark (event-controlled)" operating mode	96
9.20	Torque feedforward control	100
9.21	Tool correction	101
9.21.1	Correction window and tool correction limitation	101
9.21.2	Tool correction mark window	103
9.21.3	Tool correction mark stack	104
	Index	105
	Your opinion is important to us	106

1 About this document

This documentation ...

- contains detailed information on the functionalities of the "Cross Cutter State" 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	04/2020	TD17	Error corrections & supplements
2.0	12/2019	TD06	Error corrections & supplements
1.0	09/2019	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

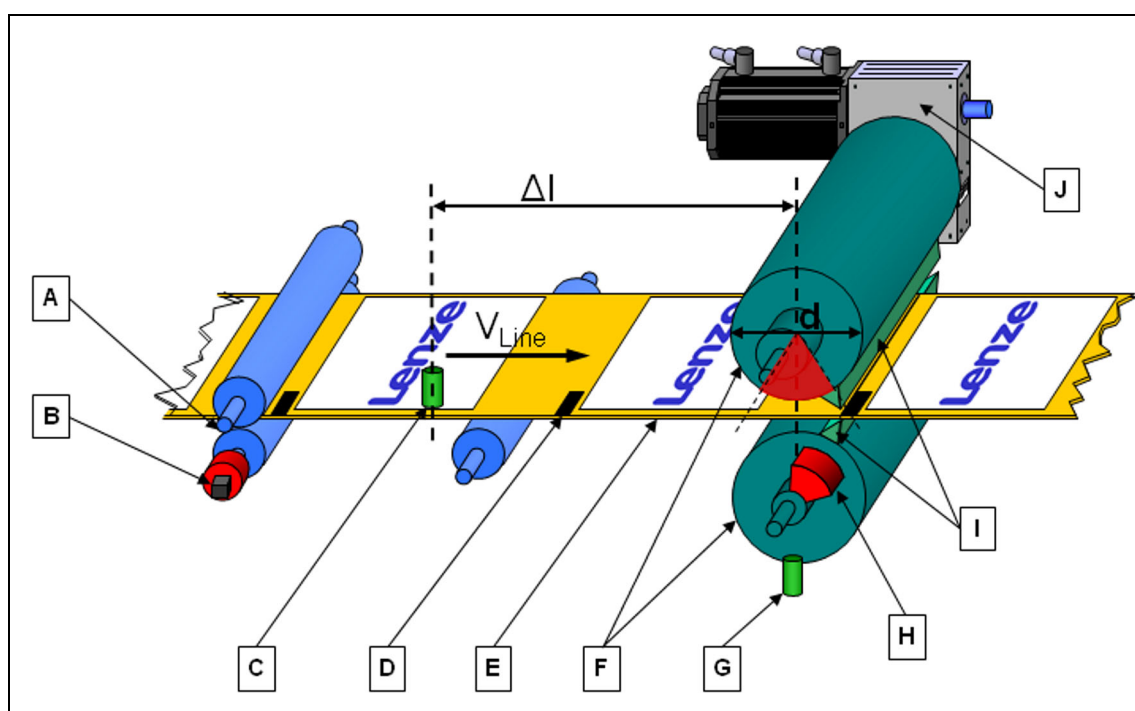
3.1

Functional description

The "Cross Cutter" technology module calculates the cam for the synchronous motion of a cross sealing roller and synchronises it with a master axis.

The drive's movement for a cross cutter is based on a motion profile (curve) saved as a path-path function that is performed depending on a master angle defined by a master axis.

Each position of the master angle is assigned to a setpoint position of the cross-cutter axis via the motion profile. The motion of the cross-cutter axis is thus carried out at a fixed angle to the master axis – even if the master speed varies.



[3-1] Typical mechanics of the technology module

- [A] Feeding roller
- [B] Detection of master speed
- [C] Cutting mark sensor (touch probe sensor) mounted at a distance Δl in front of the sealing rollers
- [D] Cutting mark
- [E] Material web
- [F] Sealing rollers with an adjustable number of sealing jaws
- [G] Reference sensor of the sealing roller
- [H] Reference cam of the sealing roller
- [I] Sealing jaws
- [J] Motor and gearbox

The technology module supports the following process modes:

Length-controlled: Motion profiles are calculated in real time using a 5th degree polynomial for the length control. This allows a flexible response to product gaps or process data changes (cutting length, cutting asynchronicity and sealing area).

Length cutting by detecting marks: Constant lengths are hereby cut and corrected using detected marks.

Event-controlled cut: The cut is executed on detected marks. The length of the cut is variable.

3.2 Functional overview of the TM

Functionality	Description
Basic Motion Base (V2):	
Manual jog (jogging) (□ 38)	Traverse axis in manual operation
Homing (□ 40)	Home the axis
QSP application (□ 43)	Stop motion via an application quick stop deceleration ramp (AxisRef)
Halt (□ 44)	Stop motion via a stop deceleration ramp
Relative positioning (□ 47)	Move axis to a target position relative to the current position
Continuous travel with a defined velocity (□ 50)	Execute continuous travel
Aborting and completing a positioning process (□ 53)	Stop positioning
Override for speed (□ 55)	Specify overrides for speed
Cross Cutter State (V2):	
Parameterisation of the sealing roller (□ 57)	Specify angle of the synchronous area of the sealing roller
Positioning to the parking position (□ 58)	Specify the parking position and the position of the tool.
"Length cutting" operating mode (□ 60)	The profiles are calculated in the "Length cutting" operating mode. The cross cutter cuts the cutting lengths defined according to the <i>IrProcutLength</i> input.
"Flow packer" operating mode (□ 63)	The cycle length of the master axis is decisive for the cut in the "Flow packer" operating mode.
Positioning the cross cutter to static material (□ 66)	The cross cutter is positioned at the material position (X position and resulting Y position from the cam).
Clutching the cross cutter to moving material (□ 68)	The cross cutter is synchronised to the material position (X position and resulting Y position from the cam).
Declutching the cross cutter (□ 70)	Declutching the cross cutter
Positive opening operation / Emergency opening operation (□ 71)	Declutch and stop the slave axis immediately at the current curve position via deceleration.
Trimming and offset of the cutting position (□ 73)	Possibility to shift the cut in relation to the product cycle.
Method of elimination (□ 79)	The elimination function serves to leave out cuts in the material.
Increasing/decreasing the speed (overspeed) (□ 81)	The motion in the cutting phase must be undersynchronous or oversynchronous with the material speed by an adjustable factor.
Calculating the cutting length (□ 82)	Calculating the cutting length
Touch probe connection (□ 82)	Connect touch probe to the technology module
"Length cutting with mark correction" operating mode (□ 85)	In the "Length cutting with mark correction" operating mode the integrated register controller controls the position relative to a mark detected on the material (touch probe) as the higher-level control loop.
Partial cutting lengths in product length (print/cut format) (□ 87)	Define partial cutting lengths in the product length
Mark window master value correction (□ 88)	Activate mark window to prevent the mark sensor from being triggered unintentionally.
Mark stack master value correction (□ 91)	The mark stack is required if, due to the design, the sensor cannot be mounted within one cycle in front of the tool (e.g. cutting blade, print heads). The function ensures that the detected mark is only evaluated when it is located in the cycle in front of the tool.
Gearbox factor correction Master value correction (□ 93)	Adjusting the gearbox factor eliminates the constant portion of the position errors and the corrections are made equally in the positive and negative directions.

Functionality	Description
"Cutting on a mark (event-controlled)" operating mode (📖 96)	The "Cutting on a mark" operating mode is an event-controlled operation. The cross cutter is stopped in the parking position. A cut is executed if a mark is detected on the material.
Torque feedforward control (📖 100)	The real time calculation of the feedforward control value allows flexible changes to the parameters for the torque feedforward control during the cutting process.
Tool correction (📖 101)	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.

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

3.3

Important notes on how to operate the technology module

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 \(jogging\)](#) (438)
 - [QSP application](#) (443)
- 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](#) (440)



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.
 - $xHomeExecute = FALSE \rightarrow TRUE$
==> **HOMING** state

Setting of the operating mode

The operating mode for the cross cutter axis has to be set to "cyclically synchronous position" (csp) because the axis is led via the master position value.

4 Application ranges

The "Cross Cutter State" technology module is based on the "electronic cam" functionality.

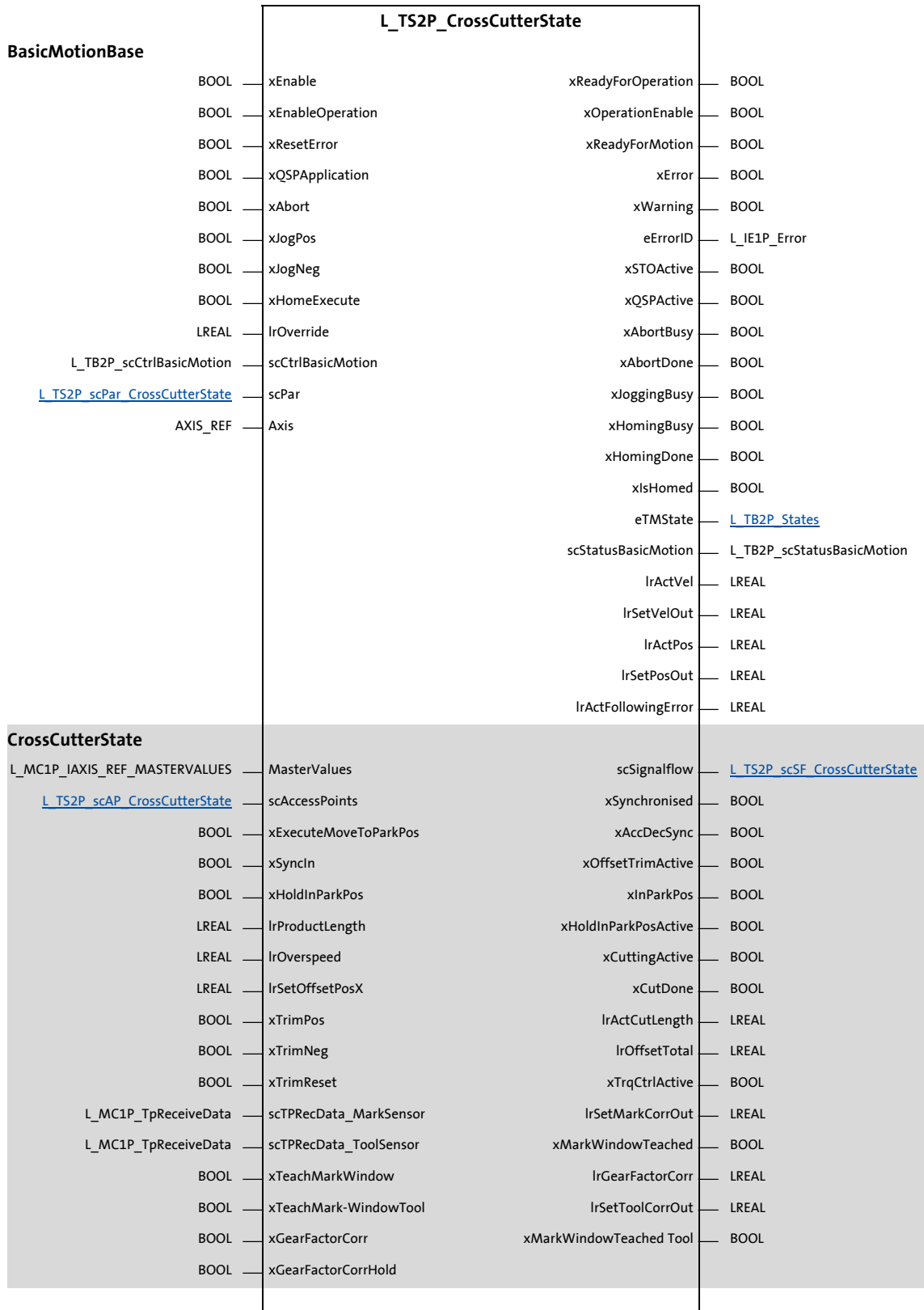
It can be used to realise applications for synchronous cross cutters and rotary cutters, as well as applications for synchronised transfer stations, e.g. in the field of packaging machines.

Typical applications:

- Cross cutters
- Punches
- Sealing rollers

5 L_TS2P_CrossCutterState 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	The input is edge-controlled and evaluates the rising edge.
		TRUE The axis is switched to ready for operation. Controller enable is issued.
xResetError	BOOL	The input is edge-controlled and evaluates the rising edge.
		FALSE Reset axis error or software error. TRUE
xQSPApplication	BOOL	The input is level-controlled and evaluates a high level.
		TRUE Active motion is aborted and axis is brought to a standstill via the QSP application deceleration ramp (43). • 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>)
xJogPos	BOOL	The input is level-controlled and evaluates a 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 a 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_CrossCutterState		The parameter structure contains the parameters of the technology module (22).
Axis AXIS_REF		Reference to the axis
MasterValues L_MC1P_IAXIS_REF_MASTERVERVALUES		Connection for the master values (master axis) The following types of axes can be connected to this input: • <i>AXIS_REF</i> • <i>AXIS_REF_i900</i> • <i>AXIS_REF_ENC_i900_Drive</i> • <i>AXIS_REF_VIRTUAL</i> • <i>AXIS_REF_VIRTUALM_i900</i>

Designator	Data type	Description
scAccessPoints L_TS2P_scAP_CrossCutterState		Structure of the access points (36) The access points can be used to influence the signal flow or technology module functions.
xSyncIn	BOOL	The input is level-controlled and evaluates a high level.
		TRUE Clutching in to the cam FALSE Declutching in the parking position
xHoldInParkPos	BOOL	Rejection process (leave out cuts)
		TRUE After the synchronous phase, the slave axis is declutched in the parking position. - The cutting process is not continued. - The product position (position of the X axis) at the <i>IrSetXPosOut</i> output, however, is still calculated. FALSE After the next synchronous phase, the slave axis is clutched in to the subsequent synchronous phase. The cutting process is continued.
IrProductLength	LREAL	Length of the product or material in the cutting process depending on the operating mode (<i>scPar.eCuttingMode</i> parameter) ▶ L_TS2P_scPar_CrossCutterState (22) - Unit: Unit of the master axis (e.g. mm) - Initial value: 0.0
IrOverspeed	LREAL	Factor for speeding during the synchronous phase. The scaled value refers to the current material speed (master speed). Example: <i>IrOverspeed</i> = 1.0 means that the tool (e.g. cutting blade) has twice the circumferential speed than the material speed. - Value range: -0.9 ... 2.0: - Initial value: 0
IrSetOffsetPosX	LREAL	Position offset of the master axis The resulting X position of the cam is produced by addition of the master axis position to this input. - Unit: unit: - Initial value: 0
xTrimPos	BOOL	Trimming in positive direction If <i>xTrimPos</i> AND <i>xTrimNeg</i> = TRUE, the first selected travel direction is maintained. The evaluation of this input depends on the set trimming mode (<i>scPar.eTrimMode</i> parameter). Initial value: FALSE
		TRUE eTrimMode = 0 (trimming per velocity offset): The input is level-controlled and evaluates a high level.
		FALSE → eTrimMode = 1 (trimming per position offset): The input is edge-controlled and evaluates the rising edge. TRUE
xTrimNeg	BOOL	Trimming in negative direction If <i>xTrimPos</i> AND <i>xTrimNeg</i> = TRUE, the first selected travel direction is maintained. The evaluation of this input depends on the set trimming mode (<i>scPar.eTrimMode</i> parameter). Initial value: FALSE
		TRUE eTrimMode = 0 (trimming per velocity offset): The input is level-controlled and evaluates a high level.
		FALSE → 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 → The position offset resulting from the trimming is reset and moved to zero via the profile generator. TRUE
scTPRecData_MarkSensor 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

Designator	Data type	Description
xTeachMarkWindow	BOOL	TRUE Teach mark window of the master value correction
xTeachMark-WindowTool	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) (□ 38) function is active. The technology module is in the "JOGGING" state.
xHomingBusy	BOOL	TRUE The Homing (□ 40) 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

Designator	Data type	Description
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: unit/s
IrSetVelOut	LREAL	Setpoint speed • Unit: unit/s
IrActPos	LREAL	Current position • Unit: unit
IrSetPosOut	LREAL	Setpoint position • Unit: unit
IrActFollowingError	LREAL	Current following error • Unit: unit
scSignalflow L_TS2P_scSF_CrossCutterState		Structure of the signal flow (33)
xInParkPos	BOOL	TRUE The axis is in the parking position. This output can be set by various functions: • Positioning into the parking position • Holding the axis in the parking position during the cutting process • The cutting length in the cutting process is so long that the axis comes to a stop in the parking position • Decutching to the parking position with the <i>xSyncln</i> input = FALSE
xSynchronised	BOOL	TRUE The axis is coupled with respect to the master axis. The technology module is in the "CUTTING" 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.
xHoldInParkPosActive	BOOL	TRUE During the rejection process (<i>xHoldInParkPos</i> input = TRUE), the slave axis is in the parking position.
xCuttingActive	BOOL	TRUE The tool (e.g. cutting blade) is in the cutting area, i.e. in the synchronous phase.
xCutDone	BOOL	FALSE TRUE The cut has been effected. With a rising edge at the <i>xCutDone</i> output, the cut length of material is calculated and output at the <i>IrActCutLength</i> output. The output is active for one PLC clock cycle.
IrActCutLength	LREAL	Cut length of the material. The cut length results from the difference between the current position of the master axis and the current cutting position of the sealing roller. • Unit: mm
IrOffsetTotal	LREAL	Total position offset between the master axis and the X axis from the cam.
xOffsetTrimActive	BOOL	TRUE An offset function (position offset, velocity offset) and/or the trimming function is active.
xTrqCtrlActive	BOOL	TRUE Torque feedforward control is active.
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

Designator	Data type	Description
IrSetToolCorrOut	LREAL	Current position error of the tool correction limited to the maximum value.
xMarkWindowTeachedTool	TRUE BOOL	Mark window of the tool correction taught.

5.3

Parameter

L_TS2P_scPar_CrossCutterState

Designator	Data type	Description
IrVel	LREAL	Speed for movements that are commissioned via the scCtrlBasicMotion structure. - Unit: unit/s - Initial value: 3600
IrAcc	LREAL	Acceleration for motions activated via the scCtrlBasicMotion structure. Selection of the value for the maximum acceleration speed variation. - Unit: unit/s² - Initial value: 36000
IrDec	LREAL	Deceleration for motions activated via the scCtrlBasicMotion structure. Selection of the value for the maximum deceleration speed variation until standstill. - Unit: unit/s² - Initial value: 36000
IrJerk	LREAL	Jerk for motions activated via the scCtrlBasicMotion structure. - Unit: unit/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.
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: unit/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: unit/s² - 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: unit/s² - Initial value: 100

Designator	Data type	Description
IrSetOffsetTrimJerk	LREAL	Jerk for the offset and trimming functions - Unit: unit/s³ - Initial value: 10 000
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_TS2P_SyncModeCrossCutter	Synchronisation of the cross cutter Initial value: 0 (ProfGenAbsolute)
	0	Cam_in: Clutch cross cutter to moving material
	1	Positioning: Position the cross cutter to the stationary material.
IrMasterSyncInDist	LREAL	Clutch-in distance for the master axis in the "length cutting" operating mode (<i>eCuttingMode</i> parameter = 0) for the "Cam_in" synchronisation mode (<i>scPar.eSyncMode</i> = 0) For the <i>IrMasterSyncInDist</i> setting = 0, an optimum clutch-in distance is calculated. - Unit: unit - Initial value: 0
IrMasterSyncOutDist	LREAL	Declutch distance for the master axis in the "length cutting" operating mode (<i>eCuttingMode</i> parameter = 0) for the "Cam_in" synchronisation mode (<i>scPar.eSyncMode</i> = 0) For the setting <i>IrMasterSyncOutDist</i> = 0, an optimum declutch distance is calculated. - Unit: unit - Initial value: 0
xMasterAbsolute	BOOL	Reference to the position of the master axis Initial value: TRUE
	TRUE	Absolute reference between the position of the master axis and the x axis of the cam.
	FALSE	Relative reference: The x starting position is loaded from the <i>IrStarttXPosRelative</i> parameter.
IrStartXPosRelative	LREAL	Starting position for the X axis of the cam when setting the <i>xMasterAbsolute</i> = FALSE - Unit: unit - Initial value: 0
byStartToolNo	BYTE	Number of the tool (e.g. cutting blade) on the sealing roller which executes the first cut. Initial value: 0 (automatic tool selection)
byNumOfTools	BYTE	Number of the tools on the sealing roller The tools are spread symmetrically on the circumference of the roller. Initial value: 1
IrCuttingAngle	LREAL	Angle of the synchronous area of the sealing roller - Unit: Degree - Value range: from 1° to 359° / number of tools (<i>byNumOfTools</i>) - Initial value: 10
IrToolCircumference	LREAL	Circumference of the sealing roller - Unit: Unit of the master axis (e.g. mm) - Initial value: 75.55555

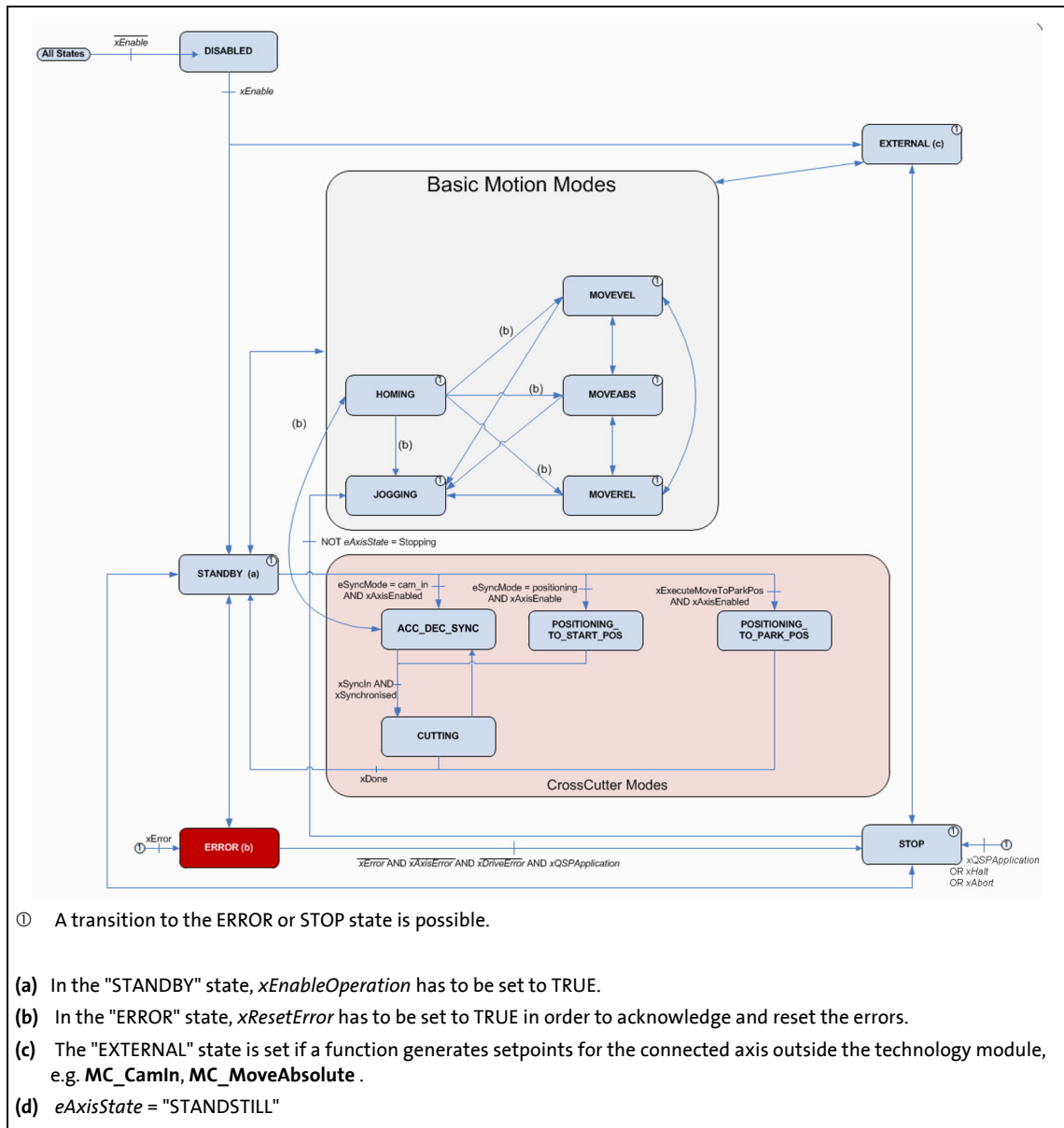
Designator	Data type	Description
xCalcCamBounds	BOOL	TRUE The extreme values for the slave axis (<i>scSignalFlow.MP_IrSlaveMaxVel</i> , <i>scSignalFlow.MP_IrSlaveMaxAcc</i>) are calculated depending on the <i>IrMasterMaxVel</i> and <i>IrMasterMaxAcc</i> parameters. • Initial value: FALSE
IrMasterMaxVel	LREAL	Maximum speed of the master axis for checking the cams • Unit: unit/s • Initial value: 100
IrMasterMaxAcc	LREAL	Maximum acceleration of the master axis for checking the cams • Unit: unit/s ² • Initial value: 1000
eCuttingMode L_TS2P_CuttingModeState		Selection of the operating mode in the cutting process • Initial value: 0 (length cutting)
		0 Length cutting The product length corresponds to the cutting length.
		1 Flow packer The product length is only used to calculate the speed in the synchronous phase. The cut is always executed at the end of the cycle of the master axis.
		2 Length cutting with mark correction The product length corresponds to the register length and is accepted once before the cutting process. The cutting length is set once within the product length by means of the <i>scPar.alrCutLength[1..10]</i> array.
		3 Cutting on a mark (event-controlled) The product length is not evaluated in this mode.
alrCutLengthInProduct ARRAY[1..10] OF LREAL		Cutting lengths within the product length (<i>IrProductLength</i> input) Use only in the "length cutting with mark correction" cutting process operating mode (<i>scPar.eCuttingMode</i> = 2) • Initial value: 0 (cutting lengths deactivated)
IrMinProductLength	LREAL	Minimum cutting length = minimum distance between two (touch probe) marks Use only in the "Cutting on a mark" cutting process operating mode (<i>scPar.eCuttingMode</i> = 3) • Unit: unit • Initial value: 50
IrTrqCtrlGain	LREAL	Gain factor for the torque feedforward control • Initial value: 0 (torque feedforward control deactivated)
IrTrqCtrlRateTime	LREAL	Rate time for the torque feedforward control • Unit: s • Initial value: 0
IrMIInertia	LREAL	Moment of inertia for the torque feedforward control relative to the sealing rollers at the gearbox output of the motor. • Unit: kg/cm ² • Initial value: 0 (torque feedforward control deactivated)
xCorrMarkErrorMaster	BOOL	TRUE Activate correction of the touch probe deviation. • Initial value: TRUE
IrSensorToolDistMaster	LREAL	Distance between the touch probe sensor and the position where the tool (e.g. cutting blade) engages the material. This parameter is needed for the automatic calculation of the mark register. If the touch probe sensor and axis are within one register cycle, this value can be set to "0". • Unit: unit • Initial value: 0
IrMaxCorrPosMaster	LREAL	Maximum positive correction distance per register cycle (x units) • Unit: unit • Initial value: 30
IrMaxCorrNegMaster	LREAL	Maximum negative correction distance per register cycle (x units) • Unit: unit • Initial value: -30

