SEW EURODRIVE

Manual

MOVIKIT®

MultiMotion, MultiMotion Gearing, MultiMotion Camming

Edition 03/2020

29179459/EN

Table of contents

29179459/EN – 03/2020

29179459/EN – 03/2020

29179459/EN – 03/2020

29179459/EN – 03/2020

1 General information

1.1 About this documentation

This documentation is an integral part of the product. The documentation is intended for all employees who perform work on the product.

Make sure this documentation is accessible and legible. Ensure that persons responsible for the systems and their operation as well as persons who work with the product independently have read through the documentation carefully and understood it. If you are unclear about any of the information in this documentation, or if you require further information, contact SEW-EURODRIVE.

1.2 Content of the documentation

The descriptions in this documentation apply to the software and firmware versions applicable at the time of publication. These descriptions might differ if you install later software or firmware versions. In this case, contact SEW-EURODRIVE.

1.3 Structure of the safety notes

1.3.1 Meaning of signal words

The following table shows the grading and meaning of the signal words for safety notes.

Signal word	Meaning	Consequences if disregarded
⚠ **DANGER**	Imminent hazard	Severe or fatal injuries
⚠ **WARNING**	Possible dangerous situation	Severe or fatal injuries
⚠ **CAUTION**	Possible dangerous situation	Minor injuries
NOTICE	Possible damage to property	Damage to the product or its environment
INFORMATION	Useful information or tip: Simplifies handling of the product.	

1.3.2 Structure of section-related safety notes

Section-related safety notes do not apply to a specific action but to several actions pertaining to one subject. The hazard symbols used either indicate a general hazard or a specific hazard.

This is the formal structure of a safety note for a specific section:

SIGNAL WORD

Type and source of hazard.

Possible consequence(s) if disregarded.

- Measure(s) to prevent the hazard.

29179459/EN – 03/2020

Meaning of the hazard symbols

The hazard symbols in the safety notes have the following meaning:

Hazard symbol	Meaning
	General hazard

1.3.3 Structure of embedded safety notes

Embedded safety notes are directly integrated into the instructions just before the description of the dangerous action.

This is the formal structure of an embedded safety note:

⚠ **SIGNAL WORD!** Type and source of hazard. Possible consequence(s) if disregarded. Measure(s) to prevent the hazard.

1.4 Decimal separator in numerical values

In this document, a period is used to indicate the decimal separator.

Example: 30.5 kg

1.5 Rights to claim under limited warranty

Read the information in this documentation. This is essential for fault-free operation and fulfillment of any rights to claim under limited warranty. Read the documentation before you start working with the product.

1.6 Product names and trademarks

The brands and product names in this documentation are trademarks or registered trademarks of their respective titleholders.

1.6.1 Trademark of Beckhoff Automation GmbH

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

1.7 Copyright notice

29179459/EN – 03/2020

1.8 Other applicable documentation

Observe the corresponding documentation for all further components.

Always use the latest edition of the documentation and the software.

The SEW-EURODRIVE website (www.sew-eurodrive.com) provides a wide selection of documents for download in various languages. If required, you can also order printed and bound copies of the documentation from SEW-EURODRIVE.

1.9 Short designation

The following short designations are used in this documentation:

Type designation	Short designation
MOVIKIT® MultiMotion	Software module
MOVIKIT® MultiMotion Camming	Software module
MOVIKIT® MultiMotion Gearing	Software module

29179459/EN – 03/2020

2 Safety notes

2.1 Preliminary information

The following general safety notes serve the purpose of preventing injury to persons and damage to property. They primarily apply to the use of products described in this documentation. If you use additional components, also observe the relevant warning and safety notes.

2.2 Target group

Software specialist

Any work with the software may only be performed by a specialist with suitable training. A specialist in this context is someone who has the following qualifications:

- Appropriate training
- Knowledge of this documentation and other applicable documentation
- SEW-EURODRIVE recommends additional training for products that are operated using this software.

2.3 Network security and access protection

A bus system makes it possible to adapt electronic drive technology components to the particulars of the machinery within wide limits. There is a risk that a change of parameters that cannot be detected externally may result in unexpected but not uncontrolled system behavior and may have a negative impact on operational safety, system availability, or data security.

Ensure that unauthorized access is prevented, especially with respect to Ethernet-based networked systems and engineering interfaces.

Use IT-specific safety standards to increase access protection to the ports. For a port overview, refer to the respective technical data of the device in use.

2.4 Designated use

MOVIKIT® MultiMotion, MultiMotion Gearing, MultiMotion Camming are software modules for MOVI-C® CONTROLLER that provide the user with a wide range of motion functions via an interface in the IEC program.

Use the device-independent MOVISUITE® engineering software to start up and configure the axes and to download the complete configuration to a MOVI-C® CONTROLLER.

29179459/EN – 03/2020

3 System description

3.1 Module description

- MOVIKIT® MultiMotion

  MOVIKIT® MultiMotion provides universal motion functions for interpolating axes. Various time or master based motion profiles can be activated conveniently by means of a defined IEC interface. Overlay of motion profiles is also possible.

- MOVIKIT® MultiMotion Gearing

  MOVIKIT® MultiMotion Gearing is based on MOVIKIT® MultiMotion and expands the range of functions by position-dependent synchronism between two or more axes.

- MOVIKIT® MultiMotion Camming

  MOVIKIT® MultiMotion Camming includes the range of functions of the MOVIKIT® MultiMotion Gearing software module and expands the range of functions to include the electronic cam as a master-based motion profile.

The range of functions of the software modules can also be expanded using add-ons. For further information, refer to chapter "Add-ons" (→ 🗎 14).

3.1.1 Advantages

The software module offers the following advantages:

- Startup and configuration using the user interfaces in MOVISUITE®
- Control and diagnostics using a monitor tool integrated in MOVISUITE®
- Integration into the application program through automatic code generation
- Simple control of the functions via global variable interfaces

29179459/EN – 03/2020

3.1.2 Areas of application

The software modules are used to coordinate or synchronize complex motion sequences of multiple axes. Applications in the areas of the examples listed below, in particular, can be implemented quickly and conveniently using the motion functions provided:

Packaging technology

- FFS machines
- Cartonizer
- Carton erecting machine
- Smart belts

Transport and logistics

- Multi-axis, lifting, or horizontal drives (e.g. for indoor cranes with multiple drives)
- Load handling devices on storage/retrieval systems with synchronized drives

Processing machines

- Flying saw
- Rotating knife

3.2 Functions

3.2.1 MOVIKIT® MultiMotion

Overview of functions:

- Time-based interpolating operating modes: jog, speed specification, relative/absolute positioning, and referencing
- Master-based interpolating operating mode: Tracking
- Structured variable interface in the programming environment
- Configuration in MOVISUITE® using a graphical user interface
- Diagnostic monitor for monitoring and controlling the axis in the MOVISUITE® engineering software
- Uniform OOP interface for new, state-of-the-art software concepts
- Use of virtual axes, for example for master axes in machines with synchronized axis movements
- Touchprobe function, for example for cut length control
- Processing of additional encoders, for example for detecting slip or misalignment
- Overlapping of profiles of various operating modes, for example for compensating slip or misalignment

29179459/EN – 03/2020

3.2.2 MOVIKIT® MultiMotion Gearing

Overview of functions:

- Scope of functions of the basic module MOVIKIT® MultiMotion
- Configuration via graphical user interface of the MOVISUITE® engineering software
- Additional master-based interpolating operating mode: Synchronism (gearing)
- Position- and time-related synchronizing/desynchronizing with a moving master axis
- Position- and time-related correction of phase and amplitude on the fly.
- Scaling factors based on user units
- Provision of an adjust function

3.2.3 MOVIKIT® MultiMotion Camming

Overview of functions:

- Scope of functions of MOVIKIT® MultiMotion Gearing
- Additional master-based interpolating operating mode: electronic cam (camming)
- Position- and time-related synchronizing/desynchronizing with a moving master axis
- Position- and time-related correction of phase and amplitude on the fly.
- Controllable, on-the-fly curve switching
- Administration of curve descriptions from within the CAM Editor
- Configuration of curve descriptions from within the IEC program
- End-to-end user units, ranging from the configuration to the drive
- Integration of user-specific curve profiles
- Provision of an adjust function

29179459/EN – 03/2020

3.3 Add-ons

INFORMATION

The add-ons are activated in the configuration menu "Basic settings" of the software module in the "Functions used" section. After activation, an additional configuration menu is displayed in the configuration. Please note that a "license" (→ 🗎 17) might be required to use the add-on.

3.3.1 MOVIKIT® MultiMotion add-on PositionController

MOVIKIT® MultiMotion add-on PositionController extends the scope of functions of a MOVIKIT® MultiMotion basic module by central position control and conventional encoder evaluation. When using the MOVIKIT® MultiMotion add-on PositionController in combination with the MOVIKIT® MultiMotion add-on "CombinedEncoderEvaluation", then combined encoder evaluation can be used as the data source.

For further information, refer to the chapters "Configuration menus" (→ 🗎 46) and "IEC programming" (→ 🗎 79).

3.3.2 MOVIKIT® MultiMotion add-on CombinedEncoderEvaluation

The MOVIKIT® MultiMotion add-on CombinedEncoderEvaluation extends the scope of functions of the MOVIKIT® MultiMotion basic module by a combined encoder evaluation. This combined encoder evaluation allows for combining a high-resolution motor encoder and a low-resolution encoder and external encoder subject to dead time. The low-resolution external encoder reliably ensures the reference to the machine even in the case of non-positive connection between motor and machine. The add-on generates a machine-related, dynamically usable and high-resolution encoder signal.

For further information, refer to the chapters "Configuration menus" (→ 🗎 46) and "IEC programming" (→ 🗎 81).

3.3.3 MOVIKIT® MultiMotion Camming add-on AntiSlosh

The MOVIKIT® MultiMotion Camming add-on AntiSlosh extends the scope of functions of MOVIKIT® MultiMotion Camming by a function for generating travel profiles to reduce oscillations in the cyclical movement of liquids. Configuration can be performed via graphical user interface of the MOVISUITE® engineering tool or via the user program. Areas of application for this add-on include filling and closing machines in the food industry, filling machines for solutions or ampoules in the pharmaceutical industry or filling machines for paints, lacquers and the like in the chemical industry.

For more information, refer to the chapters "Configuration menus" (→ 🗎 48), "IEC programming" (→ 🗎 174) and "Application examples" (→ 🗎 243).

29179459/EN – 03/2020

3.3.4 MOVIKIT® MultiMotion Camming add-on Interpolation

The MOVIKIT® MultiMotion Camming add-on Interpolation extends the scope of functions of MOVIKIT® MultiMotion Camming by a function for generating travel profiles based on the interpolation curve point tables that are exported from another tool, for example. Curve point tables can be stored either as ASCII files on the memory card of the MOVI-C® CONTROLLER or can be generated directly from the user program.

Using the add-on is particularly useful if the mathematical function required for a motion profile in a specific application is not included in the selection of MOVIKIT® MultiMotion Camming, or if the motion profile cannot be defined using a mathematical function or the function is not known.

With linear or cubic interpolation, up to 1024 curve points can be interpolated, and with spline interpolation, up to 64 curve points can be interpolated.

For further information, refer to the chapters "IEC programming" (→ 🗎 155) and "Application examples" (→ 🗎 238).

3.3.5 MOVIKIT® Motion add-on AntiSway

INFORMATION

The function is not modulo-capable. Do not use this function to operate modulo or endless axes.

The MOVIKIT® Motion add-on AntiSway extends the range of functions by a function for generating travel profiles to suppress oscillations. For this purpose, the add-on provides various application types for configuration in MOVISUITE®.

For more information, refer to the chapters "Configuration menus" (→ 🗎 50), "IEC programming" (→ 🗎 89) and "Application examples" (→ 🗎 251).

29179459/EN – 03/2020

4 Project planning information

4.1 Requirement

Correct project planning and proper installation of the devices are required for successful startup and operation.

For detailed project planning information, refer to the documentation of the respective devices.

4.2 Hardware

The following hardware is required:

- MOVI-C® CONTROLLER (all power classes)
- MOVIDRIVE® system **or** MOVIDRIVE® technology (as interpolating device)

4.3 Software

The following software is required:

- MOVISUITE® engineering software

  (includes MOVIRUN® flexible and the IEC Editor)

For more detailed information on the hardware requirements of the individual software components, see the documentation for the respective software.

29179459/EN – 03/2020

4.4 Licensing

The following licenses are available and are required:

- MOVIRUN® flexible

  License for the MOVIRUN® flexible software platform that also includes the license for the basic module MOVIKIT® MultiMotion.
- MOVIKIT® MultiMotion Gearing

  License for MOVIKIT® MultiMotion Gearing
- MOVIKIT® MultiMotion Camming

  License for MOVIKIT® MultiMotion Camming
- MOVIKIT® MultiMotion add-on CombinedEncoderEvaluation

  License for MOVIKIT® MultiMotion add-on CombinedEncoderEvaluation
- MOVIKIT® MultiMotion add-on PositionController

  License for MOVIKIT® MultiMotion add-on PositionController
- MOVIKIT® MultiMotion Camming add-on AntiSlosh

  License for MOVIKIT® MultiMotion Camming add-on AntiSlosh
- MOVIKIT® MultiMotion Camming add-on Interpolation

  License for MOVIKIT® MultiMotion Camming add-on Interpolation
- MOVIKIT® Motion add-on AntiSway

  License for MOVIKIT® Motion add-on AntiSway

For detailed information on the scope of functions of the individual licenses, refer to the chapters "Functions" (→ 🗎 12) and "Add-ons" (→ 🗎 14).

The license/licenses are referred to as performance licenses. They have to be purchased only once per MOVI-C® CONTROLLER and can then be used for any number of axes.

For further information on licensing, refer to the document "MOVI-C® Software Components". You can download the document from the SEW-EURODRIVE website (www.sew-eurodrive.com).

29179459/EN – 03/2020

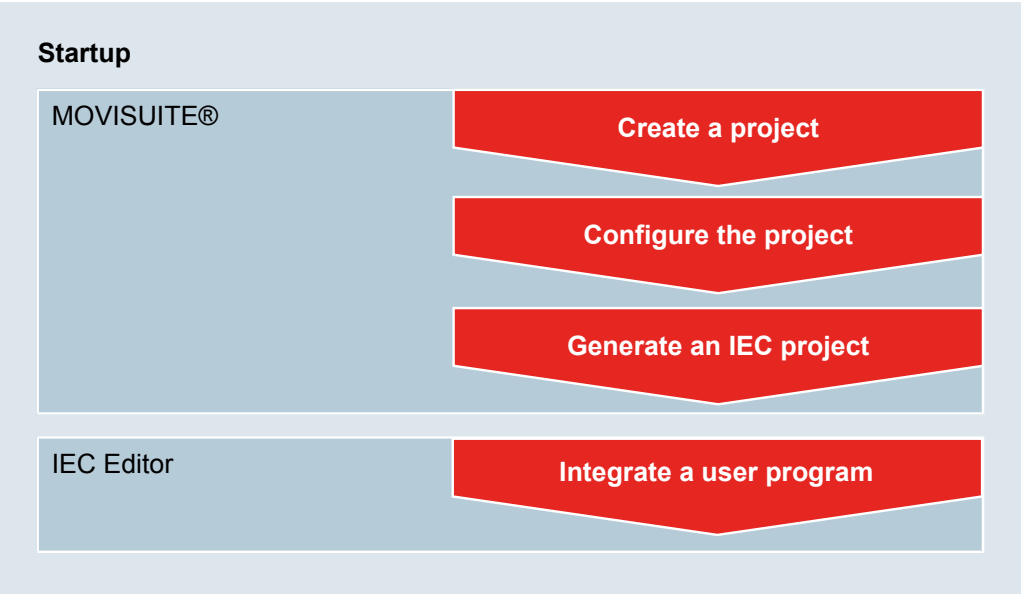
5 Startup

5.1 Requirements

- Check the installation of the inverters and, if installed, also check the encoder connection.
- Observe the installation notes in the documentation of the respective device and software components.
- The devices to be started up are displayed in MOVISUITE®.

5.2 Startup procedure

The schematic diagram below shows the startup procedure:



18014421798899467

The startup steps specific to these software modules are explained in detail in the following chapters of this manual. For startup, also observe the documentation of all the other components in use.

29179459/EN – 03/2020

5.3 Configuring a project

INFORMATION

For detailed information on how to operate the MOVISUITE® engineering software, refer to the corresponding documentation.

✓ A MOVISUITE® project has been created and is open.

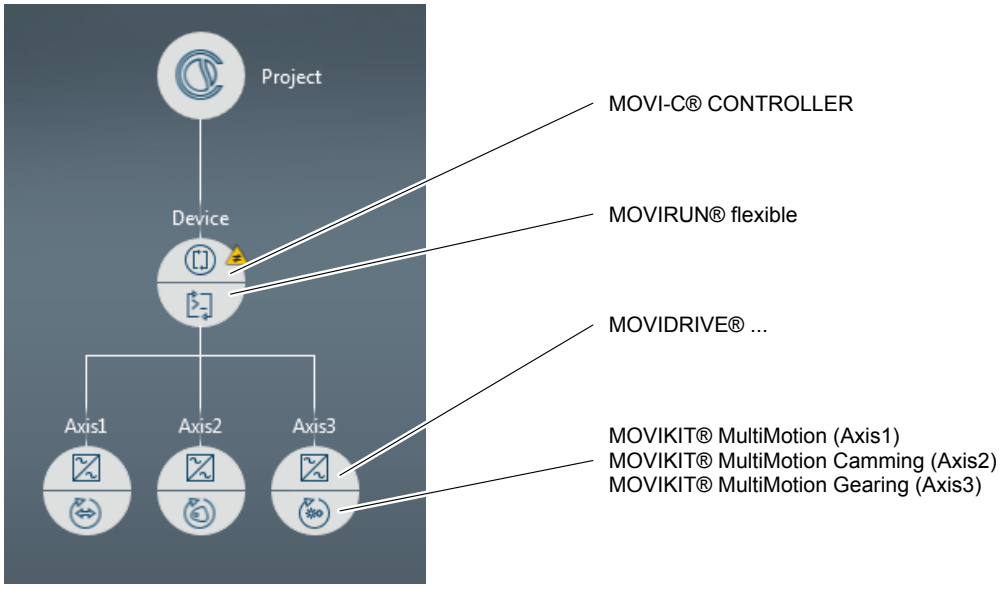
1. Add required device nodes, software nodes (MOVI-C® SoftwareNode) and software modules to the project.

