

Technology module



Winder Tension-controlled _ _ _ _ _

Reference manual

EN

Contents

1	About this document	4
1.1	Document history	6
1.2	Conventions used	7
1.3	Layout of the safety instructions	8
2	Safety instructions	9
3	Overview of the technology module functions	11
3.1	Functional description	11
3.2	Functional overview of the TM	13
3.3	Important notes on how to operate the technology module	14
4	Application ranges	16
5	L_TW2P_WinderTensionCtrlState function block	18
5.1	Inputs and outputs	20
5.2	inputs	20
5.3	outputs	24
5.4	Parameter	26
6	State machine	32
7	Signal flow diagram	33
7.1	Structure of the signal flow	35
7.2	Structure of the access points	37
8	Description of the basic functions	39
8.1	Manual jog (jogging)	40
8.2	Homing	42
8.3	QSP application	45
8.4	Halt	46
8.5	Absolute positioning	47
8.6	Relative positioning	49
8.7	Continuous travel with a defined velocity	52
8.8	Aborting and completing a positioning process	55
8.9	Override for speed	57
9	Description of the technology function	59
9.1	Defining the winding direction (winding/unwinding)	59
9.2	Defining the material feed to the winder	61
9.3	Master value source for the diameter calculation	63
9.4	Diameter calculation	64
9.5	Holding the diameter	66
9.6	Defining the diameter (sensor)	67
9.7	Material length counter	69
9.7.1	Material length counter via 'IrSetLineVel' input	70
9.7.2	Material length counter via 'MaterialCounterAxis' reference axis	71
9.8	Inching mode with 1/d evaluation	72
9.9	Synchronisation to the line velocity	73
9.10	Open-loop tension control based on torque controller	74
9.11	Speed feedforward control	76
9.12	Open-loop tension control via characteristic function	77
9.12.1	Linear tensile force characteristic (dwSelectTensCurve = 0)	78
9.12.2	Linear torque characteristic (dwSelectTensCurve = 1)	79
9.12.3	Freely definable characteristics with 65 interpolation points (dwSelectTensCurve = 2) _	80

Contents

9.13	Diameter-dependent tensile force reduction	81
9.14	Closed-loop tension control	82
9.15	Reduced controller dynamics at low system deviation	85
9.16	Speed limitation	86
9.17	Web break monitoring	88
9.18	Persistent variables	89
9.19	Acceleration compensation	92
9.20	Identification of the moments of inertia	94
9.21	Friction identification	98
9.22	Friction compensation	101
9.23	Adaptation of the speed controller gain	103
9.24	Limitation of the master line speed	108
9.25	Termination of the winding process	111
	Index	112
	Your opinion is important to us	113

1 About this document

This documentation ...

- contains detailed information on the functionalities of the "Winder Tension-controlled" 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	07/2020	TD06	Error corrections & supplements
2.0	12/2019	TD06	Error corrections & supplements
1.0	07/2019	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	 7)	Reference to further information: Page number in PDF file.

Variable identifiers

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

1 About this document

1.3 Layout of the safety instructions

DANGER!

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

WARNING!

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

CAUTION!

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

NOTICE

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

Application notes

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

2 Safety instructions

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



The device documentation contains safety instructions which must be observed!

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

DANGER!

High electrical voltage

Injury to persons caused by dangerous electrical voltage

Possible consequences

Death or severe injuries

Protective measures

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

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

Observe the corresponding information plates on the device.

DANGER!

Injury to persons

Risk of injury is caused by ...

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

Possible consequences

Death or severe injuries

Protective measures

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

NOTICE

Damage or destruction of machine parts

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

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

Protective measures

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

3 Overview of the technology module functions

3.1 Functional description

3 Overview of the technology module functions

3.1 Functional description

Variants of winding drives

In many technological processes, winding drives are a key component of an entire system. Depending on the material and the winding process, different open and closed-loop control methods are used:

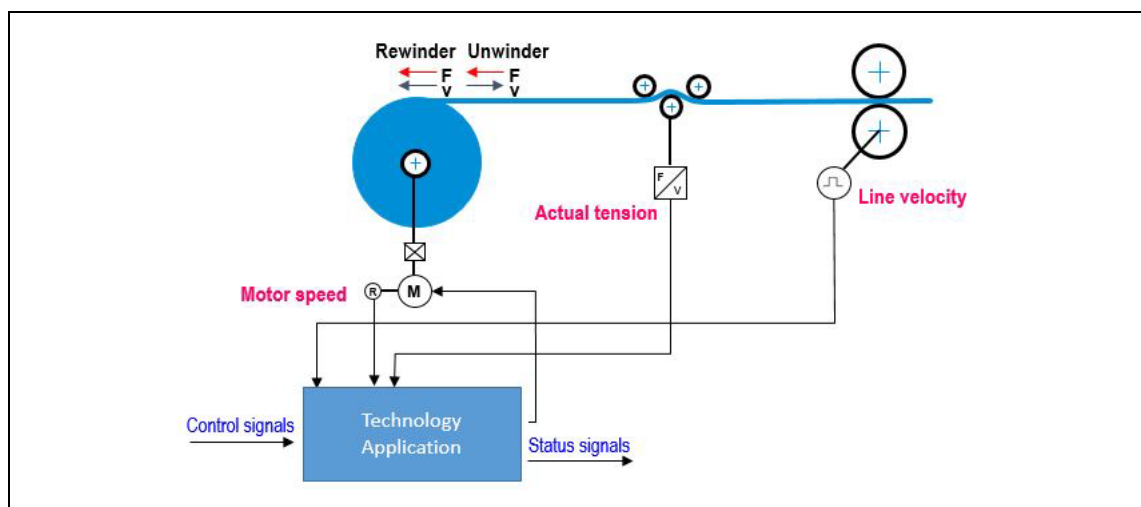
- Dancer position control
- Open-loop tension control
- Closed-loop tension control

Short description

This technology module can be used to design a tension-controlled (open loop)/tension-controlled (closed-loop) winding drive.

The material is rewound or unwound by a centre-driven winding shaft. The winding motor torque indirectly determines the tensile force on the material. The winder speed changes in inverse proportion to the diameter ($n \sim 1/d$); the torque increases proportionally with the diameter ($M \sim d$).

The speed in the machine module is not determined by the winder. The material must have a fixed point so that a web tension can be built via the winder.



[3-1] Structure of a closed-loop tension-controlled winder

If the current tensile force is measured on the material in a winder and fed back in, the system is a closed-loop system. The current tensile force is fed back in as an additive correction value.

The friction profile can also be identified (depending on the winder speed) and used to compensate for friction.

The "State" version also offers an option for identifying the moment of inertia of the winder axis and integrating this value into the technology module configuration for the purpose of acceleration compensation. Furthermore, speed controller gain can be adapted during operation depending on the current moment of inertia.

closed-loop speed control

Higher-level speed control only takes corrective action in the event of a web break to limit the speed of the drive. To prevent the setpoint torque from being affected by the speed limitation during normal operation, a velocity offset must be added to the speed setpoint calculated from the current line speed and the current diameter. The torque setpoint is calculated from the tensile force setpoint multiplied by the current radius, the correcting signal for compensating mechanical friction and the correcting signal for compensating acceleration torque.

Closed-loop tension control

In order to achieve a good winding result, the friction and acceleration compensation must not substantially exceed the lowest load torque. If too great deviations with regard to the tensile force in spite of friction and acceleration compensation are to be expected or are identified, the tensile force setpoint can be corrected correspondingly by recording and closed-loop control of the tensile force.

3.2 Functional overview of the TM

In addition to the basic functions for actuating the **L_MC1P_AxisBasicControlV2** function block, the **Quick stop application function** and the **Holding function**, the technology module offers the following functionalities:

Functionality
BasicMotionBase:
Manual jog (jogging) (📖 40)
Homing (📖 42)
QSP application (📖 45)
Halt (📖 46)
Absolute positioning (📖 47)
Relative positioning (📖 49)
Continuous travel with a defined velocity (📖 52)
Aborting and completing a positioning process (📖 55)
Override for speed (📖 57)
Winder Tension-controlled:
Defining the winding direction (winding/unwinding) (📖 59)
Automatic detection of the winding direction (📖 59)
Master value source for the diameter calculation (📖 63)
Speed feedforward control (📖 76)
Diameter calculation (📖 64)
Holding the diameter (📖 66)
Defining the diameter (sensor) (📖 67)
Material length counter (📖 69)
Open-loop tension control via characteristic function (📖 77)
Reduced controller dynamics at low system deviation (📖 85)
Web break monitoring (📖 88)
Persistent variables (📖 89)
Acceleration compensation (📖 92)
Identification of the moments of inertia (📖 94)
Friction identification (📖 98)
Friction compensation (📖 101)
Adaptation of the speed controller gain (📖 103)
Limitation of the master line speed (📖 108)
Termination of the winding process (📖 111)



»PLC Designer« online help

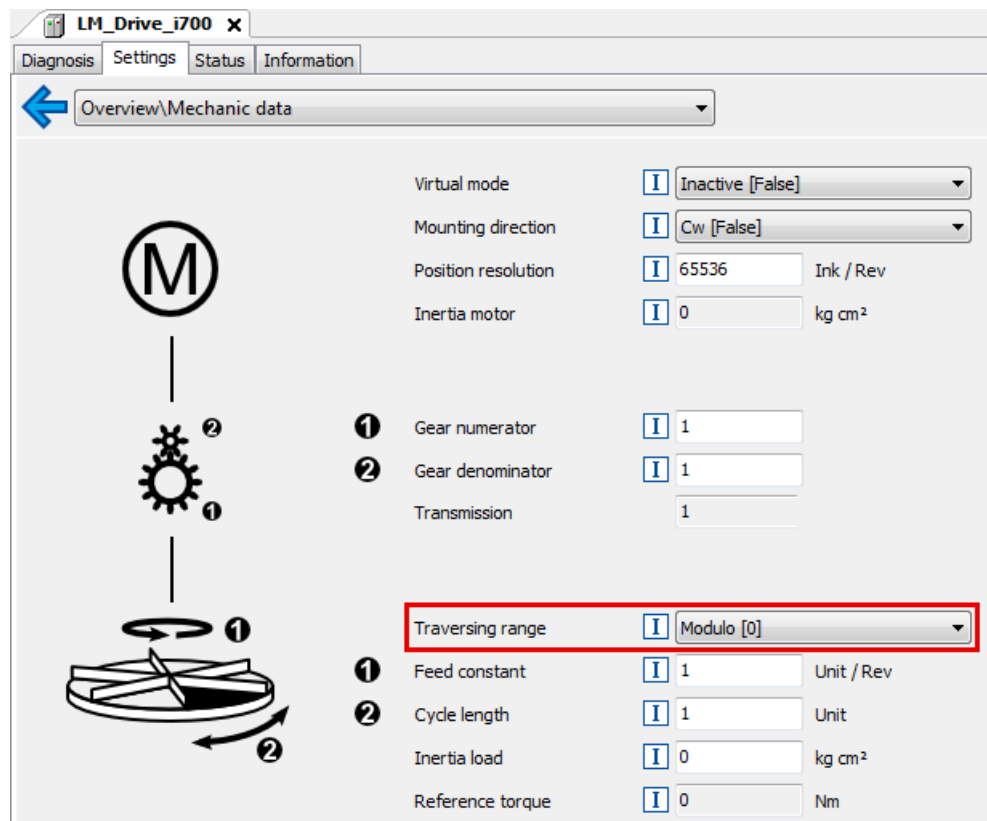
Here you'll find some detailed information with regard to the **L_MC1P_AxisBasicControlV2** function block, the **Quick stop application function** and the **Holding function**.

3.3 Important notes on how to operate the technology module

The technology module ...

- does not support the simulation mode in the »PLC Designer«;
- does not support any virtual axes;
- only supports the Modulo traversing range.

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



Observe that the line speed is specified in the unit [mm/s].

Setting of the operating mode

The operating mode for the winder axis is specified via the technology module. The change-over between "cyclic sync position mode (csp)", "cyclic synchronous speed (csv)" and "cyclic synchronous torque (cst)" in this process internally depends on the active status of the module.

Start of the axes

The method of starting the axes after the controller has been blocked depends on the assigned movement type.

A distinction is made between a movement type whose function is activated by a level and a movement type whose function is activated by an edge.

- The movement type whose function is level-activated is automatically activated after a controller release as long as no higher-priority functions are active. This movement type includes
 - [Manual jog \(jogging\)](#) (40)
 - [QSP application](#) (45)
- The movement type whose function is edge-activated must be activated after a controller release with a FALSE↗TRUE edge. This movement type includes
 - [Homing](#) (42)



Example of a movement type whose function is level-activated: manual jog (jogging)

1. In the inhibited axis state ($xOperationEnabled = FALSE$), $xJogPos$ is set to TRUE.
 - $xEnableOperation = FALSE$ (axis is blocked.)
==> "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 whose function is edge-activated: reference run (homing)

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

4 Application ranges

In the winding process of the torque-controlled center winders, on which this technology module is based, indirect tension control takes place via the variable motor drive torque.

We differentiate:

1. Tension control - Torque controlled method

For tension-controlled winders, the drive is switched to the "torque-controlled method" and receives the setpoint torque, which is determined by the resulting feedforward control value. The feedforward control consists of the tensile force setpoint, the friction compensation and the acceleration compensation. Since no tension controller is active, no measured values from the web tension measurement are required. This means that expensive and complex sensor technology for web tension measurement can be dispensed with.

Typical applications are found in low-cost winders of insensitive materials with low elasticity and medium line speeds such as:

- Coil winders in the metal-processing industry
- Cable winders
- Margin winding units
- Unwinders in the packaging industry

2. Tension control (M) - Torque controlled method

This method has a tension controller in addition to 1. The tension controller is dependent on the measurement of the actual tensile force value. The actual tensile force value is measured by a sensor in the material web and is controlled by the tension controller. The feedforward control consists of the tension setpoint, the friction compensation and the acceleration compensation.

Typical applications are the winding of insensitive materials with low elasticity and medium line speeds such as:

- Films
- Composite materials
- Paper
- Label materials

3. Tension control (n) - Speed controlled method

For tension-controlled winders, the drive is switched to the "speed-controlled method". The drive is operated via the set speed, which is composed of the speed feedforward control and the correcting variable of the tension controller. The traction force controller necessarily requires a traction force measurement. The actual tensile force value is recorded by a sensor in the material web and is controlled by the tensile force controller. The speed feedforward control is determined from the line speed. The torque feedforward control consists of the tensile force setpoint and the acceleration compensation.

The tension control (n) in the speed-controlled method offers a decisive advantage: There is no direct influence by friction and therefore the influence of acceleration is significantly lower.

The disadvantage of this method is that the mass inertia ratio must be observed for stable speed control.

It is also recommended to use a diameter sensor for the correct start diameter if there is no clear start diameter after a roll change.

The tension control (n) is used if there are high demands on tensile force quality and tensile force characteristic or if not determinable disturbance variables like friction act on the winder.

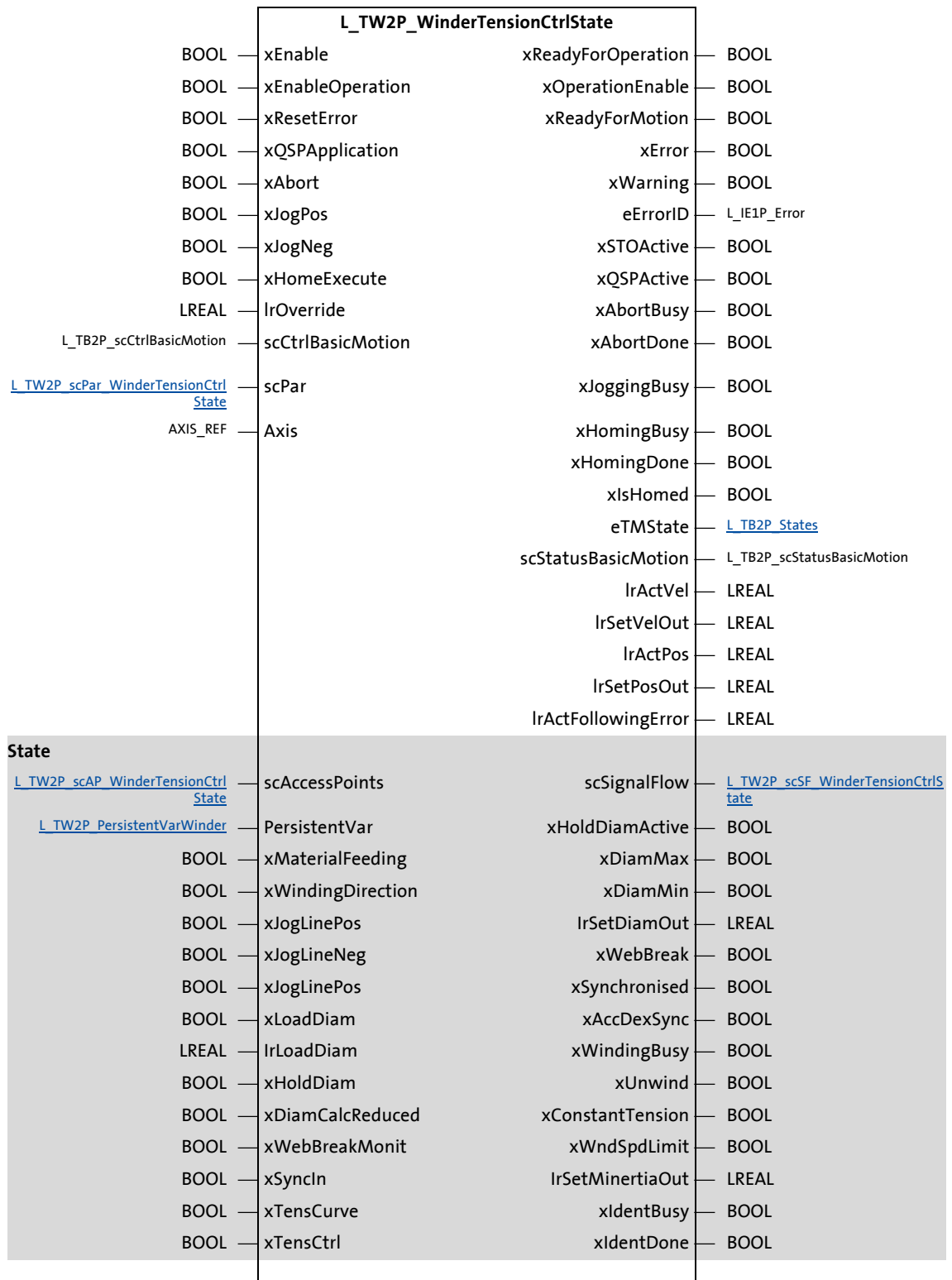
Typical applications are the winding of insensitive materials with low to high elasticity and medium line speeds such as:

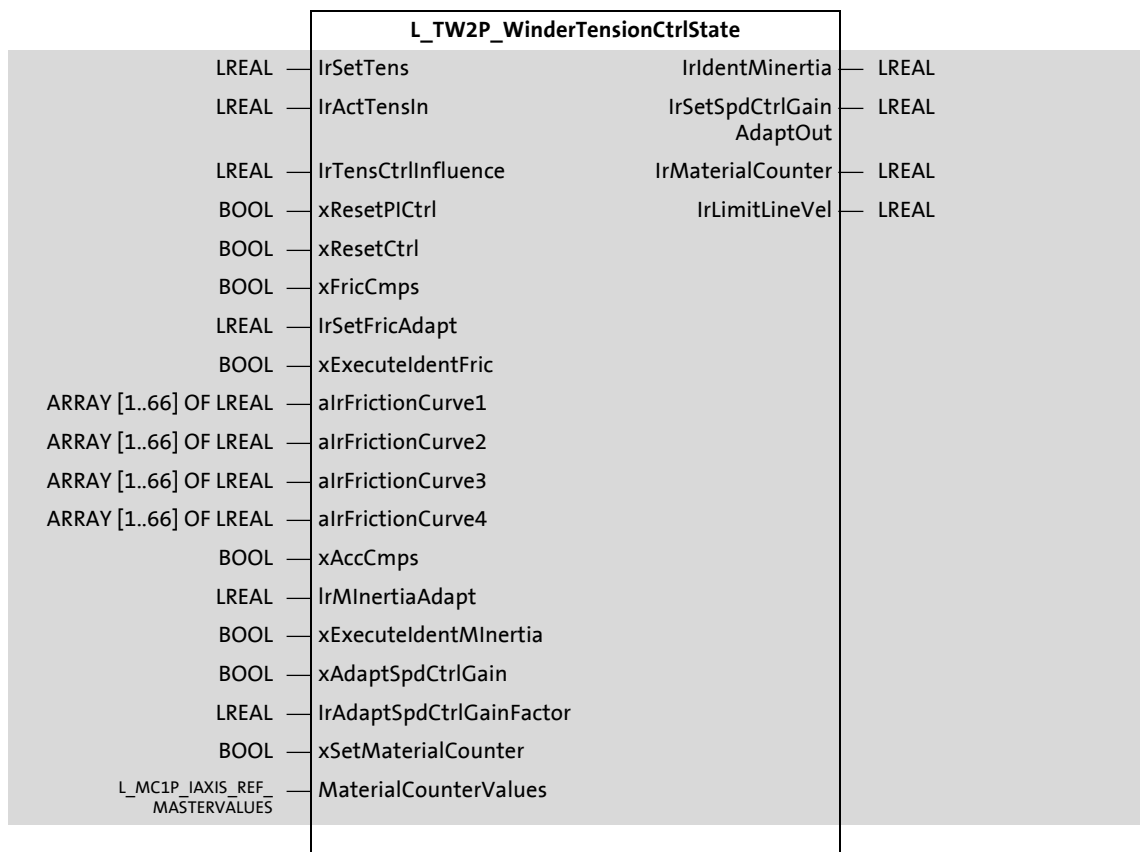
- Films
- Composite materials
- Paper
- Label materials

5 L_TW2P_WinderTensionCtrlState function block

The illustration shows that the inputs and outputs are part of the function block. The additional inputs and outputs of the "State" version are shaded.

► [L_TW2P_scSF_WinderTensionCtrlState](#) (35)





5.1

Inputs and outputs

Identifier	Data type	Description
PersistentVar L_TW2P_PersistentVarWinder		Reference to persistent variables In the reference, the following data are managed: Diameter calculated
alrFrictionCurve1 ARRAY [1..66] OF LREAL		Friction characteristics 1 ... 4 consisting of 65 data points each The reference speed for the data points is saved in field 66. The interpolation points are entered automatically in the course of the friction identification process.
alrFrictionCurve2 ARRAY [1..66] OF LREAL		
alrFrictionCurve3 ARRAY [1..66] OF LREAL		
alrFrictionCurve4 ARRAY [1..66] OF LREAL		

5.2

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.
xResetError	BOOL	TRUE Reset axis error or software error.
xQSPApplication	BOOL	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>)
xJogPos	BOOL	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	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 Start homing.
		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>

Identifier	Data type	Description
scCtrlBasicMotion L_TB2P_scCtrlBasicMotion		
scPar L_TW2P_scPar_WinderTensionCtrlState		The parameter structure contains the parameters of the technology module.
Axis	AXIS_REF	Reference to the axis
scAccessPoints L_TW2P_scAP_WinderTensionCtrlState		Structure of the access points Via the access points, the signal flow or the functions of the technology module can be influenced.
xMaterialFeeding	BOOL	Material feeding at the reels from the top or from the bottom - Initial value: FALSE ⚠ WARNING! Injury to persons Risk of injury is caused by changing the setting at input <i>xMaterialFeeding</i> during a movement. This can lead to a speed jump of the winder axle. Possible consequences Death or severe injuries Protective measures Only change the setting of input <i>xMaterialFeeding</i> if the technology module is not running and the inverter is inhibited: - Input <i>xEnable</i> = FALSE - Input <i>xEnableOperation</i> = FALSE
		TRUE Material feeding from the top
		FALSE Material feeding from the bottom
xWindingDirection	BOOL	Function of the winder for positive line speed (<i>lrSetLineVel</i> > 0) - Initial value: FALSE ⚠ WARNING! Injury to persons Risk of injury is caused by changing the setting at input <i>xWindingDirection</i> during a movement. This can lead to a speed jump of the winder axle. Possible consequences Death or severe injuries Protective measures Only change the setting of input <i>xWindingDirection</i> if the technology module is not running and the inverter is inhibited: - Input <i>xEnable</i> = FALSE - Input <i>xEnableOperation</i> = FALSE
		TRUE Unwinder
		FALSE Rewinder
xWindingDirectionStandstill	BOOL	Option for specifying the winding direction at standstill.
		TRUE Unwinder (the tension-reducing friction is taken into account)
		FALSE Winder (the tension-increasing friction is taken into account)
xJogLinePos	BOOL	TRUE Drive axis in positive material flow direction (manual jog). The path parameters (speed, acceleration and deceleration) refer to the circumferential speed of the reel.
xJogLineNeg	BOOL	TRUE Drive axis in negative material flow direction (manual jog). The path parameters (speed, acceleration and deceleration) refer to the circumferential speed of the reel.
lrSetLineVel	LREAL	Current line speed as a setpoint for the winder Unit: mm/s

Identifier	Data type	Description
xLoadDiam	BOOL	TRUE Load the (start) diameter from the <i>IrLoadDiam</i> input. • Initial value: FALSE
IrLoadDiam	LREAL	Defining a (start) diameter The diameter is loaded cyclically when the <i>xLoadDiam</i> is set to TRUE. • Unit: mm • Initial value: 0
xHoldDiam	BOOL	Hold the current diameter Initial value: FALSE
		TRUE Current diameter is held.
		FALSE The current diameter is not held.
xDiamCalcReduced	BOOL	Mode change-over of the diameter calculation between a long/short distance • Initial value: FALSE
		TRUE Diameter is updated after the short distance. • Settings via <i>IrDiamCalcReducedDist</i> parameter
		FALSE Diameter is updated after the regular distance. • Settings via <i>IrDiamCalcRegularDist</i> parameter
xWebBreakMonit	BOOL	Activate/Deactivate web break monitoring • Initial value: FALSE
		TRUE Activate web break monitoring.
		FALSE Deactivate web break monitoring.
xSyncln	BOOL	The winder axis is synchronised to the line speed <i>IrSetLineVel</i> . The input is level-controlled and evaluates the HIGH level. Enable of the axis via <i>xEnableOperation</i> required. Commissioning function to make the winding shaft (without mounted material) follow the line speed.
		TRUE Synchronisation using acceleration <i>IrSynclnAcc</i> . Speed follows <i>IrSetLineVel</i> .
		FALSE No synchronous motion to <i>IrSetLineVel</i> . Come to a standstill over the deceleration time period <i>IrSynclnDec</i> .
xTensCurve	BOOL	Enable tensile force characteristic Initial value: TRUE
		TRUE Enable tensile force characteristic.
xTensCtrl	BOOL	Open-loop/closed-loop tension-controlled mode enabled
		TRUE Open-loop/closed-loop tension-controlled mode enabled
		FALSE Exit open-loop/closed-loop tension-controlled mode depending on the <i>scPar.eTensCtrlStopMode</i>
IrSetTens	LREAL	Tensile force setpoint • Unit: N • Initial value: 0
IrActTensln	LREAL	Current actual tensile force value • Unit: x 100 % (1 = 100 %) • Initial value: 0
IrCtrlInfluenceScaled	LREAL	Influence of the tensile force process controller • Unit: N • Initial value: 0
xResetPICtrl	BOOL	TRUE The PI controller is switched off and the correcting variable (output of the controller) from the PID component is led to '0' via a ramp function. After the controller is switched back on, the setpoint/actual value is synchronised. The correcting variable is managed via a ramp generator to generate a smooth, continuous progression of setpoints for the drive. The ramp is specified by the <i>scPar.IrTensRamp</i> parameter in the unit [N/s].