Designator	Data type	Description
xEnableMarkWindowMaster	BOOL	Activation of the mark window for the master value Initial value: FALSE
		FALSE Mark window deactivated
		TRUE Mark window activated
lrMarkWindowSizeMaster	LREAL	Size of the touch probe window relative to the x axis of the cam. The window is placed symmetrically around the expected touch probe position. - Unit: unit - Initial value: 20
lrSetOffsetMarkWindow-Master	LREAL	Offset for shifting the touch probe window relative to the register measuring system. - Unit: unit - Initial value: 0
lrGearFactorCorrGainMaster	LREAL	Gain factor of the gearbox factor correction - Unit: - - Initial value: 0.1
lrMaxGearFactorCorrMaster	LREAL	Maximum deviation of the gearbox factor correction - Unit: unit - Initial value: 10
dwMaxNumberVirtual-MarksMaster	DWORD	Maximum number of permitted touch probe failures If no touch probes are detected within the touch probe window, an artificial mark is generated. This happens as long as the number of marks set here is not exceeded. If the number of marks is exceeded, the <i>xWarning</i> output is set to TRUE. - Unit: - - Initial value: 5
xLoadRegPos	BOOL	TRUE The register position at the lrLoadRegPos input is loaded manually. This input is only evaluated if the register control is inactive (<i>xSyncPos</i> input = FALSE). Initial value: FALSE
lrLoadRegPos	LREAL	Position to be loaded for the register (when <i>xLoadRegPos</i> = TRUE). - Unit: unit - Initial value:
xCalcToolError	BOOL	Tool correction mode Initial value: FALSE
		FALSE Cross cutter without tool correction
		TRUE Cross cutter with tool correction
xCorrToolError	BOOL	Enabling to correct the calculated tool error Initial value: FALSE
		FALSE Correction movement deactivated
		TRUE Correction movement activated
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
xEnableMarkWindowTool	BOOL	Activation of the mark window for the tool correction Initial value: FALSE
		FALSE Mark window deactivated
		TRUE Mark window activated
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

Designator	Data type	Description
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 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
dwMaxNumberVirtualMarks Tool	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

6 State machine

6.1 State machine of the technology module



[6-1] State machine of the technology module

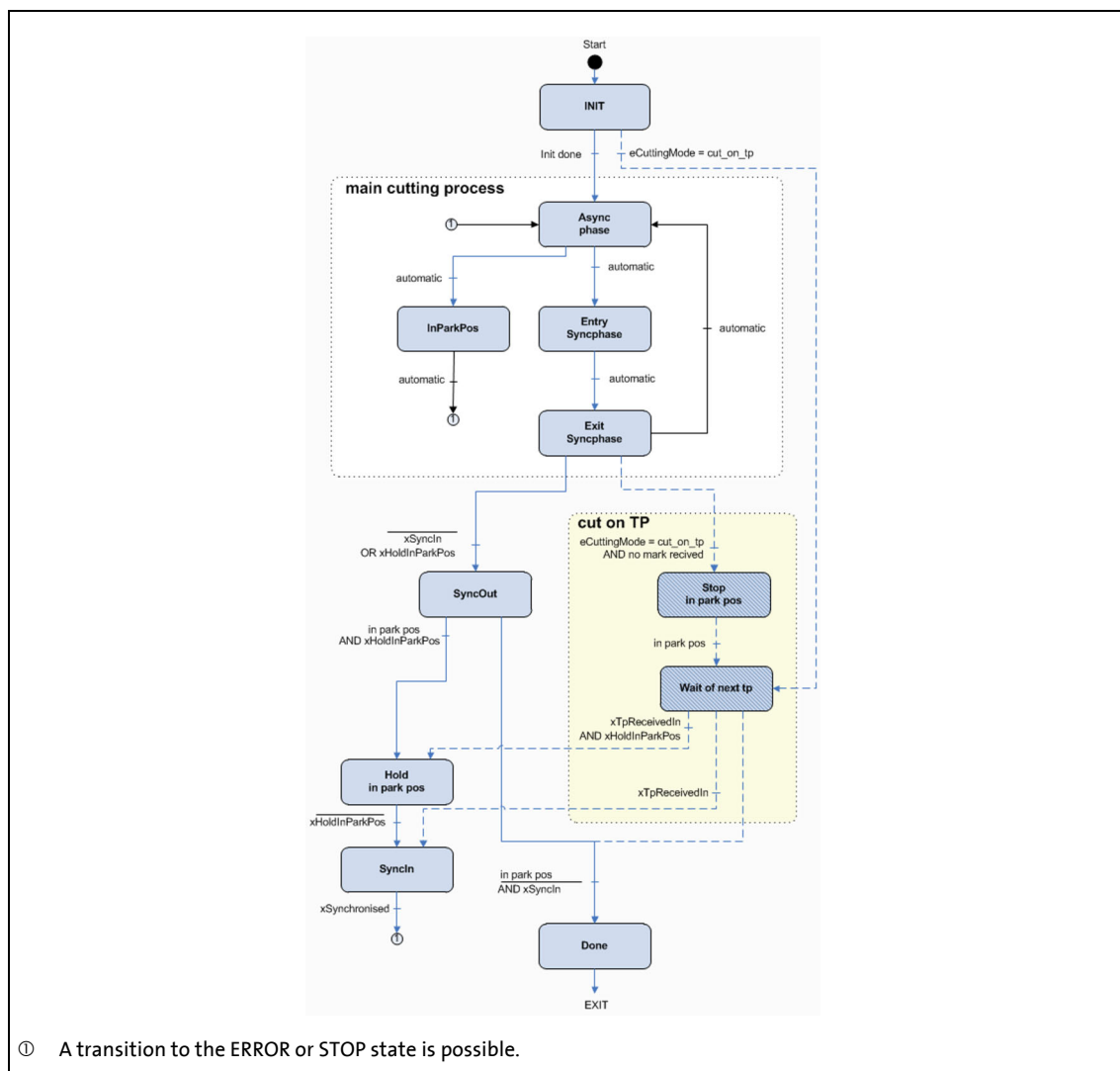
States of the eTMState output (L_TB2P_States)

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

No.	L_TB2P_States	Description
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
32	POSITIONING_TO_START_POS	Movement to start position active
35	POSITIONING_TO_PARK_POS	Movement to parking position active
170	CUTTING	Cutting function active
1000	SYSTEMFAULT	System fault

6.2

State machine of the cutting process



[6-2] State machine of the cutting process

States of the eTMState output (L_TB2P_States)

No.	L_TB2P_States	Description
-1	DISABLED	Cutting mode deactivated.
172	ASYNC PHASE	The cross cutter roller passes through the asynchronous area.
171	IN_PARK_POS	The axis is in the parking position.
173	ENTRY_IN_SYNC_PHASE	The cross-cutter axis is in the synchronous phase.
175	EXIT_SYNC_PHASE	Stop the synchronous phase after the cut.
176	HOLD IN PARKPOS	The axis is declutched in the parking position. The cam continues to be run down in the background.
177	START_CAM	The starting curve is active. The cross cutter axis is accelerated up to the setpoint speed in the synchronous area.
178	STOP_CAM	The stop curve is activated. The cross-cutter axis is held in the parking position.

No.	L_TB2P_States	Description
180	STOP IN PARKPOS	Declutching in the parking position.
179	WAIT OF NEXT TP	The TM is waiting for a valid mark for the next cut.

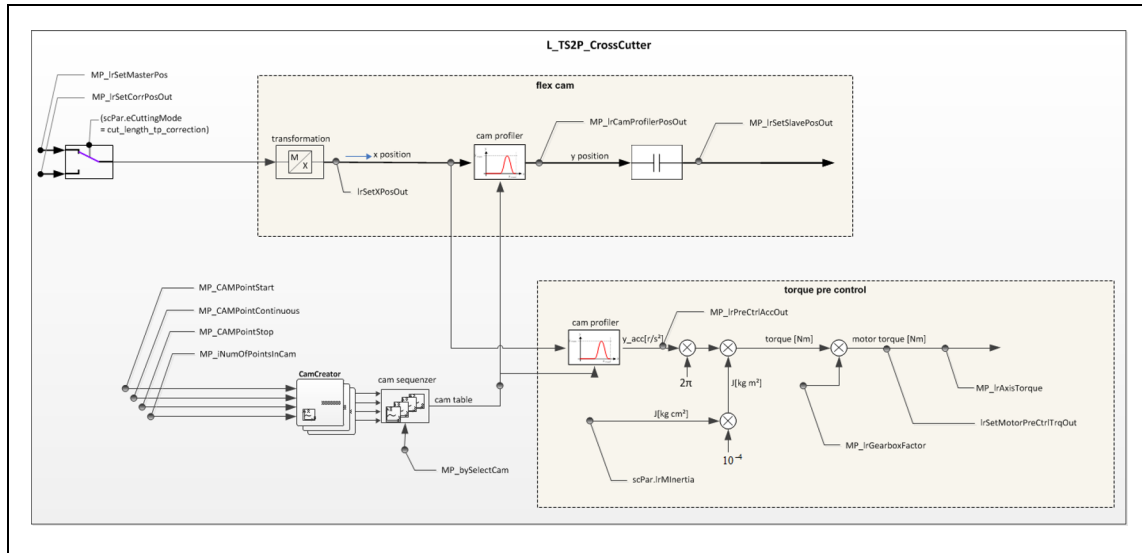
7 Signal flow diagram

7.1 Main signal flow (incl. torque feedforward control)

7 Signal flow diagram

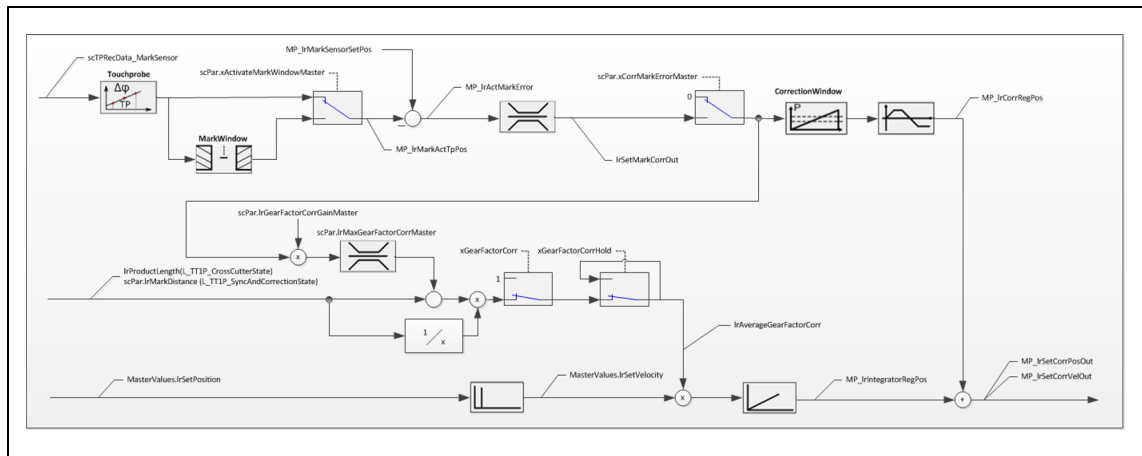
7.1 Main signal flow (incl. torque feedforward control)

The following figure shows the main signal flow of the function:



[7-1] Signal flow of the technology module

7.1.1 Master value correction/register control partial signal flow

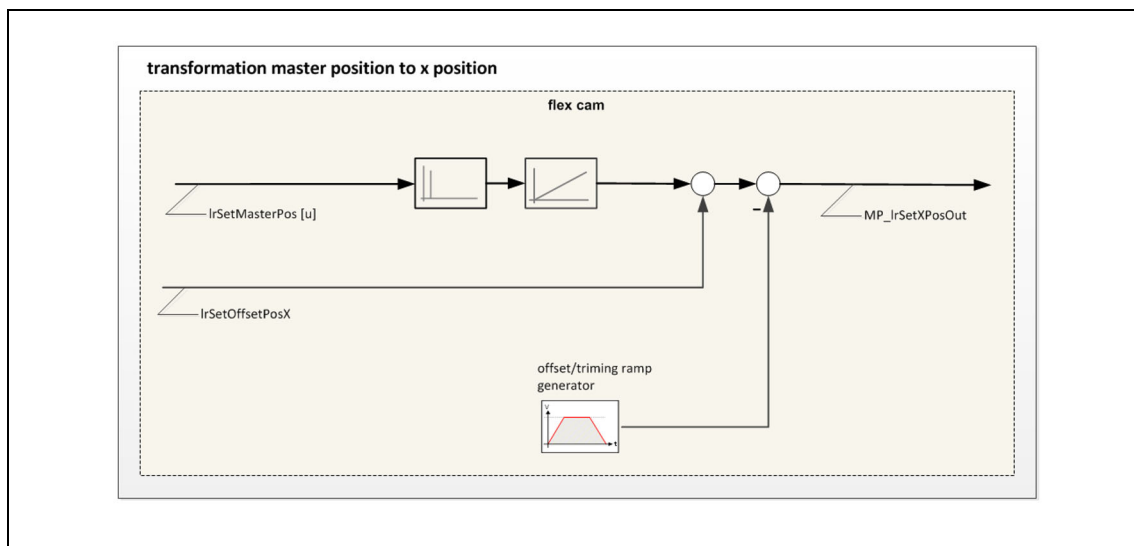


[7-2] Master value correction/register control signal flow

7 Signal flow diagram

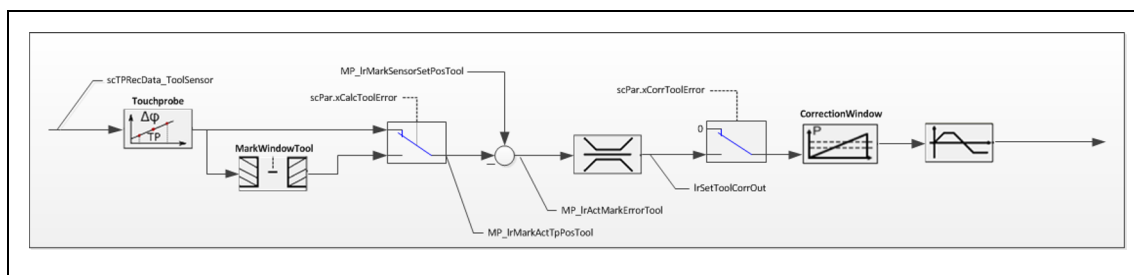
7.1 Main signal flow (incl. torque feedforward control)

7.1.2 Cross cutter partial signal flow



[7-3] Cross cutter signal flow

7.1.3 Tool correction partial signal flow



[7-4] Tool correction signal flow

7.2 Structure of the signal flow

L_TS2P_scSF_CrossCutterState

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

► [Signal flow diagram](#) (31)

Designator	Data type	Description
MP_lrSetMasterPos	LREAL	Setpoint position of the master value axis
MP_lrCamProfilerPosOut	LREAL	The Y position directly from the cam generator • Unit: unit
MP_lrSetSlavePosOut	LREAL	Set position of the slave axis • Unit: unit
MP_CAMPointStart ARRAY[0...3] OF SMC_CAMXYVA		Interpolation points of the start cam
MP_CAMPointContinuous ARRAY[0...3] OF SMC_CAMXYVA		Interpolation points of the cutting cam
MP_CAMPointStop ARRAY[0...3] OF SMC_CAMXYVA		Interpolation points of the stop cam
MP_iNumOfPointsInCam	INT	Number of points in the current cam
MP_bySelectCam	BYTE	Cam used
		1 Cutting curve
		2 Cutting curve
		3 Starting curve
		4 Stop curve
MP_eCuttingProcessState L_TB2P_States		State of the cutting process in the technology module (output <i>eTMState</i> = CUTTING)
MP_lrYStart	LREAL	Internally calculated starting position of the Y axis for the cam
MP_lrYPark	LREAL	Internally calculated parking position of the Y axis for the cam
MP_lrYEnd	LREAL	Internally calculated target position of the Y axis for the cam
MP_lrXStart	LREAL	Internally calculated starting position of the X axis for the cam
MP_lrXEndSyncPos	LREAL	Internally calculated position of the X axis for the cam in the transition to the asynchronous area
MP_lrXStartParkPos	LREAL	Internally calculated starting position of the X axis for the parking position of the Y axis for the cam
MP_lrXEndParkPos	LREAL	Internally calculated target position of the X axis for the parking position of the Y axis for the cam
MP_lrXStartSyncPos	LREAL	Internally calculated position of the X axis for the cam in the transition to the synchronous area
MP_lrXEnd	LREAL	Internally calculated target position of the X axis for the cam
MP_xXAbsolute	BOOL	TRUE The absolute reference between X axis and master axis is activated or already active.
MP_xSlaveAbsolute	BOOL	TRUE The absolute reference between X axis and slave axis is activated or already active.

Designator	Data type	Description	
MP_xEndOfProfile	BOOL	TRUE	The last cycle in the current cam profile is active. • The current values are extrapolated for detection. • The signal is applied for one clock cycle.
MP_xEndOfXCycle	BOOL	TRUE	The last position on the X axis is reached. • The current values are extrapolated for detection. • The signal is applied for one clock cycle.
MP_xEnablePositionFollower	BOOL	TRUE	Enable position follower for the setpoint positions of the slave axis from "Flex Cam".
MP_lrOverspeed	LREAL	Effective factor for speeding during the synchronous phase. The scaled value is relative to the current material speed (<i>Master.lrSetVelocity</i>).	
MP_lrOffset	LREAL	Offset between the master and x axis resulting from the function via the <i>lrSetOffsetPosX</i> input.	
MP_lrOffsetTrim	LREAL	Trimming offset between the master and x axis resulting from the function via the inputs <i>xTrimPos</i> and <i>xTrimNeg</i> .	
MP_xEnableTorqueFollower	BOOL	TRUE	Enable torque follower for the torque feedforward control of the slave axis.
MP_lrPreCtrlAccOut	LREAL	Internally calculated acceleration from the cam for the next setpoint	
MP_lrGearboxFactor	LREAL	Transmission factor of torque from cross cutter axis to motor axis	
MP_lrAxisTorque	LREAL	Setpoint torque of the slave axis • Unit: Nm	
MP_lrIntegratorRegPos	LREAL	Integrated master speed	
MP_lrCorrRegPos	LREAL	Position curve for the correction movement	
MP_lrMarkSensorSetPos	LREAL	Position of the mark sensor within the mark cycle length	
MP_lrMarkActTpPos	LREAL	Measured mark position within the mark cycle length	
MP_xMarkReceived	BOOL	TRUE	Mark occurred
MP_lrActMarkError	LREAL	Current unlimited mark error	
MP_lrActMarkDist	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 master value correction 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 beyond the mark window
MP_xMarkNotDetected	BOOL	TRUE	No mark has been detected in the mark window
MP_xGearFactorCorrLimited	BOOL	TRUE	The gearbox factor correction is limited to the maximum value <i>scPar.lrMaxGearFactorCorrMaster</i>
MP_lrSetCorrPosOut	LREAL	Mark-corrected master position at the master value correction output	
MP_lrSetCorrVelOut	LREAL	Mark-corrected master speed at the master value correction output	

Designator	Data type	Description
MP_lIrSetMasterVel	LREAL	Setpoint position of the master value axis
MP_lIrMarkSensorSetPosTool	LREAL	Position of the tool sensor within the slave cycle length
MP_lIrMarkActTpPosTool	LREAL	Measured tool position within the slave cycle length
MP_lIrActMarkErrorTool	LREAL	Unlimited tool error
MP_xMarkReceivedTool	BOOL	TRUE Tool correction mark detected
MP_xCorrActiveTool	BOOL	TRUE The setpoint of the slave axis is within the tool correction window
MP_xCorrLimitedTool	BOOL	TRUE The tool correction movement is limited
MP_xMarkWindowOpenTool	BOOL	TRUE The mark window of the tool correction is open
MP_xMarkOutOfWindowTool	BOOL	TRUE Mark outside the mark window of the tool correction
MP_xMarkNotDetectedTool	BOOL	TRUE No mark has occurred in the mark window of the tool correction
MP_xMarkWindowTeachedTool	BOOL	TRUE Mark window of the tool correction taught
MP_bySetToolNoOut	BYTE	The number of the tool (e.g. cutting blade) used for cutting when the synchronisation function is active (<i>xAccDecSync</i> output = TRUE). - The first tool is in the zero position of the axis. - Additional tools on the circumference of the axis are numbered consecutively in a positive direction of rotation.
MP_bySetAreaOut	BYTE	The number of the area between two tools on the circumference of the axis. The first area is between the first and second tool.
MP_lIrSetXPosOut	LREAL	Position of the X axis from the cam The X axis always runs from 0 to the cutting length. Unit: unit
MP_lIrSetXVelOut	LREAL	Velocity of the X axis from the cam Unit: unit/s
MP_lIrSetXAccOut	LREAL	Acceleration of the X axis from the cam Unit: unit/s²
MP_lIrSlaveMaxVel	LREAL	Maximum speed of the slave axis. This value will be reached if the master axis is executed at the maximum speed in the <i>scPar.lIrMasterVelMax</i> parameter. The calculation begins when the <i>xCamBounds</i> input = TRUE. Unit: unit/s²
MP_lIrSlaveMaxAcc	LREAL	Maximum acceleration of the slave axis. This value will be reached if the master axis is executed at the maximum speed in the <i>scPar.lIrMasterVelMax</i> parameter and the maximum acceleration in the <i>scPar.lIrMasterAccMax</i> parameter. The calculation begins when the <i>xCamBounds</i> input = TRUE. Unit: unit/s³
MP_lIrSetMotorPreCtrlTrqOut	LREAL	Setting of the torque feedforward control value The torque refers to the output end of the gearbox (gearbox output). Unit: Nm
MP_xRegisterCtrlActive	BOOL	TRUE The register control is activated.

7.3 Structure of the access points

L_TS2P_scAP_CrossCutterState



Tip!

The structure of the access points is not available on this technology module.

8 Description of the basic functions

Overview of the basic functions

- [Manual jog \(jogging\)](#) (📖 38)
- [Homing](#) (📖 40)
- [QSP application](#) (📖 43)
- [Halt](#) (📖 44)
- [Absolute positioning](#) (📖 45)
- [Relative positioning](#) (📖 47)
- [Continuous travel with a defined velocity](#) (📖 50)
- [Aborting and completing a positioning process](#) (📖 53)
- [Override for speed](#) (📖 55)

8 Description of the basic functions

8.1 Manual jog (jogging)

8.1 Manual jog (jogging)

Prerequisite

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

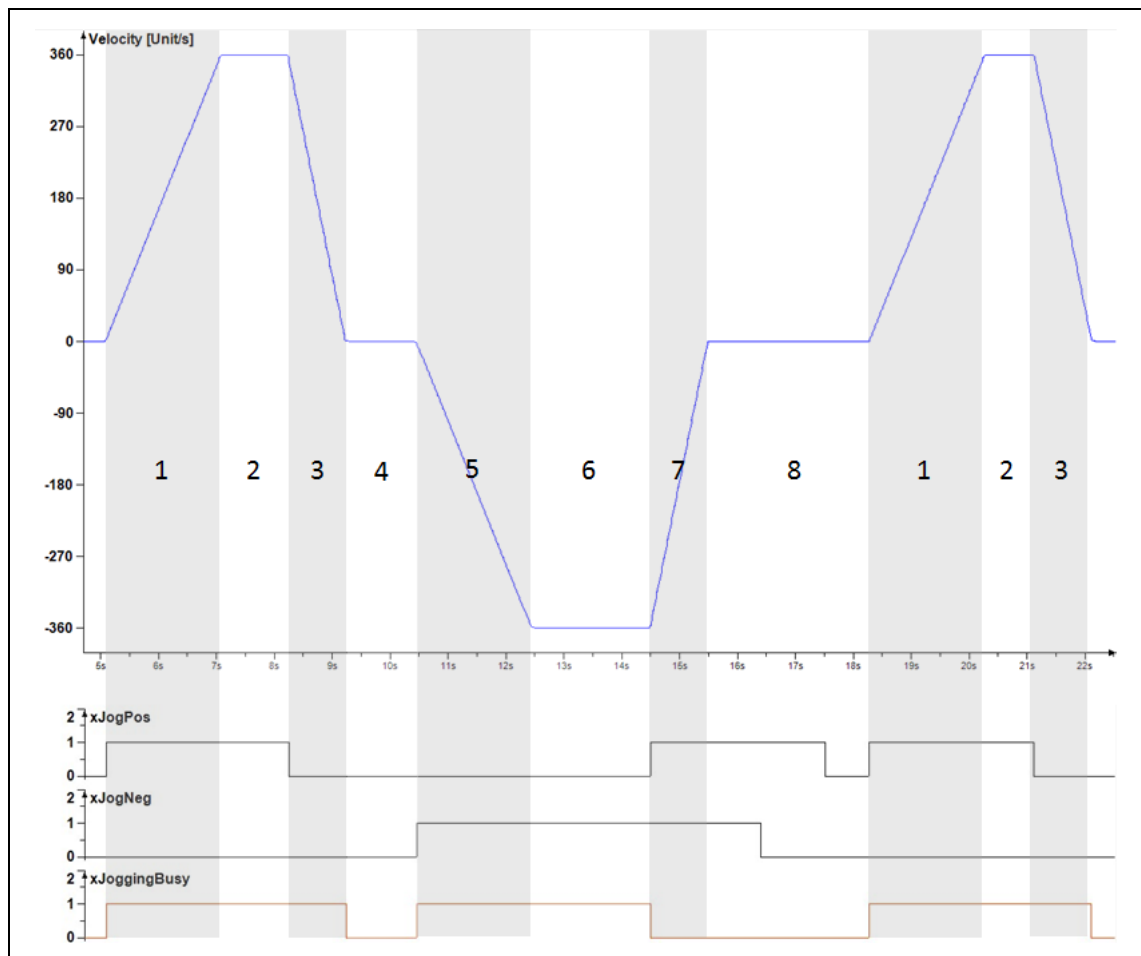
Execution

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 *xJogNeg* = 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.

