

Technology module



Flying Saw -----

Reference Manual

EN

Contents

1	About this documentation	4
1.1	Document history	6
1.2	Conventions used	6
1.3	Design of the safety instructions	7
2	Safety instructions	8
3	Overview of the functions of the technology module	10
3.1	Functional description	10
3.2	TM Functional Overview	12
3.3	Important notes on how to operate the technology module	13
4	Application areas	14
5	Function block L_TS2P_FlyingSawState	15
5.1	Inputs	16
5.2	Outputs	19
5.3	Parameters	21
6	State Machine	24
6.1	State Machine of the technology module	24
7	Signal flow diagram	26
7.1	Structure of the signal flow	27
7.2	Structure of the access points	28
8	Description of the basic functions	29
8.1	Manual jog (jogging)	30
8.2	Homing	32
8.3	QSP application	35
8.4	Halt	36
8.5	Absolute positioning	37
8.6	Relative positioning	39
8.7	Continuous travel with a defined velocity	42
8.8	Aborting and completing a positioning process	45
8.9	Override for speed	47
9	Description of the technology functions	49
9.1	Set motion profiles (linear/S ramps)	49
9.2	Optimized dynamics	50
9.3	First synchronization of the saw carriage to the material velocity	51
9.3.1	Parameter eFlyingSawMode = 0 (mark-controlled operation)	51
9.3.2	Parameter eFlyingSawMode = 1	53
9.3.3	Parameter eFlyingSawMode = 2	55
9.4	Mark-controlled operation (cutting on position marks)	57
9.5	Length-controlled operation (cutting relative product lengths)	61
9.6	Separation of cut materials (Inserting a gap)	68
9.7	Restore target position	70
9.7.1	Mark-controlled operation	70
9.7.2	Length-controlled operation	73
9.8	Delete mark stack	75
9.9	Move to parking position	77
9.10	Window for following error monitoring	79
9.11	Override in the synchronous phase	81

Contents

Index	84
Your opinion is important to us	85

1 About this documentation

This documentation ...

- contains detailed information about the functionalities of the technology module "Flying Saw (V2)";
- is classified in the "Controller-based Automation" manual collection. It consists of the following 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« (parameter setting, configuration, diagnostics) • »VisiWinNET® Smart« (visualisation) • »Backup & Restore« (data backup, recovery, update)

More technical documentation with regard to Lenze components

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

Planning / 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 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.

1.1

Document history

Version			Description
2.0	04/2021	TD06	Error corrections & supplements
1.0	04/2019	TD06	First edition

1.2

Conventions used

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

Type of information	Highlighting	Examples/notes
Numeric notation		
Decimal separator	Point	The decimal point is always used. For example: 1234.56
Hexadecimal number	0x	For hexadecimal numbers, the "0x" prefix is used. Example: 0x60F4
Text		
Program name	» «	»PLC Designer« ...
Variable names	<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; ...
Hyperlink	<u>underlined</u>	Optically highlighted reference to another topic. It is activated with a mouse-click in this online documentation.
Symbols		
Page reference	 6)	Optically highlighted reference to another page. It is activated with a mouse-click in this online documentation.
Step-by-step instructions		Step-by-step instructions are indicated by a pictograph.

Variable names

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 documentation

1.3 Design of the safety instructions

DANGER!

Indicates an exceptionally hazardous situation. If this information is not observed, serious irreversible injuries or death may result.

WARNING!

Indicates an exceptionally hazardous situation. Failure to observe this warning may result in serious irreversible or fatal injuries.

CAUTION!

Indicates a hazardous situation. Failure to observe this information may result in minor or moderate injury.

NOTE

Indicates property hazards. If this note is not observed, material damage may occur.

Application notes

Pictograph	Signal word	Meaning
	Tip!	Useful tip for easy operation
		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).
-

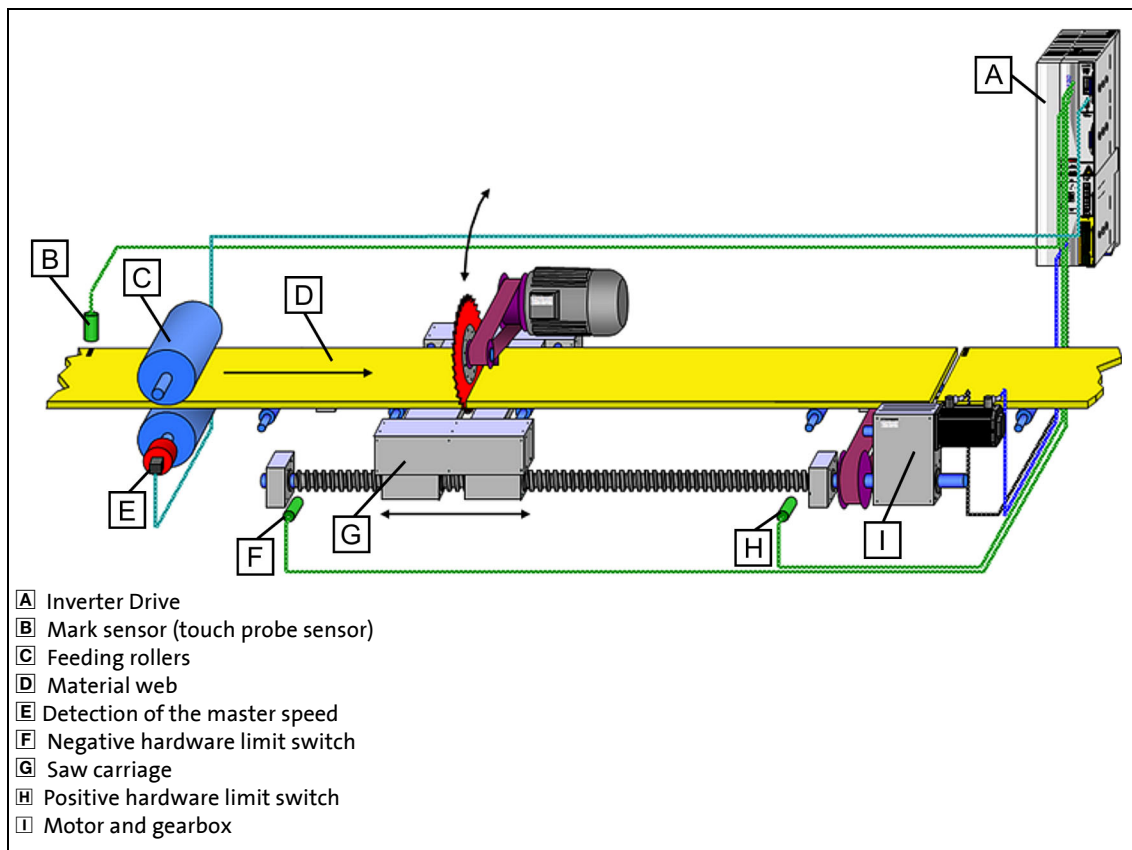
3

Overview of the functions of the technology module

3.1

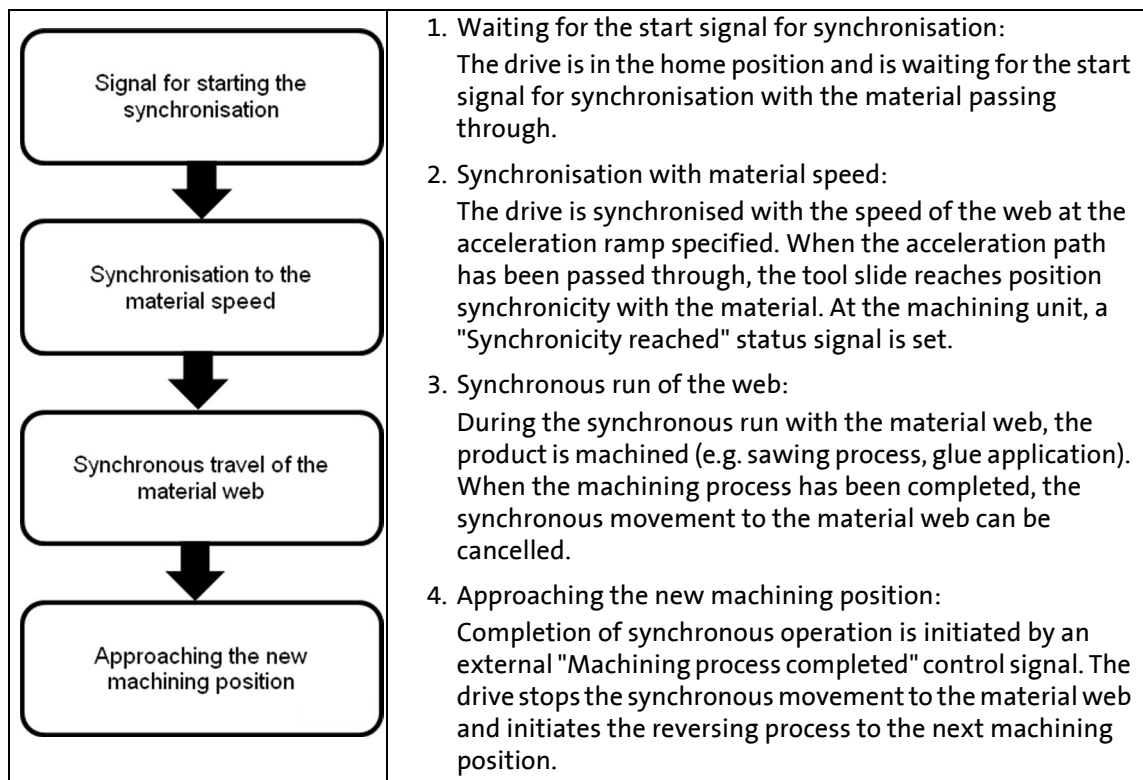
Functional description

The "Flying Saw" allows the application of a machining process (e.g. sawing or glue application) to a continuously moving material. For this purpose, the technology module enables length-controlled or mark-controlled operation with synchronization of the saw carriage to the material velocity.



[3-1] Typical mechanics of the technology module

A typical flying sawing process is divided into the following movement sections:



3.2 TM Functional Overview

In addition to the functions for operating the function block **L_MC1P_AxisBasicControlV2**, the technology module offers the following functionalities:

Functionality	Description
Basic Motion (V2):	
Manual jog (jogging) (□ 30)	Traverse axis in manual operation
Homing (□ 32)	Home the axis
OSP application (□ 35)	Stop motion via an application quick stop deceleration ramp (AxisRef)
Halt (□ 36)	Stop motion via a stop deceleration ramp
Absolute positioning (□ 37)	Move axis to an absolute target position
Relative positioning (□ 39)	Move axis to a target position relative to the current position
Continuous travel with a defined velocity (□ 42)	Execute continuous travel
Aborting and completing a positioning process (□ 45)	Stop positioning
Override for speed (□ 47)	Specify overrides for speed
Flying Saw (V2):	
Set motion profiles (linear/S ramps) (□ 49)	Set motion profiles for positioning or for jerk limited coupling operations
Optimized dynamics (□ 50)	Switch between fast and optimized profiles
First synchronization of the saw carriage to the material velocity (□ 51)	Synchronize the saw carriage to the material velocity
Mark-controlled operation (cutting on position marks) (□ 57)	Saw cuts are made to match the position marks of the material
Length-controlled operation (cutting relative product lengths) (□ 61)	Saw cuts have a desired relative distance (cut length) to each other
Separation of cut materials (Inserting a gap) (□ 68)	Suitable as cut edge protection for sensitive material or for separating the cut material.
Restore target position (□ 70)	One-time restoration of a target position after starting the sawing process
Delete mark stack (□ 75)	In mark-controlled mode, delete all stored marks in the mark stack
Move to parking position (□ 77)	Saw carriage is travelled to the parking position
Window for following error monitoring (□ 79)	Check whether the following error is within a defined window within a certain time
Override in the synchronous phase (□ 81)	In the cutting phase the material velocity is increased or decreased by an adjustable factor

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



»PLC Designer« online help

Here you can find detailed information on the function block **L_MC1P_AxisBasicControlV2**.

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 \(jogging\)](#) (□ 30)
 - [QSP application](#) (□ 35)
- 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](#) (□ 32)



Example of a movement type with level-activated function: Jogging

1. The input `xJogPos` = TRUE is set in the inhibited axis state.
 - Input `xEnableOperation` = FALSE (axis is disabled.)
==> state "STANDBY" (output `xOperationEnabled` = FALSE)
 - Output `xJogPos` = TRUE (Jogging is to be executed.)
2. Enable axis.
 - Input `xEnableOperation` = TRUE
==> state "JOGGING" (`xOperationEnabled` = TRUE)



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

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

Setting of the operating mode

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

4 Application areas

In mechatronics, the term "flying saw" stands for the synchronization of a tool to a continuously running material web. This means that machining no longer has to take place when the material is at a standstill. In this way, a convenient material transport, an optimal resource utilization of the machine and in many cases a higher production rate can be achieved.

In contrast to rotary processing tools, flying saw kinematics are translational and therefore limited dimensional systems. The flying saw technology is based on the principle of the electric shaft (synchronous travel of the tool slide with the material flow).

To move the saw carriage from one machining position to the next, the synchronous travel with the material is discontinued after the machining process and then the tool carriage synchronizes to the next machining position on the material web. The switching between the processing positions represents a time-controlled positioning process and can be adjusted by the user according to his machine requirements.

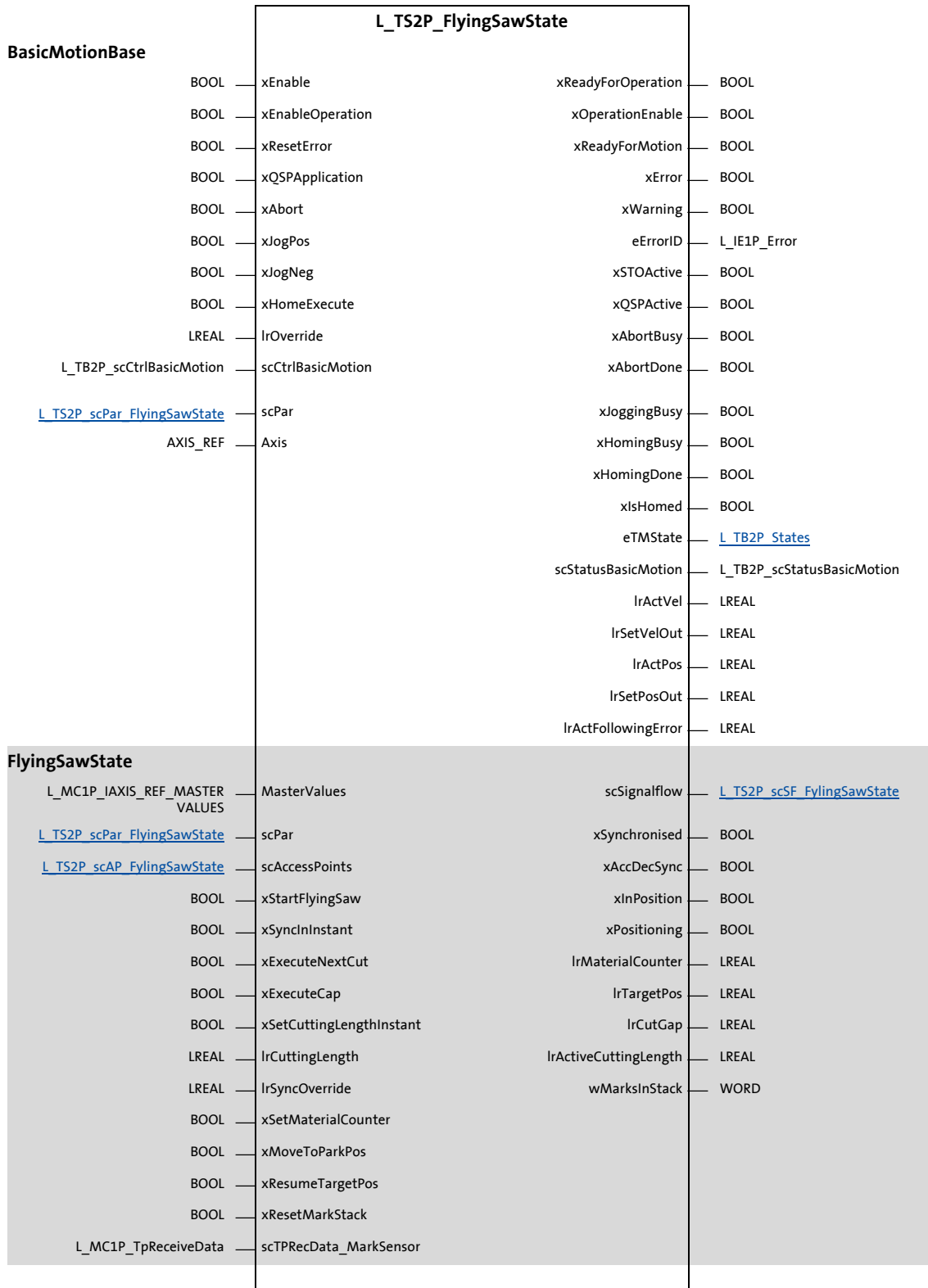
With the technology module "Flying Saw", applications for synchronous sawing and cutting processes on a continuously moving material can be realized.

The typical application is material processing while the material web is running:

- Sawing
- Cutting
- Punching
- Printing
- Drilling
- Sealing
- Grinding

5 Function block L_TS2P_FlyingSawState

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



5.1

Inputs

Designator	Data type	Description
xEnable	BOOL	Execution of the function block
		TRUE The function block is executed.
		FALSE The function block is not executed.
xEnableOperation	BOOL	TRUE The axis is switched to ready for operation. Controller enable is issued.
		FALSE The axis is not ready for operation.
xResetError	BOOL	The input is edge-controlled and evaluates the rising edge.
		FALSE Reset axis or software errors.
xQSPApplication	BOOL	The input is level controlled and evaluates the high level.
		TRUE Active motion is aborted and axis is brought to a standstill via the OSP application deceleration ramp. (35) - The state changes to "STOP". - The technology module remains in the "STOP" state as long as <i>xQSPApplication</i> = TRUE.
xAbort	BOOL	TRUE Active motion is aborted and axis is brought to a standstill via the active profile deceleration. The state changes to "STOP". The deceleration ramp can be aborted by activating a new motion (e.g. <i>xMoveVelExecute</i>)
		FALSE The active motion is not aborted.
xJogPos	BOOL	The input is level controlled and evaluates the high level.
		TRUE Traverse axis in positive direction (manual jog). If <i>xJogNeg</i> is also TRUE, the "InvalidInputJogPosJogNeg" error message is output via the <i>eErrorID</i> output and the axis stops.
xJogNeg	BOOL	The input is level controlled and evaluates the high level.
		TRUE Traverse axis in negative direction (manual jog). If <i>xJogPos</i> is also TRUE, the "InvalidInputJogPosJogNeg" error message is output via the <i>eErrorID</i> output and the axis stops.
xHomeExecute	BOOL	The input is edge-controlled and evaluates the rising edge.
		FALSE Start homing. TRUE The function is aborted via the <i>scCtrlBasicMotion.xHalt</i> structure input.
lrOverride	LREAL	Using the <i>lrOverride</i> factor, the specified setpoint speed is multiplied (e.g. an override value of 0.5 generates half the setpoint speed) - Value range: 0 ... 1.0 - Initial value: 1.0 Note: The acceptance of the value is independent of the status in <i>xContinuousUpdate</i>
scCtrlBasicMotion L_TB2P_scCtrlBasicMotion		Structure of the status data Refer to the documentation of the "Basic Motion (V2)" technology module.
scPar L_TS2P_scPar_FlyingSawState		The parameter structure contains the parameters of the technology module.
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: - AXIS_REF - AXIS_REF_i900 - AXIS_REF_ENC_i900_Drive - AXIS_REF_VIRTUAL - AXIS_REF_VIRTUALM_i900

Designator	Data type	Description
scAccessPoints L_TS2P_scAP_FlyingSawState		Structure of the access points (28) The access points can be used to influence the signal flow or technology module functions.
xStartFlyingSaw	BOOL	The technology module is activated. The input is level controlled and evaluates a high level.
	TRUE	The saw carriage is synchronized to the material position. • Parameter <i>xSyncFlyingSaw</i> = TRUE: The saw carriage is synchronized to the material position. • Parameter <i>xSyncFlyingSaw</i> = FALSE: The carriage is waiting for a signal to synchronize to the material position.
	FALSE	The saw carriage is disengaged from the material position and brakes to a standstill. • If the saw carriage is in the waiting position, it is only decoupled from the material web. • Parameter <i>eFlyingSawStopMode</i> = 1 (MoveToParkPos) The saw carriage is first decoupled and then moved to the parking position (parameter <i>IrParkPosition</i>).
xSyncInInstant	BOOL	The input is edge-controlled and evaluates the rising edge.
	FALSE	The saw carriage is synchronized to the material position. This can be done from the parking position or during a return travel.
	TRUE	
xExecuteNextCut	BOOL	The input is edge-controlled and evaluates the rising edge.
	FALSE	The saw carriage moves to the next machining position.
	TRUE	
xExecuteGap	BOOL	The input is edge-controlled and evaluates the rising edge.
	FALSE	The carriage moves the distance set in the parameter <i>IrGap</i> . This distance is added to the relative travel distance (input <i>IrCuttingLength</i>).
	TRUE	
xSetCuttingLengthInstant	BOOL	The input is edge-controlled and evaluates the rising edge.
	FALSE	The new cutting position is loaded while returning to the next cutting position. The path already travelled is taken into account for the new cutting length.
	TRUE	
IrCuttingLength	LREAL	Cutting length (relative positioning) or cutting position (absolute positioning) • Unit: units from the master axis (mm)
IrSyncOverride	LREAL	Override factor for synchronous motion of the technology module with the cutting position. • Value range: 0.5 ... 3.0 • Initial value: 1.0
	= 1.0	The saw carriage moves synchronously with the material velocity.
	> 1.0	The saw carriage is over-synchronized (faster)
	< 1.0	The saw carriage moves under-synchronized (slower)
xSetMaterialCounter	BOOL	The input is only evaluated if the technology module is in "READY" state. The input is edge-controlled and evaluates the rising edge.
	FALSE	Sets the material position (output <i>IrMaterialCounter</i>) to the value of the parameter <i>IrSetMaterialPos</i> .
	TRUE	
xMoveToParkPos	BOOL	The input is only evaluated if the technology module is in "READY" state. The input is edge-controlled and evaluates the rising edge
	FALSE	The carriage is moved to the parking position (parameter <i>IrParkPosition</i>).
	TRUE	
xResumeTargetPos	BOOL	The input is edge-controlled and evaluates the rising edge.
	FALSE	The carriage is moved to the position from the output <i>IrTargetPos</i> .
	TRUE	

