

Technology module



Speed Control

Reference manual

EN

Contents

1	About this document	3
1.1	Document history	5
1.2	Conventions used	6
1.3	Layout of the safety instructions	7
2	Safety instructions	8
3	Overview of the technology module functions	10
3.1	Functional overview of the TM	10
3.2	Important notes on how to operate the technology module	11
4	Application ranges	12
5	L_TF2P_SpeedControlBase function block	13
5.1	inputs	14
5.2	outputs	16
5.3	Parameter	18
6	State machine	19
7	Signal flow diagram	20
7.1	Structure of the signal flow	21
7.2	Structure of the access points	21
8	Description of the basic functions	22
8.1	Manual jog (jogging)	23
8.2	Homing	25
8.3	QSP application	28
8.4	Halt	29
8.5	Absolute positioning	30
8.6	Relative positioning	32
8.7	Continuous travel with a defined velocity	35
8.8	Aborting and completing a positioning process	38
8.9	Override for speed	40
9	Description of the technology function	42
9.1	Starting and stopping speed operation	43
9.2	Speed operation with preset setpoints	45
9.2.1	Travel profiles	45
9.2.2	Profile switch-over	48
9.3	Inverting a setpoint	50
9.4	External travel profile parameters	52
9.5	Engage speed offset	55
	Index	57
	Your opinion is important to us	58

1 About this document

This documentation ...

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

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

Further sets of technical documentation with regard to Lenze products

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

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

Symbols:

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



Tip!

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

www.lenze.com

Target group

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

1 About this document

1.1 Document history

1.1 Document history

Version			Description
4.0	04/2021	TD06	Error corrections & supplements
3.0	04/2020	TD17	Error corrections
2.0	12/2019	TD06	Error corrections & supplements
1.0	09/2019	TD06	First edition

1 About this document

1.2 Conventions used

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

Type of information	Writing	Examples/notes
Numeric notation		
Decimal separator	Point	The decimal point is always used. For example: 1234.56
Text		
Program name	» «	»PLC Designer« ...
Variable identifiers	<i>Italics</i>	By setting <i>bEnable</i> to TRUE...
Function blocks	Bold	The L_MC1P_AxisBasicControl function block ...
Function libraries		The L_TT1P_TechnologyModules function library ...
Source code	Font "Courier new"	... dwNumerator := 1; dwDenominator := 1; ...
Symbols		
Page reference	 6	Reference to further information: Page number in PDF file.

Variable identifiers

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

1 About this document

1.3 Layout of the safety instructions

DANGER!

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

WARNING!

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

CAUTION!

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

NOTICE

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

Application notes

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

2 Safety instructions

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



The device documentation contains safety instructions which must be observed!

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

DANGER!

High electrical voltage

Injury to persons caused by dangerous electrical voltage

Possible consequences

Death or severe injuries

Protective measures

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

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

Observe the corresponding information plates on the device.

DANGER!

Injury to persons

Risk of injury is caused by ...

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

Possible consequences

Death or severe injuries

Protective measures

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

NOTICE

Damage or destruction of machine parts

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

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

Protective measures

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

3

Overview of the technology module functions

3.1

Functional overview of the TM

Apart from the basic functions of the L_TB2P_BasicMotionBase technology module, the technology module also offers the following functionalities to move an axis:

- Speed operation
- Speed operation with up to 16 speed profiles
- Setpoint inversion
- Speed offset
- Overrides for speed, acceleration and jerk

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



Tip!

The basic functions of the technology module are also described in these instructions.

► [Description of the basic functions](#) (📖 22)

3.2

Important notes on how to operate the technology module

Start of the axes

The start of the axes after a locked controller depends on the type of movement commissioned.

A differentiation is made between the movement type with a level-activated function and the movement type with an edge-activated function.

- The movement type with a level-activated function is activated automatically after the controller release as long as no higher priority function is active. This movement type includes
 - [Manual jog \(jogging\)](#) (📖 23)
 - [QSP application](#) (📖 28)
- The movement type with an edge-activated function must be activated after the controller release with a FALSE→TRUE-edge. This movement type includes
 - [Homing](#) (📖 25)



Example of a movement type with a level-activated function: manual mode (jogging)

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



Example of a movement type with an edge-activated function: reference run (homing)

1. In the inhibited axis state ($xOperationEnabled = FALSE$), $xHomeExecute$ is set to TRUE.
 - $xEnableOperation = FALSE$ (axis is inhibited.)
==> **STANDBY** state ($xEnableOperation = FALSE$)
 - $xHomeExecute = TRUE$ (referencing is to be executed.)
2. Enable axis.
 - $xEnableOperation = TRUE$
==> **STANDBY** state (output $xOperationEnabled = TRUE$)
3. Perform homing.
 - $xHomeExecute = FALSE \rightarrow TRUE$
==> **HOMING** state

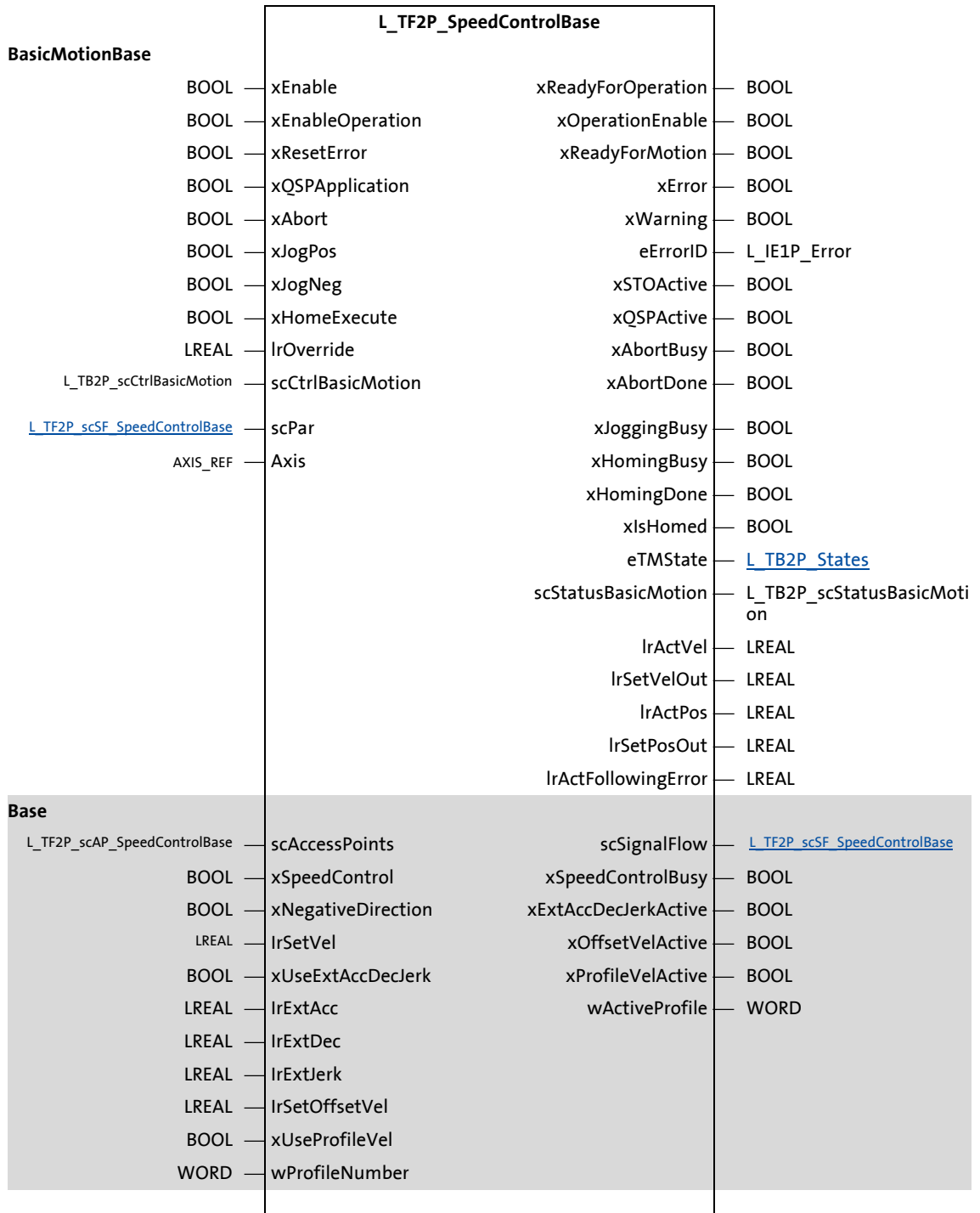
4 Application ranges

General servo drive for the speed control of

- Conveying drives (in a group)
- Extruders
- Test stands
- Vibrators
- Propulsion drives
- Presses
- Processing machines
- Dosers

5 L_TF2P_SpeedControlBase function block

The illustration shows that the inputs and outputs are part of the function block.



5.1

inputs

Identifier	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 the "ready for operation" state. 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 The reset error or software error is cleared.
		TRUE The reset error or software error is set.
xQSPApplication	BOOL	The input is level-controlled and evaluates the high level.
		TRUE The active movement is cancelled and the axis is brought to a standstill via the QSP application stop ramp. • The state changes to "Stop". • The technology module remains in the "STOP" status as long as <i>xQSPApplication</i> remains TRUE.
xAbort	BOOL	TRUE The active movement is cancelled and the axis is brought to a standstill via the active profile deceleration. • The state changes to "Stop". The deceleration ramp can be aborted by commissioning of a new movement (e.g. <i>xMoveVelExecute</i>)
		FALSE The active movement is not cancelled.
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 <i>eErrorID</i> output displays the InvalidInputJogPosJogNeg error message 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 <i>eErrorID</i> output displays the InvalidInputJogPosJogNeg error message and the axis stops.
xHomeExecute	BOOL	The input is edge-controlled and evaluates the rising edge.
		FALSE The homing process is not started. TRUE The function is aborted via the <i>scCtrlBasicMotion.xHalt</i> structure input.
lrOverride	LREAL	With the <i>lrOverride</i> factor, the defined 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: Acceptance of the value is carried out irrespective of the status in <i>xContinuousUpdate</i>
scCtrlBasicMotion L_TB2P_scCtrlBasicMotion		Structure input for controlling basic motion functions • See description for the L_TB2P_BasicMotionBase technology module
scPar L_TB2P_scPar_SpeedControlBase		The parameter structure contains the parameters of the technology module.
Axis AXIS_REF		Reference to the axis
scAccessPoints L_TB2P_scAP_SpeedControlBase		Structure of the access points Via the access points, the signal flow or the functions of the technology module can be influenced.
xSpeedControl	BOOL	The input is level-controlled and evaluates the high level.
		TRUE Enables speed operation. Resetting stops the axis. Alternatively, the speed operation can also be aborted via <i>xHalt</i> .
xNegativeDirection	BOOL	TRUE Inversion of the direction of rotation Note: inversion does not apply for jogging!
		FALSE The direction of rotation is not inverted.