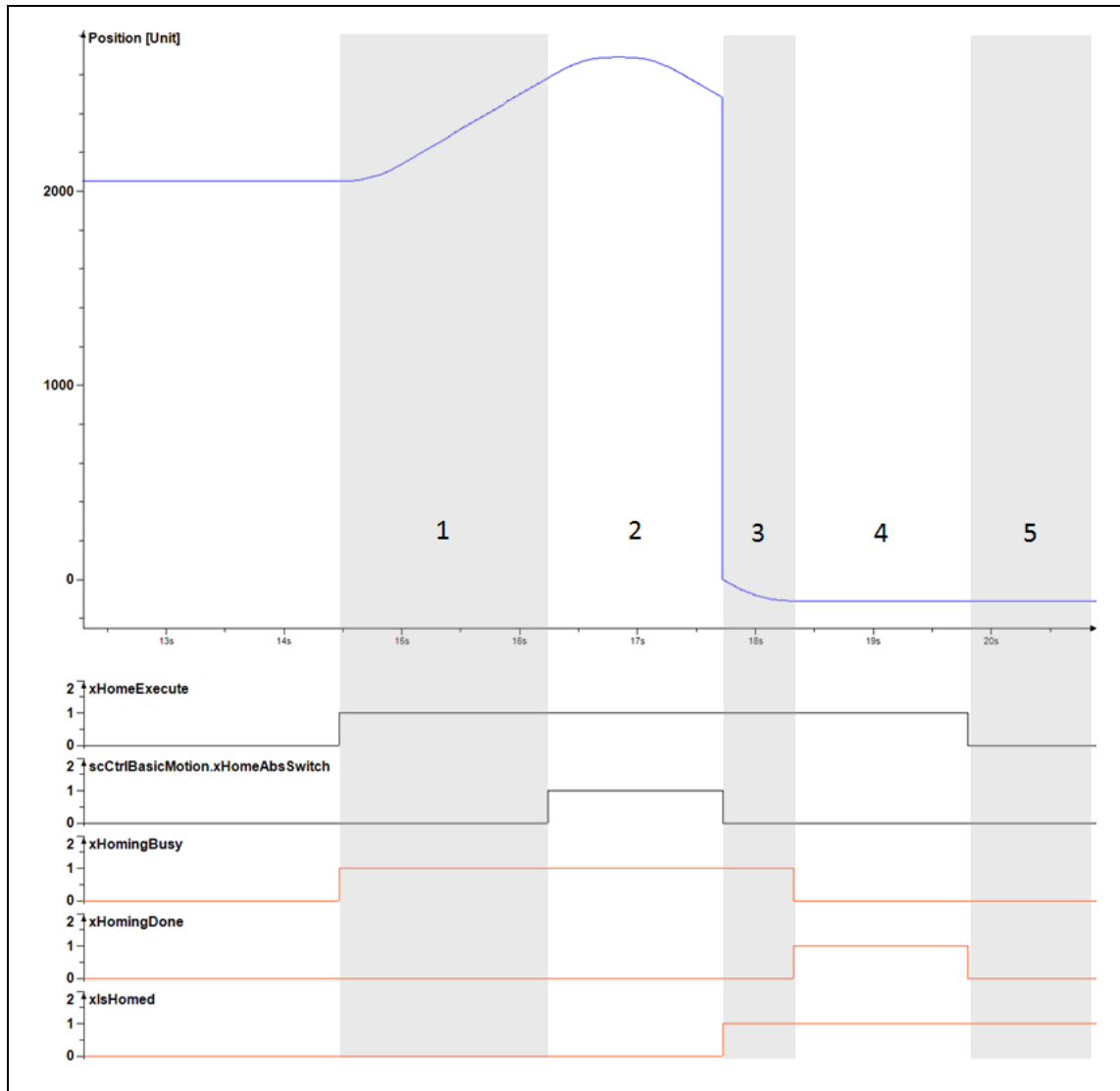
If *xHomeExecute* = FALSE, the state is changed to "STANDBY" and *xHomingDone* is set = FALSE.

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

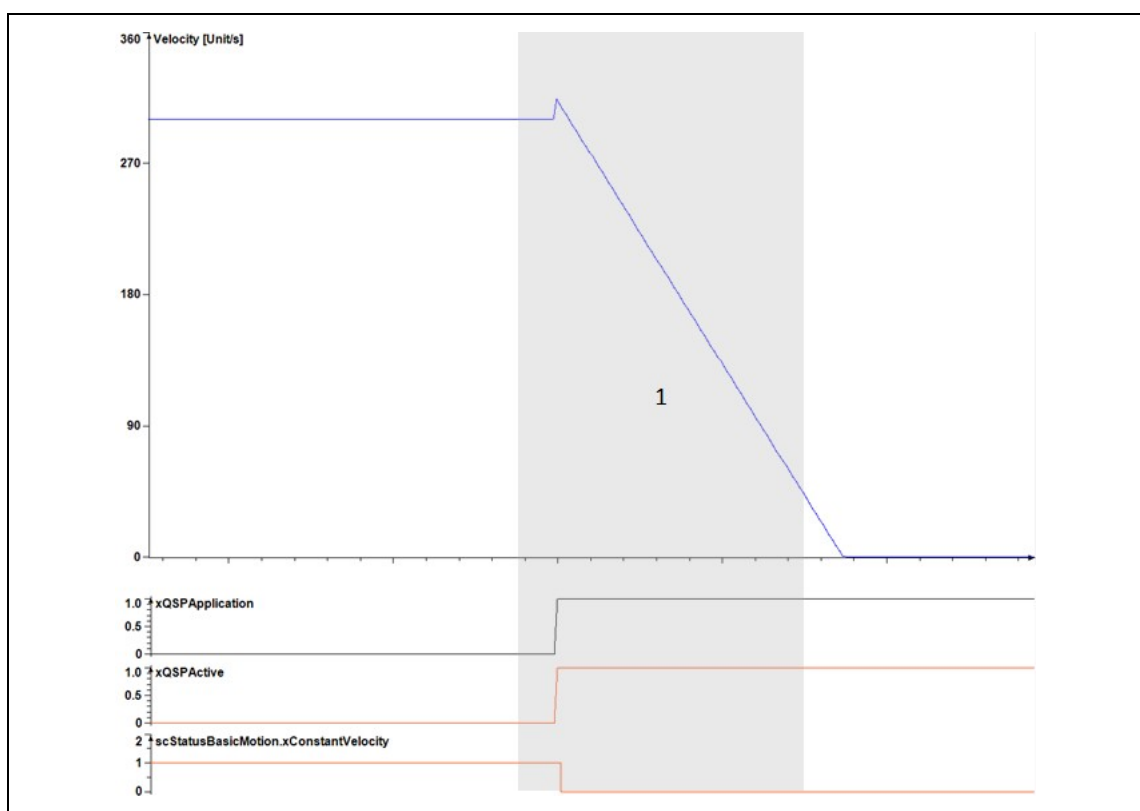
Prerequisite

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

Execution

With a rising edge (FALSE→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→TRUE	FALSE→TRUE	TRUE→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

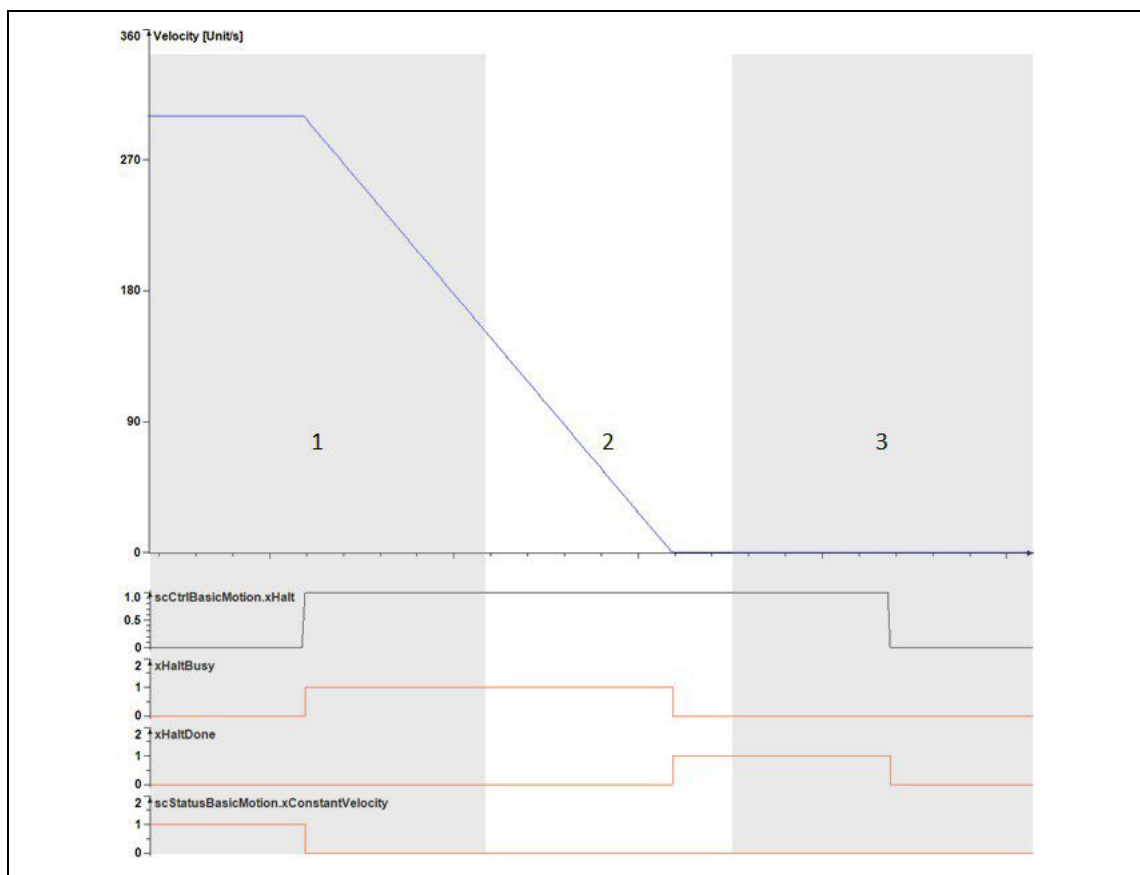
Prerequisite

- 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	$TRUE \rightarrow FALSE$	$FALSE \rightarrow TRUE$	$TRUE \rightarrow FALSE$	$FALSE$	$STOP \rightarrow STANDBY$	$xHalt = FALSE$ $xHaltDone = TRUE \rightarrow FALSE$.

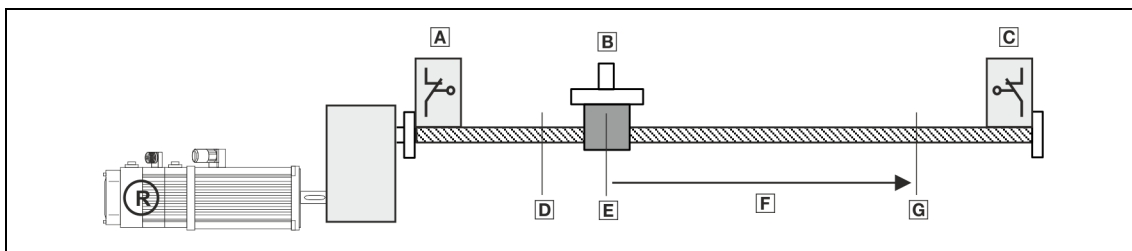
Parameters to be set

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.

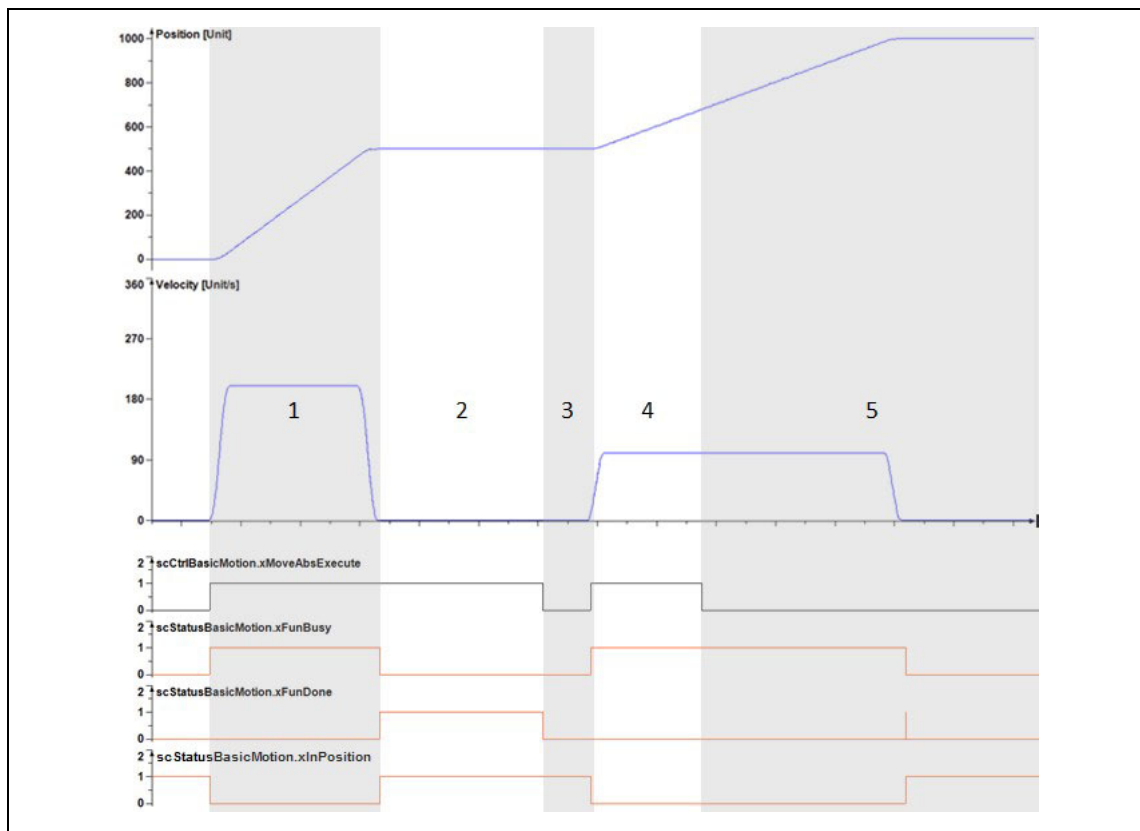
In order to reach the target position, the axis is brought to a standstill via the *lrDec* parameter. The axis reaches a standstill exactly in the target position.

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

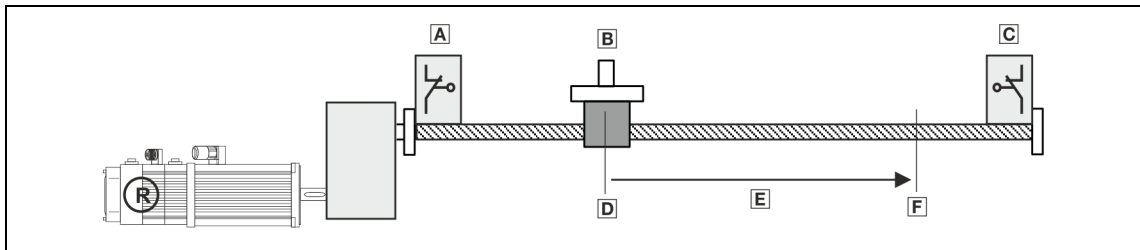
- *IrVel*, initial value: 3600 [units/s]
- *IrAcc*, initial value: 36000 [units/s²]
- *IrDcc*, initial value: 36000 [units/s²]
- *IrJerk*, initial value: 0 [units/s³]

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.

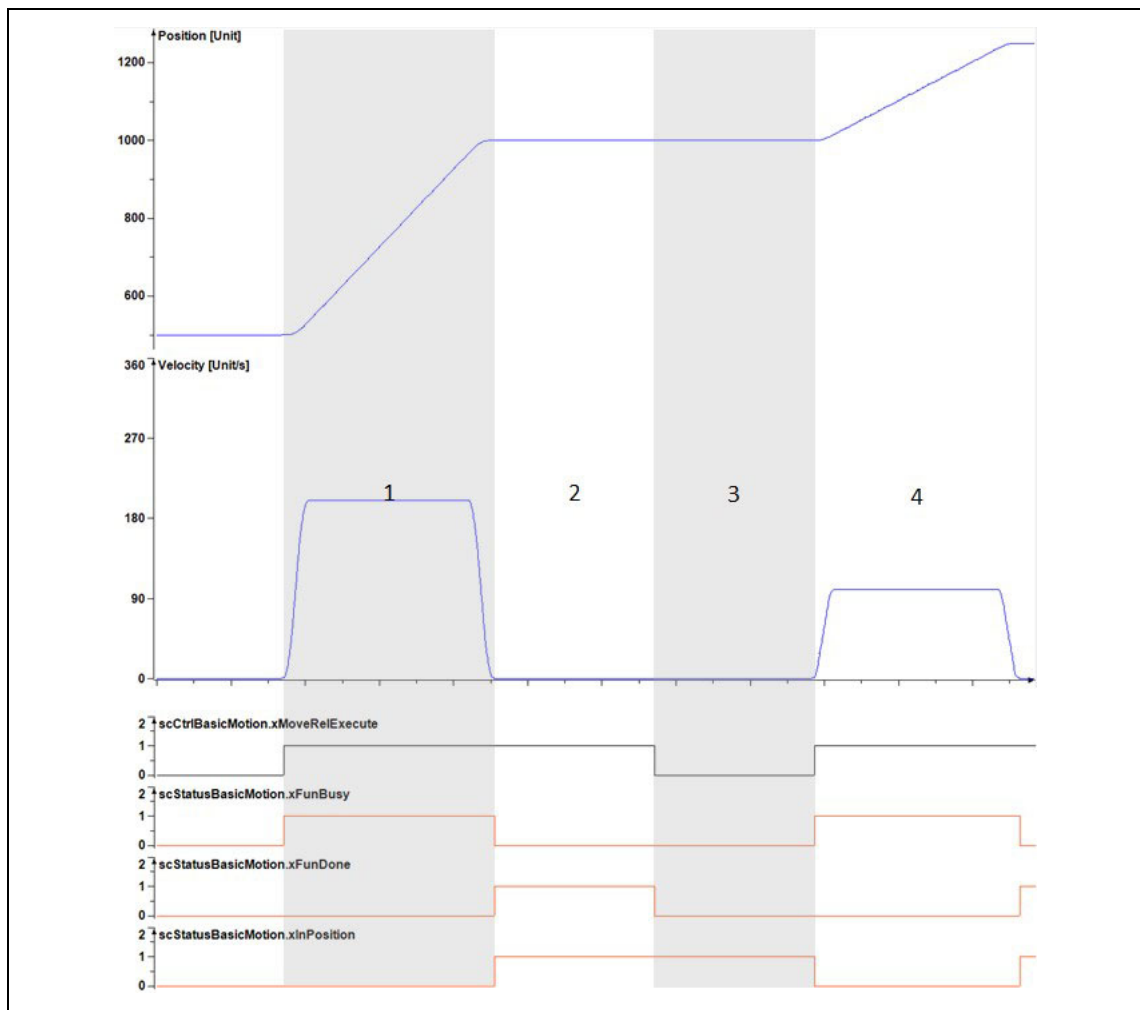
In order to reach the target position, the axis is brought to a standstill via the *lrDec* parameter. The axis reaches a standstill exactly in the target position.

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:

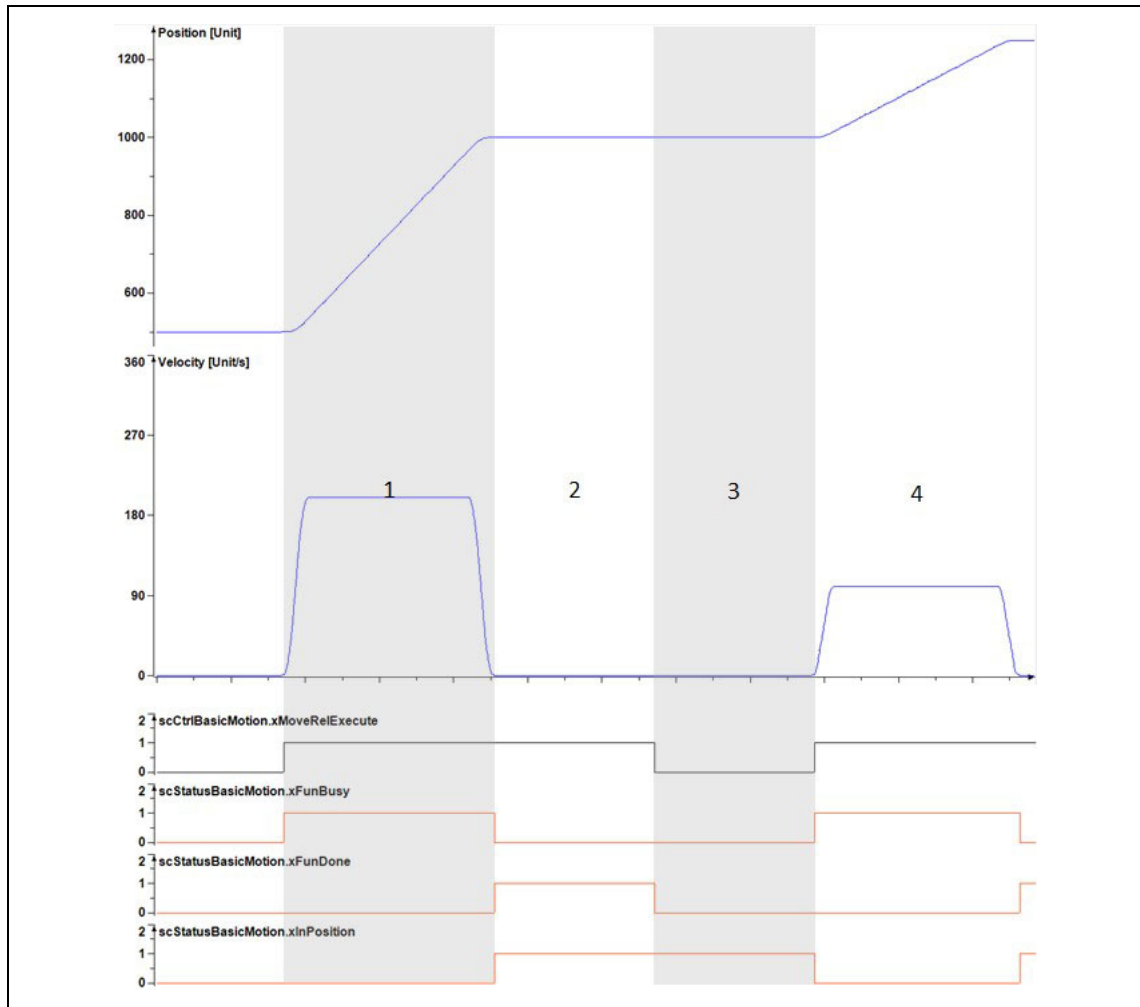
- Input *xDisableSWLimit* = FALSE
- Output *xSwLimitEnabled* = TRUE
- Output *xSwLimitSwitchPos/Neg* = TRUE
- Output *xHwLimitSwitchPos/Neg* = TRUE

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	$xMoveVelExecute$	$xAbort$	$xFuncBusy$	$xFuncDone$	$xConstantVelocity$	$eTMState$	Info
1	FALSE $\nearrow$ TRUE $\searrow$ FALSE	FALSE	FALSE $\nearrow$ TRUE	FALSE	FALSE	STANDBY $\rightarrow$ MOVEVEL	The axis accelerates to the target speed of 200 units/s.
2	FALSE	FALSE	TRUE	FALSE $\nearrow$ TRUE	FALSE $\nearrow$ TRUE	MOVEVEL	After reaching the target speed, $xFuncDone$ changes from FALSE $\nearrow$ TRUE and $xConstantVelocity$ changes from FALSE $\nearrow$ TRUE.
3	FALSE	FALSE $\nearrow$ TRUE	TRUE $\searrow$ FALSE	TRUE $\searrow$ FALSE $\nearrow$ TRUE	TRUE $\searrow$ FALSE	MOVEVEL $\rightarrow$ STANDBY	The speed run is terminated by $xAbort = TRUE$ and brought to a standstill with the profile deceleration. When standstill has been reached, $xFuncBusy$ 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**

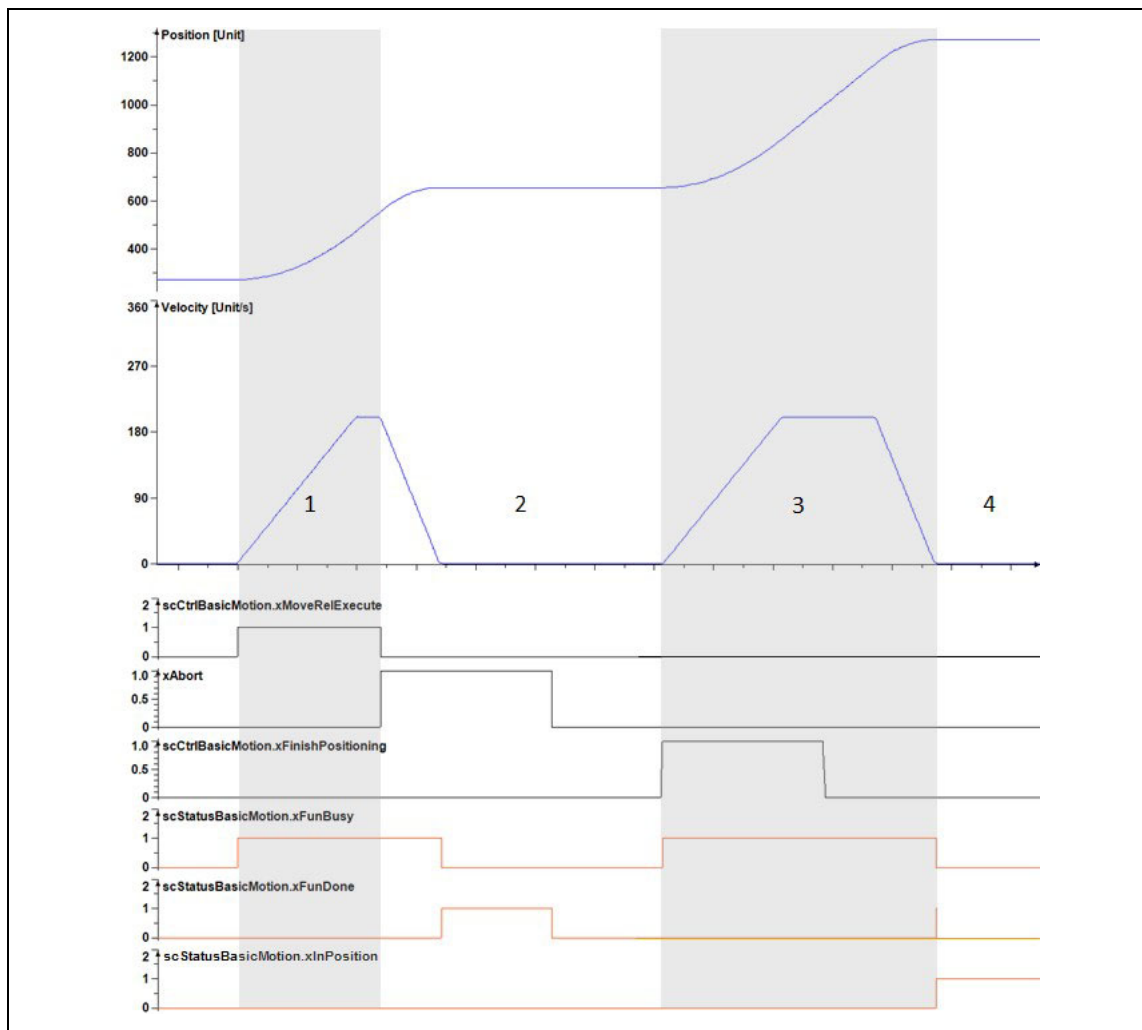
Both technology modules provide several profile data sets, one of which can be active.

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 ($xReadyForMotion = \text{TRUE}$).