Designator	Data type	Description
xResetMarkStack	BOOL	The input is edge-controlled and evaluates the rising edge.
	FALSE	All marks in the mark stack are removed.
	TRUE	If this input is activated during the engaged state, the axis automatically moves to the waiting position (parameter <i>lrLowerLimit</i>).
scTPRecData_MarkSensor L_MC1P_TpReceiveData		Touch probe structure input for detecting correction

5.2

Outputs

Designator	Data type	Description
xReadyForOperation	BOOL	The axis has no errors. The DC link voltage is present. A "Safe Torque Off" (STO) is not active.
xOperationEnabled	BOOL	The axis is enabled. The brake is released.
xReadyForMotion	BOOL	The axis is enabled and has no errors. 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. Reference Manual "FAST & Technology modules": Here you will find the information about 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) (□ 30) function is active. The technology module is in the "JOGGING" state.
xHomingBusy	BOOL	TRUE The Homing (□ 32) is executed. The technology module is in the "HOMING" state.
xHomingDone	BOOL	TRUE Homing has been successfully completed.
xIsHomed	BOOL	TRUE The axis has been referenced ("Reference known"). <i>xIsHomed</i> may only be evaluated when the <i>scStatusBasicMotion.xAxisAvailable</i> status is reported.
eTMState	L_TB2P_States	Current state of the technology module
scStatusBasicMotion	L_TB2P_scStatusBasicMotion	Output structure of the basic motion function See the documentation for the "Basic Motion (V2)" technology module.
IrActVel	LREAL	Current speed • Unit: units/s
IrSetVelOut	LREAL	Setpoint speed • Unit: units/s
IrActPos	LREAL	Current position • Unit: units
IrSetPosOut	LREAL	Setpoint position • Unit: units
IrActFollowingError	LREAL	Current following error • Unit: units
scSignalflow	L_TS2P_scSF_FlyingSawState	Structure of the signal flow (□ 27)

Designator	Data type	Description	
xSynchronised	BOOL	TRUE	The axis is synchronized with the material web. It considers the tolerance window for the following error monitoring and the status is set to TRUE when the saw carriage travels synchronously to the cutting position <i>IrTargetPos</i> . The tolerance window is configured using these parameters: • <i>xPosInWindow</i> • <i>IrTolerancePosInWindow</i> • <i>IrDwellTimePosInWindow</i>
xAccDecSync	BOOL	TRUE	The synchronization function is active. The axis is synchronized or desynchronized (the clutch opens or closes). These parameters are used for synchronization: • <i>IrSyncVel</i> • <i>IrSyncAcc</i> • <i>IrSyncDec</i> • <i>IrSyncClerk</i>
xInPosition	BOOL	TRUE	The axis has reached the parking position.
xPositioning	BOOL	TRUE	The axis moves to the parking position.
IrMaterialCounter	LREAL	Continuous material position • Unit: units	
IrTargetPos	LREAL	Target position of the saw carriage • Unit: units	
IrCutGap	LREAL	Distance between two cutting positions • Unit: units	
IrActiveCuttingLength	LREAL	Active cutting length of the material that the technology module targets. The cutting length is the difference between the last cut position and the next cut position of the material. • Unit: units of the master axis	
wMarksInStack	WORD	Number of the marks in the stack that still need to be processed.	

5.3

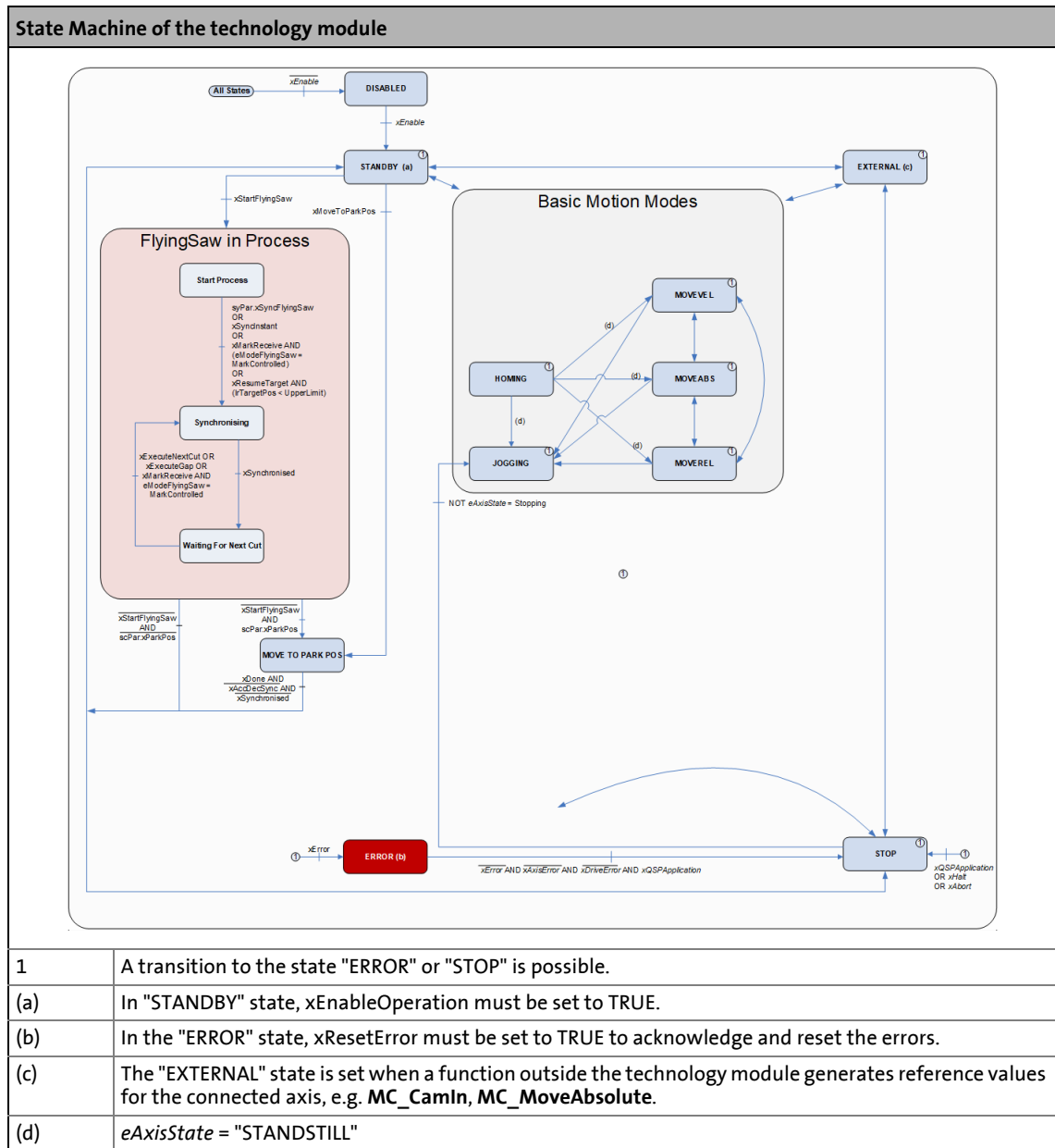
Parameters

L_TS2P_scPar_FlyingSawState

Designator	Data type	Description
IrVel	LREAL	Velocity for the motion commands that are assigned using the scCtrlBasicMotion structure. • Unit: units/s • Initial value: 3600
IrAcc	LREAL	Acceleration for the motion commands that are assigned using the scCtrlBasicMotion structure. Specification of the maximum velocity variation which is to be used for acceleration. • Unit: units/s ² • Initial value: 36000
IrDec	LREAL	Deceleration for the motion commands that are assigned using the scCtrlBasicMotion structure. Specification of the maximum velocity variation which is to be used for deceleration. • Unit: units/s ² • Initial value: 36000
IrJerk	LREAL	Jerk for the motion commands that are assigned using the scCtrlBasicMotion structure. • Unit: units/s ³ • Initial value: 0
eDirection	MC_DIRECTION	Traversing direction: • <i>mcCurrentDirection</i> : Retain current direction (only for modulo axis) • <i>mcPositiveDirection</i> : Positive direction • <i>mcNegativeDirection</i> : Negative direction • <i>mcShortestWay</i> : Shortest distance (only for 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.
eProfileMode	ENUM	The profile mode is used to select the type of profile generator for the flying saw. • Initial value: 0 ▶ Optimized dynamics (📖 50)
	0	Default (default setting) After a cut, the flying saw moves as quickly as possible to the waiting position for the next cut.
	1	Balanced The flying saw moves to the next cutting position with an optimized profile without having to wait long in the parking position.
IrSyncVel	LREAL	Maximum speed for the synchronization and desynchronization process • Unit: units/s • Initial value: 100
IrSyncAcc	LREAL	Acceleration for the synchronization and desynchronization process Specification of the maximum velocity variation which is allowed for acceleration. • Unit: units/s ² • Initial value: 1000
IrSyncDec	LREAL	Deceleration for the synchronization and desynchronization process Specification of the maximum velocity variation which is allowed for deceleration. • Unit: units/s ² • Initial value: 1000

Designator	Data type	Description						
IrSyncJerk	LREAL	Jerk for the synchronization and desynchronization process Specification of the maximum change in acceleration/deceleration allowed for the synchronization and desynchronization process. - Unit: units/s³ - Initial value: 1000000						
IrSetMaterialPos	LREAL	Default value for the material counter The value of the material counter results from the value assigned here plus the actual position of the carriage. The value is accepted if a rising edge (FALSE↗TRUE) is detected at the input <i>xSetMaterialPos</i> . - Unit: units - Initial value: 0						
IrUpperLimit	LREAL	The upper limit for the saw carriage motion - Unit: units - Initial value: 1000						
IrLowerLimit	LREAL	The lower limit for the saw carriage motion - Unit: units - Initial value: 0						
IrGap	LREAL	Distance to be travelled after a saw cut In the length-controlled operation, this value is added with the relative travel distance (input <i>IrCuttingLength</i>). The distance is travelled after a rising edge (FALSE↗TRUE) at the input <i>xExecuteGap</i> . - Unit: units - Initial value: 0						
IrBladeWidth	LREAL	Blade width of the saw blade The value is added to the input <i>IrCuttingLength</i> in the operating mode "Length-controlled operation". - Unit: units - Initial value: 0						
eFlyingSawStopMode L_TS2P_FlyingSawStopMode		This parameter influences the effect of a falling edge at the input <i>xStartFlyingSaw</i> . Initial value: 0 <table><tr><td>0</td><td>Halt: The saw carriage is immediately brought to a standstill.</td></tr><tr><td>1</td><td>MoveToParkPos: The carriage is moved to the parking position (parameter <i>IrParkPosition</i>).</td></tr></table>	0	Halt: The saw carriage is immediately brought to a standstill.	1	MoveToParkPos: The carriage is moved to the parking position (parameter <i>IrParkPosition</i>).		
0	Halt: The saw carriage is immediately brought to a standstill.							
1	MoveToParkPos: The carriage is moved to the parking position (parameter <i>IrParkPosition</i>).							
IrParkPosition	LREAL	The parking position of the carriage - Unit: units - Initial value: 0						
IrMarkSensorPos	LREAL	Position of the touch probe sensor (relating to the measuring system of the saw carriage) Unit: units						
eFlyingSawMode L_TS2P_FlyingSawMode		Selecting the operating mode Initial value: 0 (mark-controlled operation) <table><tr><td>0</td><td>Parameter eFlyingSawMode = 0 (mark-controlled operation) (📖 51) The saw carriage is only synchronized to the material position when a mark has been detected at the connected mark sensor (input <i>xMarkReceive</i> = TRUE).</td></tr><tr><td>1</td><td>Parameter eFlyingSawMode = 1 (📖 53) (length-controlled operation) The saw carriage is only synchronized to the material position when a rising edge (FALSE↗TRUE) is detected at input <i>xSynclnInstant</i>.</td></tr><tr><td>2</td><td>Parameter eFlyingSawMode = 2 (📖 55) (length-controlled operation with automatic synchronization) The carriage is immediately synchronized to the material position if the input <i>xStartFlyingSaw</i> = TRUE. This can be done regardless of the current position of the carriage.</td></tr></table>	0	Parameter eFlyingSawMode = 0 (mark-controlled operation) (📖 51) The saw carriage is only synchronized to the material position when a mark has been detected at the connected mark sensor (input <i>xMarkReceive</i> = TRUE).	1	Parameter eFlyingSawMode = 1 (📖 53) (length-controlled operation) The saw carriage is only synchronized to the material position when a rising edge (FALSE↗TRUE) is detected at input <i>xSynclnInstant</i> .	2	Parameter eFlyingSawMode = 2 (📖 55) (length-controlled operation with automatic synchronization) The carriage is immediately synchronized to the material position if the input <i>xStartFlyingSaw</i> = TRUE. This can be done regardless of the current position of the carriage.
0	Parameter eFlyingSawMode = 0 (mark-controlled operation) (📖 51) The saw carriage is only synchronized to the material position when a mark has been detected at the connected mark sensor (input <i>xMarkReceive</i> = TRUE).							
1	Parameter eFlyingSawMode = 1 (📖 53) (length-controlled operation) The saw carriage is only synchronized to the material position when a rising edge (FALSE↗TRUE) is detected at input <i>xSynclnInstant</i> .							
2	Parameter eFlyingSawMode = 2 (📖 55) (length-controlled operation with automatic synchronization) The carriage is immediately synchronized to the material position if the input <i>xStartFlyingSaw</i> = TRUE. This can be done regardless of the current position of the carriage.							

Designator	Data type	Description
xPosInWindow	BOOL	Activation of the tolerance window for monitoring the following error Initial value: FALSE
		TRUE The tolerance window (following error monitoring) is activated.
IrTolerancePosInWindow	LREAL	Size of the tolerance window for monitoring the following error - Unit: units - Initial value: 0
IrDwellTimePosInWindow	LREAL	Duration of the following error within the tolerance window - Unit: s - Initial value: 0.050



1	A transition to the state "ERROR" or "STOP" is possible.
(a)	In "STANDBY" state, <code>xEnableOperation</code> must be set to TRUE.
(b)	In the "ERROR" state, <code>xResetError</code> must be set to TRUE to acknowledge and reset the errors.
(c)	The "EXTERNAL" state is set when a function outside the technology module generates reference values for the connected axis, e.g. <code>MC_CamIn</code> , <code>MC_MoveAbsolute</code> .
(d)	<code>eAxisState</code> = "STANDSTILL"

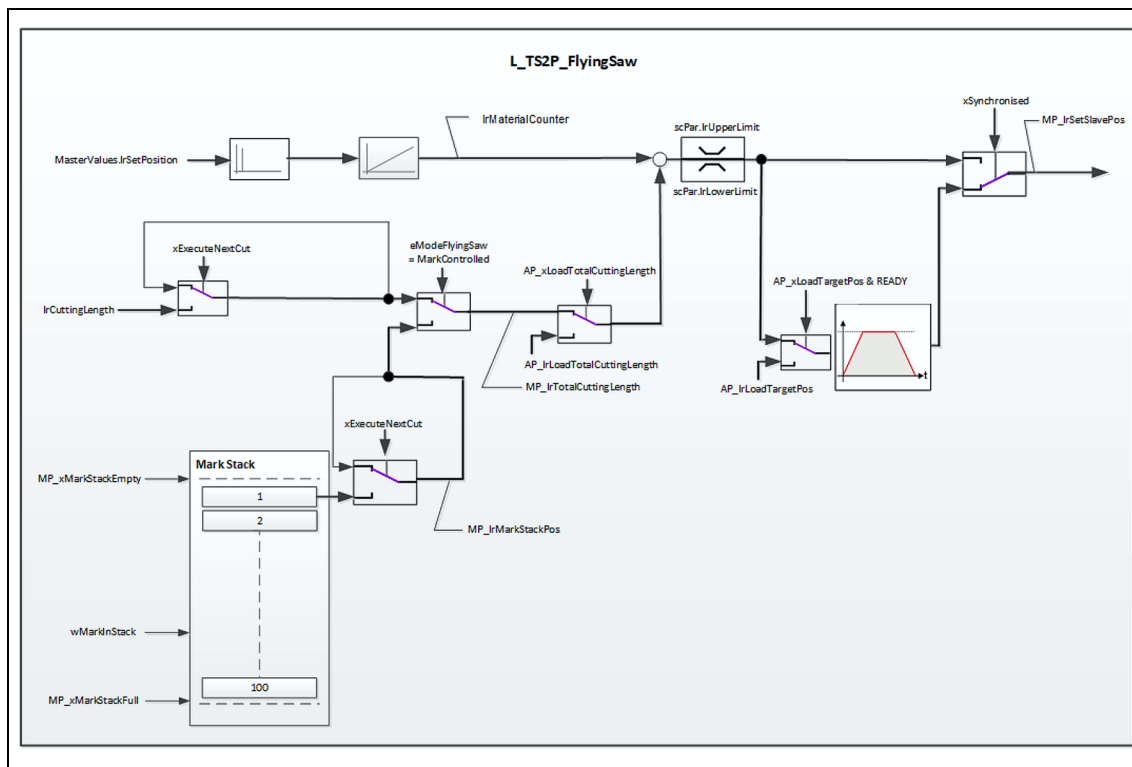
Zustände des Ausgangs eTMState (L_TB2P_States)

No.	L_TB2P_States	Description
-1	DISABLED	Technology module deactivated.
0	STANDBY	Technology module is on standby.
1	STOP	Stop/Halt active.
2	HOMING	Homing active.
3	JOGGING	Manual jog active.
4	MOVEVEL	Endless travel with defined velocity is active.
5	MOVEABS	Absolute positioning active.
6	MOVEREL	Relative positioning active.
7	EXTERNAL	An external function generates reference values for the connected axis.
8	ERROR	Error active.
200	SAW_IN_PROCESS	Technology module active Internal states: • "Start Process": Synchronization function started. • "Synchronizing": Synchronizing function active. • "Waiting For Next Cut": Waiting for the next cut.
210	Move To Park Pos	Movement to the parking position is active.