Identifier	Data type	Description
lrSetVel	LREAL	Setpoint speed • Unit: units/s
xUseExtAccDecJerk	BOOL	TRUE Accept acceleration deceleration values and jerk from the <i>lrExtAcc</i> , <i>lrExtDec</i> and <i>lrExtJerk</i> inputs.
lrExtAcc	LREAL	External acceleration value • Unit: units/s ²
lrExtDec	LREAL	External deceleration value • Unit: units/s ²
lrExtJerk	LREAL	External jerk value • Unit: units/s ³
lrSetOffsetVel	LREAL	Speed offset • Unit: units/s Note: Acceptance of the value is carried out irrespective of the status in <i>xContinuousUpdate</i> . The value does not influence the jogging speed!
xUseProfileVel	BOOL	TRUE The speed setpoint from the profile data is used. The value at the <i>lrSetVel</i> input is ignored.
wProfileNumber	WORD	Selection of the active travel profile • Initial value: 0 (profile 0) 16 different travel profiles (0 - 15) can be selected: 0: profile 0 1: profile 1 ... 15: profile 15

5.2

outputs

Identifier	Data type	Description
xReadyForOperation	BOOL	The axis is faultless, the DC-bus voltage is available. STO is not active.
xOperationEnabled	BOOL	The axis is enabled. The brake is released.
xReadyForMotion	BOOL	The axis is enabled and faultless. The brake is released.
xEError	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>xEError</i> = TRUE or <i>xWarning</i> = TRUE. "FAST technology modules" reference manual: Here you can find information on error or warning messages.
xSTOActive	BOOL	TRUE A Safe Torque Off (STO) has been activated and is active. The axis is torqueless, the controller is inhibited.
xQSPActive	BOOL	TRUE The <i>xQSPApplication</i> input has been 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 a standstill.
xAAbortBusy	BOOL	TRUE The function activated is aborted by the <i>xAAbort</i> input.
xAAbortDone	BOOL	TRUE The function activated has been aborted by the <i>xAAbort</i> input.
xJoggingBusy	BOOL	TRUE The manual jog function is active. The technology module is in the JOGGING state.
xHomingBusy	BOOL	TRUE Homing is carried out. The technology module is in the HOMING state.
xHomingDone	BOOL	TRUE Homing has been completed successfully.
xIsHomed	BOOL	TRUE The axis has been referenced ("Reference known"). <i>xIsHomed</i> must 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_scStatusBasicMotionL_TB2P_scStatusBasicMotion	Structure of the status data of the function block
IrActVel	LREAL	Current speed • Unit: unit / s
IrSetVelOut	LREAL	Setpoint speed • Unit: unit / s
IrActPos	LREAL	Current position • Unit: unit
IrSetPosOut	LREAL	Setpoint position • Unit: unit
IrActFollowingError	LREAL	Current following error • Unit: unit
scSignalFlow	L_TF2P_scSF_SpeedControlBase	Structure of the signal flow
SpeedControlBusy	BOOL	TRUE The speed operation is released and active.

Identifier	Data type	Description	
xExtAccDecJerkActive	BOOL	TRUE	The <i>xUseExtAccDecJerk</i> input is TRUE. The acceleration and deceleration values are accepted by the <i>lrExtAcc</i> , <i>lrExtDec</i> and <i>lrExtJerk</i> inputs.
xOffsetVelActive	BOOL	TRUE	The offset set at the <i>lrSetOffsetVel</i> input is non-zero.
xProfileVelActive	BOOL	TRUE	The setpoint speed is <u>not</u> accepted by the <i>lrSetVel</i> input, but from the profile data.
wActiveProfile	WORD	Number of the current speed profile 0: profile 0 1: profile 1 ... 15: profile 15	

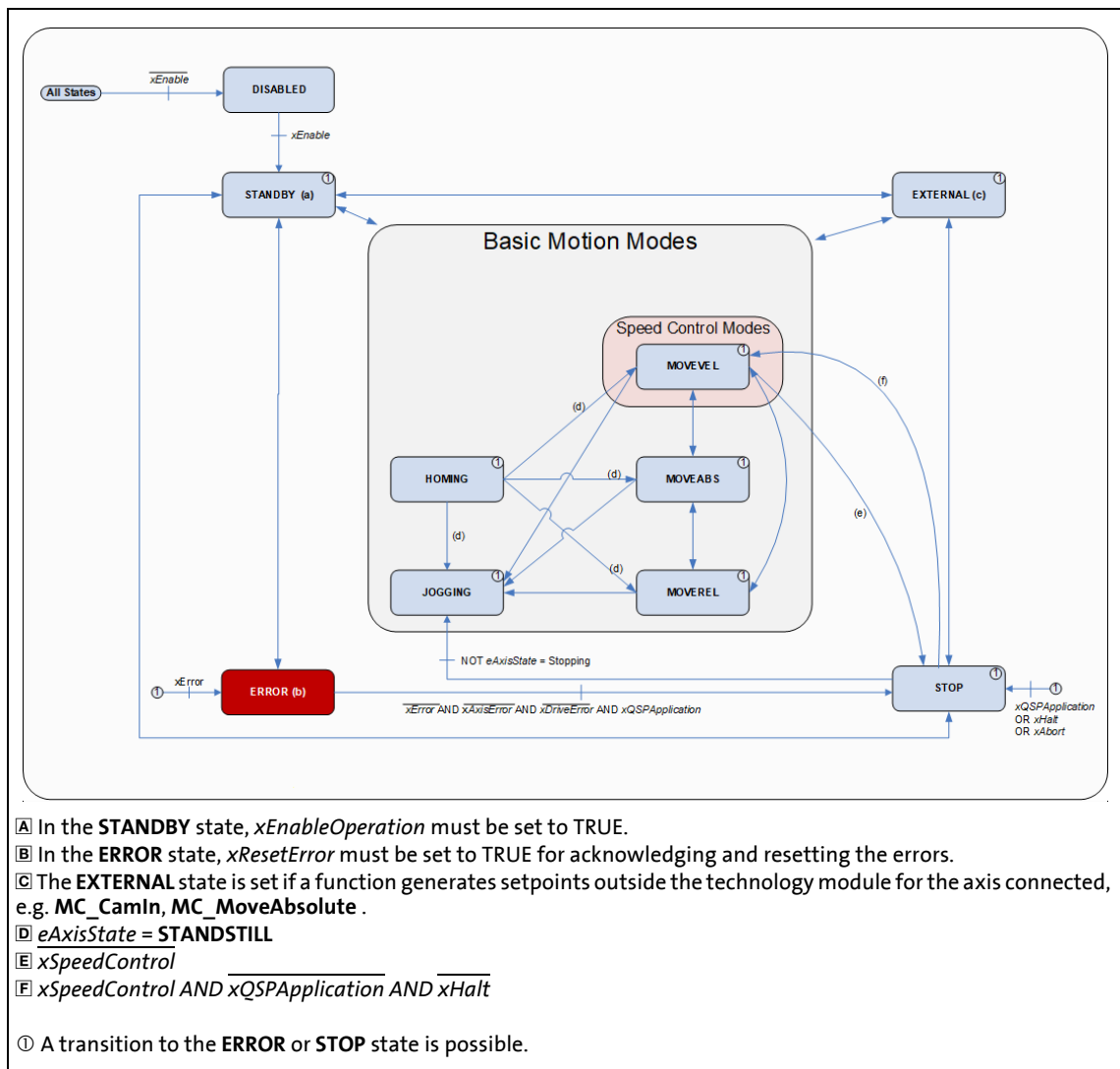
5.3

Parameter

L_TF2P_scPar_SpeedControlBase

Identifier	Data type	Description
IrVel	LREAL	Speed for movements that are commissioned via the scCtrlBasicMotion structure. - Unit: units/s - Initial value: 3600
IrAcc	LREAL	Acceleration for movements that are commissioned via the scCtrlBasicMotion structure. - Selection of the value for the maximum acceleration speed variation. - Unit: units/s² - Initial value: 36000
IrDec	LREAL	Deceleration for movements that are commissioned via the scCtrlBasicMotion structure. - Selection of the value for the maximum deceleration speed variation until standstill. - Unit: units/s² - Initial value: 36000
IrJerk	LREAL	Jerk for movements that are commissioned via the scCtrlBasicMotion structure. - Unit: units/s³ - Initial value: 0
eDirection	MC_DIRECTION	Traversing direction: <i>mcCurrentDirection</i>: Retain current direction (only in case of modulo axis) <i>mcPositiveDirection</i>: Positive direction <i>mcNegativeDirection</i>: Negative direction <i>mcShortestWay</i>: Shortest distance (only in case of modulo axis) - Initial value: <i>mcPositiveDirection</i>
scHomeExtTP	MC_TRIGGER_REF	Transfer of an external touch probe event: - Only relevant for "external encoder" touch probe configuration. - For describing the MC_TRIGGER_REF structure, see the MC_TouchProbe function block.
ascProfiles	ARRAY[0 - 15] OF L_TF2P_scVelocityProfile	Field with 16 travel profile datasets. Each data structure contains all of the data to generate a speed profile (speed, acceleration, deceleration, jerk).

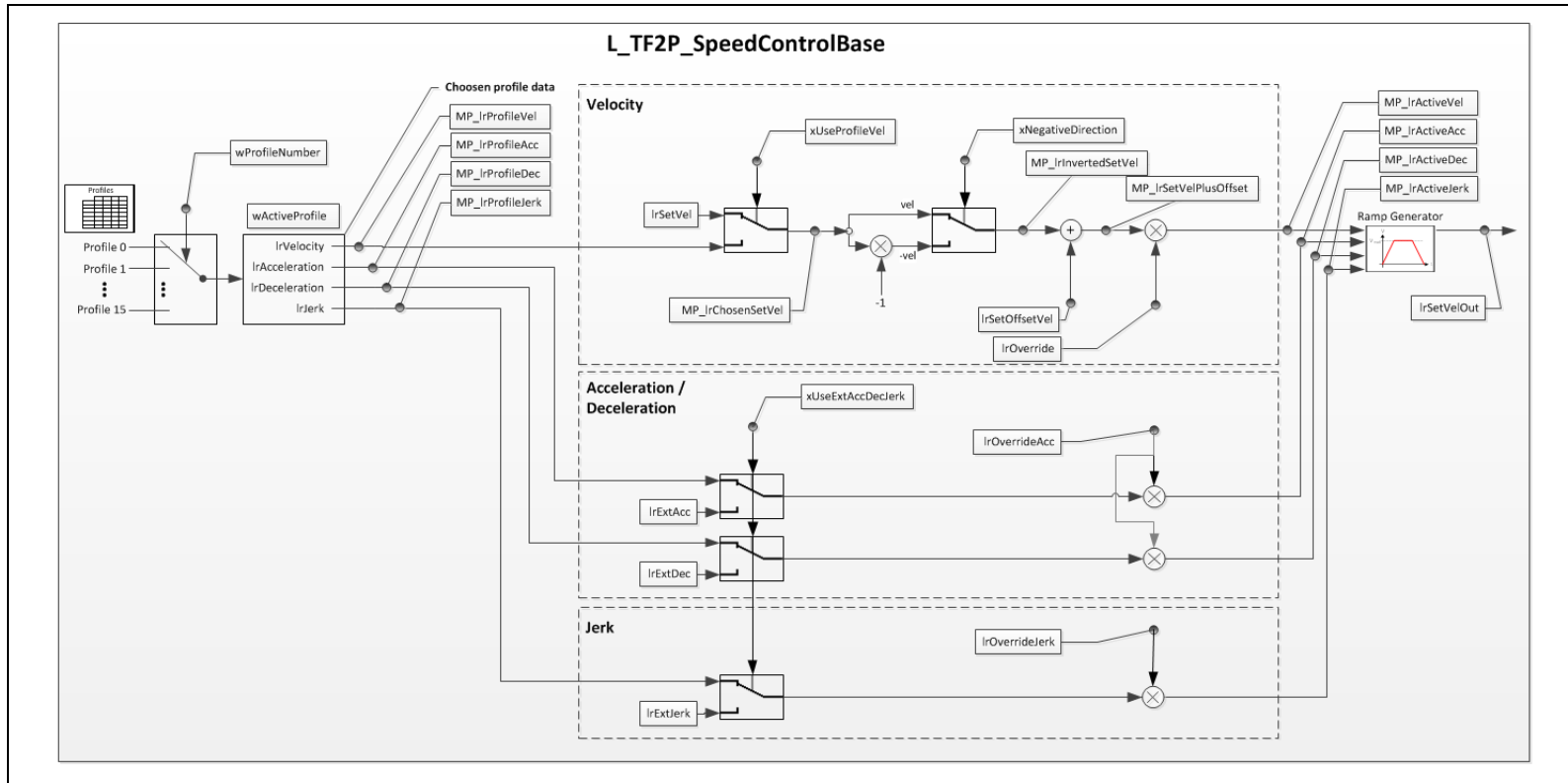
6



[6-1] State machine of the technology module

Signal flow diagram

In the following illustration, the main signal flow of the function is represented:



[7-1] Signal flow of the technology module

7 Signal flow diagram

7.1 Structure of the signal flow

L_TF2P_scSF_SpeedControlBase

The content of the structure is read-only and offers a practical diagnostics option within the signal flow.