Execution

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

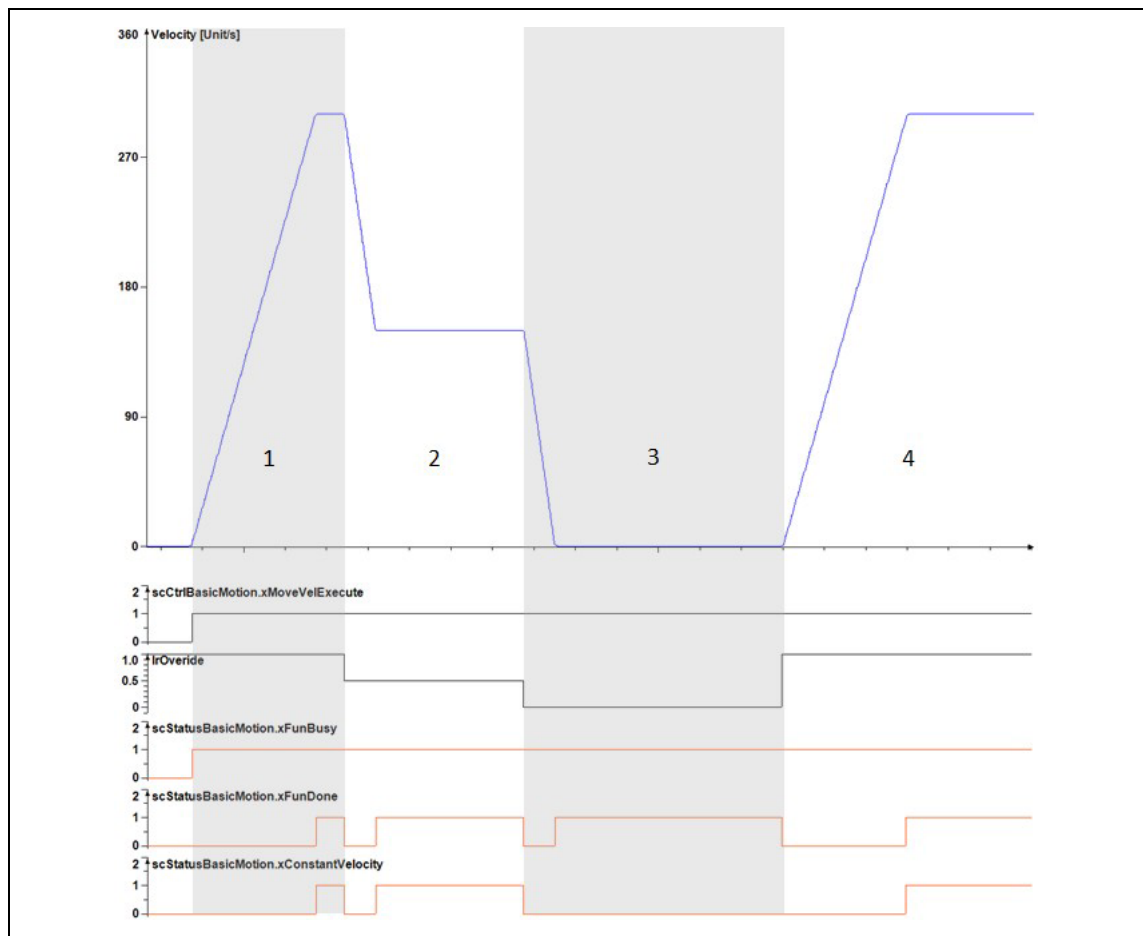
- $IrOverride$, initial value: 1.0
- $scCtrlBasicMotion.IrOverrideAcc$, initial value: 1.0 (100 %)
- $scCtrlBasicMotion.IrOverrideJerk$, initial value: 1.0 (100 %)

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

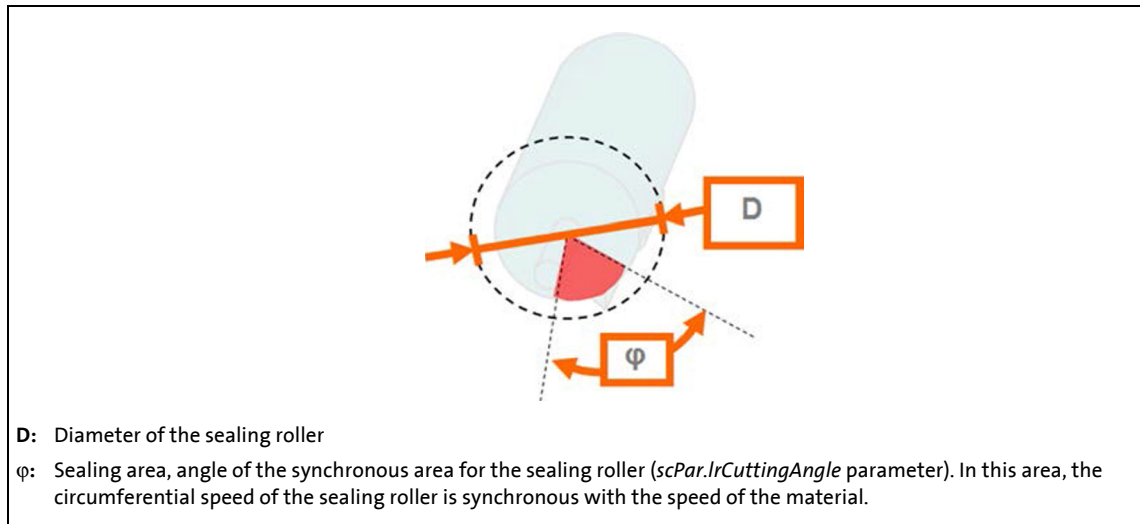
8.9 Override for speed

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 Parameterisation of the sealing roller

The technology module calculates the cam for synchronous motion of a cross sealing axis and synchronises it with a master axis.



[9-1] Definition of the sealing roller diameter and the sealing area

In order to calculate the synchronous circumferential speed, the diameter ($D \times \Pi$) of the sealing roller is specified using the *scPar.IrToolCircumference* parameter in the master axis unit (e.g. mm). The value can no longer be changed during the cutting operation.

The sealing area is specified via the *scPar.IrCuttingAngle* parameter in the (angle) degree unit. The sealing area can be changed at any time during operation and thus adjusted to the cutting process. New values are always accepted after the synchronous phase.

The permissible value range for the *scPar.IrCuttingAngle* parameter is from 1 to 359 per number of tools (*scPar.byNumOfTools* parameter).

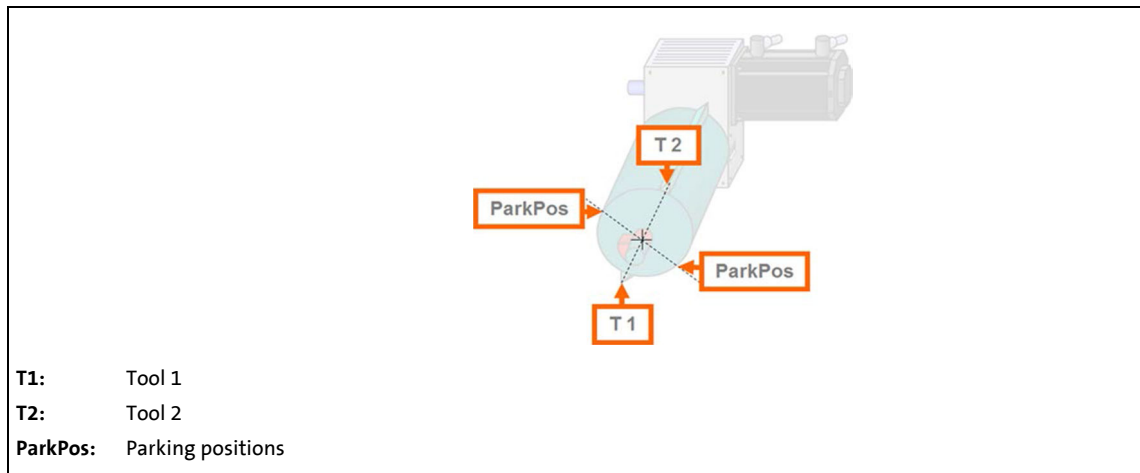
Parameters relevant for the function

- *scPar.xMasterAbsolute*, initial value: *TRUE*
- *scPar.IrToolCircumference*, initial value: 75 55555 unit
- *scPar.IrCuttingAngle*, initial value: 10°
- *scPar.byNumOfTools*, initial value: 1

9.2 Positioning to the parking position

The technology module supports the calculation of the profiles of several tools (e.g. cutting blades) on the sealing roller. The number of tools per sealing roller is specified by the *scPar.byNumOfTools* parameter. The first tool is always on the zero coordinate of the sealing roller and is defined as tool number 1. Any further tools are distributed symmetrically on the roller circumference in positive cutting direction.

The parking positions are exactly in the middle between two adjacent tools. The parking position and the position of the tools is thus defined automatically and cannot be moved.



[9-2] Definition of the sealing roller diameter and the sealing area

The tool to be used for the first cut into the material is selected via the *scPar.byStartToolNo* parameter. This results in the associated parking position in front of the selected tool.

If the tools on the sealing roller are not different, set the automatic tool selection by setting the *scPar.byStartToolNo* parameter to 0. Then the next possible parking position will be used as target parking position.

A rising edge ($\text{FALSE} \rightarrow \text{TRUE}$) at the *xExecuteMoveToParkPos* input starts positioning the slave axis into the parking position. The slave axis is then moved to the parking position via the motion parameters *scPar.eDirection*, *scPar.lrvVel*, *scPar.lrvAcc* and *scPar.lrvDec*.

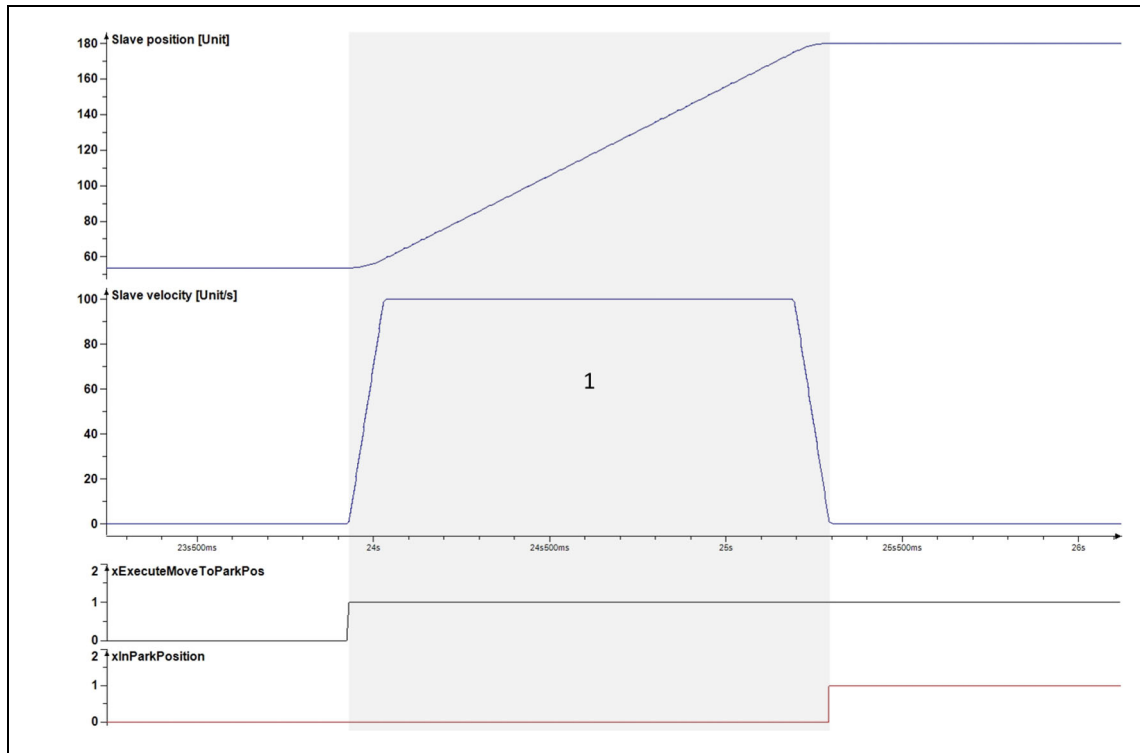
Inputs relevant for the function

- *xExecuteMoveToParkPos*

Parameters relevant for the function

- *scPar.byNumOfTools*, initial value: 1
- *scPar.byStartToolNo*, initial value: 0
- *scPar.eDirection*, initial value: *mcPositiveDirection*
- *scPar.lrvVel*, initial value: 3 600 unit/s
- *scPar.lrvAcc*, initial value: 36 000 unit/s²
- *scPar.lrvDec*, initial value: 36 000 unit/s²

Example



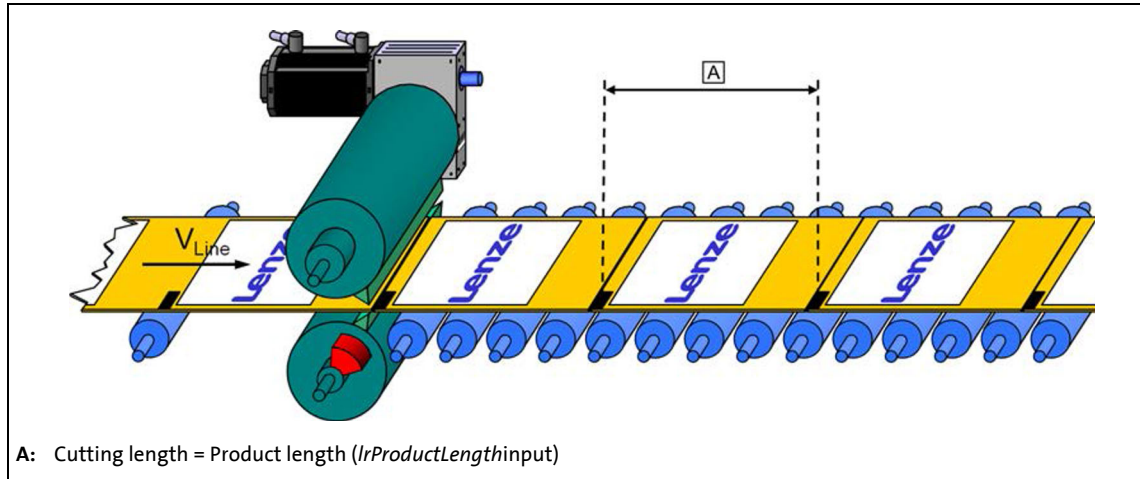
[9-3] Positioning to the parking position for a tool

No	xExecuteMoveToParkPos	xInParkPosition	INFO
1	FALSE → TRUE	FALSE	The cross cutter is moved to the parking position.
2	TRUE	TRUE	The parking position has been reached.

9.3 "Length cutting" operating mode

The profiles are calculated in real time in the "Length cutting" operating mode. This means that the cutting lengths and the synchronous phases can be changed flexibly during the cutting process.

The "Length cutting" operating mode is selected by setting the *scPar.eCuttingMode* parameter to 0 (Length cutting).



[9-4] Definition of the sealing roller diameter and the sealing area

The product/material is cut to the length defined in the *lrProductLength* input. The length is defined in the unit of the master axis (e.g. mm).

All changes are accepted after the synchronous phase and are effective for the next cut. The synchronous speed of the sealing roller is calculated depending on the circumference of the roller in the *scPar.lrToolCircumference* parameter, the speed of the master axis and the cycle length of the sealing roller.

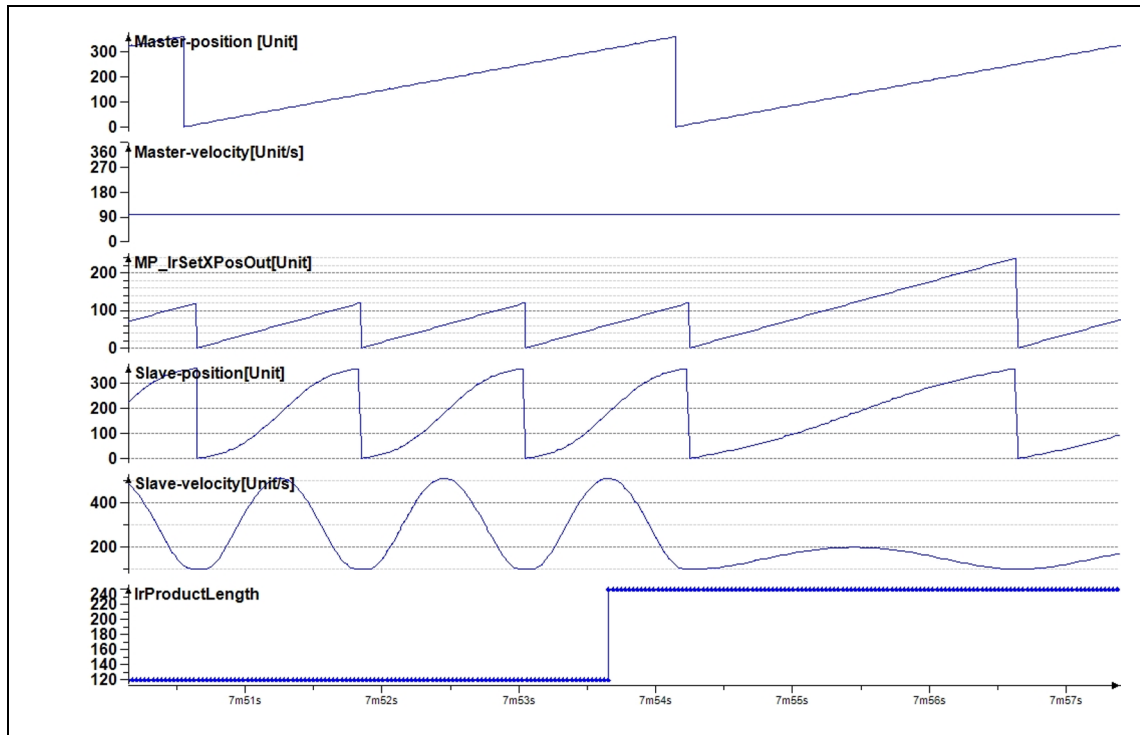
Inputs relevant for the function

- *lrProductLength*

Parameters relevant for the function

- *scPar.eCuttingMode*, initial value: 0 (length cutting)
- *scPar.lrToolCircumference*, initial value: 75.55555 unit
- *scPar.xMasterAbsolute*, initial value: TRUE
- *scPar.lrStartXPosRelative*, initial value: 0 unit
- *scPar.lrCuttingAngle*, initial value: 10°
- *scPar.byNumOfTools*, initial value: 1
- *scPar.byStartToolNo*, initial value: 0

Example



[9-5] Length-controlled cutting process

No	IrProductLength	INFO
1	120	The cross cutter cuts the cutting length = 120. The x axis is hereby moved from 0 to 120 (<i>MP_IrSetXPosOut</i>).
2	240	The cutting length = 240 is accepted in the synchronous area. The x axis is moved accordingly from 0 to 240.

9.3.1 Top cut

Only if the "Length cutting" operating mode is active (*scPar.eCuttingMode* parameter = 0).

The first cutting length of the material is called top cut. The top cutting length can be defined in the technology module via the parameterisation.

If the *xSyncln* input is set to TRUE, the material speed is integrated and thus the resulting position of the material which is depicted on the position on the X axis (*lrSetXPosOut* output) is calculated.

The end position on the X axis and thus the cutting position of the material results from the sum of the starting position for the X axis (*lrStartXPosRelative* parameter) and half of the path of the synchronous phase. Therefore, after the FALSE→TRUE edge at the *Syncln* input, the X axis will cover the following distance until the cut:

$$lrXDist = lrMasterSynclnDist \times \frac{lrCuttingAngle}{2} \times \frac{lrCrossCutterCircumference}{360} \times lrSetOffsetMaster$$

Inputs relevant for the function

- *xSyncln*

Outputs relevant for the function

- *lrSetXPosOut*

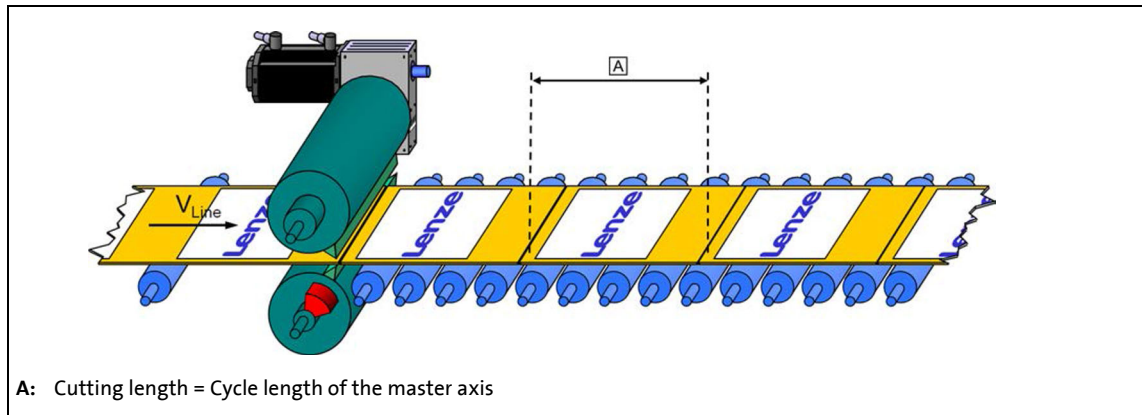
Parameters relevant for the function

- *scPar.eCuttingMode*, initial value: 0 (length cutting)
- *lrStartXPosRelative*, initial value: 0

9.4 "Flow packer" operating mode

In the "Flow packer" operating mode, the cycle length of the master axis is decisive for the cut. In this case, the cross cutter is coupled to the master axis via the cycle length. The technology module always makes a cut at the end of the cycle of the master axis, independent of the product length set at the *lrProductLength* input.

The "Flow packer" operating mode is selected by setting the *scpar.eCuttingMode* parameter to 1 (Flow packer).



[9-6] Cutting length in the "Flow packer" operating mode

It is necessary to set the product length at the *lrProductLength* input to calculate the synchronous speed. The synchronous speed of the sealing roller is calculated in the following dependencies:

- Circumference of the roller in the *scPar.lrToolCircumference* parameter [unit]
- Speed of the master axis
- Cycle length of the master axis
- Cycle length of the sealing roller

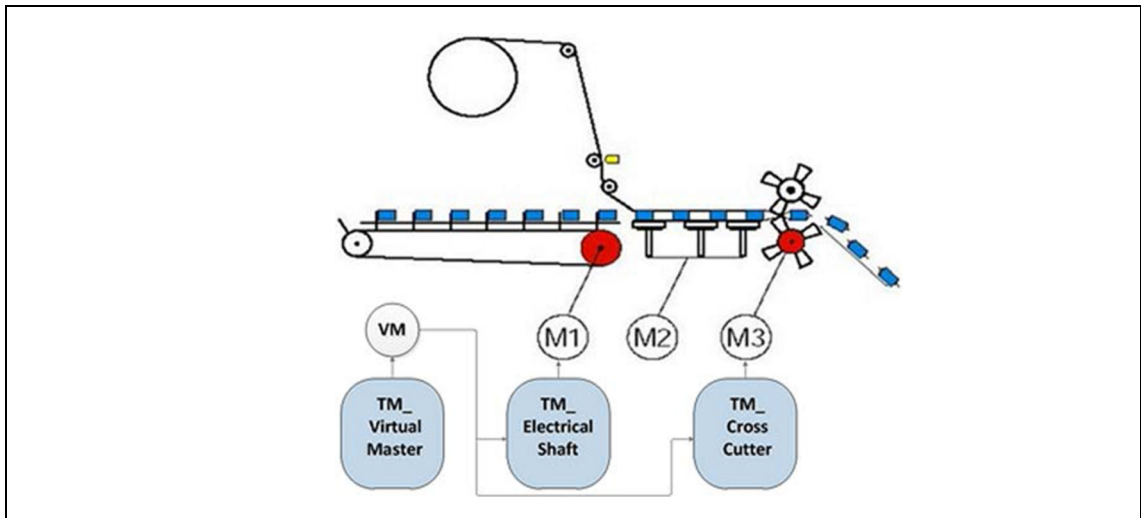
Inputs relevant for the function

- *lrProductLength*

Parameters relevant for the function

- xMasterAbsolute, initial value: TRUE
- lrStartXPosRelative, initial value: 0 unit
- eSyncMode, initial value: 1
- scPar.lrToolCircumference, initial value: 75.55555 unit
- byNumOfTools; initial value: 1
- scPar.eCuttingMode, initial value:

Example



[9-7] Sample structure: "Flow packer"

The "Virtual Master" technology module (TM) controls a virtual axis with the modulo cycle 0 ... 360°. The virtual axis is the virtual master for all workstations in the "Flow packer".

The conveyor belt M1 is controlled by the "Electrical Shaft Position" TM. The cycle length of M1 is defined equal to the length of the tray in which the product is transported (here 100 mm).

When the master performs a cycle revolution of 360°, the conveyor belt travels 100 mm, in other words the exact length of a tray.

The M3 cross cutter is operated by the "Cross Cutter" TM in the "Flow packer" operating mode. The product length of 100 mm is transferred to the "Cross Cutter" TM. Thus the TM also specifies the correct synchronous speed for the cut. The cross cutter makes a cut at the end of the cycle of the virtual master, i.e. at position 360.

The virtual master thus controls the entire cutting process. With each cycle of the virtual master, the conveying belt covers another 100 mm and the cross cutter makes a cut.

Calculation of the synchronous speed from the cycle speed of the master axis

$$V_{\text{sync}} = \frac{L_{\text{Prod}}}{L_{\text{MCycle}}} \times V_{\text{M, Sync}}$$

V_{sync} : Cycle speed of the material [mm/s]

L_{Prod} : Product length [mm]

L_{MCycle} : Length of the master clock pulse [deg]

$V_{\text{M, Sync}}$: Cycle speed of the master axis [deg/s]

The synchronous speed can equally be derived from the angular velocity of the cross-cutter axis:

$$V_{\text{cc, sync}} = \frac{\text{Circ}_{\text{cc}}}{L_{\text{SCycle}}} \times V_{\text{cc}}$$

$V_{\text{cc, sync}}$: Synchronous circumferential speed of the cross-cutter axis [mm/s]

Circ_{cc} : Circumference of the cross-cutter axis [mm]

L_{SCycle} : Length of the slave clock pulse [deg]

V_{cc} : Angular velocity of the cross-cutter axis [deg/s]

9.5 Positioning the cross cutter to static material

The "Positioning to static material" clutching type is selected by setting the *scPar.eSyncMode* parameter to 1 (Positioning).

The cross cutter is synchronised to the material position (X position and resulting Y position from the cam) using the input *SyncIn* = TRUE.

There are two possibilities to specify the current material position:

- *scPar.xMasterAbsolute* parameter = FALSE. The material position is defined via the *scPar.lrStartXPosRelative* parameter.
- *scPar.xMasterAbsolute* parameter = TRUE. This parameter sets the current material position to the position of the master axis and thus defines an absolute reference between these two positions for positioning the slave axis. The slave axis is positioned from its current position to the resulting Y position from the cam via the parameters *scPar.eDirection*, *scPar.lrVel*, *scPar.lrAcc* and *scPar.lrDec*.