7 Signal flow diagram

7 Signal flow diagram

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



[7-1] Signal flow of the technology module

7.1 Structure of the signal flow

L_TS2P_scSF_FlyingSawState

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

► [Signal flow diagram](#) (26)

Designator	Data type	Description
MP_lrcutLengthProcessed	LREAL	Integrated distance, travelled since the cut. • Value range: 0 ... Cutting length
MP_lrcutTotalCuttingLength	LREAL	Total sum of all the cutting lengths • Unit: units
MP_lrcutSetMasterPos	LREAL	Setpoint position of the Master-Axis • Unit: units
MP_lrcutSetSlavePos	LREAL	Setpoint position of the Slave-Axis • Unit: units
MP_xPosSynchronised	BOOL	TRUE The reference values of the master axis and the slave axis are synchronized with each other.
MP_xMarkStackEmpty	BOOL	TRUE There is no marker left in the mark stack.
MP_xMarkStackFull	BOOL	TRUE The mark stack is completely filled. The next recognized mark overwrites the oldest registered mark in the mark stack.
MP_lrcutStackMarkPos	LREAL	The position of the marker that is currently coming out from the mark stack. • Unit: units
MP_wMarkIndex	WORD	Index of the active mark from the mark stack

7 Signal flow diagram

7.2 Structure of the access points

7.2 Structure of the access points

L_TS2P_scAP_FlyingSawState

Designator	Data type	Description
AP_xLoadTotalCuttingLength	BOOL	Selection of a value for the total sum of all the cut lengths (variable <i>MP_lrTotalCuttingLength</i>)
		TRUE The value in AP_lrLoadTotalCuttingLength is set.
AP_lrLoadTotalCuttingLength	LREAL	Value for the sum of all the cut lengths (variable <i>MP_lrTotalCuttingLength</i>) • Unit: units • Initial value: 0
AP_xLoadTargetPos	BOOL	Selection of a value for the target position of the carriage (output <i>lrTargetPos</i>) The prerequisite for this is that the technology module should be in the "STANDBY" state.
		TRUE The value in AP_lrLoadTargetPos is set.
AP_lrLoadTargetPos	LREAL	Value for the target position of the carriage (output <i>lrTargetPos</i>) • Unit: units • Initial value: 0

8 Description of the basic functions

Overview of the basic functions

- [Manual jog \(jogging\)](#) (📖 30)
- [Homing](#) (📖 32)
- [QSP application](#) (📖 35)
- [Halt](#) (📖 36)
- [Absolute positioning](#) (📖 37)
- [Relative positioning](#) (📖 39)
- [Continuous travel with a defined velocity](#) (📖 42)
- [Aborting and completing a positioning process](#) (📖 45)
- [Override for speed](#) (📖 47)

8 Description of the basic functions

8.1 Manual jog (jogging)

8.1 Manual jog (jogging)

Prerequisite

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

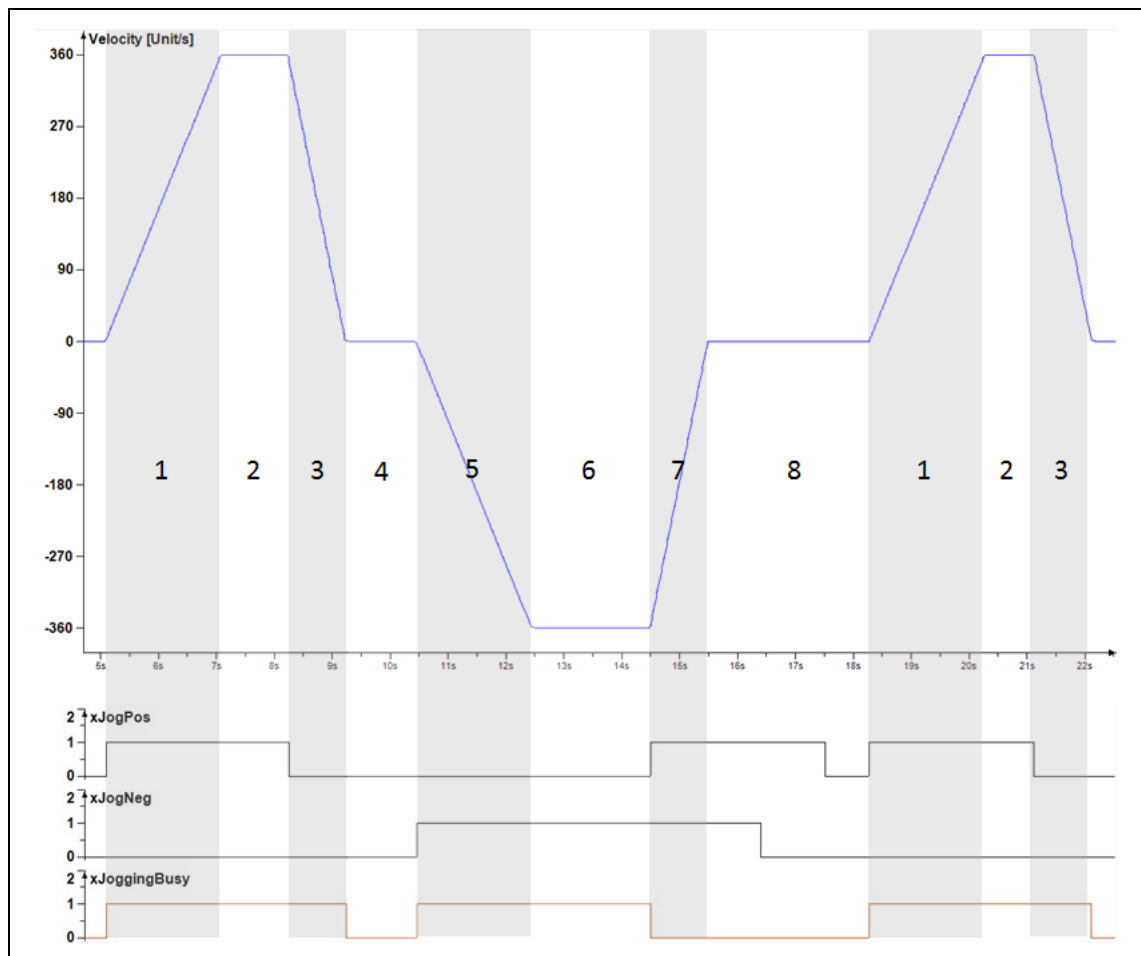
Execution

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

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

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

Travel diagram



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

Parameters to be set

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

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

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

8.2**Homing****Prerequisites**

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

Execution

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

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

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

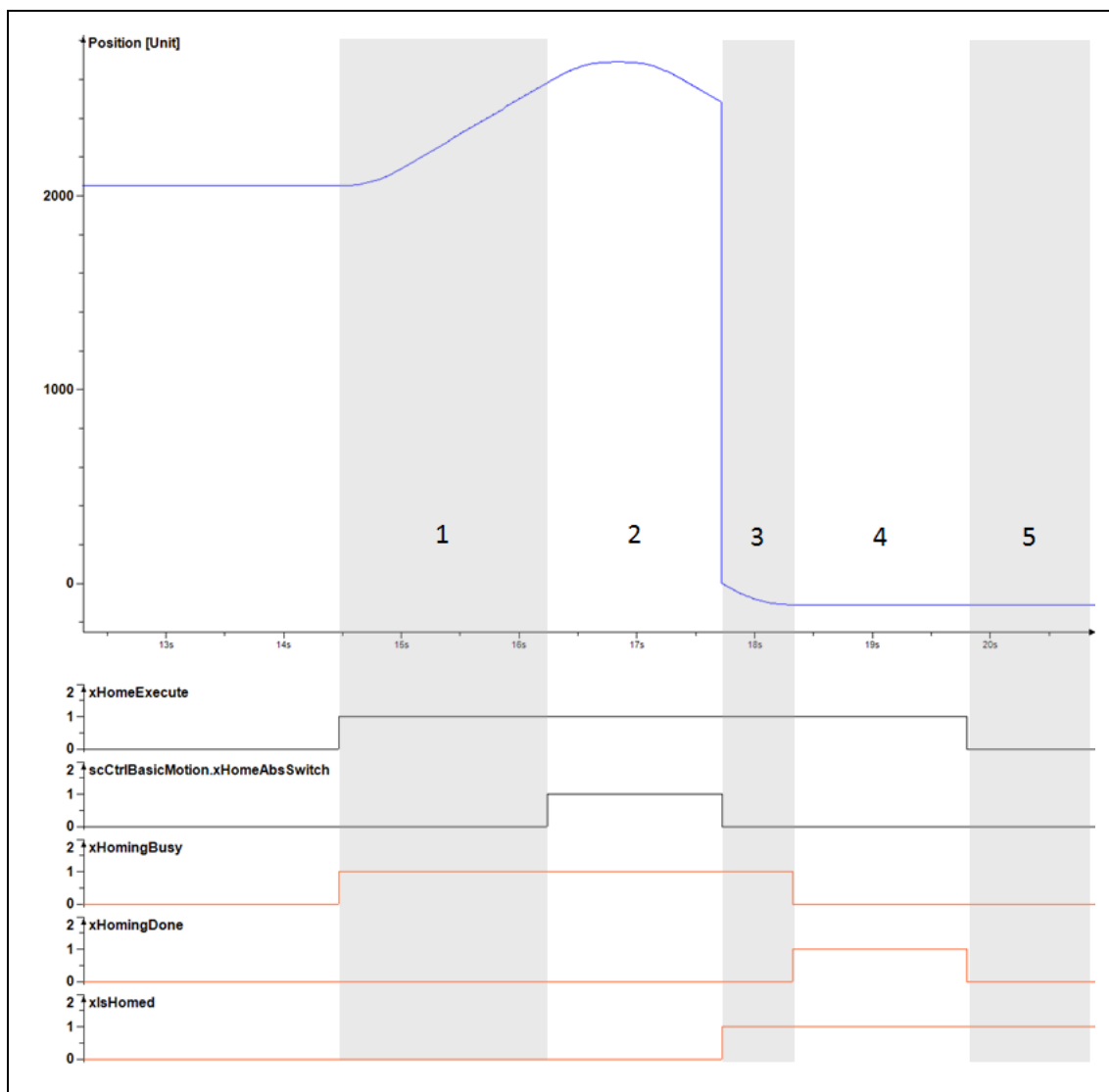
If *xHomeExecute* = FALSE, the state is changed to "STANDBY" and *xHomingDone* is set = FALSE.

Abort of the homing process

The function is aborted via the *xHalt* input.

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

Travel diagram



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

No	xHome Execute	xHome AbsSwitch	xHoming Busy	xHoming Done	xisHomed	eTMState	Info
1	FALSE → TRUE	FALSE	FALSE → TRUE	FALSE	FALSE	STANDBY → HOMING	The axis starts homing in positive direction.
2	TRUE	TRUE → FALSE	TRUE	FALSE	FALSE	HOMING	The axis changes the direction of rotation due to the positive edge of <i>xHomeAbsSwitch</i> .
3	TRUE	FALSE → TRUE	TRUE	FALSE	FALSE → TRUE	HOMING	The negative edge of <i>xHomeAbsSwitch</i> 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, <i>xHomingBusy</i> is set = FALSE and <i>xHomingDone</i> is set = TRUE. <i>xHomingDone</i> remains TRUE for as long as <i>xHomeExecute</i> 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

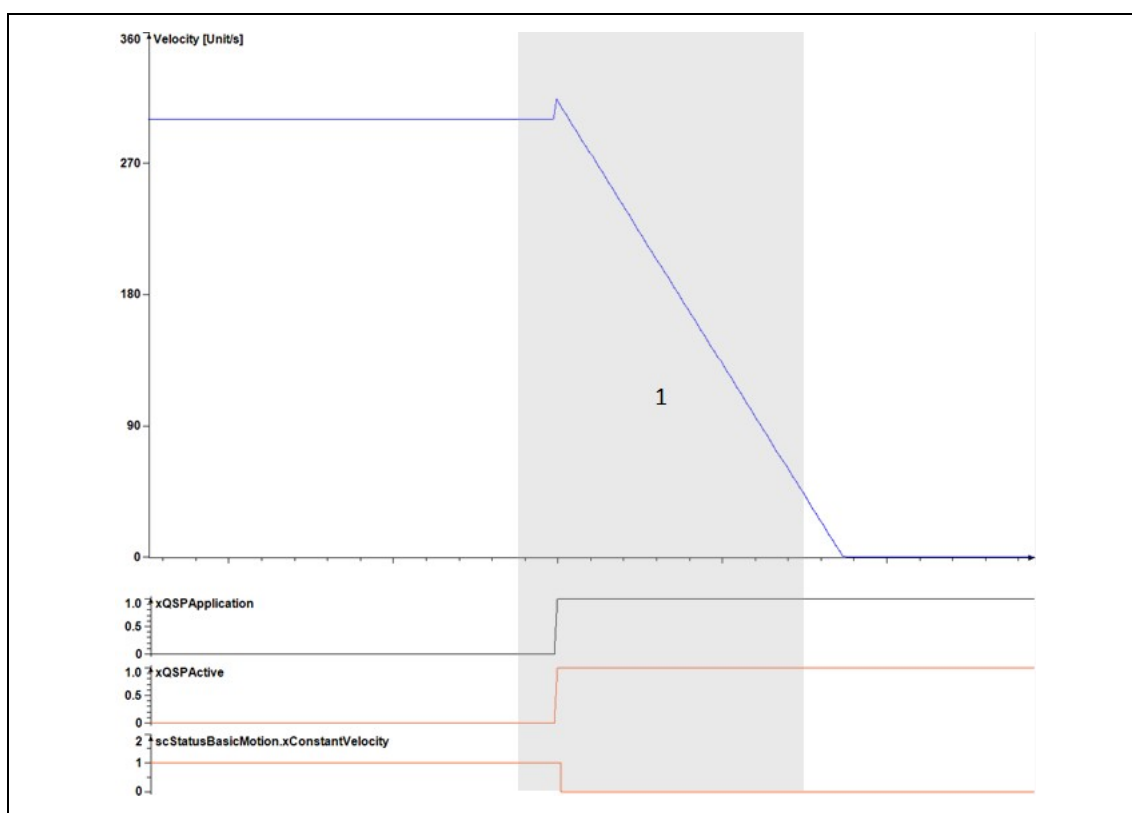
Prerequisite

- 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

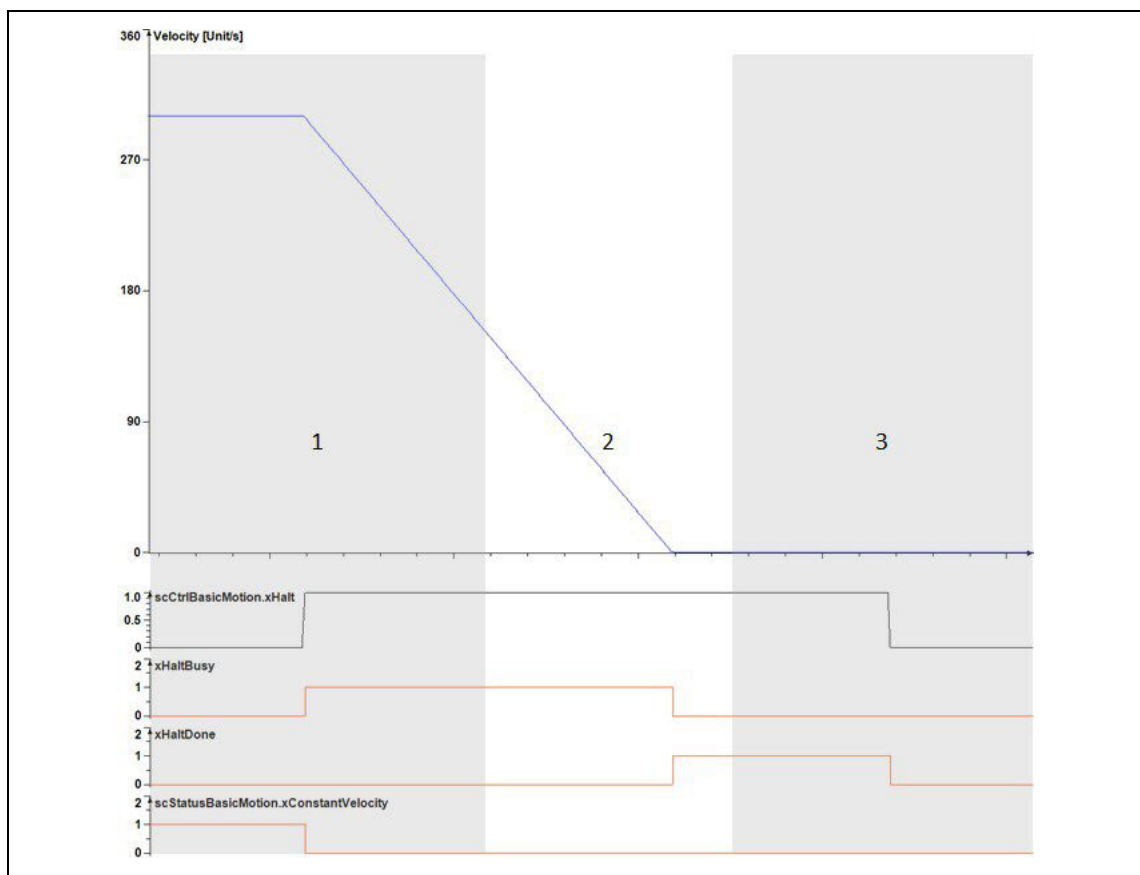
Prerequisite

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

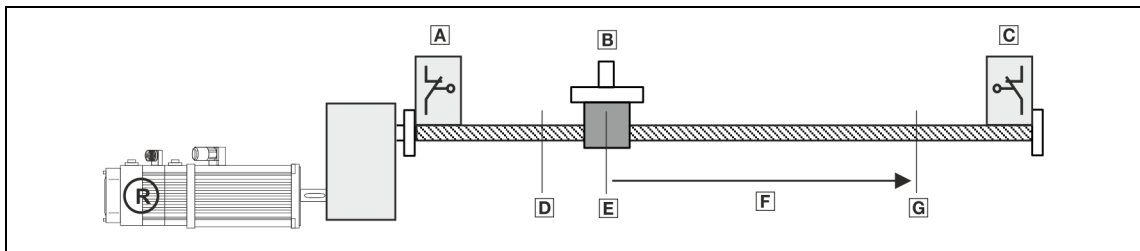
Parameters to be set

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>IrPos_Dist</i> input)

With absolute positioning, the target position in the *IrPos_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 *IrSetPos_Dist* input is started on the basis of the parameters for the motion profile (see below). The technology module changes to the "MOVEABS" state.

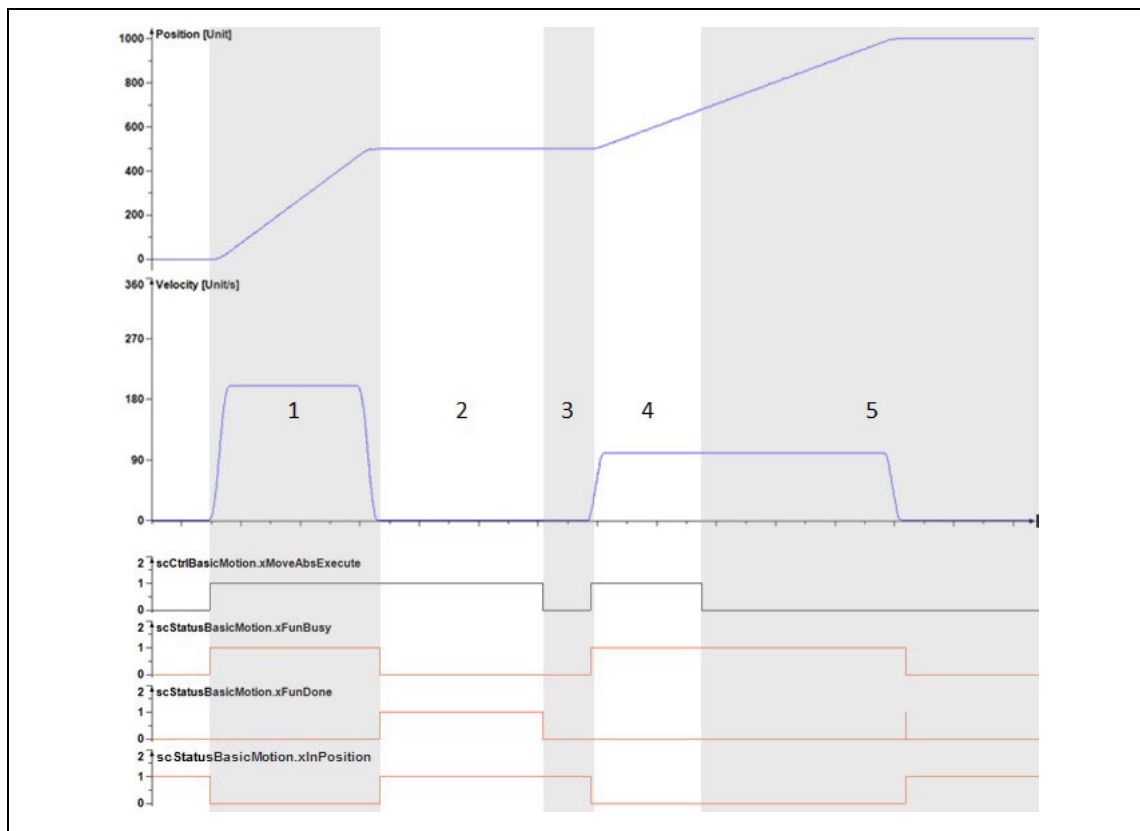
In order to reach the target position, the axis is brought to a standstill via the *IrDec* parameter. The axis reaches a standstill exactly in the target position.

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

During the travel process, the absolute target position in the *IrPos_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

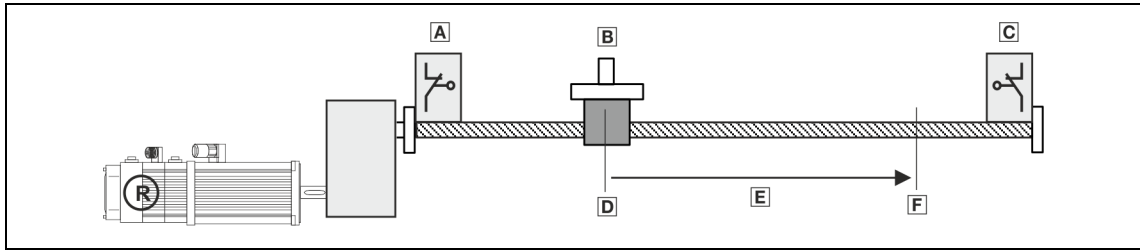
- *IrVel*, initial value: 3600 [units/s]
- *IrAcc*, initial value: 36000 [units/s²]
- *IrDcc*, initial value: 36000 [units/s²]
- *IrJerk*, initial value: 0 [units/s³]

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.