► [Signal flow diagram](#) (📖 20)

Identifier	Data type	Description
MP_IrProfileVel	LREAL	Speed from chosen profile
MP_IrProfileAcc	LREAL	Acceleration from chosen profile
MP_IrProfileDec	LREAL	Deceleration from chosen profile
MP_IrProfileJerk	LREAL	Jerk from chosen profile
MP_IrChosenSetVel	LREAL	Selected setpoint speed
MP_IrInvertedSetVel	LREAL	Setpoint speed after negation point
MP_IrSetVelPlusOffset	LREAL	Setpoint speed incl. speed offset
MP_IrActiveVel	LREAL	Actual speed used
MP_IrActiveAcc	LREAL	Actual acceleration used
MP_IrActiveDec	LREAL	Actual deceleration used
MP_IrActiveJerk	LREAL	Actual jerk used

7.2 Structure of the access points



Tip!

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

8 Description of the basic functions

Overview of the basic functions

- [Manual jog \(jogging\)](#) (📖 23)
- [Homing](#) (📖 25)
- [QSP application](#) (📖 28)
- [Halt](#) (📖 29)
- [Absolute positioning](#) (📖 30)
- [Relative positioning](#) (📖 32)
- [Continuous travel with a defined velocity](#) (📖 35)
- [Aborting and completing a positioning process](#) (📖 38)
- [Override for speed](#) (📖 40)

8 Description of the basic functions

8.1 Manual jog (jogging)

8.1 Manual jog (jogging)

Prerequisite

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

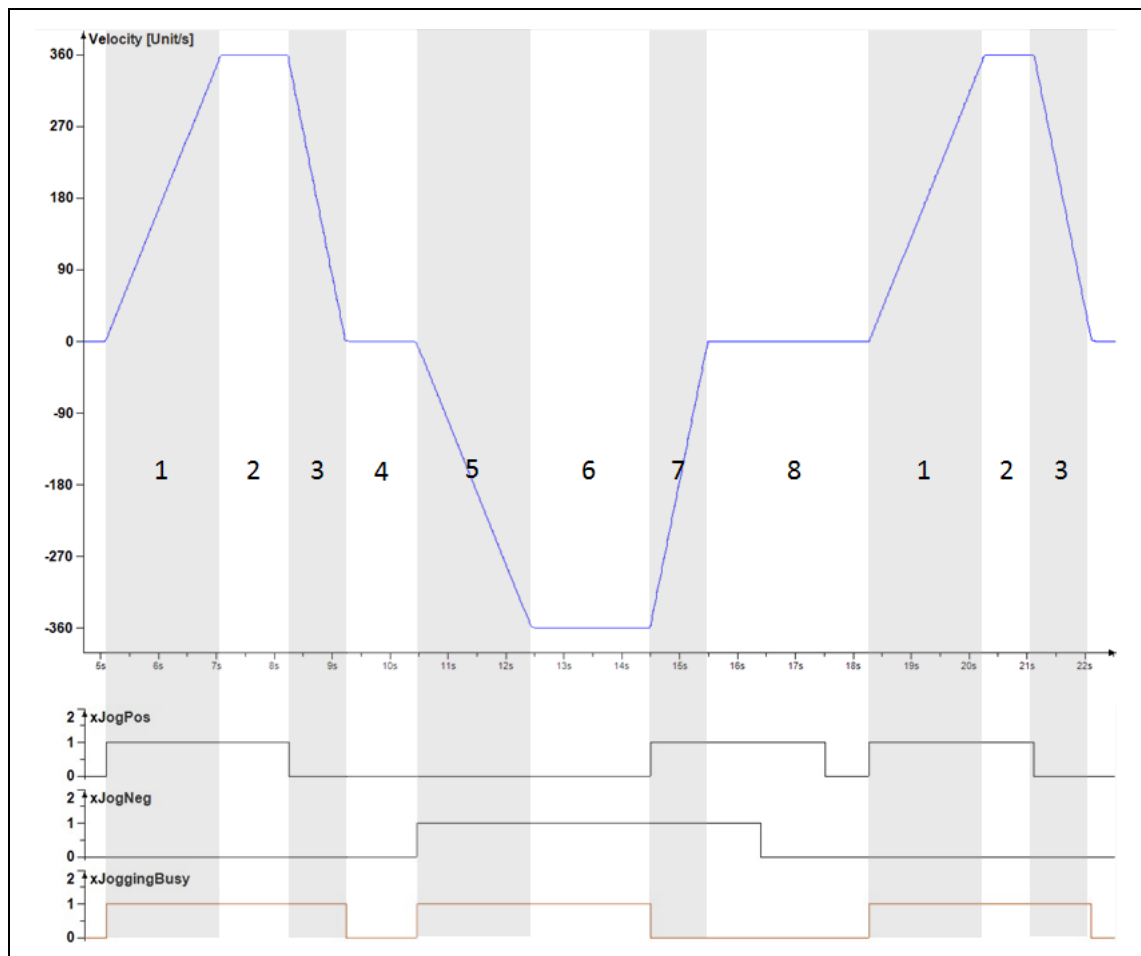
Execution

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.

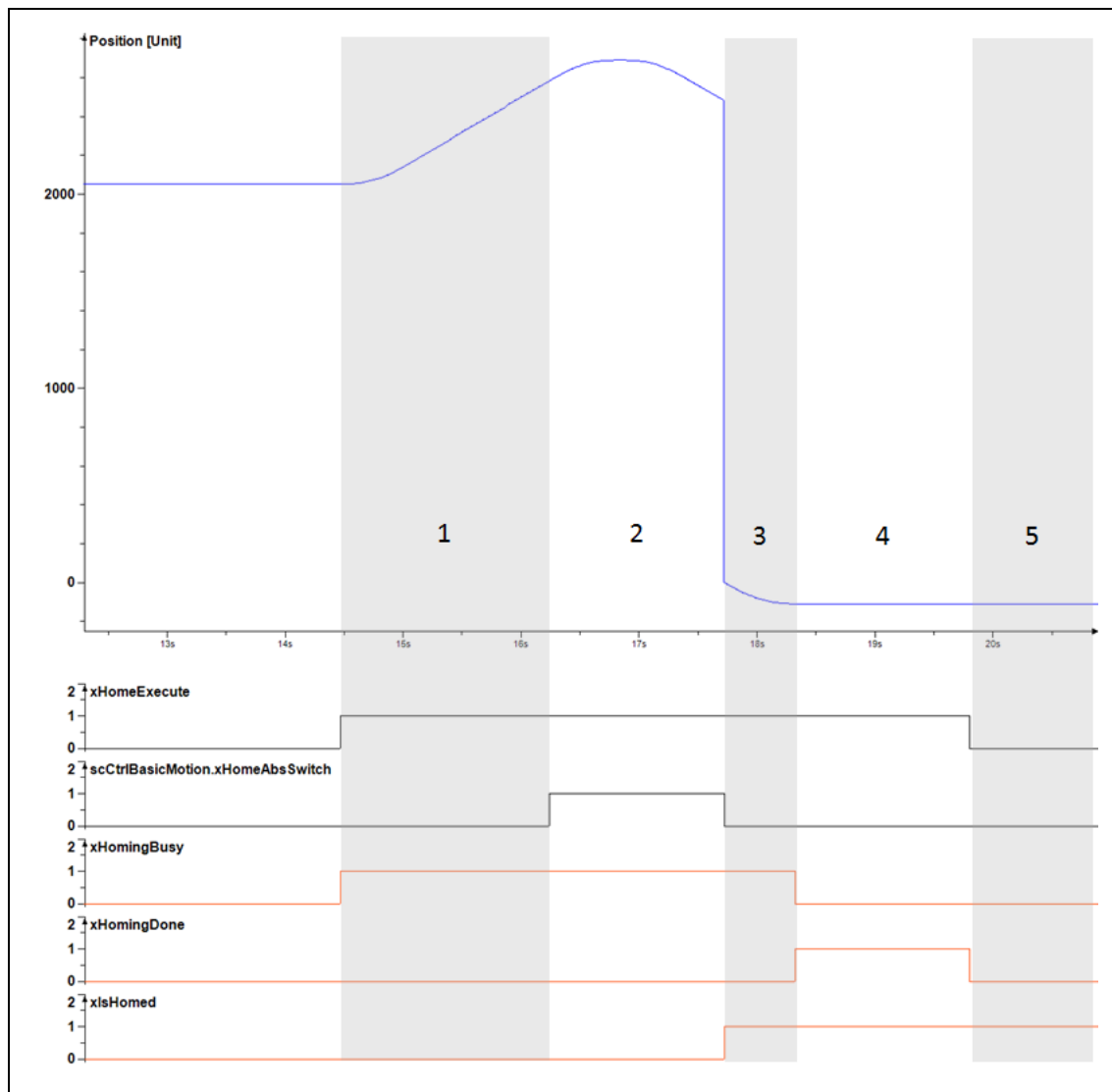
If *xHomeExecute* = FALSE, the state is changed to "STANDBY" and *xHomingDone* is set = FALSE.