Identifier	Data type	Description
xResetCtrl	BOOL	TRUE The I component of the PI controller can be switched off and the correcting variable (output of the controller) from the I component can be led to the value 0 via the ramp function. The correcting variable from the P component is not affected.
xFricCmps	BOOL	TRUE Activate friction compensation during the open loop/closed loop tension control process.
IrSetFricAdapt	LREAL	Multiplicative evaluation of the current friction factor, e.g. in order to factor in the influences of temperature.
xExecutIdentFric	BOOL	The input is edge-controlled and evaluates the rising edge.
		FALSE Activate identification of the friction characteristic. TRUE
xAccCmps	BOOL	Activate acceleration compensation during the open-loop/closed-loop tension control process. • Initial value: FALSE
IrMInertiaAdapt	LREAL	Multiplicative evaluation of the current variable moment of inertia, e.g. in order to factor in varying material density and widths. • Initial value: 0
xExecutIdentMInertia	BOOL	The input is edge-controlled and evaluates the rising edge.
		FALSE The moment of inertia at the winder shaft is detected. TRUE The IrIdentMInertia output displays the detected moment of inertia in kgcm ² .
xAdaptSpdCtrlGain	BOOL	Switch adaptation of the speed controller gain on/off. • Initial value: FALSE
		TRUE Enable adaptation of the speed controller gain.
		FALSE Adaptation of the speed controller gain inactive.
IrAdaptSpdCtrlGainFactor	LREAL	The resulting value of the adaptation of the speed controller gain can be influenced multiplicatively via this input. • Value range: 0 ... 1 • Initial value: 1
xSetMaterialCounter	BOOL	The input is edge-controlled and evaluates the FALSE→TRUE edge.
		TRUE Sets the material length counter (<i>IrMaterialCounter</i> output) to the value set under the <i>IrSetMaterialPos</i> parameter.
MaterialCounterValues L_MC1P_IAXIS_REF_MASTERVALUES		Here a modulo axis of a measuring wheel can be connected to the material. If an axis is connected to the input, the material length is determined on the basis of the data from the reference axis. This procedure is also suitable for noisy signals. If no axis is connected here, the material length is determined by integrating the material speed (<i>IrSetLineVel</i> input). ▶ Material length counter (69) ▶ Sources for the material length counting (69) Supported axes: • Encoder • Servo inverter 9400 HighLine • Servo inverter i700 • i950 servo inverter

5.3

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.
xError	BOOL	TRUE There is an error in the technology module.
xWarning	BOOL	TRUE There is a warning in the technology module.
eErrorID	L_IE1P_Error	ID of the error or warning message if <i>xError</i> = TRUE or <i>xWarning</i> = TRUE. "FAST technology modules" reference manual: Here you can find information on error or warning messages.
xSTOActive	BOOL	TRUE 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.
xAbortBusy	BOOL	TRUE The function activated is aborted by the <i>xAbsort</i> input.
xAbortDone	BOOL	TRUE The function activated has been aborted by the <i>xAbsort</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").
eTMState	L_TB2P_States	Current state of the technology module ► State machine (32)
scStatusBasicMotion	L_TB2P_scStatusBasicMotion	Structure of the status data of the function block
IrActVel	LREAL	Current velocity • Unit: unit / s
IrSetVelOut	LREAL	Setpoint speed • Unit: unit / s
IrActPos	LREAL	Actual position • Unit: unit
IrSetPosOut	LREAL	Setpoint position • Unit: unit
IrActFollowingError	LREAL	Current following error • Unit: unit
scSignalFlow	L_TW2P_scSF_WinderTensionCtrlState	Structure of the signal flow

Identifier	Data type	Description
xHoldDiamActive	BOOL	Signals whether the diameter is held. Note: When this output is toggled, we recommend that the hysteresis function is used (<i>scPar.IrHystlineStandstill</i> parameter). ► Signal flow diagram (📖 33)
		TRUE Internal or conditional external holding is active.
		FALSE The diameter is updated again after the distance has passed.
xDiamMax	BOOL	TRUE The maximum diameter has been reached.
xDiamMin	BOOL	TRUE The minimum diameter has been reached.
IrSetDiamOut	LREAL	Current diameter calculated • Unit: mm
xWebBreak	BOOL	TRUE A web break has been detected.
xSynchronised	BOOL	TRUE The axis setpoint is synchronous with <i>IrSetLineVel</i> . This status is not active if <i>xWindingBusy</i> = TRUE.
xAccDecSync	BOOL	TRUE The synchronisation function is active. Synchronisation of the winder is carried out or cancelled.
xWindingBusy	BOOL	TRUE Winding operation has been activated.
xUnwind	BOOL	Status bit for unwinder and rewinder Note: When this output is toggled, we recommend that the hysteresis function is used (<i>scPar.IrHystlineStandstill</i> parameter). ► Signal flow diagram (📖 33)
		TRUE Unwinding mode active
		FALSE Winding mode active
xConstantTension	BOOL	TRUE Constant tension has been set automatically.
xWndSpdLimit	BOOL	TRUE The speed limit has been reached
IrSetMIInertiaOut	LREAL	Current moment of inertia at the winder shaft • Unit: kgcm ²
xIdentBusy	BOOL	TRUE The identification of the moment of inertia or the friction is being determined.
xIdentDone	BOOL	TRUE The identification of the mass inertia or the friction has been completed.
IrIdentMIInertia	LREAL	Identified moment of inertia at the winder shaft • Unit: kgcm ²
IrSetSpdCtrlGainAdaptOut	LREAL	Adaptation of the speed controller gain • Unit: x 100 % (1 = 100 %)
IrMaterialCounter	LREAL	Display the material length counter status Depending on the definition of the winding direction, the material length counter is incremented or decremented • Unit: mm ► Defining the winding direction (winding/unwinding) (📖 59).
IrLimitLineVel	LREAL	The maximum permitted line speed is determined from the calculated diameter and the <i>scPar.IrMaxWndSpd</i> parameter. • Unit [mm/s]

5.4

Parameter

L_TW2P_scPar_WinderTensionCtrlState

Identifier	Data type	Description
IrVel	LREAL	Speed for movements that are commissioned via the scCtrlBasicMotion structure. - Unit: unit/s - Initial value: 3600
IrAcc	LREAL	Acceleration for movements that are commissioned via the scCtrlBasicMotion structure. Selection of the value for the maximum acceleration speed variation. - Unit: unit/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: unit/s² - Initial value: 100
IrJerk	LREAL	Jerk for movements that are commissioned via the scCtrlBasicMotion structure. - Einheit: Unit/s³ - Initial value: 0
eDirection	MC_DIRECTION	Traversing direction: <i>mcCurrentDirection</i>: Retain current direction (only in case of modulo axis) <i>mcPositiveDirection</i>: Positive direction <i>mcNegativeDirection</i>: Negative direction <i>mcShortestWay</i>: Shortest distance (only in case of modulo axis) - Initial value: <i>mcPositiveDirection</i>
scHomeExtTP	MC_TRIGGER_REF	Transfer of an external touch probe event: - Only relevant for "external encoder" touch probe configuration. - For describing the MC_TRIGGER_REF structure, see the MC_TouchProbe function block.
IrLineJerk	LREAL	Jerk for inching mode with the 1 / d evaluation and for the purpose of compensation with regard to the clutch function - Unit: rev / s³ - Initial value: 100000
IrJogLineAcc	LREAL	Acceleration for the inching mode with the 1 / d evaluation. Selection of the value for the maximum acceleration speed variation. - Unit: mm/s² - Initial value: 100
IrJogLineDec	LREAL	Deceleration for the inching mode with the 1 / d evaluation. Selection of the value for the maximum deceleration speed variation. - Unit: mm/s² - Initial value: 10
IrJogLineVel	LREAL	Speed for the inching mode with the 1 / d evaluation. Selection of the maximum speed used for traversing. - Unit: mm/s - Initial value: 10
IrSyncLineAcc	LREAL	Acceleration for synchronising to line velocity - Unit: mm/s² - Initial value: 100
IrSyncLineDec	LREAL	Deceleration for synchronising to line velocity - Unit: mm/s² - Initial value: 100

Identifier	Data type	Description
IrWebBreakWindow	LREAL	Web break window The current diameter is compared to the previous diameter across the web break window from the last calculation cycle. - Unit: x 100 % (1.0 = 100 %) - Initial value: 0.1 (10 %)
IrMaxDiam	LREAL	Maximum diameter - Unit: mm - Initial value: 180
IrMinDiam	LREAL	Minimum diameter - Unit: mm - Initial value: 50
rFiltTimeDiam	REAL	PT1 filter time for the current diameter (<i>IrSetDiamOut</i>) - Unit: s - Initial value: 0.05
IrDiamCalcRegularDist	LREAL	Regular calculation distance for the diameter - Unit: rev - Initial value: 1
IrDiamCalcReducedDist	LREAL	Reduced calculation distance for the diameter - Unit: rev - Initial value: 0.1
alrAdaptDiamX ARRAY [1...9] OF LREAL		Grid points of the curve function for diameter loading - Values that may be applied to the analog input <i>IrSetDiam</i>. - Unit: mm - Initial values: 0, 100, 200, 300, 400, 500, 600, 700, 800
alrAdaptDiamY ARRAY [1...9] OF LREAL		Grid points of the curve function for diameter loading - Function values for the diameter - Unit: mm - Initial values: 0, 100, 200, 300, 400, 500, 600, 700, 800
IrTensCurveCtrlScaled	LREAL	Gradient of the characteristic for the tension control open loop - Unit: x 100 % (1 = 100 %) - Initial value: 0 The value '1' causes a constant tensile force profile and thus a proportionally increasing setpoint for the diameter.
IrTensCurveStartDiamScaled	LREAL	Initial point of the characteristic for the tension control open loop - Unit: x 100 % 1 = 100 % = <i>IrMaxDiam</i> parameter - Initial value: 0 (0 %)
IrLineVelRef	LREAL	Maximum line velocity - Unit: mm/s - Initial value: 1000
IrMinLineVel	LREAL	Minimum line velocity Up to this velocity, the diameter is held. - Unit: mm/s - Initial value: 1
rFiltTimeMaterialCounter	REAL	Filter time constant for the material length counter (<i>IrMaterialCounter</i> output) - Unit: s - Initial value: 0 (filter is deactivated.)
IrSetMaterialPos	LREAL	Position of the material length counter With a FALSE→TRUE edge at the <i>xSetMaterialCounter</i> input, the material length counter (<i>IrMaterialCounter</i> output) is set to the value in <i>IrSetMaterialPos</i> . - Unit: mm - Initial value: 0
IrMaxWndSpd	LREAL	Definition of the maximum winder shaft speed (at the output end). - Unit: rev/s - Initial value: 100

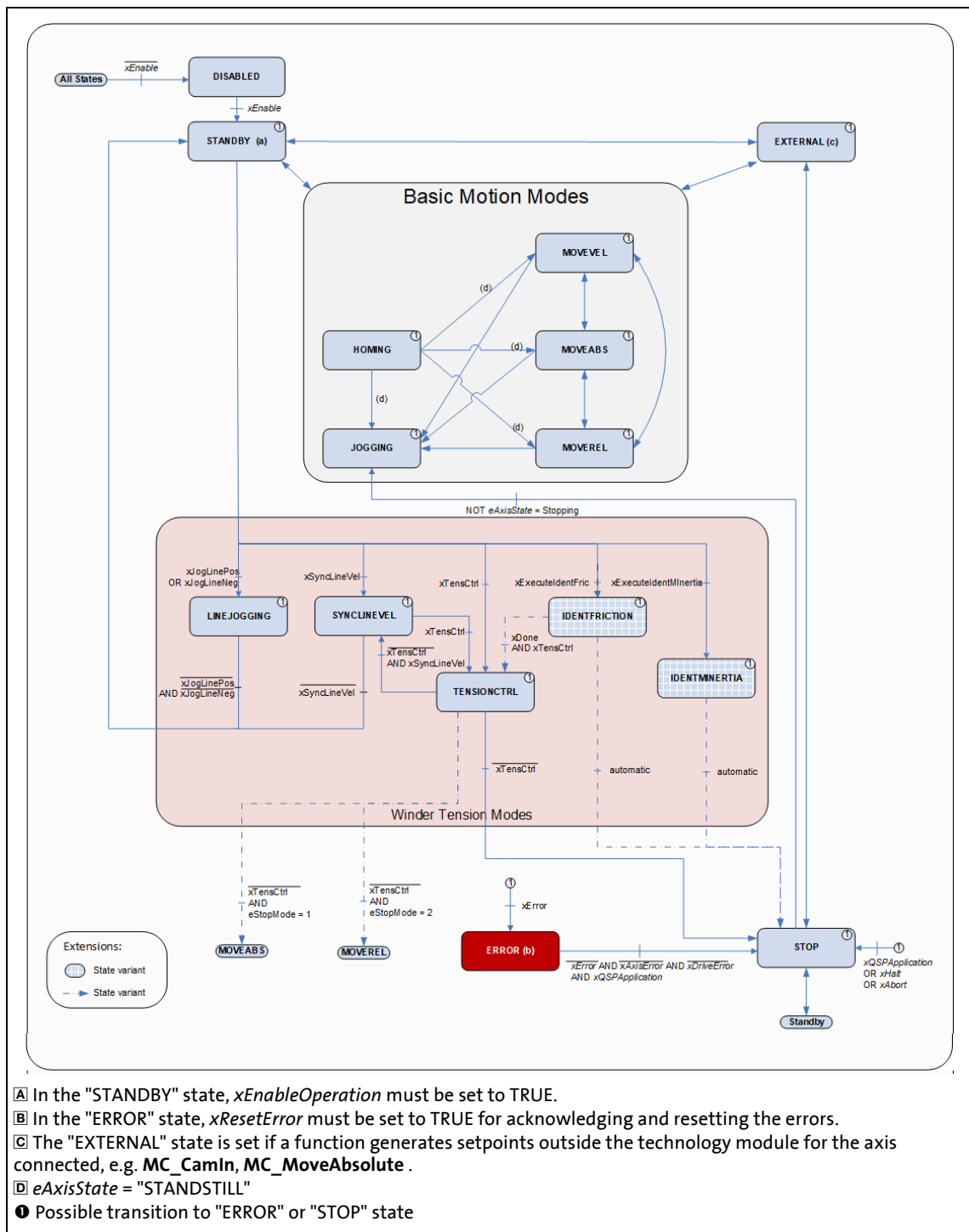
Identifier	Data type	Description
lrLineVelOffsetScaled	LREAL	Line speed offset for speed limitation - Unit: x 100 % (1 = 100 % = lrLineVelRef parameter) - Initial value: 0.1
lrTensRef	LREAL	Maximum permissible tensile force - Unit: N - Initial value: 50
lrTensRamp	LREAL	Acceleration ramp for the tensile force setpoint - Unit: N/s - Initial value: 50
rFiltTimeWndSpd	REAL	PT1 filter time for the winder shaft speed - Unit: s - Initial value: 0.01
rFiltTimeAccSpd	REAL	PT1 filter time for the line speed which is calculated for the speed via the diameter, for acceleration compensation - Unit: s - Initial value: 0.005
lrAccCmpsDeadBandTrqScaled	LREAL	Lagging range (dead band) for the current acceleration torque - Unit: x 100 % (1.00 = 100 %); reference variable: rated motor torque [Nm] - Initial value: 0.1 (10 %)
lrAccCmpsGainAcc	LREAL	Gain factor for the acceleration torque in positive direction - Unit: x 100 % (1.00 = 100 %) - Initial value: 1.05 (105 %)
lrAccCmpsGainDec	LREAL	Gain factor for the acceleration torque in negative direction - Unit: x 100 % (1.00 = 100 %) - Initial value: 0.95 (95 %)
lrConstMIInertia	LREAL	Constant moment of inertia at the winder shaft - Unit: kgcm² - Initial value: 1
lrMaxMIInertia	LREAL	Maximally permissible moment of inertia at the winder shaft - Unit: kgcm² - Initial value: 10
lrFricCurveStartTrq	LREAL	Initial friction for the linear friction compensation if the winder shaft is at a standstill. - Unit: Nm - Initial value: 0
lrFricCurveEndTrq	LREAL	Final friction for the linear friction compensation if the winder shaft is at a standstill. - Unit: Nm - Initial value: 0
rFiltTimeFricSetSpd	REAL	PT1 filter time for the winder shaft speed for friction compensation - Unit: s - Initial value: 0.01
eTensCtrlMode L_TW2P_WinderTensionControlMode		Selection of the axis control mode for the reel The mode is effective in winding mode when the tension control/regulation is activated. Initial value: 0
	0	"TorqueBased" The torque control is activated in the drive when the technology module changes to the "TENSIONCTRL" state.
	1	"SpeedBased" Speed control is activated in the drive when the technology module changes to the "TENSIONCTRL" state.

Identifier	Data type	Description
dwSelectTensCurve	DWORD	Selection of the characteristic for tension control Initial value: 0
		0 Linear tensile force profile
		1 Linear torque profile
		2 Tensile force profile on the basis of a specified characteristic
alrTensCurve	ARRAY[1..65] OF LREAL	Characteristic for tension control open loop consisting of 65 values.
lrTensCtrlGain	LREAL	Controller gain Initial value: 0
lrTensCtrlResetTime	LREAL	Controller reset time - Unit: s - Initial value: 0 (reset time deactivated)
eResetCtrlMode	L_TW2P_ResetCtrlMode	Mode for setting the function for resetting the correcting variable from the I component via the <i>xResetCtrl</i> input = TRUE Initial value: 1
		0 Instant With a rising edge <i>xResetCtrl</i> = TRUE, the correcting variable from the I component is set to 0 immediately (abruptly).
		1 Ramp With a rising edge <i>xResetCtrl</i> = TRUE, the correcting variable from the I component is ramped to zero via the <i>scPar.lrDancerPosCtrlResetTime</i> parameter.
lrTensCtrlLimitPos	LREAL	Limitation of the tensile force controller correcting variable (output of the controller) in positive direction - Unit: [N] - Initial value: 2147483648 (0.5×2^{32})
lrTensCtrlLimitNeg	LREAL	Limitation of the tensile force controller correcting variable (output of the controller) in negative direction - Unit: [N] - Initial value: -2147483648 (-0.5×2^{32})
lrIdentFricMaxSpdScaled	LREAL	Speed for friction identification - Unit: x 100 % (1 = 100 % = max. speed at the <i>lrWndSpdRef</i> output) - Initial value: 0.9
lrIdentFricAccDec	LREAL	Acceleration for friction identification - Unit: revs/s² - Initial value: 1
rFiltTimeIdentFricSpd	REAL	PT1 filter time for the winder shaft speed for friction identification - Unit: s - Initial value: 0.0
rFiltTimeIdentFricTrq	REAL	PT1 filter time for the winder shaft friction torque for friction identification - Unit: s - Initial value: 0.05
dwSelectFricCurve	DWORD	For friction identification (input <i>xFricCmps</i> = TRUE): Selection of the memory area in which the friction values identified are stored. A maximum of 4 characteristics can be stored. - Initial value: 1 1 ... 4: memory area 1 ... 4
		For friction compensation (input <i>xExecuteIdentFric</i> = FALSE → TRUE) Selection of the characteristic for friction compensation 0: Linear friction characteristic (adjustable via <i>lrFricCurveStartTrq</i> parameter) 1 ... 4: identified friction characteristic (memory area 1 ... 4)

Identifier	Data type	Description						
rFiltTimeActTensIn	REAL	PT1 filter time for the current tensile force (<i>IrActTensIn</i> input) - Unit: s - Initial value: 0.005						
IrHystLineVelFricComp	LREAL	Hysteresis for detecting the direction of line speed. Only used with friction compensation <i>xFricComp</i> = TRUE and a noisy line signal. Maximum amplitude for line speed noise. - Unit: mm/s - Initial value: 0						
IrActTensInGain	LREAL	Gain factor for the current tensile force (<i>IrActTensIn</i> input) Initial value: 1						
IrActTensInOffset	LREAL	Offset for the current tensile force (<i>IrActTensIn</i> input) Initial value: 0						
IrHystlineStandStill	LREAL	Hysteresis width for the line speed This stabilises the noisy line speed and the direction recognition. After detecting the maximum noise amplitude of the input signal, set this value. - Unit: mm/s - Initial value: 0						
rFiltTimeIdentMIInertiaSpd	REAL	PT1 filter time for the speed at the winder shaft during the identification of the moment of inertia - Unit: s - Initial value: 0.01						
rFiltTimeIdentMIInertiaTrq	REAL	PT1 filter time for the torque at the winder shaft during the identification of the moment of inertia - Unit: s - Initial value: 0.005						
IrIdentMIInertiaMaxSpd Scaled	LREAL	Maximum speed of the winder shaft during the identification of the moment of inertia - Unit: x 100 % (1.0 = 100 % = <i>IrWndSpdRef</i>) - Initial value: 0.2 (20 %)						
IrIdentMIInertiaMaxTrq Scaled	LREAL	Maximum torque of the winder shaft during the identification of the moment of inertia - Unit: x 100 % (1.0 = 100 %) reference variable: rated motor torque - Initial value: 0.2 (20 %)						
eAdaptSpdCtrlGainMode L_TW2P_AdaptSpdCtrlGainMode		Mode selection for adaptation of the speed controller gain. Initial value: 2 <table><tr><td>0</td><td>DiamToSquare; $VP = f(d^2)$</td></tr><tr><td>1</td><td>Diam; $VP = f(d)$</td></tr><tr><td>2</td><td>Inertia; $VP = f(J)$</td></tr></table>	0	DiamToSquare; $VP = f(d^2)$	1	Diam; $VP = f(d)$	2	Inertia; $VP = f(J)$
0	DiamToSquare; $VP = f(d^2)$							
1	Diam; $VP = f(d)$							
2	Inertia; $VP = f(J)$							
IrAdaptSpdCtrlLowLimit	LREAL	Lower limit of speed controller gain in the drive. The <i>IrSetSpdCtrlGainAdaptOut</i> adaptation value must not be smaller than the <i>scPar</i> value. <i>IrAdaptSpdCtrlLowLimit</i> . - Value range: 0 to 1 - Initial value: 1.0 (100 %)						
alrSpdCtrlGainAdaptX ARRAY[1..9] OF LREAL		Characteristic function for speed control gain. The X axis corresponds to the scaled moment of inertia. - Unit: x 100 % (1 = 100 % = parameter <i>IrMaxMIInertia</i>) - Initial values: [0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6]						
alrSpdCtrlGainAdaptY ARRAY[1..9] OF LREAL		Characteristic function for speed control gain. The Y axis corresponds to the gain factor of the speed controller. - Unit: x 100 % (1 = 100 %) - Initial values: [0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6]						

Identifier	Data type	Description
IrReducedGainWindow	LREAL	Area of the system deviation with a reduced gain / sensitivity. The <i>IrReducedGainWindow</i> parameter is used to set a tolerance range in which the system deviation is transmitted to the controller with a lower gain. The tolerance range is set above and below the setpoint. • Initial value: 0.0
IrReducedGain	LREAL	Gain of system deviation within the reduced sensitivity The <i>IrReducedGain</i> parameter serves to set the value to which the gain is to be reduced in the defined tolerance zone. This means, in the tolerance zone, the reduced gain is active (<i>IrReducedGain</i>). • Initial value: 0.0
eTensCtrlStopMode L_TW2P_TensionCtrlStopMode		Mode in which the winding process ("TENSIONCTRL" state) is stopped. • Initial value: 0
	0	Holding The axis is stopped via the deceleration (<i>IrHaltDec</i>) and the jerk (<i>IrJerk</i>).
	1	Move ABS Absolute travel/position: The axis travels to the target position (<i>IrPos_Dist</i>) with the velocity (<i>IrVel</i>), acceleration (<i>IrAcc</i>), deceleration (<i>IrDec</i>) and jerk (<i>IrJerk</i>).
	2	Move Rel Relative travel position: the axis is brought to a standstill with the velocity (<i>IrVel</i>), acceleration (<i>IrAcc</i>), deceleration (<i>IrDec</i>) and jerk (<i>IrJerk</i>) after the distance (<i>IrPos_Dist</i>) has been travelled.