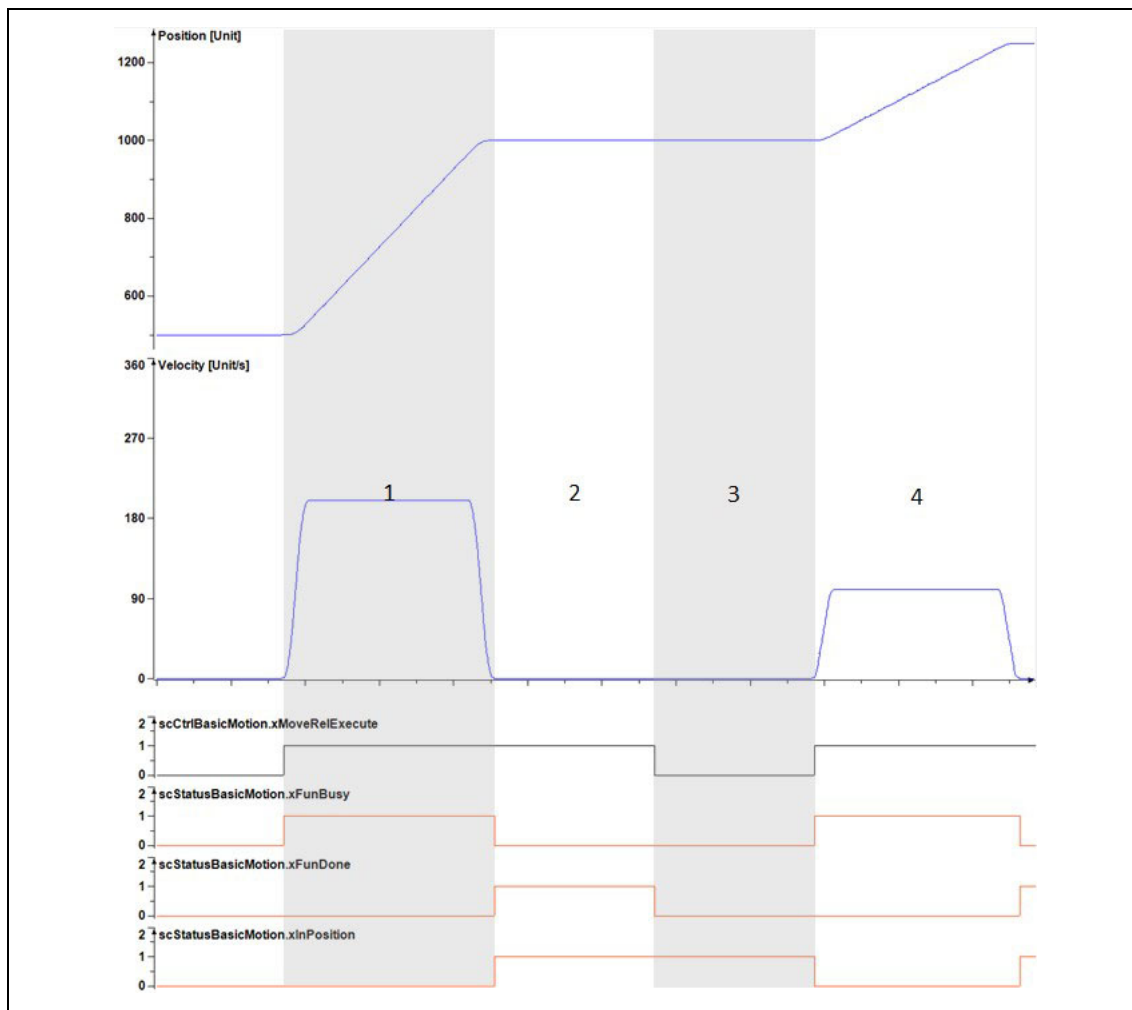
In order to reach the target position, the axis is brought to a standstill via the *lrDec* parameter. The axis reaches a standstill exactly in the target position.

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.

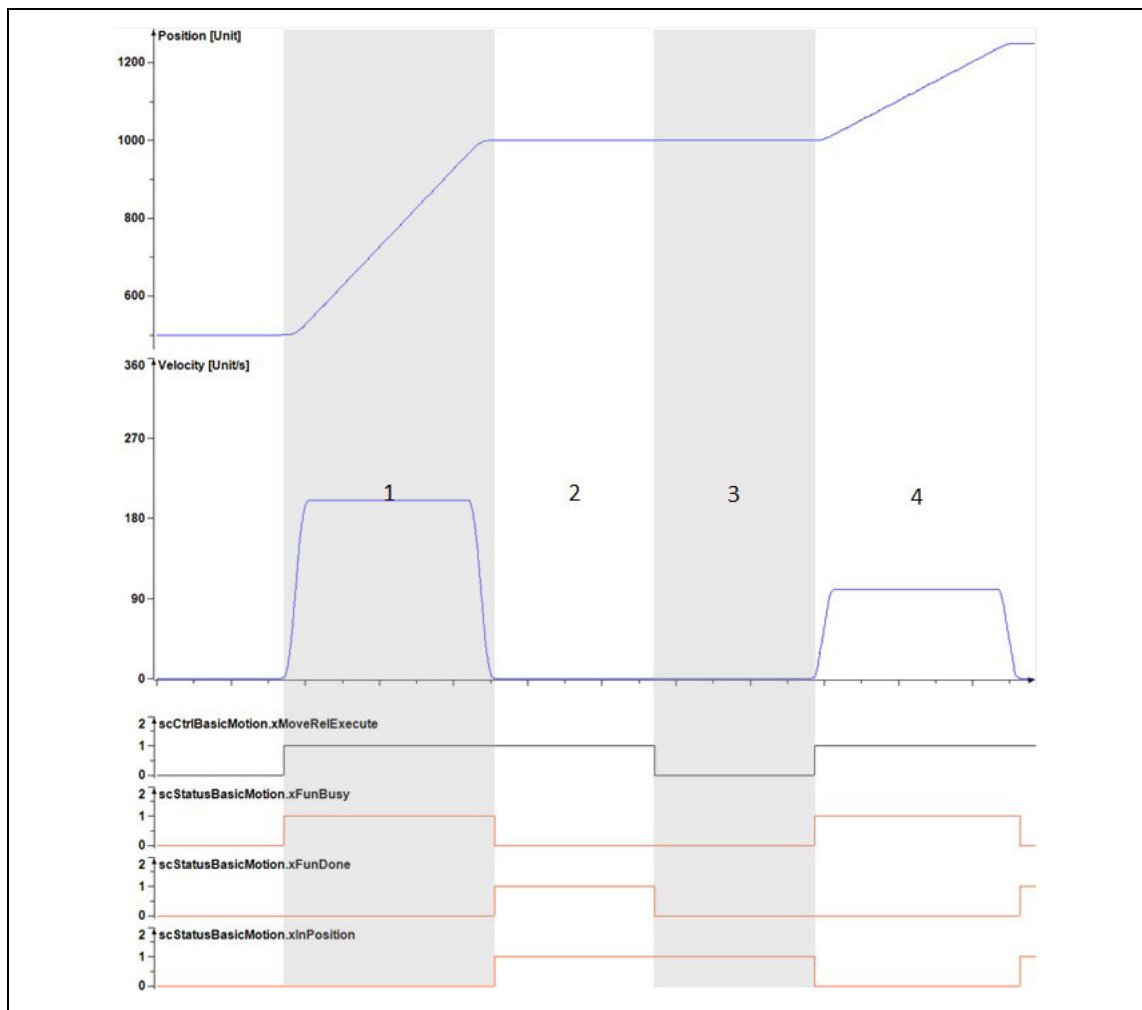
When a hardware limit switch has been reached, the axis is brought to a standstill via the QSP application deceleration ramp.

Then the state is changed back to "STANDBY".

8 Description of the basic functions

8.7 Continuous travel with a defined velocity

Travel diagram



No	xMoveVelExecute	xAbsort	xFuncBusy	xFuncDone	xConstantVelocity	eTMState	Info
1	FALSE → TRUE TRUE → FALSE	FALSE	FALSE → TRUE	FALSE	FALSE	STANDBY → MOVEVEL	The axis accelerates to the target speed of 200 units/s.
2	FALSE	FALSE	TRUE	FALSE → TRUE	FALSE → TRUE	MOVEVEL	After reaching the target speed, xFuncDone changes from FALSE → TRUE and xConstantVelocity changes from FALSE → TRUE.
3	FALSE	FALSE → TRUE	TRUE → FALSE	TRUE → FALSE FALSE → TRUE	TRUE → FALSE	MOVEVEL → 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**

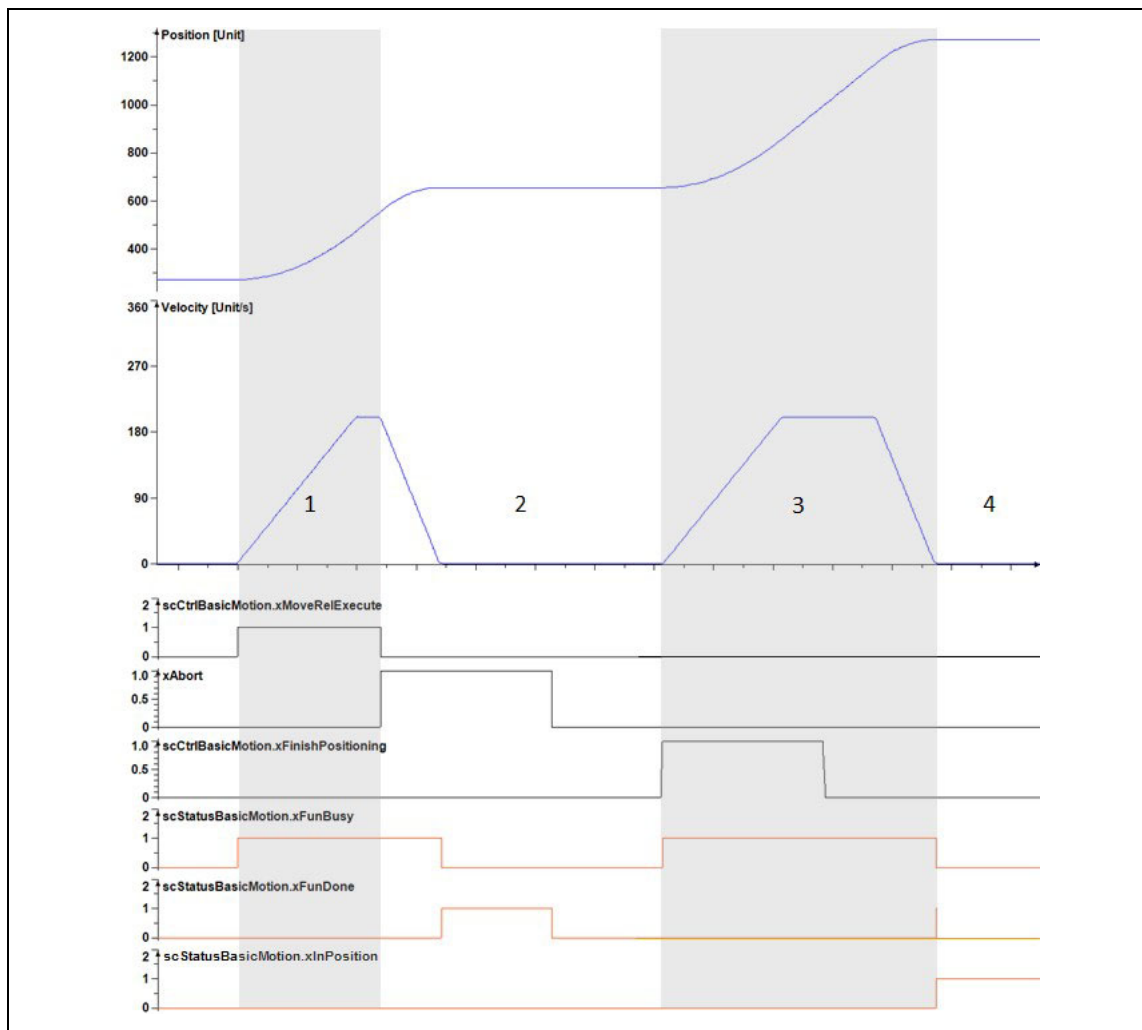
Both technology modules provide several profile data sets, one of which can be active.

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:

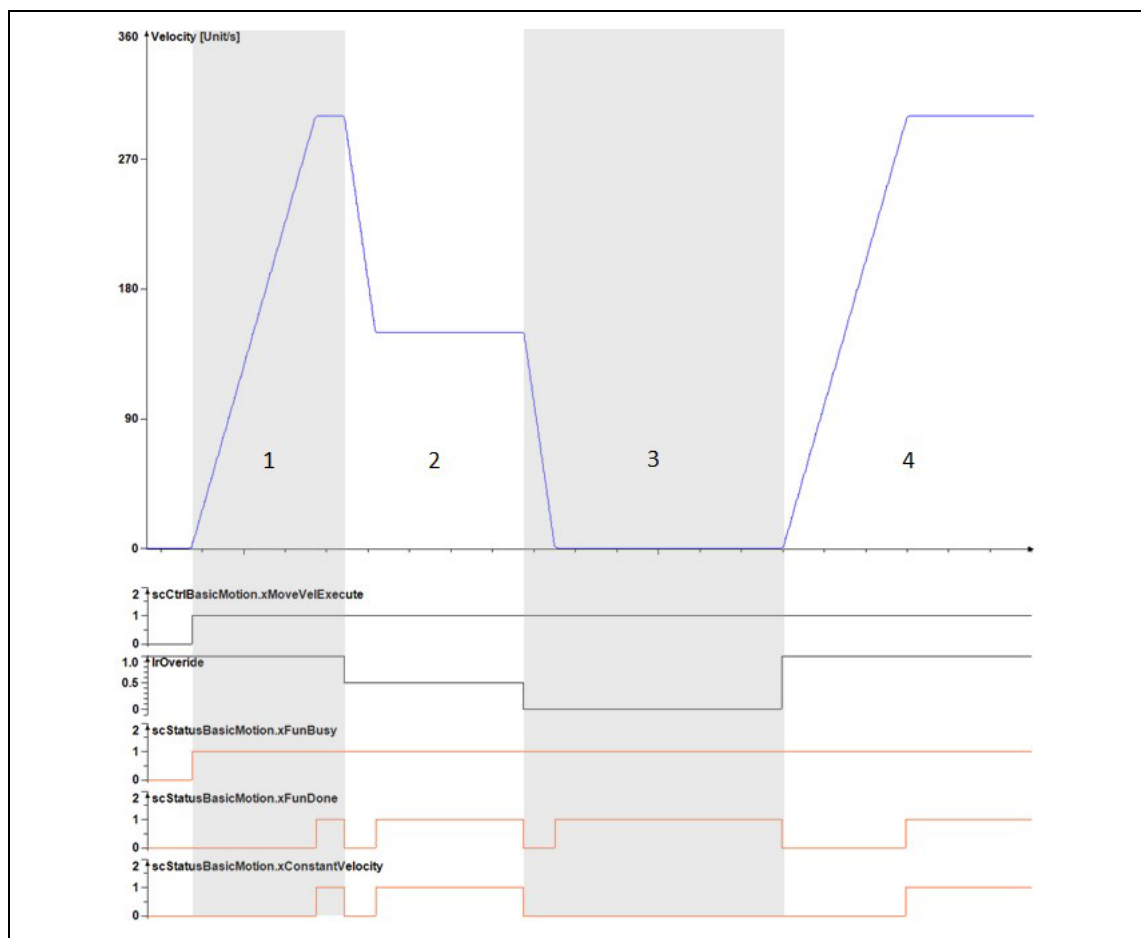
- *IrOverride*, initial value: 1.0
- *scCtrlBasicMotion.IrOverrideAcc*, initial value: 1.0 (100 %)
- *scCtrlBasicMotion.IrOverrideJerk*, initial value: 1.0 (100 %)

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.



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 Set motion profiles (linear/S ramps)

9 Description of the technology functions

9.1 Set motion profiles (linear/S ramps)

The technology module uses the jerk limited acceleration trapezoid as the motion profile for the positioning and synchronizing operations. The advantage is that this motion profile offers the possibility of parameterizing the jerk.

Parameters relevant to the function

For the positioning operations, similar to the functions *xMoveToParkPos*, *xExecuteMoveAbs*, *xExecuteMoveRel*, the following parameters are used.

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

These parameters are used for the coupling processes (synchronizing or desynchronizing):

- *IrSyncVel*, Initial value: 360 units/s
- *IrSyncAcc*, Initial value: 1000 units/s²
- *IrSyncDec*, Initial value: 1000 units/s²
- *IrSyncJerk*, Initial value: 1000000 units/s³

See the parameter structure [L TS2P_scPar FlyingSawState](#) (21).

9.2**Optimized dynamics**

The technology module makes it possible to influence the dynamics of the profiles. By setting the parameter *eProfileMode* it is possible to switch between fast and optimized profiles.

eProfileMode = 0: Default

In this mode, the fastest possible profile is used for positioning the flying saw. To calculate the profile, the set profile parameters are used. For this purpose the set profile parameters are used for the calculation of the profiles.

As a result, after a cut (after the rising edge of the *xExecuteNextCut* input) the flying saw position itself to the lower limit of the working range with the set parameters and waits there for the next cut.

eProfileMode = 1: Balanced (optimized dynamics)

Using this mode, the parameterized profile parameters are lowered as far as possible. The profile is then calculated in such an optimal way that the flying saw does not reach waiting states and always remains in motion.

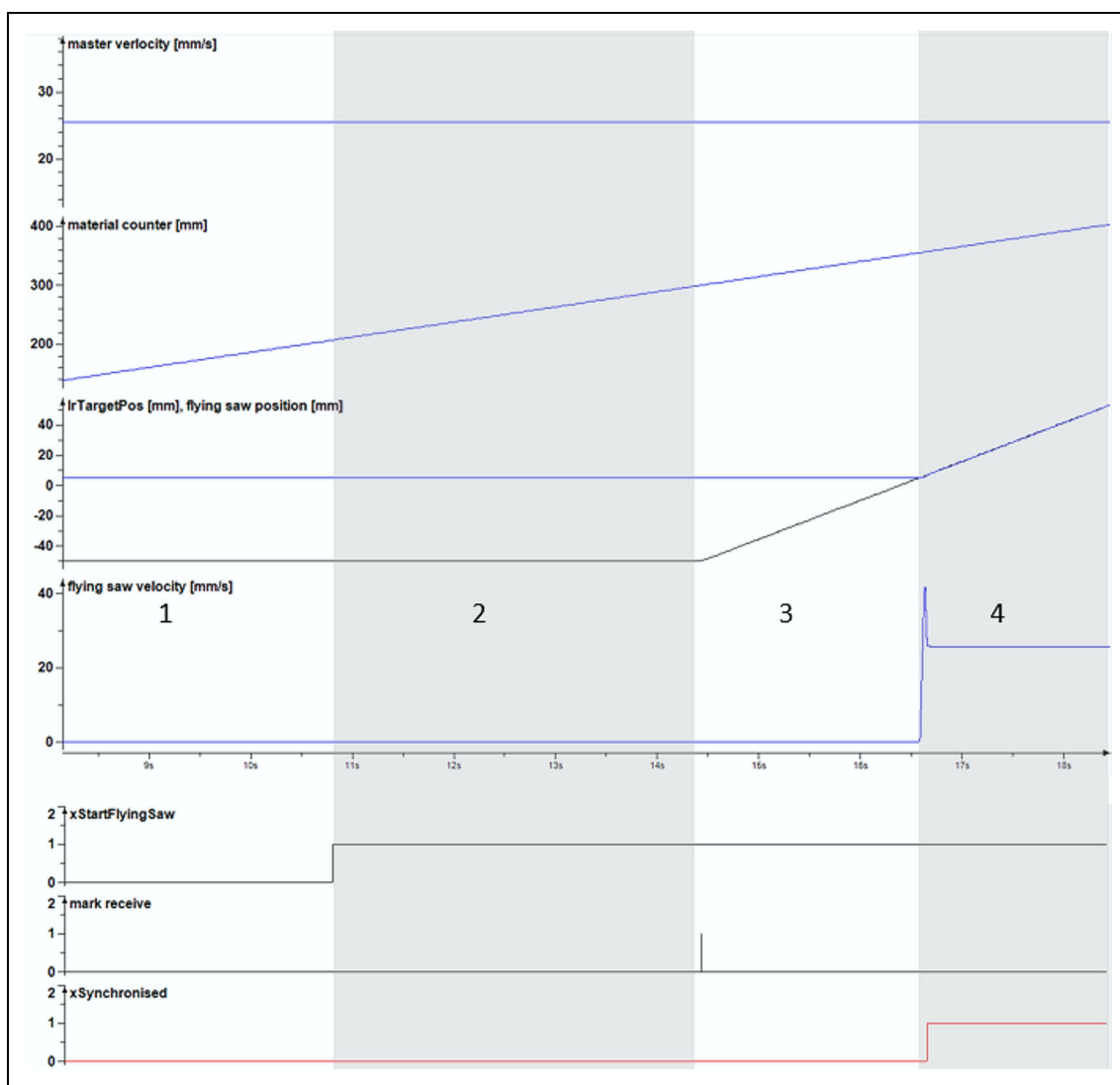
9.3 First synchronization of the saw carriage to the material velocity

When the "Flying saw" has been switched on by the input $xStartFlyingSaw = FALSE \rightarrow TRUE$, there are several possibilities of synchronising the saw carriage with the material velocity.

- ▶ [Parameter eFlyingSawMode = 0 \(mark-controlled operation\) \(51\)](#)
- ▶ [Parameter eFlyingSawMode = 1 \(53\)](#)
- ▶ [Parameter eFlyingSawMode = 2 \(55\)](#)

9.3.1 Parameter eFlyingSawMode = 0 (mark-controlled operation)

The saw carriage is only synchronized to the material position when a mark has been detected at the connected mark sensor (input $xMarkReceive = TRUE$).