The positioning process is performed in the "POSITIONING_TO_START_POS" state. **During positioning, the master axis, and thus the material too, may not move!** After successful clutching, the *xSynchronised* output is set to TRUE and the TM changes to the "CUTTING" state.

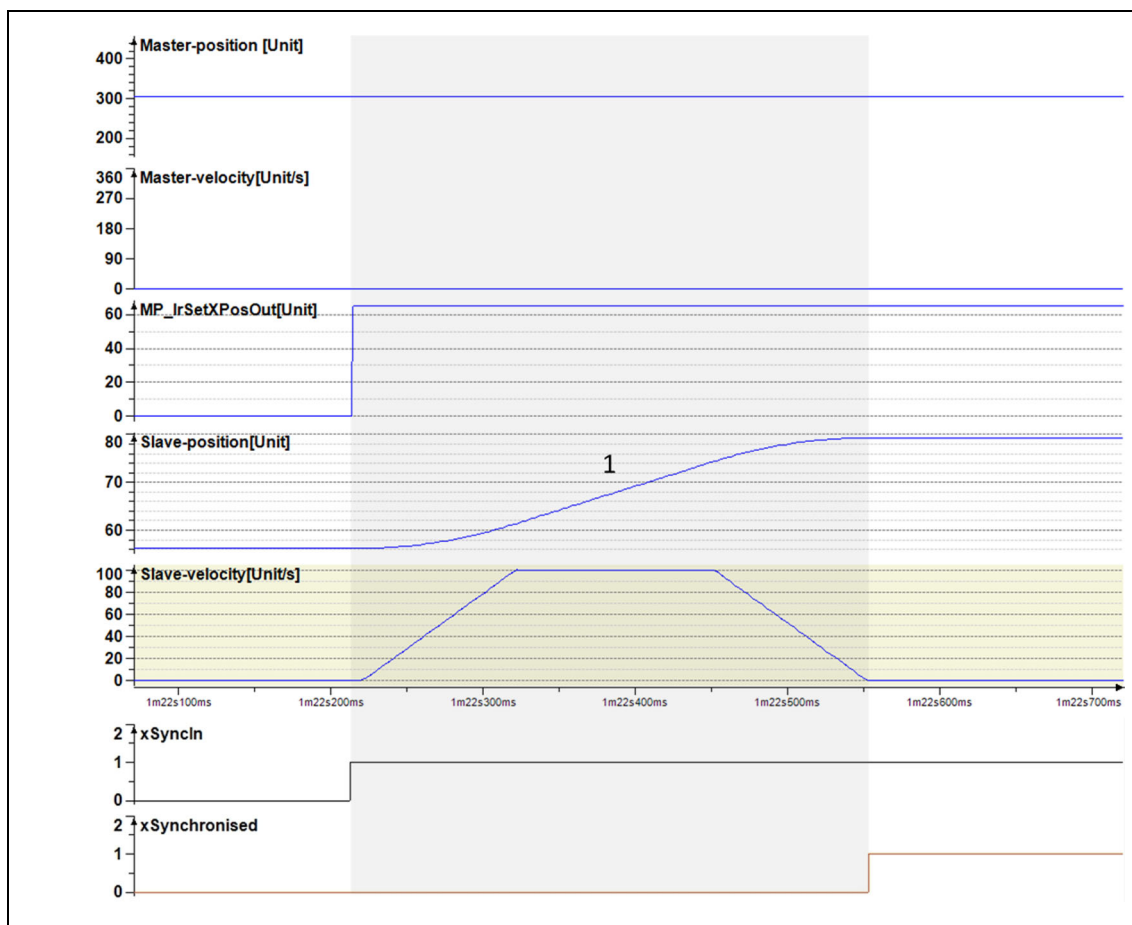
Inputs relevant for the function

- *xSyncIn*

Parameters relevant for the function

- *scPar.eSyncMode*, initial value: 0
- *scPar.xMasterAbsolute*, initial value: TRUE
- *scPar.lrStartXPosRelative*, initial value: 0
- *scPar.eDirection*, initial value: *mcPositiveDirection*
- *scPar.lrVel*, initial value: 3600
- *scPar.lrAcc*, initial value: 36 000
- *scPar.lrDec*, initial value: 36 000

Example



[9-8] Positioning to the cam

No	xSyncln	xSynchronised	INFO
1	FALSE→TRUE	FALSE	The cutter is synchronised to the master position.
2	TRUE	TRUE	The TM is coupled to the master axis via the cam.

9.6 Clutching the cross cutter to moving material

For the clutch-in process to moving material, the cross cutter must be in the parking position (*xExecuteMoveToParkPos* input = TRUE).

Positioning to the parking position

The "Clutching to moving material" clutching type is selected by setting the *scPar.eSyncMode* parameter to 1 (Cam_in).

The cross cutter is synchronised to the material position (X position and resulting Y position from the cam) using the input *SyncIn* = TRUE.

There are two possibilities to specify the current material position:

- *scPar.xMasterAbsolute* parameter = FALSE. The material position is defined via the *scPar.lrStartXPosRelative* parameter.
- *scPar.xMasterAbsolute* parameter = TRUE. This parameter sets the current material position to the position of the master axis and thus defines an absolute reference between these two positions for positioning the slave axis. The slave axis is clutched-in via the calculated start cam in the synchronous phase from the parking position.

In the "Length cutting" operating mode (*scPar.eCuttingMode* parameter = 0) the clutch-in distance of the master axis can be set via the *scPar.lrMasterSyncInDist* parameter. Then the first cutting length is defined. Setting *scPar.lrMasterSyncInDist* to 0 serves to calculate the optimum clutch-in distance.

The clutch-in process is carried out in the "SYNCHRONISING_POS" status and the *xAccDecSync* output is set to TRUE. After successful clutch-in, the output *xSynchronised* = TRUE and the "CUTTING" status is set.

Inputs relevant for the function

- *xExecuteMoveToParkPos*
- *xSyncIn*

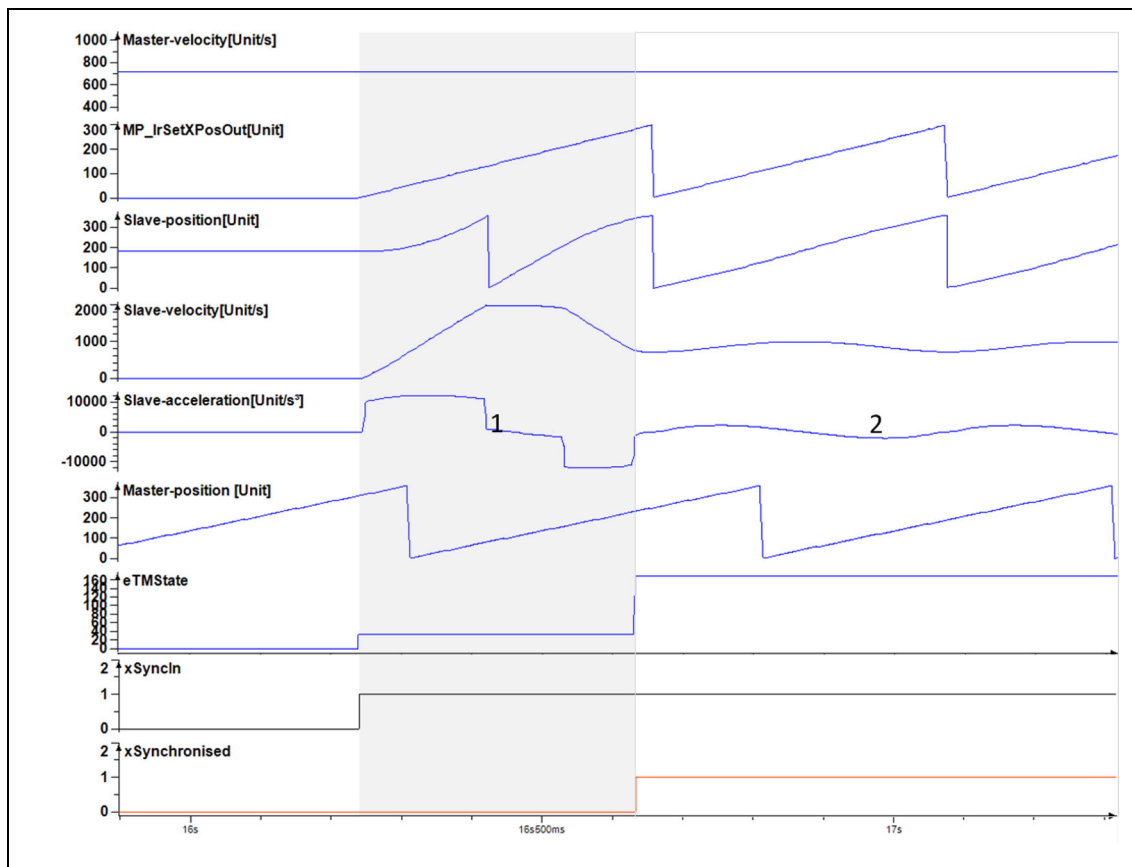
Outputs relevant for the function

- *xAccDecSync*
- *xSynchronised*

Parameters relevant for the function

- *scPar.xMasterAbsolute*, initial value: True
- *scPar.lrStartXPosRelative*, initial value: 0 unit
- *scPar.eSyncMode*, initial value: 0
- *scPar.lrMasterSyncInDist*, initial value: 0 unit
- *scPar.eCuttingMode*, initial value: 0

Example



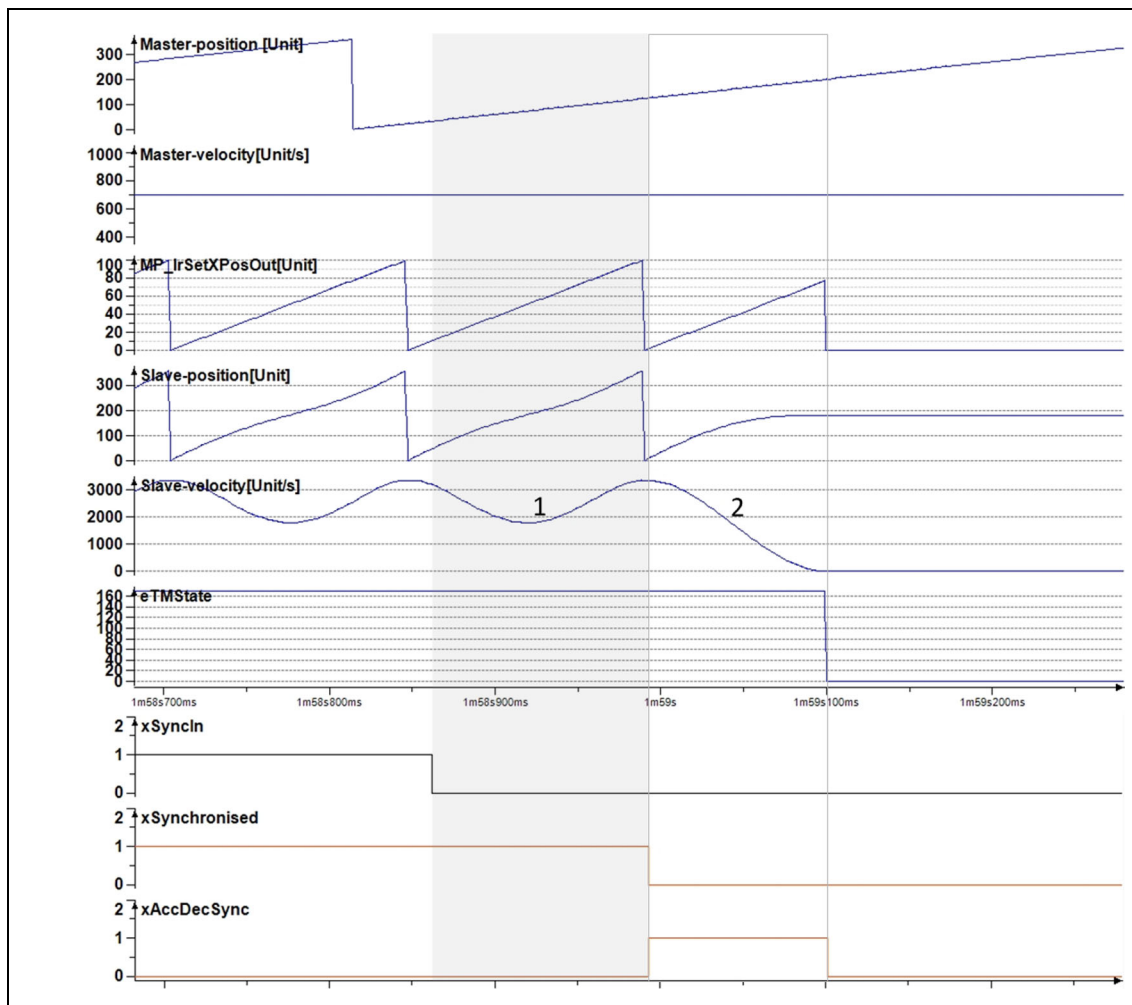
[9-9] Clutching to moving material

No	xSyncIn	xSynchronised	INFO
1	FALSE → TRUE	FALSE	The cross cutter is synchronised to the master axis via the cam.
2	TRUE	TRUE	The cross cutter is already synchronous with the master axis via the cam up to its entry into the synchronous area.

9.7 Declutching the cross cutter

The declutch process with *xSyncIn* input = FALSE can only be carried out if the master axis is moving. Its start is controlled by its position (by a 5th degree polynomial) after the synchronous phase. The cross cutter is hereby stopped in the parking position.

An ongoing declutch process is indicated via the *xAccDecSync* output = TRUE. After successful declutching, the *xAccDecSync* output = FALSE and the TM switches to the STANDBY state.



[9-10] Declutching from the cutting process

No	<i>xSyncIn</i>	<i>xSynchronised</i>	INFO
1	FALSE → TRUE	TRUE	The declutch process is requested
2	FALSE	TRUE → FALSE	The declutch process is executed
3	FALSE	FALSE	The slave axis is stopped in the parking position; its state changes to "STANDSTILL"

Inputs relevant for the function

- *xSyncIn*

Outputs relevant for the function

- *xAccDecSync*

9.8 Positive opening operation / Emergency opening operation

The slave axis is declutched and immediately stopped at the current curve position via the set deceleration for the requested function with the input *xQSPApplication* = TRUE, *xAbort* = TRUE or *scCtrlBasicMotion.xHalt* = TRUE.

The clutch remains open as long as one of the inputs *scCtrlBasicMotion.xHalt*, *xAbort* or *xQSPApplication* = TRUE. The functions *scCtrlBasicMotion.xHalt*, *xAbort* or *xQSPApplication* have a higher priority than the *xSyncIn* input.

Inputs relevant for the function

- *xQSPApplication*
- *xAbort*
- *scControlBasicMotion.xHalt*

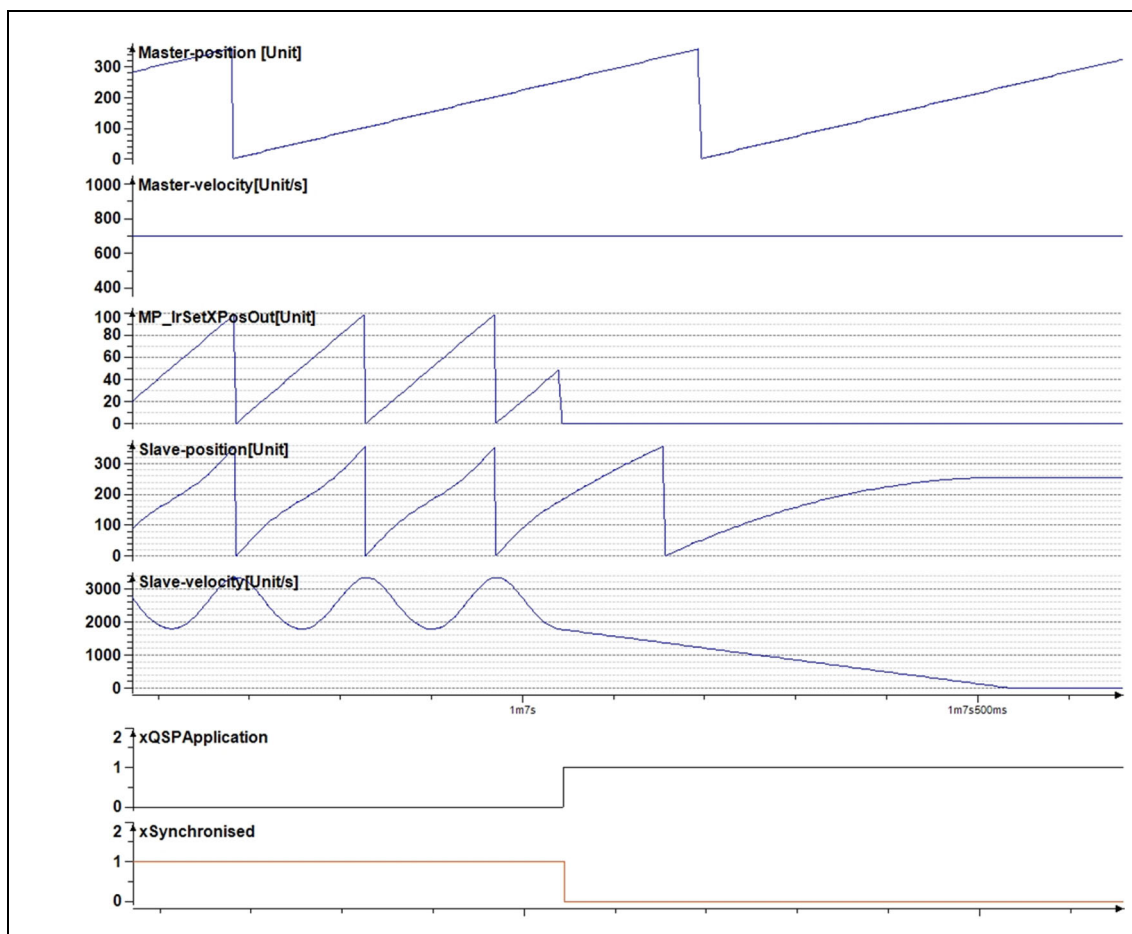
Outputs relevant for the function

- *xQSPActive*
- *xAbortBusy*
- *xAbortDone*
- *scStatusBasicMotion.xHaltBusy*
- *scStatusBasicMotion.xHaltDone*

Parameters relevant for the function

- QSP application deceleration, axis subindex: 048, initial value: 3600 unit/s²
- QSP application jerk, axis subindex: 049, initial value: 0 unit/s³
- Deceleration halt, axis subindex: 186, initial value: 1800 unit/s²
- Jerk halt, axis subindex: 187, initial value: 0 unit/s³

Example



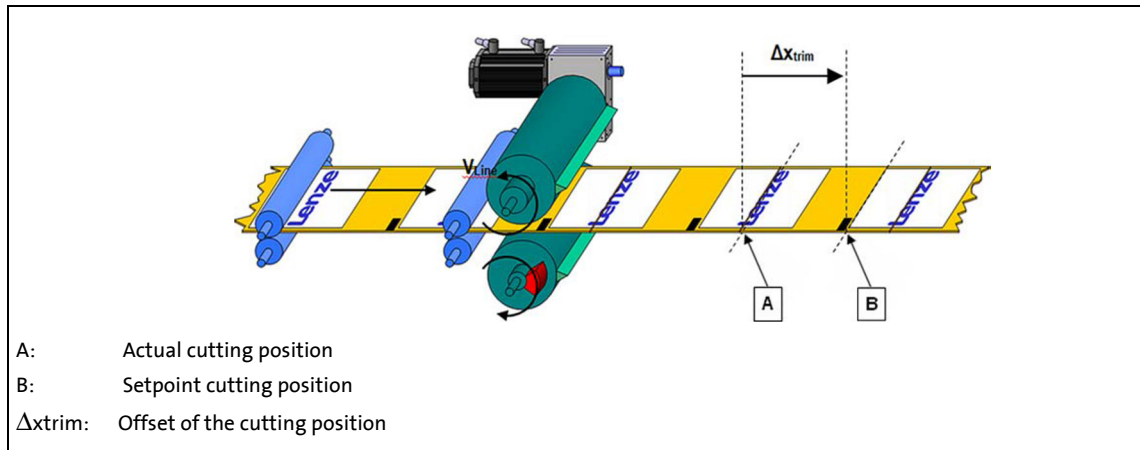
[9-11] Positive opening operation / Emergency opening operation

No	xQSPApplication	xSynchronised	INFO
1	FALSE	TRUE	The slave axis is coupled to the master axis via the cam
2	TRUE	FALSE	The slave axis is brought to an immediate standstill via the deceleration of the QSP application (Subindex 048).

9.9 Trimming and offset of the cutting position

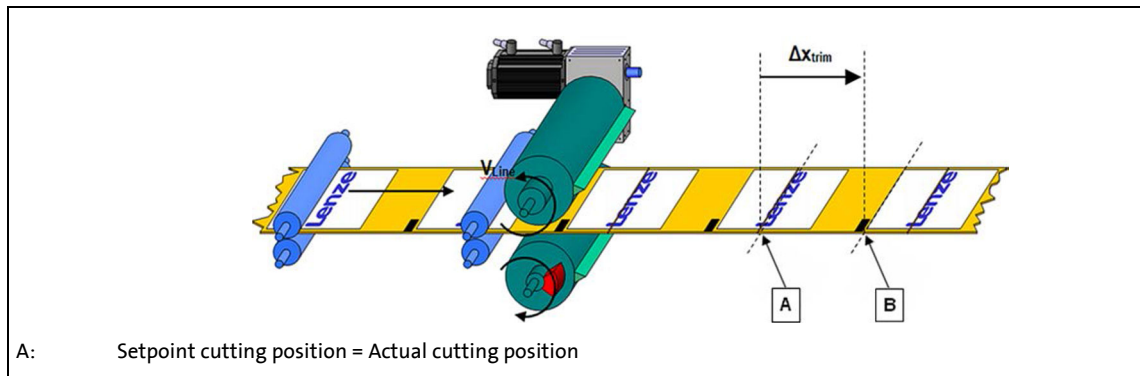
The trimming and offset functions provide a possibility to shift the cut in relation to the product cycle.

In the following figure, the cut is in the middle of the print image. In order to get the cut into the required position, the cross cutter drive must be trimmed by the corresponding differential travel Δx_{trim} with respect to the current cutting position. On completion of the trimming, the cut has been shifted (relative to the product cycle) and is ideally in the correct position.



[9-12] Cutting position: the cut is in the middle of the print image

Following figure: Setpoint cutting position = Actual cutting position

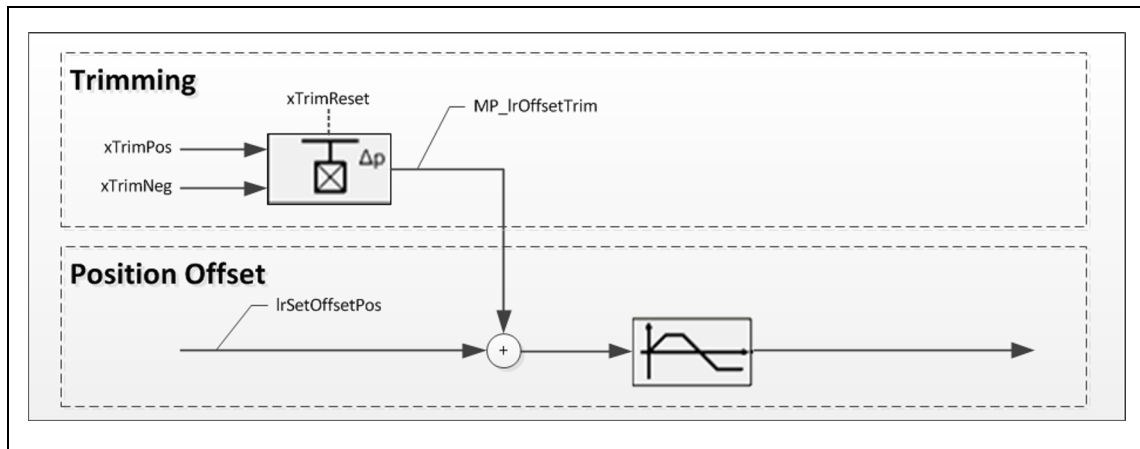


[9-13] Optimal cutting position: the cut is exactly at the cycle transition

Technology module functions for cut trimming and cut offset:

- Position cut trimming
- Speed cut trimming (Tip trimming)
- Absolute positioning of the cutting position via master offset

9.9.1 Position trimming



[9-14] Trim partial signal flow

Condition

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

Type

Position trimming makes it possible to adjust the position of the slave axis relative to the master axis. The position difference set in $scPar.lrTrimDist$ is trimmed in each case.

Increment trimming

Increment trimming can be selected by setting the $scPar.eTrimMode$ parameter to $L_TS2P_TrimMode.position$.

The trimming process corresponds to a relative positioning. Each positive edge at $xTrimPos$ increases the position to be trimmed by the value of $scPar.lrTrimDist$.

Speed trimming

Speed trimming can be selected by setting the $scPar.eTrimMode$ parameter to $L_TS2P_TrimMode.velocity$.

- Speed-based trimming is carried out in a positive direction as long as the $xTrimPos$ input is set.
- Speed-based trimming is carried out in a negative direction as long as the $xTrimNeg$ input is set.
- If both $xTrimPos$ and $xTrimNeg$ are set to TRUE, ...