6 State machine

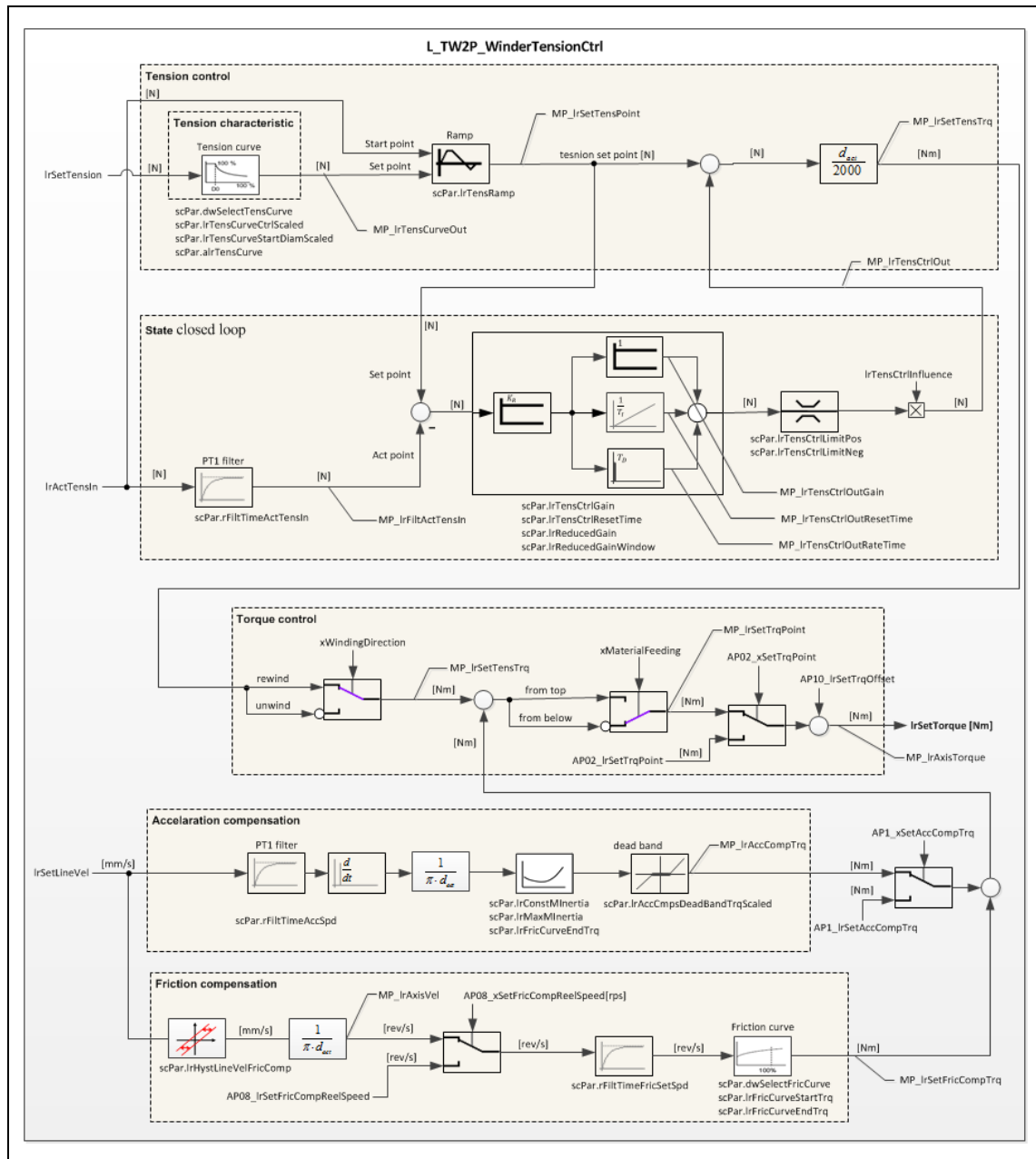


[6-1] State machine of the technology module

7 Signal flow diagram

Main signal flow of the function in torque-based operation

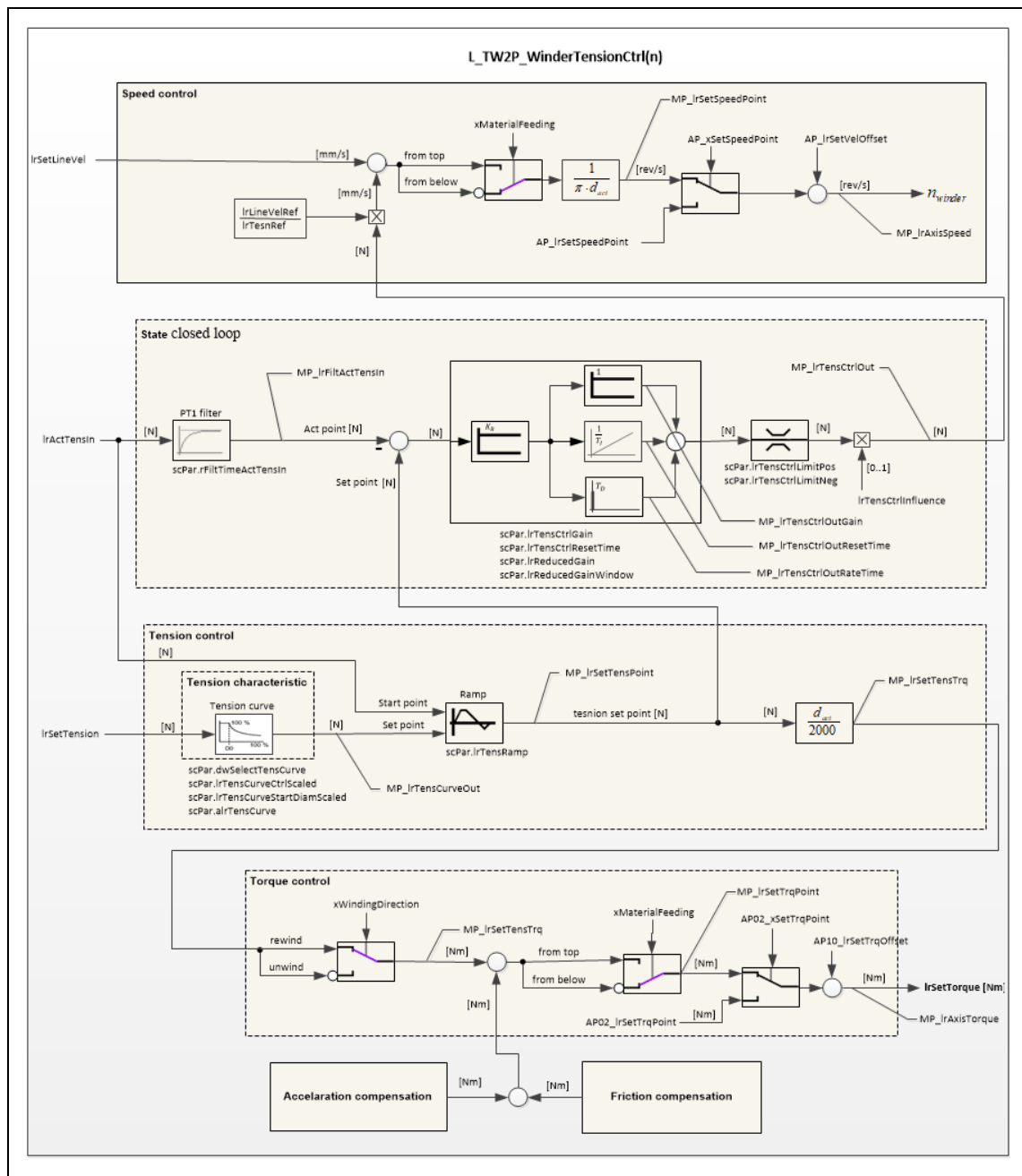
Setting: Parameter *eTensCtrlMode* = 0: "TorqueBased"



[7-1] Signal flow of the technology module in torque-based operation

Main signal flow of the function in speed-based operation

Setting: Parameter *eTensCtrlMode* = 1: "SpeedBased"



[7-2] Signal flow of the technology module in speed-based operation

7.1 Structure of the signal flow

L_TW2P_scSF_WinderTensionCtrlState

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

► [Signal flow diagram](#) (33)

Identifier	Data type	Description
MP_lrWndVelRef	LREAL	Nominal speed of the reel (output end) • Unit: unit/s
MP_lrWndSpeedRef	LREAL	Nominal speed of the reel (output end) • Unit: rev/s
MP_liRevCounter	LREAL	Counter content of the material length counter (whole numbers of complete reel revolutions)
MP_lrRevCounterResidual	LREAL	Counter content of the material length counter (residual value after one complete reel revolution) • Unit 1 x 100% • 100% = One reel revolution
MP_lrAxisSpeed	LREAL	Current speed of the reel (output end) • Unit: rev/s
MP_xDiamCalculated	LREAL	FALSE→TRUE edge: The diameter has been calculated. The measuring point is only set to TRUE for one PLC cycle
MP_lrSetDiamUnfiltered	LREAL	Current diameter calculated • Unit: mm
MP_lrSetDiamAdapted	LREAL	Diameter resulting from the lrSetDiam input. • Unit: mm
MP_lrSetDiamScaledOut	LREAL	Scaled diameter resulting from the lrSetDiam input. • Unit: 1x100% • 100% <i>lrMaxDiam</i>
MP_lrSetLineVelScaledOut	LREAL	Scaled line speed • Unit 1 x 100% • 100% <i>lrLineVelRef</i>
MP_lrActTotalTrqScaled	LREAL	Torque calculated from the measured current (scaled) • Unit x100% • 100% nominal torque
MP_lrTensCurveOut	LREAL	Tensile force setpoint evaluated using characteristic function. • Unit: N
MP_lrSetTensPoint	LREAL	Filtered tensile force setpoint with characteristic function evaluation • Unit: N
MP_lrSetTensTrq	LREAL	Torque calculated from the tensile force setpoint • Unit: Nm
MP_lrAccCompTrq	LREAL	Torque resulting from acceleration compensation • Unit: Nm
MP_lrSetFricCompTrq	LREAL	Torque resulting from friction compensation • Unit: Nm
MP_lrSetTrqPoint	LREAL	Resulting torque for reels • Unit: Nm
MP_lrSetUpperSpeedLimit	LREAL	Upper limit value for limiting reel speed • Unit: rev/s
MP_lrSetLowerSpeedLimit	LREAL	Lower limit value for limiting reel speed • Unit: rev/s
MP_lrAxisUpperSpeedLimit	LREAL	Upper limit value for limiting reel speed • Unit: units/s

Identifier	Data type	Description
MP_IrSetSpeedPoint	LREAL	Resulting speed setpoint • Unit: rev/s
MP_IrAxisLowerSpeedLimit	LREAL	Lower limit value for limiting reel speed • Unit: units/s
MP_IrAxisVel	LREAL	Reel speed (output end) • Unit: unit/s
MP_IrAxisTorque	LREAL	Reel torque (output end) • Unit: Nm
MP_IrSetTensScaledOut	LREAL	Scaled tensile force setpoint • Unit x 100% • 100% = <i>scPar.IrTensRef</i>
MP_IrSetTensTrqScaledOut	LREAL	Scaled torque calculated from the tensile force setpoint • Unit x 100% • 100% = Rated torque
MP_IrSetFricTrqScaledOut	LREAL	Scaled torque pre-control value of friction compensation • Unit x 100% • 100% = Nominal torque
MP_IrAccTrqScaledOut	LREAL	Scaled torque pre-control value of acceleration compensation • Unit x 100% • 100% = Rated torque
MP_IrFiltActTensIn	LREAL	Filtered tensile force from the <i>IrActTensIn</i> input
MP_IrTensCtrlOut	LREAL	Correcting variable from tensile force controller
MP_IrTensCtrlOutGain	LREAL	Correcting variable of the proportional component (P component) of the tensile force controller • Unit: N
MP_IrTensCtrlOutResetTime	LREAL	Correcting variable of the proportional component (I component) of the tensile force controller • Unit: N
MP_IrTensCtrlOutRateTime	LREAL	Correcting variable of the proportional component (D component) of the tensile force controller • Unit: N
MP_IrTensCtrlOutScaled	LREAL	Scaled correcting variable from tensile force controller • Unit: x100% • 100% = <i>IrTensRef</i>
MP_IrSetTensCtrlValue	LREAL	Setpoint for the PID tensile force controller • Unit: N
MP_IrActTensCtrlValue	LREAL	Actual value for the PID tensile force controller • Unit: N
MP_IrActTensScaled	LREAL	Scaled current tensile force • Unit: x 100% • 100% <i>IrSetTens</i>
MP_xLineStandStill	BOOL	TRUE Standstill of the line has been detected.

7.2 Structure of the access points

L_TW2P_scAP_WinderTensionCtrlState

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

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

Identifier	Data type	Description
AP_xSetVelOffset	BOOL	Access point AP_IrVelOffset enabled
		TRUE Enable access point <i>AP_IrVelOffset</i>
AP_IrVelOffset	LREAL	Cyclic specification of the offset for the velocity of the winder axis with regard to the winding shaft (gearbox output side) • Unit: units/s The offset value is set immediately and abruptly without ramp generator!
AP_xSetTrqOffset	BOOL	Access point <i>AP_IrSetTrqOffset</i> enabled
		TRUE The access point overwrites the values at the access point in the signal flow.
AP_IrSetTrqOffset	LREAL	Cyclic specification of the offset for the torque of the winder axis with regard to the winding shaft (gearbox output side) • Unit: Nm The offset value is set immediately and abruptly without ramp generator!
AP_xSetVelOffsetDiamCalc	BOOL	Access point <i>AP_IrVelOffsetDiamCalc</i> enabled
		TRUE The access point overwrites the values at the access point in the signal flow
AP_IrVelOffsetDiamCalc	LREAL	Cyclic specification of the offset for the velocity of the winder axis relative to the winding shaft (gearbox output side) for calculating the diameter • Unit: units/s The offset value is set immediately and abruptly without ramp generator!
AP_xSetSpeedPointDiamCalc	BOOL	Access point <i>AP_IrSetSpeedPointDiamCalc</i> enabled
AP_IrSetSpeedPointDiamCalc	LREAL	Cyclic loading of speed setpoint for the drive • Unit: rev/s The offset value is set immediately and abruptly without ramp generator!
AP_xSetAccCompTrq	BOOL	Access point <i>AP_IrAccCompTrq</i> enabled
AP_IrAccCompTrq	LREAL	Resulting torque setpoint from the acceleration compensation of the winding drive • Unit: Nm
AP_xSetTrqPoint	BOOL	Access point <i>AP_IrSetTrqPoint</i> enabled
AP_IrSetTrqPoint	LREAL	Alternative torque setpoint for the torque feedforward control. • Unit: Nm
AP_xSetUpperSpeedLimit	BOOL	Access point <i>AP_IrSetUpperSpeedLimit</i> enabled
AP_IrSetUpperSpeedLimit	LREAL	Upper limit value for the speed limitation of the winding drive • Unit: revs/s
AP_xSetLowerSpeedLimit	BOOL	Access point <i>AP_IrSetLowerSpeedLimit</i> enabled
AP_IrSetLowerSpeedLimit	LREAL	Lower limit value for the speed limitation of the winding drive • Unit: revs/s

Identifier	Data type	Description
AP_xSetFricCompReelSpeed	BOOL	Access point <i>AP_IrSetFricCompReelSpeed</i> enabled
AP_IrSetFricCompReelSpeed	LREAL	Selection of the speed for the friction compensation • Unit: revs/s The corresponding torque for friction compensation is set in the unit [Nm] based on the speed <i>AP_IrSetFricCompReelSpeed</i> .

8 Description of the basic functions

Overview

- [Manual jog \(jogging\)](#) (40)
- [Homing](#) (42)
- [QSP application](#) (45)
- [Halt](#) (46)
- [Absolute positioning](#) (47)
- [Relative positioning](#) (49)
- [Continuous travel with a defined velocity](#) (52)
- [Aborting and completing a positioning process](#) (55)
- [Override for speed](#) (57)

8 Description of the basic functions

8.1 Manual jog (jogging)

8.1 Manual jog (jogging)

Prerequisite

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

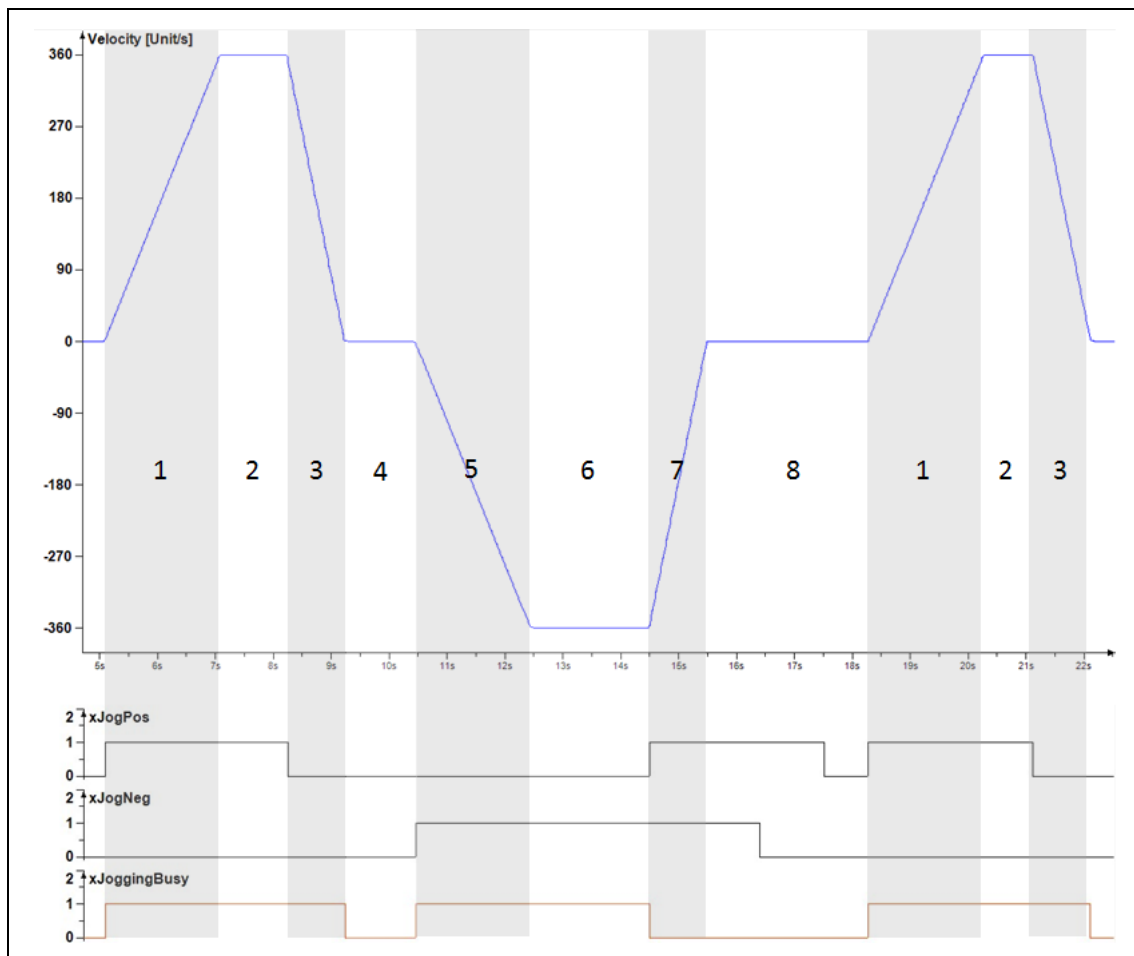
Execution

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

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

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

Travel diagram



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

Parameters to be set

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

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

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

8.2**Homing****Prerequisites**

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

Execution

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

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

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

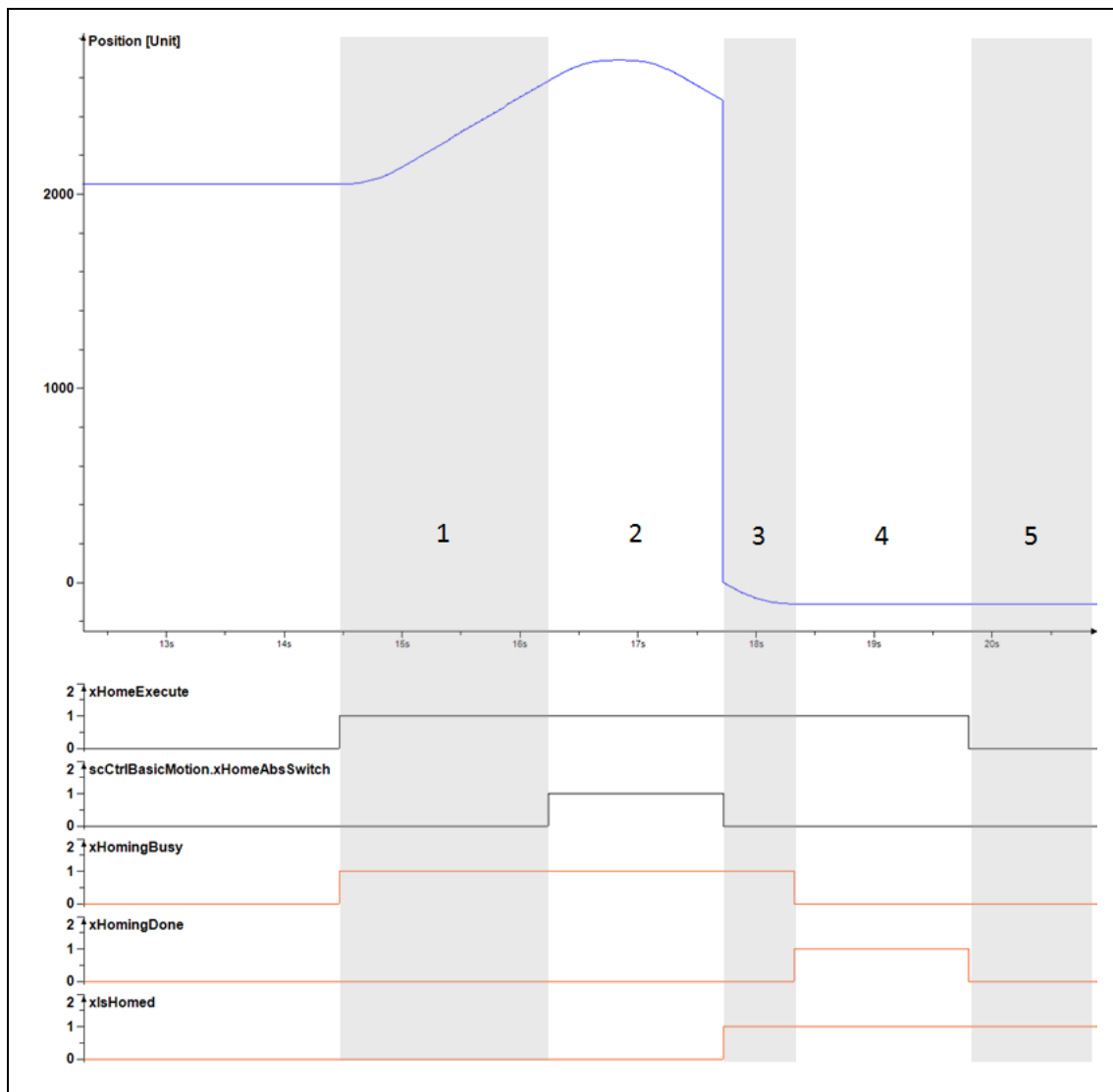
If *xHomeExecute* = FALSE, the state is changed to "STANDBY" and *xHomingDone* is set = FALSE.

Abort of the homing process

The function is aborted via the *xHalt* input.