[9-1] Example: Behaviour with parameter eFlyingSawMode = 0 (mark-controlled operation)

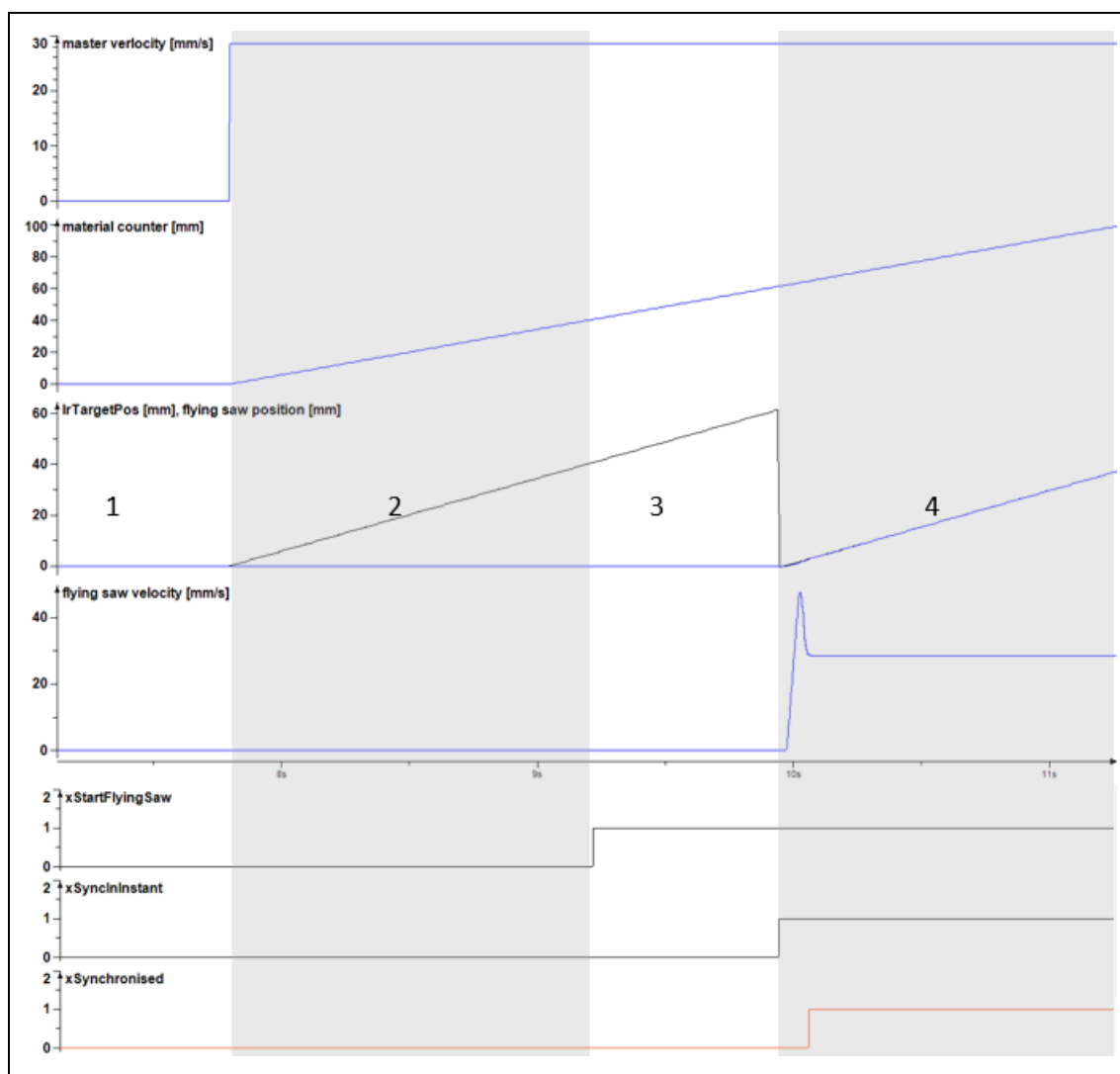
No.	xStartFlyingSaw	mark receive	eTMState	Description
1	FALSE	FALSE	STANDBY	The mark-controlled operation is activated: <i>eFlyingSawMode</i> = 0 The master axis moves, and therefore the material too.
2	TRUE	FALSE	SAW_IN_PROCESS	The state of the technology module changes to "SAW_IN_PROCESS".
3	TRUE	TRUE	SAW_IN_PROCESS	The touch probe sensor detects a mark. The target position <i>lrTargetPos</i> is set to the position of the mark, taking into account the parameter <i>lrMarkSensorPos</i> . The target position is still coupled with the master velocity.
4	TRUE	FALSE	SAW_IN_PROCESS	The slave axis is synchronized to the target position <i>lrTargetPos</i> as soon as it enters the working range of the flying saw. As soon as the slave axis moves synchronously with the target position <i>lrTargetPos</i> , the status <i>xSynchronised</i> = TRUE is set.

9.3.2 Parameter `eFlyingSawMode = 1`

The saw carriage is only synchronized to the material position when a rising edge (`FALSE` → `TRUE`) is detected at the input `xSyncInInstant`.

This function can be used, for example, to perform an initial first cut.

The initial first cut at the start of the length-controlled operation can be performed even if the material web is stationary or running.

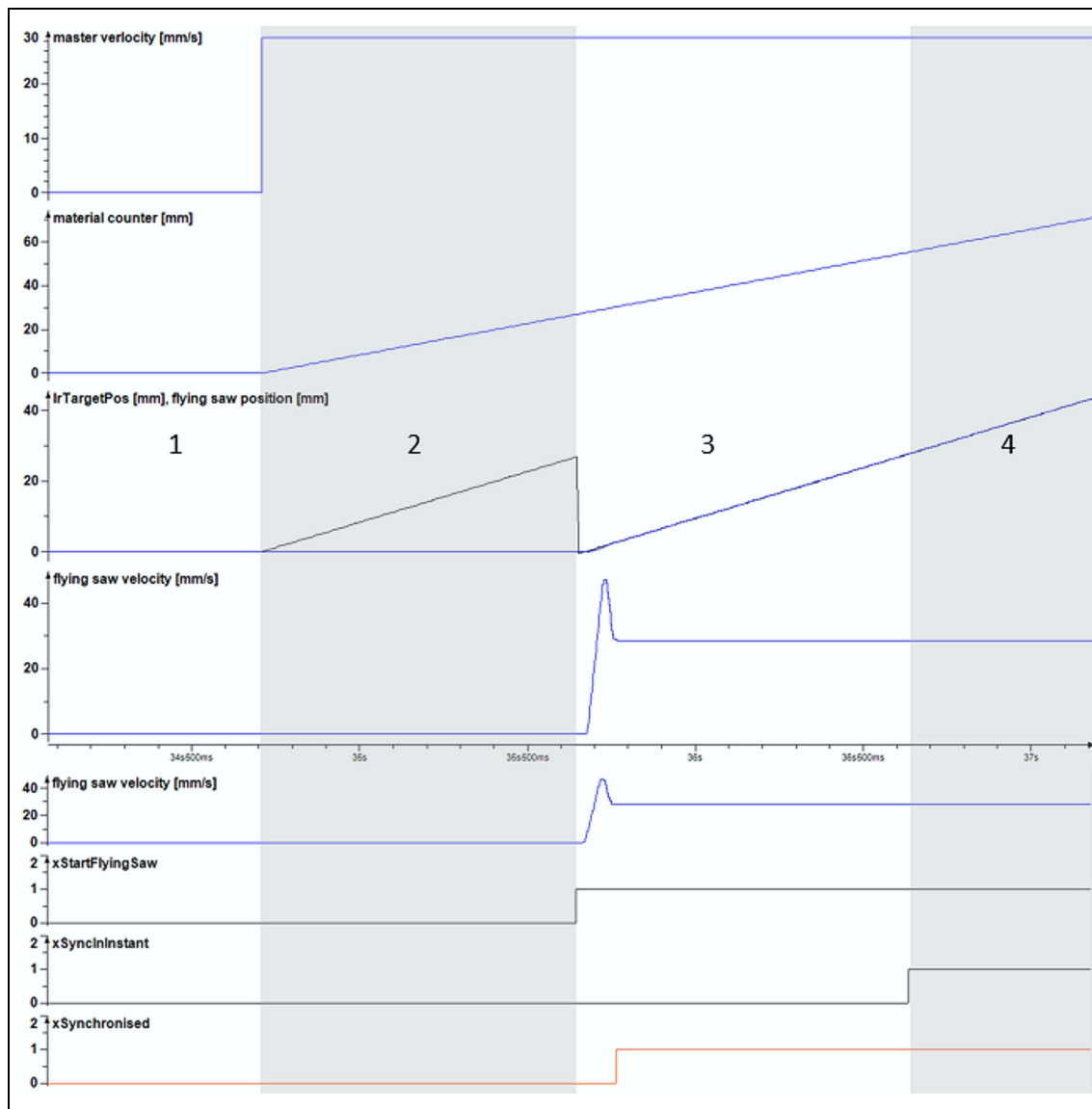


[9-2] Example: Process with parameter `eFlyingSawMode = 1`

No.	xStartFlyingSaw	xSyncInstant	eTMState	Description
1	FALSE	FALSE	STANDBY	The master axis is stationary. The start mode <i>eFlyingSawMode</i> = 1 is set.
2	FALSE	FALSE	STANDBY	The material counter is started. This integrates the material length counter with the master velocity. The target position <i>lrTargetPos</i> is also travelled with the master position.
3	TRUE	FALSE	SAW_IN_PROCESS	The state of the technology module changes to "SAW_IN_PROCESS". The slave axis is not synchronized.
4	TRUE	FALSE → TRUE	SAW_IN_PROCESS	The target position <i>lrTargetPos</i> is set to the slave position. The target position is still synchronized with the master velocity. The slave axis is coupled to the target position. The status <i>xSynchronised</i> = TRUE is set.

9.3.3 Parameter *eFlyingSawMode* = 2

The carriage is immediately synchronized to the material position if the input *xStartFlyingSaw* = TRUE. This can be done irrespective of the current position of the carriage.

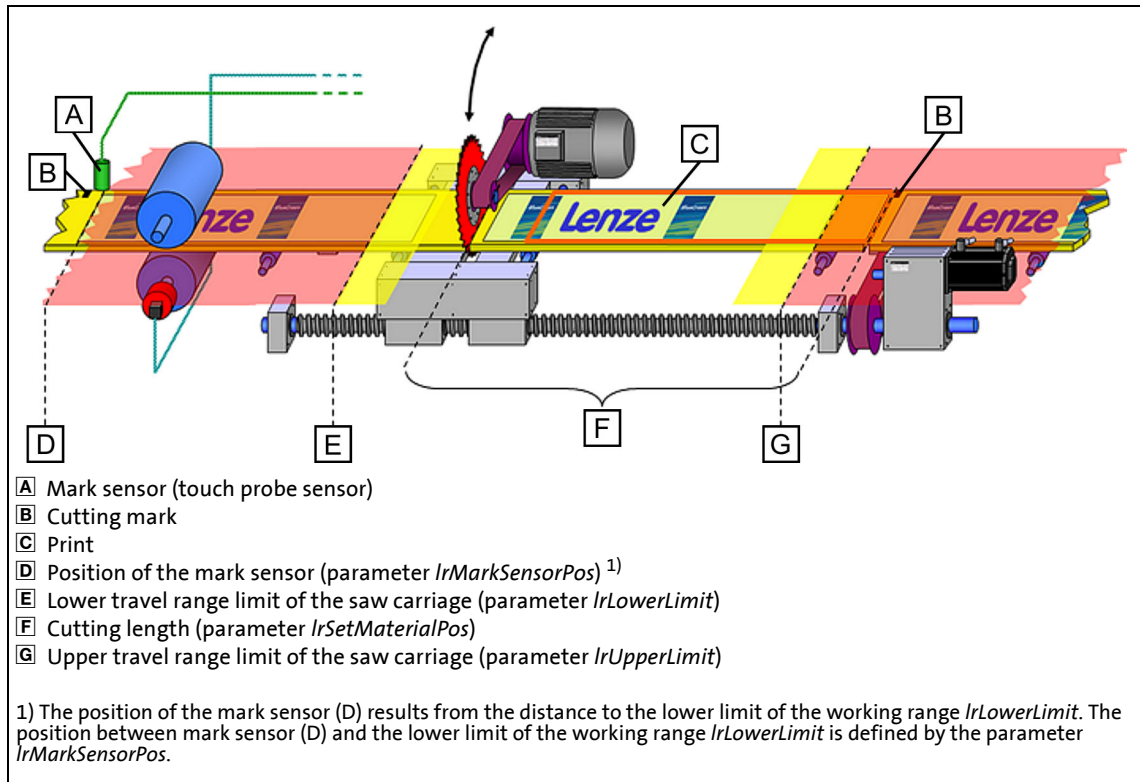


[9-3] Example: Process with parameter *eFlyingSawMode* = 2

No.	xStartFlyingSaw	xSyncInstant	eTMState	Description
1	FALSE	FALSE	STANDBY	The master axis is stationary. The start mode <i>eFlyingSawMode</i> = 2 is set.
2	FALSE	FALSE	STANDBY	The material counter is started. This integrates the material length counter with the master velocity. The target position <i>lrTargetPos</i> is also travelled with the master position.
3	TRUE	FALSE	SAW_IN_PROCESS	The state of the technology module changes to "SAW_IN_PROCESS". The target position <i>lrTargetPos</i> is set to the slave position. The target position is still synchronized with the master velocity. The slave axis is coupled to the target position. The status <i>xSynchronised</i> = TRUE is set.
4	TRUE	FALSE → TRUE	SAW_IN_PROCESS	The <i>xSyncInstant</i> input has no influence.

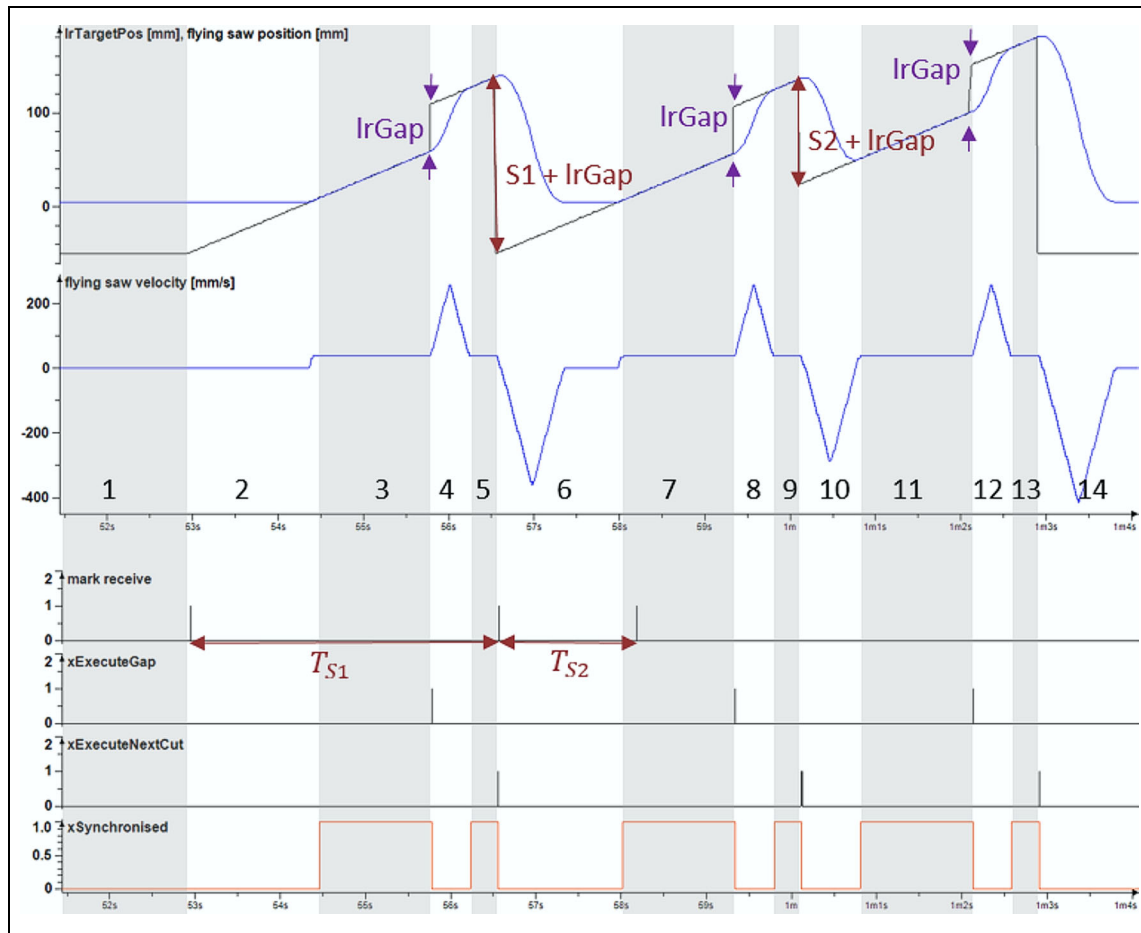
9.4

Mark-controlled operation (cutting on position marks)



[9-4] Cutting length for mark-controlled operation

Mark-controlled operation is used if the material already features partitions by position marks, e.g. in the case of a print image. In this case, the saw cuts must not be placed arbitrarily on the material, rather it must be executed in line with the position marks. An additional mark sensor that is placed in a sufficient distance in front of the operating range of the saw detects the cutting positions beforehand. These cutting positions are then stored in a mark stack. The mark stack contains a maximum of 100 marks.



[9-5] Example: Process of the mark-controlled operation

No.	xExecuteGap	xExecuteNextCut	xSynchronised	Description
1	FALSE	FALSE	FALSE	The master axis, and thus the material, is travelled. The slave axis is at the left limit <i>lrLowerLimit</i> of the work area.
2	FALSE	FALSE	FALSE	A mark is recorded. The target position is set to the sensor position <i>lrMarkSensorPos</i> . The target position <i>lrTargetPos</i> moves synchronously to the material velocity below the limit <i>lrLowerLimit</i> .
3	FALSE	FALSE	TRUE	The target position <i>lrTargetPos</i> moves into the work area. The slave axis is synchronized to the target position <i>lrTargetPos</i> .
4	FALSE→TRUE	FALSE	FALSE	With the command "Execute gap" the target position <i>lrTargetPos</i> jumps by the gap width <i>lrGap</i> in positive direction. The slave axis eliminates the jump of the target position using the profile generator.
5	FALSE	FALSE	TRUE	After a compensation movement, the slave axis again moves synchronously with the target position.
6	FALSE	FALSE→TRUE	FALSE	Another mark is recorded. A FALSE→TRUE edge at the input <i>xExecuteNextCut</i> results in a travel to the next cut position. The target position jumps to the next cut position.
7	FALSE	FALSE	TRUE	The slave axis synchronizes to the new target position. Another mark is detected and tracked in the stack of the technology module.
8	FALSE→TRUE	FALSE	FALSE	With the command "Execute gap" the target position <i>lrTargetPos</i> jumps by the gap width <i>lrGap</i> in positive direction. The slave axis eliminates the jump of the target position using the profile generator.
9	FALSE	FALSE	TRUE	After a compensation movement, the slave axis again moves synchronously with the target position.
10	FALSE	FALSE→TRUE	FALSE	A FALSE→TRUE edge at input <i>xExecuteNextCut</i> moves to the next cutting position. The target position jumps by the mark distance and the gap width. The slave axis eliminates the jump of the target position using the profile generator.
11	FALSE	FALSE	TRUE	The slave axis synchronizes to the new target position. The slave axis synchronizes to the new target position.
12	FALSE→TRUE	FALSE	FALSE	With the command "Execute gap" the target position <i>lrTargetPos</i> jumps by the gap width <i>lrGap</i> in positive direction. The slave axis eliminates the jump of the target position using the profile generator.

No.	xExecuteGap	xExecuteNextCut	xSynchronised	Description
13	FALSE	FALSE	TRUE	After a compensation movement, the slave axis again moves synchronously with the target position.
14	FALSE	FALSE→TRUE	FALSE	A FALSE→TRUE edge at the input <i>xExecuteNextCut</i> is to move to the next cutting position. Since there is no further marks in the stack, the target position jumps to the sensor position and remains there. The slave axis is positioned to the lower limit of the working range via the profile generator.

Ablauf

In the time diagram in Fig. [9-5] three marks are detected in succession. The first mark starts the process. During the processing phase (cutting and spacing) of the first detected mark, two more marks are detected and placed on the mark stack. With a rising edge (FALSE→TRUE) at the input *xExecuteNextCut* = TRUE the next valid marking position is approached.

Approaching the lower travel range limit

If there are no more valid marks on the mark stack, the carriage moves to the lower travel range limit (parameter *lrLowerLimit*) and waits for the next mark.

Cutting length

The cutting length is the distance between the detected marks. As soon as the technology module is switched on, the position at the input *lrActMarkPosIn* is converted to the target position of the saw carriage for each detected mark at the input *xReceiveMark* and stored on the mark stack. With a rising edge (FALSE→TRUE) at the input *xExecuteNextCut*, it is possible to switch to the next mark in the stack or a new or changed cutting length is accepted.