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

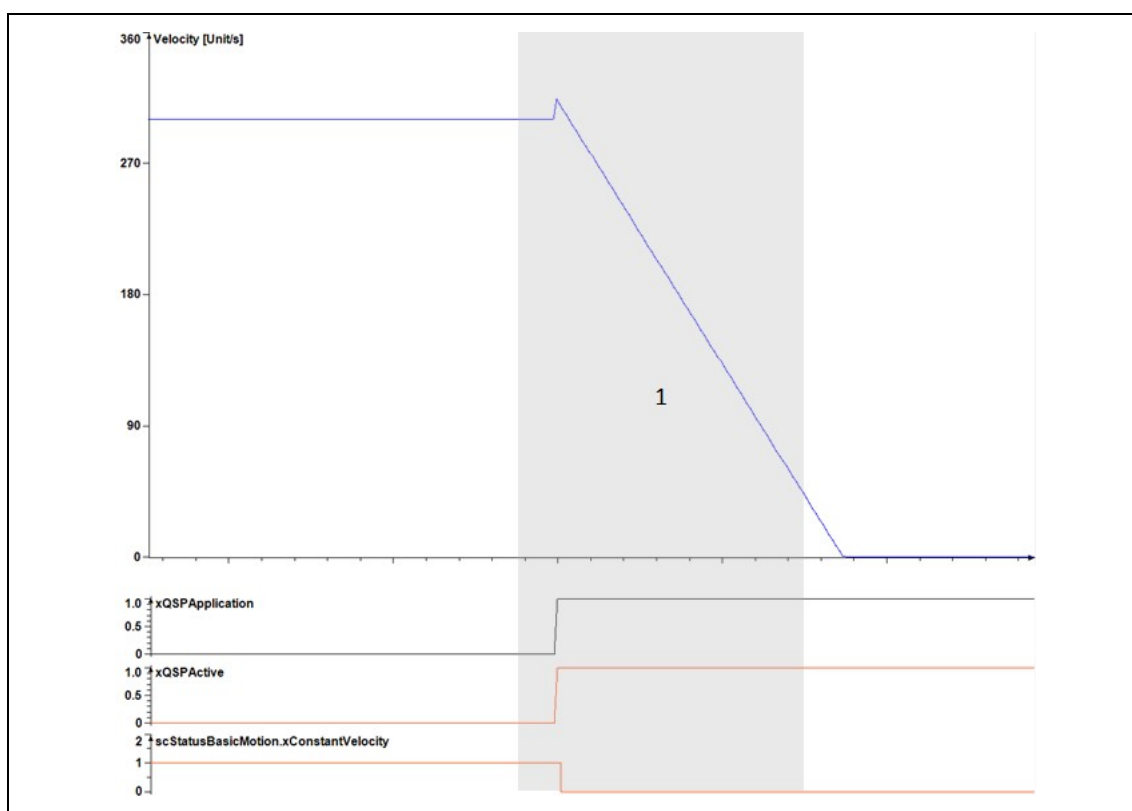
Prerequisite

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

Execution

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

Travel diagram



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

Parameters to be set

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

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

8.4 Halt

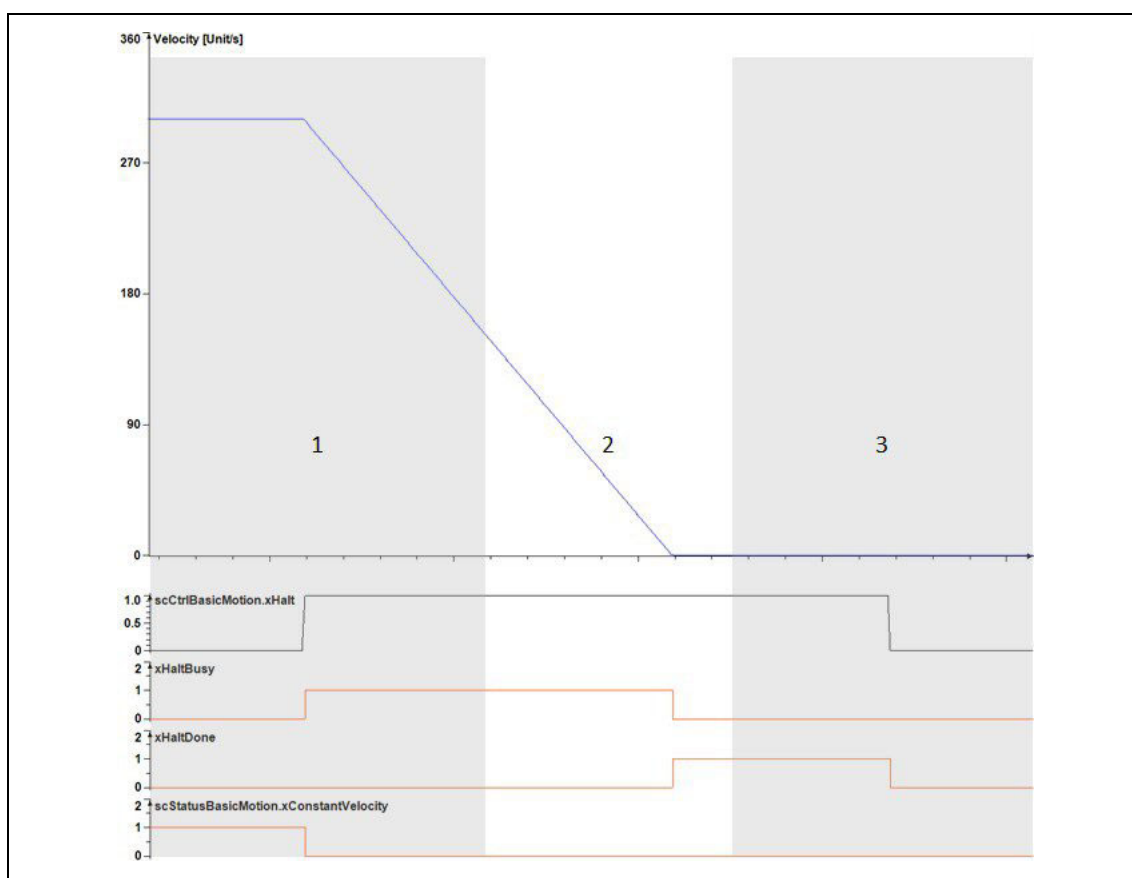
Prerequisite

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

Execution

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

Travel diagram



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

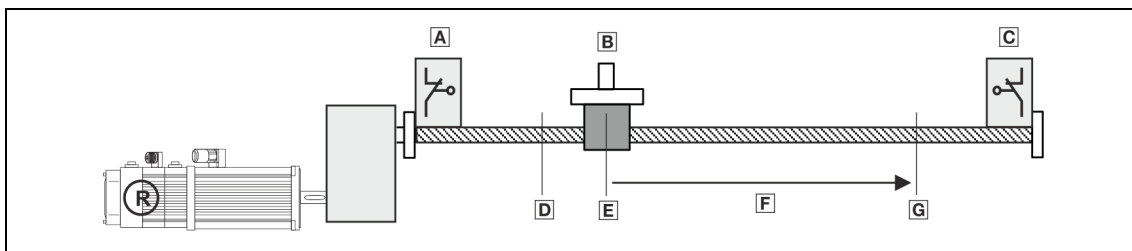
Parameters to be set

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

8 Description of the basic functions

8.5 Absolute positioning

8.5 Absolute positioning



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

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

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

Prerequisites

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

Execution

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

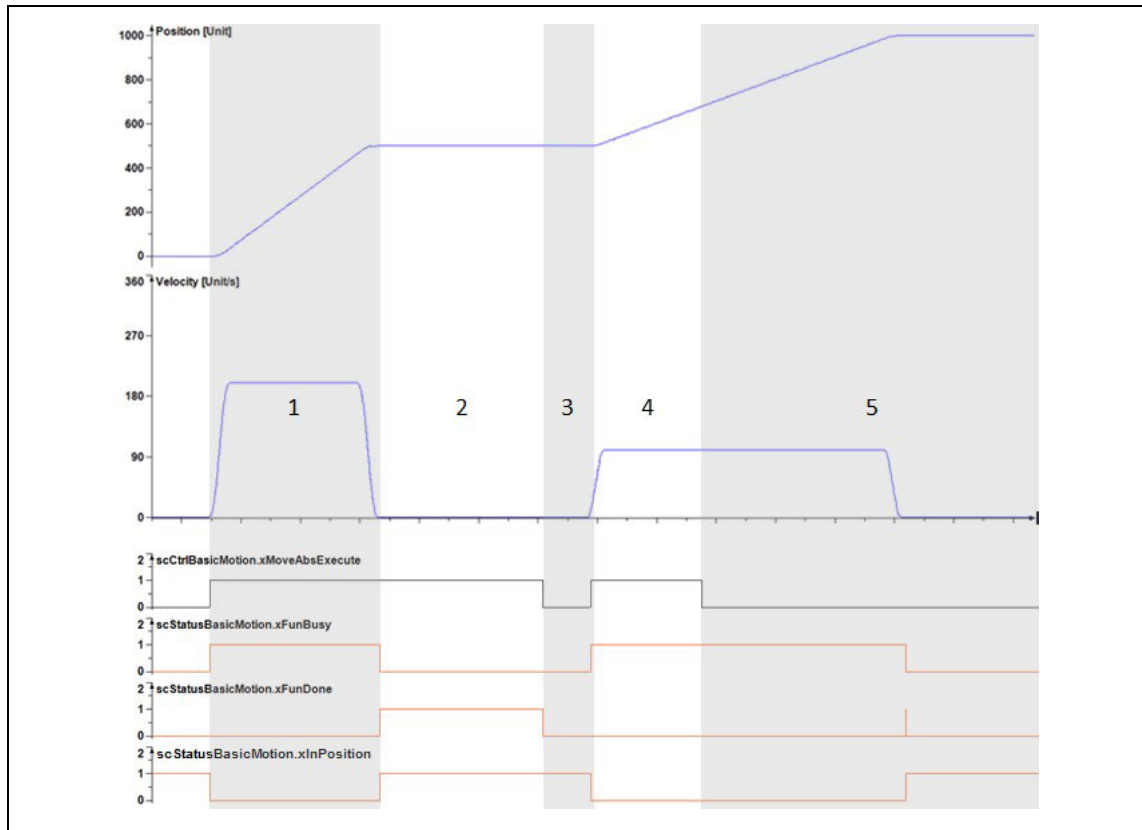
In order to reach the target position, the axis is brought to a standstill via the *lrDec* parameter. The axis reaches a standstill exactly in the target position.

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

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

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

Travel diagram



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

Parameters to be set

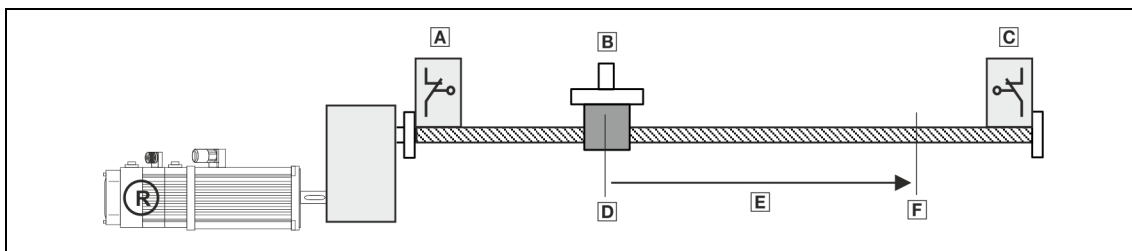
- *IrVel*, initial value: 3600 [units/s]
- *IrAcc*, initial value: 36000 [units/s²]
- *IrDcc*, initial value: 36000 [units/s²]
- *IrJerk*, initial value: 0 [units/s³]

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

8 Description of the basic functions

8.6 Relative positioning

8.6 Relative positioning



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

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

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

Prerequisite

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

Execution

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

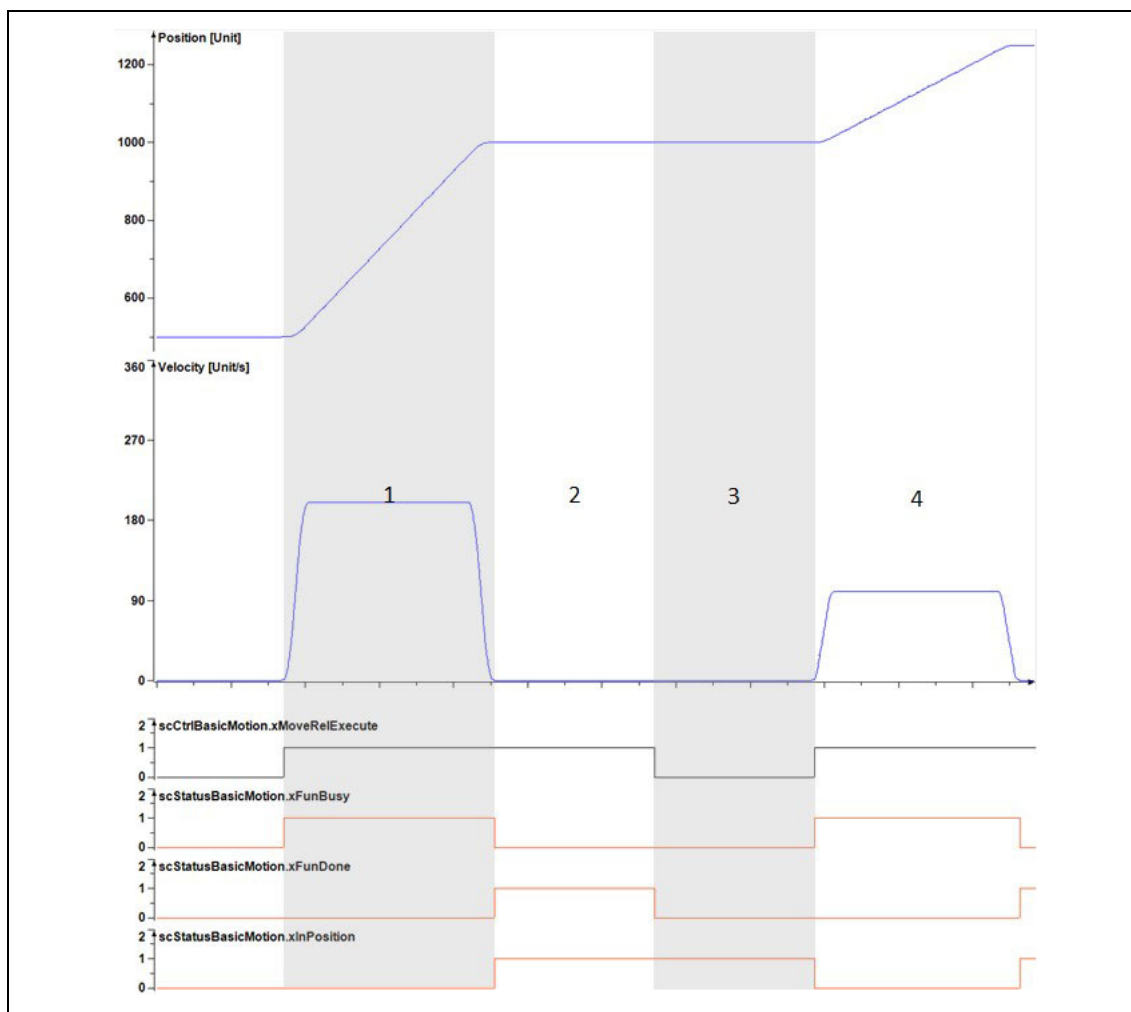
In order to reach the target position, the axis is brought to a standstill via the *lrDec* parameter. The axis reaches a standstill exactly in the target position.