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

Travel diagram



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

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

Parameters to be set

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

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

8.3 QSP application

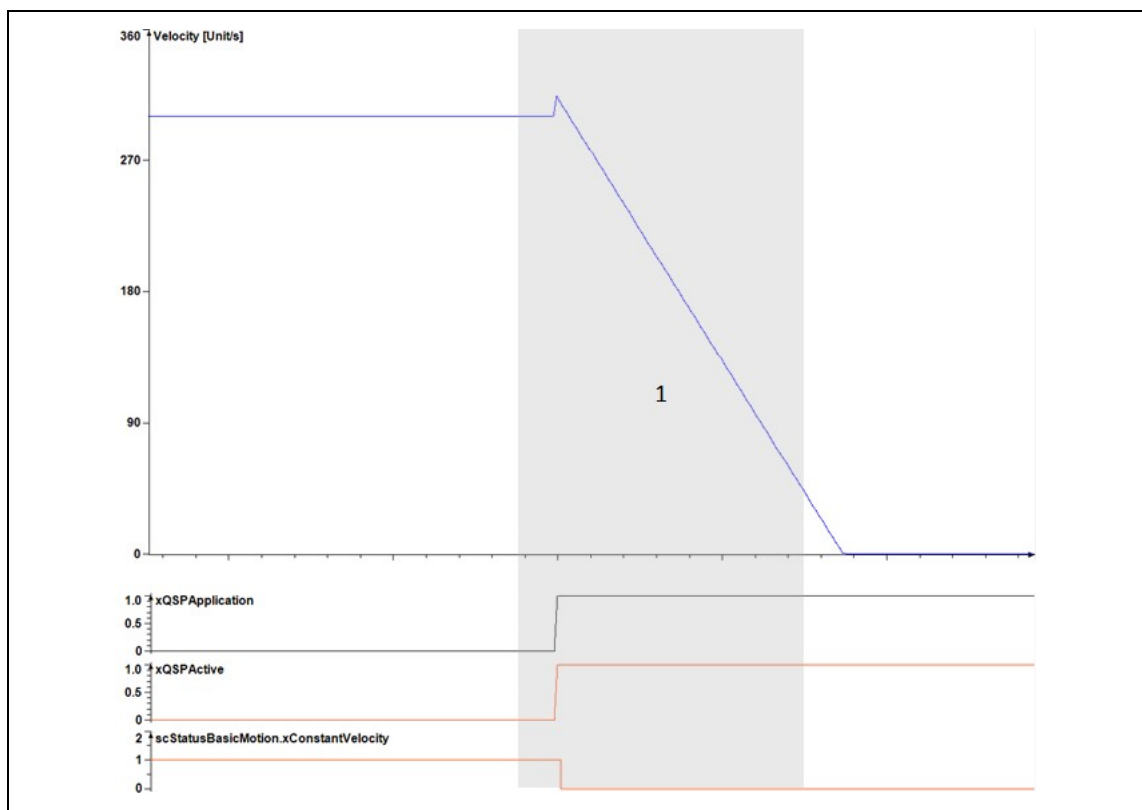
Prerequisite

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

Execution

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

Travel diagram



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

Parameters to be set

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

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

8.4 Halt

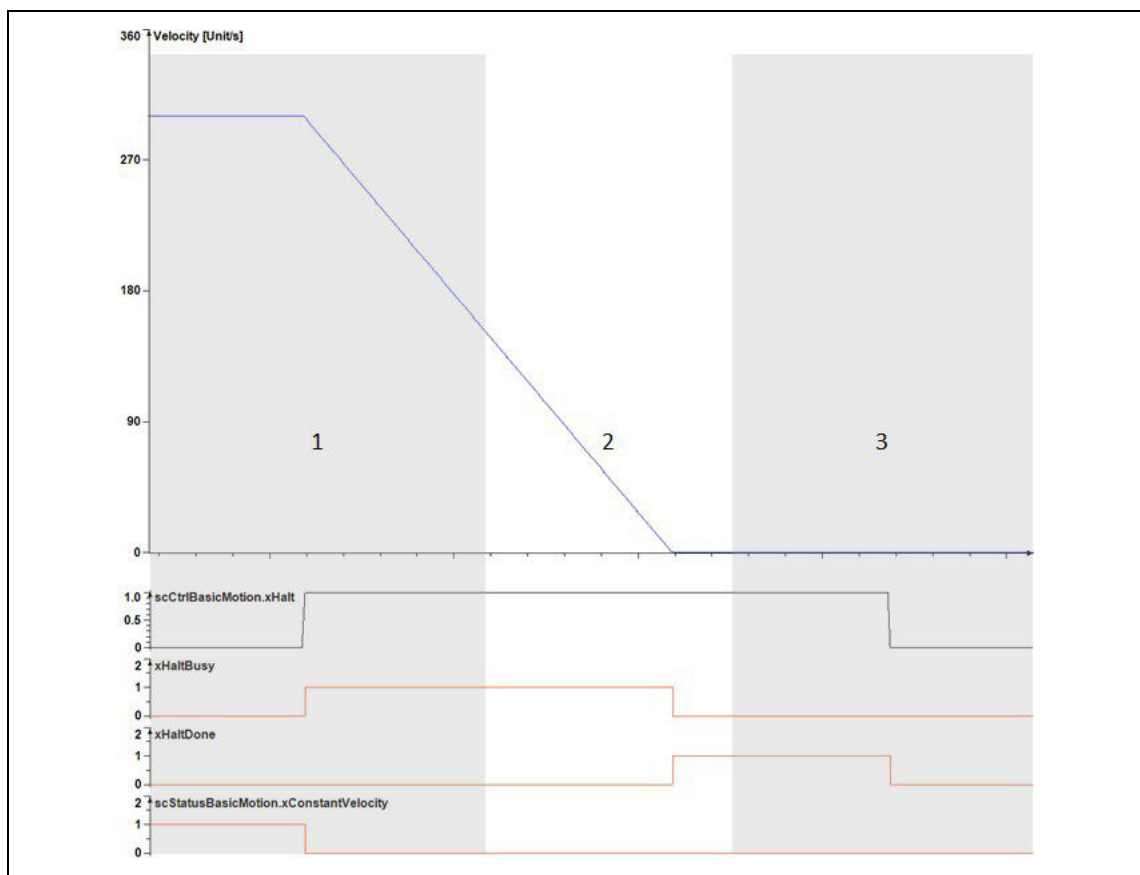
Prerequisite

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

Execution

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

Travel diagram



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

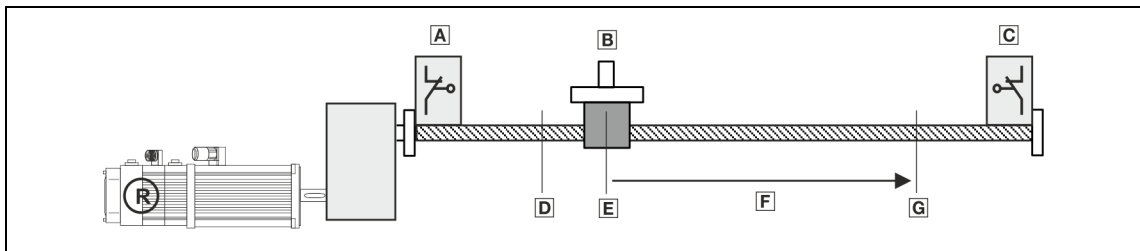
Parameters to be set

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

8 Description of the basic functions

8.5 Absolute positioning

8.5 Absolute positioning



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

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

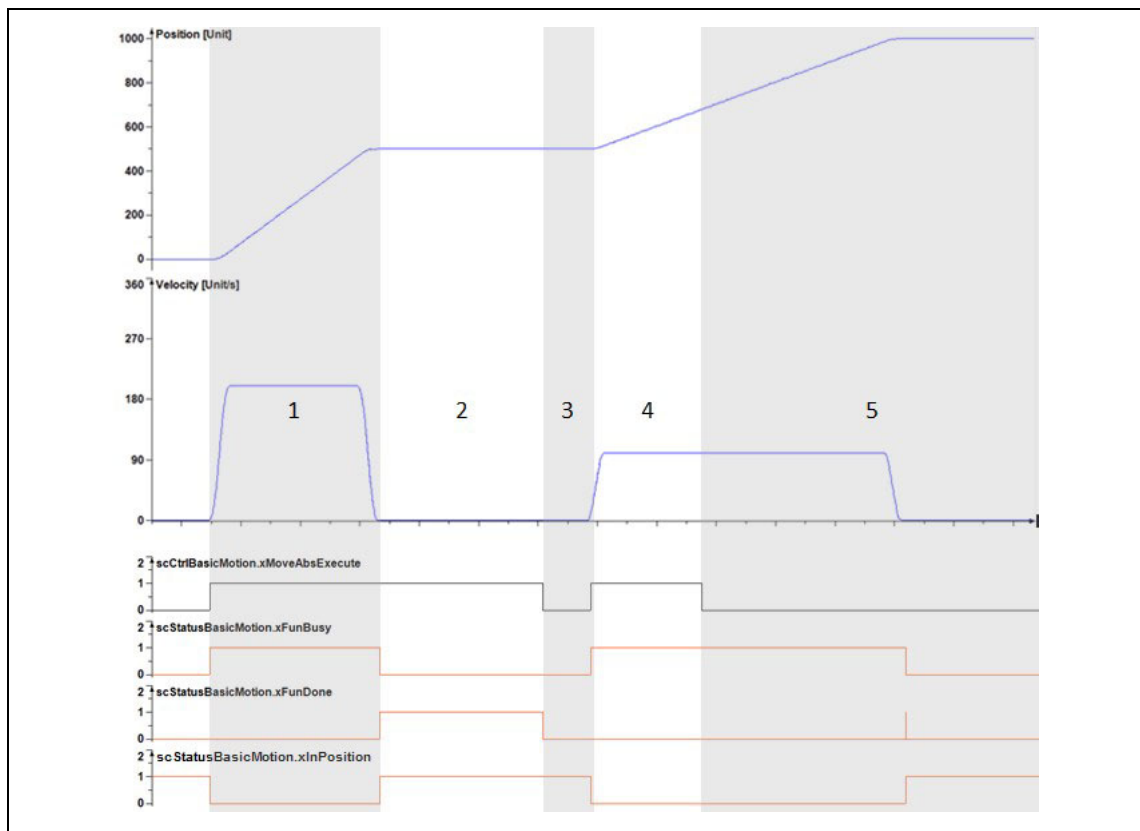
In order to reach the target position, the axis is brought to a standstill via the *IrDec* parameter. The axis reaches a standstill exactly in the target position.

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

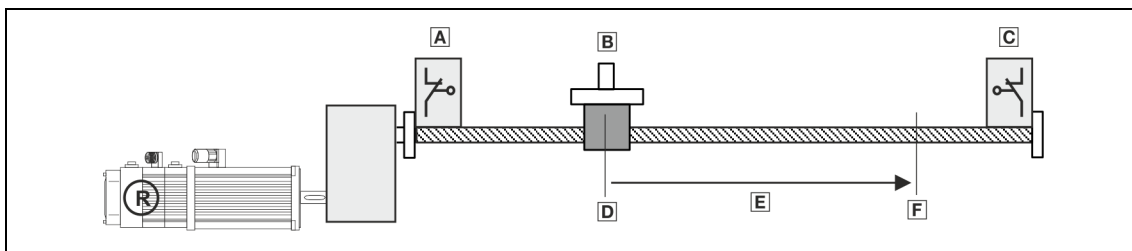
- *IrVel*, initial value: 3600 [units/s]
- *IrAcc*, initial value: 36000 [units/s²]
- *IrDcc*, initial value: 36000 [units/s²]
- *IrJerk*, initial value: 0 [units/s³]

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.

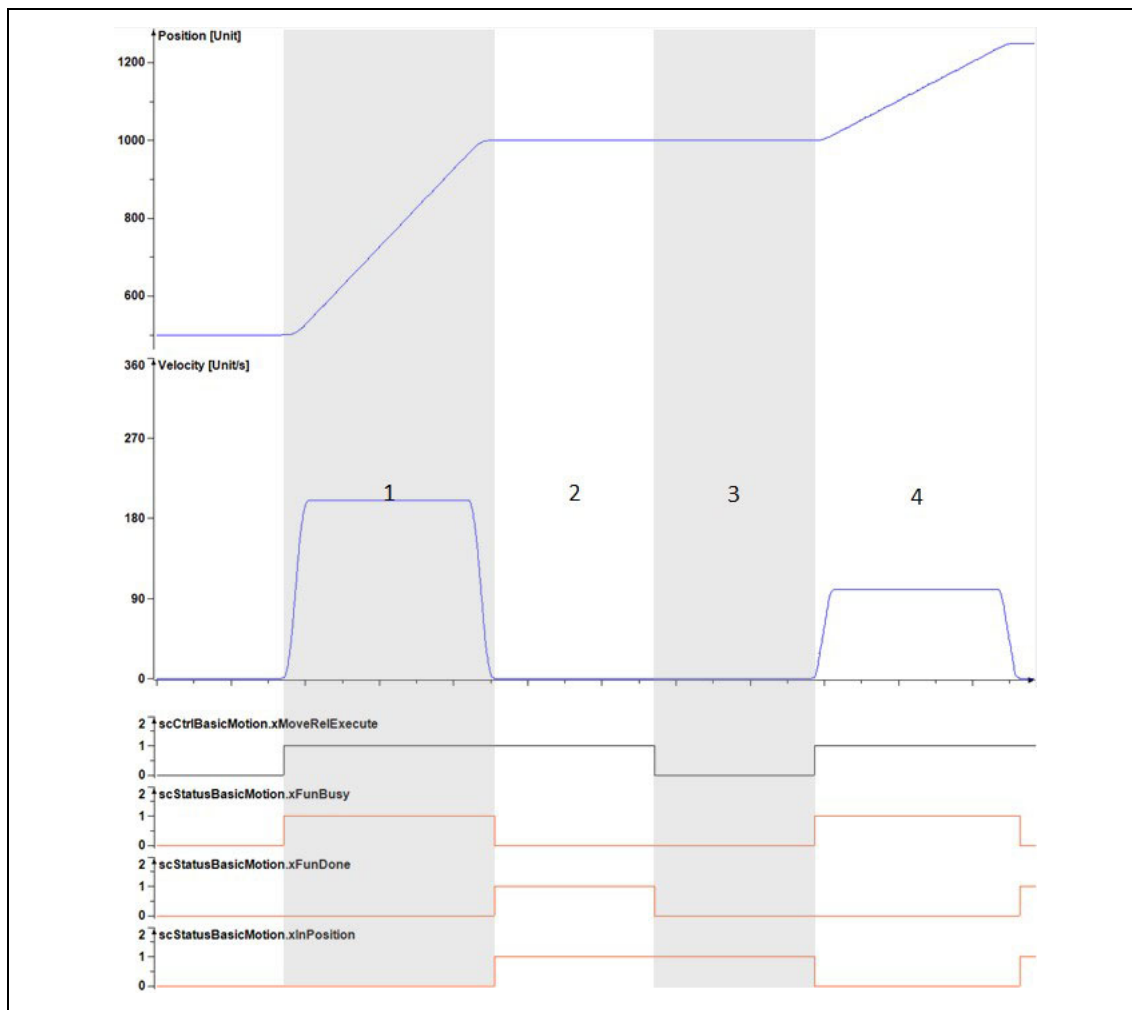
In order to reach the target position, the axis is brought to a standstill via the *lrDec* parameter. The axis reaches a standstill exactly in the target position.

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.

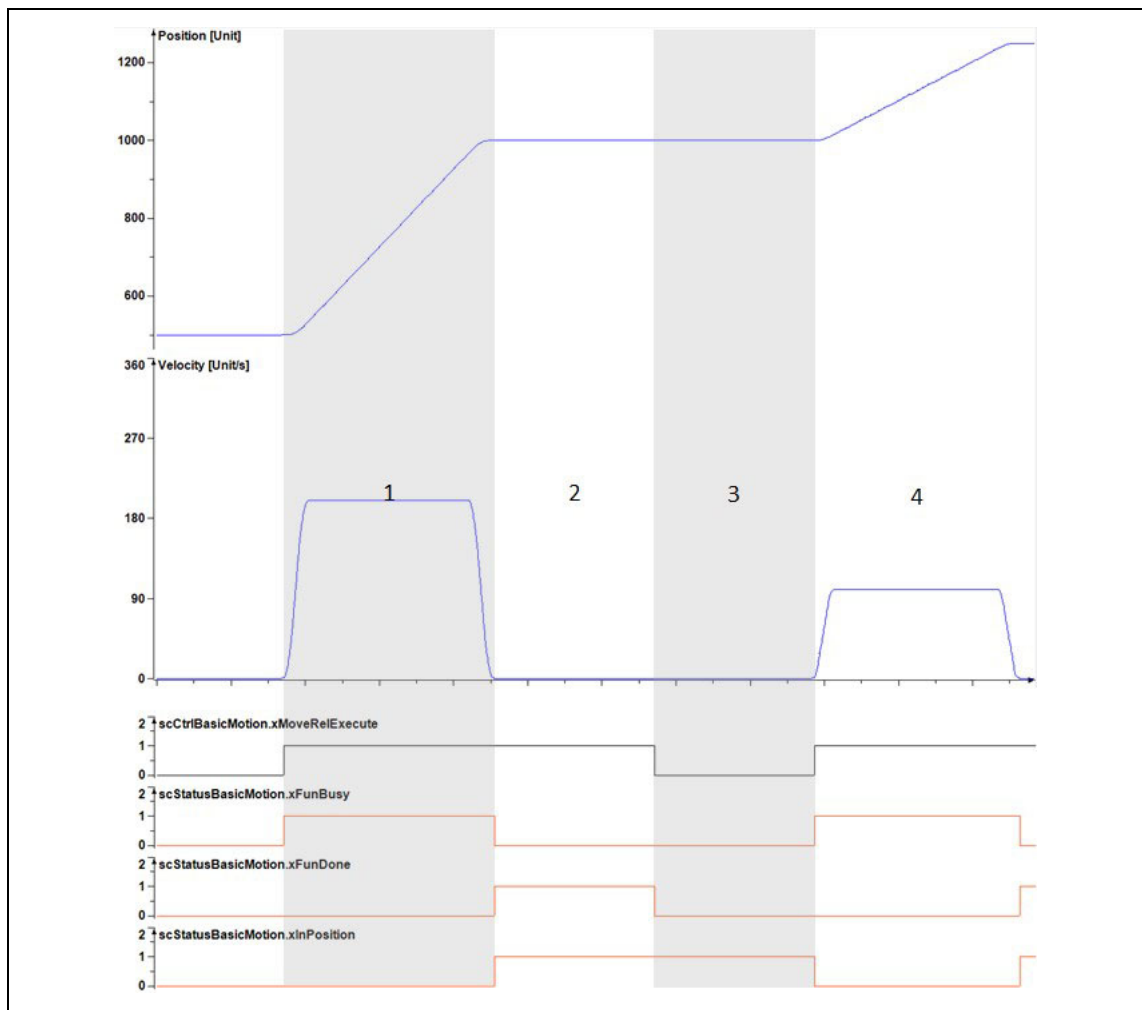
When a hardware limit switch has been reached, the axis is brought to a standstill via the QSP application deceleration ramp.

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**

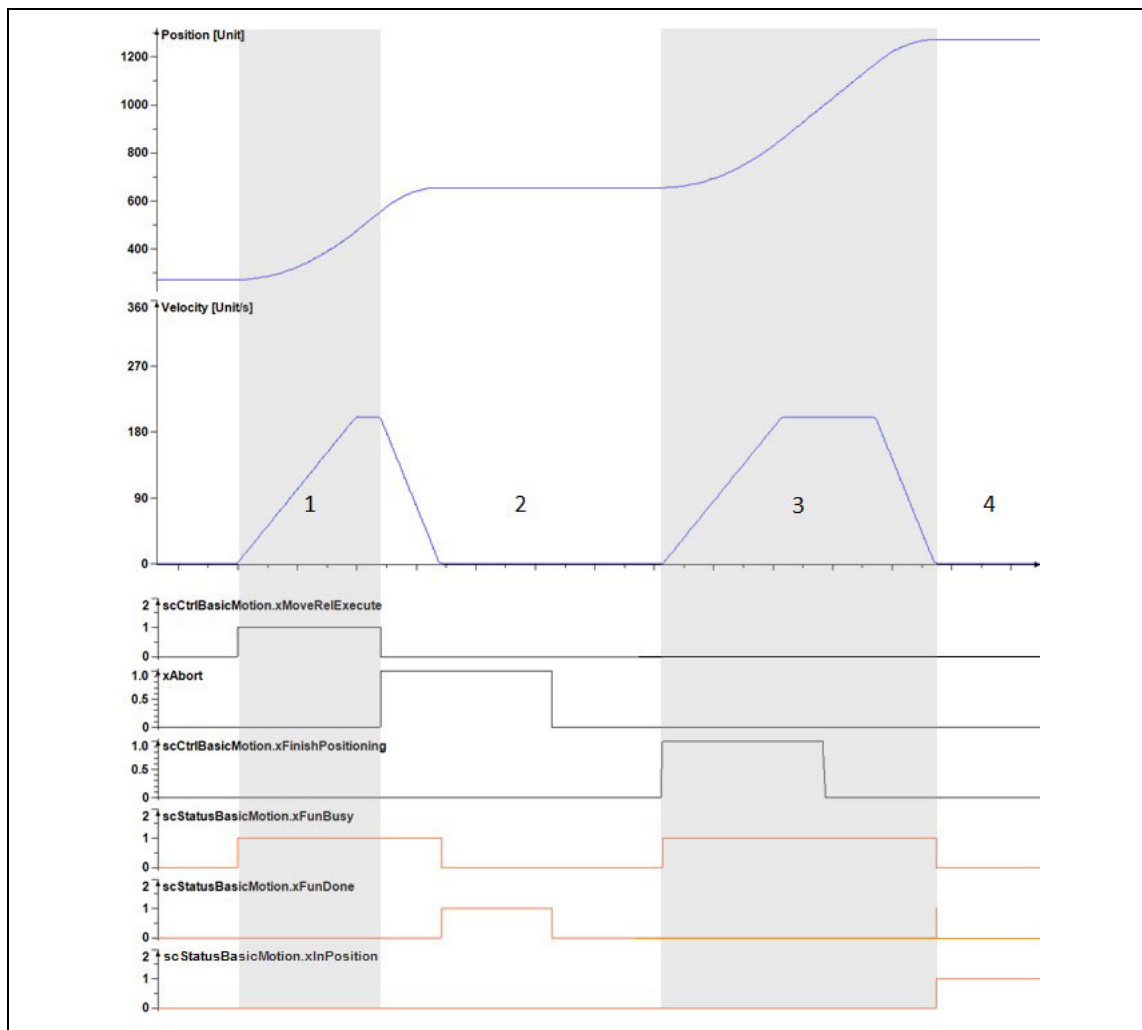
Both technology modules provide several profile data sets, one of which can be active.

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:

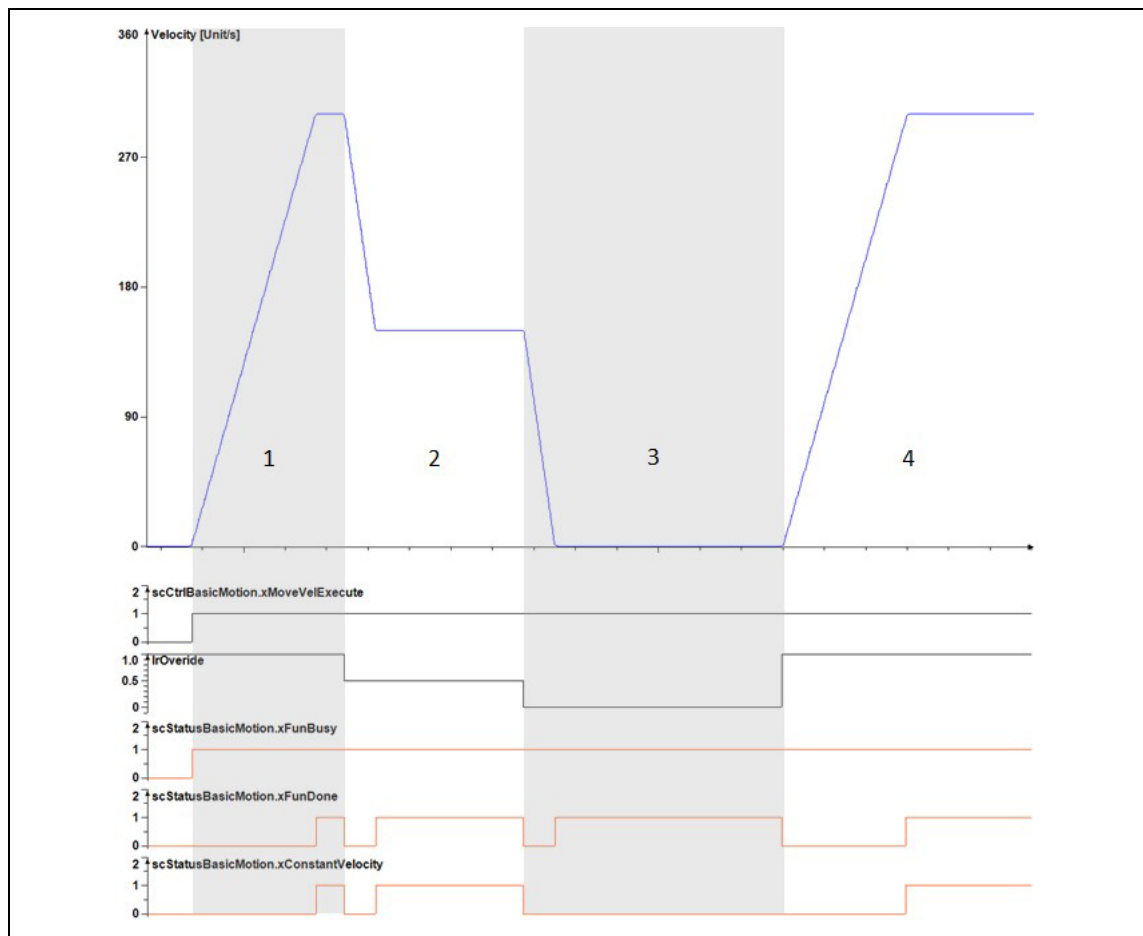
- *IrOverride*, initial value: 1.0
- *scCtrlBasicMotion.IrOverrideAcc*, initial value: 1.0 (100 %)
- *scCtrlBasicMotion.IrOverrideJerk*, initial value: 1.0 (100 %)

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 function

9.1 Defining the winding direction (winding/unwinding)

In order that the feedforward control values, the disturbance compensation and the correcting signal of the dancer position controller always act in the required direction, the "normal winding direction" must be defined once.

The *xWindingDirection* input serves to set whether the winding drive is to operate as unwinder or rewinder with regard to the normal material flow direction with positive line velocity.

- *xWindingDirection* = TRUE: Unwinder (material is unwound.)
- *xWindingDirection* = FALSE: Rewinder (material is rewound.)

⚠ WARNING!

Injury to persons

Risk of injury is caused by changing the setting at input *xWindingDirection* during a movement. This can lead to a speed jump of the winder axle.

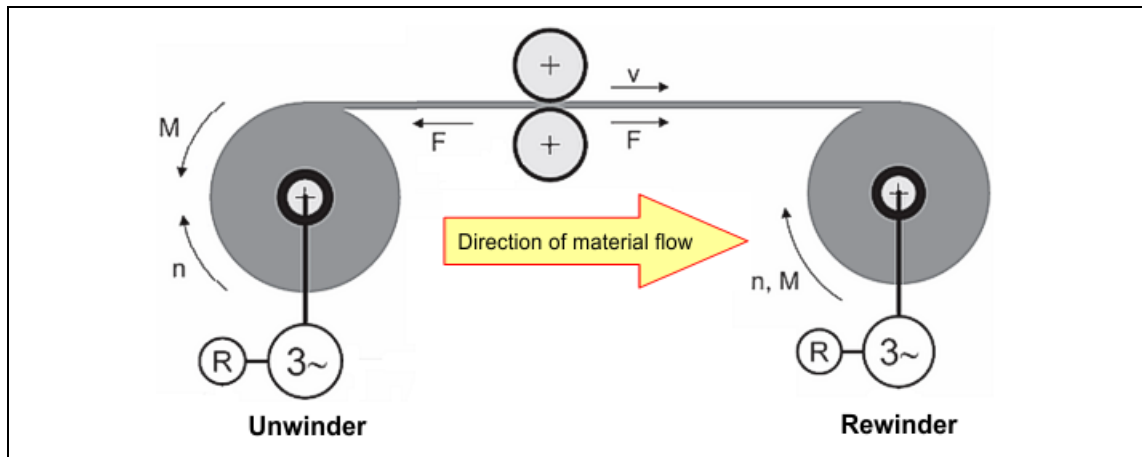
Possible consequences

Death or severe injuries

Protective measures

Only change the setting of input *xWindingDirection* if the technology module is not running and the inverter is inhibited:

- Input *xEnable* = FALSE
- Input *xEnableOperation* = FALSE



[9-1] Effective direction of speed and torque as a function of the material flow

Automatic detection of the winding direction

After defining the winding direction, the winding drives can be actuated with a negative line speed even in the opposite direction. A modification of the signal flow is not required if the direction of the material flow is inverted. The current winding direction is shown at the *xUnwind* output.

**Tip!**

Special case: winding direction detection with line at a standstill

- Unwinder: input *xWindingDirectionStandstill* = TRUE
- Rewinder: input *xWindingDirectionStandstill* = FALSE

9.2 Defining the material feed to the winder

The *xMaterialFeeding* input serves to define whether the material is fed to the winder from the top or the bottom.

► [L_TW2P_WinderTensionCtrlState function block](#) (18)

⚠ WARNING!

Injury to persons

Risk of injury is caused by changing the setting at input *xMaterialFeeding* during a movement. This can lead to a speed jump of the winder axle.

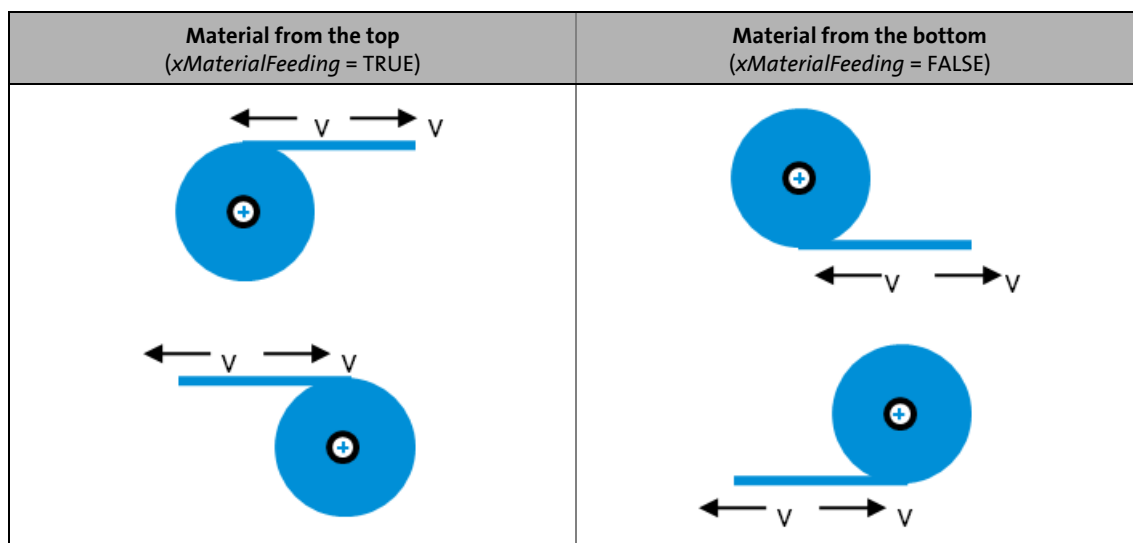
Possible consequences

Death or severe injuries

Protective measures

Only change the setting of input *xMaterialFeeding* if the technology module is not running and the inverter is inhibited:

- Input *xEnable* = FALSE
- Input *xEnableOperation* = FALSE



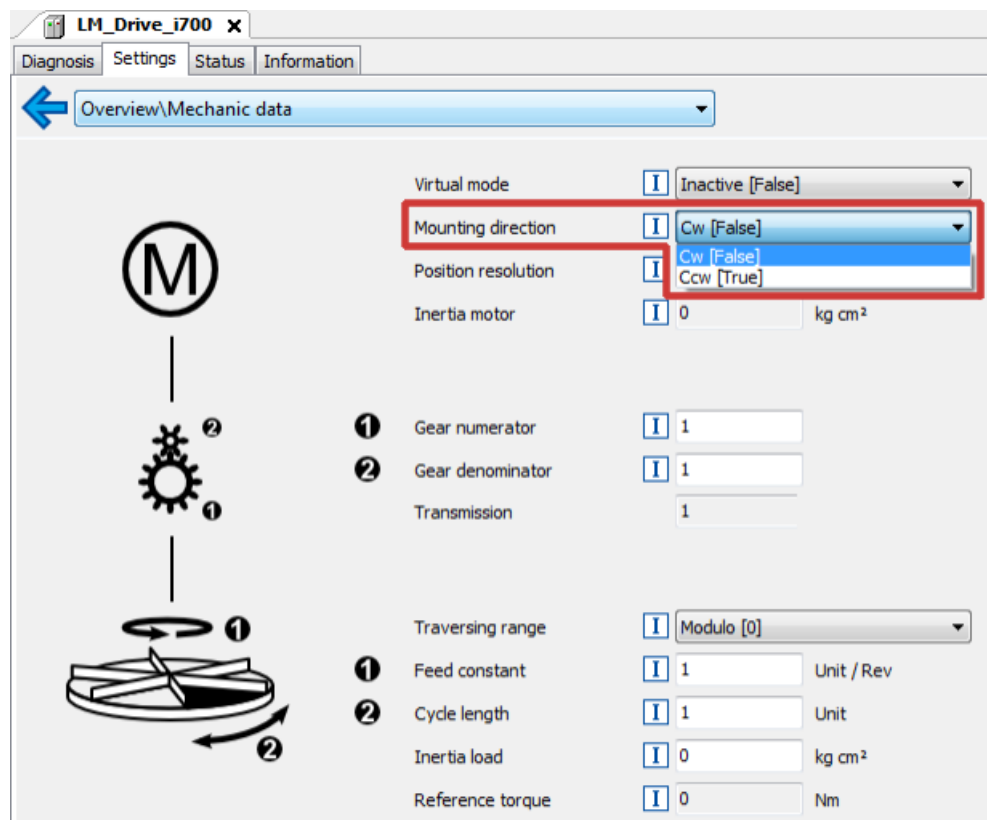
In principle, the rotation direction of the winder shaft is adapted to the material flow through the mounting direction of the motor.