The acceleration and velocity of the trimming superimpose the ones of the master axis. Hence, the results for the axis to be trimmed are as follows:

- resulting speed of: $v_{AxisRes} = v_{master\ axis} + lrSetOffsetTrimVel$
- resulting acceleration of: $a_{AxisRes} = a_{master\ axis} + lrSetOffsetTrimAcc$

Inputs relevant for the function

- *xTrimPos*
- *xTrimNeg*

Outputs relevant for the function

- *xOffsetTrimActive*
- *lrOffsetTotal*

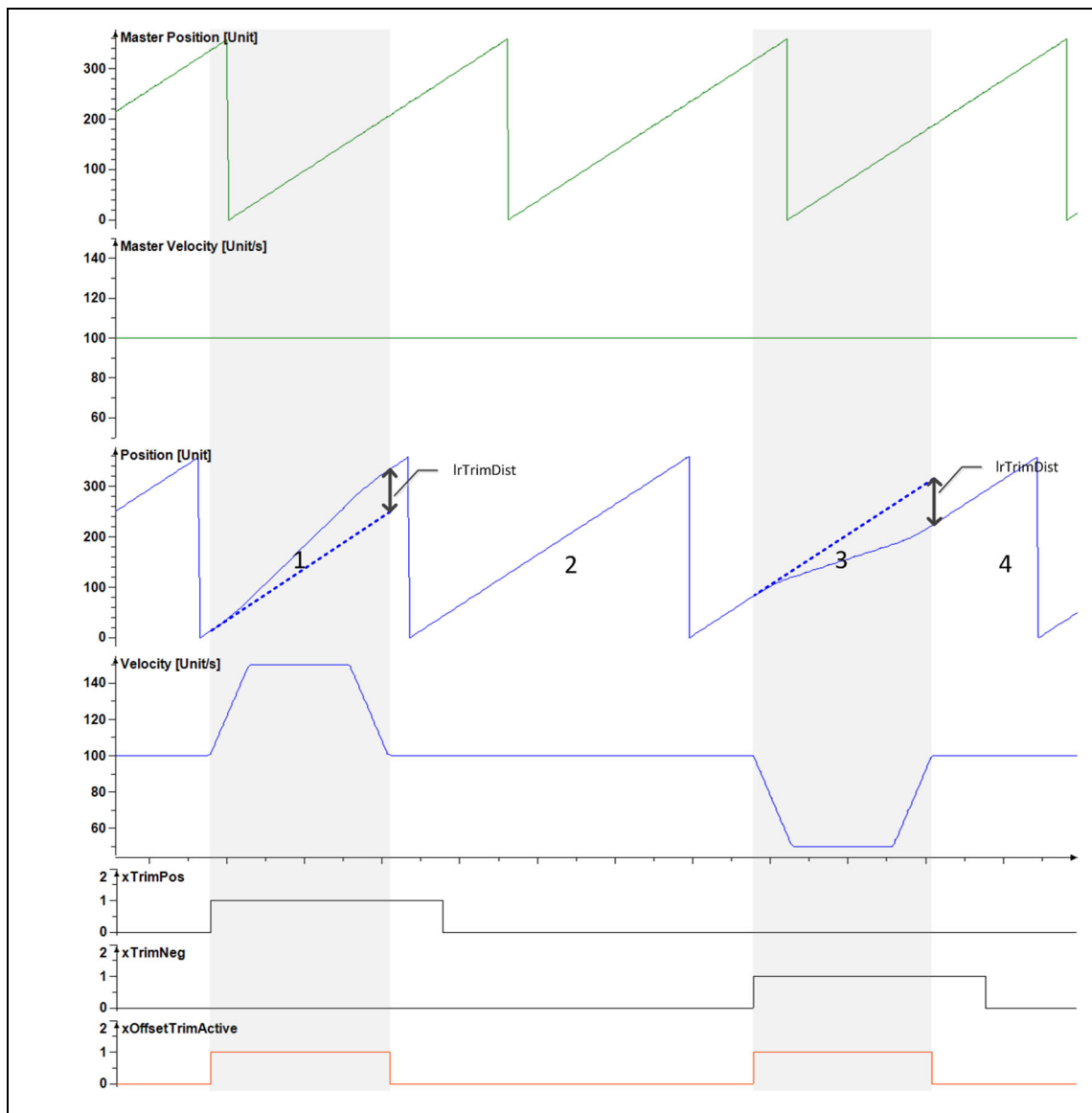
Parameters relevant for the function

- *scPar.eTrimMode*, initial value: 1 (trimming by position offset)
- *scPar.lrTrimDist*, initial value: 1 unit
- *scPar.lrSetOffsetTrimVel*, initial value: 10 unit/s
- *scPar.lrSetOffsetTrimAcc*, initial value: 100 unit/s²
- *scPar.lrSetOffsetTrimDec*, initial value: 100 unit/s²
- *scPar.lrSetOffsetTrimJerk*, initial value: 10 000 unit/s³

Measuring points relevant for the function

- *scSignalflow.MP_lrOffsetTrim*

Example



[9-15] Trimming in positive/negative direction by 90 units at a maximum trimming speed of 50 units/s

No	xTrimPos	xTrimNeg	xOffsetTrimActive	eTMState	INFO
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 a negative direction by -IrTrimDist = -90 units
4	FALSE	TRUE → FALSE	TRUE → FALSE	SYNCHRONIZED	Synchronous travel

9.9.2 Position offset via profile generator

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

The basic conditions for calculating the profile are specified via *scPar.IrSetOffsetTrimVel*, *scPar.IrSetOffsetTrimAcc*, *scPar.IrSetOffsetTrimDec* and *scPar.IrSetOffsetTrimJerk*.

Inputs relevant for the function

- *IrSetOffsetPosX*, initial value: 0 unit

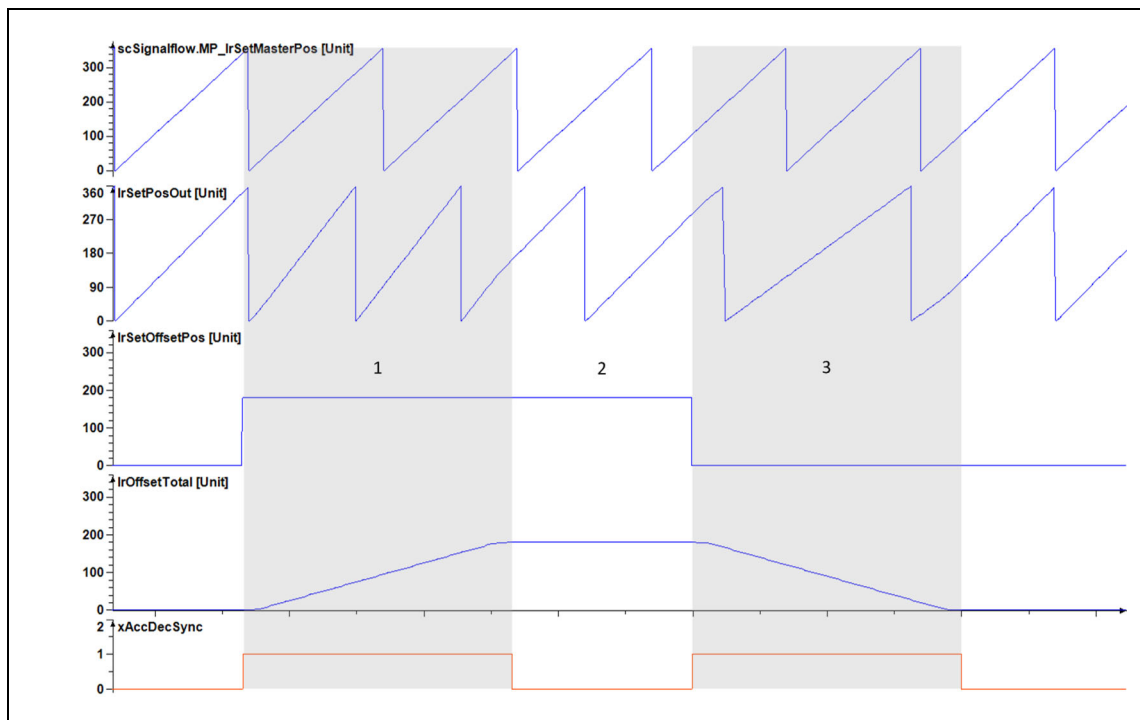
Outputs relevant for the function

- *IrSetOffset*, initial value: 0 unit
- *IrSetOffsetTotal*, initial value: 0 unit

Parameters relevant for the function

- *scPar.IrSetOffsetTrimVel*, initial value: 10 unit/s
- *scPar.IrSetOffsetTrimAcc*, initial value: 100 unit/s²
- *scPar.IrSetOffsetTrimDec*, initial value: 100 unit/s²
- *scPar.IrSetOffsetTrimJerk*, initial value: 10000 unit/s³

Example



[9-16] Single cycle operation

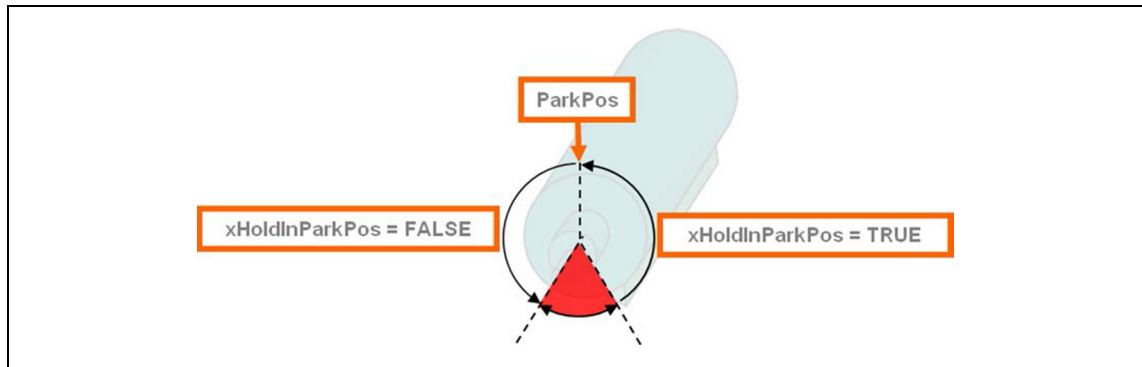
No	IrSetOffsetPosX	IrOffsetTotal	xOffsetTrimActive	INFO
1	0 units ↗ 200 units	0 units	FALSE ↗ TRUE	The offset is corrected using the offset and trim profile parameters with the help of a profile generator to the <i>IrSetOffsetPosX</i> target value (200 units)
2	200 units	200 units	TRUE ↘ FALSE	The offset is corrected
3	200 units ↘ 0 units	200 units	FALSE ↗ TRUE	The offset is corrected using the offset and trim profile parameters with the help of a profile generator to the <i>IrSetOffsetPosX</i> target value (0 units)

9.10 Method of elimination

The elimination function serves to leave out cuts in the material. For this purpose the cross cutter is brought to the parking position in order not to carry out the cutting process.

If the *xHoldInParkPos* input is set to TRUE, the cross cutter is declutched in the parking position after the synchronous phase. As soon as the parking position has been reached the *xInParkPos* output is set to TRUE. The position of the material continues to be determined. The cross cutter remains in the parking position until the *xHoldInParkPos* input is set to FALSE.

The cutting process is resumed after the synchronous phase, i.e. the cross cutter is clutched into the next synchronous phase.



[9-17] Parking position (ParkPos) in the method of elimination

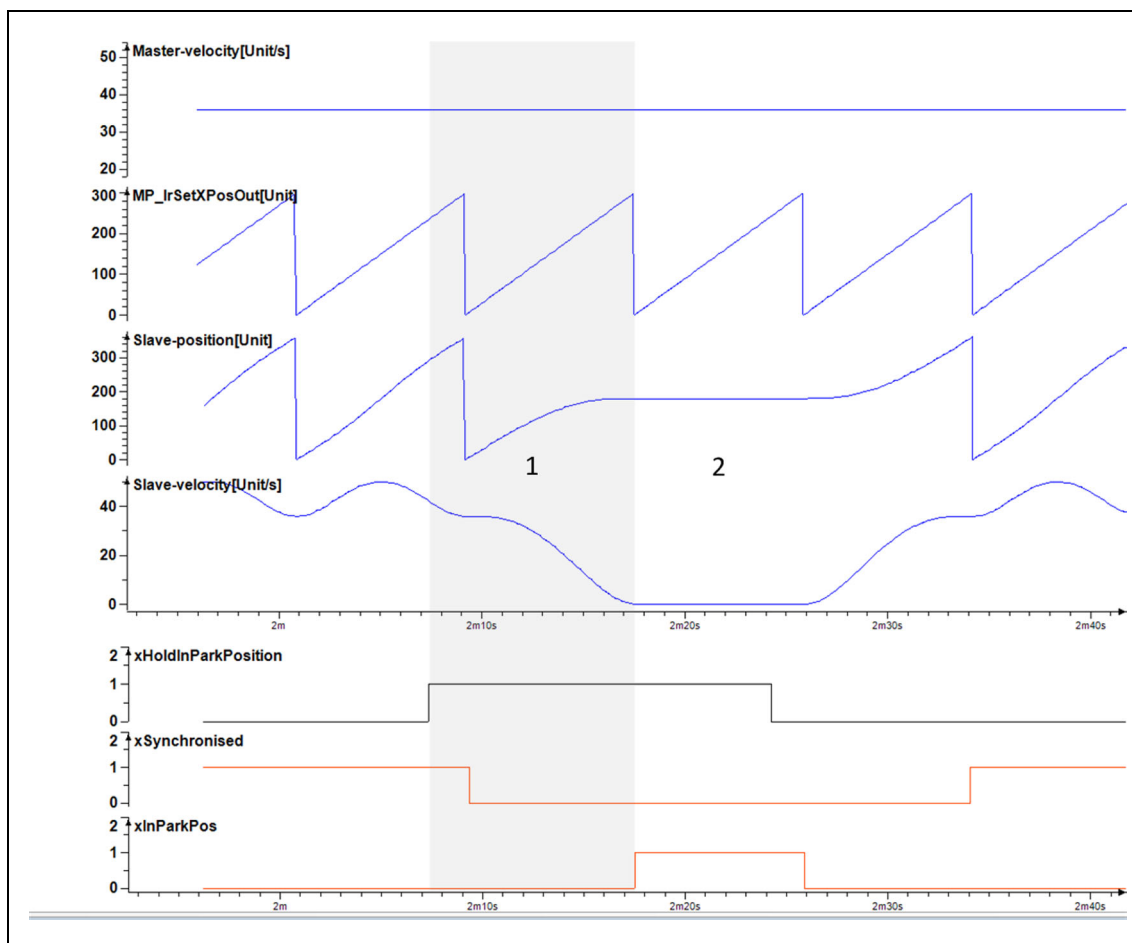
Inputs relevant for the function

- *xHoldInParkPos*

Outputs relevant for the function

- *xInParkPos*

Example



[9-18] Continuous driving operation

No	xHoldInParkPosition	xSynchronised	xInParkPos	INFO
1	FALSE → TRUE	TRUE → FALSE	FALSE	The cross cutter is moved to the parking position after it leaves the synchronous area.
2	TRUE	FALSE	FALSE → TRUE	The cross cutter is kept in the parking position until <i>xHoldInParkPosition</i> is once again set to FALSE.

9.11 Increasing/decreasing the speed (overspeed)

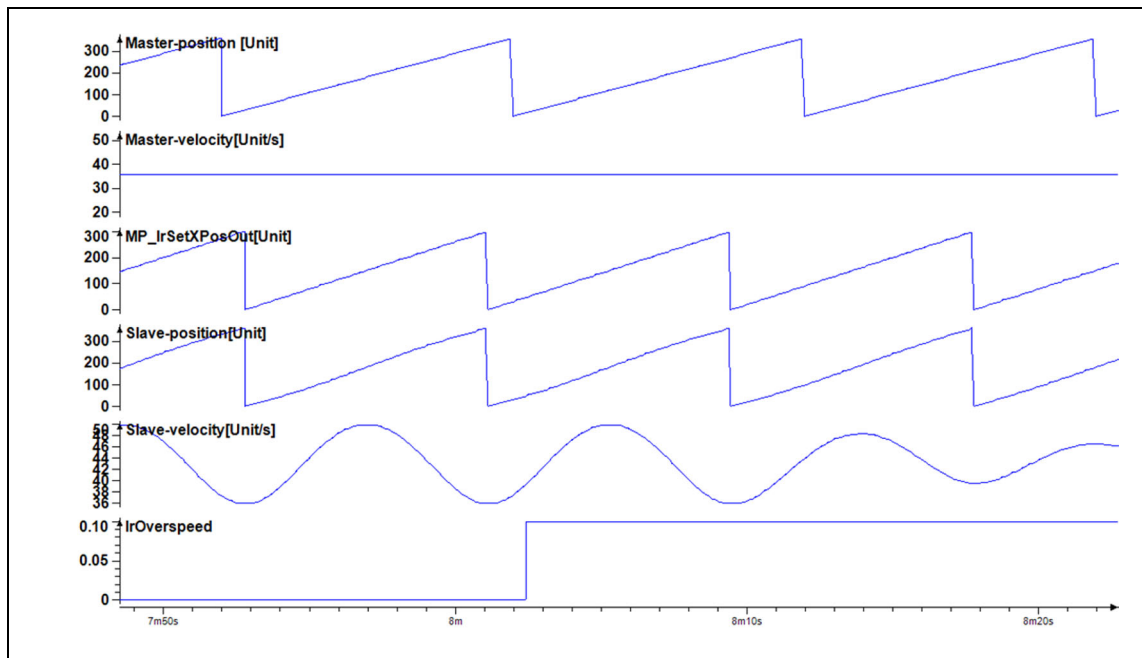
In certain applications, the motion in the cutting phase must be undersynchronous or oversynchronous with the material speed by an adjustable factor. Increasing or decreasing the speed is made possible by the *IrOverspeed* input. A scaled value with reference to the current material speed is expected here.

If, for example, the synchronous speed of the sealing roller is to be decreased by 20%, the *IrOverspeed* input must be set to -0.2. Setting it to 0.5 results in an increase in the synchronous speed by 50%. The reference value of 100% is the current material speed. The valid value range of the *IrOverspeed* input is -0.9 to 2.0. Any change to the value is accepted after the synchronous phase and is effective for the next step.

Inputs relevant for the function

- *IrOverspeed*

Example



[9-19] Oversynchronous cutting process

9.12 Calculating the cutting length

The technology module calculates the cut length of material after every cut (`xCutDone` output = TRUE). The cut length results from the difference between the current position of the master axis and the current cutting position of the sealing roller. The current cutting length is output at the `IrActCutLength` output.

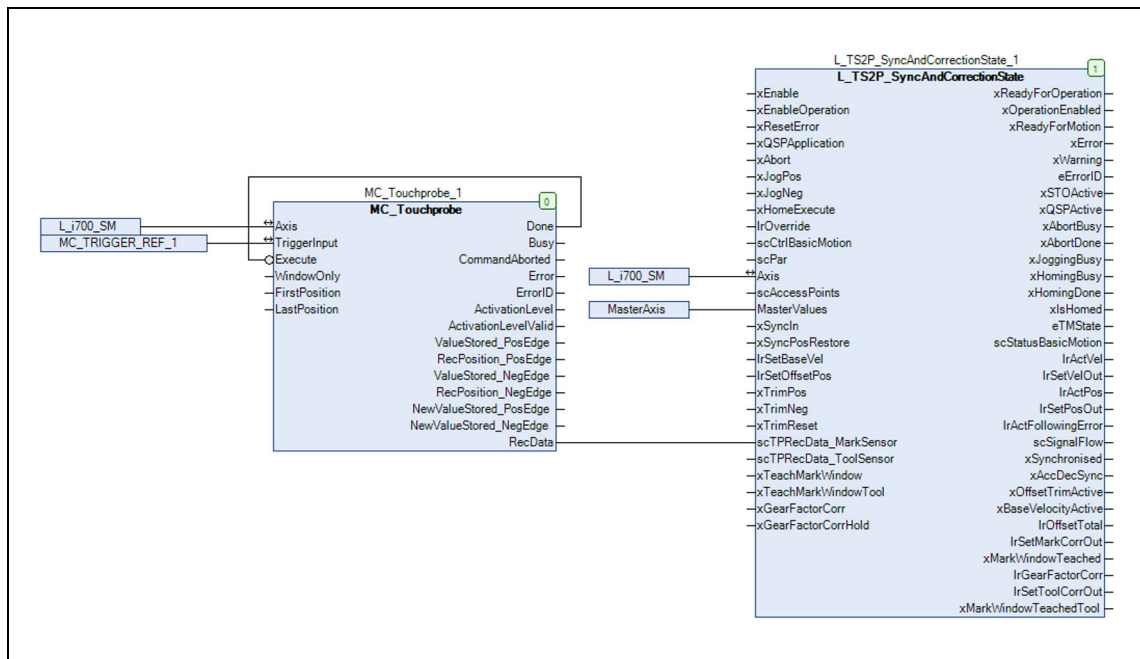
9.13 Touch probe connection

This chapter describes the connection of a touch probe to a V2 technology module by way of example.

In order to be able to use the touch probe functionality, one instance of the `MC_Touchprobe` function block must be used in the application. The user can choose which touch probe source is to be used for the mark sensor via the parameters of this function block.

The touch probe is connected to the technology module via the `scTpRecData_MarkSensor` input structure.

This input is connected to the `RecData` output of the "MC Touchprobe" module, as can be seen in the following figure.



[9-20] Interconnection example for touch probe connection

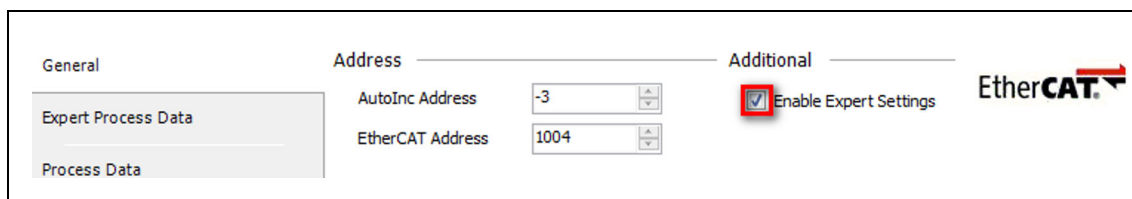
Example

A digital slave axis input was used for this 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.

Note!

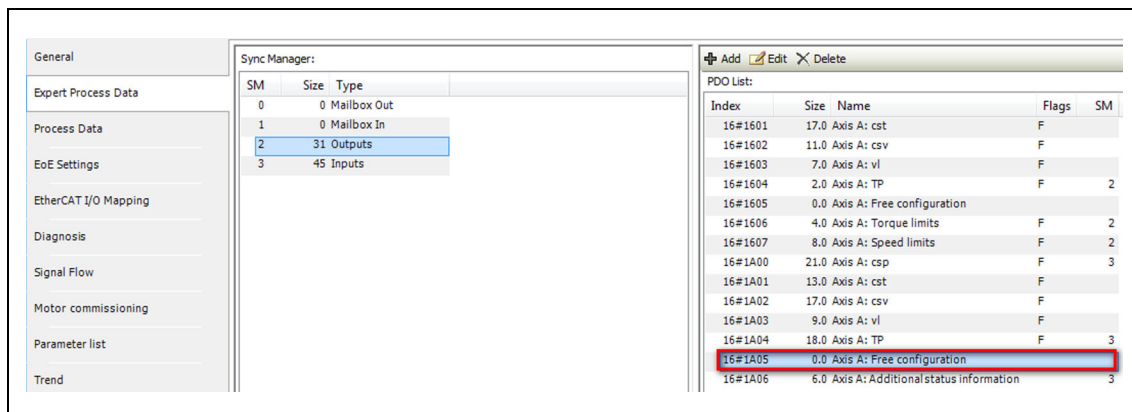
The touch probe evaluation of this technology module functions via a time stamp. If the technology module is used on a controller, the EtherCAT time stamp of the digital input used must be added to the EtherCAT mapping by the user. This step is not necessary for the i950 PLC device. Here, all necessary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

1. Activating expert settings



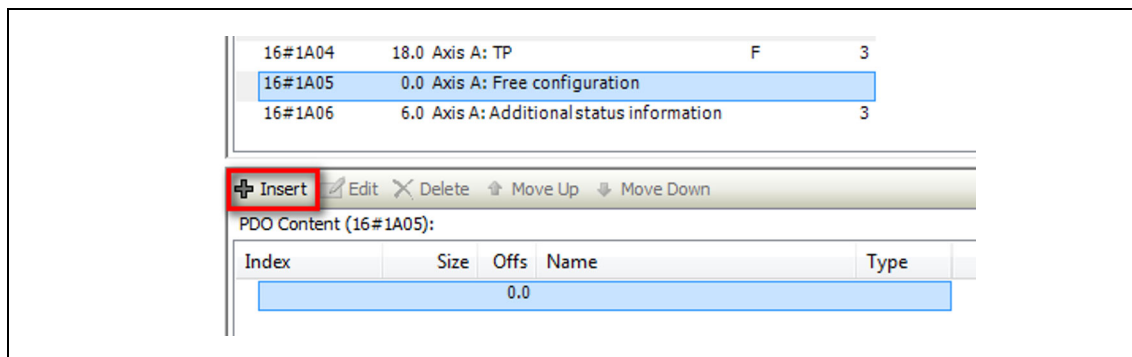
[9-21] Interconnection example for touch probe connection

2. Selecting the mapping to be extended

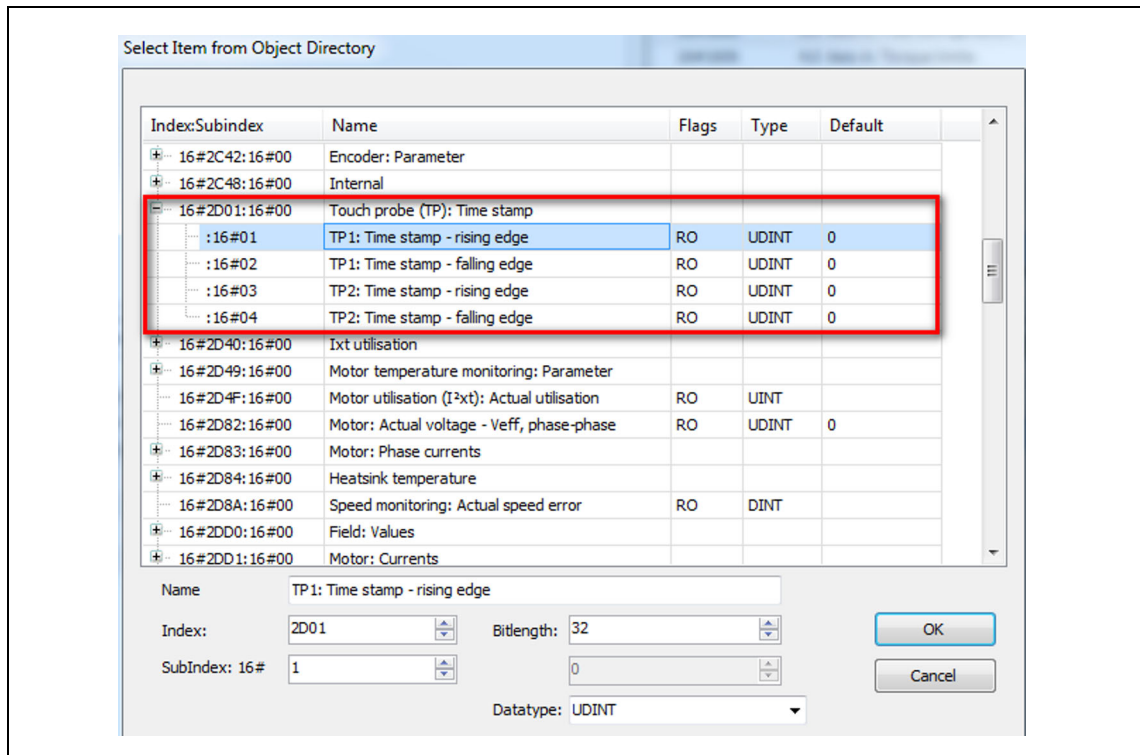


[9-22] Select the mapping to be processed

3. Add time stamp objects to the free mapping

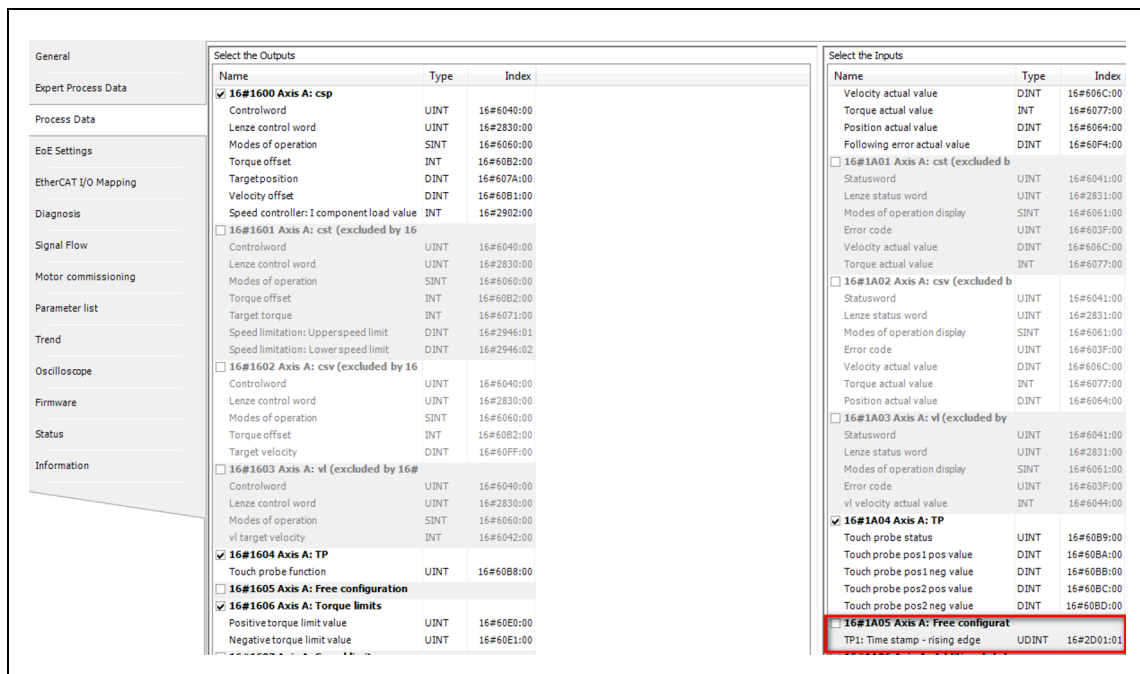


[9-23] Add objects 1



[9-24] Add objects 2

4. Add free mapping objects to the EtherCAT mapping



[9-25] Add free mapping to mapped objects

9.14 "Length cutting with mark correction" operating mode

In the "Length cutting with mark correction" operating mode the integrated register controller controls the position relative to a mark detected on the material (touch probe) as the higher-level control loop.