   ⇨ See "Example project".

2. Configure the added devices or software modules. If available, observe the specific notes in the following chapters that apply to MOVIKIT® MultiMotion, MultiMotion Gearing, MultiMotion Camming. For detailed information on the configuration of devices or other software modules, refer to the respective documentation.

5.3.1 Example project

The following figure shows an example project:



31220465291

5.3.2 Adding MOVIKIT® MultiMotion, MultiMotion Gearing, MultiMotion Camming

INFORMATION

For detailed information on how to operate the MOVISUITE® engineering software, refer to the corresponding documentation.

✓ A MOVISUITE® project has been created and is open.

1. Click on the empty software module section of the required node.
   ⇨ The catalog section opens and displays the available software modules.
2. In the catalog section, click on MOVIKIT® MultiMotion, MultiMotion Gearing, MultiMotion Camming.
   ⇨ A context menu opens.
3. Select the version from the respective drop-down list in the context menu and confirm your selection with [Apply].

⇨ MOVIKIT® MultiMotion, MultiMotion Gearing, MultiMotion Camming is assigned to the node, the configuration is created, and the basic settings are performed.

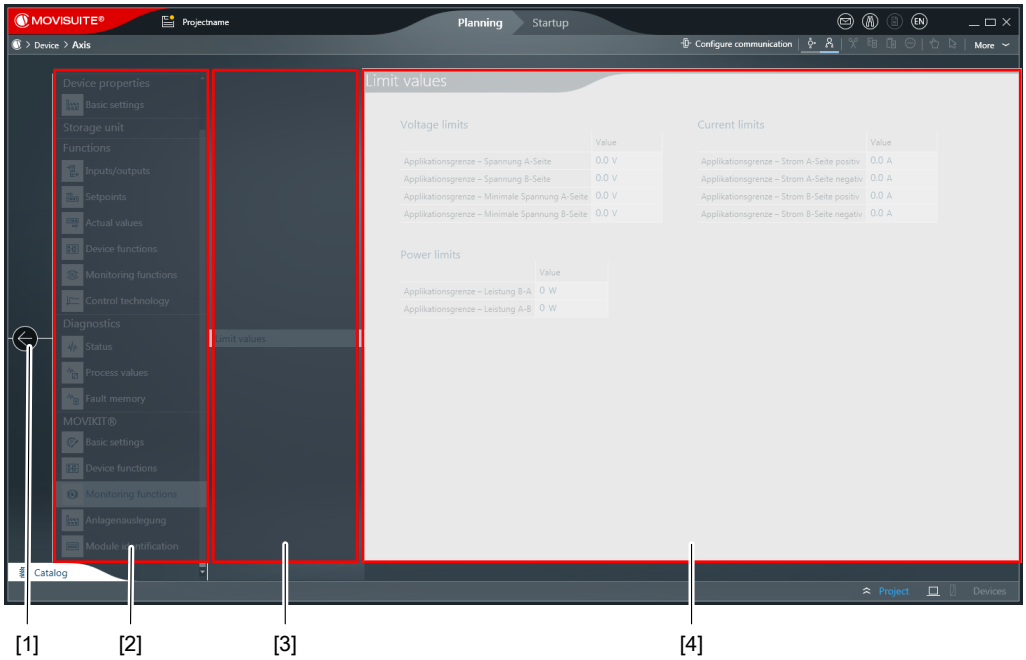
29179459/EN – 03/2020

5.3.3 Configuring MOVIKIT® MultiMotion, MultiMotion Gearing, MultiMotion Camming

INFORMATION

For detailed information on how to operate the MOVISUITE® engineering software, refer to the corresponding documentation.

1. In MOVISUITE®, click on MOVIKIT® MultiMotion, MultiMotion Gearing, MultiMotion Camming.
   ⇨ The configuration menus of the software module are displayed. The configuration menus are explained in the following subchapters.



9007228165413771

[1] Button to return to the project overview
[2] Main menu of the software module configuration (MOVIKIT® section)
[3] Submenus of the configuration
[4] Setting fields of the respective submenu

2. Configure the software module using the respective setting fields.
3. Click button [1] after having completed the configuration.
   ⇨ The project overview is displayed.

INFORMATION

For the changes made to the configuration to take effect, you have to update the configuration data. To do so, click [Update configuration data] in the respective notification at the node or in the context menu of the MOVI-C® CONTROLLER. The MOVI-C® CONTROLLER is stopped and restarted for updating the configuration data.

Basic settings

Parameter name	Value
General	
Activate simulation	• Yes – Simulate MultiMotion axis functions • No – Do not simulate MultiMotion axis functions
	Index: 50000.2
	IEC name: Inverter.In.xSimulation
Functions used	
Position controller **Information:** "Configuration menu" (→ 🗎 46) becomes visible when activated.	Extend the software module by the ability to perform central positioning control and encoder evaluation. • On • **Off**
	Index: 50000.102
	IEC name: _fbController._stConfig.stOptionalModes.xAddonPositionController
Anti-sway control **Information:** "Configuration menu" (→ 🗎 50) becomes visible when activated.	Activation or deactivation of the option to configure vibration suppression measures. • On • **Off**
	Index: 50010.108
	IEC name: _fbController._stConfig.stOptionalModes.xAddonAntiSway
Combined encoder evaluation **Information:** "Configuration menu" (→ 🗎 46) becomes visible when activated.	Extend the software module by the ability to perform combined encoder evaluation. • On • **Off**
	Index: 50010.107
	IEC name: _fbController._stConfig.stOptionalModes.xAddonAdvancedEncoderEvaluation
Interpolation **Information:** "Configuration menu" (→ 🗎 47) becomes visible when activated.	Extend the software module by the ability to generate travel profiles based on the interpolation of curve point tables. • On • **Off**
	Index: 50006.50
	IEC name: _fbController._stConfig.stOptionalModes.xAddonInterpolation

29179459/EN – 03/2020

Parameter name	Value
AntiSlosh **Information:** "Configuration menu" (→ 🗎 48) becomes visible when activated.	Extend the software module by the ability to generate optimized travel profiles with oscillation reduction for cyclical movements of liquids. • On • **Off**
	Index: 50006.51
	IEC name: _fbController._stConfig.stOptionalModes.xAddonAntiSlosh
Touchprobe 1 **Information:** "Configuration menu" (→ 🗎 43) becomes visible when activated.	Configure touchprobe 1 of the inverter in such a way that it can be controlled via the software module and the detected value can be read. • On • **Off**
	Index: 50000.20
	IEC name: -

Monitoring functions

Software limit switches

Parameter name	Value
Software limit switches	
Monitoring negative SW limit switch	• On Activate monitoring for negative software limit switch(es) • **Off** Deactivate monitoring for negative software limit switch(es)
	Index: 8572.3
	IEC name: LimitSwitchEvaluation.SoftwareLimitSwitch.In.xActivateMonitoringNegative
SW limit switch negative	Position of the negative software limit switch (in user units)
	Index: 8572.4
	IEC name: LimitSwitchEvaluation.SoftwareLimitSwitch.In.lrLimitNegative

29179459/EN – 03/2020

Parameter name	Value
Monitoring positive SW limit switch	• On Activate monitoring for positive software limit switch(es) • **Off** Deactivate monitoring for positive software limit switch(es)
	Index: 8572.5
	IEC name: LimitSwitchEvaluation.SoftwareLimitSwitch.In.xActivateMonitoringPositive
SW limit switch positive	Position of the positive software limit switch (in user units)
	Index: 8572.6
	IEC name: LimitSwitchEvaluation.SoftwareLimitSwitch.In.lrLimitPositive

Limit values

Parameter name	Value
Application limits	
Positive speed	Limits the maximum positive speed permitted for moving the system. (in user units)
	Index: 8357.10
	IEC name: ConfigHandling._stAxisConfig.lrAppLimitVelocityPositive
Negative speed	Limits the maximum negative speed permitted for moving the system. (in user units)
	Index: 8357.11
	IEC name: ConfigHandling._stAxisConfig.lrAppLimitVelocityNegative
Acceleration	Limits the maximum acceleration permitted for accelerating the system. (in user units)
	Index: 8357.12
	IEC name: ConfigHandling._stAxisConfig.lrAppLimitAcceleration
Deceleration	Limits the maximum deceleration permitted for braking the system. (in user units)
	Index: 8357.13
	IEC name: ConfigHandling._stAxisConfig.lrAppLimitDeceleration

29179459/EN – 03/2020

Parameter name	Value
Jerk time	Limits the jerk time in [ms] The jerk time takes effect in torque control (FCB 07), speed control (FCB 05), and positioning control (FCB 09), as well as in manual mode. The positioning process extends to twice the set jerk time.
	Index: 8357.14
	IEC name: ConfigHandling._stAxisConfig.lrAppLimitJerkTime
Torque	Limits the maximum torque that may be applied to the system in [Nm]
	Index: 8357.15
	IEC name: -
Limits	
Emergency stop deceleration	Deceleration for the ramp that is activated during an emergency stop. An emergency stop can be programmed as a response to a fault (in user units).
	Index: 8357.20
	IEC name: ConfigHandling._stAxisConfig.lrRapidStopDeceleration
Cycle limit	
Modulo minimum	Lower modulo limit (in user units)
	Index: 8357.30
	IEC name: ConfigHandling._stAxisConfig.lrModuloMin
Modulo maximum	Upper modulo limit (in user units)
	Index: 8357.31
	IEC name: ConfigHandling._stAxisConfig.lrModuloMax
Lag error	
Lag error window DT1	Lag error from which drive train 1 signals a fault (in user units).
	Index: 8510.4
	IEC name: -
Limit values from startup	
Maximum speed at motor shaft	Maximum permitted speed at the motor shaft calculated from motor and gear unit data during startup in [1/min].
	Index: 8360.9
	IEC name: -

29179459/EN – 03/2020

Parameter name	Value
Maximum torque at motor shaft	Maximum permitted torque at the motor shaft calculated from motor and gear unit data during startup in [Nm].
	Index: 8360.11
	IEC name: -
Limit values	
Skew error window	Position window for the permitted skew error between the axis group members (in user units). If this window is exceeded, an error will be issued and the set fault response will be performed. **Default value: 0**
	Index: 50011.24
	IEC name: Controller.MAC.Config.SkewLeveling.lrSkewErrorWindow
Lag error window	Window of the actual lag error (in user units) No lag error is created when the setting is "0". **Default value: 0**
	Index: 50012.03
	IEC name: Controller.PositionController.Config.lrLagErrorWindow

Drive functions

Scaling

Parameter name	Value
Encoders	
Actual position source	Encoder that acts as a source for generating the actual position.
	Index: 8565.3
Inverter scaling	
For setting the scaling of the inverter using the position, speed, and acceleration parameters.	
Index: 8554.1-4 (position), 8557.1-4 (speed), 8560.1-4 (acceleration)	
Recommended resolution	
Calculated recommendation for setting the resolution.	
Index: -	
"Apply resolution to drive train" button	Apply the set resolution to the drive train.

29179459/EN – 03/2020

FCB 10 Interpolated position control

Parameter name	Value
FCB 10 Interpolated position control	
Mean value filter time	Setting the filter time constant for interpolating positioning control.
	Index: 8510.3

FCB 12 reference travel

Parameter name	Value
FCB 12 Reference travel	
Type	• Deactivated • Zero pulse – negative direction • Reference cam – negative end • Reference cam – positive end • Positive limit switch • Negative limit switch • Reference cam flush – limit switch positive • Reference cam flush – limit switch negative • Referencing without reference travel • Fixed stop positive • Fixed stop negative
	Index: 8552.1
Reference to zero pulse	Activates or deactivates referencing to zero pulse
	Index: 8552.2
Reference offset	Deviation of the cam from the machine zero
	Index: 8552.5
Search speed	Search speed for reference travel
	Index: 8552.8
Retraction speed	Retraction speed for reference travel
	Index: 8552.9
Acceleration	Acceleration of reference travel
	Index: 8552.11
Deceleration	Deceleration of reference travel
	Index: 8552.12
Advanced settings	
Go to home position	Activates or deactivates homing
	Index: 8552.3

29179459/EN – 03/2020

Parameter name	Value
Home position	Home position that is approached automatically after reference travel is complete.
	Index: 8552.7
Homing speed	Speed for approaching the home position after referencing.
	Index: 8552.10
Jerk time	Homing jerk time
	Index: 8552.13
Speed changeover before fixed stop	For setting whether search speed changes over to retraction speed.
	Index: 8552.4
Dwell time at fixed stop	Dwell time at fixed stop
	Index: 8552.15
Torque limit fixed stop	Limits the torque when referencing to the fixed stop.
	Index: 8552.14

Controller functions

Basic settings

Parameter name	Description
Basic settings	
Fault response	Behavior of profile generation in the event of a fault in the axis assigned to it: • Stop without ramps The profile generator abruptly freezes at the current target position. • Stop at application limit The profile generator creates a stop profile with the deceleration specified in the application limits. • Stop with emergency stop deceleration The profile generator creates a stop profile with the specified emergency stop deceleration. • Following the axis The profile generator creates a stop profile based on the course of the actual position of the axis.
	Index: 50000.11
	IEC name: ProfileGeneration.Config.eErrorReaction

Reference travel

Parameter name	Value
Reference travel	

29179459/EN – 03/2020

Parameter name	Value
Reference travel type	• Deactivated • Reference cam – negative end • Reference cam – positive end • Positive limit switch • Limit switch negative • Referencing without reference travel with enable • Reference cam flush – limit switch positive • Reference cam flush – limit switch negative
	Index: 50007.1
	IEC name: ProfileGeneration.Homing.Config.eReferenceTravelType
Reference offset	Deviation of the cam from the machine zero
	Index: 50007.2
	IEC name: ProfileGeneration.Homing.Config.lrReferenceOffset
Search speed	Search speed for reference travel
	Index: 50007.4
	IEC name: ProfileGeneration.Homing.Config.lrSearchVelocity
Retraction speed	Retraction speed for reference travel
	Index: 50007.5
	IEC name: ProfileGeneration.Homing.Config.lrClearVelocity
Acceleration	Acceleration of reference travel
	Index: 50007.6
	IEC name: ProfileGeneration.Homing.Config.lrAcceleration
Deceleration	Deceleration during reference travel
	Index: 50007.7
	IEC name: ProfileGeneration.Homing.Config.lrDeceleration
Limit switch debouncing time	Debouncing time for the limit switches
	Index: 50007.3
	IEC name: ProfileGeneration.Homing.Config.lrLimitSwitchDebouncingTime
Advanced settings	
Go to home position	Activates or deactivates homing
	Index: 50007.9
	IEC name: ProfileGeneration.Homing.Config.xMoveToStartPosition

29179459/EN – 03/2020

Parameter name	Value
Home position	Position that is approached automatically after referencing.
	Index: 50007.10
	IEC name: ProfileGeneration.Homing.Config.lrStartPosition
Homing speed	Speed for approaching the home position after referencing.
	Index: 50007.11
	IEC name: ProfileGeneration.Homing.Config.lrStartPosVelocity
Jerk	Jerk time for reference travel
	Index: 50007.8
	IEC name: ProfileGeneration.Homing.Config.lrJerk

Speed specification

Parameter name	Value
Speed setpoint	
Stop at position	• **Off** When the operating mode is stopped, the drive stops directly at its position. • Absolute When the operating mode is stopped, the drive moves to a certain absolute position. • Relative When the operating mode is stopped, the drive moves on from its current position again by the values specified at the stop position.
	Index: 50002.1
	IEC name: ProfileGeneration.Velocity.Config.stStopAtPosition.eStopMode
Stop position	Stop position that is approached when the operating mode is stopped (in user units).
	Index: 50002.2
	IEC name: ProfileGeneration.Velocity.Config.stStopAtPosition.lrStopPosition

Absolute positioning

Parameter name	Value
Absolute positioning	

29179459/EN – 03/2020

Parameter name	Value
Without referenced encoder	• Yes Allow positioning if the inverter has not yet been referenced. • **No** Do not allow positioning if the inverter has not yet been referenced.
	Index: 50003.1
	IEC name: ProfileGeneration.Positioning.Config.xWithoutReferencedEncoder
Target position monitoring	• On Check whether the target position is within the set software limit switches. If the target position is outside the set software limit switches, the motion is not started and an error is displayed. • **Off** Do not check whether the target position is within the set software limit switches.
	Index: 50003.2
	IEC name: ProfileGeneration.Positioning.Config.xTargetPositionMonitoring

Relative positioning

Parameter name	Value
Relative positioning	
Continue relative movement	• Yes Continue an interrupted relative positioning if the axis has meanwhile left the interpolating operating mode, e.g. due to enable being canceled. • **No** Do not continue triggered relative positioning after interruption.
	Index: 50004.1
	IEC name: ProfileGeneration.PositioningRelative.Config.xContinueRelativeMove

29179459/EN – 03/2020

Parameter name	Value
Behavior on target position change	• **Off** The target position cannot be changed during movement. • Based on start position The target position can be changed during ongoing movement. The new target position refers to the original start position. • Based on actual position The target position can be changed during ongoing movement. The new target position refers to the current position of profile generation. • Based on target position The target position can be changed during ongoing movement. The new target distance is based on the last target position and is applied with a positive edge change of input signal *xAcceptNewDistance* (and a pending start signal).
	Index: 50004.2
	IEC name: ProfileGeneration.PositioningRelative.Config.eMode
Target position monitoring	• Yes Before traveling to the target position, check whether the target position is within the set software limit switches. If the target position is outside the set software limit switches, the motion is not started and an error is displayed. • **No** Before traveling to the target position, do not check if the target position is within the set software limit switches.
	Index: 50004.3
	IEC name: ProfileGeneration.PositioningRelative.Config.xTargetPositionMonitoring