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

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

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

Travel diagram



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

Parameters to be set

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

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

8.7 Continuous travel with a defined velocity

Prerequisite

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

Execution

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

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

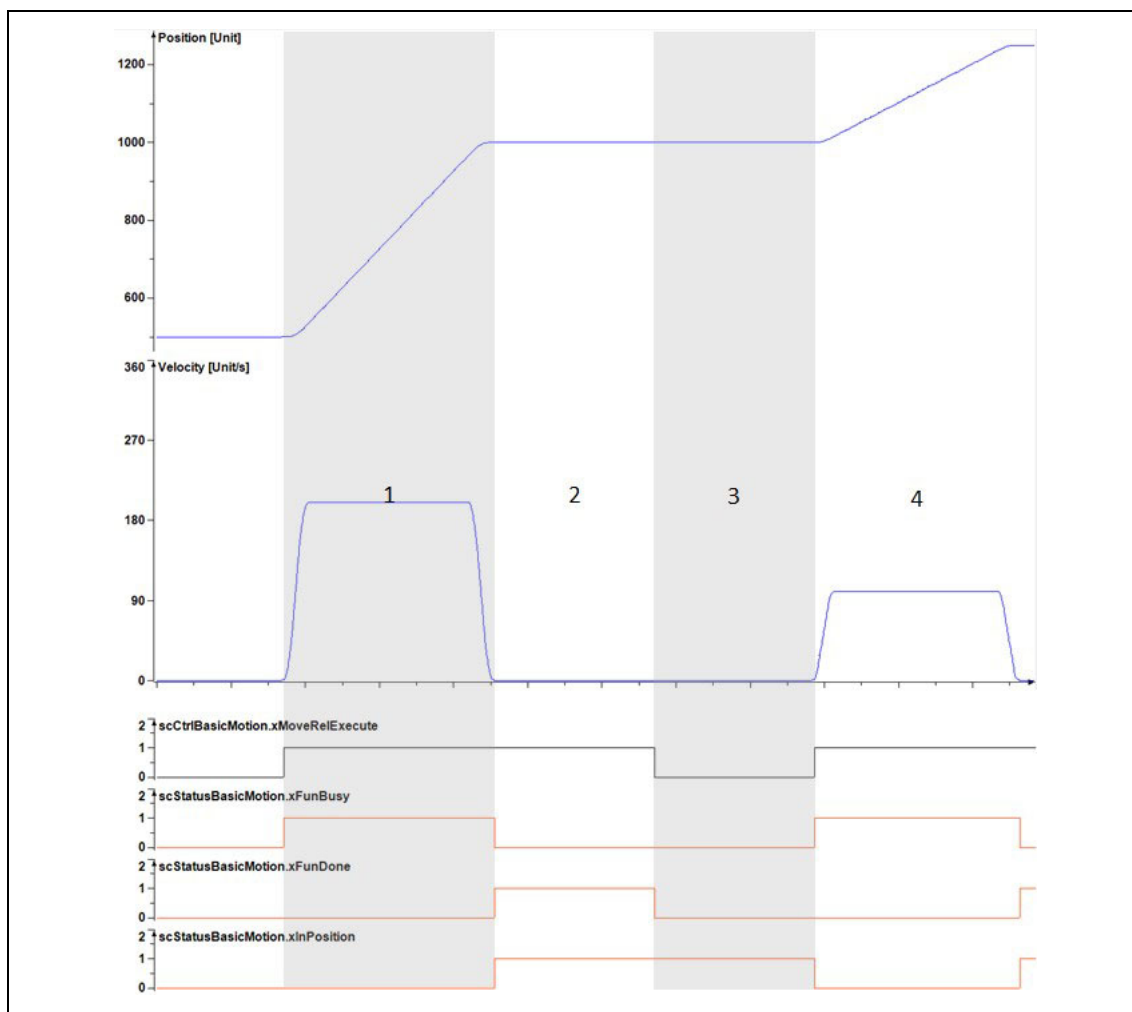
- Input *xDisableSWLimit* = FALSE
- Output *xSwLimitEnabled* = TRUE
- Output *xSwLimitSwitchPos/Neg* = TRUE
- Output *xHwLimitSwitchPos/Neg* = TRUE

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

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

Then the state is changed back to "STANDBY".

Travel diagram



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

Parameters to be set

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

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

8.8 Aborting and completing a positioning process

Prerequisite

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

Execution

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

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

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

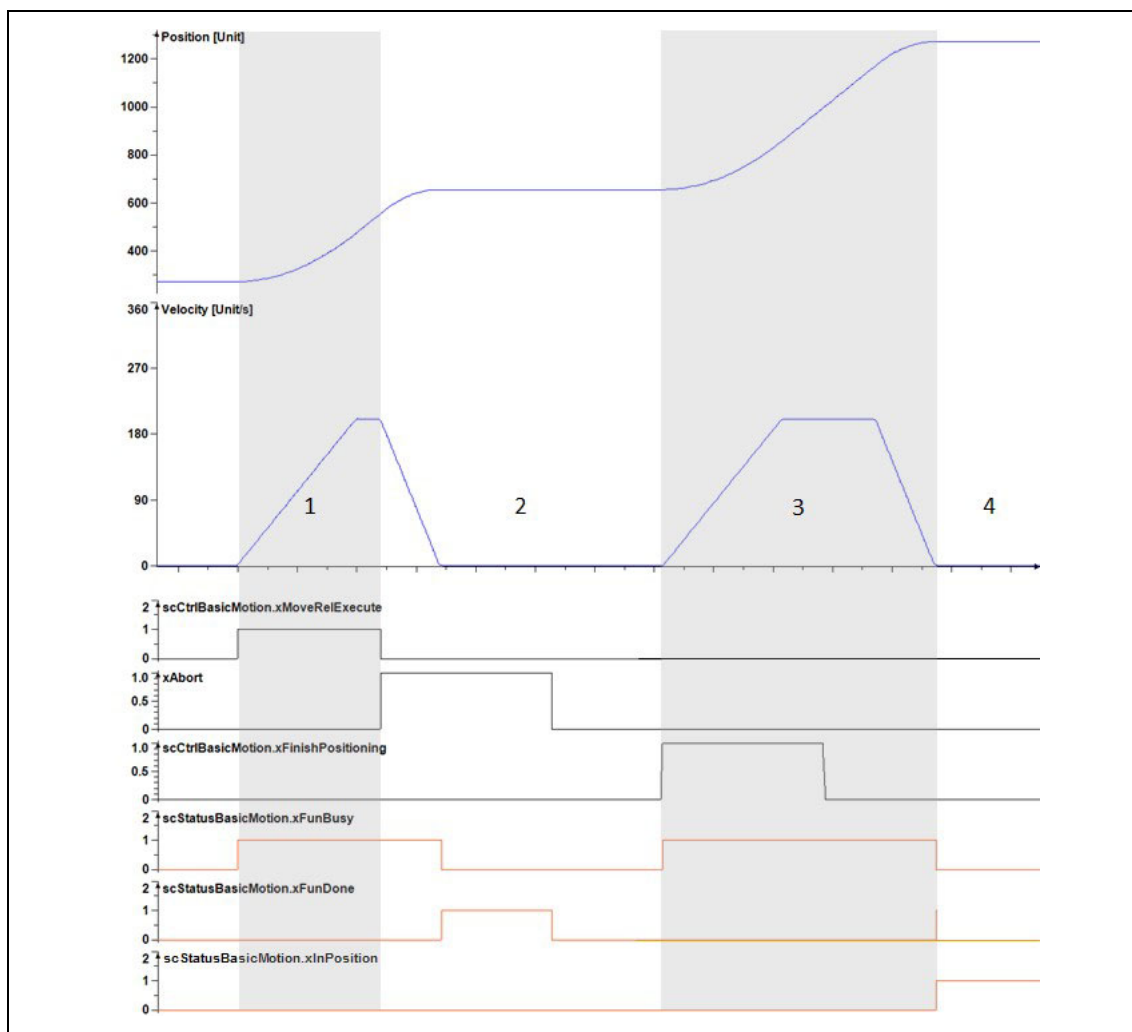
Both technology modules provide several profile data sets, one of which can be active.

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

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

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

Travel diagram



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

8.9 Override for speed

Prerequisite

- The axis is enabled ($xReadyForMotion = \text{TRUE}$).

Execution

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

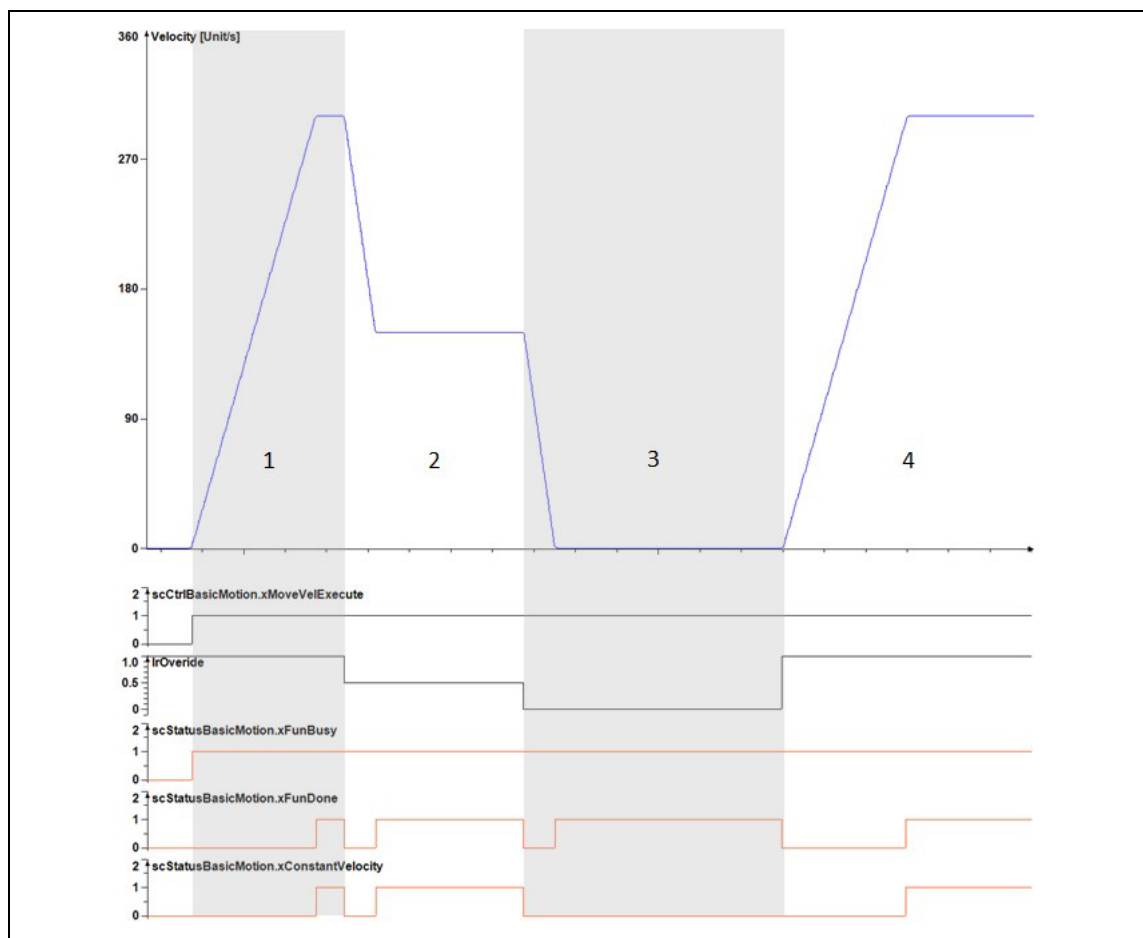
- *IrOverride*, initial value: 1.0
- *scCtrlBasicMotion.IrOverrideAcc*, initial value: 1.0 (100 %)
- *scCtrlBasicMotion.IrOverrideJerk*, initial value: 1.0 (100 %)