To this end, a register axis is created in the technology module that contains the corrected position. This register position is used as the master value for the cutting formats. The "Length cutting with mark correction" operating mode is selected by setting the *eCuttingMode* parameter to 2.

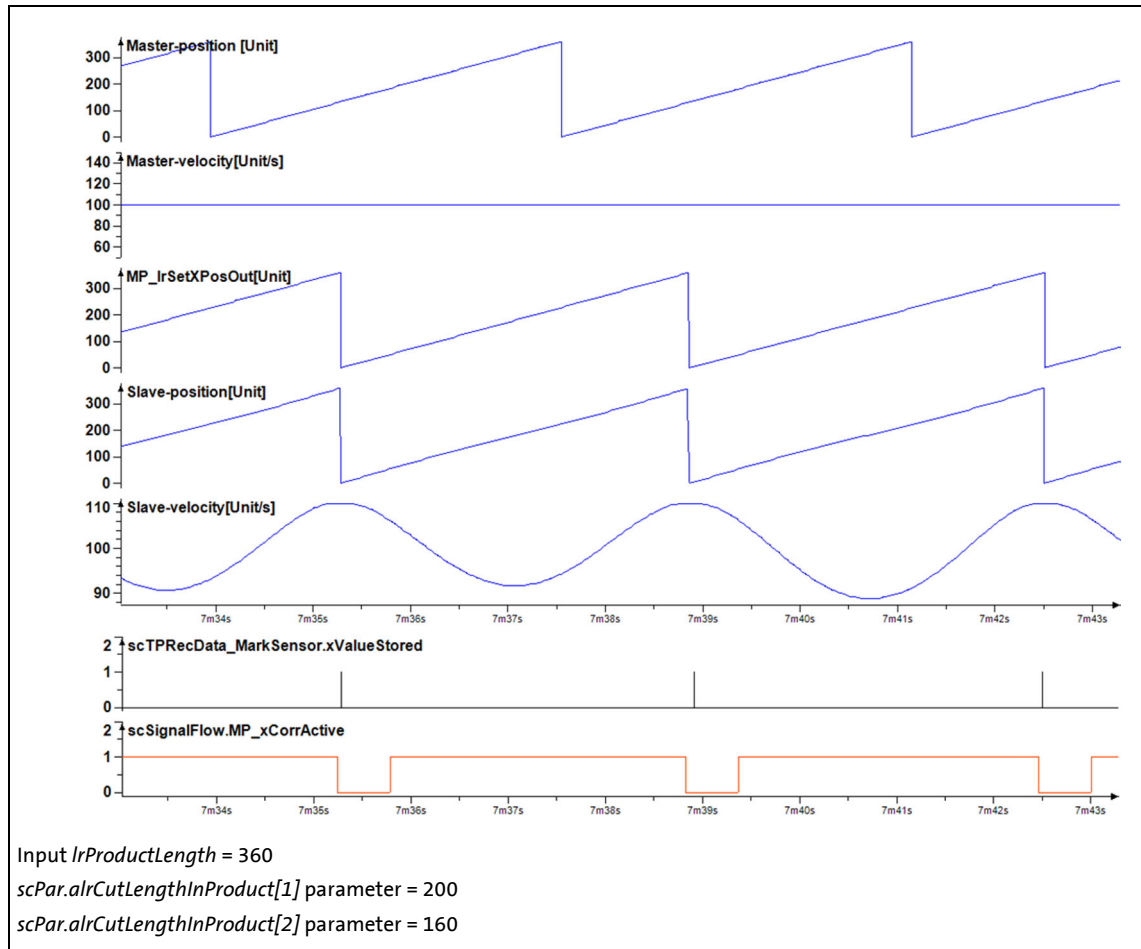
Before the cutting process ...

- a touch probe source must be selected
- a mark position (touch probe) homing must be executed for register control

Parameters relevant for the function

- *scPar.xMasterAbsolute*, initial value: *True*
- *lrStartXPosRelative*, initial value: 0 unit
- *eCuttingMode*, initial value: 0
- *lrSensorToolDist*, initial value: 0 unit of MasterAxis
- *lrToolCircumference*, initial value: 75.55555 unit of MasterAxis
- *lrCuttingAngle*, initial value: 10°
- *byNumOfTools*, initial value: 1
- *byStartToolNo*, initial value: 0
- *xMarkCorrection*, initial value: *TRUE*
- *lrMarkWindowSize*, initial value: 90 unit of MasterAxis
- *lrMaxCorrPos*, initial value: 30 unit of MasterAxis
- *lrMaxCorrNeg*, initial value: -30 unit of MasterAxis

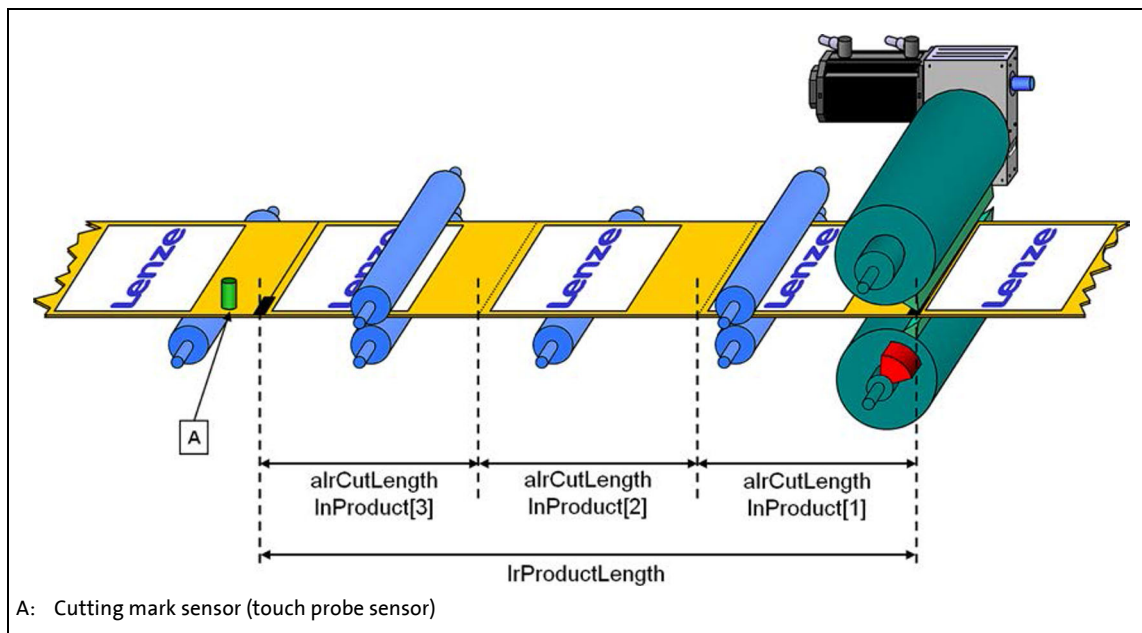
Example



[9-26] Cutting process with mark correction via register control

9.15

Partial cutting lengths in product length (print/cut format)



[9-27] Cutting lengths in product length (print/cut format)

The product length is defined by the *lrProductLength* input. In this way you also define the cycle length of the register. This value is always constant for the cutting process and cannot be changed. Cuts are always expected at the interval of the product length.

The *scPar.alrCutLengthInProduct[1...10]* parameter (array) serves to define up to 10 partial cuts with different cutting lengths within in the product length (or the register cycle). The first cut *alrCutLengthInProduct[1]* is made at the beginning of the product, the second cut *alrCutLengthInProduct[2]* subsequently, etc. The sum of all cutting lengths in the array must correspond to the product length (the register cycle).

All partial cutting lengths defined with the value "0" are deactivated. In the event that all partial cutting lengths are defined with the value "0", the cutting length is equal to the product length (*lrProductLength* input).

Inputs relevant for the function

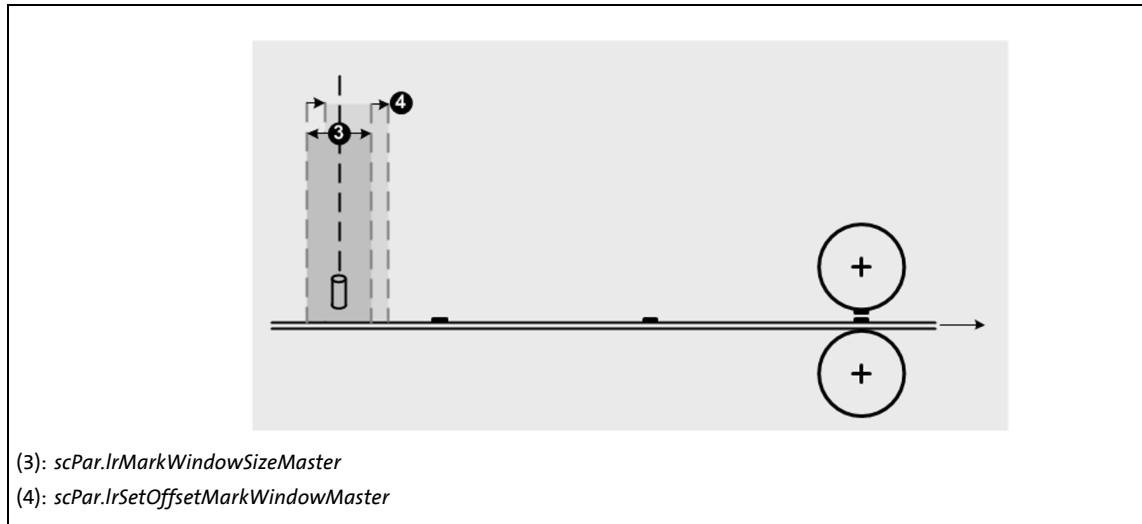
- *lrProductLength*

Parameters relevant for the function

- *scPar.eCuttingMode*, initial value: 0
- *scPar.lrMarkWindowSize*, initial value: 90 unit

9.16 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.



[9-28] Continuous driving operation

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).

If the mark window is taught, the *scPar.IrMarkWindowSizeMaster* 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.

If the mark window has to be shifted this can be done with the help of the *scPar.IrSetOffsetMarkWindowMaster* parameter.

Inputs relevant for the function

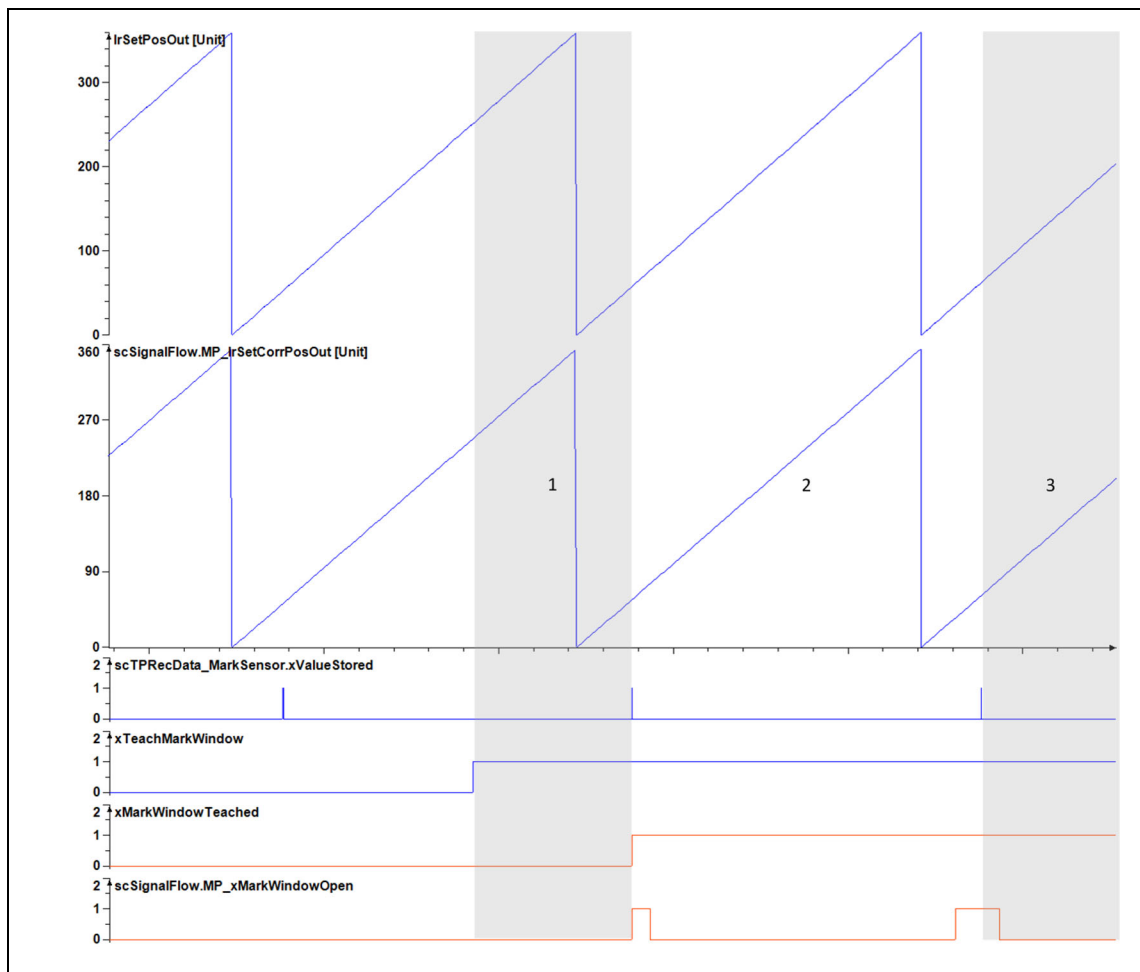
- *xTeachMarkWindow*

Outputs relevant for the function

- *xMarkWindowTeached*

Parameters relevant for the function

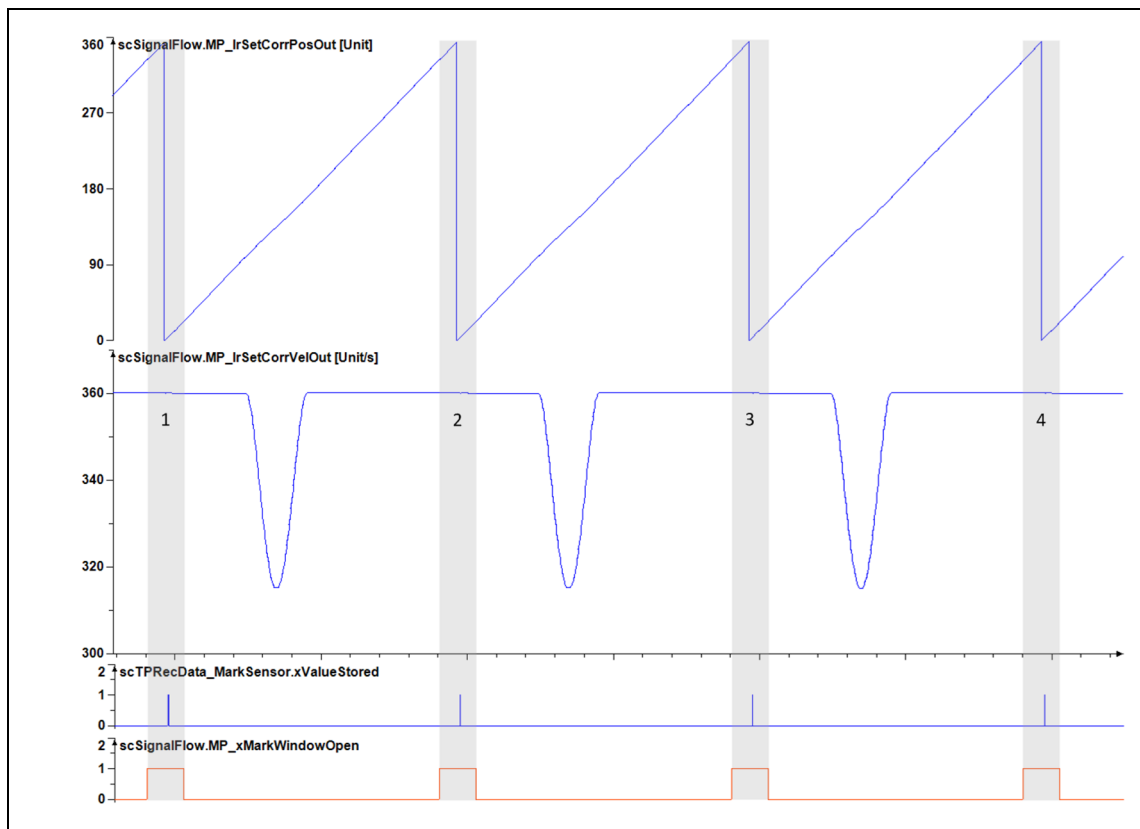
- *scPar.xEnableMarkWindowMaster*
- *scPar.IrMarkWindowSizeMaster*, initial value: 90 unit
- *scPar.IrSetOffsetMarkWindowMaster*, initial value: 0 unit

Example: 1

[9-29] Signal characteristic when the teaching function is executed (xTeachMarkWindow = TRUE)

No	scTPRecData_MarkSensor.xValueStored	xTeachMarkWindow	xMarkWindowTeached	scSignal-Flow.MP_xMarkWindowOpen	INFO
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.

Example 2



[9-30] Signal characteristic for a configured mark window

No	scTPRecData_MarkSensor.xValueStored	scSignal-Flow.MP_xMarkWindowOpen	INFO
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.

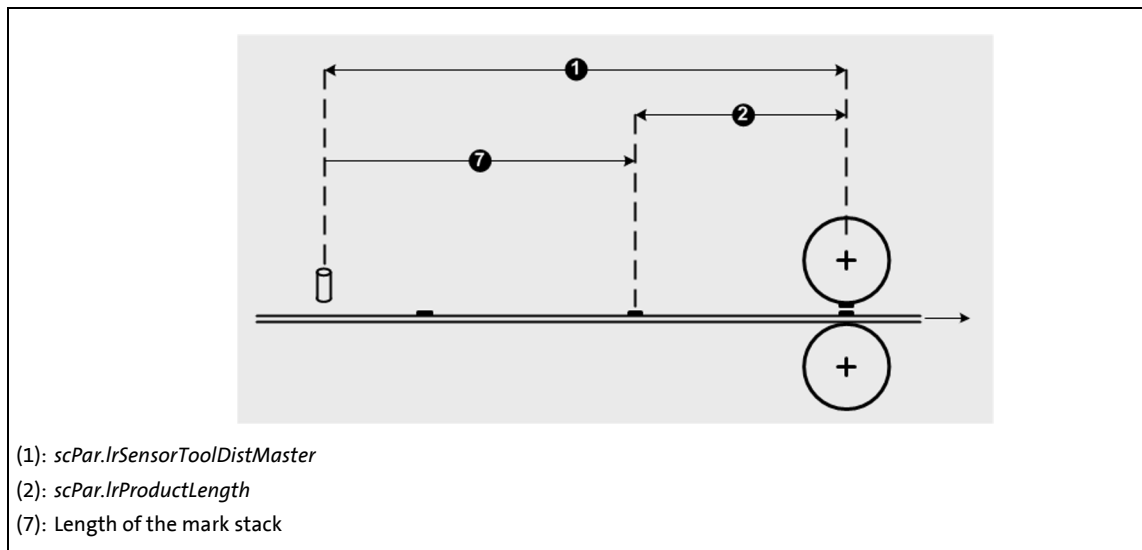
9.17 Mark stack master value correction

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, print heads). $scPar.IrSensorToolDistMaster > scPar.IrProductLength$

The function ensures that the detected mark is only evaluated if 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 = $scPar.IrSensorToolDistanceMaster - scPar.IrProductLength$

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 and cause cutting inaccuracies.

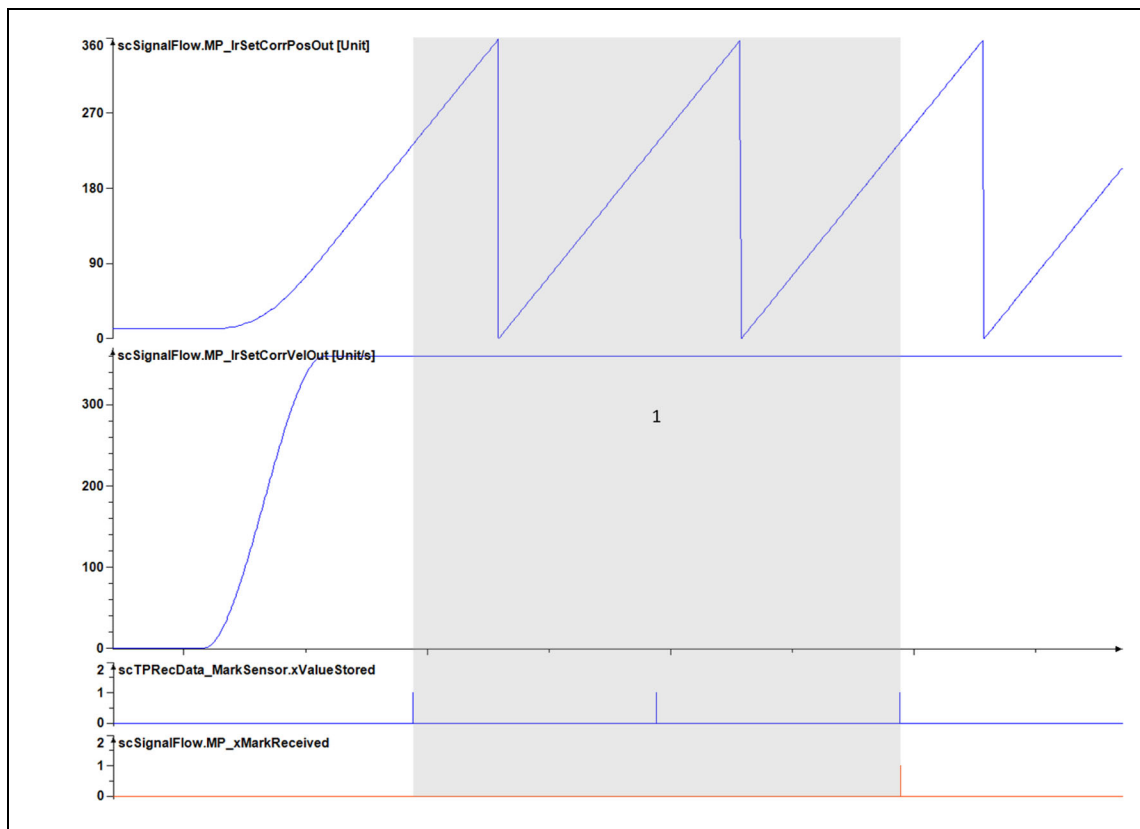


[9-31] Systematic representation of the mark register

Parameters relevant for the function

- $scPar.IrSensorToolDistMaster$
- $scPar.IrProductLength$

Example



[9-32] Signal characteristic when the mark stack is used

No	scTPRecData_MarkSensor.xValueStored	scSignalFlow.MP_xMarkReceived	INFO
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 left the mark stack. The length of the mark stack can be read off by the distance covered between scSignalFlow.MP_IrSetCorrPosOut[1] and scSignalFlow.MP_IrSetCorrPosOut[2].

9.18 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.

Caution: If the selected gain is too high, the setpoint speed may build up.