The parameters for the direction of rotation of the winder shaft are located in the parameter structure

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

Example

The direction of rotation of the axis can be set in the »PLC Designer« under the **Settings Configuration** tab:



9 Description of the technology function

9.3 Master value source for the diameter calculation

9.3 Master value source for the diameter calculation

The technology module always operates with the line speed at the *lrSetLineVel* input.

The line speed is expected in the unit [mm/s].

9.4

Diameter calculation

The current diameter is calculated by dividing the line velocity by the winder speed and the figure π :

Calculation of the current diameter		
$d_{act} = \frac{v_{Line}}{n_{Winder} \times \pi}$		
Symbol	Description	Dimension unit
d_{act}	Current diameter	mm
v_{Line}	Line velocity	mm/s
n_{Winder}	Winder speed	rev/s

In fact, however, for the calculation integrated speed values are used instead of instantaneous values. This causes a average determination. The number of revolutions causing a recalculation of the diameter is specified via the parameter *IrDiamCalcRegularDist*. The initial value of this parameter is set to 1 winding shaft revolution.

Averaging can be extended to greater distances for thin materials and untidy reels. A lower calculation distance is specified in the *IrDiamCalcReducedDist* parameter. In this case, the initial value is 1/10 of a winder shaft rotation.

For quick diameter changes of *IrDiamCalcRegularDist*, it can be switched to the fast calculation mode by setting the *xDiamCalcReduced* input = TRUE. The lower calculation distance is set with the *IrDiamCalcReducedDist* parameter. Here, the initial value 1/10 is defined for the winder shaft revolution.

The smaller calculation distance is also activated automatically when a starting diameter is loaded. This state remains until a new diameter has been calculated. The function is required if the real diameter of the reel may deviate significantly from the loaded diameter. Thus, the winder shaft only rotates for a short distance with the "wrong" diameter. After the diameter has been calculated, a suitable value is available again.

The minimum (*Min. diameter*) and maximum diameter (*Max. diameter*) must be specified in the unit [mm] as the basic parameter. These values function as limits. Furthermore, all percentage or scaled diameter values refer to *Max. diameter*.

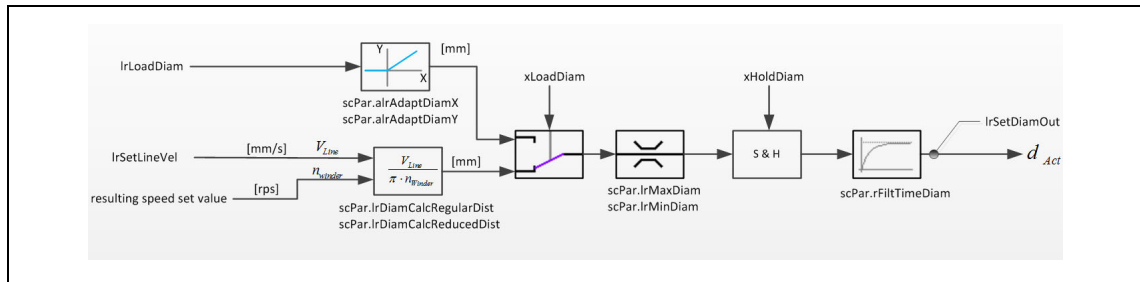
Parameters to be set

The parameters for the diameter calculation are located in the parameter structure, see extract:

Identifier	Data type	Description
IrMaxDiam	LREAL	Maximum diameter • Unit: mm • Initial value: 180
IrMinDiam	LREAL	Minimum diameter • Unit: mm • Initial value: 50
IrDiamCalcRegularDist	LREAL	Regular calculation distance for the diameter • Unit: rev • Initial value: 1
IrDiamCalcReducedDist	LREAL	Reduced calculation distance for the diameter • Unit: rev • Initial value: 0.1

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

Signal flow for calculating the diameter



9.5 Holding the diameter

For some operating states of the winder, in which the line velocity does not correspond to the circumferential velocity of the reel, the current diameter cannot be calculated from the line velocity and the motor speed. In this case, the calculation of new values must be prevented and the diameter must be held at the old value.

If the diameter is held, the *xHoldDiamActive* output is = TRUE

This is done automatically under the following conditions:

- Line speed < minimum line speed
 - *lrMinLineVel* [mm/s] from the *scPar* parameter structure
- In the "STOP", "ERROR", "HOMING", "JOGGING" and "SYNCLINEVEL" states.

For the user holding of the diameter, set the *xHoldDiam* input to TRUE.

If the actual line speed is noisy, the "Hold diameter" function can toggle.

Recommendation: Use the hysteresis function via the parameter *lrHystlineStandstill*.

Via the parameter *lrHystlineStandstill* the hysteresis width is defined. The current line speed is evaluated via the hysteresis. The actual speed resulting from the hysteresis function is compared with the minimum line speed.

Parameters to be set

Identifier	Data type	Description
<i>lrMinLineVel</i>	LREAL	Minimum line velocity Up to this velocity, the diameter is held. • Unit: mm/s • Initial value: 1
<i>xHoldDiam</i>	BOOL	Hold the current diameter Initial value: FALSE
		TRUE Current diameter is held.
		FALSE The current diameter is not held.
<i>lrHystlineStandStill</i>	LREAL	Hysteresis width for the line speed This stabilises the noisy line speed and the direction recognition. After detecting the maximum noise amplitude of the input signal, set this value. • Unit: mm/s • Initial value: 0

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

9.6 Defining the diameter (sensor)

At the start of a winding process, it may be required to define a start diameter or use the signal of a diameter sensor to be able to properly start the setpoint generation for the winding drive and also the interference compensation.

The *IrLoadDiam* input serves to define a start diameter that is accepted with the highest priority and loaded cyclically when *xLoadDiam* = TRUE.

An external diameter value, e.g. of an ultrasonic sensor, can also be connected to the *IrSetDiam* input. This analog value can be adapted via a curve function $Y = f(x)$. The curve function is set with nine grid points via the parameters *alrAdaptDiamX*[1...9] and *alrAdaptDiamY*[1...9]. In order that the analog value is used as start diameter, the adapted curve progression is initialised with *alrAdaptDiamY* = *alrAdaptDiamX*. The sensor signal can also be loaded permanently.

In practice, ultrasonic sensors with an analog output variable (0 ... 10 V) are often used for recording a diameter. A curve function $Y = f(x)$ serves to compensate for potential nonlinearities of such a sensor. The function acts on the value which can be diagnosed via the *MP_IrSetDiam* measuring point. The curve function is set using nine data points via the *scPar.alrAdaptDiamX* and *scPar.alrAdaptDiamY* parameters. The value is not adapted when the Lenze setting of the curve function is used. The start diameter is passed through 1:1.

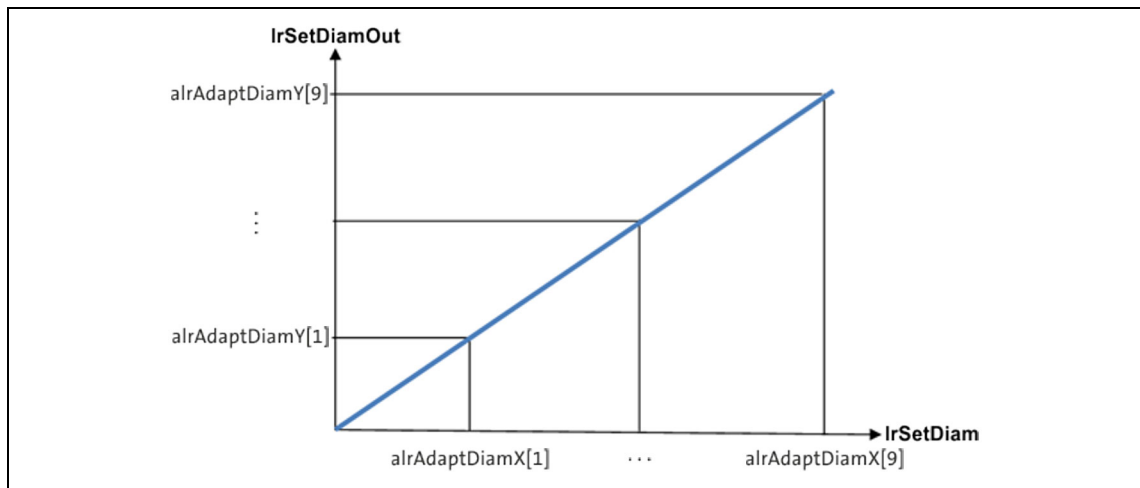
Parameters to be set

Identifier	Data type	Description
<i>IrLoadDiam</i>	LREAL	Defining a (start) diameter The diameter is loaded cyclically when the <i>xLoadDiam</i> is set to TRUE. • Unit: mm • Initial value: 0
<i>alrAdaptDiamX</i> ARRAY [1...9] OF LREAL		Grid points of the curve function for diameter loading • Values that may be applied to the analog input <i>IrSetDiam</i> . • Unit: mm • Initial values: 0, 100, 200, 300, 400, 500, 600, 700, 800
<i>alrAdaptDiamY</i> ARRAY [1...9] OF LREAL		Grid points of the curve function for diameter loading • Function values for the diameter • Unit: mm • Initial values: 0, 100, 200, 300, 400, 500, 600, 700, 800

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

Example: Loading of a diameter via a curve function**NOTICE**

If the real diameter for loading is not known (no sensor, no operator input possible), a mean diameter should be loaded.



9.7

Material length counter**Enable of the material length counter**

The material length counter is activated with the *xEnable* input = TRUE.

The material length is calculated by integrating the line speed at the *IrSetLineVel* input and is shown at the *IrMaterialCounter* output (in millimetres). The material length is incremented or decremented depending on the winding direction.

► [Defining the winding direction \(winding/unwinding\)](#) (📖 59)

For the analog signal of the line speed, the current value of the material length can be filtered with a PT1 characteristic. The filter time is set with the *rFiltTimeMaterialCounter* parameter.

The current value of the material length is stored in the persistent data.

► [L_TW2P_PersistentVarWinder](#) (📖 91)

Preloading the material length counter

For the initialisation of the material length, an initial material length can be set via the *IrSetMaterialPos* parameter. With a FALSE→TRUE edge at the *xSetMaterialCounter* input, the initial material length is accepted with the highest priority.

Parameters to be set

Identifier	Data type	Description
IrSetMaterialPos	LREAL	Position of the material length counter With a FALSE→TRUE edge at the <i>xSetMaterialCounter</i> input, the material length counter (<i>IrMaterialCounter</i> output) is set to the value in <i>IrSetMaterialPos</i> . • Unit: mm • Initial value: 0
rFiltTimeMaterialCounter	REAL	Filter time constant for the material length counter (<i>IrMaterialCounter</i> output) • Initial value: 0 (filter is deactivated.)

► [L_TW2P_scPar_WinderTensionCtrlState](#) (📖 26)

Sources for the material length counting

The material length can be counted from one of three different sources, and the counting procedure can be carried out in two different ways.

Source	Selection of the source		INFO
	Connection to <i>MaterialCounterAxis</i>	<i>xLineVelDiamCalc</i>	
IrSetLineVel	no	FALSE	Material length counter via 'IrSetLineVel' input
MaterialCounterAxis	Yes	is not evaluated	Material length counter via 'MaterialCounterAxis' reference axis

9.7.1 Material length counter via 'IrSetLineVel' input

Conditions

- No reference axis is connected to the *MaterialCounterAxis* input.
- *xLineVelDiamCalc* parameter = FALSE

Operating mode

In addition to the position (*IrSetMaterialPos*), the line speed at the *IrSetLineVel* input is integrated into the material length count.

The resulting value is shown as material length at the *IrMaterialCounter* output and stored persistently.

With a FALSE→TRUE edge at the *xSetMaterialCounter* input, the material length from the *IrSetMaterialPos* parameter is loaded, setting the material length at the *IrMaterialCounter* output directly to the value of *IrSetMaterialPos*. The further count is added to the value of the material length set at the output.

NOTICE

In the case of a noisy signal, the material counting process is falsified by integration of the line speed. In this process, the material length counter shows a drift, even if the line is stable.

9.7.2 Material length counter via 'MaterialCounterAxis' reference axis

Conditions

A reference axis (modulo axis) is connected to the *MaterialCounterAxis* input.

Operating mode

The loss-free number of revolutions determined is used as a basis for the material length counting. This procedure is suitable for noisy signals!

In this process, whole revolutions and the fractional part of a revolution are recorded. Via the feed constant of the reference axis (modulo axis), the material length is shown at the *lrMaterialCounter* output.

The number of revolutions counted can be read out via the *MP_liRevCounter* measuring point. The fractional amount of a revolution is shown via the *MP_lrRevCounterResidual* measuring point. The values of these measuring points are persistently stored.

With a FALSE↗TRUE edge at the *xSetMaterialCounter* input, the material length is loaded from the *lrSetMaterialPos* parameter. In this process, the material length is converted to the number of revolutions and stored via the feed constant of the axis.

The material length from the *lrSetMaterialPos* parameter is shown at the *lrMaterialCounter* output.

NOTICE

An exact material length counting process can only be carried out with a slip-free measuring wheel. A measuring wheel that is prone to slip on the material causes errors in the material length counting.

9.8 Inching mode with 1/d evaluation



Tipp!

The limitation function "Safe speed" (SLS1, SLS2 ...) can limit the speed from the function "Manual jog (jogging)" (*xLineJogPos*, *xLineJogNeg*), if the parameter "160" (Follower SLS Reaction) of the winder axis is set to TRUE.

With this mode, manual inching of the line is executed. The function differs with regard to the **Manual jog** BaseMotion function. The feed constant in [mm/rev] of the winder is not constant, since the circumference of the winder changes with the current diameter. The actual speed of the winding shaft therefore not only depends on the manual speed parameterised, but also on the current diameter. This means that the current diameter, *lrSetDiamOut* output, should correspond to the real diameter if the inching mode is requested, otherwise the winding shaft may be moved at a speed that is not expected.

'Positive inching' means that the material is moved in the direction of the material flow.

'Negative inching' means that the material is moved opposite to the direction of the material flow.

The direction of rotation of the winding shaft is therefore also influenced by the material feeding which is defined via the *xMaterialFeeding* input.

When the *xJogLinePos* = TRUE, the line is driven in positive direction and when the *xJogLineNeg* = TRUE, it is driven in negative direction. The line is driven as long as the input remains set to TRUE. The running travel command cannot be replaced by the other jog command.

The parameterisable setpoints *lrJogLineVel*, *lrJogLineAcc* and *lrJogLineDec* for manual jog refer to the circumferential velocity or line velocity and not to the motor speed.

The parameter values can be changed during operation. They are accepted when resetting the inputs *xJogLinePos* = TRUE or *xJogLineNeg* = TRUE.

Parameters to be set

Identifier	Data type	Description
<i>lrJogLineAcc</i>	LREAL	Acceleration for the inching mode with the 1 / d evaluation. Selection of the value for the maximum acceleration speed variation. • Unit: mm/s ² • Initial value: 100
<i>lrJogLineDec</i>	LREAL	Deceleration for the inching mode with the 1 / d evaluation. Selection of the value for the maximum deceleration speed variation. • Unit: mm/s ² • Initial value: 10
<i>lrJogLineVel</i>	LREAL	Speed for the inching mode with the 1 / d evaluation. Selection of the maximum speed used for traversing. • Unit: mm/s • Initial value: 10

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

9.9 Synchronisation to the line velocity

In the course of commissioning, still without material, a simple speed follower function can be used. This process serves to check whether the axis generally follows the master value with the speed expected, even if the diameter varies. In practice, a manual tachometer is contacted at the circumference of the winder as test equipment, displaying the actual circumferential speed. This can be compared to the *IrSetLineVel* shown. The current diameter *IrSetDiamOut* should correspond to the real diameter if the speed follower function is requested, otherwise the winding shaft may be moved at a speed that is not expected.

The winder axis is synchronised to the line velocity when *xSynclLineVel* is set to TRUE.

The parameters *IrSynclLineAcc* and *IrSynclLineDec* refer to the circumferential velocity or line velocity and not to the motor speed.

Parameters to be set

Identifier	Data type	Description
IrSynclLineAcc	LREAL	Acceleration for synchronising to line velocity • Unit: mm/s ² • Initial value: 100
IrSynclLineDec	LREAL	Deceleration for synchronising to line velocity • Unit: mm/s ² • Initial value: 100

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

9.10 Open-loop tension control based on torque controller

Parameters to be set

Setting the *scPar.IrTensCtrlGain* parameter determines which of the two basic methods is to be used.

Control mode	Parameter- settings	
	eTensCtrlMode	IrTensCtrlGain
Torque-based open-loop	0: "TorqueBased"	= 0
Torque-based closed-loop	0: "TorqueBased"	≥ 0
Speed-based closed loop	1: "SpeedBased"	≥ 0

► L TW2P_scPar_WinderTensionCtrlState (26)

If no actual tensile force value is available for a closed tension control loop, simple open-loop tension control is used. The parameter settings are:

- *eTensCtrlMode* = 0: "TorqueBased"
- *IrTensCtrlGain* = 0 (torque-based open loop)

In open-loop tension control, motor torque is controlled directly depending on the tensile force setpoint and the current diameter. The motor speed is not regulated, it is merely limited via speed limitation in the event of a web break.

Enabling of open-loop/closed-loop tension control

The input *xTensCtrl* = TRUE activates open-loop/closed-loop tension control. The technology module switches to the "TENSIONCTRL" state and the status output *xWindingBusy* = TRUE is set.

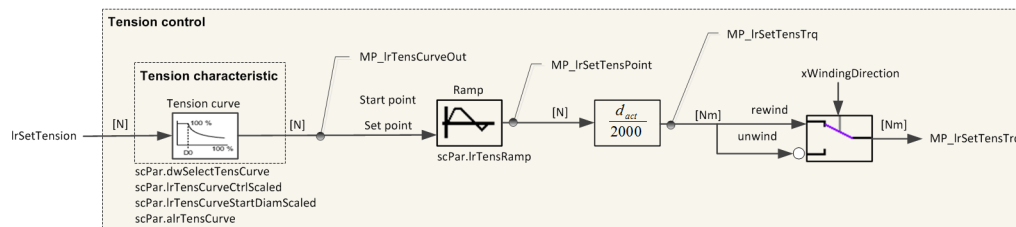
Ramp generator for the tensile force setpoint

The tensile force setpoint is initially guided by a ramp generator to prevent setpoint jumps. The ramp function must be adapted to the required time response in the application via the *scPar.IrTensRamp* parameter. For instance, if the default value of 50 N/s is used, it takes 2 s to generate 100 N of tension on the material. The ramp function is initialised with a starting value of zero as long as open-loop/closed-loop tension control is not active.

The current tensile force setpoint after the ramp function can be diagnosed in the signal flow via the *MP_IrSetTensPoint* measuring point [N].

The tensile force [N] is evaluated using the current diameter and converted into torque [Nm].

If unwinding mode is active, the resulting torque is inverted. The resulting torque setpoint from open-loop tension control is displayed in the signal flow via the *MP_IrSetTensTrq* measuring point [Nm].



Parameters to be set

Identifier	Data type	Description
eTensCtrlMode	L_TW2P_WinderTensionControlMode	Selection of the axis control mode for the reel The mode is effective in winding mode when the tension control/regulation is activated. • Initial value: 0
0		"TorqueBased" The torque control is activated in the drive when the technology module changes to the "TENSIONCTRL" state.
1		"SpeedBased" Speed control is activated in the drive when the technology module changes to the "TENSIONCTRL" state.
IrTensCtrlGain	LREAL	Controller gain • Initial value: 0
IrTensRef	LREAL	Maximum permissible tensile force • Unit: N • Initial value: 50
IrTensRamp	LREAL	Acceleration ramp for the tensile force setpoint • Unit: N/s • Initial value: 50

► [L_TW2P_scPar_WinderTensionCtrlState](#) (📖 26)

9.11 Speed feedforward control

The speed feedforward control is activated for speed-controlled operation of the winder. The speed-controlled operation is activated with the parameter *eTensCtrlMode* = 1 ("SpeedBased").

The setpoint speed for the speed feedforward control is calculated by dividing the line velocity at the *IrSetLineVel* input by the current diameter and the figure π :

Calculation of the setpoint speed for speed feedforward control		
$n_{\text{Set}} = \frac{v_{\text{Line}}}{d_{\text{act}} \times \pi}$		
Symbol	Description	Dimension unit
n_{Set}	Setpoint speed for speed feedforward control	rev/s
v_{Line}	Line velocity at the <i>IrSetLineVel</i> input	mm/s
d_{act}	Current diameter	mm

In order that the winder speed complies with the motor setpoint speed and the line velocity signal, the suitable setting for the motor reference speed is mandatory. For this purpose, the calculation and parameterisation are carried out automatically and not by the user.

The scaled winder setpoint speed at the *IrWndSpdRef* output refers to the motor speed required at minimum diameter d_{min} for reaching the reference line speed at the circumference of the reel.



How to check the speed feedforward control:

1. Load the diameter calculator with the minimum diameter (d_{min}):
 - *IrSetDiam* input = 0 (or $\leq d_{\text{min}}$)
 - Input *xLoadDiam* = TRUE
2. With synchronisation to the line speed if the *xSyncLineVel* input = TRUE, the winder axis follows the line speed setpoint in a purely speed-controlled fashion without correcting the dancer position.
 - ▶ [Synchronisation to the line velocity](#) (73)
3. Start the line velocity master and increase the speed up to e.g. 50%. Now the winder should rotate with a speed corresponding to half the reference speed calculated at the *IrWndSpdRef* output.
4. Now, the circumferential speed of the winder has to correspond to half the *IrLineVelRef* reference. The current line velocity signal is displayed at the output of the technology module *IrSetLineVelScaledOut* = 0.5 [x 100 %] = 50 %.

If the speed or the direction of rotation is not correct, check the definition of the system data described above.

9.12 Open-loop tension control via characteristic function

For many rewinders, the tensile force must be reduced as the diameter increases to prevent the reel from being displaced, depending on the surface and type of winding material. This is referred to as the winding characteristic or tensile force characteristic.

It is common to adjust the tensile force in the winder control unit so that any adapted setpoints can be reproduced on a pneumatic actuator, for example.

The actual tensile force setpoint from the *IrSetTens* input is evaluated using a linear characteristic function on the basis of the diameter to ensure that the material-dependent characteristic is achieved.

The adaptation can be done according to different principles:

- ▶ [Linear tensile force characteristic \(dwSelectTensCurve = 0\)](#) (📖 78)
- ▶ [Linear torque characteristic \(dwSelectTensCurve = 1\)](#) (📖 79)
- ▶ [Freely definable characteristics with 65 interpolation points \(dwSelectTensCurve = 2\)](#) (📖 80)

Parameters to be set

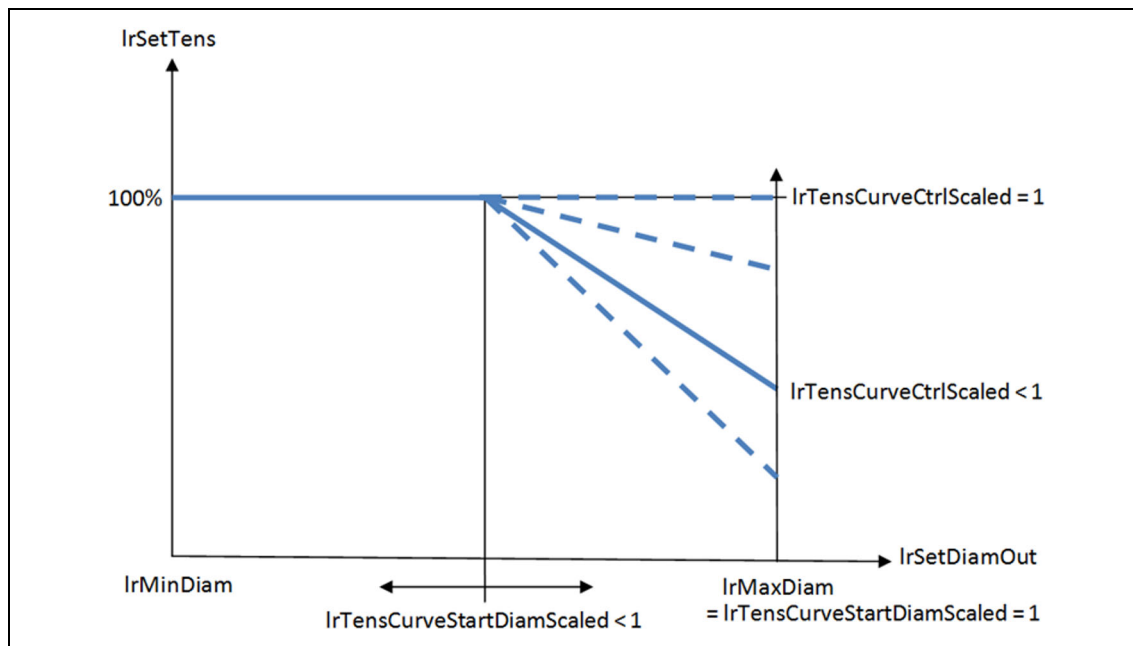
Identifier	Data type	Description
dwSelectTensCurve	DWORD	Selection of the characteristic for tension control Initial value: 0
		0 Linear tensile force profile
		1 Linear torque profile
		2 Tensile force profile on the basis of a specified characteristic
alrTensCurve	ARRAY[1..65] OF LREAL	Characteristic for tension control open loop consisting of 65 values.
IrTensCurveStartDiamScaled	LREAL	Initial point of the characteristic for the tension control open loop - Unit: x 100 % 1 = 100 % = <i>IrMaxDiam</i> parameter - Initial value: 0 (0 %)
IrTensCurveCtrlScaled	LREAL	Gradient of the characteristic for the tension control open loop - Unit: x 100 % (1 = 100 %) - Initial value: 0 The value '1' causes a constant tensile force profile and thus a proportionally increasing setpoint for the diameter.
IrMaxDiam	LREAL	Maximum diameter - Unit: mm - Initial value: 180
IrMinDiam	LREAL	Minimum diameter - Unit: mm - Initial value: 50

- ▶ [L_TW2P_scPar_WinderTensionCtrlState](#) (📖 26)

9.12.1 Linear tensile force characteristic ($dwSelectTensCurve = 0$)

The characteristic is characterised by an initial range with constant evaluation (100 %) and a second range in which the tensile force is adapted to the diameter.