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

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

Example

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



8 Description of the basic functions

8.9 Override for speed

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

9 Description of the technology function

Overview of the technology functions

- ▶ [Starting and stopping speed operation](#) (📖 43)
- ▶ [Speed operation with preset setpoints](#) (📖 45)
- ▶ [Inverting a setpoint](#) (📖 50)
- ▶ [External travel profile parameters](#) (📖 52)
- ▶ [Engage speed offset](#) (📖 55)

9.1 Starting and stopping speed operation

Condition

- The technology module is in a fault-free state
- The drive is released (*xReadyForMotion* = TRUE)

Type

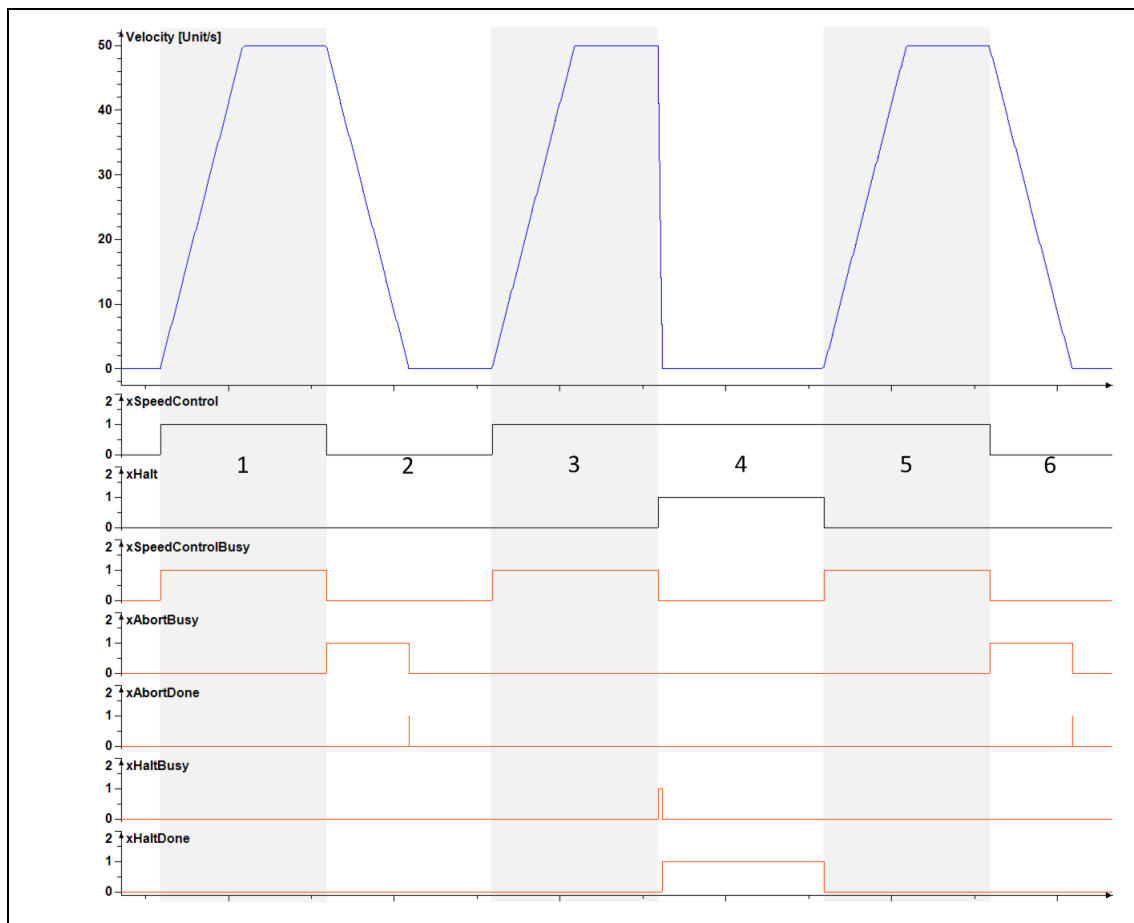
xSpeedControl = TRUE must be set to start a movement with a defined speed. The finite state machine changes to the **MOVEVEL** state and the axis accelerated with limited jerk (*IrJerk*) with the preselected acceleration *IrAcceleration* up to the setpoint speed specified at the *IrSetVel* input of the technology module. As soon as the setpoint speed is reached, the *xConstantVelocity* output of the *scStatusBasicMotion* output structure is set to TRUE. The speed is maintained up to *xSpeedControl* = FALSE. The axis is brought to a standstill according to the deceleration *IrDeceleration* and the finite state machine then changes to the **STANDBY** state.

Alternatively, the axis can also be stopped via *xHalt*, *xAbort* or *xQSPApplication*.

Parameters to be set

Identifier	Data type	Description
<i>IrSetVel</i>	LREAL	Setpoint speed • Unit: units/s
<i>scPar.ascProfiles</i> [0]. <i>IrAcceleration</i>		Initial value: [100 units/s ²]
<i>scPar.ascProfiles</i> [0]. <i>IrDeceleration</i>		Initial value: [100 units/s ²]
<i>scPar.ascProfiles</i> [0]. <i>IrJerk</i>		Initial value: 100000 [units/s ³]

Example



No	xSpeed Control	xHalt	xSpeed Control Busy	xAbortBusy	xAbortDone	xHaltBusy	xHaltDone	eTMState	INFO
1	FALSE ↗ TRUE	FALSE	FALSE ↗ TRUE	FALSE	FALSE	FALSE	FALSE	STANDBY → MOVEVEL	The axis accelerates with limited jerk up to the preselected setpoint speed and holds this.
2	TRUE↘ FALSE	FALSE	TRUE↘ FALSE	FALSE↗ TRUE↘ FALSE (xAbortDone)	FALSE↗ TRUE (for one cycle)	FALSE	FALSE	MOVEVEL → STOP	The axis decelerates to a standstill with the deceleration set in the active profile.
3	FALSE ↗ TRUE	FALSE	FALSE ↗ TRUE	FALSE	FALSE	FALSE	FALSE	STANDBY → MOVEVEL	The axis accelerates with limited jerk up to the preselected setpoint speed and holds this.
4	TRUE	FALSE ↗ TRUE	TRUE↘ FALSE	FALSE	FALSE	FALSE↗ TRUE↘ FALSE (xHaltDone)	FALSE ↗ TRUE	MOVEVEL → STOP	The axis decelerates to a standstill with the deceleration set for the holding function.
5	TRUE	TRUE↘ FALSE	FALSE ↗ TRUE	FALSE	FALSE	FALSE	TRUE↘ FALSE	STOP → MOVEVEL	The axis accelerates with limited jerk up to the preselected setpoint speed and holds this.
6	TRUE↘ FALSE	FALSE	TRUE↘ FALSE	FALSE↗ TRUE↘ FALSE (xAbortDone)	FALSE↗ TRUE (for one cycle)	FALSE	FALSE	MOVEVEL → STOP	The axis decelerates to a standstill with the deceleration set in the active profile.

9 Description of the technology function

9.2 Speed operation with preset setpoints

9.2 Speed operation with preset setpoints

9.2.1 Travel profiles

Up to 16 speed profile data sets can be defined for the speed control operation. Each profile data set consists of:

- Speed (*IrVelocity*)
- Acceleration (*IrAcceleration*)
- Deceleration (*IrDeceleration*)
- Jerk (*IrJerk*)

The profile data sets are saved in structures of the type **L_TF2P_scVelocityProfile** in an array *ascProfiles* and can be selected via the *wProfileNumber* input. The active profile in each case is shown at the *wActiveProfile* output. The array *ascProfiles* is part of the parameter structure **scPar** and can be shown with its default values as follows:

wProfileNumber	IrVelocity	IrAcceleration	IrDeceleration	IrJerk
0	0	100	100	100000
...				
15				

Condition

- The technology module is in a fault-free state
- The drive is released (*xReadyForMotion* = TRUE)

Type

The speed operation is started and stopped as explained in the section above. The switch-over between the profiles is via the *wProfileNumber* input. If the speed value is also to be accepted from the profile data, the *xUseProfileVel* input must be set to TRUE. If the *xUseProfileVel* input has the status FALSE, the setpoint speed is accepted from the *IrSetVel* input. The *xProfileVelActive* output indicates whether the speed has been accepted from the profile data.

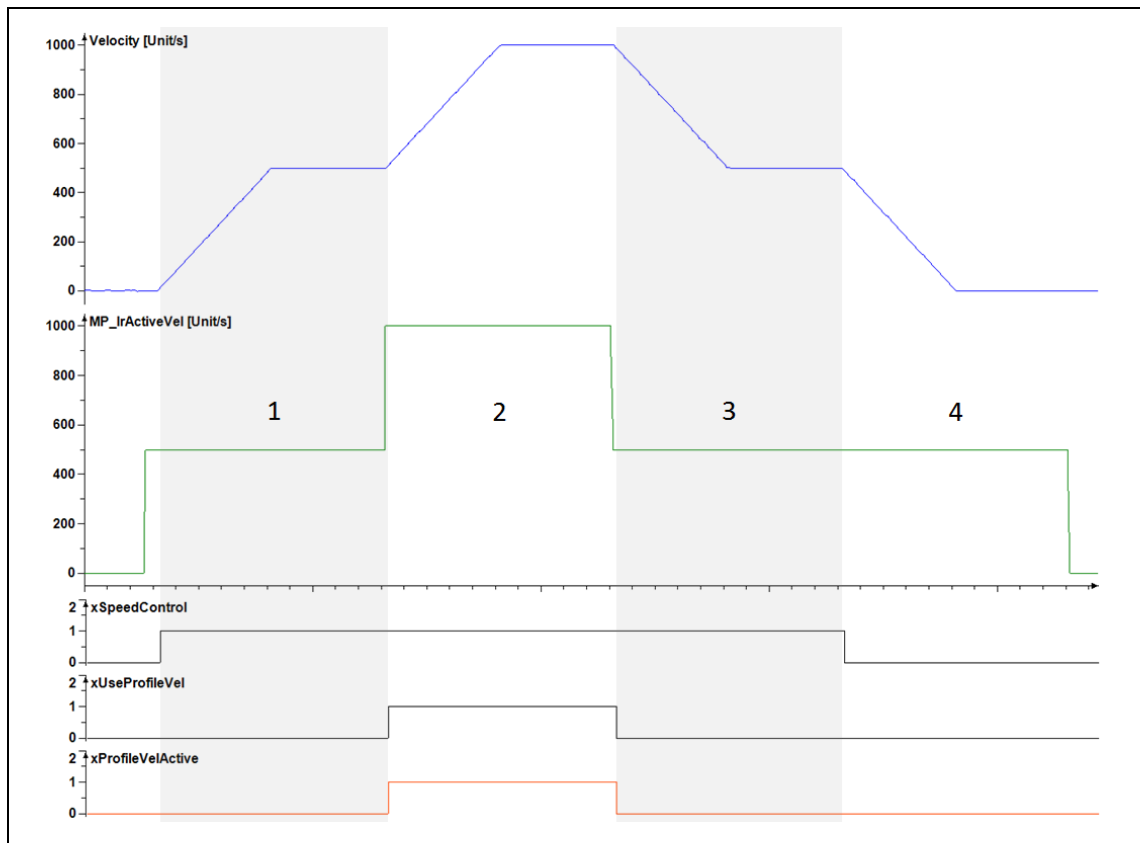
Parameters to be set

Identifier	Data type	Description
Profile data		
scPar. ascProfiles[wProfileNumber]. IrVelocity		Initial value: 0 [units/s]
scPar. ascProfiles[wProfileNumber]. IrAcceleration		Initial value: [100 units/s ²]
scPar. ascProfiles[wProfileNumber]. IrDeceleration		Initial value: [100 units/s ²]
scPar. ascProfiles[wProfileNumber]. IrJerk		Initial value: 100000 [units/s ³]
Profile data selection		
wProfileNumber		Initial value: 0
Selection of the speed source		
xUseProfileVel		Initial value: FALSE

Example

The example shows the use of the *xUseProfileVel* input bit and the resulting behavior of the technology module.