29179459/EN – 03/2020

Direct coupling

Parameter name	Value
Tracking	
Master source	• **User program** Depending on this setting, the automatic code generation creates a master-slave connection in the action *SEW_PRG.LinkInterfaces*. The structure *MasterUserProgram* from the global variable *Interface_AxisName* is used as the master source. The user must provide the master values to this structure in the cyclic task. • Configured axis The setpoints of the slave axis are adopted from the axis selected under "Master axis name". • EncoderInterface The setpoints of the slave axis are adopted from the EncoderInterface selected under "Encoder-Interface name". • None
	Index: 50005.5
	IEC name: -
Master axis name **Information:** Visible with master source "Configured axis".	Selects the master axis.
	Index: 50005.1
	IEC name: -
Name of EncoderInterface **Information:** Visible with master source "EncoderInterface".	Selection of the EncoderInterface
	Index: 50005.11
	IEC name: -
Settings of the master source	
Modulo minimum	Modulo limit minimum (in user units)
	Index: 50005.4
	IEC name: ProfileGeneration.Tracking.Config.lrMasterModuloMin
Modulo maximum	Modulo limit maximum (in user units)
	Index: 50005.3
	IEC name: ProfileGeneration.Tracking.Config.lrMasterModuloMax
Number of decimal places	Number of decimal places of the master signal
	Index: 50005.8
	IEC name: ProfileGeneration.Tracking.Config.uiMasterResolution

29179459/EN – 03/2020

Parameter name	Value
Time factor for speed	Time base applicable to the speed of the master signal: • **min**$^{-1}$ • 1/s
	Index: 50005.9
	IEC name: ProfileGeneration.Tracking.Config.stTimeBaseFactor.eVelocity
Time factor for acceleration	Time base applicable to the acceleration of the master signal: • **1/min**^{2} • 1/(min*s) • 1/s^{2}
	Index: 50005.10
	IEC name: ProfileGeneration.Tracking.Config.stTimeBaseFactor.eAcceleration
Master/slave gear ratio	
Numerator	Numerator value of the gear ratio between master and slave in user units of the slave axis
	Index: 50005.6
	IEC name: ProfileGeneration.Tracking.In.diTracking-Numerator
Denominator	Denominator value of the gear ratio between master and slave in user units of the master axis
	Index: 50005.7
	IEC name: ProfileGeneration.Tracking.In.diTracking-Denominator

29179459/EN – 03/2020

Synchronous operation (gearing)

Parameter name	Value
Synchronous operation (gearing)	
Master source	• User program (in preparation) Depending on this setting, the automatic code generation creates a master-slave connection in the action *SEW_PRG.LinkInterfaces*. The structure *MasterUserProgram* from the global variable *Interface_AxisName* is used as the master source. The user must provide the master values to this structure in the cyclic task. • Configured axis The setpoints of the slave axis are adopted from the axis selected under "Master axis name". • EncoderInterface The setpoints of the slave axis are adopted from the EncoderInterface selected under "EncoderInterface name". • None
	Index: 50009.5
	IEC name: -
Master axis name **Information:** Visible with master source "Configured axis".	Selects the master axis.
	Index: 50009.1
	IEC name: -
Name of EncoderInterface **Information:** Visible with master source "EncoderInterface".	Selection of the EncoderInterface
	Index: 50009.11
	IEC name: -
Settings of the master source	
Modulo minimum	Modulo minimum
	Index: 50009.4
	IEC name: ProfileGeneration.Gearing.Config.lrMasterModuloMin
Modulo maximum	Modulo maximum
	Index: 50009.3
	IEC name: ProfileGeneration.Gearing.Config.lrMasterModuloMax
Number of decimal places	Number of decimal places
	Index: 50009.28
	IEC name: ProfileGeneration.Gearing.Config.uiMasterResolution

29179459/EN – 03/2020

Parameter name	Value
Time factor for speed	Time factor for speed
	Index: 50009.29
	IEC name: ProfileGeneration.Gearing.Config.stTimeBaseFactor.eVelocity
Time factor for acceleration	Time factor for acceleration
	Index: 50009.30
	IEC name: ProfileGeneration.Gearing.Config.stTimeBaseFactor.eAcceleration
Master/slave gear ratio	
Numerator	Numerator factor of the synchronous operation in user units of the slave axis
	Index: 50009.6
	IEC name: ProfileGeneration.Gearing.In.dGearingNumerator
Denominator	Denominator factor of the synchronous operation in user units of the master axis
	Index: 50009.7
	IEC name: ProfileGeneration.Gearing.In.dGearingDenominator
Synchronizing	
Synchronization behavior	• Direct with master in positive direction of movement • Direct with master in negative direction of movement • Direct with master in positive or negative direction of movement • With reference position and master in positive direction of movement • With reference position and master in negative direction of movement • With reference position and master in positive or negative direction of movement • With point of synchronism
	Index: 50009.13
	IEC name: ProfileGeneration.Gearing.CONFIG.Start.eStartMode
Synchronization transition	• None • Master based • Time based
	Index: 50009.22
	IEC name: ProfileGeneration.Gearing.CONFIG.Start.eStartTransition

29179459/EN – 03/2020

Parameter name	Value
Synchronization distance	Synchronization distance in user units (master)
	Index: 50009.8
	IEC name: ProfileGeneration.Gearing.CONFIG.Start.lrGearInDistance
Synchronization time	Synchronization time in [s]
	Index: 50009.20
	IEC name: ProfileGeneration.Gearing.CONFIG.Start.lrGearInTime
Synchronization offset	Offset in user units (master)
	Index: 50009.9
	IEC name: ProfileGeneration.Gearing.CONFIG.Start.lrGearInOffset
Reference position during synchronization	Reference position during synchronization
	Index: 50009.10
	IEC name: ProfileGeneration.Gearing.CONFIG.Start.lrGearInReferencePosition
Desynchronizing	
Desynchronization behavior	• Direct with master in positive direction of movement • Direct with master in negative direction of movement • Direct with master in positive or negative direction of movement • With stop position of external master and positive direction of movement • With stop position of external master and negative direction of movement • With stop position of external master and positive or negative direction of movement
	Index: 50009.14
	IEC name: ProfileGeneration.Gearing.CONFIG.Stop.eStopMode
Desynchronization transition	• None • Master based • Time based
	Index: 50009.23
	IEC name: ProfileGeneration.Gearing.CONFIG.Stop.eStopTransition
Desynchronization distance	Desynchronization distance in user units (master)
	Index: 50009.11
	IEC name: ProfileGeneration.Gearing.CONFIG.Stop.lrGearOutDistance

29179459/EN – 03/2020

Parameter name	Value
Desynchronization time	Desynchronization time in [s]
	Index: 50009.20
	IEC name: ProfileGeneration.Gearing.CONFIG.Stop.lrGearOutTime
Stop position after desynchronization	Stop position after desynchronization
	Index: 50009.12
	IEC name: ProfileGeneration.Gearing.CONFIG.Stop.lrGearOutStopPosition
X offset correction	
Mode	• Absolute – Interpret the transferred value as absolute • Relative – Interpret the transferred value as relative
	Index: 50009.31
	IEC name: ProfileGeneration.Gearing.XOffsetCorrection.eOffsetCorrectionMode
Transition	Type of travel profile: • Master based • Profile based
	Index: 50009.32
	IEC name: ProfileGeneration.Gearing.XOffsetCorrection.eTransitionType
Master distance	Distance of the master axis within which a master-based offset correction is performed.
	Index: 50009.33
	IEC name: ProfileGeneration.Gearing.XOffsetCorrection.lrMasterDistance
Y offset correction	
Mode	• Absolute - Interpret the transferred value as absolute • Relative – Interpret the transferred value as relative
	Index: 50009.34
	IEC name: ProfileGeneration.Gearing.YOffsetCorrection.eOffsetCorrectionMode
Transition	Type of travel profile: • Master based • Profile based
	Index: 50009.35
	IEC name: ProfileGeneration.Gearing.YOffsetCorrection.eTransitionType

29179459/EN – 03/2020

Parameter name	Value
Master distance	Distance of the master axis within which a master-based offset correction is performed.
	Index: 50009.36
	IEC name: ProfileGeneration.Gearing.YOffsetCorrection.lrMasterDistance

Camming

INFORMATION

This configuration menu is only available with the "MOVIKIT® MultiMotion Camming" license.

Parameter name	Value
Electronic cam (camming)	
Master source	• **User program** Depending on this setting, the automatic code generation creates a master-slave connection in the action *SEW_PRG.LinkInterfaces*. The structure *MasterUserProgram* from the global variable *Interface_AxisName* is used as the master source. The user must provide the master values to this structure in the cyclic task. • Configured axis The setpoints of the slave axis are adopted from the axis selected under "Master axis name". • EncoderInterface The setpoints of the slave axis are adopted from the EncoderInterface selected under "EncoderInterface name". • None
	Index: 50006.5
	IEC name: -
Master axis name **Information:** Visible with master source "Configured axis".	Selects the master axis.
	Index: 50006.1
	IEC name: -
EncoderInterface name **Information:** Visible with master source "EncoderInterface".	Selection of the EncoderInterface
	Index: 50006.11
	IEC name: -
General	

29179459/EN – 03/2020

Parameter name	Value
Behavior at end of cycle	• Adjust reference position of slave axis • Keep reference position of slave axis
	Index: 50006.19
	IEC name: ProfileGeneration.Camming.CONFIG.Start.eCycleMode
Synchronizing	
Synchronization behavior	• Direct synchronization with master in positive direction of movement • Direct synchronization with master in negative direction of movement • Direct synchronization with master in positive or negative direction of movement • Synchronization with reference position and master in positive direction of movement • Synchronization with reference position and master in negative direction of movement • Synchronization with reference position and master in positive or negative direction of movement
	Index: 50006.13
	IEC name: ProfileGeneration.Camming.CONFIG.Start.eStartMode
Synchronization distance	Synchronization distance in user units (master)
	Index: 50006.8
	IEC name: ProfileGeneration.Camming.CONFIG.Start.lrGearInDistance
Synchronization offset	Offset during synchronization to the reference position in user units (master)
	Index: 50006.9
	IEC name: ProfileGeneration.Camming.CONFIG.Start.lrGearInOffset
Reference position during synchronization	Reference position in user units (master)
	Index: 50006.10
	IEC name: ProfileGeneration.Camming.CONFIG.Start.lrGearInReferencePosition
Keep last configured phase correction	• ON – When deactivating and re-activating the operating mode, keep the last configured phase correction. • OFF – When deactivating and activating the operating mode again, initialize the last configured phase correction with 0.
	Index: 50006.15
	IEC name: ProfileGeneration.Camming.CONFIG.Start.xStartWithLastCorrection
Desynchronizing	

29179459/EN – 03/2020

Parameter name	Value
Desynchronization behavior	• Direct desynchronization with master in positive direction of movement • Direct desynchronization with master in negative direction of movement • Direct desynchronization with master in positive or negative direction of movement • Desynchronization with stop position of internal master and in positive direction of movement • Desynchronization with stop position of internal master and in negative direction of movement • Desynchronization with stop position of internal master and in positive or negative direction of movement • Desynchronization with stop position of external master and in positive direction of movement • Desynchronization with stop position of external master and in negative direction of movement • Desynchronization with stop position of external master and in positive or negative direction of movement
	Index: 50006.14
	IEC name: ProfileGeneration.Camming.CONFIG.Stop.eStopMode
Desynchronization distance	Desynchronization distance in user units (master)
	Index: 50006.11
	IEC name: ProfileGeneration.Camming.CONFIG.Stop.lrGearOutDistance
Stop position after desynchronization	Stop position after desynchronization in user units (master)
	Index: 50006.12
	IEC name: ProfileGeneration.Camming.CONFIG.Stop.lrGearOutStopPosition
Curve change	

29179459/EN – 03/2020

Parameter name	Value
Curve change trigger	• Direct • With position of internal master and positive direction of movement • With position of internal master and negative direction of movement • With position of internal master and positive or negative direction of movement • With position of external master and positive direction of movement • With position of external master and negative direction of movement • With position of external master and positive or negative direction of movement
	Index: 50006.17
	IEC name: ProfileGeneration.Camming.CONFIG.Change.eChangeMode
Behavior on curve change	• Direct
	Index: 50006.16
	IEC name: ProfileGeneration.Camming.CONFIG.Change.eTransitionMode
Reference position on curve change	Reference position on curve change
	Index: 50006.18
	IEC name: ProfileGeneration.Camming.CONFIG.Change.lrChangeReferencePosition
X offset correction	
Mode	• Absolute – Interpret the transferred value as absolute • Relative – Interpret the transferred value as relative
	Index: 50009.31
	IEC name: ProfileGeneration.Gearing.XOffsetCorrection.eOffsetCorrectionMode
Transition	Type of travel profile: • Master based • Profile based
	Index: 50009.32
	IEC name: ProfileGeneration.Gearing.XOffsetCorrection.eTransitionType
Master distance	Distance of the master axis within which a master-based offset correction is performed.
	Index: 50009.33
	IEC name: ProfileGeneration.Gearing.XOffsetCorrection.lrMasterDistance
Y offset correction	

29179459/EN – 03/2020

Parameter name	Value
Mode	• Absolute – Interpret the transferred value as absolute • Relative – Interpret the transferred value as relative
	Index: 50009.34
	IEC name: ProfileGeneration.Gearing.YOffsetCorrection.eOffsetCorrectionMode
Transition	Type of travel profile: • Master based • Profile based
	Index: 50009.35
	IEC name: ProfileGeneration.Gearing.YOffsetCorrection.eTransitionType
Master distance	Distance of the master axis within which a master-based offset correction is performed.
	Index: 50009.36
	IEC name: ProfileGeneration.Gearing.YOffsetCorrection.lrMasterDistance

Touchprobe 1

INFORMATION

Only included if the function is activated in the "Basic settings" configuration menu under "Functions used".

Parameter name	Value
General	
Touchprobe source	Selection of the touchprobe source: • Inverter • Configured axis • User program
	Index: 50008.1
	IEC name: -
Mode	Selection of the touchprobe mode: • Single • Multiple
	Index: 50008.2
	IEC name: TouchProbe.Config.eTouchProbeMode
Trigger	
Source	Source for activating the trigger for recording a signal
	Index: 8352.10
	IEC name: -

Parameter name	Value
Event	Selection of the type of edge that triggers the system: • Rising edge • Falling edge • Rising and falling edge
	Index: 8352.11
	IEC name: -
Sensor dead time rising edge	Dead time of the sensor used for the rising edge at the trigger input. This time will be included in the calculation of the touchprobe event value.
	Index: 8352.12
	IEC name: Parameter.TouchProbe1.TriggerSensorDeadTimeRisingEdge
Sensor dead time falling edge	Dead time of the sensor used for the falling edge at the trigger input. This time will be included in the calculation of the touchprobe event value.
	Index: 8352.13
	IEC name: Parameter.TouchProbe1.TriggerSensorDeadTimeFallingEdge
Counter	Counter of trigger events. This value is incremented by the value 1 with each trigger event.
	Index: 8352.14
	IEC name: Parameter.TouchProbe1.TriggerCounter
Data source	
Data source	Data source for trigger recording
	Index: 8352.30
	IEC name: -
PO data format	Selection of the format of the process data: • 16 bits • 32 bits – Big Endian • 32 bits – Little Endian The data format is specified when accessing the PO data words. The PO data has a word width of 16 bits and can be compiled as a 32-bit value using the PO data format. This parameter has no effect for data sources with a word width of 32 bits.
	Index: 8352.31
	IEC name: -
Process data - Modulo minimum	Modulo minimum of the source. Is only required if the data source is to be recorded at the time of the modulo change.
	Index: 8352.32
	IEC name: -

29179459/EN – 03/2020

Parameter name	Value
Process data - Modulo maximum	Modulo maximum of the source. Is only required if the data source is to be recorded at the time of the modulo change.
	Index: 8352.33
	IEC name: -
Process data – dead time	Setting to compensate for the runtime of the process data
	Index: 8352.34
	IEC name: -
Process data – cycle time	Setting to compensate for the runtime of the process data
	Index: 8352.35
	IEC name: -

29179459/EN – 03/2020

Position controller

Parameter name	Description
Position controller	
Position controller	Switching on/off the position controller
	Index: 50012.1
	IEC name: Controller.PositionController.Config.xDisable
P gain	Position controller gain for minimizing lag errors
	Index: 50012.2
	IEC name: Controller.PositionController.Config.lrP-Gain

Encoder evaluation

INFORMATION

Bear in mind that the modulo reference could be lost when acknowledging an encoder fault.

Parameter name	Description
Encoder evaluation	
Encoder type	Encoder type selection: • **Motor encoder** • External encoder • Motor encoder and external encoder • Motor encoder and low-resolution EtherCAT® encoder • High-resolution EtherCAT® encoder • Low-resolution EtherCAT® encoder • High and low-resolution EtherCAT® encoders
	Index: 50013.1
	IEC name: Controller.EncoderEvaluation.Config.eActPos_EncSelector
New initialization of encoder when changing encoder source	Turn on encoder re-initialization when changing encoder source • **Yes** • No
	Index: 50013.6
	IEC name: Controller.EncoderEvaluation.Config.xDontInitializeAtEncSelectorChange

29179459/EN – 03/2020

Parameter name	Description
Time constant	Integral time for encoder adjustment in [s] **Default value: 0.1**
	Index: 50013.3
	IEC name: Controller.EncoderEvaluation.Config.lrInputFilterTime_ExtEnc
Dead time	Dead time of the external encoder in [s] Contact SEW-EURODRIVE or the manufacturer of the external encoder to obtain the relevant value. **Default value: 0**
	Index: 50013.4
	IEC name: Controller.EncoderEvaluation.Config.lrDeadtime_ExtEnc
Filter of the low-resolution EtherCAT® encoder	Switching on/off the low-resolution EtherCAT® encoder
	Index: 50013.5
	IEC name: Controller.EncoderEvaluation.Config.xInterpolationFilterOn
Advanced settings	
P gain Option only visible with combined encoder evaluation.	Amplification factor of encoder evaluation
	Index: 50013.2
	IEC name: _fbController._fbEncoderEvaluation.stConfig.lrActPos_EncSelector

Interpolation

Parameter name	Description
Interpolation mode	
Mode	Selection of the interpolation method: • **Linear** • Polynominal 3 • Spline
	Index: 50006.120
	IEC name: -

29179459/EN – 03/2020

AntiSlosh

INFORMATION

Only included if the function is activated in the "Basic settings" configuration menu under "Functions used".