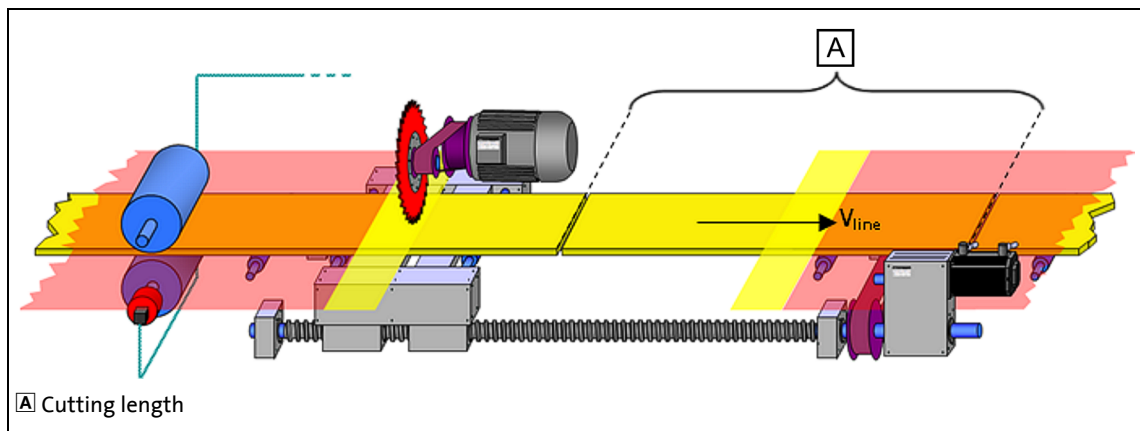
Parameters to be set

- *lrLowerLimit*, Initial value: 0 units
- *lrUpperLimit*, Initial value: 1000 units
- *lrGap*, Initial value: 50 units
- *eFlyingSawMode*, Initial value: 0 (mark-controlled operation)
- *lrMarkSensorPos*, Initial value: 0 units

See the parameter structure [L TS2P_scPar_FlyingSawState](#) (21).

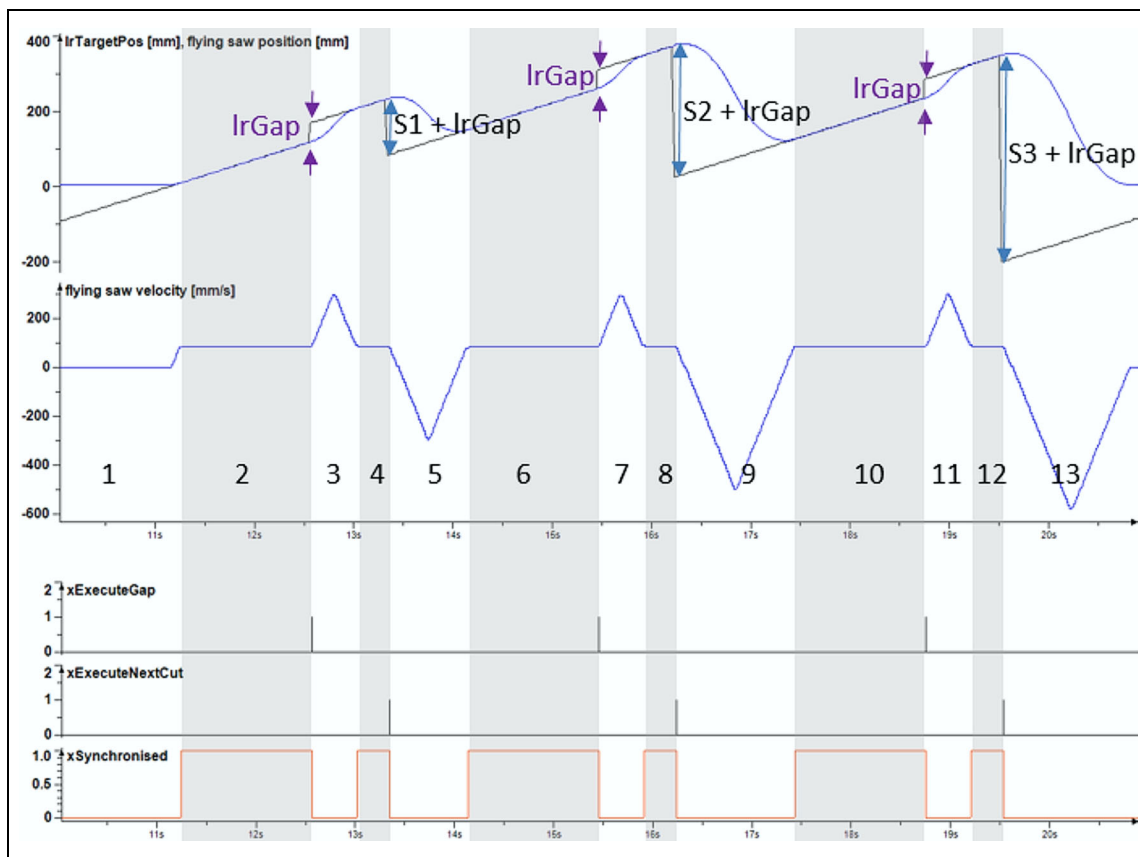
9.5

Length-controlled operation (cutting relative product lengths)



[9-6] Cutting length for length-controlled operation

In length-controlled operation, the first machining operation (first cut) starts at any position. After that, all the subsequent saw cuts have the desired relative distance (cutting length) to the respective preceding cut.



[9-7] Example: Process of the length-controlled operation with the command "Execute gap"

No.	xExecuteGap	xExecuteNextCut	xSynchronised	Description
1	FALSE	FALSE	FALSE	The slave axis is at the left end position <i>lrLowerLimit</i> of the working range. The target position <i>lrTargetPos</i> moves synchronously with the material velocity below the limit <i>lrLowerLimit</i> .
2	FALSE	FALSE	TRUE	The target position <i>lrTargetPos</i> moves into the work area. The slave axis is coupled to the target position <i>lrTargetPos</i> . The cutting process is then performed on the material.
3	FALSE → TRUE	FALSE	FALSE	With the command "Execute gap" the target position <i>lrTargetPos</i> jumps by the gap width <i>lrGap</i> in positive direction. The slave axis smooths out the jump of the target position <i>lrTargetPos</i> via the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> .
4	FALSE	FALSE	TRUE	After the jump is smoothed out, the slave axis continues to travel synchronously to the target position <i>lrTargetPos</i> .
5	FALSE	FALSE → TRUE	FALSE	With the command "Next Cut" (<i>xExecuteNextCut</i>) the target position jumps by the cutting length <i>lrCuttingLength</i> = <i>S1</i> added with the gap width <i>lrGap</i> in negative direction. The slave axis smooths out the jump of the target position <i>lrTargetPos</i> via the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> .
6	FALSE	FALSE	TRUE	After the jump is smoothed out, the slave axis is travelled relatively by the cutting length plus the gap <i>lrGap</i> to the material in the negative direction. The slave axis moves synchronously with the target position <i>lrTargetPos</i> . The cutting process is then performed on the material.
7	FALSE → TRUE	FALSE	FALSE	With the command "Execute gap" the target position <i>lrTargetPos</i> jumps by the gap width <i>lrGap</i> in positive direction. The slave axis smooths out the jump of the target position <i>lrTargetPos</i> via the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> .
8	FALSE	FALSE	TRUE	After the jump is smoothed out, the slave axis continues to travel synchronously to the target position <i>lrTargetPos</i> .

No.	xExecuteGap	xExecuteNextCut	xSynchronised	Description
9	FALSE	FALSE↗TRUE	FALSE	With the command "Next cut" (<i>xExecuteNextCut</i>) the target position jumps by the cutting length <i>lrCuttingLength</i> = S2 added with the gap width <i>lrGap</i> in negative direction. The slave axis smoothes out the jump of the target position <i>lrTargetPos</i> over the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> . If the target position is below the limit <i>lrLowerLimit</i> , the slave axis is stopped exactly at the <i>lrLowerLimit</i> . The slave axis is synchronized to the target position <i>lrTargetPos</i> via the profile generator as soon as the target position <i>lrTargetPos</i> reaches the work area.
10	FALSE	FALSE	TRUE	After the jump is smoothed out, the slave axis is travelled relatively by the cutting length plus the gap <i>lrGap</i> to the material in the negative direction. The slave axis moves synchronously with the target position <i>lrTargetPos</i> . The cutting process is then performed on the material.
11	FALSE↗TRUE	FALSE	FALSE	With the command "Draw gap" the target position <i>lrTargetPos</i> jumps by the gap width <i>lrGap</i> in the positive direction. The slave axis smoothes out the jump of the target position <i>lrTargetPos</i> via the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> .
12	FALSE	FALSE	TRUE	After the jump is smoothed out, the slave axis continues to travel synchronously to the target position <i>lrTargetPos</i> .
13	FALSE	FALSE↗TRUE	FALSE	With the command "Next cut" (<i>xExecuteNextCut</i>) the target position jumps by the cutting length <i>lrCuttingLength</i> = S3 added with the gap width <i>lrGap</i> in the negative direction. The slave axis smoothes out the jump of the target position <i>lrTargetPos</i> via the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> .

Procedure

The timing diagram in Fig. [\[9-7\]](#) shows three different saw cuts S1 ... S3 are performed. After each cut a certain distance (parameter *IrGap*) is travelled.

The movement from one machining position to the next is carried out via a superimposed relative positioning. The positioning distance S1 ... S3 results from the cutting length at the input *IrCuttingLength* added with the length of the moved distance in the parameter *IrGap*.

Next cut

With a rising edge (FALSE↗TRUE) at the input *xExecuteNextCut* the relative positioning is triggered or a new or changed cutting length is accepted.

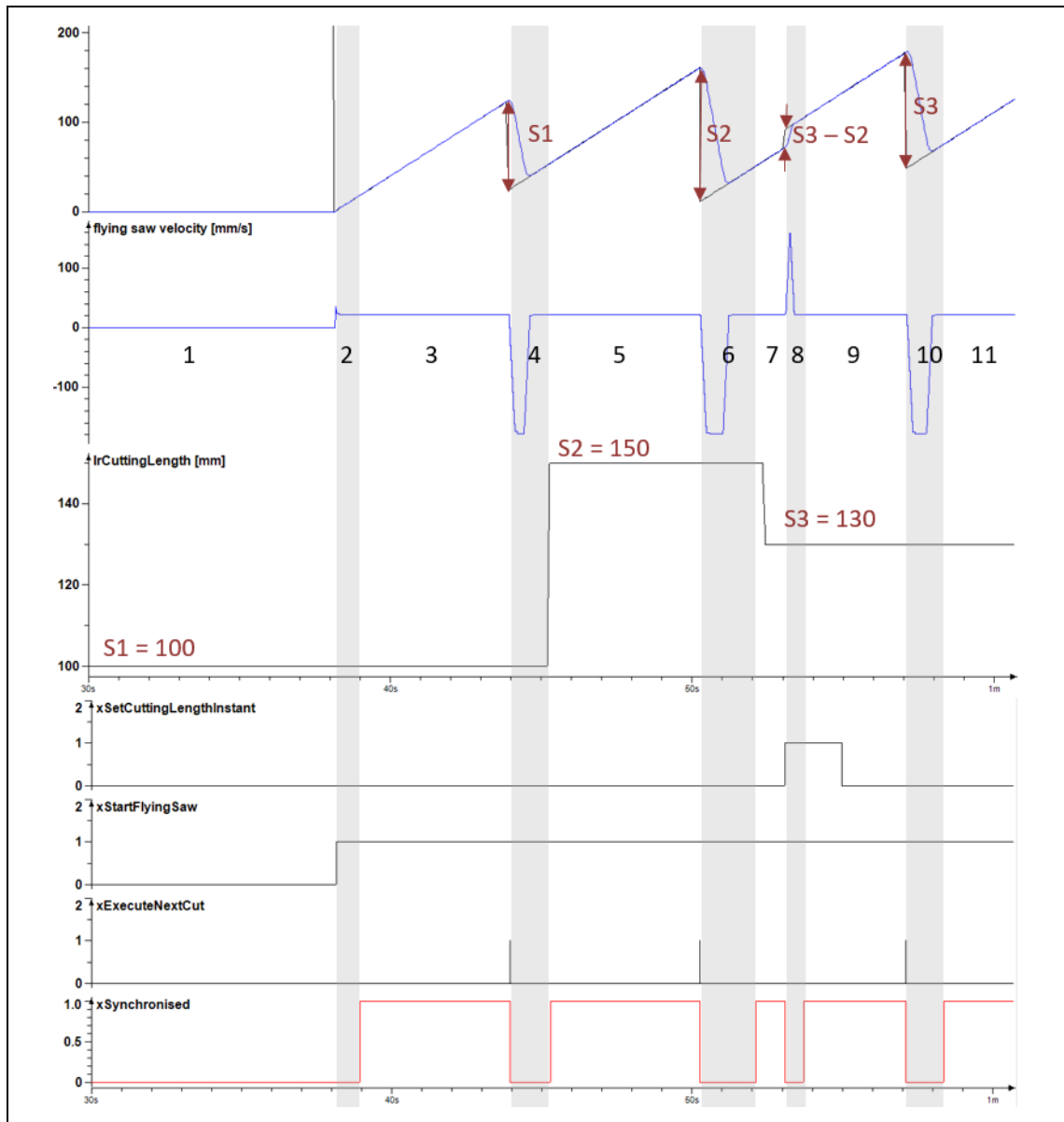
The grey-shaded areas in Fig. [\[9-7\]](#) indicate when the carriage is synchronous with the material position. In the example, the saw carriage is always synchronized to the material position if a new target position is requested with input *xExecuteNextCut* = TRUE or if a distance (parameter *IrGap*) is travelled after the saw cut with the input *xExecuteGap* = TRUE.

Changing the cutting length

The cutting length is defined at the input *IrCuttingLength*. The change of the cutting length is accepted with the next cut, i.e. with a rising edge (FALSE↗TRUE) at the input *xExecuteNextCut*.

It is also possible to update the cutting length in the middle of a cutting process. With a rising edge (FALSE↗TRUE) at the input *xSetCuttingLengthInstant*, the active cutting length is discarded and a new cutting length is loaded at input *IrCuttingLength*. The distance already travelled since the last cut is considered in this new cutting length.

The cutting length cannot be updated via a rising edge at the input *xSetCuttingLengthInstant* if the technology module is already moving synchronously to the cutting position.



[9-8] Example: Process "Change of cutting length"

No.	xExecuteNextCut	xSetCuttingLength Instant	xSynchronised	Description
1	FALSE	FALSE	FALSE	The material is travelled. The target position <i>lrTargetPos</i> is also traversed with the master position.
2	FALSE	FALSE	FALSE	With <i>xStartFlyingSaw</i> = TRUE the state of the technology module changes to "SAW_IN_PROCESS". The target position <i>lrTargetPos</i> is set to the slave position. The target position is further coupled with the master velocity. The slave axis is coupled to the target position.
3	FALSE	FALSE	TRUE	The slave axis moves synchronously to the target position <i>lrTargetPos</i> .
4	TRUE	FALSE	FALSE	With the command "Next cut" (<i>xExecuteNextCut</i>) the target position jumps by the cutting length <i>lrCuttingLength</i> = S1 in the negative direction. The slave axis smoothes out the jump of the target position <i>lrTargetPos</i> via the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> .
5	FALSE	FALSE	TRUE	The slave axis moves synchronously to the target position <i>lrTargetPos</i> . With the input <i>lrCuttingLength</i> = S2 a new cutting length is defined. The technology module initially does not update the new cutting length.
6	TRUE	FALSE	FALSE	With the command "Next cut" (<i>xExecuteNextCut</i>) the target position jumps by the cutting length <i>lrCuttingLength</i> = S2 in the negative direction. Thus, the new cutting length has been updated. The slave axis smoothes out the jump of the target position <i>lrTargetPos</i> via the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> .
7	FALSE	FALSE	TRUE	The slave axis moves synchronously to the target position <i>lrTargetPos</i> . With the input <i>lrCuttingLength</i> = S3 a new cutting length is defined. The technology module initially does not update the new cutting length.
8	FALSE	TRUE	FALSE	The technology module takes over the new cutting length from the synchronous drive. This replaces the cutting length S2 with the new cutting length S3. The target position <i>lrTargetPos</i> jumps to the difference of the cutting lengths S2 - S3. The slave axis smoothes out the jump of the target position <i>lrTargetPos</i> via the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> .
9	FALSE	FALSE	TRUE	The slave axis moves synchronously to the target position <i>lrTargetPos</i> .

No.	xExecuteNextCut	xSetCuttingLengthInstant	xSynchronised	Description
10	TRUE	FALSE	FALSE	With the command "Next cut" (<i>xExecuteNextCut</i>) the target position jumps by the cutting length <i>IrCuttingLength</i> = S3 in the negative direction. Thus, the new cutting length has been updated. The slave axis smoothes out the jump of the target position <i>IrTargetPos</i> via the profile generator with the defined parameters <i>IrSyncVel</i> , <i>IrSyncAcc</i> and <i>IrSyncDec</i> .
11	FALSE	FALSE	TRUE	The slave axis moves synchronously to the target position <i>IrTargetPos</i> .

Working range

The parameters *IrLowerLimit* and *IrUpperLimit* can be used to set the working range of the saw carriage. In the request for the saw cut S3, it can be clearly seen that the target position (output *IrTargetPos*) is below the limit *IrLowerLimit*. In this case, the carriage moves to the specified lower limit and waits there until the target position enters the working area, and then synchronizes to it.

Parameters to be set

- *IrLowerLimit*, Initial value: 0 units
- *IrUpperLimit*, Initial value: 1000 units
- *IrGap*, Initial value: 50 units
- *eFlyingSawMode*, Initial value: 0 (mark-controlled operation)
- *IrSyncVel*, Initial value: 360 units/s
- *IrSyncAcc*, Initial value: 1000 units/s²
- *IrSyncDec*, Initial value: 1000 units/s²

See the parameter structure [L TS2P_scPar_FlyingSawState](#) (21).

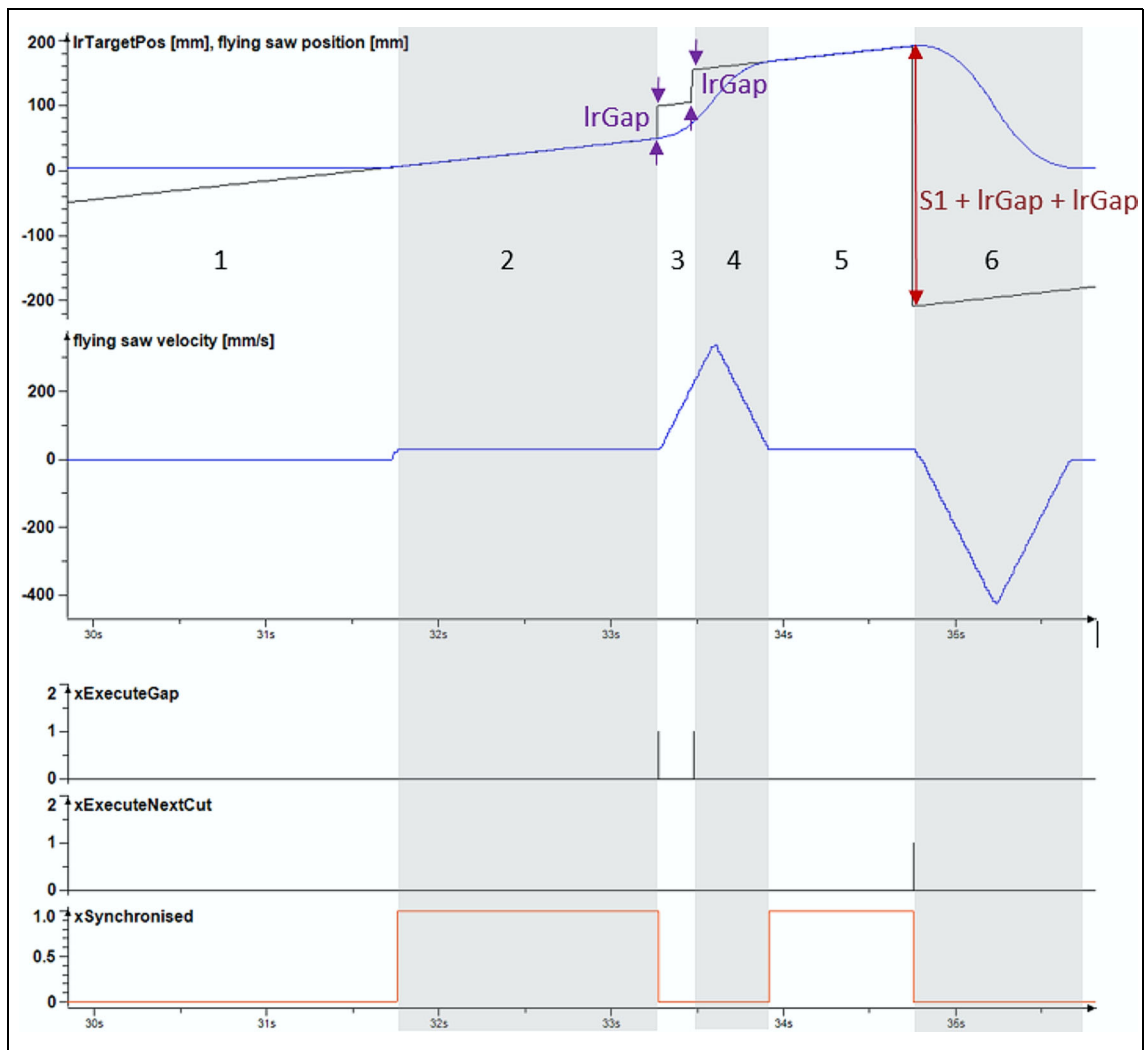
9.6 Separation of cut materials (Inserting a gap)

This function is suitable for:

- Edge protection for sensitive material
- Separation of the cut material

The gap prevents leaving any marks of the saw blade on the cutting edge.