A speed of 500 units/s is preselected at the *IrSetVel* input. The profile speed of profile 0 is set to 1000 units/s. The speed operation starts and the axis ramps up to 500 units/s. The *xUseProfileVel* bit is now set and the speed setpoint accepted from the profile data (1000 units/s). The *xProfileVelActive* output bit shows that this function is active. The *MP_IrActiveVel* signal flow variable indicates that the speed has been accepted. The axis accelerates to 1000 units/s. After *xUseProfileVel* has been reset, the speed setpoint is once again accepted from the *IrSetVel* input. The axis decelerates to 500 units/s.



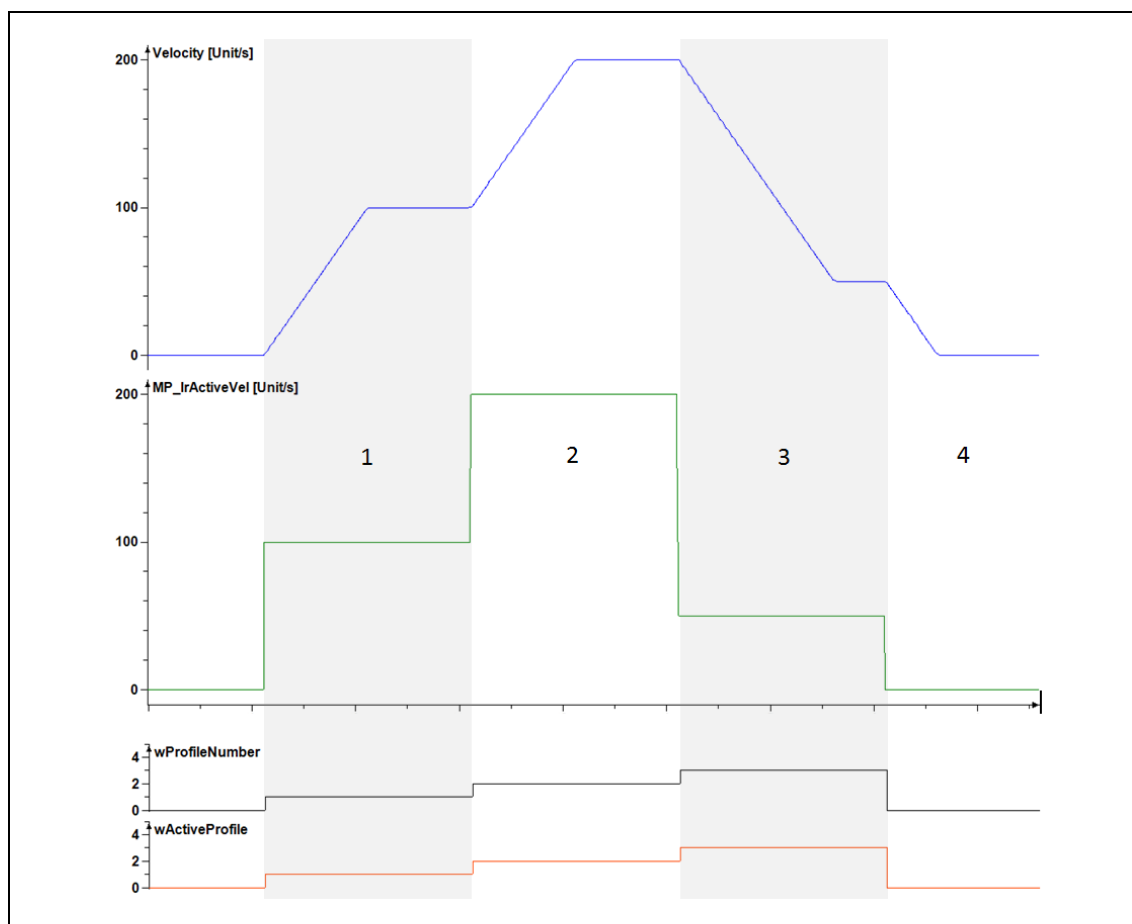
No	xSpeedControl	xUseProfileVel	xProfileVelActive	eTMState	INFO
1	FALSE ↗ TRUE	FALSE	FALSE	STANDBY → MOVEVEL	The axis accelerates with limited jerk to the setpoint speed 500 units/s preselected at the <i>IrSetVel</i> input and holds this.
2	TRUE	FALSE ↗ TRUE	FALSE ↗ TRUE	MOVEVEL	The active profile's speed is activated and the axis accelerates to 1000 units/s and holds this speed
3	TRUE	TRUE ↘ FALSE	TRUE ↘ FALSE	MOVEVEL	The profile speed is deactivated and the axis decelerates to the setpoint speed 500 units/s preselected at the <i>IrSetVel</i> input and holds this.
4	TRUE ↘ FALSE	FALSE	FALSE	MOVEVEL	The axis decelerates to a standstill with the deceleration set for the holding function.

9.2.2 Profile switch-over

You can switch over to individual parameters or complete profile data sets during operation. Changes become effective immediately. Several profile data sets are available for switch-over in the following example.

Example

Speed operation is enabled and switch over between profiles 1, 2 and 3 in succession.



No	wProfile Number	wActive Profile	eTMState	INFO
1	1	1	STANDBY → MOVEVEL	The axis accelerates with limited jerk to the setpoint speed 100 units/s set in profile 1 and holds this.
2	2	2	MOVEVEL	The axis accelerates with limited jerk to the setpoint speed 200 units/s set in profile 2 and holds this.
3	3	3	MOVEVEL	The axis decelerates with limited jerk to the setpoint speed 50 units/s set in profile 3 and holds this.
4	0	0	MOVEVEL → STANDBY	The axis decelerates to a standstill.

The following three profile data sets were parametrized for the example:

Profile number (wProfileNumber)	Speed (IrVelocity)	Acceleration (IrAcceleration)	Deceleration (IrDeceleration)	Jerk (IrJerk)
1	100	1000	1000	100000
2	200	1000	1000	100000
3	50	1000	1000	100000

Parameters to be set

Identifier	Description
wProfileNumber	Initial value: 0

9.3 Inverting a setpoint

Type

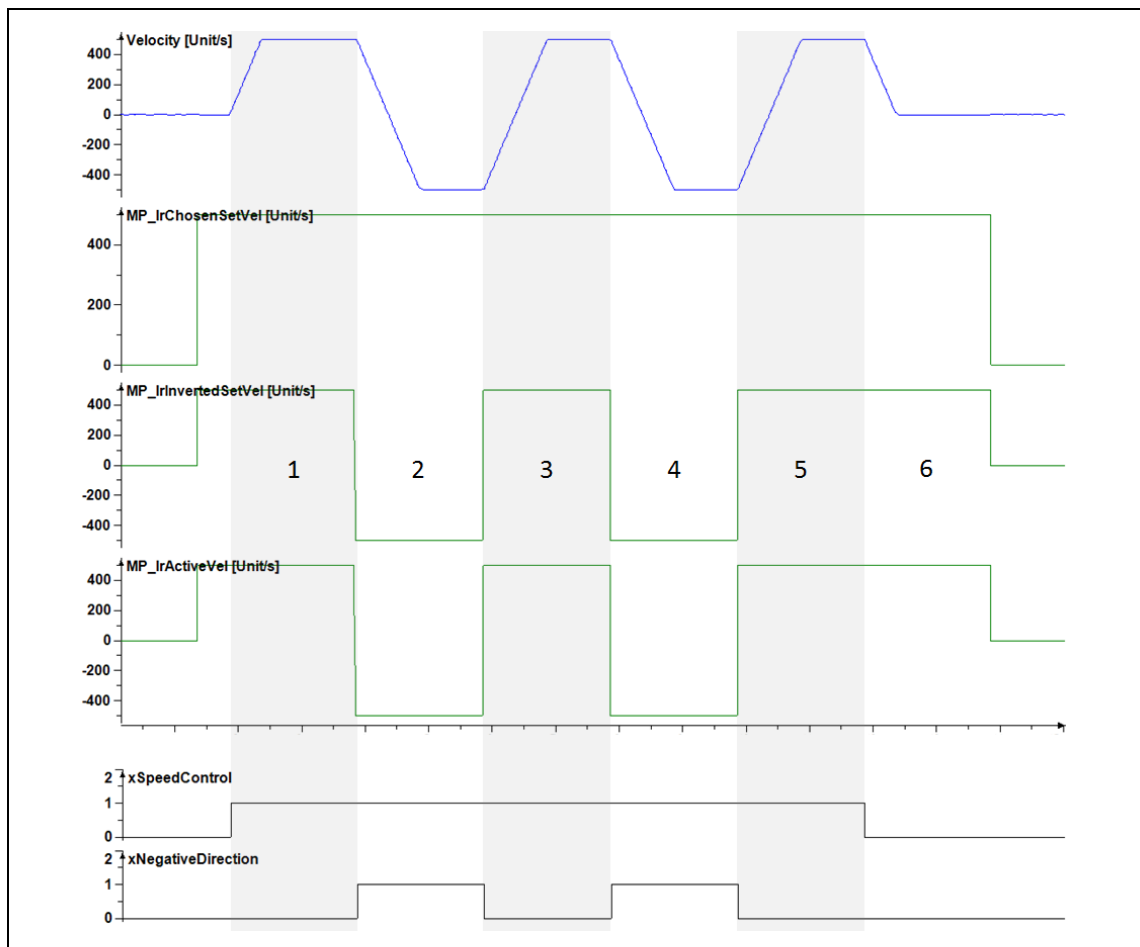
The negation (inversion) of the pre-set *scSignalflow.MP_IrChosenSetVel* setpoint speed is carried out via the *xNegativeDirection* input. If this input is set to TRUE the speed setpoint is multiplied by -1. The speed value behind the negation point is shown via the *scSignalflow.MP_IrInvertedSetVel* signal flow variable. The setpoint speed of the axis is shown by the *IrSetVelOut* output.

Parameters to be set

The value to be set is an input on the technology module

Identifier	Data type	Description	
xNegativeDirection	BOOL	TRUE	Inversion of the direction of rotation Note: inversion does not apply for jogging! • Initial value: FALSE

Example



No	xSpeed Control	xNegative Direction	eTMState	INFO
1	FALSE → TRUE	FALSE	STANDBY → MOVEVEL	The axis accelerates with limited jerk to 500 units/s and holds this.
2	TRUE	FALSE → TRUE	MOVEVEL	The setpoint is inverted and the axis ramps to -500 units/s.
3	TRUE	TRUE → FALSE	MOVEVEL	The setpoint inversion is deactivated and the axis ramps to 500 units/s and holds this.
4	TRUE	FALSE → TRUE	MOVEVEL	The setpoint is inverted and the axis ramps to -500 units/s.
5	TRUE	TRUE → FALSE	MOVEVEL	The setpoint inversion is deactivated and the axis ramps to 500 units/s and holds this.
6	TRUE → FALSE	FALSE	MOVEVEL → STANDBY	The axis decelerates to a standstill.

9.4

External travel profile parameters

Type

Apart from the possibility of defining acceleration and deceleration / jerk values via the profile data, the profile data can also be defined directly via the inputs. The *lrExtAcc*, *lrExtDec* and *lrExtJerk* inputs are available for this purpose. If the *xUseExtAccDecJerk* input is set to = TRUE, the values are no longer accepted from the profile data, but directly from the inputs of the technology module. The *xExtAccDecJerkActive* output indicates whether this function is active or not.