When the gearbox factor correction is activated, the current value of the corrected gearbox factor is output via the *scSignalFlow.IrAverageGearFactorCorr* 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

- *xGearFactorCorr*
- *xGearFactorCorrHold*

Outputs relevant for the function

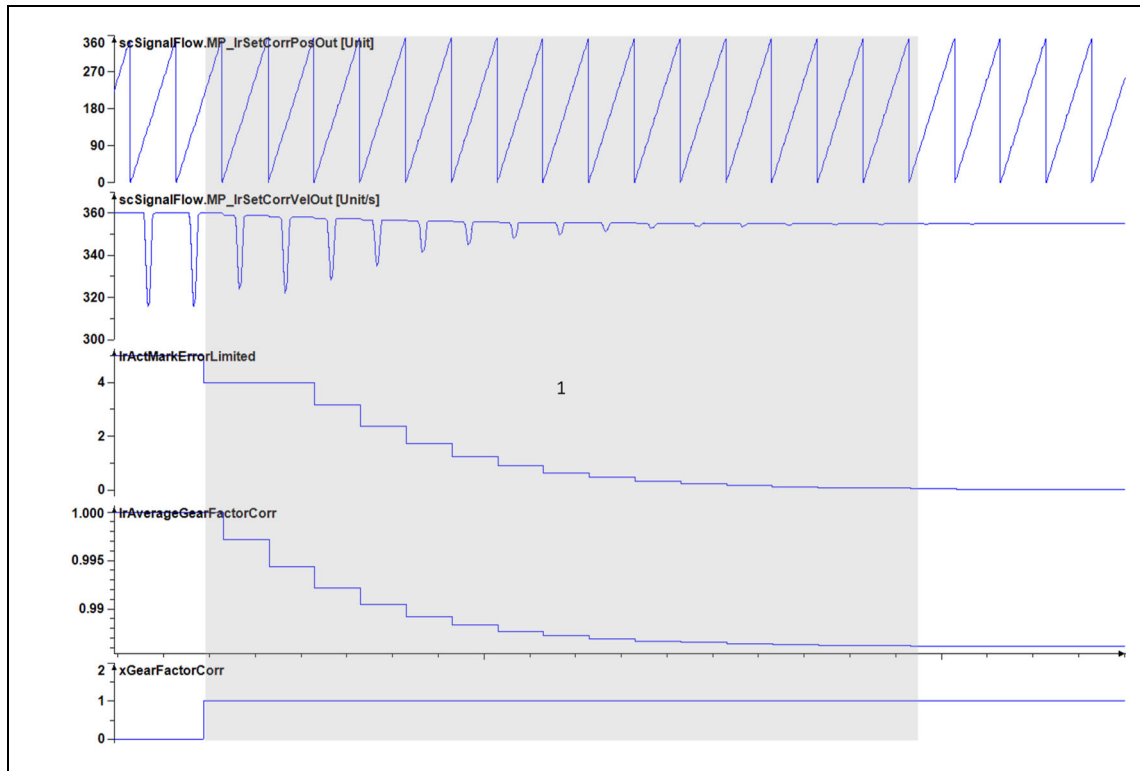
- *IrGearFactorCorr*

Parameters relevant for the function

- *scPar.IrGearFactorCorrGainMaster*, initial value: 0.1
- *IrMaxGearFactorCorrMaster*, initial value: 10

Signal flow structures relevant for the function

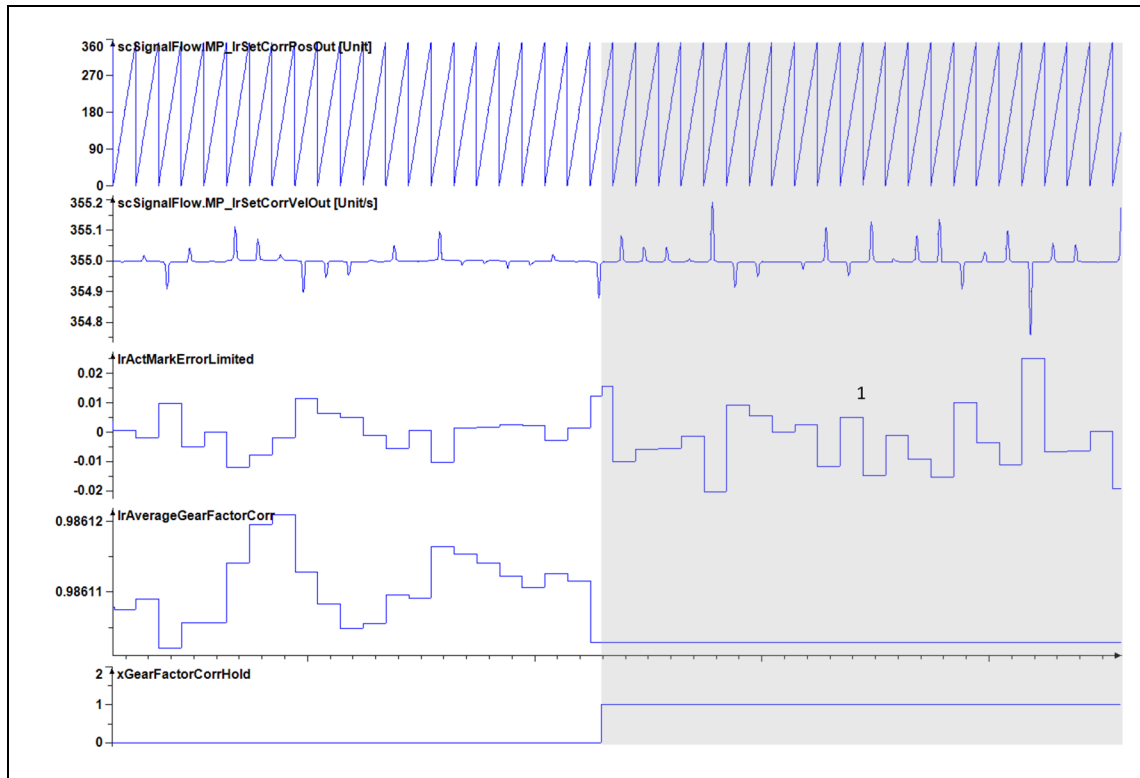
- *scSignalflow.MP_xGearFactorCorrLimited*

Example: 1

[9-33] "Activate gearbox factor correction" signal characteristic

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

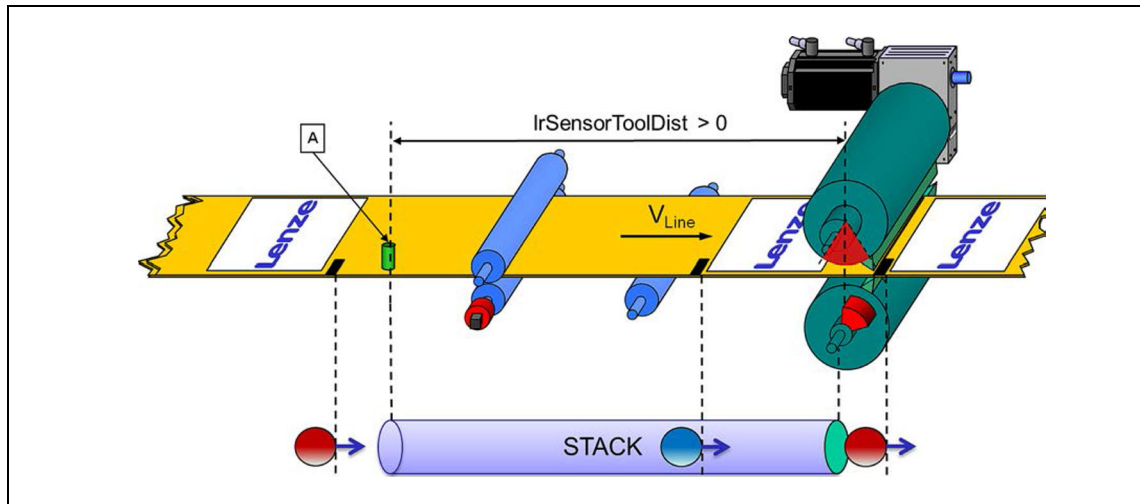
Example 2



[9-34] "Maintain optimized gearbox factor" signal characteristic

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

9.19 "Cutting on a mark (event-controlled)" operating mode



[9-35] "Cutting on a mark (event-controlled)" operating mode

A: Cutting mark sensor (touch probe sensor)

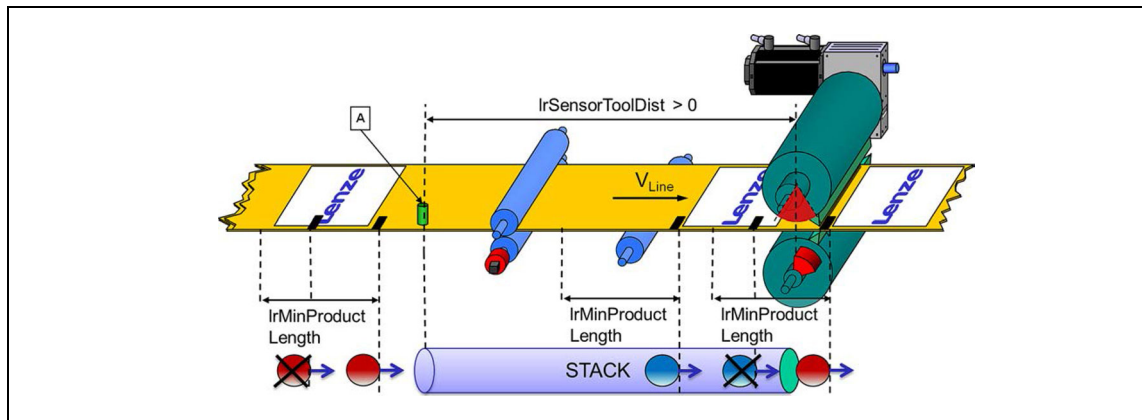
The "Cutting on a mark (event-controlled)" operating mode is selected by setting the *scPar.eCuttingMode* parameter to 3. It is only possible to use this operating mode if the cutting mark sensor is installed in front of the cross cutter. The distance between cross cutter and the cutting mark sensor is set by setting the *scPar.IrSensorToolDistance* to > 0 .

The profiles are calculated in real time in this operating mode, allowing the synchronous phase to be changed during the cutting process. All changes are accepted after the synchronous phase and are effective for the next cut. The synchronous speed of the sealing roller is calculated depending on the circumference of the roller in the *scPar.IrToolCircumference* parameter, the speed of the master axis and the cycle length of the sealing roller.

If the *SyncIn* input is set to TRUE, the starting curve is calculated from the current position in the synchronous phase. The cutting process starts as soon as a valid mark (detected touch probe with the current position) is present. The cross cutter makes the cut at the detected marks. If no other valid marks are detected, the cross cutter is moved to the parking position. The cross cutter remains at standstill until another valid mark is detected. A valid mark serves to re-initiate the cutting process.

All valid marks are stored in a register. The register can keep a maximum of 100 marks. After each cut, the next mark from the register is processed for the next step.

Several marks can be stored and processed consecutively during the cutting process. In order to hide marks, a minimum distance between two marks must be set using the *scPar.IrMinProductLength* parameter. After a valid mark, the following marks within the minimum distance are hidden by the technology module.



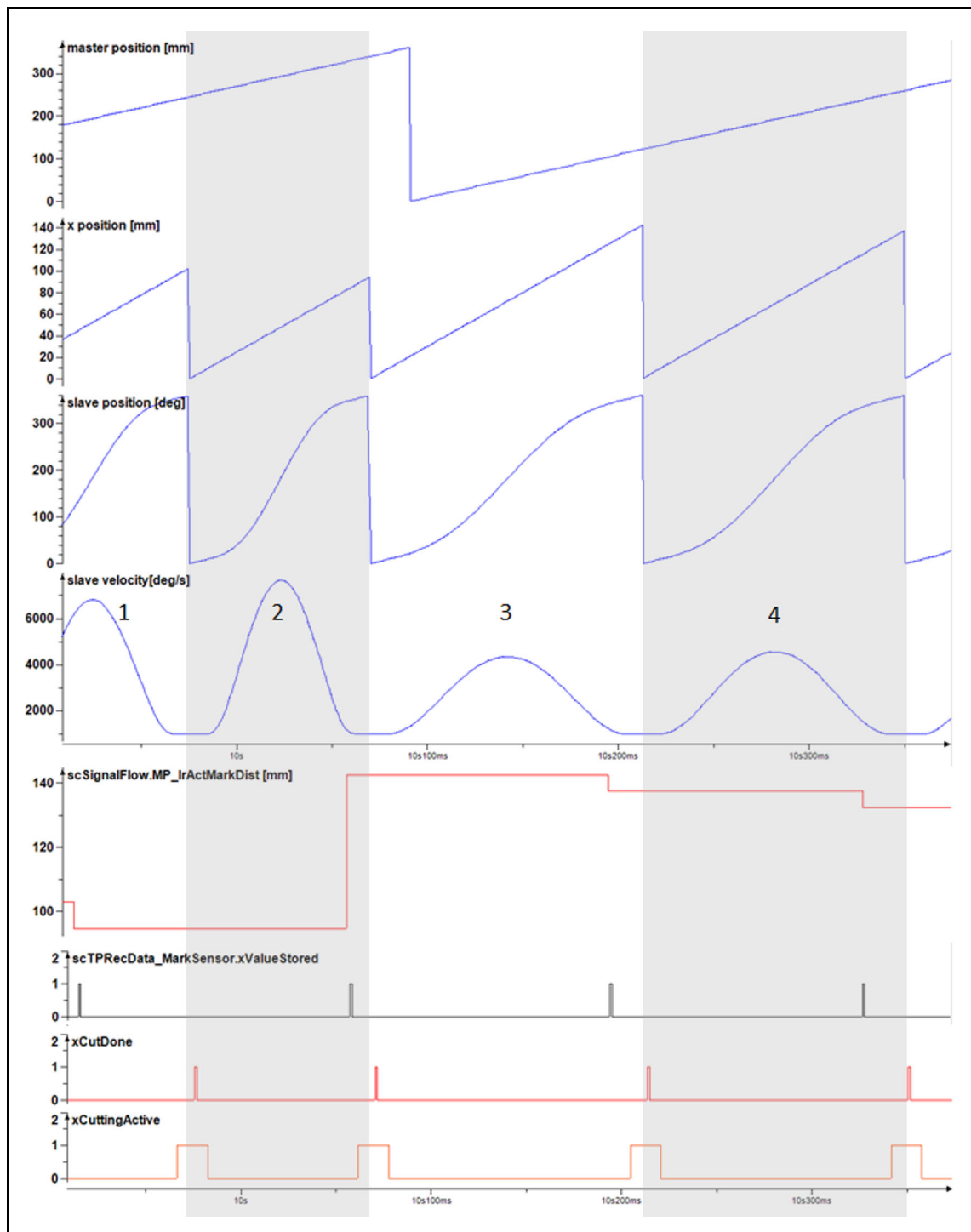
[9-36] Cutting on a mark with the minimum product length (parameter `IrMinProductLength`)

A: Cutting mark sensor (touch probe sensor)

Parameters relevant for the function

- `scPar.xMasterAbsolute`, initial value: TRUE
- `scPar.IrStartXPosRelative`, initial value: 0 unit
- `scPar.eCuttingMode`, initial value: 0
- `scPar.IrSensorToolDist`, initial value: 0 unit
- `scPar.IrMinProductLength`, initial value: 50 unit
- `scPar.IrToolCircumference`, initial value: 75 55555 unit
- `scPar.IrCuttingAngle`, initial value: 10 unit
- `scPar.byNumOfTools`, initial value: 1 unit
- `scPar.byStartToolNo`, initial value: 0 unit

Example



[9-37] Event-controlled cutting process on a detected mark

No	scTPRecData_MarkSensor.xValueStored	INFO
1	Rising edge at the start of area 1	The TM is executed in the <i>scPar.eCuttingMode</i> = CutOnTP mode. Distance between the TP sensor and the cutting position: <i>scPar.IrSensorToolDist</i> = 160 The marks on the material are sporadic and have no fixed distance.
2	Rising edge at the start of area 2	The TM cuts in area 2; Cutting length = Mark distance to the mark in area 1

No	scTPRecData_MarkSensor.xValueStored	INFO
3	Rising edge at the start of area 3	The TM cuts in area 3; Cutting length = Mark distance to the mark in area 2
4	Rising edge at the start of area 4	The TM cuts in area 4; Cutting length = Mark distance to the mark in area 3

9.20 Torque feedforward control

The torque feedforward control is activated if the *scPar.lTrqCtrlGain* parameter > 0 and the *xTrqCtrlActive* output is set to TRUE. The *scPar.lTrqCtrlGain* parameter is the multiplicative gain factor of the feedforward control value.

The feedforward control value is calculated in the axis by multiplying the inertia with the acceleration / deceleration of the sealing roller.

The inertia of the sealing roller is defined via the *scPar.lMInertia* parameter in kg/cm².

The feedforward control value is provided via the *lrSetMotorPreCtrlTrqOut* output.

The acceleration / deceleration of the sealing roller is calculated from the cam information. The temporal difference with which the acceleration / deceleration is calculated is specified in seconds with the *scPar.lTrqCtrlRateTime* parameter.

If the *scPar.lTrqCtrlRateTime* parameter is set to 0, the current setpoint values of the acceleration / deceleration of the sealing roller are used.

The real time calculation of the feedforward control value allows flexible changes to the parameters for the torque feedforward control during the cutting process.

All changes become effective immediately. This means that the feedforward control values can be adjusted and optimised during the cutting process.

Outputs relevant for the function

- *lrSetMotorPreCtrlTrqOut*

Parameters relevant for the function

- *scPar.lTrqCtrlGain*, initial value: 0
- *scPar.lTrqCtrlRateTime*, initial value: 0
- *scPar.lMInertia*, initial value: 0 [kg/cm²]

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

In order to adjust the calculated correction value, the tool correction profile generation has to be activated (*scPar.xCorrToolError* = TRUE).

The positioning error is corrected within the parameterised correction window.

9.21.1 Correction window and tool correction limitation

Inputs relevant for the function

- *scTPRecData_ToolSensor*

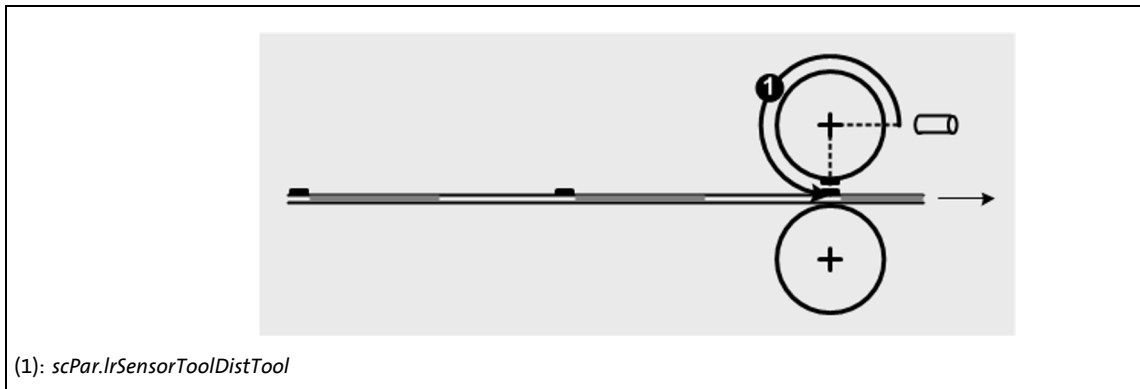
Outputs relevant for the function

- *lrSetToolCorrOut*

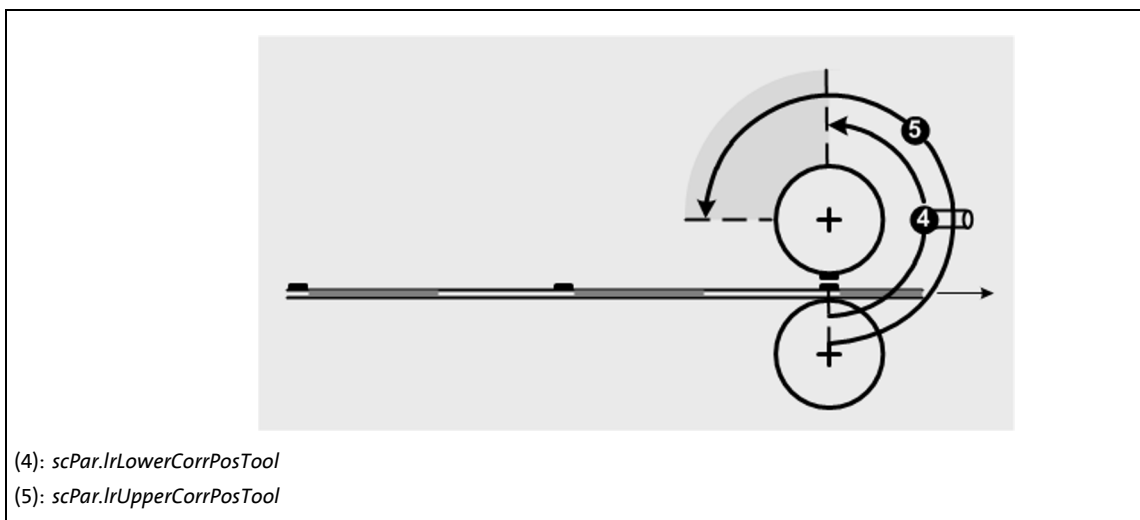
Parameters relevant for the function

- *scPar.xCalcToolError*, initial value: FALSE
- *xCorrToolError*, initial value: FALSE
- *lrSensorToolDistTool*, initial value: unit
- *scPar.lrLowerCorrPosTool*, initial value: 90 unit
- *scPar.lrUpperCorrPosTool*, initial value: 180 unit
- *scPar.lrMaxCorrPosToo*, initial value: 30 unit
- *scPar.lrMaxCorrNegTool*, initial value: -30 unit

Measuring points relevant for the function



[9-38] Tool correction sensor parameterization



[9-39] Tool correction window

9.21.2 Tool correction mark window

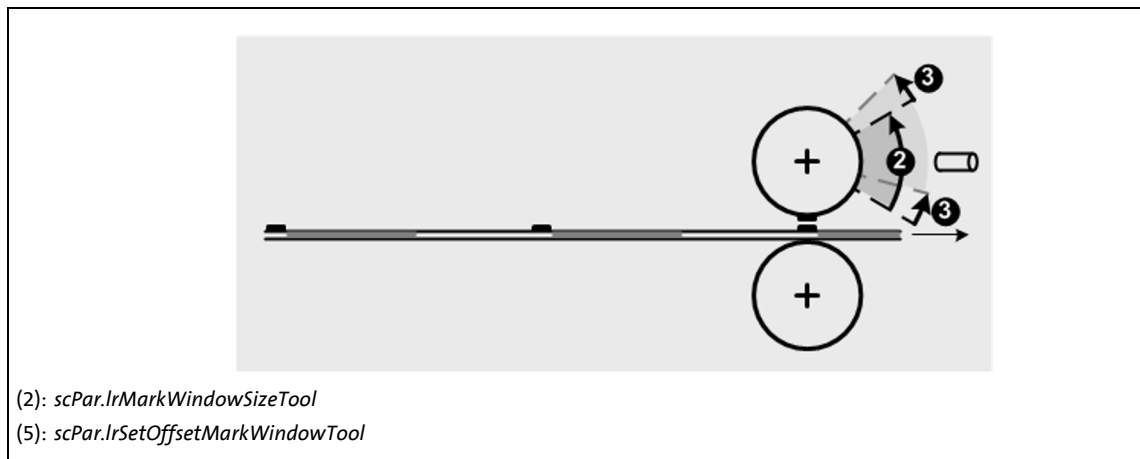
Tool correction also offers the functionality of a mark window. To activate this, the *scPar.xEnableMarkWindowTool* parameter must be set to TRUE.

Each touch probe is evaluated when the mark window is deactivated.

The first mark has to be taught so the mark window can work properly. To this end, 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 the *scPar.lrSetOffsetMarkWindowTool* parameter.



[9-40] Tool correction window

Mark failure detection

If the *scSF.MP_xMarkNotDetectedTool* signal flow variable is set, no mark has been detected within the parameterised mark window. In this case, no position error is calculated so that downstream functions can still be executed.

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

Inputs relevant for the function

- *xTeachMarkWindowTool*

Outputs relevant for the function

- *xMarkWindowTeachedTool*

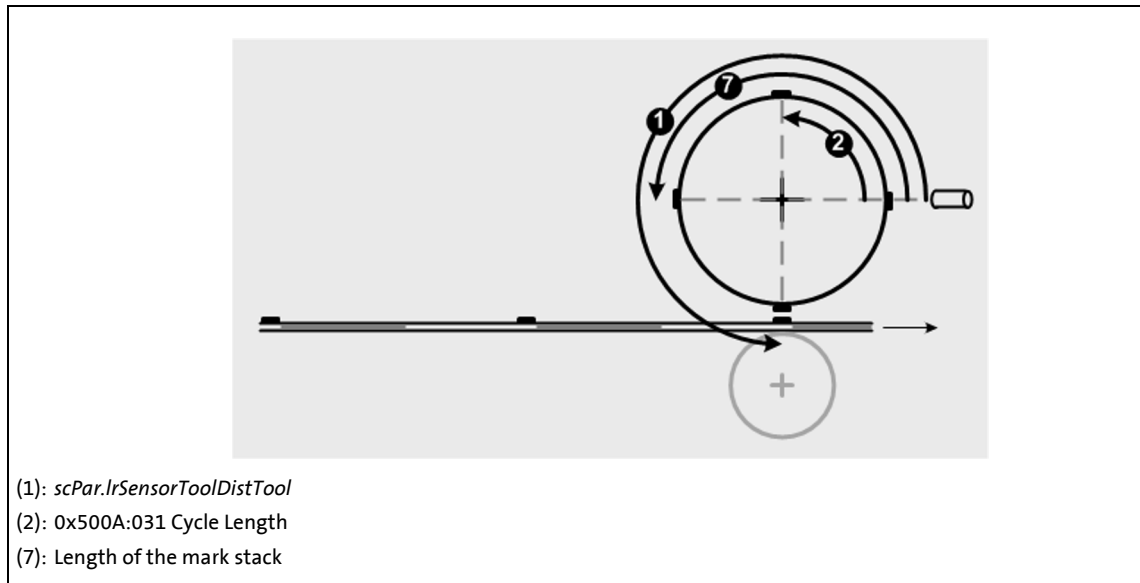
Parameters relevant for the function

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

9.21.3 Tool correction mark stack

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 figure shows the use of a mark stack. Here, the distance from the mark sensor to the tool is greater than the set mark distance.



[9-41] Systematic representation of the mark stack for a tool with feed constant = 4 cycle lengths

If possible, the mark sensor should be mounted as close as possible to the axis. The farther away it is mounted from the axis, the more changes in the material flow remain undetected and cause 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.IrSensorToolDistTool* parameter. The mark stack function is automatically activated when the *scPar.IrSensorToolDistTool* parameter > 0x500A:031 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

- *scPar.IrSensorToolDistTool*, initial value 0

Symbole

"Cross Cutter (V2)" TM parameters [22](#)
"Cross Cutter State (V2)" functional description [11](#)
"Cross Cutter State (V2)" technology module signal flow [33](#)
"Cross Cutter State" technology module signal flow [31](#)

A

Aborting a positioning process [53](#)
Absolute positioning [45](#)
Application notes [8](#)

C

Competing a positioning process [53](#)
Continuous travel with a defined velocity [50](#)
Conventions used [7](#)
Cross cutter partial signal flow [32](#)
Cross Cutter State (functional description) [11](#)
Cross cutter state signal flow plan [31](#)

D

Document history [6](#)

E

E-mail to Lenze [106](#)
eTMState [27](#), [29](#)

F

Feedback to Lenze [106](#)

H

Halt, [44](#)
Homing [40](#)

I

inputs [18](#)

L

L_TB2P_States [27](#), [29](#)
L_TF2P_CrossCutterState function block [17](#)
L_TS2P_CrossCutterState [17](#)
L_TS2P_scPar_CrossCutterState [22](#)
L_TS2P_scPar_CrossCutterState parameter structure [22](#)
L_TS2P_scSF_CrossCutterState [33](#)
Layout of the safety instructions [8](#)

M

Main signal flow [31](#)
Manual jog (jogging) [38](#)
Master value correction/register control partial signal flow [31](#)

N

Notes on how to operate the technology module [15](#)

O

Operating mode [15](#)
outputs [20](#)
Override for speed [55](#)

Q

QSP application [43](#)

R

Relative positioning [47](#)

S

Safety instructions [9](#)
Speed override [55](#)
Start of the axes [15](#)
State machine [27](#)
States [27](#)
States of the eTMState output [27](#), [29](#)
Structure of the "Cross Cutter State (V2)" access points [36](#)
Structure of the L_TS2P_scSF_CrossCutterState signal flow [33](#)

T

Target group [5](#)
Technology module functions (overview) [13](#)
Tool correction partial signal flow [32](#)
Torque feedforward control partial signal flow [31](#)

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