Parameter name	Description
General parameters	
Natural frequency	Natural frequency of the "container-liquid" system in [Hz]
	Index: 50006.100
	IEC name: stActiveConfig.lrNaturalFrequency
Damping	Damping of the "container-liquid" system
	Index: 50006.101
	IEC name: stActiveConfig.lrDamping
Nominal speed of application	Nominal speed of the application in [1/min]
	Index: 50006.102
	IEC name: stActiveConfig.lrVelocity
Travel profile parameter	
Start of cycle	Position of the master at the beginning of the cycle
	Index: 50006.103
	IEC name: stActiveConfig.lrCycleStart
End of cycle	Position of the master at the end of the cycle
	Index: 50006.104
Cycle length	Length of the master cycle, i.e. the difference between "end of cycle" and "start of cycle".
	Index: 50006.111
	IEC name: -
Starting point of movement X1	Position of the master at the beginning of the movement of the slave axis
	Index: 50006.106
	IEC name: stActiveConfig.lrStartPos
Distance of movement ΔX2	Distance covered by the master during the movement of the slave axis
	Index: 50006.107
	IEC name: stActiveConfig.lrDeltaX2
Stroke of movement ΔY	Distance covered by the slave axis during the movement.
	Index: 50006.105
	IEC name: stActiveConfig.lrStepDistance

29179459/EN – 03/2020

Parameter name	Description
X offset	Offset of the curve in x-direction
	Index: 50006.112
	IEC name: -
Y offset	Offset of the curve in y-direction
	Index: 50006.113
	IEC name: -
Jerk in the middle of movement	Jerk in the middle of movement $\frac{d^3 y(x_m)}{dx^3}$ If "Snap at the starting point of movement" and "Snap at the end point of movement" are 0, favorable values for "Jerk in the middle of movement" are between 0 and -100.
	Index: 50006.110
	IEC name: stActiveConfig.lrJerkMid
Snap at the starting point of the movement	Derivation of the jerk at the starting point of the movement $\frac{d^4 y(x_a)}{dx^4}$ The favorable value range for "Snap at the starting point of movement" is between 0 and 2000.
	Index: 50006.108
	IEC name: stActiveConfig.lrSnapStart
Snap at the end point of movement	Derivation of the jerk at the end point of the movement $\frac{d^4 y(x_e)}{dx^4}$ The favorable value range for "Snap at the end point of movement" is between -2000 and 0.
	Index: 50006.109
	IEC name: stActiveConfig.lrSnapEnd

29179459/EN – 03/2020

Anti-sway control

INFORMATION

Only included if the function is activated in the "Basic settings" configuration menu under "Functions used".

Anti-sway control

Parameter name	Value
Anti-sway control	
Application type	Selection of the application type • No sway • Tower sway • Pendulum sway (in preparation) • Belly sway (in preparation) • Fluid sway (in preparation) • Spring sway (in preparation)
	Index: 50014.1
	IEC name: Controller.AntiSway.Config.eApplicationType
Conversion factor for user units in meters	Position relationship between the user unit from the drive train and one meter. If the value is set to "0", the user unit corresponds to one meter. For all other values, one meter is the product of the user unit and the specified value.
	Index: 50014.2
	IEC name: Controller.AntiSway.Config.lrUserUnitToMeter
Setpoint correction selection	• Off • Anti-sway • Bandstop • Tension build-up time filter
	Index: 50014.30
	IEC name: Controller.AntiSway.Config.SetpointCorrection.eSelector
Source of lifting height	• **No master** Use "Distance between lifting and traveling trolley" as source • Axis group • Axis
	Index: 50014.13
	IEC name: -
Basic settings (setting fields visible depending on application type)	

29179459/EN – 03/2020

Parameter name	Value
Height of the tower	Tower height in [m]
	Index: 50014.10
	IEC name: Controller.AntiSway.Config.DriveTrain.lrHeightTower
Distance between lifting and traveling trolley	Distance from lifting axis to travel axis in [m]
	Index: 50014.11
	IEC name: Controller.AntiSway.Config.DriveTrain.lrDistanceHoistToCar
Mass of the trolley	Mass of the lifting gear without payload mass and without shuttle in [kg]
	Index: 50014.13
	IEC name: Controller.AntiSway.Config.DriveTrain.lrMassHoist
Mass of the payload	Mass of the payload in [kg] The shuttle is included in the payload.
	Index: 50014.14
	IEC name: Controller.AntiSway.Config.DriveTrain.lrMassPayload
Mass of the tower	Mass of the tower without trolley, lifting gear, load mass and shuttle in [kg]
	Index: 50014.15
	*IEC name: Controller.AntiSway.Config.DriveTrain.*lrMassTower
Support for parameter determination See "Configuration menu" (→ 🗎 52)	Activation of the support to determine the parameters "spring stiffness between tower and trolley" and "damping between tower and trolley". If support is enabled, the additional configuration menu "Support for parameter determination" is displayed. • **Not active** • Active Information: The calculations in the configuration menu "Support for parameter determination" (→ 🗎 52) are based, among others, on the values entered in this configuration menu.
	Index: 50014.250
	IEC name: -
Spring stiffness between tower and trolley	Spring constant between tower and trolley in [Nm/wheel]
	Index: 50014.16
	*IEC name: Controller.AntiSway.Config.DriveTrain.*lrSpringTowerToCar

29179459/EN – 03/2020

Parameter name	Value
Damping between tower and trolley	Damping stiffness between tower and trolley in [Nm/(wheel/s)]
	Index: 50014.17
	IEC name: Controller.AntiSway.Config.DriveTrain.lrDampTowerToCar
Time window	
Jerk time tension build-up	Jerk time for the mechanical tension build-up in [s] Maximum ≤ 2000 * PLC cycle time
	Index: 50014.40
	IEC name: Controller.AntiSway.Config.SetpointCorrection.TensionTimes.lrJerkTime
Ramp time tension build-up	Ramp time for the mechanical tension build-up in [s] Maximum ≤ 2000 * PLC cycle time
	Index: 50014.41
	IEC name: Controller.AntiSway.Config.SetpointCorrection.TensionTimes.lrRampTime

Support for parameter determination

INFORMATION

Only included if "Support for parameter determination" is activated in the "Anti-sway control" configuration menu.

Parameter name	Value
Damping	
Oscillation amplitude	First peak of the oscillation amplitude
	Index: 50014.110
	IEC name: -
Oscillation amplitude	Second peak of the oscillation amplitude
	Index: 50014.111
	IEC name: -
Oscillation period	Time between first and second oscillation amplitude
	Index: 50014.112
	IEC name: -
Degree of damping between tower and trolley	Damping behavior of the oscillation in [Nm/(Rad/s)]
	Index: 50014.20
	IEC name: Controller.AntiSway.Config.DriveTrain.lrDampRatioTowerToCar
Basic settings (setting fields visible depending on application type)	

29179459/EN – 03/2020

Parameter name	Value
Spring stiffness between tower and trolley	Spring constant between tower and trolley in [Nm/wheel]
	Index: 50014.16
	*IEC name: Controller.AntiSway.Config.DriveTrain.*lrSpringTowerToCar
Resonant frequency	
Resonant frequency	Frequency at which the oscillating system can oscillate with maximum amplitude
	Index: 50014.55
	IEC name: -
General data	
Maximum acceleration	Acceleration for deflection calculation in [m/s]
	Index: 50014.113
	IEC name: -
Auxiliary tower mass	Value resulting from all other parameters in [kg]
	Index: 50014.127
	IEC name: -
Auxiliary tower height	Value resulting from all other parameters in [m]
	Index: 50014.126
	IEC name: -
Auxiliary deflection	Value resulting from all other parameters in [rad]
	Index: 50014.128
	IEC name: -
Deflection at lifting height	Deflection at lifting height calculated from the parameter setting in [m]. This value is used for the plausibility check against reality and construction calculations. If the deflection is not plausible, an incorrect parameterization can be assumed.
	Index: 50014.114
	IEC name: -

Advanced settings

Set parameters

Parameter name	Value
Delivery state	
Factory setting	Initialize the software module with default values or suggested values. A possibly configured readjustment will be overwritten. All the other settings of the "controller functions" (→ 🗎 28) remain unchanged.

29179459/EN – 03/2020

Parameter name	Value
Suggested values	
Apply all suggested values	Overwrite all values in the configuration of the software module to which a suggested value is assigned with the corresponding suggested value.

Process data profile

Parameter name	Value
Select process data profile	
Process data profile	Setting regarding how much and which data is exchanged between inverter and MOVI-C® CONTROLLER.
	Index: 50000.10
	IEC name: -
Apply process data settings	
"Apply process data settings" button	Configure the process data interface according to the selected process data profile.

Module identification

Parameter group	Description
Module identification	Includes name and version for identifying the software module.

29179459/EN – 03/2020

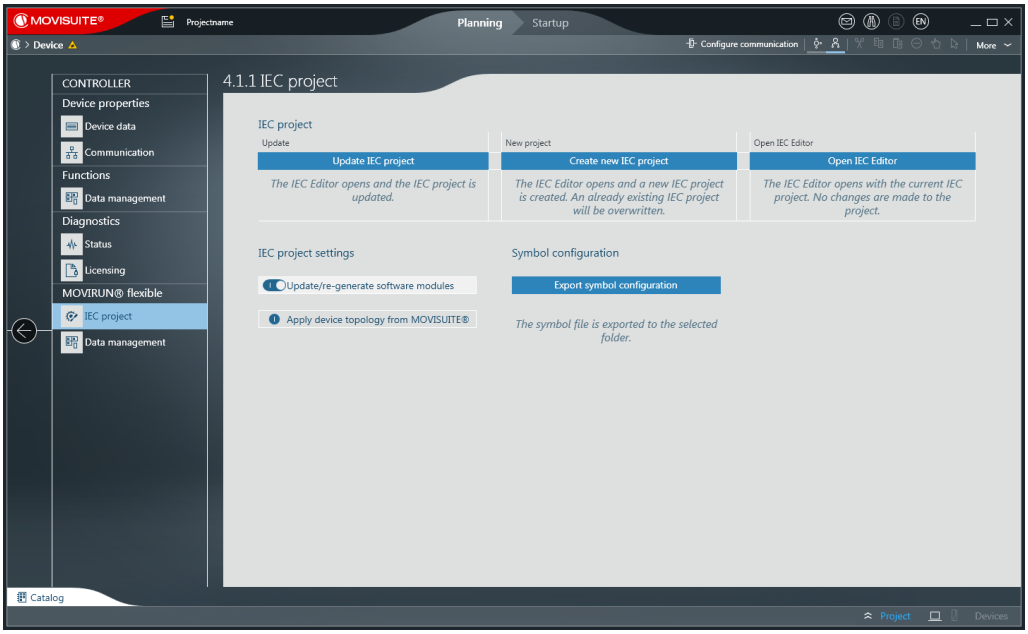
5.4 Generating an IEC project

Carry out the following steps to create an IEC project using automatic code generation and based on the configuration settings in MOVISUITE®.

✓ Configuration of the MOVISUITE® project has been completed.

1. In the function view of MOVISUITE®, click the software module section of the MOVI-C® CONTROLLER.

   ⇨ The "IEC project" menu opens.



27021618448637067

INFORMATION

If you have carried out the configuration in MOVISUITE® using the "Startup" mode and the message "Device cannot be reached" appears, proceed as follows:

- If the MOVI-C® CONTROLLER is not available via the network, switch over to "Planning" mode.
- If the MOVI-C® CONTROLLER is available via the network, carry out a network scan and connect the MOVI-C® CONTROLLER in the network view with the MOVI-C® CONTROLLER in the function view.

2. Click [Create new IEC project].

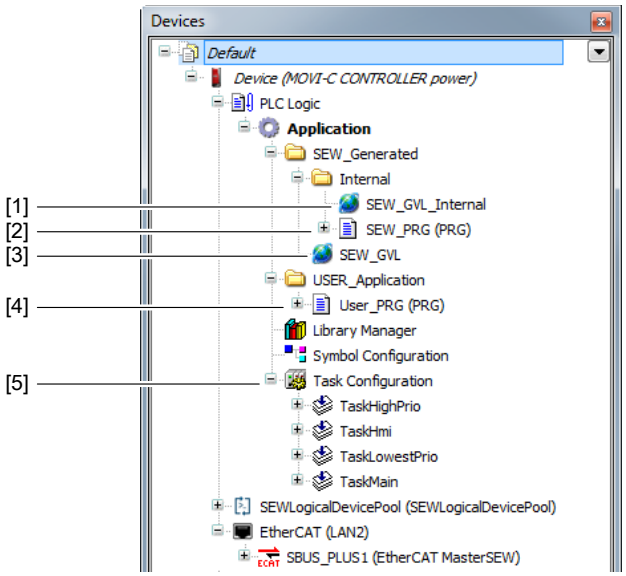
   ⇨ The IEC Editor opens and a new IEC project is created.

INFORMATION

If changes are made to the project structure, to inverter data sets, or to a software module configuration after the IEC project is generated for the first time, a notification symbol is displayed on the MOVI-C® CONTROLLER node. Click on the message icon for more information about the change, and to update the IEC project.

5.4.1 IEC project structure

The IEC project has the following basic structure:



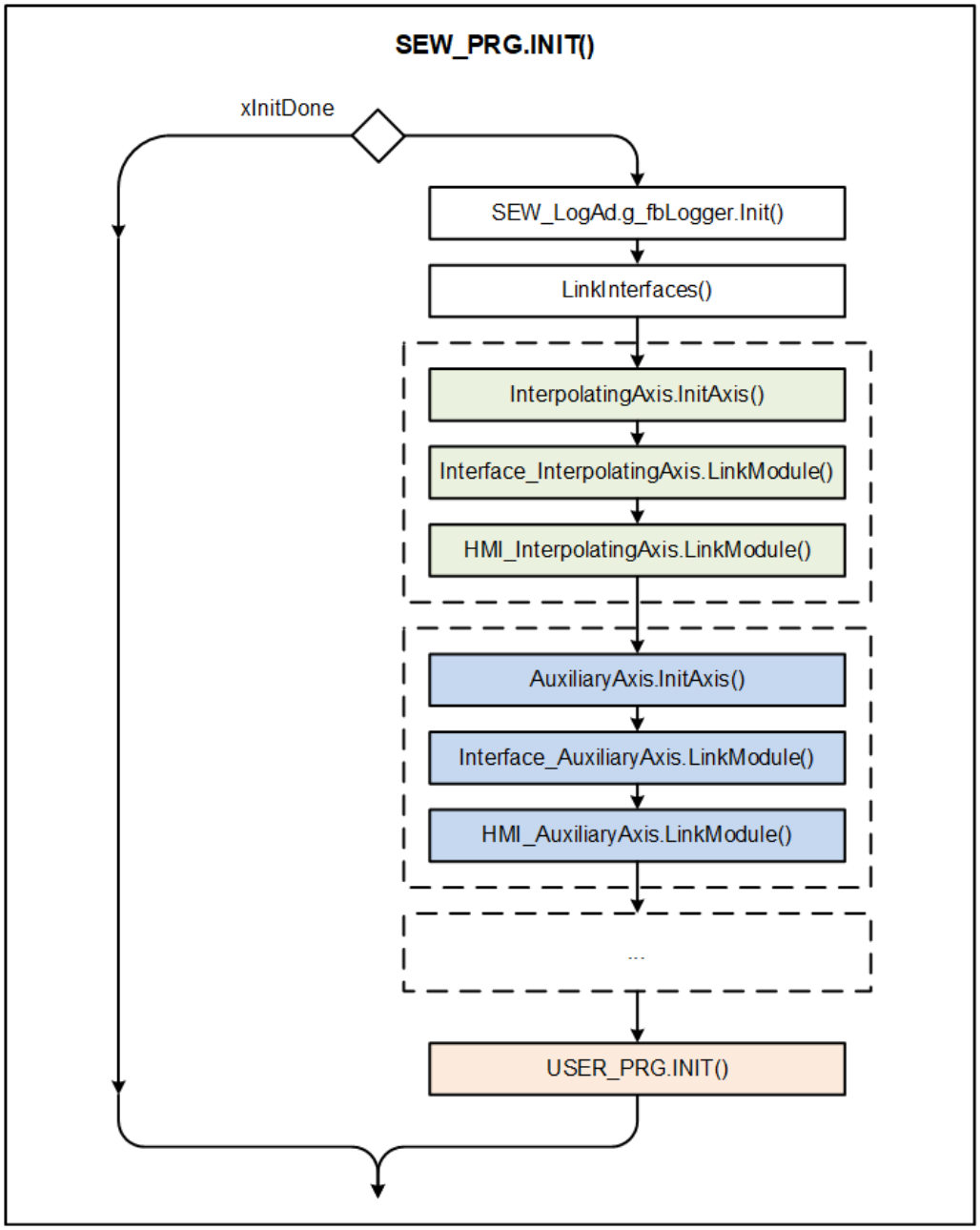
18014423003085323

No.	Name	Description
[1]	SEW_GVL_Internal	The SEW_GVL_Internal global list of variables contains the instances that correspond to the software module used. These variables may not be written to from the user program. In addition, the structure contains an instance as a communication buffer for controlling or monitoring the software module by means of a monitor.
[2]	SEW_PRG	Program that contains all the important instance calls. Automatic code generation recreates this program in accordance with the configuration made in MOVISUITE® each time the IEC project is created, thereby overwriting the previous version. Therefore, you should not make any changes to this program.
[3]	SEW_GVL	The SEW_GVL global list of variables is the interface for accessing the software module features.
[4]	User_PRG	The user program is created once, initially, by automatic code generation. Since the program is not overwritten with each subsequent creation, this is the appropriate place for integrating user programs. The program is divided into five actions. These actions differ in the time at which they are called during the program sequence.
[5]	Task configuration	The list of tasks created in the project. Automatic code generation initially adds tasks that differ in how they are prioritized. The user can add additional programs to existing tasks or create new tasks. It is the responsibility of the user to design the capacity utilization of the tasks to enable the tasks to be processed within the required cycle time. Moving beyond the cyclical tasks, in particular, prevents setpoints for the interpolating axes being generated in time, which means that these axes cannot be operated properly.