The parameter *IrTensCurveStartDiamScaled* defines the diameter at which the reduction in tensile force should start. With the parameter *IrTensCurveCtrlScaled* the tensile force at the maximum diameter is evaluated.

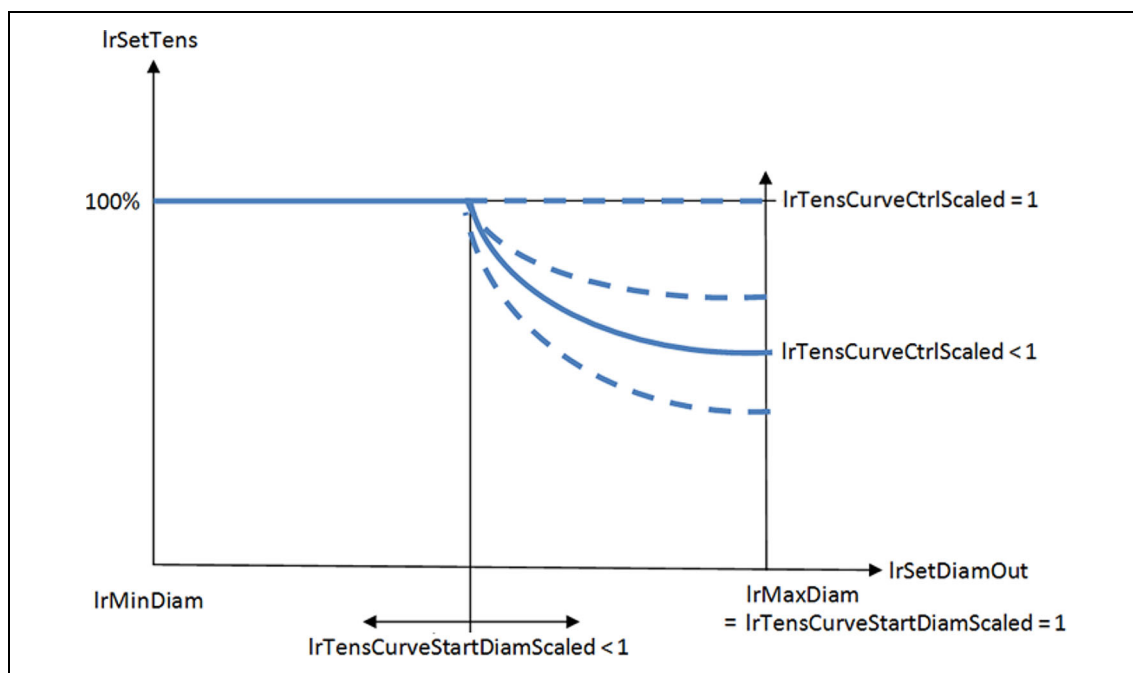


[9-2] Linear tensile force characteristic

9.12.2 Linear torque characteristic (dwSelectTensCurve = 1)

The characteristic is characterised by an initial range with constant evaluation (100 %) and a second range in which the tensile force is adapted linearly over the torque curve. Consequently, the characteristic is adapted via non-linear functions over the diameter.

The parameter *IrTensCurveStartDiamScaled* defines the diameter at which the reduction in tensile force starts. The parameter *IrTensCurveCtrlScaled* defines the percentage of the tensile force at the maximum diameter.

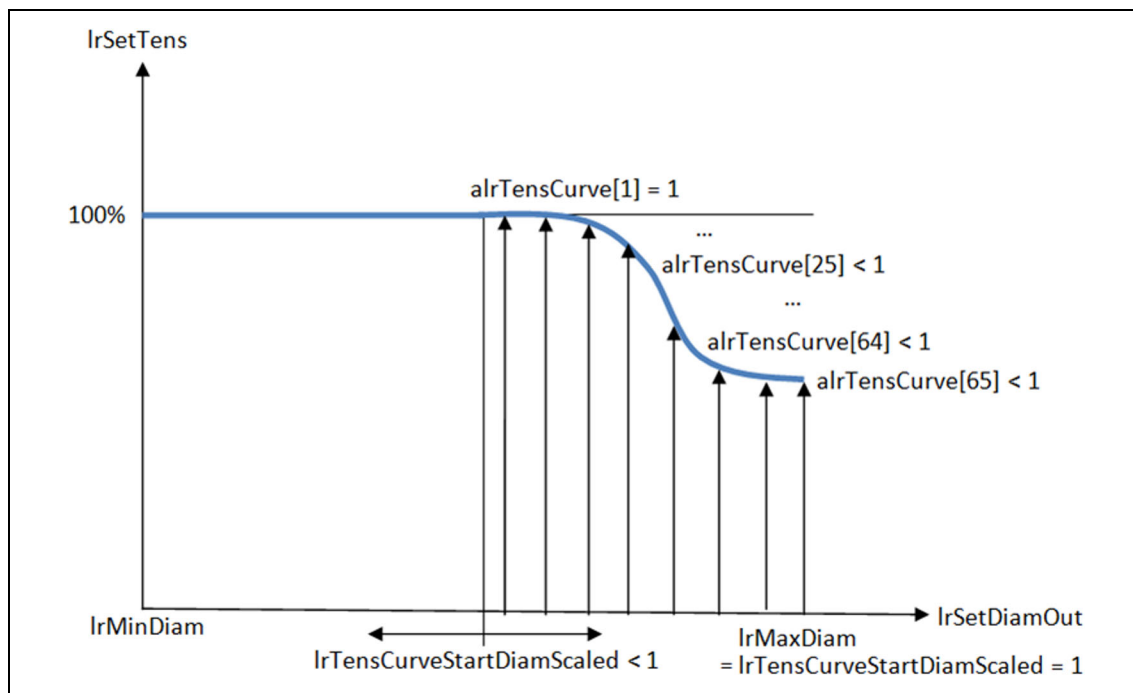


[9-3] Linear torque characteristic

9.12.3 Freely definable characteristics with 65 interpolation points ($dwSelectTensCurve = 2$)

The characteristic can be parameterized via the *alrTensCurve* array. It comprises 65 values, evenly distributed over the diameter range of the parameters *IrMinDiam* and *IrMaxDiam* (minimum and maximum diameter).

All 65 values must be specified for the characteristic.



[9-4] Freely definable interpolation points

9.13 Diameter-dependent tensile force reduction

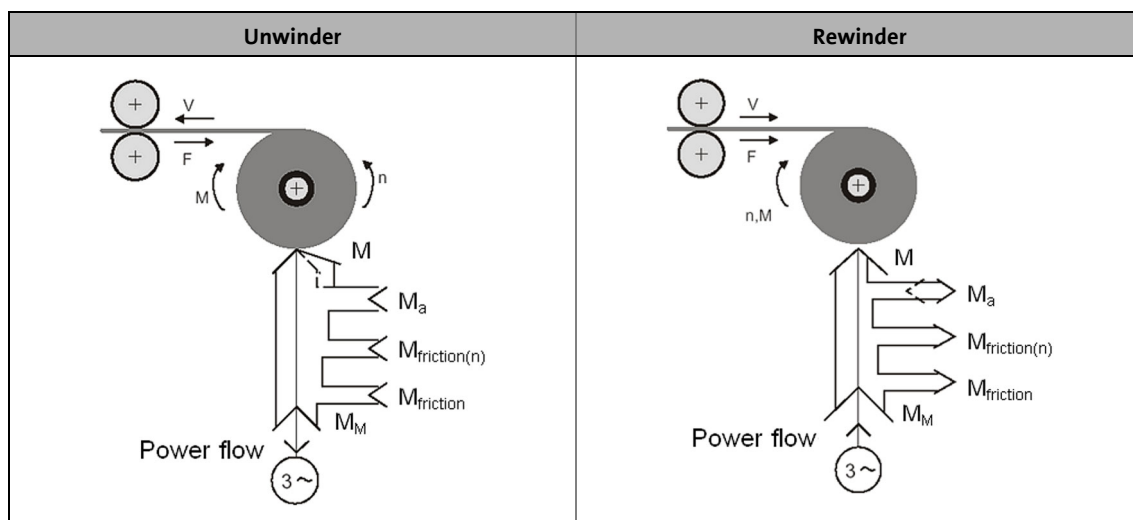
Diameter evaluation / conversion to torque setpoint

The resulting tensile force setpoint from the characteristic evaluation and open loop tension control in a final step must be converted to a motor setpoint torque for the selection via the basic function "Torque follower".

The motor setpoint torque results from the addition of the total tensile force setpoint to the correction values from the friction compensation and the acceleration compensation.

► [Acceleration compensation](#) (192)

► [Friction compensation](#) (101)



Calculation of the motor setpoint torque

$$M_M = M + M_{\text{friction}} + M_{\text{friction}(n)} + M_a$$

with

$$M = F \cdot \frac{d}{2}$$

Symbol	Description	Dimension unit
M_M	Motor setpoint torque	Nm
M	Torque at the reel	
M_{friction}	Static friction torque	
$M_{\text{friction}(n)}$	Speed-dependent friction torque	
M_a	Acceleration torque	
F	Acting force	N
d	Diameter	mm

9.14 Closed-loop tension control

A closed control loop can be added to the concept of simple open-loop tension control. It is used if excessive tensile force deviations are expected or detected despite friction and acceleration compensation. A closed-loop tension controller with PID structure has a corrective function and an additive effect on the torque feedforward control path from open-loop tension control.

For tension control, the drive can be switched to torque-based or speed-based mode. The tension controller with PID structure works corrective and has an additive effect on the torque feedforward control path from the tension control if the *eTensCtrlMode* parameter is set = 0 ("TorqueBased") or set to the speed feedforward control path from the speed control if *eTensCtrlMode* is set = 1 ("SpeedBased").

Closed-loop tension control is activated as soon as a factor greater than 0% is present at the *IrTensCtrlInfluence* input with a P gain of *scPar.IrTensCtrlGain* > 0. The closed-loop tension controller has a direct multiplicative effect on the correcting variable of the controller.

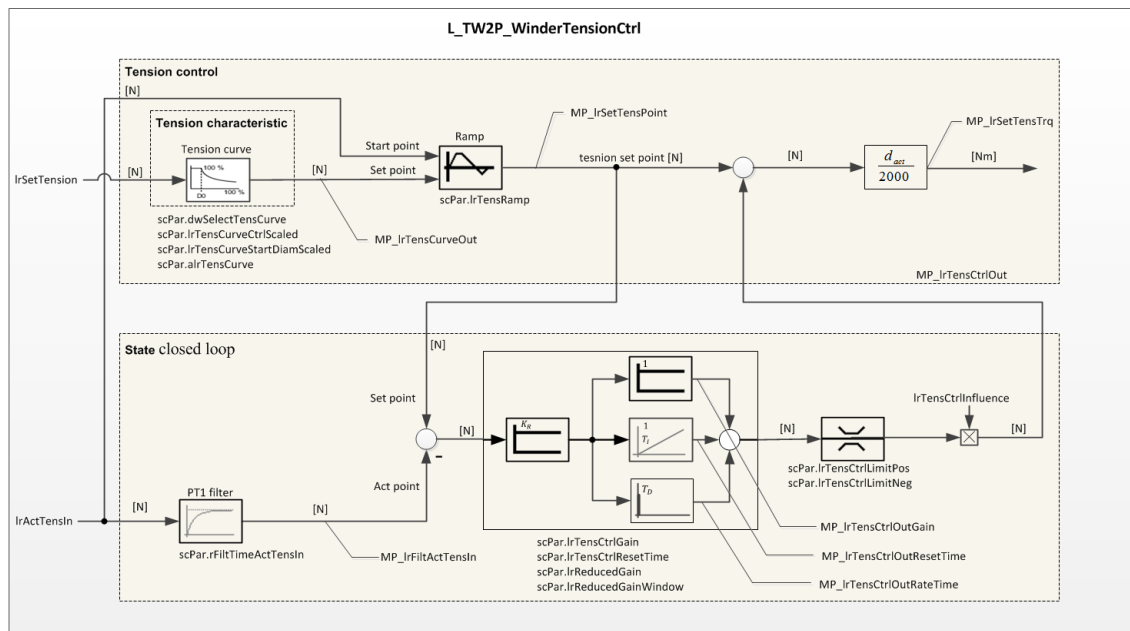
The actual tensile force is expected from the *IrActTensIn* input in the unit [N] to be used in closed-loop tension control. Filtering the actual tensile force value using a PT1 filter can have a positive effect on the stability of the control loop. However, using a PT1 filter can result in dead times in the control loop that must be avoided if high dynamic performance is required. The filter time can be optimised via the *scPar.rFiltTimeActTensIn* parameter.

The correcting variable of the PI controller is also guided via a limiter before multiplying the influencing factor. This allows the *scPar.IrTensCtrlLimitPos* and *scPar.IrTensCtrlLimitNeg* parameters to be used for limitation.

Torque based mode

Settings for the implementation of this control procedure:

- Parameter *eTensCtrlMode* = 0: "TorqueBased"
- Parameter *IrTensCtrlGain* > 0

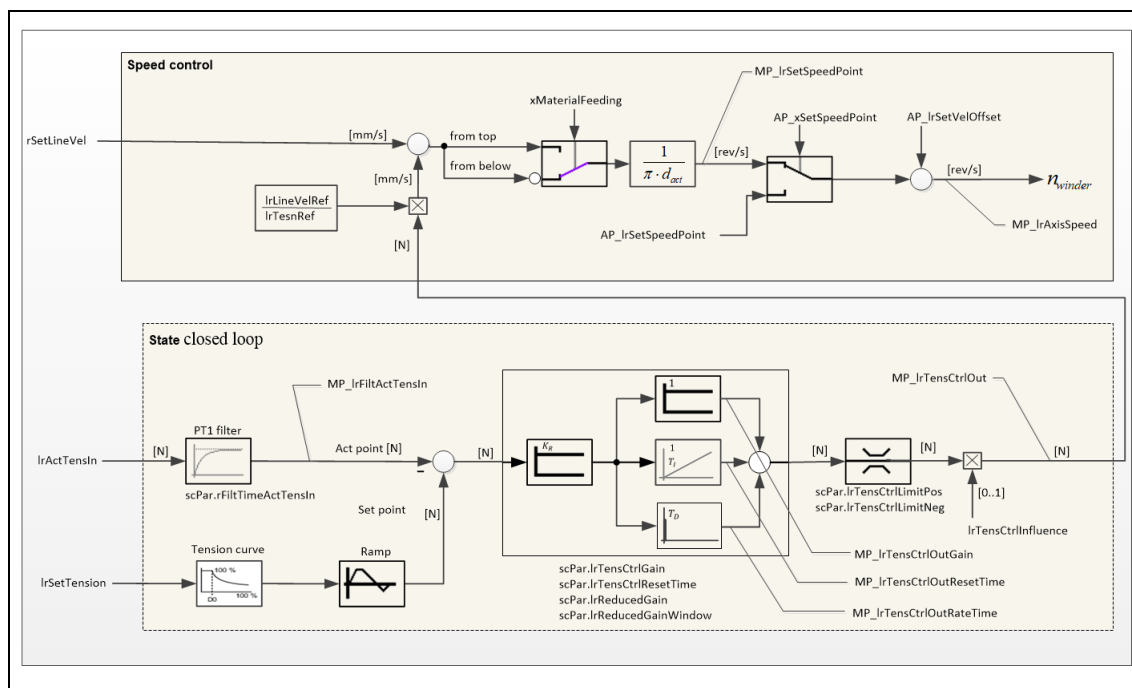


[9-5] Signal flow for tension control in torque based mode

Speed based mode

Settings for the implementation of this control procedure:

- Parameter *eTensCtrlMode* = 1: "SpeedBased"
- Parameter *IrTensCtrlGain* > 0



[9-6] Signal flow for tension control in speed based mode

Parameters to be set

Identifier	Data type	Description
eTensCtrlMode L_TW2P_WinderTensionControlMode		Selection of the axis control mode for the reel The mode is effective in winding mode when the tension control/regulation is activated. • Initial value: 0
	0	"TorqueBased" The torque control is activated in the drive when the technology module changes to the "TENSIONCTRL" state.
	1	"SpeedBased" Speed control is activated in the drive when the technology module changes to the "TENSIONCTRL" state.
IrTensCtrlGain	LREAL	Controller gain • Initial value: 0
IrTensCtrlResetTime	LREAL	Controller reset time • Unit: s • Initial value: 0 (reset time deactivated)
IrTensCtrlLimitPos	LREAL	Limitation of the tensile force controller correcting variable (output of the controller) in positive direction • Unit: [N] • Initial value: 2147483648 (0.5×2^{32})
IrTensCtrlLimitNeg	LREAL	Limitation of the tensile force controller correcting variable (output of the controller) in negative direction • Unit: [N] • Initial value: -2147483648 (-0.5×2^{32})

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

9.15 Reduced controller dynamics at low system deviation

A reduced dynamic performance of the controller with minor system deviations in most cases has a beneficial influence on the control loop damping behaviour.

The system deviation results from the difference between the setpoint value and the actual value, i.e. the difference between the *IrSetTens* input and the *IrActTensIn* output.

The *IrReducedGain* parameter serves to set the value to which the gain is to be reduced in the defined tolerance zone. This means, in the tolerance zone, the reduced gain is active (*IrReducedGain*).

Parameters to be set

Identifier	Data type	Description
IrReducedGainWindow	LREAL	Area of the system deviation with a reduced gain / sensitivity. The <i>IrReducedGainWindow</i> parameter is used to set a tolerance range in which the system deviation is transmitted to the controller with a lower gain. The tolerance range is set above and below the setpoint. • Initial value: 0.0
IrReducedGain	LREAL	System deviation gain within the reduced sensitivity range The <i>IrReducedGain</i> parameter serves to set the value to which the gain is to be reduced in the defined tolerance zone. This means, in the tolerance zone, the reduced gain is active (<i>IrReducedGain</i>). • Initial value: 0.0

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

9.16 Speed limitation

The speed limitation is only relevant for torque-based operation. The torque-based operation is set with the *eTensCtrlMode* parameter = 0 ("TorqueBased").

Speed limitation in closed-loop/open-loop tension control mode

In the closed-loop/open-loop tension-controlled operation mode with mounted material, the winder speed is determined by the circumferential speed and the current diameter. Speed limitation must take effect in the event of a web break or an overcompensation of a disturbance and prevent the reel from being accelerated towards $n_{\max}$.

The *scPar.lrlLineVelOffsetScaled* parameter is used to determine a percentage lead (reference variable *scPar.lrlLineVelRef* = 100%) in relation to the current line speed *lrSetLineVel* that can, however, never be achieved when material is mounted in the system.

It is not generally necessary to change the default limitation threshold value of 10% (*scPar.lrlLineVelOffsetScaled* = 0.10).

The currently effective limitation threshold values can be diagnosed in *MP_lrlSetUpperSpeedLimit* and *MP_lrlSetLowerSpeedLimit*.

If the limitation threshold and consequently speed regulation become active, the status *xWndSpdLimit* = TRUE is set at the output.

Speed limitation in the event of a web break

DANGER!

Injury to persons and damage or destruction of machine parts

Injury to persons and damage or destruction of machine parts is caused by unpredictable motor movements, especially by too high velocities.

Possible consequences

Death or severe injuries

Protective measures

On an unwinder, an overcompensation of friction and/or acceleration should be avoided at all costs!

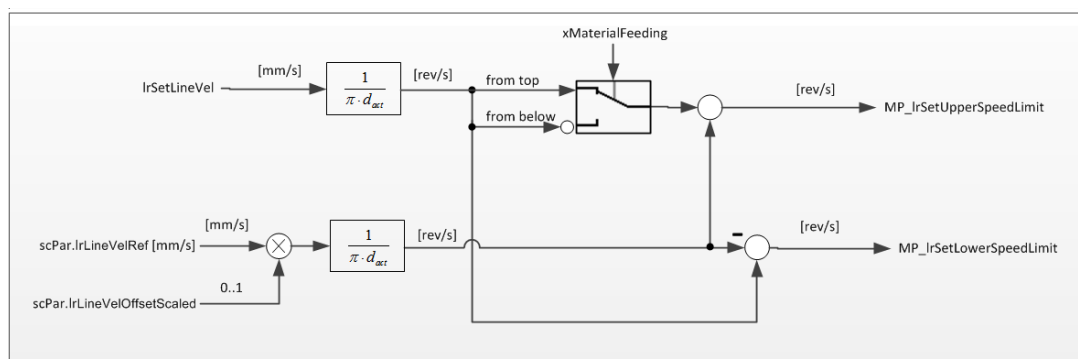
Speed limitation can also not prevent the generation of material slack. This results in the diameter approaching $D_{\min}$ and the winder rotating even faster (positive feedback).

For the speed limitation in torque-controlled operation, the speed offset is converted separately into a rotational speed offset by multiplying by $1/d$. This value is added to / subtracted from the speed setpoint to calculate the upper and lower speed limits.

During normal operation, the calculated diameter values are continuously rising (rewinder) or falling (unwinder). A web break leads to a change of direction in the diameter calculation.

In this process, the diameter of the rewinder may approach $D_{\min}$, which, without speed limitation, would cause the winder to accelerate up to $n_{\max}$.

On an unwinder, a web break would cause the diameter to approach $D_{\max}$. If speed limitation is implemented, the speed in the winding direction will not exceed the configured *scPar.lrlLineVelOffsetScaled*, assessed with the current diameter.



Parameters to be set

Identifier	Data type	Description
IrLineVelRef	LREAL	Maximum line velocity - Unit: mm/s - Initial value: 1000
IrLineVelOffsetScaled	LREAL	Line speed offset for speed limitation - Unit: x 100 % (1 = 100 % = IrLineVelRef parameter) - Initial value: 0.1

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

9.17 Web break monitoring

In the case of a web break, the diameter calculated develops in the opposite direction of the winding direction (unwinding or rewinding), since the actual circumferential speed no longer corresponds to the line speed signal.

The monitoring function is activated with the *xWebBreakMonit* input = TRUE, making a diameter change opposed to the winding direction only permissible within the window set in the *IrWebBreakWindow* parameter.

Rewinding or unwinding is automatically detected via the sign of the line velocity and the winding direction set via the *xWindingDirection* input.

NOTICE

Web break monitoring may only be activated when the calculated diameter corresponds to the real diameter.

If web break monitoring is activated (*xWebBreakMonit* = TRUE), a diameter change contrary to the winding direction specified via the *xUnwind* output.

After loading a starting diameter which deviates considerably from the real diameter contrary to the winding direction, this may lead to an unintentional triggering of the monitoring function. If, for instance, a starting diameter of 50% is loaded for a rewinder and the real diameter only is 45%, then a change of the diameter value to the real 45% is prevented when web break monitoring is activated.

Parameters to be set

Identifier	Data type	Description
IrWebBreakWindow	LREAL	Web break window The current diameter is compared to the previous diameter across the web break window from the last calculation cycle. • Unit: x 100 % (1.0 = 100 %) • Initial value: 0.1 (10 %)

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

9.18 Persistent variables

The technology module provides the possibility of persistently saving the parameters determined such as the winding diameter. For this purpose, the following setting has to be made in the »PLC Designer« for the technology module.

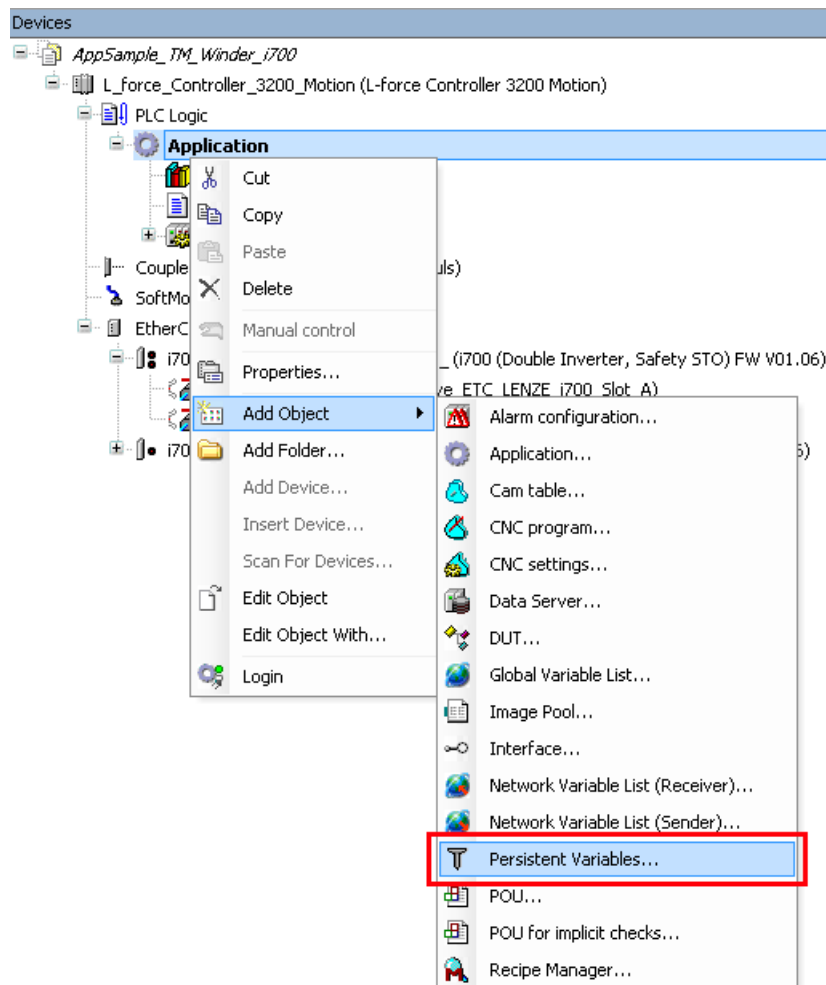


How to create persistent variables in the »PLC Designer«:

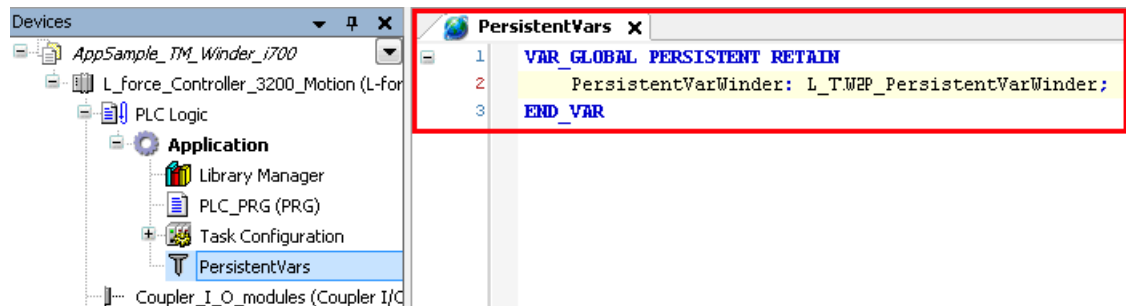
Note!