The *xExecuteGap* input can be used to separate the cut material after the cutting process. This function shortly drives the saw carriage oversynchronous (faster) to the material before the saw blade is withdrawn. This creates a gap between the cutting edge and the saw blade, which can be adjusted via the parameter *IrGap*. The input can be activated several times in succession. This increases the gap by a multiple of *IrGap*.



[9-9] Example: Separation of cut material (inserting a gap)

No.	xExecuteGap	xExecuteNextCut	xSynchronised	Description
1	FALSE	FALSE	FALSE	The slave axis is at the left end position <i>lrLowerLimit</i> of the working range. The target position <i>lrTargetPos</i> moves synchronously with the material velocity below the limit <i>lrLowerLimit</i> .
2	FALSE	FALSE	TRUE	The target position <i>lrTargetPos</i> moves into the work area. The slave axis is coupled to the target position <i>lrTargetPos</i> . The cutting process is then performed on the material.
3	FALSE → TRUE	FALSE	FALSE	With the command "Execute gap" the target position <i>lrTargetPos</i> jumps by the gap width <i>lrGap</i> in the positive direction. The slave axis smoothes out the jump of the target position <i>lrTargetPos</i> via the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> .
4	FALSE → TRUE	FALSE	FALSE	An additional "Execute gap" command is requested via the input <i>xExecuteGap</i> . The target position <i>lrTargetPos</i> jumps by the gap width <i>lrGap</i> in the positive direction. The slave axis now smoothes out the jump in the target position and also the total gap that was generated via the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> .
5	FALSE	FALSE	TRUE	After the jump is smoothed out, the slave axis continues to travel synchronously to the target position <i>lrTargetPos</i> .
6	FALSE	FALSE → TRUE	FALSE	With the command "Next cut" (<i>xExecuteNextCut</i>) the target position jumps by the cutting length <i>lrCuttingLength</i> = S1 added with the two gap widths (<i>lrGap</i> + <i>lrGap</i>) in the negative direction. The slave axis smoothes out the jump of the target position <i>lrTargetPos</i> via the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> .

Parameters to be set

- *lrLowerLimit*, Initial value: 0 units
- *lrGap*, Initial value: 50 units
- *lrSyncVel*, Initial value: 360 units/s
- *lrSyncAcc*, Initial value: 1000 units/s²
- *lrSyncDec*, Initial value: 1000 units/s²

See the parameter structure [L_TS2P_scPar_FlyingSawState](#) (p. 21).

9.7 Restore target position

This function is used to restore a target position (output *lrTargetPos*) and can only be executed once after the start of the sawing process (*xStartFlyingSaw* = TRUE).

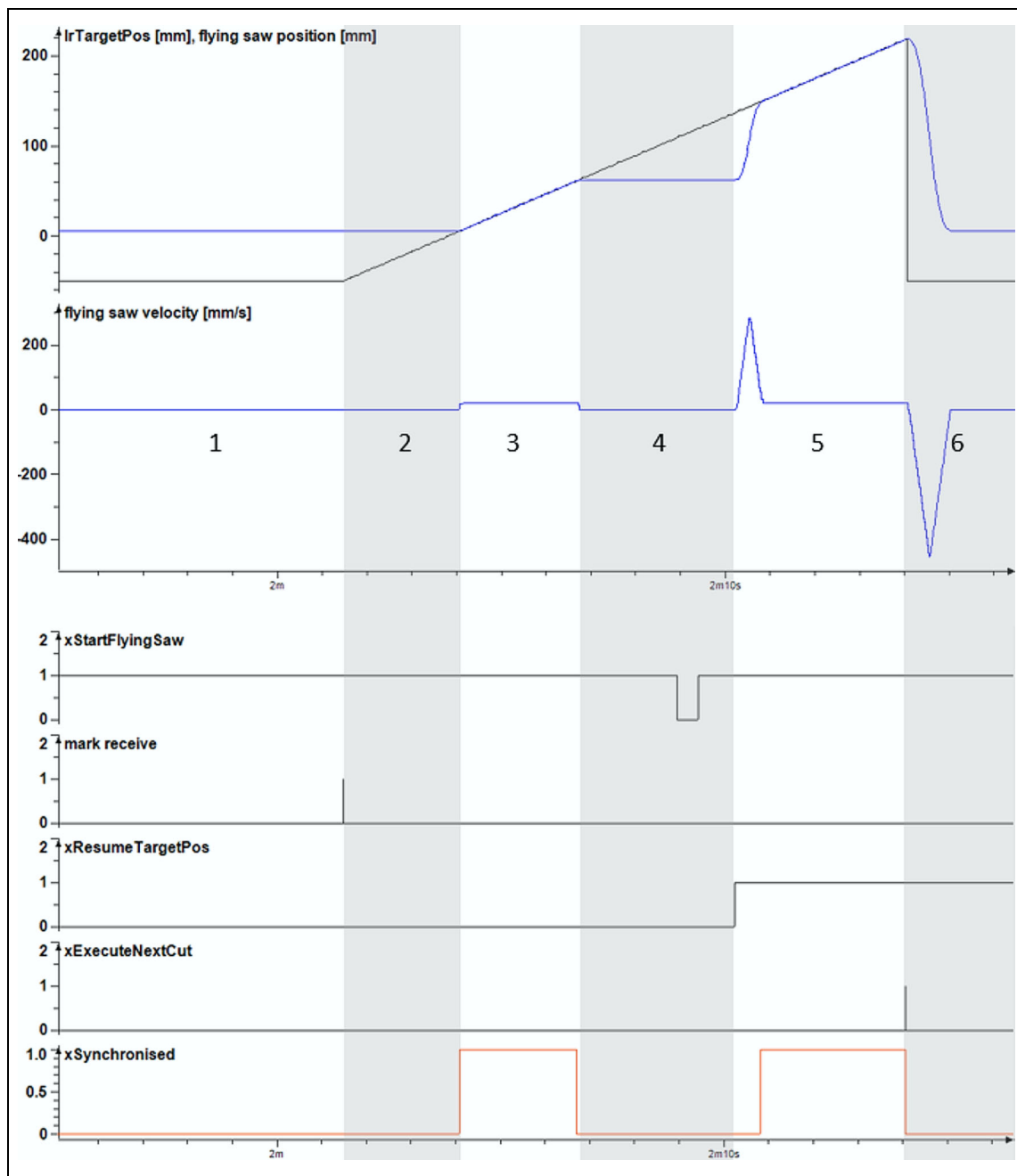
In mark-controlled operation and in length-controlled operation, the restoration of the target position is different. In addition to that, in both the operating modes a check is made whether the requested target position is below the upper travel range limit (parameter *lrUpperLimit*).

9.7.1 Mark-controlled operation

In the mark-controlled operation, only one valid mark can be restored at a time. In this case, the target position (output *lrTargetPos*) must be before the upper movement range limit (parameter *lrUpperLimit*).

If the target position has already exceeded the upper travel range limit, a rising edge (FALSE→TRUE) at the input *xExecuteNextCut* can be used to switch to the next mark in the mark stack. If there is no valid marker in the stack, then no target position is approached.

After a rising edge (FALSE→TRUE) at the input *xResumeTargetPos* the target position is approached again.

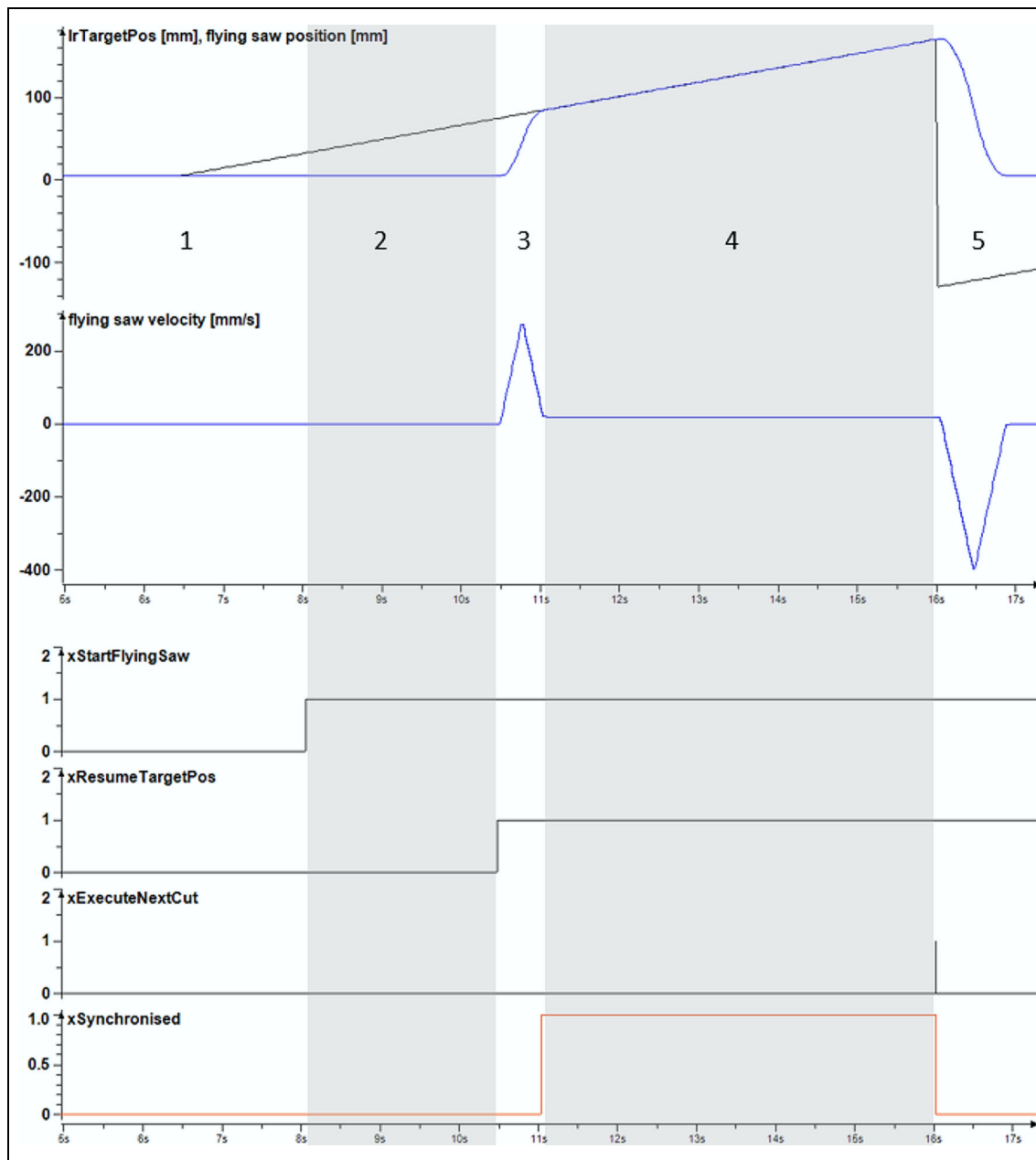


[9-10] Example: Process "Restore target position" in mark-controlled operation

No.	xStartFlyingSaw	xResumeTargetPos	xExecuteNextCut	xSynchronised	Description
1	FALSE	FALSE	FALSE	FALSE	The master axis, and along with the material, is moved. The slave axis is at the left limit <i>lrLowerLimit</i> of the work area.
2	TRUE	FALSE	FALSE	FALSE	A mark is recorded. The target position is set to the sensor position <i>lrMarkSensorPos</i> . The target position <i>lrTargetPos</i> moves synchronously to the material velocity below the limit <i>lrLowerLimit</i> .
3	TRUE	FALSE	FALSE	TRUE	The target position <i>lrTargetPos</i> moves into the work area. The slave axis is synchronized to the target position <i>lrTargetPos</i> .
4	TRUE↗FALSE↗TRUE	FALSE	FALSE	FALSE	The technology module is stopped via the "Halt" or "Stop" function. In the background, the target position continues to run synchronously with the mark. The technology module is switched back to the "SAW_IN_PROCESS" state via the <i>xStartFlyingSaw</i> input.
5	TRUE	TRUE	FALSE	FALSE↗TRUE	The FALSE↗TRUE edge at the input <i>xResumeTargetPos</i> synchronizes the slave axis to the restored target position.
6	TRUE	TRUE	FALSE↗TRUE	FALSE	A FALSE↗TRUE edge at the input <i>xExecuteNextCut</i> normally moves to the next cutting position. Since there is no further mark in the stack, the target position jumps to the sensor position and remains there. The slave axis is positioned to the lower limit of the working range via the profile generator.

9.7.2 Length-controlled operation

After a rising edge (FALSE→TRUE) at the input *xResumeTargetPos* the target position (output *IrTargetPos*) is approached again.



[9-11] Example: Process "Restore target position" in length-controlled operation

No.	xStartFlyingSaw	xResumeTargetPos	xExecuteNextCut	xSynchronised	Description
1	FALSE	FALSE	FALSE	FALSE	After a stop, the master axis, and thus the material, is started again. The target position is also moved along.
2	TRUE	FALSE	FALSE	FALSE	The technology module is switched to the "SAW_IN_PROCESS" state via the <i>xStartFlyingSaw</i> input and the parameter <i>eStartMode</i> = 0.
3	TRUE	TRUE	FALSE	FALSE	The FALSE→TRUE edge at the input <i>xResumeTargetPos</i> synchronizes the slave axis to the restored target position.
4	TRUE	TRUE	FALSE	TRUE	The slave axis moves synchronously to the target position <i>lrTargetPos</i> .
5	TRUE	TRUE	FALSE→TRUE	FALSE	With the command "Next cut" (<i>xExecuteNextCut</i>) the target position jumps by the cutting length <i>lrCuttingLength</i> = S1 added with the two gap widths (<i>lrGap</i> + <i>lrGap</i>) in the negative direction. The slave axis smoothes out the jump of the target position <i>lrTargetPos</i> via the profile generator with the defined parameters <i>lrSyncVel</i> , <i>lrSyncAcc</i> and <i>lrSyncDec</i> .

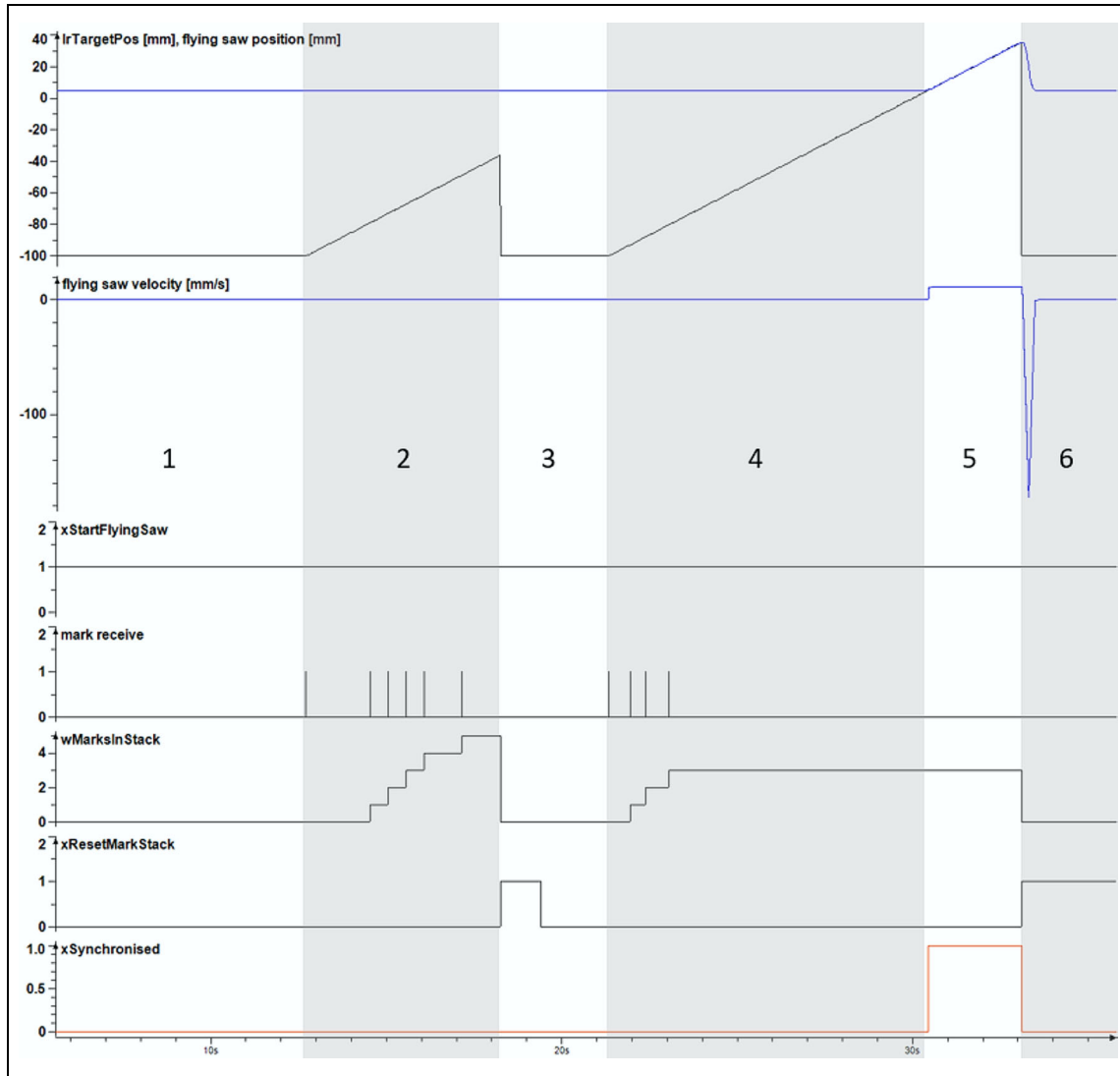
9.8 Delete mark stack

In mark-controlled mode, a rising edge (FALSE→TRUE) at the input *xResetMarkStack* deletes all the stored marks in the mark stack.

The output *byRemainMarkStack* shows how many marks the mark stack can hold.

If the function is executed while the flying saw is in synchronous state (input *xStartFlyingSaw* = TRUE), the carriage moves to the waiting position (parameter *IrLowerLimit*) and is synchronized with the next mark.

Both the possibilities are shown in the time diagram in Figure [9-12].



[9-12] Example: Process "Delete mark stack"

No.	mark receive	xResetMarkStack	xSynchronised	Description
1	FALSE	FALSE	FALSE	The master axis, and thus the material, is moved. The technology module is in the "SAW_IN_PROCESS" state.
2	4 x FALSE → TRUE	FALSE	FALSE	Four marks are recorded. The first mark is followed via the target position <i>lrTargetPos</i> . The remaining targets are loaded into the stack and tracked. The output <i>wMarksInStack</i> shows the number of stored marks. The target position <i>lrTargetPos</i> moves synchronously to the material velocity below the limit <i>lrLowerLimit</i> .
3	FALSE	TRUE	FALSE	If the input <i>xResetMarkStack</i> = TRUE, the mark stack is deleted and <i>wMarksInStack</i> = 0 is set. The entire mark stack is then freely available again for a maximum of 100 marks. The target position <i>lrTargetPos</i> is reset and is therefore no longer coupled to the material velocity. The slave axis remains at the lower limit of the working range.
4	3 x FALSE → TRUE	FALSE	FALSE	Four marks are recorded. The first mark is followed via the target position <i>lrTargetPos</i> . The remaining targets are loaded into the stack and tracked. The output <i>wMarksInStack</i> shows the number of stored marks. The target position <i>lrTargetPos</i> moves synchronously to the material velocity below the limit <i>lrLowerLimit</i> .
5	FALSE	FALSE	TRUE	The target position <i>lrTargetPos</i> enters the working range of the flying saw. This synchronizes the slave position to the target position. The saw carriage is synchronized with the mark.
6	FALSE	TRUE	FALSE	If the input <i>xResetMarkStack</i> = TRUE, the mark stack is deleted and <i>wMarksInStack</i> = 0 is set. The entire mark stack is then freely available again. The target position <i>lrTargetPos</i> is reset and is therefore no longer coupled to the material velocity. The slave axis remains at the lower limit of the working range.
7	FALSE	TRUE	FALSE	The slave axis waits for a valid mark.

9.9 Move to parking position

There are two ways to travel to the parking position:

A. The flying saw is not active.

The technology module is in the "READY" state.