29179459/EN – 03/2020

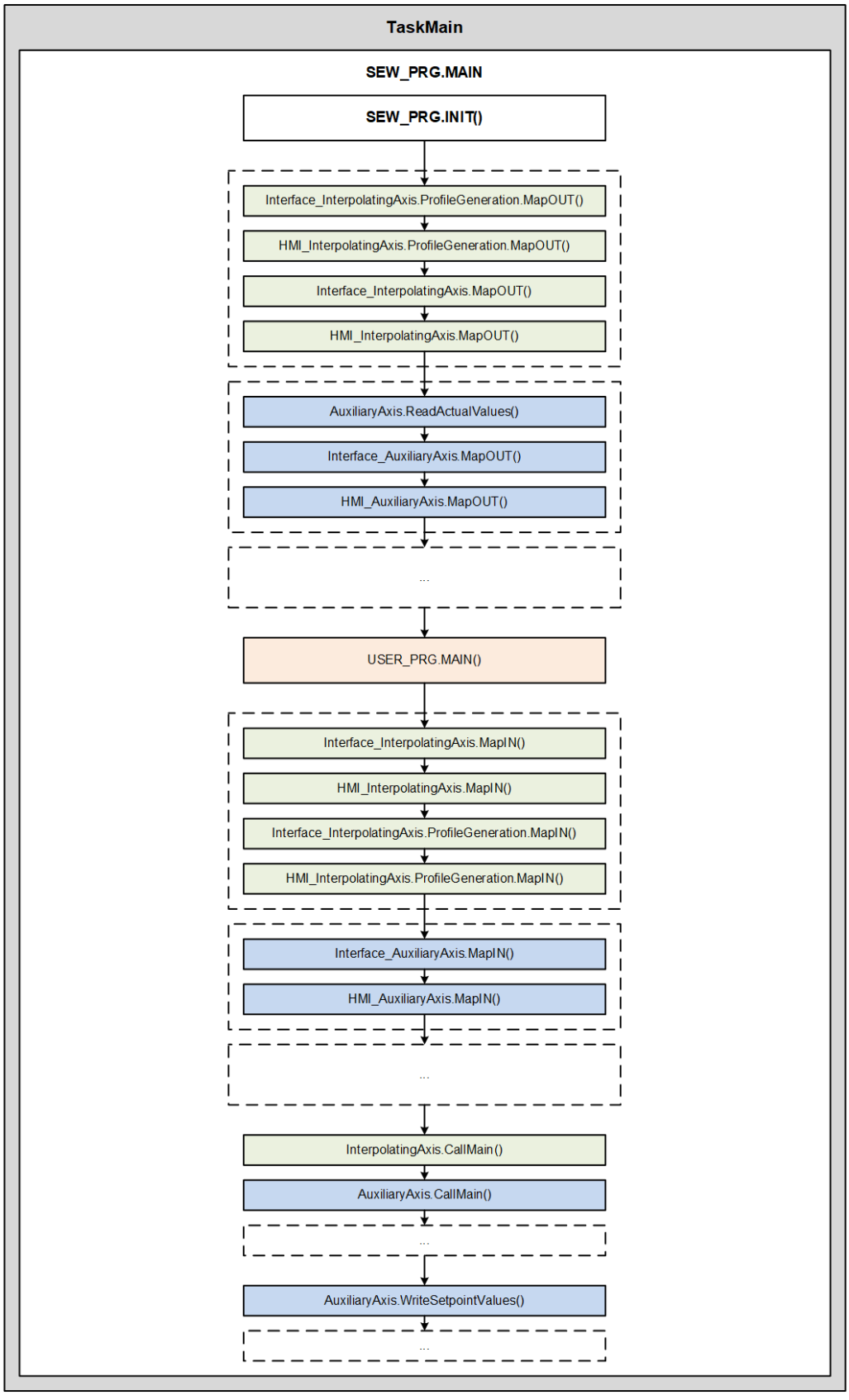
5.5 Integrating a user program

There are various ways to integrate a user program into the IEC project created. The figure below shows the basic structure of the software. In addition, you can see where the actions and methods are processed during program execution. The actions or methods highlighted in red are available for executing the user program. Program calls of an auxiliary axis are indicated in blue. Program calls of an interpolating axis are indicated in green.



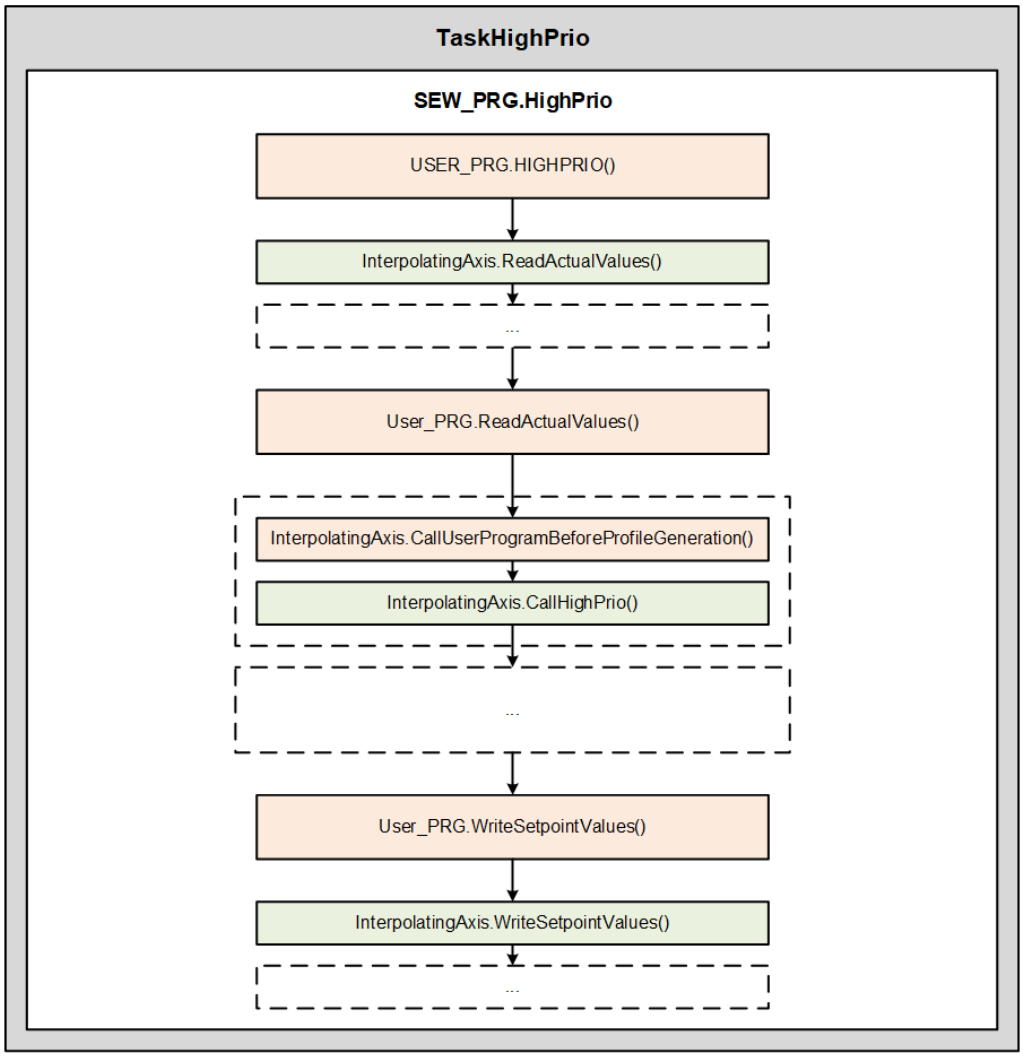
25291027851

29179459/EN – 03/2020



25291030411

29179459/EN – 03/2020



25291032971

5.5.1 Tasks of the mapping functions

Automatic code generation places the mapping to the interfaces in the free-wheeling task because in most cases, the user program is also processed within the free-wheeling task (e.g. in the *Main* action in the *User_PRG* program). Keep in mind that for cyclical tasks (e.g. in profile generation) all information written from the free-wheeling task to the interfaces (e.g. *Interface_Axis1*) is processed asynchronously and, as a result, in a potentially inconsistent manner.

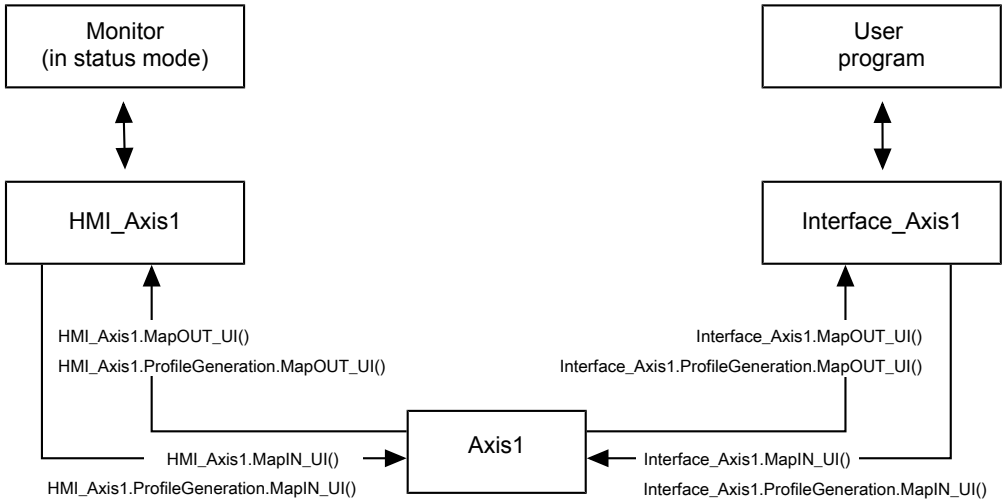
In certain cases, however, it might be important to control the system synchronously to the cyclical task from the user program (e.g. during an on-the-fly changeover). In such cases, the respective control actions of the user program then have to be processed in the cyclical task (e.g. in the *HighPrio* action in the *User_PRG* program). Furthermore, the mapping functions of the status signals or control signals processed within the cyclical task must also be called from within the cyclical task. This means the corresponding calls must be shifted by the *Main* action in the *SEW_PRG* program to the *HighPrio* action in the *SEW_PRG* program.

The following diagram shows an overview of the mapping functions and their tasks that can be used to decide which calls might have to be modified.

29179459/EN – 03/2020

INFORMATION

It is the responsibility of the user to ensure consistency by taking the appropriate measures if necessary.



18014423329103755

- Interface_Axis1.MapOUT_UI() copies the status information from "Axis1" to "Interface_Axis1"
- Interface_Axis1.ProfileGeneration.MapOUT_UI() copies the status information of profile generation from "Axis1" to "Interface_Axis1"
- Interface_Axis1.MapIN_UI() copies the control information from "Interface_Axis1" to "Axis1"
- Interface_Axis1.ProfileGeneration.MapIN_UI() copies the control information of profile generation from "Interface_Axis1" to "Axis1"
- HMI_Axis1.MapOUT_UI() copies the status information from "Axis1" to "HMI_Axis1"
- HMI_Axis1.ProfileGeneration.MapOUT_UI() copies the status information of profile generation from "Axis1" to "HMI_Axis1"
- HMI_Axis1.MapIN_UI() copies the control information from "HMI _Axis1" to "Axis1"
- HMI_Axis1.ProfileGeneration.MapIN_UI() copies the control information of profile generation from "HMI _Axis1" to "Axis1"

29179459/EN – 03/2020

5.5.2 Using the "CallUserProgramBeforeProfileGeneration" method

The *CallUserProgramBeforeProfileGeneration* method is used to integrate a user program immediately before the profile of a specific axis is generated. This allows the system, for example, to calculate free profiles that are not included in the MultiMotion axis functions.

The *CallUserProgramBeforeProfileGeneration* method has a special role. It is an optional method and not included in the project template.

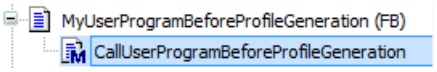
Carry out the following steps to integrate a user program using this method:

1. Create a new function block that implements the *IUserProgramBeforeProfileGeneration* interface.

```
MyUserProgramBeforeProfileGeneration
1  FUNCTION_BLOCK MyUserProgramBeforeProfileGeneration IMPLEMENTS  IUserProgramBeforeProfileGeneration
2  VAR_INPUT
3  END_VAR
4  VAR_OUTPUT
5  END_VAR
6  VAR
7  END_VAR
```

20764032779

2. Implement the code using the *CallUserProgramBeforeProfileGeneration* method.



20764035851

3. Declare an instance of the function block, e.g. as a global variable.

```
GVL_User
1  VAR_GLOBAL
2
3  fbMyUserProgramBeforeProfileGeneration: MyUserProgramBeforeProfileGeneration;
4
5  Interface_Axis1 : FB_MultiMotionCammingAxisUI;
6  CamDescription_Axis1 : CamDescription;
7  Interface_Axis2 : FB_MultiMotionCammingAxisUI;
8  CamDescription_Axis2 : CamDescription;
9  END_VAR
```

20764038923

4. Connect the interface of the respective axis instance with the instance of the function block by making a call from within *USER_PRG.Init()*.

```
User_PRG.INIT
1  Axis1.itfUserProgramBeforeProfileGeneration := fbMyUserProgramBeforeProfileGeneration;
2  xInitDone := TRUE;
```

20764041995

⇨ The method of the fieldbus defined is always called from the corresponding position *Axis1.CallUserProgramBeforeProfileGeneration()* during program execution. At runtime, you can switch between the methods of various function blocks.

6 IEC programming

6.1 Opening the IEC project

- If an IEC project has already been generated, select the entry [IEC Editor] under "Tools" from the context menu of the MOVI-C® CONTROLLER in MOVISUITE®.
- If no IEC project has been generated, follow the steps described in the "Generating an IEC project" (→ 🗎 55) chapter.

6.2 User interface

The user interface is implemented in the IEC program by an instance in the global variable list *SEW_GVL*.

Depending on the software module used, the basic MultiMotion functions contained in all software modules are supplemented by additional structures.

The following figure shows the IEC interface of MOVIKIT® MultiMotion:

Interface_Axis1	SEW_MK_MultiMotion.MultiMotionStandardAxis_UI	
xError	BOOL	FALSE
xWarning	BOOL	FALSE
udiMessageID	UDINT	16#00000000
sAdditionalText	STRING(Constants.gc_udiLengthAdditionalText)	''
xReset	BOOL	FALSE
xGetAccessControl	BOOL	TRUE
xControlActive	BOOL	FALSE
xInitDone	BOOL	FALSE
Basic	SEW_IAx.ST_Basic	
Inverter	SEW_IAx.ST_Inverter	
EnergySaving	SEW_IAx.ST_EnergySaving2	
Brake	SEW_IAx.ST_Brake	
SoftwareLimitSwitch	SEW_SWLS.SoftwareLimitSwitch_UI	
PosHoldCtrl	SEW_IAx.ST_PosHoldCtrl	
RotorPosIdentification	SEW_IAx.ST_RotorPosIdentification	
SendObject	SEW_IAx.ST_SendObject	
Homing	SEW_UIDM.ModeHoming_UI	
TouchProbe	SEW_TP.TouchProbe_UI	
Controller	SEW_Ctrlr.Controller_UI	
ProfileGeneration	ProfileGeneration_MultiMotionStandardAxis_UI	

18014419006975499

The sequence of the following chapters corresponds to this structure.

29179459/EN – 03/2020

6.3 Basic functions

6.3.1 Diagnostics

Variables for reporting and writing errors and warnings.

Variable name	Description
xError	Data type – BOOL
	• TRUE – Error present • FALSE – No error present
xWarning	Data type – BOOL
	• TRUE – Warning present • FALSE – No warning present
udiMessageID	Data type: UDINT
	Message ID number
sAdditionalText	Data type: STRING
	Additional message text
xReset	Data type – BOOL
	• TRUE – Reset messages • FALSE – Do not reset messages

6.3.2 Access management

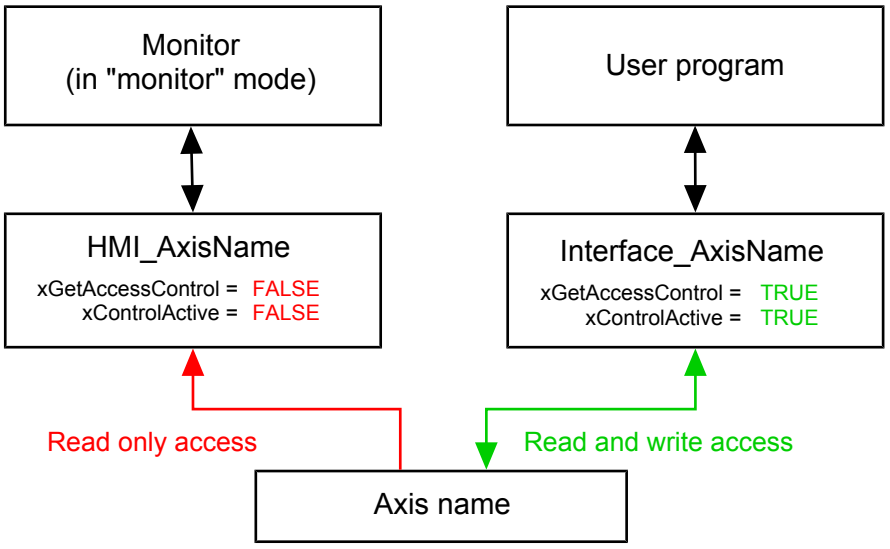
The access management system controls access to the user interface and ensures that no collisions occur during concurrent access to the axis functions.