This procedure does not apply to the Application Template, since it already provides structures for persistent data of the machine modules.

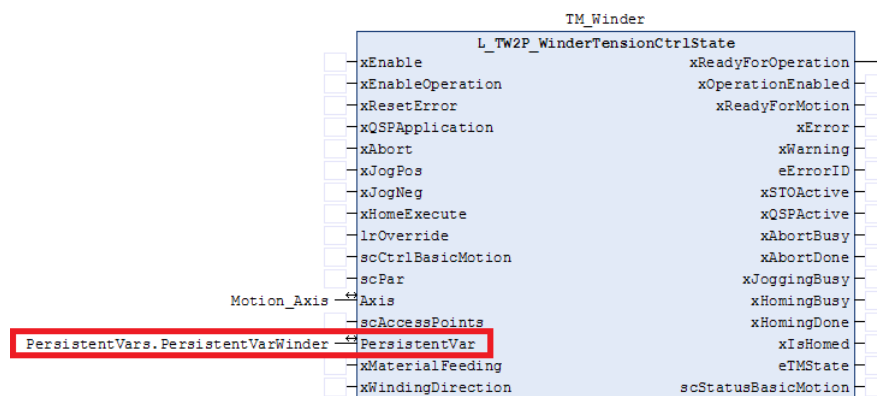
1. Add the global list for managing persistent variables in the context menu to **Application** with the **Add object → Persistent variables...** command.



2. Use the reference for the persistent variables "L_TW2P_PersistentVarWinder" in the global structure of the persistent variables.



3. Interconnect the instance of the persistent variables with the *PersistentVar* input/output.



L_TW2P_PersistentVarWinder

Content of the persistent variables within the **L_TW2P_PersistentVarWinder** structure:

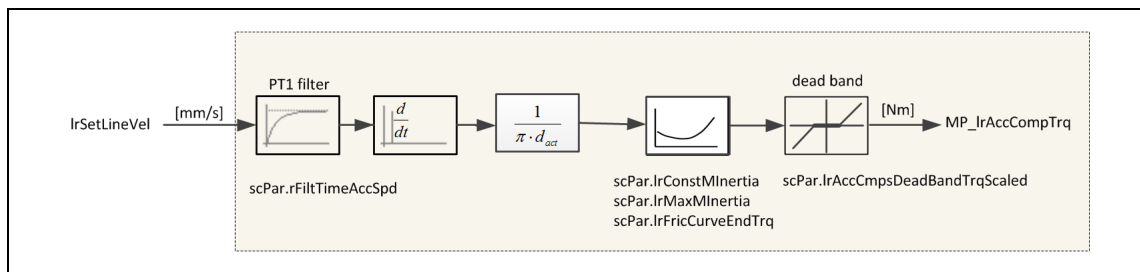
Identifier	Data type	Description
IrDiam	LREAL	The diameter calculated • Unit: mm
IrMaterialCounter	LREAL	Counter content of the material • Unit: mm
liRevCounter	LINT	Number of revolutions counted for the material length counter • Unit: rev
IrRevCounterResidual	LREAL	The fractional part of an incomplete revolution for the material length counter • Unit: rev

9.19 Acceleration compensation

The acceleration in the line velocity setpoint is a disturbance in the winding process. The torque consumed for acceleration is lacking for the tensile force.

Therefore, the acceleration torque has to be calculated and pilot-controlled as an additional torque.

Calculation of the acceleration torque		
$M_a = 2 \cdot \pi \cdot \left(\frac{\partial n}{\partial t} \right) \cdot (J_{const} + J_{var})$ with $J_{var} = (J_{max} - J_{const}) \cdot \left(\frac{d_{act}^4 - d_{min}^4}{d_{max}^4 - d_{min}^4} \right) \cdot B$		
Symbol	Description	Dimension unit
M_a	Acceleration torque	Nm
∂n	(Delta) motor speed	revs/s
∂t	(Delta) time	s
J_{const}	Constant moment of inertia	kgm ²
J_{var}	Variable (diameter-dependent) moment of inertia	
J_{max}	Maximum moment of inertia	
d_{act}	Current diameter	mm
d_{min}	Minimum diameter (sleeve diameter)	
d_{max}	Maximum diameter	
B	Material width	mm



The line speed signal is differentiated for generating an acceleration.

In practice, a line speed signal that is not ideal with a constant characteristic can be expected. Depending on the resolution and stability of this signal, the signal may have to be smoothed via the PT1 filter. The filter time is defined via the *rFiltTimeAccSpd* parameter.

The dead band function is applied for the purpose of noise suppression in the compensation torque calculated. In this way it is avoided that small acceleration changes only resulting from the quantisation or the noise of the line speed signal lead to a compensation torque. The *IrAccCmpsDeadBandTrqScaled* parameter defines up to which value the dead band takes effect.

The acceleration compensation is enabled when the *xAccCmp* input = TRUE.

Different material widths or material densities can be taken into consideration as a percentage via the *IrMInertiaAdapt* input.

The *IrAccCmpsGainAcc* and *IrAccCmpsGainDec* parameters evaluate the compensation torque calculated as a function of the effective direction. In order to avoid slack material, compensation is to be stronger during acceleration processes than during deceleration processes.

Different material widths or material densities can be taken into consideration as a percentage via the *IrMInertiaAdapt* input.

Parameters to be set

Identifier	Data type	Description
rFiltTimeAccSpd	REAL	PT1 filter time for the winder shaft speed for acceleration compensation - Unit: s - Initial value: 0.005
IrAccCmpsDeadBandTrqScaled	LREAL	Lagging range (Dead band) for the current acceleration torque - Unit: x 100 % (1.00 = 100 %); reference variable: rated motor torque [Nm] - Initial value: 0.1 (10 %)
IrAccCmpsGainAcc	LREAL	Gain factor for the acceleration torque in positive direction - Unit: x 100 % (1.00 = 100 %) - Value range: 0 ... 2 (0 ... 200%) - Initial value: 1.05 (105 %)
IrAccCmpsGainDec	LREAL	Gain factor for the acceleration torque in negative direction - Unit: x 100 % (1.00 = 100 %) - Value range: 0 ... 2 (0 ... 200%) - Initial value: 0.95 (95 %)
IrConstMInertia	LREAL	Constant moment of inertia at the winder shaft - Unit: kgcm² - Initial value: 1
IrMaxMInertia	LREAL	Maximally permissible moment of inertia at the winder shaft - Unit: kgcm² - Initial value: 10

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

9.20 Identification of the moments of inertia

A suitable moment of inertia is required if the acceleration compensation or the adaption of the speed controller gain is to be used.

For this purpose, the constant moment of inertia *scPar.IrConstMInertia* and the maximum moment of inertia *scPar.MaxMInertia* must be appropriately configured. During operation, the current moment of inertia is automatically calculated as a function of the current diameter and output at *IrSetMInertiaOut*.

The constant and maximum moment of inertia can be determined via manual calculation or by means of identification.

The maximum moment of inertia corresponds to the reel with the maximum moment of inertia, for which the winder is designed. In order to adapt the reel actually used, which can differ with regard to the material density, the maximum diameter and / or the material width, evaluation factors are provided for the acceleration compensation and speed controller gain in each case.

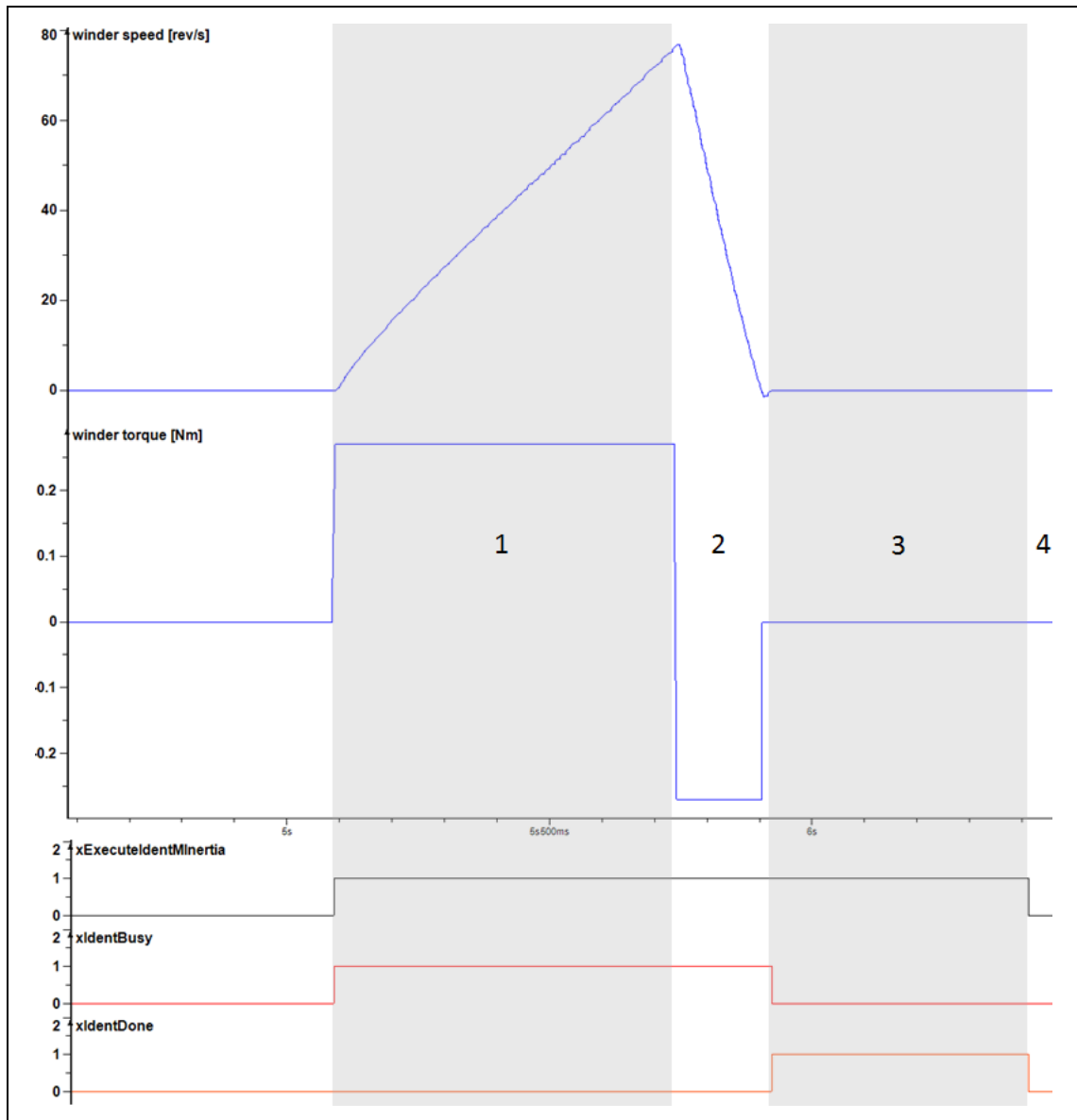
Identification of the constant moment of inertia

The winder shaft is empty (no material).

The *IrIdentMInertiaMaxSpdScaled* parameter serves to define the maximum motor speed in [x 100 %] regarding the maximally reachable winder speed *IrWndSpdRef*. Typically, speeds between 50 ... 60 % are sufficient.

The *IrIdentMInertiaMaxTrqScaled* parameter serves to define the acceleration torque. This value always has to be higher than the maximally occurring friction. Recommendation: 25 %.

The mass moment of inertia is detected with a rising edge (FALSE→TRUE) at the *xExecutIdentMInertia* input. The *IrIdentMInertia* output displays the detected moment of inertia.



No	xExecutIdentMI nertia	xIdentBusy	xIdentDone	eTMState	INFO
1	FALSE ↗ TRUE	FALSE ↗ TRUE	FALSE	STANDBY ↗ IDENTMINER TIA	The axis is controlled in the CST torque mode. With the positive torque, which is defined by the parameter setting for the identification process, the axis is accelerated up to the identified speed defined.
2	TRUE	TRUE	FALSE	IDENTMINER TIA	The speed that is determined via the parameter setting for the identification process has been reached. The torque is commissioned in negative direction. The axis is now decelerated.
3	TRUE	TRUE ↘ FALSE	FALSE ↗ TRUE	IDENTMINER TIA ↘ STANDBY	The axis reaches standstill and the torque is switched off. The histories recorded are evaluated and the mass inertia is determined.
4	TRUE ↘ FALSE	FALSE	TRUE ↘ FALSE	STANDBY	If the assignment for identification is reset, the Done status is also reset.

The control stability of the drive and the actual value resolution of speed and torque are directly included in the calculation result of the identification process. Filtering of the current values often improves the result of the identification process:

- Filter time for the current torque: *rFiltTimIdentMI nertiaTrq*
- Filter time for the current winder speed: *rFiltTimIdentMI nertiaSpd*

Identification of the maximum moment of inertia

The winder is loaded with the maximum possible reel (maximum diameter and maximum width).

The maximum motor speed *IrIdentMInertiaMaxSpdScaled* must be configured so that the maximum permissible circumferential speed of the winder will not be exceeded (e.g. *IrIdentMInertiaMaxSpdScaled* = 10% at $d_{\max}/d_{\min} = 10$) – recommended: 25%.

The mass moment of inertia is detected with a rising edge (FALSE→TRUE) at the *xExecutIdentMInertia* input.

Termination of the identification

Identification is completed as soon as the motor has reached standstill again and no errors have been reported.

The detected inertia of the winder shaft (J_{Winder}) is displayed at the *IrIdentMInertia* output and must be checked for plausibility.

The identification of both types of inertia should be executed several times. In doing so, the filter time for the *rFiltTimeIdentMInertiaSpd* speed can be altered.

NOTICE

A significant non-linear friction in the system adversely affects the calculation of the moment of inertia in the technology module.

Parameters to be set

Identifier	Data type	Description
<i>rFiltTimeIdentMInertiaSpd</i>	REAL	PT1 filter time for the speed at the winder shaft during the identification of the moment of inertia - Unit: s - Initial value: 0.01
<i>rFiltTimeIdentMInertiaTrq</i>	REAL	PT1 filter time for the torque at the winder shaft during the identification of the moment of inertia - Unit: s - Initial value: 0.005
<i>IrIdentMInertiaMaxSpdScaled</i>	LREAL	Maximum speed of the winder shaft during the identification of the moment of inertia - Unit: x 100 % (1.0 = 100 % = <i>IrWndSpdRef</i>) - Value range: 0 ... 1 (0 ... 100 %) - Initial value: 0.2 (20 %)
<i>IrIdentMInertiaMaxTrqScaled</i>	LREAL	Maximum torque of the winder shaft during the identification of the moment of inertia - Unit: x 100 % (1.0 = 100 %) - reference variable: rated motor torque - Value range: 0 ... 1 (0 ... 100 %) - Initial value: 0.2 (20 %)

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

Calculation of the maximum moment of inertia

If no reel is available, the maximum moment of inertia can be calculated as follows:

Calculation of the maximum moment of inertia		
The <u>density</u> of the winding material is known:		
$J_{MaxWinder} = i^2 \cdot J_{Motor} + \left(\frac{\pi}{32 \cdot 10^8} \right) \cdot B \cdot \rho \cdot (d_{max}^4 - d_{min}^4)$		
The <u>mass</u> of the winding material is known:		
$J_{MaxWinder} = i^2 \cdot J_{Motor} + \frac{m \cdot d_{max}^4}{800}$		
Symbol	Description	Dimension unit
$J_{MaxWinder}$	Maximum moment of inertia of the winder shaft	kgcm ²
J_{Motor}	Moment of inertia of the motor shaft	kgcm ²
i	Gearbox factor	
B	Material width	mm
ρ	Material density	kg/dm ³
d_{max}	Maximum diameter	mm
d_{min}	Minimum diameter (sleeve diameter)	mm
m	Mass	kg

The constant moment of inertia is basically determined by the motor. Its moment of inertia is specified in the catalogue.

The values calculated or identified have to be transferred to the respective parameter:

- *scPar.IrConstMinertia*
- *scPar.IrMaxMinertia*

In order to store the values safe against mains failure in the device, the parameter set must be saved!

9.21

Friction identification

NOTICE

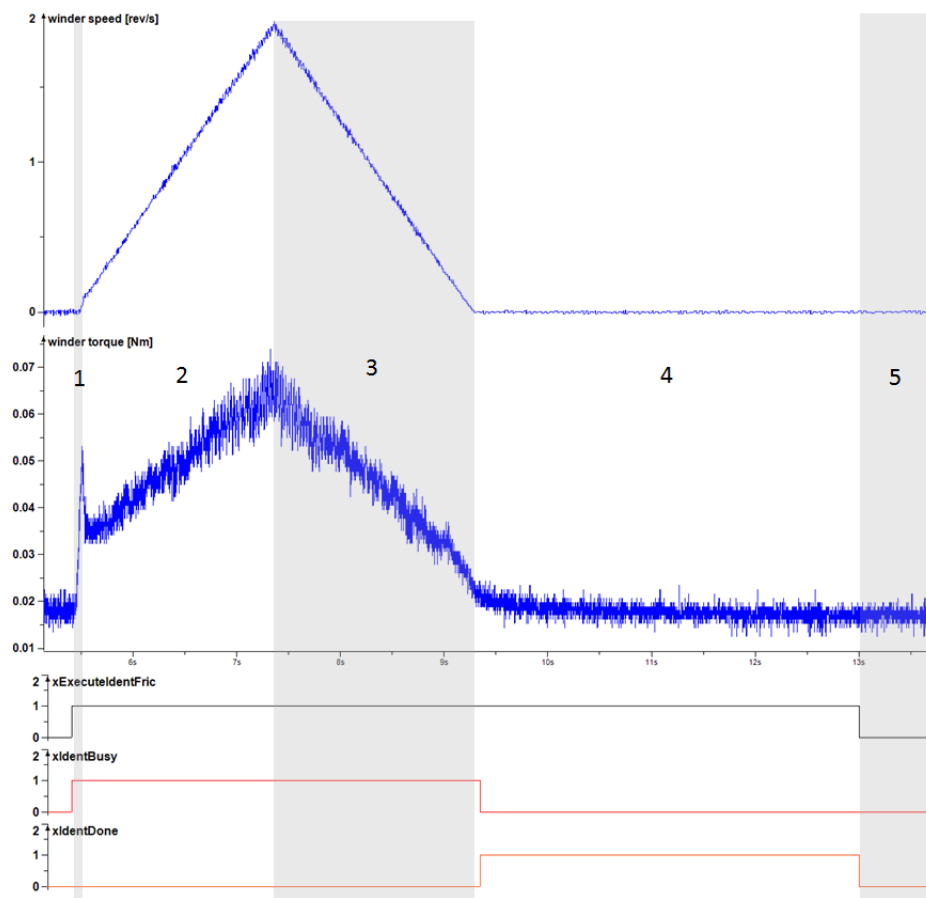
If the winder is controlled via an oil-filled gearbox, the friction very much depends on the temperature of the oil. The friction identification should be carried out using gearbox oil that is as warm as possible. For this it is sufficient to previously actuate the motor in a speed-controlled manner for some minutes.

Friction only depends on speed, not on load and diameter. The winder shaft must be empty (no material mounted) when identifying the friction. This allows the maximum speed to be achieved during the identification run movement.

Use the setting *IrIdentFricMaxSpdScaled* parameter = 1 to define the maximum motor speed (100 %) relating to the maximum winder speed at the *IrWndSpdRef* output, which may be reached during the identification run. (Usually the motor will also reach the *IrWndSpdRef* speed with the minimum diameter and at maximum line speed.)

The *scPar.IrIdentFricAccDec* acceleration for the identification run must be adapted to the current basic conditions. The maximum possible torque must be sufficient to accelerate the mass inertia up to the *scPar.IrIdentFricMaxSpdScaled* parameter over the ramp.

During identification, the drive is accelerated and decelerated using the configured parameters. This movement saves the current torques evenly distributed across an array of 66 elements. The first 1 to 65 elements of the array form a friction characteristic. The reference speed that was reached during identification is set in element 66. The speed-dependent torque must be compensated during subsequent winder operation.



No	xExecutIdentFric	xIdentBusy	xIdentDone	eTMState	INFO
1	FALSE ↗ TRUE	FALSE ↗ TRUE	FALSE	STANDBY IDENTFRIC TION	Identification is activated. The winder is moved into speed mode CSV. Torque is built up until the winder begins to move. The torque required for the winder axis to break loose is referred to as Coulomb friction (static friction) and is saved in the friction characteristic.
2	TRUE	TRUE	FALSE	IDENTFRIC TION	The speed of the winder drive is accelerated over the defined ramp for identifying friction up to the maximum speed set for identifying friction in the parameter setting. The torque identified via the speed is saved as the torque for viscous friction (sliding friction) in the friction characteristic. The maximum speed achieved is the reference speed for the friction characteristic and is saved in the friction characteristic under index 65.
3	TRUE	TRUE	FALSE ↗ TRUE	IDENTFRIC TION	The speed is decelerated until standstill is achieved. The torque is evaluated using the speed and corrected in the friction characteristic.
4	TRUE ↘ FALSE	TRUE ↘ FALSE	TRUE ↘ FALSE	IDENTFRIC TION STANDBY	The winder drive comes to a standstill. This completes friction identification. The Done status is set for identification.
5	FALSE	FALSE	TRUE ↘ FALSE	STANDBY	If the friction identification request is reset, the Done status is also reset.

The control stability of the drive and the actual value resolution of speed and torque are directly included in the characteristic. Filtering the current values often improves the continuous progression of the identified characteristic: *scPar.rFiltTimeIdentFricSpd*, *scPar.rFiltTimeIdentFricTrq*.

Optionally up to 4 friction characteristics can be recorded and stored. A change-over between different friction characteristics is for instance required when variable speed drives are used.

The friction characteristic is detected with a rising edge (FALSE↗TRUE) at the *xExecutIdentFric* input.

Termination of the identification

The identification is completed as soon as the motor has reached standstill again, no errors have been reported and *xIdentDone* has been set to TRUE.

The detected torque setpoints can be checked for plausibility and can be corrected if required.

The friction compensation is activated with *xFricCmps* = TRUE.

Parameters to be set

Identifier	Data type	Description
rFiltTimeIdentFricSpd	REAL	PT1 filter time for the winder shaft speed for friction identification - Unit: s - Initial value: 0.0
rFiltTimeIdentFricTrq	REAL	PT1 filter time for the winder shaft friction torque for friction identification - Unit: s - Initial value: 0.05
IrIdentMIInertiaMaxSpdScaled	LREAL	Maximum speed of the winder shaft during the identification of the moment of inertia - Unit: x 100 % (1.0 = 100 % = <i>IrWndSpdRef</i>) - Value range: 0 ... 1 (0 ... 100 %) - Initial value: 0.2 (20 %)
IrIdentFricAccDec	LREAL	Acceleration for friction identification - Unit: revs/s² - Initial value: 1
dwSelectFricCurve	DWORD	For friction identification (input <i>xFricCmps</i> = TRUE): Selection of the memory area in which the friction values identified are stored. A maximum of 4 characteristics can be stored. - Initial value: 1 1 ... 4: memory area 1 ... 4
		For friction compensation (input <i>xExecuteIdentFric</i> = FALSE → TRUE) Selection of the characteristic for friction compensation 0: Linear friction characteristic (adjustable via <i>IrFricCurveStartTrq</i> parameter) 1 ... 4: identified friction characteristic (memory area 1 ... 4)

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

9.22 Friction compensation

Friction is determined by the winder mechanics and depends on speed. Oil-filled toothed gears have approx. 2% friction in each transmission level. Compensation is necessary, particularly if small torques must be set precisely.

Compensation is achieved by including an additive extra torque *MP_IrSetFricCompTrq*.

In the default setting, simplified friction compensation is activated by a linear characteristic that it is not necessary to identify during commissioning.

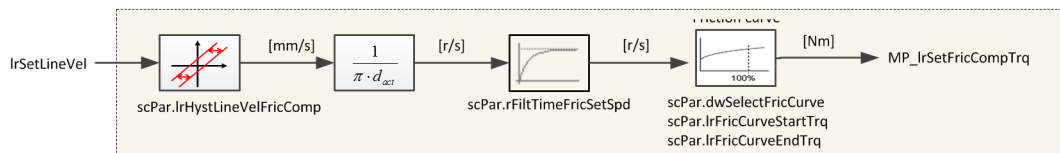
This linear friction characteristic only requires you to specify a start value *scPar.IrFricCurveStartTrq* and an end value *scPar.IrFricCurveEndTrq*. Other values are interpolated in a linear manner between these points.

If non-linear mechanics are used, the characteristic must be identified during commissioning. The *scPar.dwSelectFricCurve* parameter is used to select the identified characteristic (*alrFrictionCurve1*, *alrFrictionCurve2*, *alrFrictionCurve3* or *alrFrictionCurve4*) for compensation.

The reference speed of the characteristic during friction compensation is derived from the line speed signal. Compensation is thus based on the setpoint. Friction compensation stability can be improved by filtering the reference speed. The filter time is set in the *scPar.rFiltTimeFricSetSpd* parameter. However, excessively high filter times can reduce the dynamic response of compensation.

The additional torque exerted by friction can be adapted by multiplication using this factor via the *IrSetFricAdapt* input. This is used, for example, if the gearbox oil temperature is known and must be compensated.

The effective direction of the compensation torque changes depending on whether the system is winding or unwinding. This adaptation is implemented automatically depending on the line speed sign.



The winding direction cannot be automatically detected with the system at a standstill (line speed = 0). In this case, the effective direction of friction compensation can be controlled manually.

If the input *xWindingDirectionStandstill* = TRUE is set, compensation reduces the overall setpoint torque, as for the unwinding process.

If material is to be wound with the line speed at a standstill, set the input *xWindingDirectionStandstill* = FALSE.

Hysteresis

The hysteresis function in friction compensation is only activated if *scPar.IrHystlineStandstill* > 0. If *scPar.IrHystlineStandstill* = 0, friction compensation is not affected.

It is not necessary to use the hysteresis function if a setpoint speed or signal without noise is present at the *IrSetLineVel* input.

If the line speed is detected by an encoder, a noisy signal may occur at the output. This can also effect the torque feedforward control because the torque signal switches between positive and negative although the axis is at a standstill.

This can be prevented by using the hysteresis function. The hysteresis function output is the filtered line speed that is used for friction compensation.

The *scPar.lrHystLineVelFricComp* parameter sets the hysteresis range for line speed and stabilises noisy line speeds. This value must then be set on *scPar.lrHystLineVelFricComp* after the maximum amplitude of input signal noise has been determined.