Parameters to be set

Identifier	Data type	Description
Acceleration and deceleration values		
lrExtAcc	LREAL	External acceleration value • Initial value: 100 units/s ²
lrExtDec	LREAL	External deceleration value • Initial value: 100 units/s ²
lrExtJerk	LREAL	External jerk value • Initial value: 100000 units/s ³
Select inputs as source		
xUseExtAccDecJerk	BOOL	TRUE Accept acceleration deceleration values and jerk from the <i>lrExtAcc</i> , <i>lrExtDec</i> and <i>lrExtJerk</i> inputs. • Initial value: FALSE

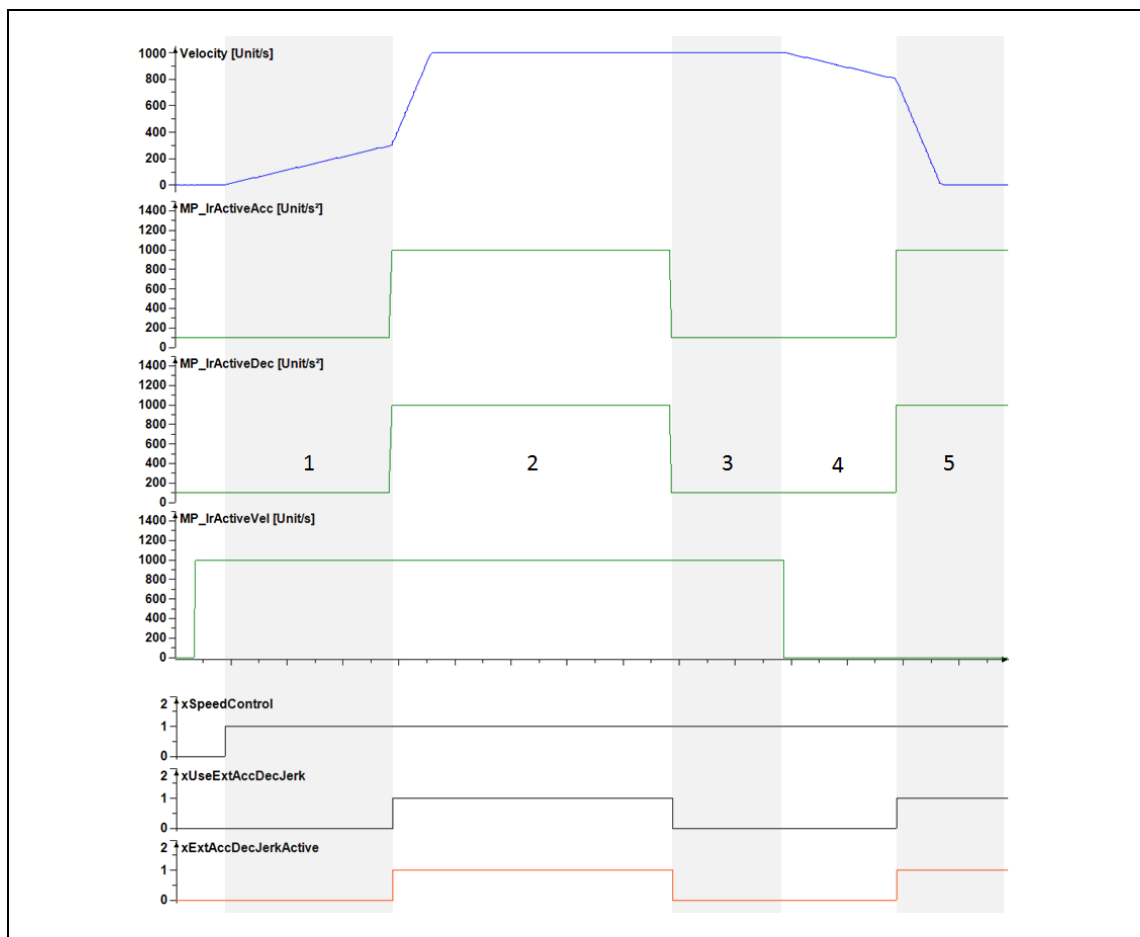
Example

The following example shows the use of the *lrExtAcc* and *lrExtDec* inputs, which are activated via the *xUseExtAccDecJerk* input. In the example, a *lrSetVelocity* setpoint speed of 1000 units/s is preselected and the axis accelerates with the acceleration set in the profile data of 100 units/s².

During the acceleration phase, the external acceleration values are active (*lrExtAcc* = *lrExtDec* = 1000 units/s²). The switch-over results in the immediate use of the input values and the axis now accelerates to the preselected setpoint speed faster. Which acceleration and deceleration values are active can be monitored via the *MP_lrActiveAcc* and *MP_lrActiveDec* measuring points of the signal flow. The *xExtAccDecActive* output indicates that the external acceleration and deceleration values are active.

Once the setpoint speed has been reached, the external values are deactivated again and the profile data are once again active.

The setpoint speed is set to zero and the axis begins to decelerate at 100 units/s². During this deceleration, the external values are once again activated and the axis is brought to a standstill - much faster - with a deceleration of 1000 units/s².



No	xSpeedControl	xUseExtAccDecJerk	xExtAccDecJerkActive	eTMState	INFO
1	FALSE ↗ TRUE	FALSE	FALSE	STANDBY → MOVEVEL	The axis accelerates with limited jerk with 100 units/s ² .
2	TRUE	FALSE ↗ TRUE	FALSE ↗ TRUE	MOVEVEL	The external profile data is activated and the axis continues to accelerate at 1000 units/s ² up to the preselected setpoint speed of 1000 units/s and holds this speed.
3	TRUE	TRUE ↘ FALSE	TRUE ↘ FALSE	MOVEVEL	The external profile data are deactivated.

No	xSpeed Control	xUseExtAcc DecJerk	xExtAccDec JerkActive	eTMState	INFO
4	TRUE	FALSE	FALSE	MOVEVEL	The setpoint speed is set to zero. The axis decelerates at 100 unit /s ² .
5	TRUE	FALSE ↗ TRUE	FALSE ↗ TRUE	MOVEVEL → STANDBY	The external profile data are activated and the axis continues to decelerate at 1000 units/s ² to a standstill.

9.5

Engage speed offset

Type

The *IrSetOffsetVel* input is available so that speed offsets can be taken into account. The value pending is added to the current speed setpoint. Whether a speed offset is effective, i.e. the *IrSetOffsetVel* value is non-zero, is indicated at the *xOffsetVelActive* output.

Parameters to be set

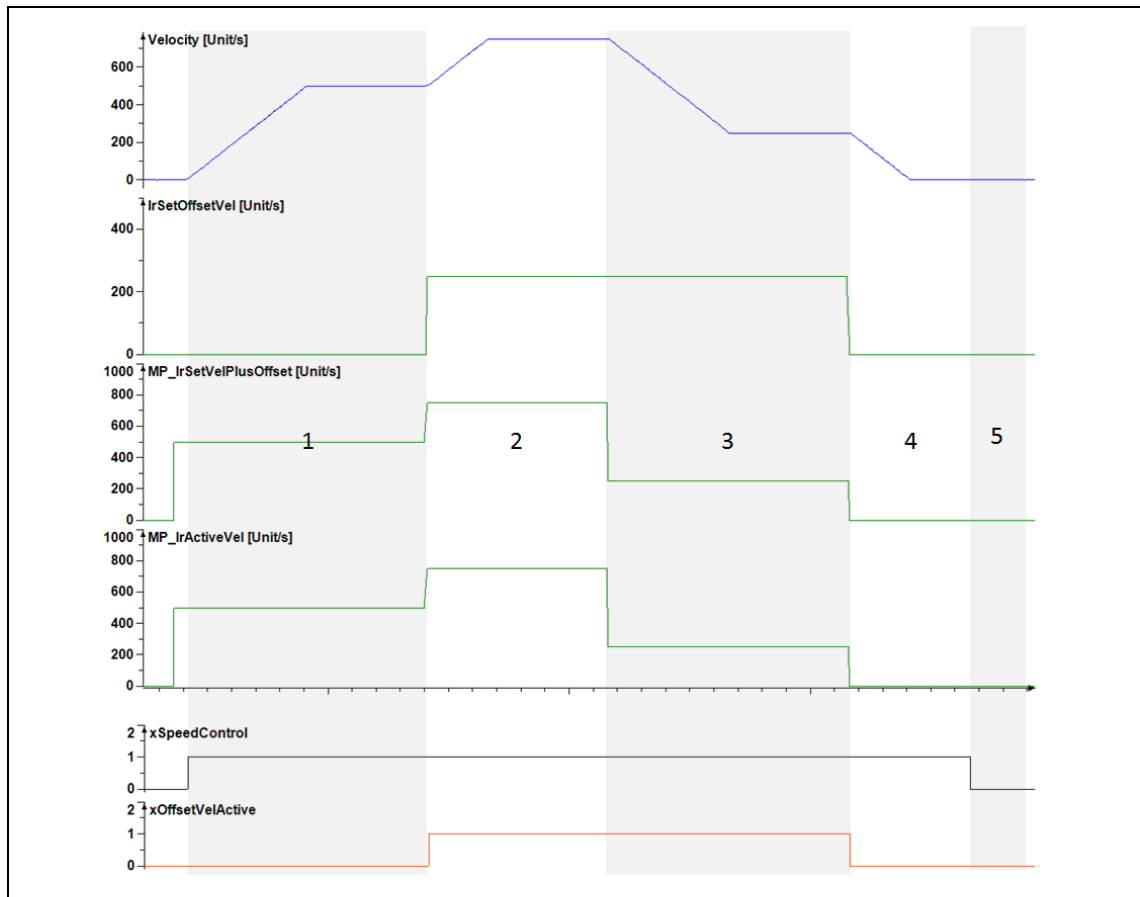
The value to be set is an input on the technology module

Identifier	Data type	Description
IrSetOffsetVel	LREAL	Speed offset Initial value: 0 units/s Note: Acceptance of the value is carried out irrespective of the status in <i>xContinuousUpdate</i> . The value does not influence the jogging speed!

Example

In the example shown here, the preselected speed at the *IrSetVel* input is 500 units/s. The axis accelerates to the preselected value. An offset of 250 units/s is now added via the *IrSetOffsetVel* input. The axis accelerates to 750 units/s. The *xOffsetVelActive* output indicates that an offset is effective.

The value of *IrSetVel* is now set to zero. The axis does not decelerate to a standstill, but holds the speed of 250 units/s. The offset is similarly set to zero and the axis decelerates to a standstill.



No	xSpeed Control	xOffset VelActive	eTMState	INFO
1	FALSE → TRUE	FALSE	STANDBY → MOVEVEL	The axis accelerates with limited jerk to the setpoint speed of 500 units/s.
2	TRUE	FALSE → TRUE	MOVEVEL	A speed offset of 250 units/s is specified and the axis accelerates to 750 units/s.
3	TRUE	TRUE → FALSE	MOVEVEL	The setpoint speed is set to zero and the axis decelerates to 250 units/s.
4	TRUE	FALSE	MOVEVEL	The speed offset is set to zero and the axis decelerates to a standstill.
5	TRUE → FALSE	FALSE	MOVEVEL → STANDBY	Standstill

A

Aborting a positioning process [38](#)

Absolute positioning [30](#)

Access points [21](#)

Application notes [7](#)

C

Competing a positioning process [38](#)

Continuous travel with a defined velocity [35](#)

Conventions used [6](#)

D

Document history [5](#)

E

E-mail to Lenze [58](#)

Engage speed offset [55](#)

External travel profile parameters [52](#)

F

Feedback to Lenze [58](#)

H

Halt, [29](#)

Homing [25](#)

I

inputs [14](#)

L

L_TF2P_scPar_SpeedControlBase [18](#)

L_TF2P_scPar_SpeedControlBase parameter structure [18](#)

L_TF2P_SpeedControlBase [13](#)

Layout of the safety instructions [7](#)

M

Manual jog (jogging) [23](#)

N

Notes on how to operate the technology module [11](#)

O

outputs [16](#)

Override for speed [40](#)

P

Profile switch-over [48](#)

Q

QSP application [28](#)

R

Relative positioning [32](#)

S

Safety instructions [8](#)

Signal flow diagrams [20](#)

Signal flow of the “SpeedControlBase” technology module [20](#)

Speed operation with preset setpoints [45](#)

Speed override [40](#)

SpeedControlBase (functional description) [10](#)

SpeedControlBase (V2) functional description [10](#)

Start and/or stop speed operation [43](#)

Start of the axes [11](#)

State machine [19](#)

States [19](#)

Structure of the signal flow L_TF2P_scSF_SpeedControlBase [21](#)

T

Target group [4](#)

Technology module functions (overview) [10](#)

V

Variable identifiers [6](#)



Your opinion is important to us

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

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

feedback-docu@lenze.com

Thank you very much for your support.

Your Lenze documentation team