Variable name	Description
xGetAccessControl	Data type – BOOL
	• TRUE – Request access • FALSE – Return access
xControlActive	Data type – BOOL
	• TRUE – Access granted • FALSE – Access denied

29179459/EN – 03/2020

User program access

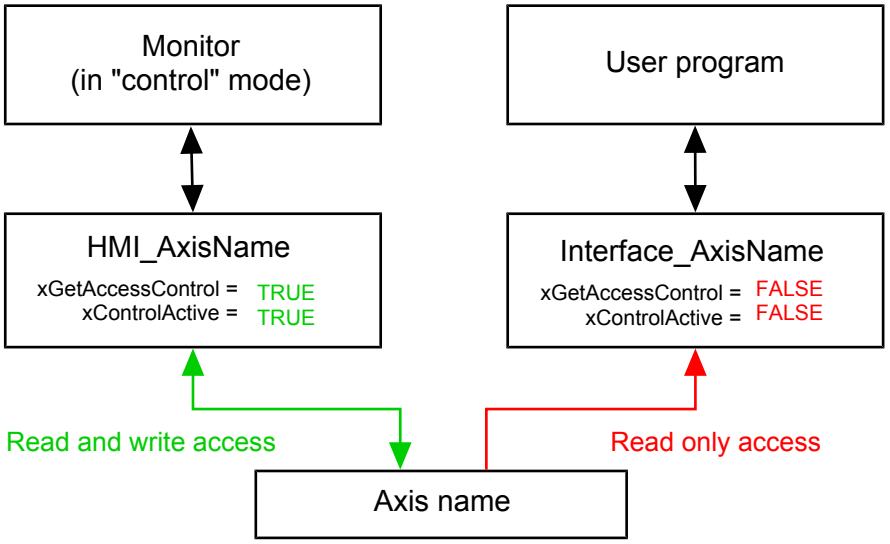
An instance requests access by setting *xGetAccessControl* to "TRUE." If *xControlActive* returns a "TRUE" value, access has been granted and is now permitted.



18014422491696011

Monitor access

If the axis function is accessed at the same time as the user program using a monitor in control mode, the *Interface_AxisName* user interface loses its access permission and *xControlActive* reports back "FALSE". Instead, access is realized using the *HMI_AxisName* interface to the monitor. In this case, *xControlActive* reports back "TRUE".



18014422491831691

29179459/EN – 03/2020

6.3.3 Basic axis functions (Basic)

The *Basic* structure contains the basic functions of the axis. This part of the user interface is identical for all axis types.

The following control and status variables are available:

Interface in the IEC Editor

9007219940779147

IN

Variable name	Description
xEnable_EmergencyStop	Data type: BOOL
	• TRUE – The software module does not execute an emergency stop using the set emergency stop ramp. (Requirement for enabling the axis) • FALSE – The software module executes an emergency stop using the set emergency stop ramp.
xEnable_ApplicationStop	Data type: BOOL
	• TRUE – The software module does not execute an application stop using the deceleration set to below the application limits. (Requirement for enabling the axis) • FALSE – The software module executes an application stop using the deceleration set under the application limits.

Comment:

To enable the axis, set *xEnable_EmergencyStop* and *xEnable_ApplicationStop* to "TRUE".

OUT

The dynamic parameters are scaled in user units.

Variable name	Description
lrActualPosition	Data type: LREAL – Floating-point number
	Actual position
lrActualVelocity	Data type: LREAL – Floating-point number
	Actual speed
xStandstill	Data type: BOOL
	• TRUE – Axis is at a standstill • FALSE – Axis is not at a standstill

6.3.4 Inverter functions (Inverter)

The *Inverter* structure contains control and status variables for the inverter. This part of the user interface represents a hardware device and, as a result, only exists with real axes.

The following control and status variables are available:

Interface in the IEC Editor

Inverter	SEW_IAx.ST_Inverter2	
In	ST_Inverter_In	
xInhibit	BOOL	FALSE
wDigitalOutputs	WORD	0
xSimulation	BOOL	FALSE
lrTorqueLimit	LREAL	32.767
Out	ST_Inverter_Out2	
xConnected	BOOL	TRUE
xPowered	BOOL	FALSE
xReady	BOOL	TRUE
xReferenced	BOOL	FALSE
xSetpointActive	BOOL	FALSE
xSafeStop	BOOL	FALSE
xPositionValid	BOOL	FALSE
wDigitalInputs	WORD	0
lrActualTorque	LREAL	0
eActualInverterMode	E_INVERTERMODE	EmergencyStop
usiErrorID	USINT	0
usiErrorSubID	USINT	0
xSimulation	BOOL	FALSE

18014419195524875

IN

Variable name	Description
xInhibit	Data type: BOOL
	• TRUE – Inhibit the output stage of the device. When this signal is activated, the brake might engage; otherwise, the axis coasts to a halt. • FALSE – Enable the output stage of the device.
wDigitalOutputs	Data type: WORD
	Control of the digital outputs of the device **Information:** Depending on the software module used, digital outputs are already reserved and cannot be controlled here.
xSimulation	Data type – BOOL
	• TRUE – Simulate the frequency inverter of the software module (e.g. when testing without hardware). • FALSE – Do not simulate the frequency inverter.
lrTorqueLimit	Data type: LREAL – Floating-point number
	Torque limit (1.0 = 100% M_N)

29179459/EN – 03/2020

OUT

Variable name	Description
xConnected	Data type – BOOL
	• TRUE – Communication connection with controller • FALSE – No communication connection with controller
xPowered	Data type – BOOL
	• TRUE – Output stages enabled (provide output voltage) • FALSE – Output stages not enabled
xReady	Data type – BOOL
	• TRUE – Ready for control by the controller • FALSE – Not ready for control by the controller
xReferenced	Data type: BOOL
	• TRUE – Referenced • FALSE – Not referenced
xSetpointActive	Data type – BOOL
	• TRUE – Setpoints are processed • FALSE – Setpoints are not processed
xSafeStop	Data type – BOOL
	• TRUE – Axis is at standstill (STO active) • FALSE – Axis is not at a standstill (STO is not active)
xPositionValid	Data type: BOOL
	• TRUE – Encoder position is valid • FALSE – Encoder fault is present (e.g. due to a bird strike)
wDigitalInputs	Data type: WORD
	State of the digital inputs **Information:** Depending on the software module used, digital inputs are already reserved and cannot be controlled here.
lrActualTorque	Data type: LREAL – Floating-point number
	Actual torque (1.0 = 100% M_N)

29179459/EN – 03/2020

Variable name	Description
eActualInverterMode	Data type: E_INVERTERMODE
	Operating mode of the inverter (FCB of the inverter): • Unknown • Default • OutputDisabled (FCB 01) • ManualMode (FCB 04) • Stop (FCB 02) • Homing (FCB 12) • JogMode (FCB 20) • BrakeTest (FCB 21) • Positioning (FCB 09) • PositioningInterpolated (FCB10) • Velocity (FCB 05) • VelocityInterpolated (FCB 06) • Torque (FCB 07) • TorqueInterpolated (FCB 08) • MotorParamMeasurement (FCB 25) • PosHoldCtrl (FCB 19) • RotorPosIdentification (FCB 18) • ApplicationStop (FCB 13) • EmergencyStop (FCB 14) • UserStop (FCB 26) *Library: SEW DeviceHandler Interfaces*
usiErrorID	Data type: USINT
	Error ID
usiErrorSubID	Data type: USINT
	Suberror ID
xSimulation	Data type – BOOL
	• TRUE – Simulation of the axis activated • FALSE – Simulation of the axis not activated

29179459/EN – 03/2020

6.3.5 Energy-saving function (EnergySaving)

The *EnergySaving* structure contains the control and status variables of the energy-saving function.

INFORMATION

The energy-saving function can only be activated when the axis is not enabled. When using a double axis, both axes must not be enabled. In this case, the control signal always sets both axes to energy-saving mode.

The following control and status variables are available:

Interface in the IEC Editor

9007219941252491

IN

Variable name	Description
xActivateStandBy	Data type – BOOL
	• TRUE – Activate energy-saving mode • FALSE – Deactivate energy-saving mode

OUT

Variable name	Description
xStandByActive	Data type – BOOL
	• TRUE – Energy-saving mode active • FALSE – Energy-saving mode not active

6.3.6 Brake control (Brake)

⚠ WARNING

Uncontrolled system behavior when the brake is released.

Severe or fatal injuries, e.g. due to falling hoist.

- Make sure that the system does not automatically start moving when the brake is released.
- Make sure that in hoist applications unintentional lowering of the load is prevented by suitable protective measures.

The *Brake* structure contains control variables and status variables for controlling the brake in special cases, e.g. if a drive is not supplied with current (is in controller inhibit) and is to be moved manually.

In the configuration of the axis in MOVISUITE®, set the parameter "Release brake with inhibited output stage –enable?" under "Drive functions" > "FCB 01 Output stage enable (FCB 01 Output stage inhibit)" to *Yes*. The brake can then be released by setting *xBrakeOpen* to the value "TRUE".

The following control and status variables are available:

Interface in the IEC Editor

Brake	SEW_MOS_IAxis.ST_Brake	
IN	ST_Brake_IN	
xBrakeOpen	BOOL	FALSE
OUT	ST_Brake_OUT	
xBrakeReleased	BOOL	FALSE

20686516235

IN

Variable name	Description
xBrakeOpen	Data type – BOOL
	• TRUE – Release the brake. If the output stage of the device is inhibited (e.g. via *xInhibit* = "TRUE"), the user can use this variable to accurately control the release (opening) of the brake. • FALSE – Apply the brake.

OUT

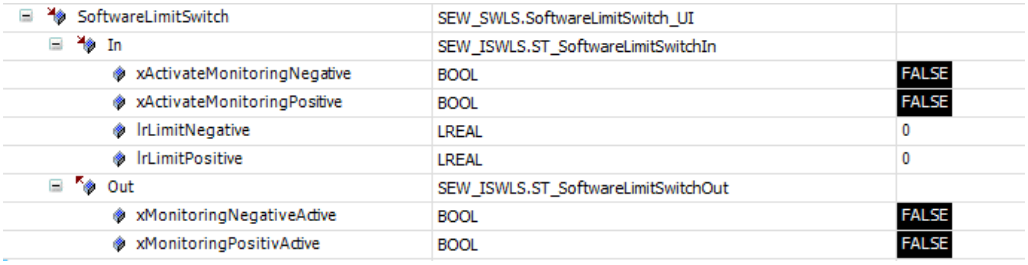
Variable name	Description
xBrakeReleased	Data type: BOOL
	• TRUE – The brake is released. • FALSE – The brake is applied.

29179459/EN – 03/2020

6.3.7 Software limit switches (SoftwareLimitSwitch)

The *SoftwareLimitSwitch* structure contains the control and status variables of the software limit switches.

Interface in the IEC Editor



9007223531035915

IN

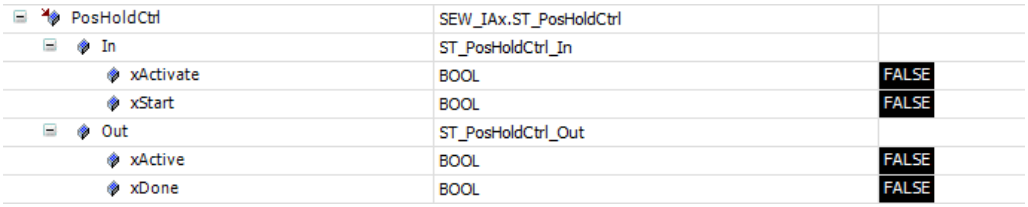
Variable name	Description
xActivateMonitoring-Negative	Data type – BOOL
	• TRUE – Activate monitoring of software limit switch in negative direction • FALSE – Deactivate monitoring of software limit switch in negative direction
xActivateMonitoring-Positive	Data type – BOOL
	• TRUE – Activate monitoring of software limit switch in positive direction • FALSE – Deactivate monitoring for a software limit switch in a positive direction.
lrLimitNegative	Data type: LREAL – Floating-point number
	Position of the software limit switch in negative direction (in user units)
lrLimitPositive	Data type: LREAL – Floating-point number
	Position of the software limit switch in positive direction (in user units)

OUT

Variable name	Description
xMonitoringNegat-iveActive	Data type – BOOL
	• TRUE – Monitoring for software limit switch in a negative direction is active. • FALSE – Monitoring of software limit switch in negative direction is not active
xMonitoringPositive-Active	Data type: BOOL
	• TRUE – Monitoring for software limit switch in a positive direction is active. • FALSE – Monitoring for software limit switch in a positive direction is not active.

6.3.8 Position hold control (PosHoldCtrl)

Interface in the IEC Editor



32294011019

IN

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate function • FALSE – Deactivate function
xStart	Data type – BOOL
	• TRUE – Start function • FALSE – Stop function

Note:

If *xActivate* and *xStart* have the value "TRUE", position hold control (FCB 19) is activated on the inverter. If one of these signals is removed, FCB 19 is deactivated.

Out

Variable name	Description
xActive	Data type – BOOL
	• TRUE – Activated • FALSE – Not activated
xDone	Data type – BOOL
	• TRUE – Process is complete • FALSE – Process is not complete

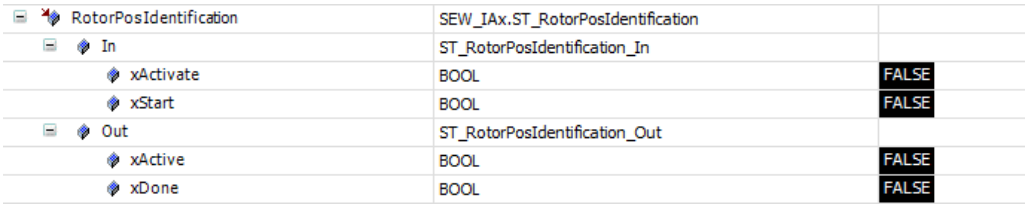
Note:

Both signals are "TRUE" as long as position hold control (FCB 19) is activated on the inverter. Otherwise, the signals output the value "FALSE".

29179459/EN – 03/2020

6.3.9 Rotor position identification (RotorPosIdentification)

Interface in the IEC Editor



32294128651

IN

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate function • FALSE – Deactivate function
xStart	Data type – BOOL
	• TRUE – Start function • FALSE – Stop function

Note:

Rotor position identification (FCB 18) is activated on the inverter if *xActivate* and *xStart* have the value "TRUE" If one of these signals is removed, FCB 18 is deactivated.

Out

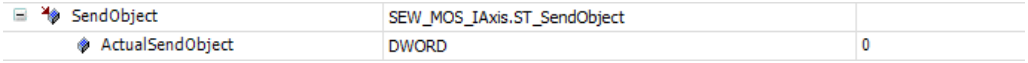
Variable name	Description
xActive	Data type – BOOL
	• TRUE – Activated • FALSE – Not activated
xDone	Data type – BOOL
	• TRUE – Process is complete • FALSE – Process is not complete

6.3.10 Process data transmission (SendObject)

The *SendObject* structure contains the data of the process data words PI 9 and PI 10 sent from the inverter to the MOVI-C® CONTROLLER as a double word. Typically, this is where the position of a second encoder is transferred. The scaling corresponds to the value connected to the corresponding process data.

The following control and status variables are available:

Interface in the IEC Editor



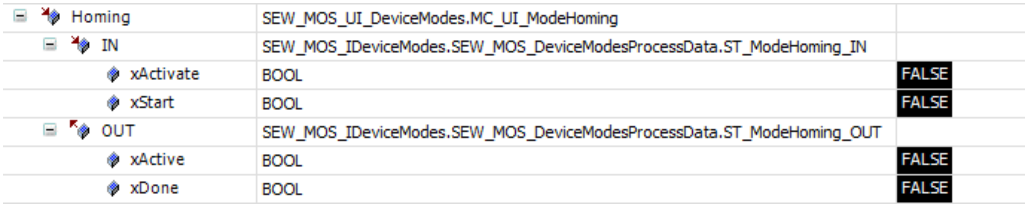
20686520971

Variable name	Description
dwActualSendObject	Data type: DWORD
	Data of process data words 9 and 10 (e.g. position of an encoder)

6.3.11 Reference travel (Homing)

The *Homing* structure contains control and status variables for the reference travels. It is used to control the FCB 12 of the connected inverter. The parameters for *FCB 12 Reference travel* set on the inverter apply (e.g. reference travel type, etc.).

Interface in the IEC Editor



20686538507

IN

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop If *xActivate* is set to "FALSE", the inverter switches to a stop FCB and stops with the ramp assigned to the stop FCB.
xStart	Data type – BOOL
	• TRUE – Start function • FALSE – Stop function

Note:

To start a reference travel, set both signals to "TRUE". To stop an ongoing reference travel, set one of the two signals to "FALSE".

29179459/EN – 03/2020

OUT

Variable name	Description
xActive	Data type – BOOL
	• TRUE – Activated • FALSE – Not activated
xDone	Data type – BOOL
	• TRUE – Process is complete • FALSE – Process is not complete

Note:

The feedback signal *xActive* becomes "TRUE" only when both input signals *xActivate* and *xStart* are set to "TRUE".

6.4 Touchprobe (TouchProbe)

The touchprobe function is used to detect the current position of the drive.