NOTICE

If hysteresis compensation is active, *scPar.rFiltTimeFricSetSpd* = 0 can be set as the filter may not be needed.

Parameters to be set

Identifier	Data type	Description
dwSelectFricCurve	DWORD	For friction identification (input <i>xFricCmps</i> = TRUE): Selection of the memory area in which the friction values identified are stored. A maximum of 4 characteristics can be stored. - Initial value: 1 1 ... 4: memory area 1 ... 4 For friction compensation (input <i>xExecutIdentFric</i> = FALSE → TRUE) Selection of the characteristic for friction compensation 0: Linear friction characteristic (adjustable via <i>lrFricCurveStartTrq</i> parameter) 1 ... 4: identified friction characteristic (memory area 1 ... 4)
lrFricCurveStartTrq	LREAL	Initial friction for the linear friction compensation if the winder shaft is at a standstill. - Unit: Nm - Initial value: 0
lrFricCurveEndTrq	LREAL	Final friction for the linear friction compensation if the winder shaft is at a standstill. - Unit: Nm - Initial value: 0
rFiltTimeFricSetSpd	REAL	PT1 filter time for the winder shaft speed for friction compensation - Unit: s - Initial value: 0.01
lrHystLineVelFricComp	LREAL	Hysteresis for detecting the direction of line speed. Only used with friction compensation <i>xFricComp</i> = TRUE and a noisy line signal. Maximum amplitude for line speed noise. - Unit: mm/s - Initial value: 0

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

9.23 Adaptation of the speed controller gain

Conditions

The winder axis must be enabled (output *xReadyForMotion* = TRUE).

Operating mode

Looking at the motor and the reel as a rigid one-mass system, the optimum gain of the speed controller is directly proportional to the moment of inertia.

Since the moment of inertia in most cases significantly changes during the winding process, it may be necessary to control the gain of the speed controller along with the moment of inertia in order to provide for a stable control response.

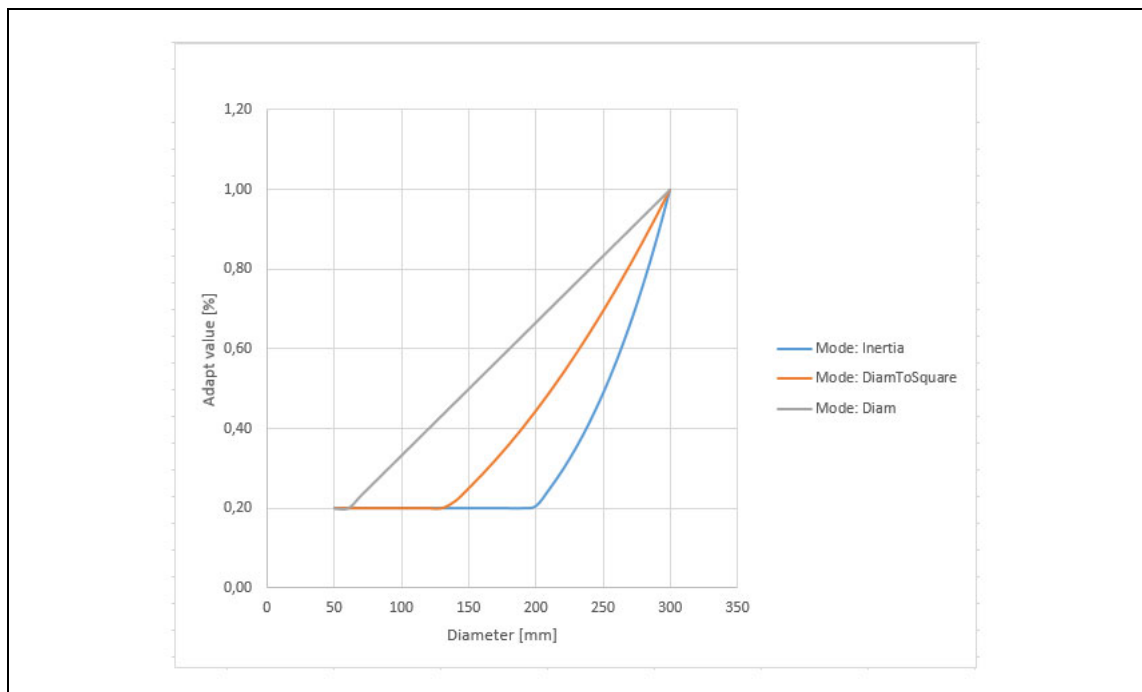
The speed controller does not interfere in open-loop/closed-loop tension control operation. However, it may be useful to work with an adapted gain in manual jog mode and during a quick stop function.

In this function, an adaptation factor is formed that is suitable for the selected method; either in direct relation to the current diameter or the moment of inertia.

This factor is used for multiplicative evaluation of the gain factor set for the speed controller. In this way, the set gain only takes effect once the maximum diameter or maximum moment of inertia has been reached.

The *scPar.eAdaptSpdCtrlGainMode* parameter provides the following methods for generating the adaptation factor:

- *DiamToSquare*: The adaptation factor corresponds to d_{act}^2 / d_{max}^2
- *Diam*: The adaptation factor corresponds to d_{act} / d_{max}
- *Inertia*: The adaptation factor corresponds to J_{act} / J_{max}



[9-7] Adaptation of the speed controller gain with an exemplary lower limitation to *scPar.lAdaptSpdCtrlLowLimit* = 0.2

In the case of very rigid mechanics and winding materials as well as minimised dead times in the control loop, the adaptation method *scPar.eAdaptSpdCtrlGainMode* = 2 (*Inertia*) can be applied.

In this case, the moments of inertia must be entered in a plausible fashion:

- *scPar.IrConstMinertia* and
- *scPar.IrMaxMinertia*

In many applications it suffices to use the default adaptation method *scPar.eAdaptSpdCtrlGainMode* = 0 (*DiamToSquare*). In this case, the moment of inertia has no impact on the adaptation.

A lower adaptation limit is specified for all three methods via the *scPar.IrAdaptSpdCtrlLowLimit* parameter.

Activating/Deactivating adaption of the speed controller gain

The adaption of the speed controller gain is independent of the state of the TM and from a currently executed function and can therefore be activated or deactivated at any time.

- Input *scPar.xAdaptSpdCtrlGain* = TRUE: Adaption of speed controller gain activated.
- Input *scPar.xAdaptSpdCtrlGain* = FALSE: adaption of speed controller gain deactivated.

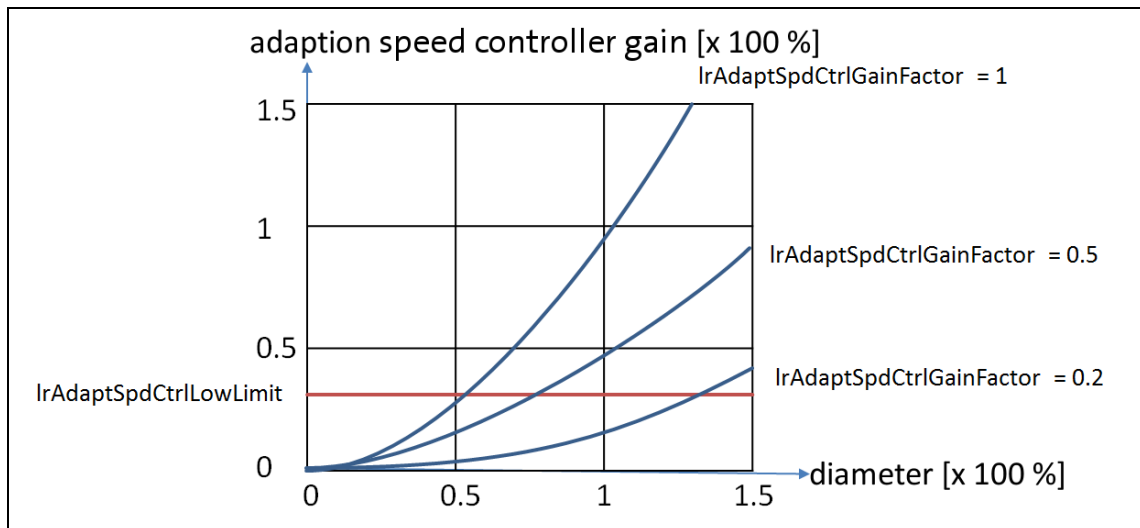
The value for the adaption is calculated in the TM, whereby the calculation regulation is specified via the *scPar.eAdaptSpdCtrlGainMode* adaption mode (see below).

Adaptation value range: 0 ... 1 (1 = 100% of the speed gain from the speed controller setting). A factor from the *scPar.IrAdaptSpdCtrlGainFactor* input can have a multiplicative effect on the resulting adaptation value set in the speed controller.

The smallest permissible value for adaption of the speed controller gain is defined via the *scPar.IrAdaptSpdCtrlLowLimit* parameter.

Adaption mode *eAdaptSpdCtrlGainMode*:= 0 (*DiamToSquare*)

In mode *eAdaptSpdCtrlGainMode*:= 0, the adaption is calculated from the scaled diameter (output *IrSetDiamScaledOut*) for the square.

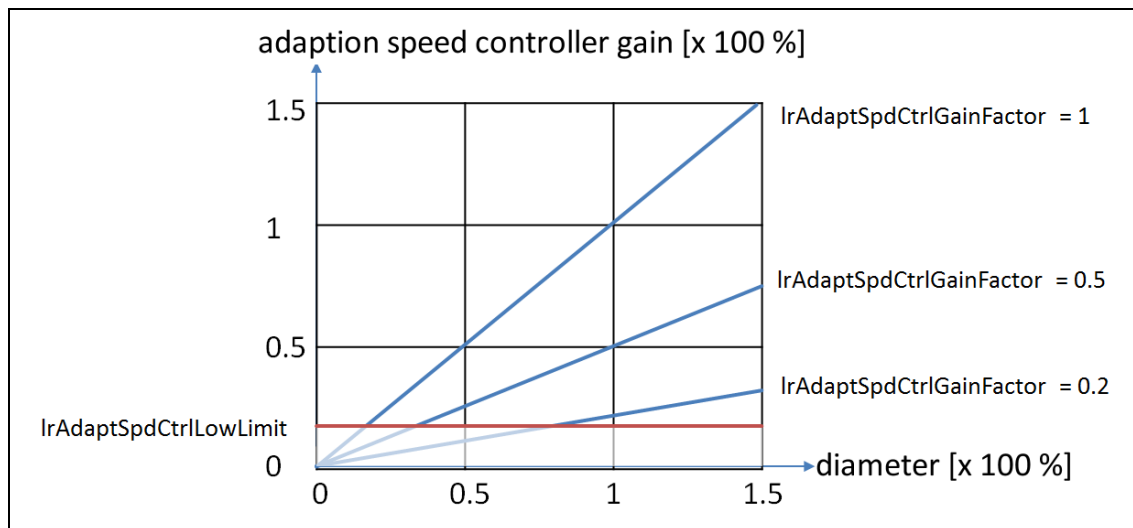


[9-8] Adaption of the speed controller depending on the diameter for the square under the influence of *IrAdaptSpdCtrlGainFactor*

Adaption value = 1 is set for the maximum diameter. The lower limit for the adaption of the speed controller gain is defined via the *scPar.IrAdaptSpdCtrlLowLimit* parameter.

Adaption mode $eAdaptSpdCtrlGainMode := 1$ (Diam)

In mode $eAdaptSpdCtrlGainMode := 1$, the speed controller adaption is calculated proportionally to the scaled diameter (output $IrSetDiamScaledOut$).



[9-9] Adaption of the speed controller depending on the diameter under the influence of $IrAdaptSpdCtrlGainFactor$

Adaption value = 1 is set for the maximum diameter. The lower limit for the adaption of the speed controller is defined via the $IrAdaptSpdCtrlLowLimit$ parameter.

Adaption mode eAdaptSpdCtrlGainMode:= 2 (Inertia)

In an ideal winder drive model, the motor and reels are seen as a fixed single-mass system. The optimum gain of the speed controller therefore behaves in direct proportion to the moment of inertia J with a d4 function.

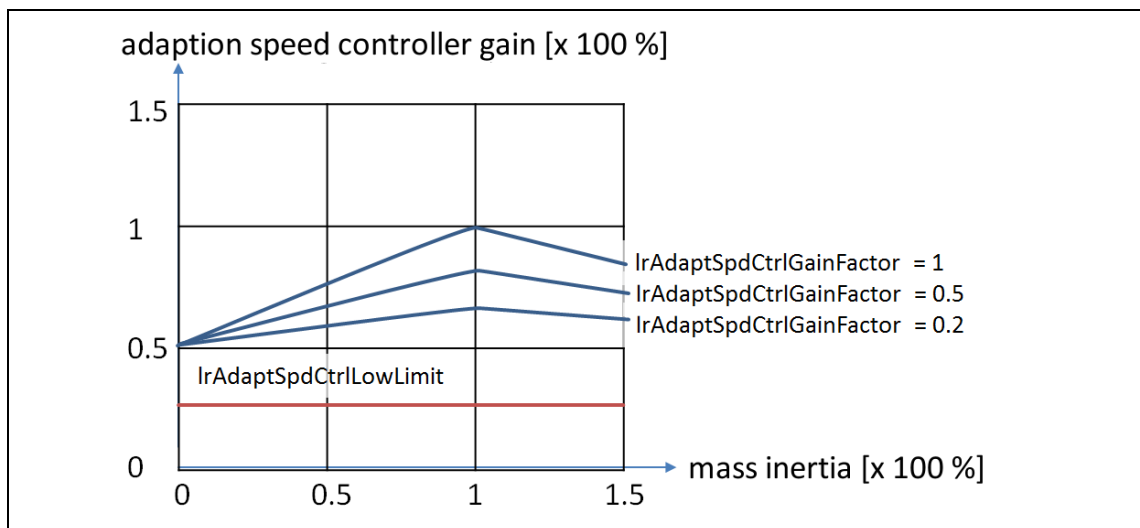
Since the moment of inertia in most cases significantly changes during the winding process, it may be necessary to control the gain of the speed controller along with the moment of inertia in order to provide for an optimum control response.

The moments of inertia must be entered in order to use the mode eAdaptSpdCtrlGainMode:= 2 (Inertia):

- Mass inertia of empty reel *scPar.IrConstMIInertia* with minimum diameter *scPar.IrMinDiam*
- Mass inertia of the reels with material *scPar.IrMaxMIInertia* for maximum diameter *scPar.IrMaxDiam*

The mass inertia can either be calculated or identified from the TM.

When the two *scPar.IrConstMIInertia* and *scPar.IrMaxMIInertia* mass inertias are known, the adaption is defined using the following characteristic:



[9-10] Adaption of the speed controller depending on the diameter under the influence of *IrAdaptSpdCtrlGainFactor*

The standard setting of this characteristic function is based on the following functions:

- Lower limitation of the adaptation: 50 %
- Upper limitation of the adaptation: 100 %
- Linear increase of the gain up to 100 % of the moment of inertia

Parameters to be set

The parameters are located in the *scPar* parameter structure.

Identifier	Data type	Description
xAdaptSpdCtrlGain	BOOL	Switch adaptation of the speed controller gain on/off. • Initial value: FALSE
		TRUE Enable adaptation of the speed controller gain.
		FALSE Adaptation of the speed controller gain inactive.

Identifier	Data type	Description
eAdaptSpdCtrlGainMode	ENUM	Mode selection for adaptation of the speed controller gain. Initial value: 2
	0	DiamToSquare; $VP = f(d^2)$
	1	Diam; $VP = f(d)$
	2	Inertia; $VP = f(J)$
lrAdaptSpdCtrlLowLimit	LREAL	Lower limitation of speed controller gain in the drive. The <i>lrSetSpdCtrlGainAdaptOut</i> adaptation value must not be smaller than the <i>scPar</i> value. <i>lrAdaptSpdCtrlLowLimit</i> . - Value range: 0 to 1 - Initial value: 1.0 (100 %)
alrSpdCtrlGainAdaptX	ARRAY [1...9] OF LREAL	Characteristic function for the speed control gain The X axis corresponds to the adaptation factor currently calculated for the adaptation of the speed controller gain. - Unit: x 100 % (1 = 100 % = <i>lrMaxMinertia</i> parameter) - Initial values: [0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6]
alrSpdCtrlGainAdaptY	ARRAY [1...9] OF LREAL	Characteristic function for the speed control gain The Y axis corresponds to the gain factor of the speed controller. - Unit: x 100 % (1 = 100 %) - Initial values: [0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6]

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

9.24 Limitation of the master line speed

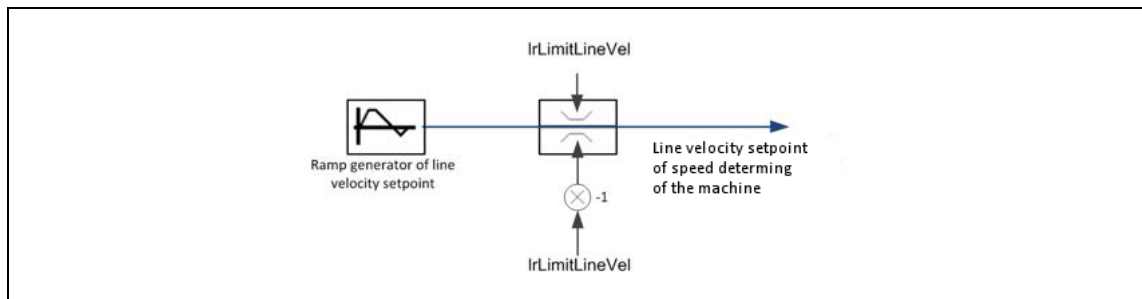
In order to reduce the drive performance for small winder diameters or not to exceed permissible gearbox input speeds, it may be necessary to limit the line speed of the system. The limitation is calculated in the winder TM.

The maximum speed of the winder shaft (drive-side) is defined via parameter *scPar.IrMaxWndSpd*. The limitation function is enabled directly when this parameter is entered.

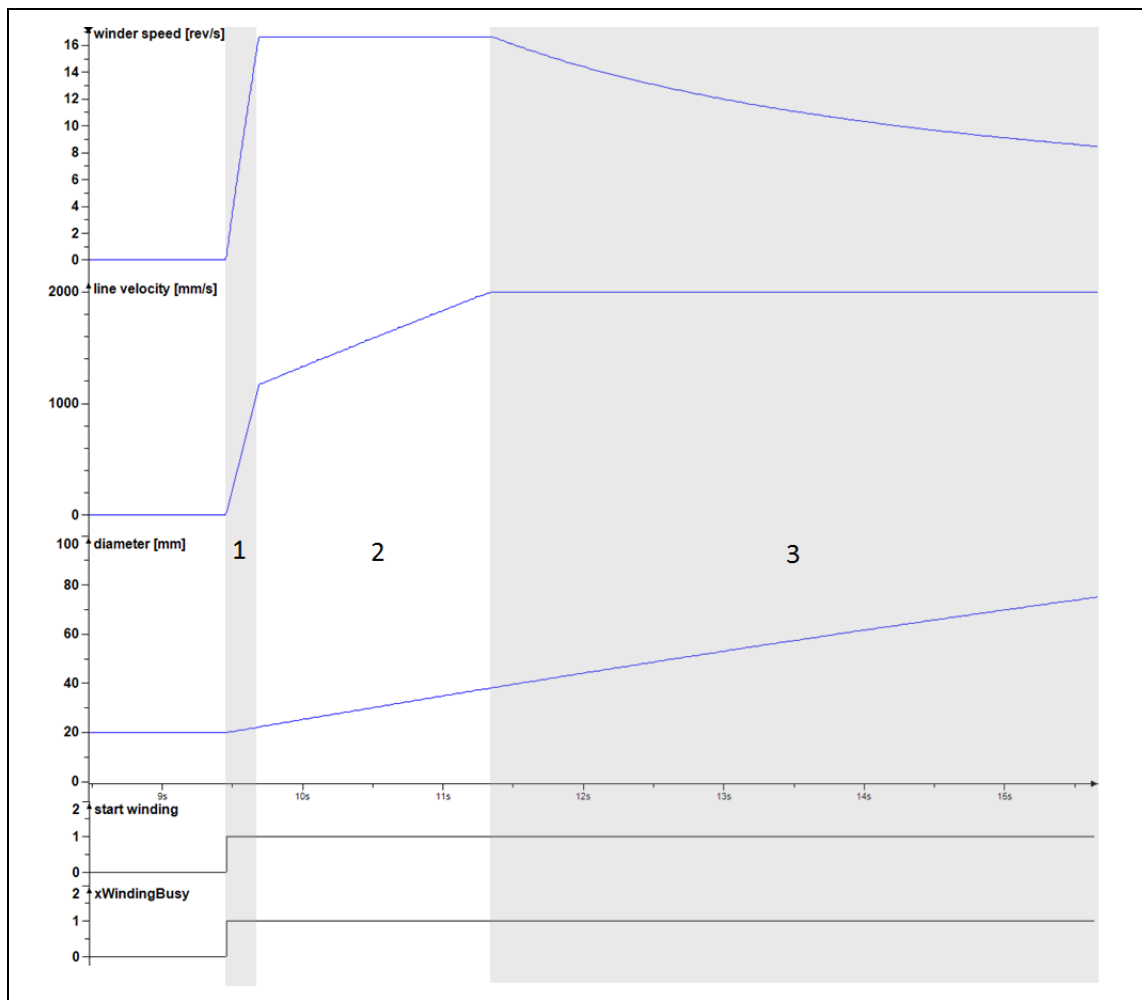
At the *IrLimitLineVel* output, the maximum permissible line speed is output in [mm/s]. Exceeding the *IrLimitLineVel* line speed inevitably means maximum winder shaft speed *scPar.IrMaxWndSpd* will also be exceeded.

Example

The following graphic illustrates the necessary limitation of the line speed between the reference line speed (V_{ref}) to the maximum line speed (V_{max}) for a rewinder which starts at the minimum diameter (D_{min}) so as not to exceed the permissible speed of the *scPar. IrMaxWndSpd* winder shaft.



[9-11] Example for limiting the system setpoint in the drive that determines the velocity (not in the winder) using the *IrLimitLineVel* from the winder TM



[9-12] Example for the limitation of the line speed

No	xWindingBusy	Limitation is active	Line speed has reached the maximum value	INFO
1	TRUE	FALSE	FALSE	The TM is in the winding mode. The line speed is increased.
2	TRUE	TRUE	FALSE	The winding drive reaches the maximum permissible speed. The line limitation from the winder limits the line speed.
3	TRUE	FALSE	TRUE	The line has reached the maximum speed and is traversed constantly. The diameter increases further. The speed of the winder is reduced.

Parameters to be set

Identifier	Data type	Description
lrMaxWndSpd	LREAL	Definition of the maximum winder shaft speed (at the output end). Unit: rev / s Initial value: 100

► [L_TW2P_scPar_WinderTensionCtrlState](#) (📖 26)

9.25 Termination of the winding process

There are two ways of terminating the winding process ("TENSIONCTRL" state):

- A. Set inputs $xTensionCtrl$ = FALSE and $xSyncLineVel$ = TRUE.

The technology module changes from the winding process to the synchronous travel. Here, the winder circumferential speed is synchronised to the line velocity ($lrSetLineVel$ input).

State change: "TENSIONCTRL" → "SYNCLINEVEL"

- B. Set inputs $xTensionCtrl$ = FALSE and $xSyncLineVel$ = FALSE.

The winding process is terminated as a function of the setting of the $eDancerCtrlStopMode$ parameter.

- $eDancerCtrlStopMode$ = 0: Stop
The axis is brought to a standstill from AXIS-REF using the deceleration and jerk from the holding function.
- $eDancerCtrlStopMode$ = 1: Move ABS
With the speed ($lrVel$), acceleration ($lrAcc$), deceleration ($lrDec$) and the jerk ($lrJerk$), the axis is traversed to the target position ($lrPos_Dist$).
The target position is specified at the input of the technology module via the $scCtrlBasicMotion.lrPos_Dist$ structure.
- $eDancerCtrlStopMode$ = 2: Move Rel
With the speed ($lrVel$), acceleration ($lrAcc$), deceleration ($lrDec$), and the jerk ($lrJerk$), the axis is brought to a standstill after the distance travelled ($lrPos_Dist$).
The path is specified via the $scCtrlBasicMotion.lrPos_Dist$ structure at the input of the technology module.

State change: "TENSIONCTRL" → "STOP"

Parameters to be set

► [L_TW2P_scPar_WinderTensionCtrlState](#) (26)

A

Aborting a positioning process [55](#)
Absolute positioning [47](#)
Acceleration compensation [92](#)
Access points [37](#)
Adaptation of the speed controller gain [103](#)
Application notes [8](#)

C

Calculate max. moment of inertia [97](#)
Competing a positioning process [55](#)
Continuous travel with a defined velocity [52](#)
Conventions used [7](#)

D

Defining a start diameter [67](#)
Defining the diameter [67](#)
Defining the winding direction (rewinding/unwinding) [59](#)
Deviation within the reduced sensitivity [85](#)
Diameter calculation [64](#)
Diameter sensor signal [67](#)
Document history [6](#)

E

E-mail to Lenze [113](#)

F

Feedback to Lenze [113](#)
Functional description for "Winder Tension-controlled" [11](#)

H

Halt, [46](#)
Holding the diameter [66](#)
Homing [42](#)

I

Identification of the moments of inertia [94](#)
inputs [20](#)
Inputs and outputs [20](#)

L

L_TW2P_scPar_WinderTensionCtrlState [26](#)
L_TW2P_scPar_WinderTensionCtrlState parameter structure [26](#)
L_TW2P_WinderTensionCtrlState [18](#)
Layout of the safety instructions [8](#)
Limitation of the master line speed [108](#)
Limiting master line speed [108](#)

M

Manual jog (jogging) [40](#)
Master value source for diameter calculation [63](#)
Material feed to the winder [61](#)
Material length counter [69](#)

Material length counting (sources) [69](#)

N

Notes on how to operate the technology module [14](#)

O

Operating mode [14](#)
outputs [24](#)
Override for speed [57](#)

P

Persistent variables [89](#)

Q

QSP application [45](#)

R

Relative positioning [49](#)

S

Safety instructions [9](#)
Signal flow diagrams [33](#)
Signal flow of the "winder dancer-controlled" technology module [34](#)
Signal from the diameter sensor [67](#)
Signalfuss des Technologiemoduls "Winder Tension-controlled" (Torque) [33](#)
Sources for the material length counting [69](#)
Speed controller gain (adaptation) [103](#)
Speed feedforward control [76](#)
Speed override [57](#)
Start of the axes [15](#)
State machine [32](#)
States [32](#)
Structure of the access points
L_TW2P_scAP_WinderTensionCtrl [37](#)
Structure of the signal flow L_TW2P_scSF_WinderTensionCtrl [35](#)
Synchronisation to the line velocity [73](#)
System deviation within the range of reduced sensitivity [85](#)

T

Target group [5](#)
Technology module functions (overview) [13](#)
Tension control via characteristic function (Base version) [77](#)
Terminating the winding process [111](#)
Termination of the winding process [111](#)

V

Variable identifiers [7](#)

W

Web break monitoring [88](#)
Winder Tension-controlled (functional description) [11](#)
Winding direction (automatic detection) [59](#)



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