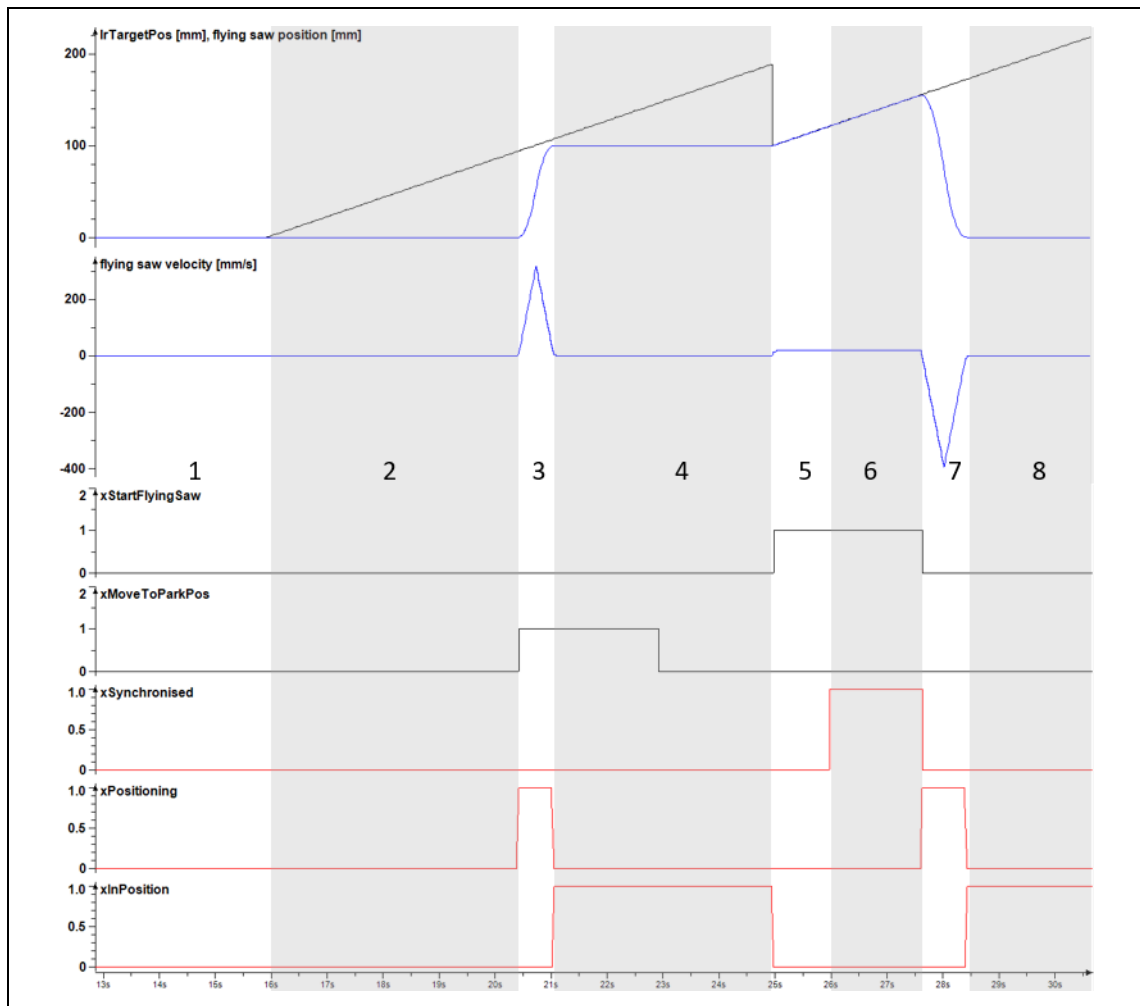
A rising edge (FALSE→TRUE) at the input *xMoveToParkPos* changes the state from "READY" to "MoveToParkPos" and the saw carriage is travelled to the parking position set in the parameter *lrParkPosition*.

B. The flying saw is switched to active with the input *xStartFlyingSaw* = TRUE. The mode *eFlyingSawStopMode* = 1 (move to parking position) is set.

If the input *xStartFlyingSaw* is then reset from TRUE to FALSE, the carriage moves to the parking position set in the parameter *lrParkPosition*.

During the positioning the output *xPositioning* = TRUE.

If the parking position is reached, the outputs *xInPosition* = TRUE and *xPositioning* = FALSE are set.



[9-13] Example: Process "Move to parking position"

No.	xStartFlyingSaw	xMoveToParkPos	Status	Description
1	FALSE	FALSE	xSynchronised = FALSE xPositioning = FALSE xInPosition = FALSE	The slave axis is at the left end position of the working area <i>lrLowerLimit</i> .
2	FALSE	FALSE	xSynchronised = FALSE xPositioning = FALSE xInPosition = FALSE	The material is moved. The slave axis is still at the left end position of the working area <i>lrLowerLimit</i> .
3	FALSE	TRUE	xSynchronised = FALSE xPositioning = TRUE xInPosition = FALSE	The slave axis is moved to the parking position <i>lrParkPosition</i> = 100.
4	FALSE	FALSE	xSynchronised = FALSE xPositioning = FALSE xInPosition = TRUE	The slave axis has reached the parking position <i>lrParkPosition</i> = 100.
5	TRUE	FALSE	xSynchronised = FALSE xPositioning = FALSE xInPosition = FALSE	The state of the technology module changes to "SAW_IN_PROCESS". The target position <i>lrTargetPos</i> is travelled along with the master position. The target position <i>lrTargetPos</i> moves into the work area. The slave axis is synchronized to the target position <i>lrTargetPos</i> .
6	TRUE	FALSE	xSynchronised = TRUE xPositioning = FALSE xInPosition = FALSE	The slave axis is synchronized into the target position <i>lrTargetPos</i> . The cutting process is then performed on the material.
7	FALSE	FALSE	xSynchronised = FALSE xPositioning = TRUE xInPosition = FALSE	With <i>xStartFlyingSaw</i> = FALSE the cutting process is terminated. With the parameter <i>eFlyingSawStopMode</i> = 1 (MoveToParkPos) the slave axis moves to the parking position <i>lrParkPosition</i> = 100.
8	FALSE	FALSE	xSynchronised = FALSE xPositioning = FALSE xInPosition = TRUE	The slave axis has reached the parking position <i>lrParkPosition</i> = 100.

Parameters to be set

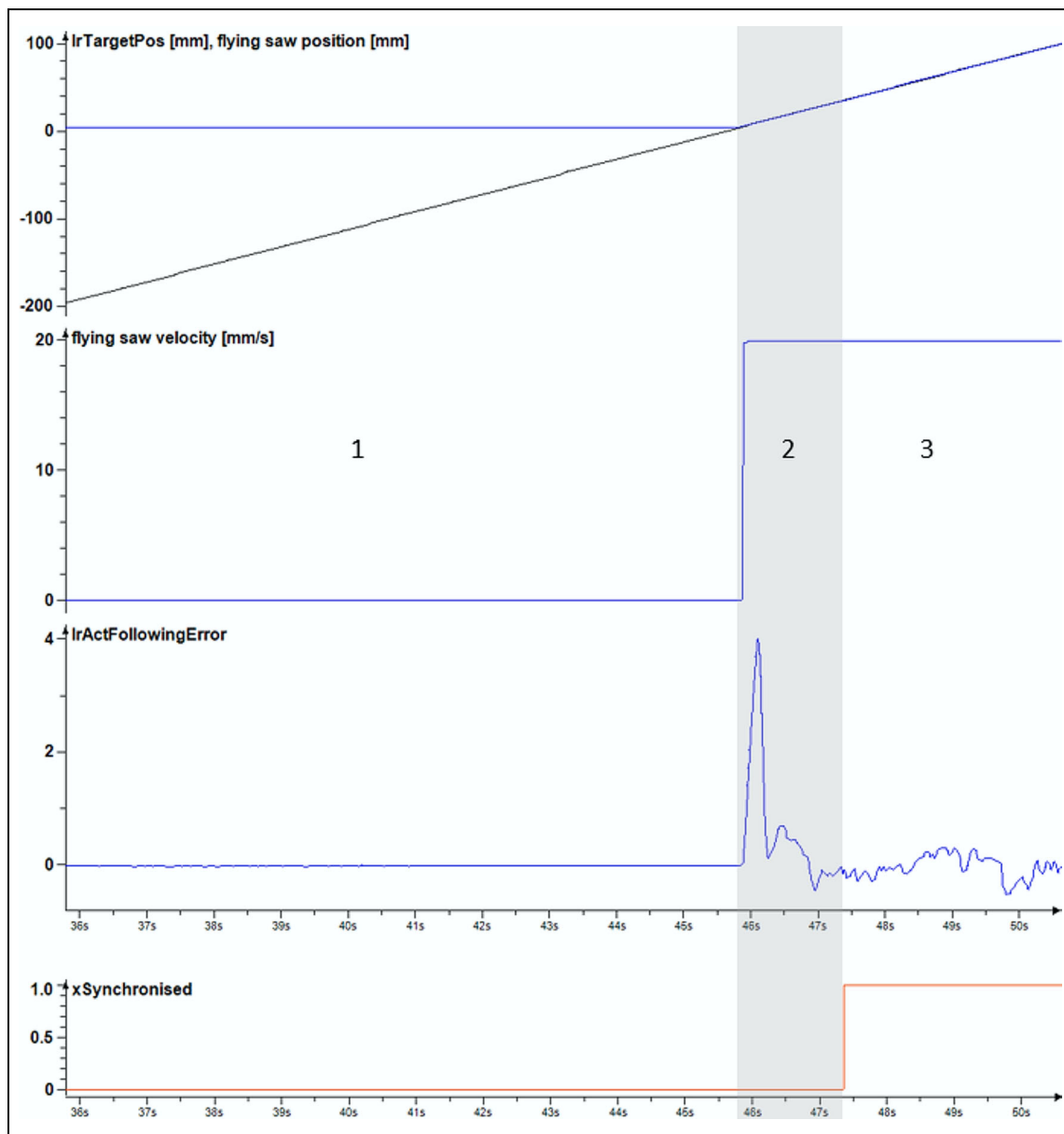
- *eFlyingSawMode*, Initial value: 0 (mark-controlled operation)
- *eFlyingSawStopMode*, Initial value: 0 (Halt)
- *lrParkPosition*, Initial value: 0 units

See the parameter structure [L_TS2P_scPar_FlyingSawState](#) (21).

9.10 Window for following error monitoring

This function checks whether the following error is within a defined window within a certain time. It is activated with the parameter `xPosInWindow = TRUE`.

The window and the time are set via the parameters `lrPosInWindow` and `lrTimePosInWindow` in the parameter structure [L TS2P_scPar_FlyingSawState](#) (21). As soon as the two reference values of the master and slave axis are synchronous and the following error is within the set window, the output `xSynchronised = TRUE` is set.



[9-14] Example: Process of the following error monitoring

No.	xSynchronised	Description
1	FALSE	The master axis, and thus the material, is moved. The technology module is in the "SAW_IN_PROCESS" state. The target position <i>lrTargetPos</i> moves synchronously to the material velocity below the limit <i>lrLowerLimit</i> .
2	FALSE	The target position <i>lrTargetPos</i> enters the working range of the flying saw. This synchronizes the slave position to the target position. The saw carriage is synchronized with the mark. The following error is corrected.
3	TRUE	The following error has remained within the parameterized window <i>lrTolerancePosInWindow</i> at least for the time <i>lrDwellTimePosInWindow</i> . This sets the output <i>xSynchronised</i> = TRUE.

Parameters to be set

- *eFlyingSawMode*, Initial value: 0 (mark-controlled operation)
- *xPosInWindow*, Initial value: 0
- *lrTolerancePosInWindow*, Initial value: 0.5 units
- *lrDwellTimePosInWindow*, Initial value: 50 ms

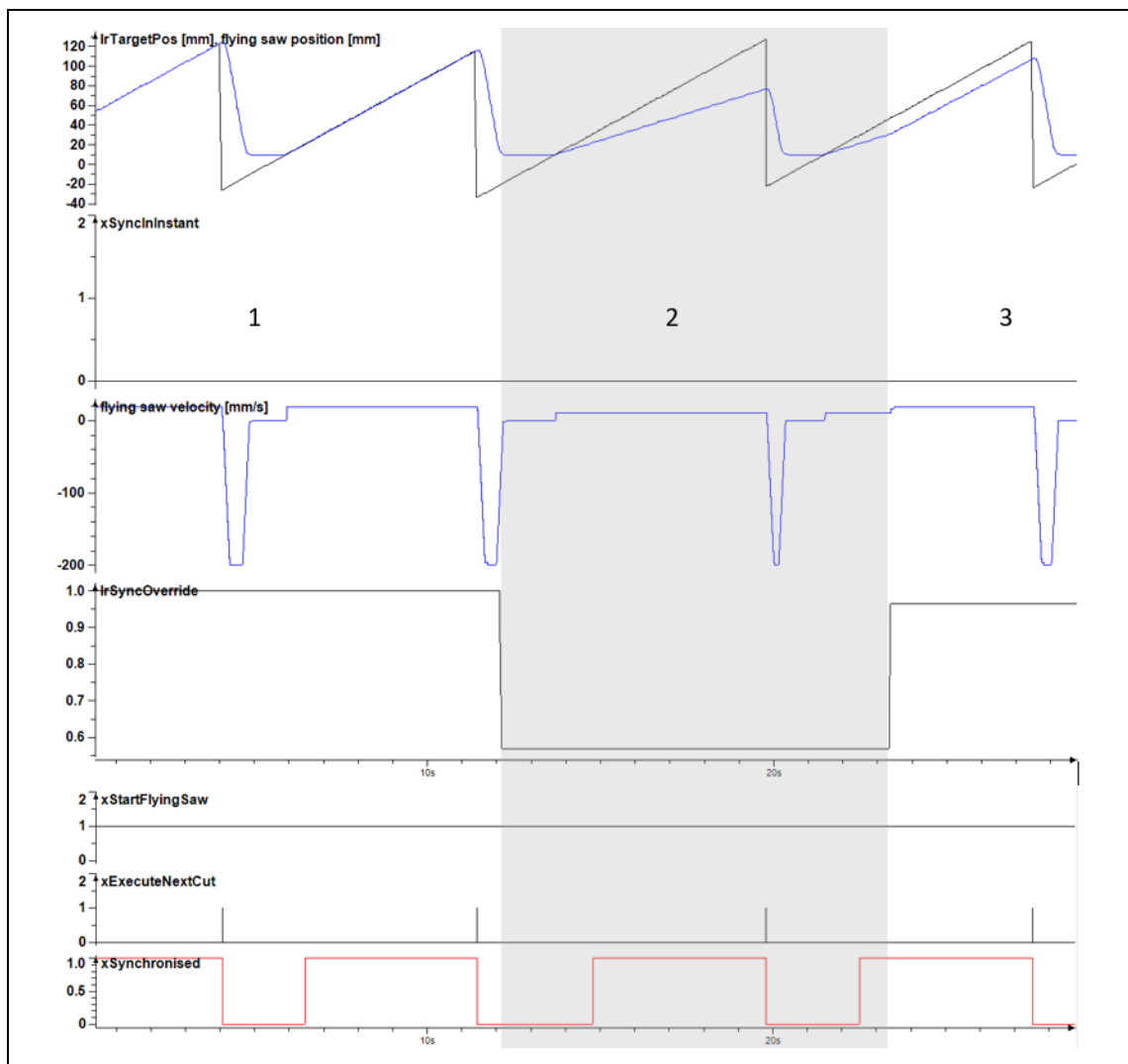
See the parameter structure [L_TS2P_scPar_FlyingSawState](#) (21).

9.11 Override in the synchronous phase

In certain applications the movement in the cutting phase must be slower or faster to the material velocity by an adjustable factor. The velocity can be reduced or increased via the *IrSyncOverride* input. Here a scaled value related to the current material velocity is expected. The factor has a multiplying effect on the material velocity. For example, if the synchronous velocity of the flying saw is to be reduced by 20 %, input *IrSyncOverride* = 0.8 must be set. Configuring with the value *IrSyncOverride* = 1.5 results in an increase of the synchronous velocity by 50 %. The valid value range of the input *IrSyncOverride* is from 0.5 to 3.0.

A change of the value is immediately effective in the synchronous phase. If the value is changed within the synchronous phase, the resulting speed is controlled via the ramp generator with the parameters *IrSyncVel*, *IrSyncAcc*, *IrSyncDec* and *IrSyncJerk*.

Calculation of the velocity of the flying saw with override		
$V_{FS} = F \cdot V_{CV}$		
Formula symbol	Description	Unit
V_{FS}	Velocity of the flying saw	units/s
V_{CV}	Velocity of the conveyor belt (material velocity)	units/s
F	Factor <i>IrSyncOverride</i> = 1.0: The flying saw moves synchronously with the velocity of the conveyor belt. > 1.0: The Flying Saw moves faster than the conveyor belt. < 1.0: The Flying Saw moves slower than the conveyor belt.	



[9-15] Example: Process "Override in the synchronous phase"

No.	<i>IrSyncOverride</i>	Description
1	1.00	The technology module is in the "SAW_IN_PROCESS" state. The slave axis moves synchronously to the target position <i>IrTargetPos</i>. With the command "Next cut" (<i>xExecuteNextCut</i>) the target position jumps by the cutting length <i>IrCuttingLength</i> in the negative direction. The slave axis smoothes out the jump of the target position <i>IrTargetPos</i> via the profile generator with the defined parameters <i>IrSyncVel</i>, <i>IrSyncAcc</i> and <i>IrSyncDec</i>.
2	0.55	When returning or in the waiting position (left end position of the working area <i>IrLowerLimit</i>) the override for the synchronous range <i>IrSyncOverride</i> = 0.55 is set. The target position <i>IrTargetPos</i> moves into the work area. The slave axis is coupled to the target position <i>IrTargetPos</i>. When synchronous travel is reached, the slave position is moved more slowly because the override factor. Furthermore, the status <i>xSynchronised</i> = TRUE is set. With the command "Next cut" (<i>xExecuteNextCut</i>) the target position jumps by the cutting length <i>IrCuttingLength</i> in the negative direction. The override has no influence on the target position <i>IrTargetPos</i> during return travel or in the coupling process of the slave axis.
3	0.90	In the synchronous phase the override for the synchronous range <i>IrSyncOverride</i> = 0.9. The slave axis moves synchronously to the target position <i>IrTargetPos</i>. The status <i>xSynchronised</i> = TRUE remains unchanged. The slave axis moves via the ramp generator with the newly set synchronous velocity, taking the new override value into account. For starting with the new resulting velocity the defined parameters <i>IrSyncVel</i>, <i>IrSyncAcc</i> and <i>IrSyncDec</i> are set for the profile generator.

Parameters to be set

- *IrSyncVel*, Initial value: 360 units/s
- *IrSyncAcc*, Initial value: 1000 units/s²
- *IrSyncDec*, Initial value: 1000 units/s²
- *IrSyncJerk*, Initial value: 1000000 units/s³

See the parameter structure [L TS2P_scPar_FlyingSawState](#) (21).

A

Aborting a positioning process [45](#)
Absolute positioning [37](#)
Application notes [7](#)

C

Competing a positioning process [45](#)
Continuous travel with a defined velocity [42](#)
Conventions used [6](#)
Cross cutter state signal flow plan [26](#)
Cutting on position marks (mark-controlled operation) [57](#)
Cutting relative product lengths (length-controlled operation) [61](#)

D

Delete mark stack [75](#)
Design of the safety instructions [7](#)
Document history [6](#)

E

eFlyingSawMode = 0 [51](#)
eFlyingSawMode = 1 [53](#)
eFlyingSawMode = 2 [55](#)
E-mail to Lenze [85](#)
eTMState [25](#)

F

Feedback to Lenze [85](#)
Flying Saw V2 (Functional description) [10](#)
Following error monitoring [79](#)
Function block L_TS2P_FlyingSawState [15](#)
Functional description "Flying Saw (V2)" [10](#)
Functions of the technology module (overview) [12](#)

H

Halt, [36](#)
Homing [32](#)

I

Inputs [16](#)
Inserting a gap (Separation of cut materials) [68](#)

L

L_TB2P_States [25](#)
L_TS2P_FlyingSawState [15](#)
L_TS2P_scAP_FlyingSawState [28](#)
L_TS2P_scPar_FlyingSawState [21](#)
L_TS2P_scSF_FlyingSawState [27](#)
Length-controlled operation (cutting relative product lengths) [61](#)

M

Manual jog (jogging) [30](#)
Mark-controlled operation (cutting on position marks) [57](#)

Motion profiles (linear/S ramps) [49](#)
Move to parking position [77](#)

N

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

O

Operating mode [13](#)
Optimized dynamics [50](#)
Outputs [19](#)
Override for speed [47](#)
Override in the synchronous phase [81](#)

P

Parameter structure L_TS2P_scPar_FlyingSawState [21](#)
Parameters TM "Flying Saw (V2)" [21](#)

Q

QSP application [35](#)

R

Relative positioning [39](#)
Restore target position [70](#)

S

Safety instructions [8](#)
Separation of cut materials (Inserting a gap) [68](#)
Speed override [47](#)
Start of the axes [13](#)
State Machine [24](#)
States [24](#)
States of the output eTMState [25](#)
Structure of the access points "Flying Saw (V2)" [28](#)
Structure of the L_TS2P_scSF_CrossCutterState signal flow [27](#)
Synchronization of the saw carriage to the material velocity [51](#)

T

Target group [5](#)

V

Variable names [6](#)

W

Window for following error monitoring [79](#)



Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

If you have suggestions for improvement, please e-mail us to:

feedback-docu@lenze.com

Thank you for your support.

Your Lenze documentation team