The following figure shows the interface in the IEC Editor:

Interface in the IEC Editor

TouchProbe	SEW_MOS_Touchprobe.MC_UI_TouchProbe	
IN	SEW_MOS_ITouchprobe.ST_TouchProbe_IN	
xActivate	BOOL	FALSE
xStart	BOOL	FALSE
lrUserProgramSourceSetpoint	LREAL	0
CONFIG	SEW_MOS_ITouchprobe.ST_TouchProbe_CONFIG	
eTouchProbeMode	E_TOUCHPROBEMODE	SINGLE
uiUserProgramSetpointDecimalPlaces	UINT	0
OUT	SEW_MOS_ITouchprobe.ST_TouchProbe_OUT	
xActive	BOOL	FALSE
xDone	BOOL	FALSE
lrValue	LREAL	0
uiCounter	UINT	0

20714210187

6.4.1 Configuration

To use the touchprobe function, the following steps are required:

- Configure the touchprobe function in MOVISUITE®

  For further information, refer to the chapter "Touchprobe 1" (→ 🗎 43).
- Initialize the touchprobe source in the *SEW_PRG.Init* action

Touchprobe source

The following options are available as touchprobe sources:

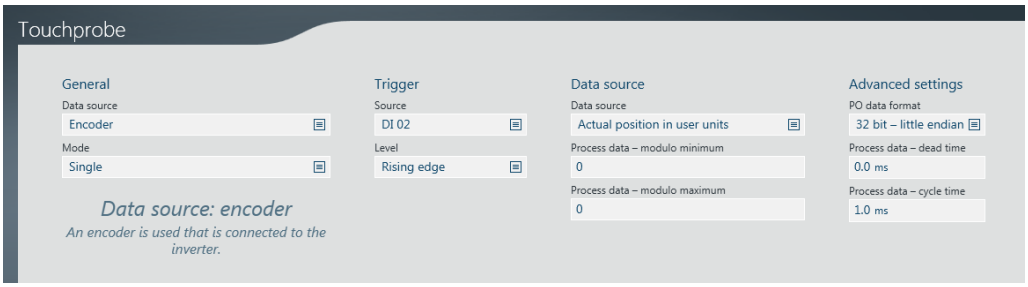
- Motor encoder or other encoder connected to the device
- Setpoint position of a configured axis
- Any signal generated by the user

Motor encoder

To configure a motor encoder as the touchprobe source, make the following settings in MOVISUITE®:

- Trigger signal source (in this example: *DI02*)
- Trigger signal level (in this example: *Rising edge*)
- Touchprobe source (in this example: *Actual position in user units – modulo*)

Interface in MOVISUITE®



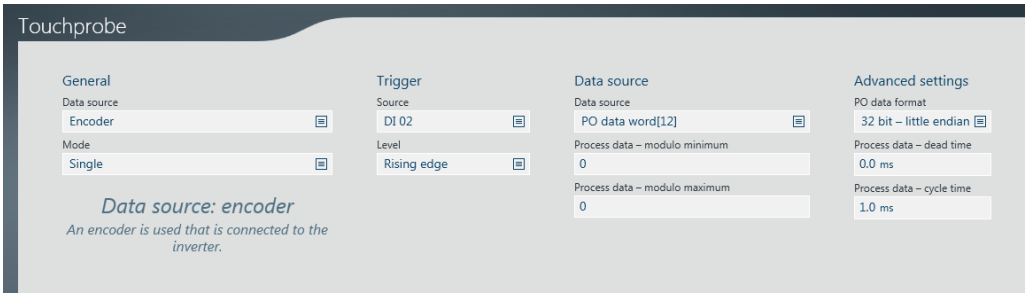
9007220002789771

Setpoint position of a configured axis

To configure a setpoint position of a configured axis as the touchprobe source, make the following settings in MOVISUITE®:

- Trigger signal source (in this example: *DI02*)
- Trigger signal level (in this example: *Rising edge*)
- Touchprobe source (in this example: *Process output data[12]*)
- PO data format
- Modulo minimum/modulo maximum
- Dead time/cycle time of the process data

Interface in MOVISUITE®

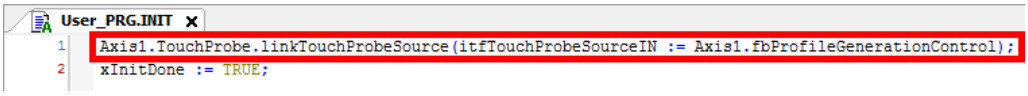


9007220004164363

Initialization

An assignment to the touchprobe source is required in the *User_PRG.INIT* action, e.g. the setpoint of *Axis1*:

Interface in the IEC Editor



20749428363

29179459/EN – 03/2020

6.4.2 IN

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop – If *xActivate* is set to "FALSE" and *lrStopDeceleration*" is ≤ 0, the operating mode stops at the last setpoint position, and speed and acceleration go to zero. – If *xActivate* is set to "FALSE" and *lrStopDeceleration*" is > 0, profile generation stops with the deceleration *lrStopDeceleration* and set set jerk *lrStopJerk*.
xStart	Data type – BOOL
	• TRUE – Start function • FALSE – Stop function
lrUserProgram SourceSetpoint	Data type: LREAL – Floating-point number
	A signal generated by the user in the IEC program that is to be used as a Touchprobe source ("User program" data source). The signal is cyclically transmitted to the inverter with the process data where it is used as the Touchprobe source.

6.4.3 Config

Variable name	Description
eTouchprobeMode	Data type: E_TOUCHPROBEMODE
	• SINGLE – Use the Touchprobe function once. • MULTIPLE – Use the Touchprobe function multiple times.
uiUserProgram SetpointDecimalPlaces	Data type: UINT
	The number of decimal places of the signal used as the Touchprobe source by the user ("User program" data source).

6.4.4 OUT

Variable name	Description
xActive	Data type – BOOL
	• TRUE – Activated • FALSE – Not activated
xDone	Data type – BOOL
	• TRUE – Process is complete • FALSE – Process is not complete

29179459/EN – 03/2020

Variable name	Description
lrValue	Data type: LREAL – Floating-point number
	Measured value
uiCounter	Data type: UINT
	Counter

29179459/EN – 03/2020

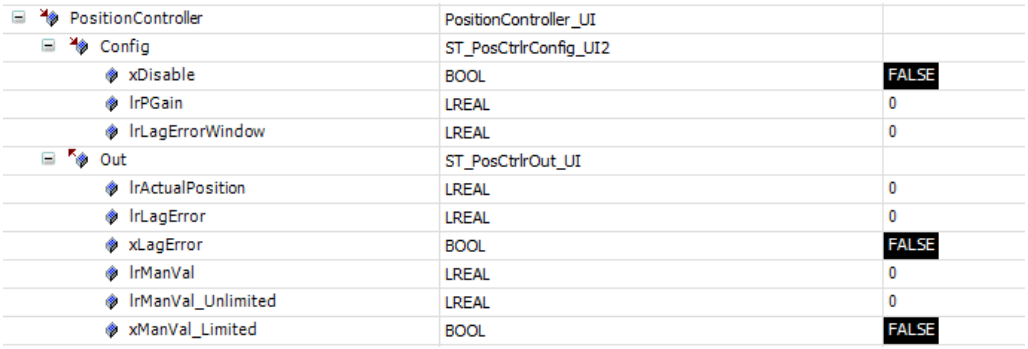
6.5 Position control (Controller)

INFORMATION

Only included if the function is activated in the "Basic settings" configuration menu under "Functions used".

6.5.1 Position controller (PositionController)

Interface in the IEC Editor



9007223528835595

Config

Variable name	Description
xDisable	Data type – BOOL
	• TRUE – Switch off position controller • FALSE – Turn on position controller.
lrPGain	Data type: LREAL – Floating-point number
	Position controller gain for minimizing lag errors Deactivated when the setting is "0".
lrLagErrorWindow	Data type: LREAL – Floating-point number
	Window of the actual lag error (in user units) Deactivated when set to "0"

OUT

Variable name	Description
lrActualPosition	Data type: LREAL – Floating-point number
	Actual position transferred to the position controller (in user units)
lrLagError	Data type: LREAL – Floating-point number
	Current lag error (in user units)
xLagError	Data type – BOOL
	• TRUE – Lag error present • FALSE – No lag error present

29179459/EN – 03/2020

Variable name	Description
lrManVal	Data type: LREAL – Floating-point number
	Correcting value of the position controller (in user units)
lrManVal_Unlimited	Data type: LREAL – Floating-point number
	Correcting value of the position controller without limitation (in user units)
xManVal_Limited	Data type – BOOL
	• TRUE – Position controller correction value limited • FALSE – Position controller correction value not limited

29179459/EN – 03/2020

6.5.2 Encoder evaluation (EncoderEvaluation)

Interface in the IEC Editor

EncoderEvaluation	EncoderEvaluation_UI	
In	ST_EncoderEvaluationIn_UI2	
lrActPos_LowResolutionEC_Encoder	LREAL	0
lrActPos_HighResolutionEC_Encoder	LREAL	0
xReferenced	BOOL	FALSE
xPositionValid	BOOL	TRUE
Config	ST_EncoderEvalConfig_UI2	
eActPos_EncSelector	E_ENCODEREVALUATION	MotorEncoder
xDontInitializeAtEncSelectorChange	BOOL	FALSE
lrInputFilterTime_ExtEnc	LREAL	0
lrDeadtime_ExtEnc	LREAL	0
xInterpolationFilterOn	BOOL	FALSE

9007223528831755

Config

INFORMATION

Bear in mind that the modulo reference could be lost when acknowledging an encoder fault.

Variable name	Description
eActPos_EncSelector	Data type: E_ENCODEREVALUATION
	Source for actual position: • MotorEncoder – Motor encoder • ExternalEncoder – External encoder • MotorAndExternalEncoder – Motor encoder + external encoder • MotorAndLowResolutionEC_Encoder – Motor encoder and low-resolution EtherCAT® encoder • HighResolutionEC_Encoder – High-resolution EtherCAT® encoder • LowResolutionEC_Encoder – Low-resolution EtherCAT® encoder • HighAndLowResolutionEC_Encoder – High-resolution and low-resolution EtherCAT® encoder
xDontInitializeAtEncSelectorChange	Data type – BOOL
	• TRUE – Initialize modulo and all filters when changing the encoder selection. • FALSE – Do not initialize modulo and all filters when changing the encoder selection (required for on-the-fly changes).
lrInputFilterTime_ExtEnc	Data type: LREAL – Floating-point number
	Integral time for encoder adjustment in [s] **Default value: 0.1**
lrDeadtime_ExtEnc	Data type: LREAL – Floating-point number
	Dead time of the external encoder in [s] **Default value: 0**

Variable name	Description
xInterpolationFilterOn	Data type – BOOL
	• TRUE – Filter external and low-resolution encoders • FALSE – Do not filter external and low-resolution encoders.

6.5.3 Actual values (ActualValues)

Interface in the IEC Editor

ActualValues	ST_ActualValuesAG_UI2	
FromSubAxis	ARRAY [1..gc_usiMaxIndexOfController] OF ST_ActualValues	
FromSubAxis[1]	ST_ActualValues_VelocityInterpolated	
xActive	BOOL	FALSE
lrPosition	LREAL	0
lrVelocity	LREAL	0
lrIntegralVelCtrlr	LREAL	0
lrAcceleration	LREAL	0
lrTorque	LREAL	0
lrAdvancedSendObject	LREAL	0
xReferenced	BOOL	FALSE
xSetpointActive	BOOL	FALSE
FromSubAxis[2]	ST_ActualValues_VelocityInterpolated	
FromSubAxis[3]	ST_ActualValues_VelocityInterpolated	
FromSubAxis[4]	ST_ActualValues_VelocityInterpolated	
ToSuperAxis	SEW_IDM.ST_ActualValues_VelocityInterpolated	
xActive	BOOL	FALSE
lrPosition	LREAL	0
lrVelocity	LREAL	0
lrIntegralVelCtrlr	LREAL	0
lrAcceleration	LREAL	0
lrTorque	LREAL	0
lrAdvancedSendObject	LREAL	0
xReferenced	BOOL	FALSE
xSetpointActive	BOOL	FALSE
ToPG	SEW_IDM.ST_ActualValues_PositioningInterpolated	
xActive	BOOL	FALSE
lrPosition	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrTorque	LREAL	0
xReferenced	BOOL	FALSE
xSetpointActive	BOOL	FALSE

9007223528827915

29179459/EN – 03/2020

FromSubAxis[1..4]

Actual values that are transmitted to the axis group by subordinate group members.

The dynamic parameters are scaled in user units.

Variable name	Description
xActive	Data type – BOOL
	• TRUE – Activated • FALSE – Not activated
lrPosition	Data type: LREAL – Floating-point number
	Position
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrIntegralVelCtrlr	Data type: LREAL – Floating-point number
	I component of the speed controller
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrTorque	Data type: LREAL – Floating-point number
	Torque in percent of the nominal motor torques (1.0 = 100% M_N)
lrAdvancedSendObject	Data type: LREAL – Floating-point number
	Actual position sent to the axis or axis group from an external source
xReferenced	Data type – BOOL
	• TRUE – All relevant encoders are referenced • FALSE – Not all the relevant encoders are referenced
xSetpointActive	Data type – BOOL
	• TRUE – Setpoints are processed • FALSE – Setpoints are not processed

29179459/EN – 03/2020

ToSuperAxis

Mean value of the actual values of all related axis group members transmitted to the higher-level axis group member via the *VelocityInterpolated* interface.

The BOOLEAN variables are linked by logical operator AND.

The dynamic parameters are scaled in user units.

Variable name	Description
xActive	Data type – BOOL
	• TRUE – Activated • FALSE – Not activated
lrPosition	Data type: LREAL – Floating-point number
	Position
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrIntegralVelCtrlr	Data type: LREAL – Floating-point number
	I component of the speed controller
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrTorque	Data type: LREAL – Floating-point number
	Torque in percent of the nominal motor torques (1.0 = 100% M_N)
lrAdvancedSendObject	Data type: LREAL – Floating-point number
	Actual position sent to the axis or axis group from an external source
xReferenced	Data type – BOOL
	• TRUE – All relevant encoders are referenced • FALSE – Not all the relevant encoders are referenced
xSetpointActive	Data type – BOOL
	• TRUE – Setpoints are processed • FALSE – Setpoints are not processed

29179459/EN – 03/2020

ToPG

Mean value of the actual values of all subordinate axis group members transmitted to the profile generator via the *PositioningInterpolated* interface.

The dynamic parameters are scaled in user units.

Variable name	Description
xActive	Data type – BOOL
	• TRUE – Activated • FALSE – Not activated
lrPosition	Data type: LREAL – Floating-point number
	Position
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrTorque	Data type: LREAL – Floating-point number
	Torque in percent of the nominal motor torques (1.0 = 100% M_N)
xReferenced	Data type – BOOL
	• TRUE – All relevant encoders are referenced • FALSE – Not all the relevant encoders are referenced
xSetpointActive	Data type – BOOL
	• TRUE – Setpoints are processed • FALSE – Setpoints are not processed

29179459/EN – 03/2020

6.5.4 Setpoints (SetpointValues)

Interface in the IEC Editor

SetpointValues	ST_SetpointValuesAG_UI	
FromPG	SEW_IDM.ST_SetpointValues_PositioningInterpolated	
xActivate	BOOL	FALSE
lrPosition	LREAL	0
lrVelocityPrecontrol	LREAL	0
lrAccelerationPrecontrol	LREAL	0
lrTorquePrecontrol	LREAL	0
lrInertiaScale	LREAL	0
lrTorqueLimit	LREAL	32.767
FromSuperAxis	SEW_IDM.ST_SetpointValues_VelocityInterpolated	
xActivate	BOOL	FALSE
lrPosition	LREAL	0
lrVelocity	LREAL	0
lrAccelerationPrecontrol	LREAL	0
lrTorquePrecontrol	LREAL	0
lrManValPosCtrlr	LREAL	0
lrInertiaScale	LREAL	0
lrTorqueLimit	LREAL	32.767
ToSubAxis	ARRAY [1..gc_usiMaxIndexOfController] OF SEW_IDM.ST_SetpointValues	
ToSubAxis[1]	SEW_IDM.ST_SetpointValues_VelocityInterpolated	
xActivate	BOOL	FALSE
lrPosition	LREAL	0
lrVelocity	LREAL	0
lrAccelerationPrecontrol	LREAL	0
lrTorquePrecontrol	LREAL	0
lrManValPosCtrlr	LREAL	0
lrInertiaScale	LREAL	0
lrTorqueLimit	LREAL	32.767
ToSubAxis[2]	SEW_IDM.ST_SetpointValues_VelocityInterpolated	
ToSubAxis[3]	SEW_IDM.ST_SetpointValues_VelocityInterpolated	
ToSubAxis[4]	SEW_IDM.ST_SetpointValues_VelocityInterpolated	

9007223531026315

FromPG

Setpoints transmitted by the profile generator to the controller via the *PositioningInterpolated* interface.

The dynamic parameters are scaled in user units.

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop – If *xActivate* is set to "FALSE" and *lrStopDeceleration*" is ≤ 0, the operating mode stops at the last setpoint position, and speed and acceleration go to zero. – If *xActivate* is set to "FALSE" and *lrStopDeceleration*" is > 0, profile generation stops with the deceleration *lrStopDeceleration* and set set jerk *lrStopJerk*.
lrPosition	Data type: LREAL – Floating-point number
	Position

Variable name	Description
lrVelocityPrecontrol	Data type: LREAL – Floating-point number
	Precontrol value of speed
lrAccelerationPrecontrol	Data type: LREAL – Floating-point number
	Precontrol value of acceleration
lrTorquePrecontrol	Data type: LREAL – Floating-point number
	Precontrol value of torque in [1/M_n]
lrIntertiaScale	Data type: LREAL – Floating-point number
	Inertia scaling in [kg*m^2]
lrTorqueLimit	Data type: LREAL – Floating-point number
	Torque limit (1.0 = 100% M_N)

FromSuperAxis

Setpoints transmitted by the higher-level axis group member via the *VelocityInterpolated* interface.

The dynamic parameters are scaled in user units.

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop – If *xActivate* is set to "FALSE" and *lrStopDeceleration*" is ≤ 0, the operating mode stops at the last setpoint position, and speed and acceleration go to zero. – If *xActivate* is set to "FALSE" and *lrStopDeceleration*" is > 0, profile generation stops with the deceleration *lrStopDeceleration* and set set jerk *lrStopJerk*.
lrPosition	Data type: LREAL – Floating-point number
	Position
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAccelerationPrecontrol	Data type: LREAL – Floating-point number
	Precontrol value of acceleration
lrTorquePrecontrol	Data type: LREAL – Floating-point number
	Precontrol value of torque in [1/M_n]
lrIntertiaScale	Data type: LREAL – Floating-point number
	Inertia scaling in [kg*m^2]
lrTorqueLimit	Data type: LREAL – Floating-point number
	Torque limit (1.0 = 100% M_N)

ToSubAxis[1..4]

Setpoints transmitted by the axis group to the subordinate axis group members via the *VelocityInterpolated* interface.

The dynamic parameters are scaled in user units.

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop – If *xActivate* is set to "FALSE" and *lrStopDeceleration*" is ≤ 0, the operating mode stops at the last setpoint position, and speed and acceleration go to zero. – If *xActivate* is set to "FALSE" and *lrStopDeceleration*" is > 0, profile generation stops with the deceleration *lrStopDeceleration* and set set jerk *lrStopJerk*.
lrPosition	Data type: LREAL – Floating-point number
	Position
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAccelerationPrecontrol	Data type: LREAL – Floating-point number
	Precontrol value of acceleration
lrTorquePrecontrol	Data type: LREAL – Floating-point number
	Precontrol value of torque in [$1/M_n$]
lrManVal	Data type: LREAL – Floating-point number
	Correcting value of the position controller
lrIntertiaScale	Data type: LREAL – Floating-point number
	Inertia scaling in [$kg*m^2$]
lrTorqueLimit	Data type: LREAL – Floating-point number
	Torque limit (1.0 = 100% M_N)

29179459/EN – 03/2020

6.5.5 Anti-sway control (AntiSway)

Input variables (In)

Variable name	Description
lrSensorSignal	Data type: LREAL – Floating-point number
	Position of the master from external source in user units

Configuration (Config)

Structure with input variables to control the function.

Variable name	Description
eApplicationType	Data type – E_APPLICATIONTYPE
	Application type of anti-sway control: • NoAntiSway • TowerSway • PendulumSway (in preparation) • BellySway (in preparation) • FluidSway (in preparation) • SpringSway (in preparation) **Default value: 1**
lrUserUnitToMeter	Data type – E_APPLICATIONTYPE
	Position relationship between the user unit from the drive train and one meter in [m/user units] If the value is set to "0", the user unit corresponds to the unit [m]. If the value is not "0", the formula user unit*lrUserUnit-ToMeter = meter applies. The computing time is longer in this case. **Default value: 0**

Setpoint correction (SetPointCorrection)

Structure with variables for configuring setpoint correction

Variable name	Description
eSelector	Data type – E_SETPOINTCORRECTIONSELECTOR
	• Off • AntiSway • Bandstop • TensionTimes

29179459/EN – 03/2020

Variable name	Description
xStart	Data type – BOOL
	• TRUE – Start positioning Must be maintained until *Out.xBusy* has the value "FALSE". • FALSE – Stop positioning Is used when *eSelector* is "Bandstop" and *eApplicationType* is "TowerSway", "SpringSway" or "BellySway". For these types of application, the bandstop has residual inaccuracies when approaching the target. These inaccuracies are eliminated when the profile is below *BandStop.lrAdjustWindow* and is finished when the residual accuracy is less than 0.0001. Is used when *eSelector* is set to "AntiSway" and the application type is "BellySway". The application type "BellySway" has residual inaccuracies when the target is approached. These inaccuracies are eliminated when the profile is stopped and the target deviation is less than 0.0001.
xAdjustEnable	Data type – BOOL
	• TRUE – Allow residual inaccuracy movement • FALSE – Do not allow residual inaccuracy movement

Delay (TensionTimes)

The original motion profile is delayed by the sum of *lrJerkTime* and *lrRampTime*. During these times, the mechanical tension is built up which corresponds to the desired acceleration.

Variable name	Description
lrJerkTime	Data type: LREAL – Floating-point number
	Jerk time for the mechanical tension build-up in [s] Maximum <= 2000*PLC CycleTime **Default value: 0.05**
lrRampTime	Data type: LREAL – Floating-point number
	Ramp time for the mechanical tension build-up in [s] Maximum <= 2000*PLC CycleTime **Default value: 0.3**

Bandstop (BandStop)

If *eSelector* is "Bandstop" and *eApplicationType* "TowerSway", "SpringSway" or "BellySway, there will be little residual inaccuracy when approaching the target. The inaccuracies are corrected if the profile has a deviation greater than 0.0001, the target position is only away from the target within a specified window, and the speed and acceleration in the standard profile are less than 10^{-6}. The dynamic values for sensor-based positioning are configured in the following parameters.

29179459/EN – 03/2020

The tension can be built up by configuring *lrMassPayload*. This tension is built up using the dynamic values configured in the following parameters. Activation starts with *xStart* and *xAdjustEnable*.

Variable name	Description
lrAdjustVelocity	Data type: LREAL – Floating-point number
	Speed in [m/s] **Default value: ApplicationLimitVelocity**
lrAdjustAccDec	Data type: LREAL – Floating-point number
	Acceleration in [m/s^2] **Default value: ApplicationLimitVelocity**
lrAdjustGain	Data type: LREAL – Floating-point number
	P gain **Default value: 40**
lrAdjustWindow	Data type: LREAL – Floating-point number
	Window for starting sensor-based positioning in [m] Sensor-based positioning is performed when the profile is only away from the target within the window defined here. The process is terminated as soon as the deviation is ≤ 0.0001 m, the speed of the preset profile is ≤ 10^{-6} m/s, and the acceleration of the preset profile is ≤ 10^{-6} m/s^2. **Default value: 0.001**

Configuration of the trolley (DriveTrain)

Variable name	Description
lrHeightTower	Data type: LREAL – Floating-point number
	Tower height in [m] **Default value: 1**
lrDistanceHoistToCar	Data type: LREAL – Floating-point number
	Distance from lifting axis to travel axis in [m] **Default value: 1**
lrSpringTowerToCar	Data type: LREAL – Floating-point number
	Spring constant between tower and trolley in [Nm/wheel] **Default value: 1**
lrMassCar	Data type: LREAL – Floating-point number
	Mass of the trolley without tower and lifting gear in [kg] **Default value: 1**
lrMassHoist	Data type: LREAL – Floating-point number
	Mass of the lifting gear without payload mass and without shuttle in [kg] **Default value: 1**

29179459/EN – 03/2020

Variable name	Description
lrMassTower	Data type: LREAL – Floating-point number
	Mass of the tower without trolley, lifting gear, load mass and shuttle in [kg] **Default value: 1**
lrMassPayload	Data type: LREAL – Floating-point number
	Mass of the payload; shuttle is counted as payload in [kg] **Default value: 1**

Status (Out)

Variable name	Description
xBusy	Data type: LREAL – Floating-point number
	For *eApplicationType* "TowerSway", "BellySway" or "SpringSway": • FALSE – Anti-sway control inactive • TRUE – Anti-sway control active xBusy detects all changes triggered by anti-sway control compared to the original travel profile. **Default value: 0**
lrMaschineState	Data type: LREAL – Floating-point number
	For *eApplicationType* "TowerSway" or "SpringSway" and *eSelector* "Bandstop" **or** for *eApplicationType* "BellySway": • 0 – Sensor-based positioning inactive • 1 – Sensor-based positioning active For *eApplicationType* "SpringSway and without *eSelector* "Bandstop": • 0 – SpringSway inactive • 1 – SpringSway active **Default value: 0**

AntiSway correcting values (ManipulatedValues)

Variable name	Description
lrPosition	Data type: LREAL – Floating-point number
	Position (in user units) **Default value: 0**
lrVelocity	Data type: LREAL – Floating-point number
	Speed (in user units) **Default value: 0**

29179459/EN – 03/2020

Variable name	Description
lrAccDec	Data type: LREAL – Floating-point number
	Acceleration (in user units) **Default value: 0**

AntiSway setpoints (ModifiedSetpoints)

Setpoints sent to the controller following the AntiSway function.

Variable name	Description
lrPosition	Data type: LREAL – Floating-point number
	Position (in user units) **Default value: 0**
lrVelocity	Data type: LREAL – Floating-point number
	Speed (in user units) **Default value: 0**
lrAccDec	Data type: LREAL – Floating-point number
	Acceleration (in user units) **Default value: 0**

6.5.6 Connecting a high-resolution/low-resolution EtherCAT® encoder

The following link commands are available for connecting a high-resolution and/or low-resolution EtherCAT® encoder:

```
Axis1.fbController.LinkILowResolutionEC_Encoder2(itfQueryInter-
faceSEW := _fbSyncExtSource);
```

```
Axis1.fbController.LinkIHighResolutionEC_Encoder2(itfQueryIn-
terfaceSEW := _fbSyncExtSource);
```

The MOVIKIT® EncoderInterface is used for connection. For more information, refer to the corresponding manual.

29179459/EN – 03/2020

6.6 MultiMotion axis functions (ProfileGeneration)

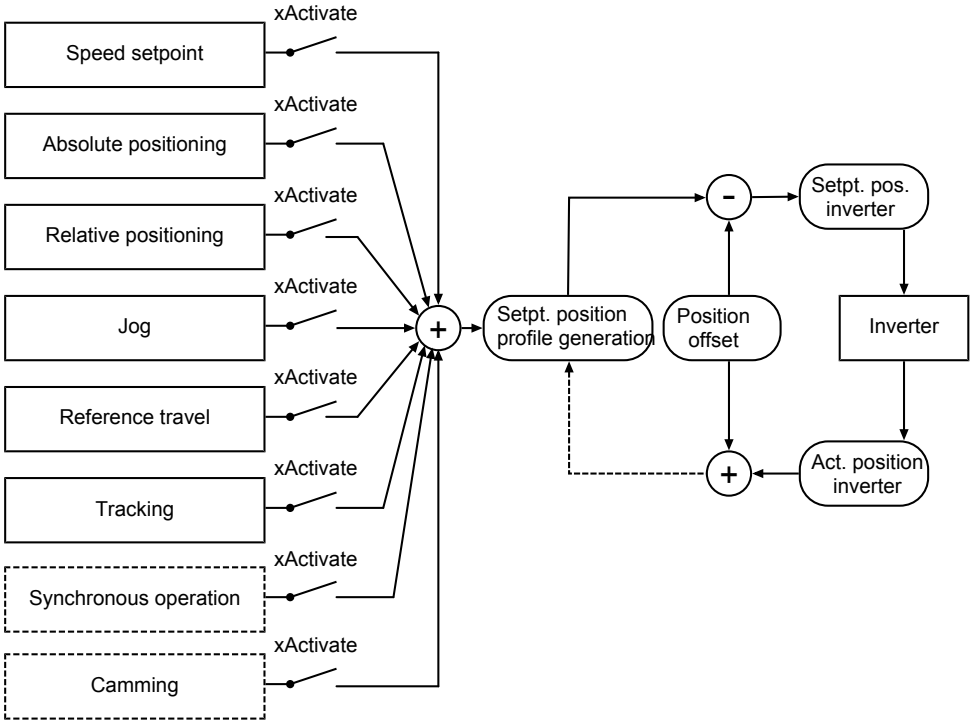
The function modules described below contain variables for configuring complex axis motion sequences.

The control and status variables of the various MultiMotion axis functions are available in the *ProfileGeneration* structure. These include special structures for the various operating modes (*Velocity*, *Positioning*, etc.) as well as structures with variables that are valid across all modes.

Profile generation procedure

The setpoints generated by profile generation are transmitted cyclically and synchronously to the inverter; the inverter then processes these setpoints into an interpolating operating mode.

A characteristic feature is that multiple operating modes are activated simultaneously and, as a result, run overlapped. The following drawing illustrates the internal structure of profile generation:



9007224555719051

If two or more operating modes are activated simultaneously, the corresponding motion profiles also overlap. This means that the speeds and accelerations of the operating modes involved in overlapping are added together. It is the responsibility of the user to use this option in such a way that no malfunctions occur.

Interface in the IEC Editor

ProfileGeneration	ProfileGeneration_MultiMotionStandardAxis_UI
In	SEW_IInterpolModes.ST_ProfileGeneration_In2
Config	SEW_IInterpolModes.ST_ProfileGeneration_Config2
Out	SEW_IInterpolModes.ST_ProfileGeneration_Out
Velocity	SEW_UIInterpolModes.ModeVelocity_UI
Positioning	SEW_UIInterpolModes.ModePositioning_UI
PositioningRelative	SEW_UIInterpolModes.ModePositioningRelative_UI
Jog	SEW_UIInterpolModes.ModeJog_UI
Tracking	SEW_UIInterpolModes.ModeTracking_UI
Homing	SEW_UIInterpolModes.ModeHoming_UI

31393605387

29179459/EN – 03/2020

6.6.1 Across all operating modes

Structures of the *ProfileGeneration* structure for all operating modes are described below.

The following control and status variables are available:

Interface in the IEC Editor

ProfileGeneration	ProfileGeneration_MultiMotionStandardAxis_UI	
In	SEW_IInterpolModes.ST_ProfileGeneration_In2	
xActivate	BOOL	TRUE
lrStopDeceleration	LREAL	0
lrStopJerk	LREAL	0
stInterpolationAdjust	ST_InterpolationAdjust	
xActivate	BOOL	FALSE
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrWindowNoAdjust	LREAL	0
xSetMotionPosition	BOOL	FALSE
lrMotionPositionSetpoint	LREAL	0
Config	SEW_IInterpolModes.ST_ProfileGeneration_Config2	
eErrorReaction	E_ERRORREACTIONINTERPOLATION	ABORT
eSetMotionPositionMode	E_SETMOTIONPOSITIONMODE	DEACTIVATED
Out	SEW_IInterpolModes.ST_ProfileGeneration_Out	
xProfileStandstill	BOOL	FALSE
xReferenced	BOOL	FALSE
lrActualMotionPositionUserUnit	LREAL	0
diActualMotionPositionPLCUnit	DINT	0
lrActualMotionOffsetUserUnit	LREAL	0
diActualMotionOffsetPLCUnit	DINT	0
stSetpointsUserUnits	ST_InterpolatedValues_UserUnits	
lrPosition	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
stSetpointsPLCUnits	ST_InterpolatedValues_PLCUnits	
diPosition	DINT	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0

18014419196028299

IN

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop – If *xActivate* is set to "FALSE" and *lrStopDeceleration*" is ≤ 0, the operating mode stops at the last setpoint position, and speed and acceleration go to zero. – If *xActivate* is set to "FALSE" and *lrStopDeceleration*" is > 0, profile generation stops with the deceleration *lrStopDeceleration* and set set jerk *lrStopJerk*.

Variable name	Description
lrStopDeceleration	Data type: LREAL – Floating-point number
	Deceleration with which profile generation stops if *xActivate*, *xEnable_EmergencyStop* or *xEnable_ApplicationStop* are set to "FALSE. If the value 0 is entered, the deceleration from the configuration data is used if *xEnable_EmergencyStop* or *xEnable_ApplicationStop* is set to "FALSE". If *xActivate* is set to "FALSE" in this case, the operating mode stops at the last setpoint position, and speed and acceleration go to zero.
lrStopJerk	Data type: LREAL – Floating-point number
	Jerk of profile generation if *xActivate*, *xEnable_EmergencyStop* or *xEnable_ApplicationStop* are set to "FALSE. If a value is specified, the required deceleration is established linearly according to the set value. If 0 is entered here, the change in deceleration occurs abruptly.
xSetMotionPosition	Data type – BOOL
	With a rising edge in profile generation, the setpoint position of the inverter (see chapter "MultiMotion axis functions (ProfileGeneration)" (→ 🗎 94)) is set according to the value of *eSetMotionPositionMode*. The setpoint position of profile generation is adjusted accordingly, whereby the position offset remains the same.
lrMotionPositionSetpoint	Data type: LREAL – Floating-point number
	Value to which the setpoint position of the inverter is set in the profile generation if the parameter *eSetMotionPositionMode* is set to "MOTIONPOSITIONSETPOINT".

stInterpolationAdjust

Adjustment function of profile generation

The dynamic parameters are scaled in user units.

Variable name	Description
xActivate	Data type: BOOL
	• TRUE – Activate If the inverter is enabled and profile generation is activated, *Interpolated position control (FCB 10)* will be switched on when *xActivate* is set to *TRUE*. • FALSE – Stop If *xActivate* is set to *FALSE*, the operating mode stops at the last setpoint position; speed and acceleration go to zero.
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration

29179459/EN – 03/2020

Variable name	Description
lrDeceleration	Data type: LREAL – Floating-point number
	Deceleration
lrWindowNoAdjust	Data type: LREAL – Floating-point number
	Position of the window within which no alignment is to be performed.

Config

Variable name	Description
eErrorReaction	Data type: E_ERRORREACTION
	Error response of profile generation: • ABORT Profile generation stops abruptly without a ramp at the current setpoint position. • APPLICATIONSTOP_DECELERATION Profile generation stops with the deceleration specified under application limits. • EMERGENCYSTOP_DECELERATION Profile generation stops with the emergency stop deceleration specified. • FOLLOWING_AXIS Profile generation follows the actual position of the axis.
eSetMotionPosition-Mode	Data type – E_SETMOTIONPOSITIONMODE
	• DEACTIVATED – It is not possible to set the setpoint position of the inverter • ACTUALPOSITION – The setpoint position of the inverter is set to the current position • MOTIONPOSITIONSETPOINT – The setpoint position of the inverter is set to the value set in *lrMotionPosition-Setpoint*

OUT

Variable name	Description
xProfileStandstill	Data type: BOOL
	• TRUE – Profile generation is in standstill. No change in setpoint occurs. • FALSE – Profile generation is not in standstill.
xReferenced	Data type: BOOL
	• TRUE – Referenced • FALSE – Not referenced
lrActualMotionPosition-UserUnit	Data type: LREAL – Floating-point number
	Profile generation setpoint position in user units

Variable name	Description
diActualMotionPosition-PLCUnit	Data type: DINT
	Profile generation setpoint position in PLC units
lrActualMotionOffset-UserUnit	Data type: LREAL – Floating-point number
	Position offset in user units
diActualMotionOffset-PLCUnit	Data type: DINT
	Position offset in PLC units

stSetpointsUserUnits

Setpoints generated by profile generation.

The dynamic parameters are scaled in user units.

Variable name	Description
lrPosition	Data type: LREAL – Floating-point number
	Position
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration

stSetpointsPLCUnits

Setpoints generated by profile generation.

The dynamic parameters are scaled in PLC units.

INFORMATION

PLC units can be defined as follows:

- Position (PLC unit) = position (user unit) x $10^{\text{Number of decimal places}}$
- Velocity (PLC unit) = position difference (PLC unit) / ms
- Acceleration (PLC unit) = speed difference (PLC unit) / ms / ms

Variable name	Description
diPosition	Data type: DINT
	Setpoint position
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration

29179459/EN – 03/2020

6.6.2 Speed control (Velocity)

In *Velocity* operating mode, profiles are generated where the position change is controlled with a specified speed (velocity).

The following control and status variables are available:

Interface in the IEC Editor

Velocity	SEW_UIInterpolModes.ModeVelocity_UI	
In	SEW_IInterpolModes.ST_Velocity_In	
xActivate	BOOL	FALSE
xStart	BOOL	FALSE
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0
Config	SEW_IInterpolModes.ST_Velocity_Config	
eModuloMode	E_MODULOMODE	MODULO_OFF
stPresetValues	ST_ProfGen_PresetValues	
lrPosition	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
ePresetMode	E_PRESETMODE	ACTPOS_MOVING
stStopAtPosition	ST_StopAtPosition	
eStopMode	E_STOPATPOSITIONMODE	STOPMODE_OFF
lrStopPosition	LREAL	0
Out	SEW_IInterpolModes.ST_Velocity_Out	
lrSetpPosition	LREAL	0
lrSetpPositionModulo	LREAL	0
lrDeltaPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xInVelocity	BOOL	FALSE
xModeActive	BOOL	FALSE
xProfileActive	BOOL	FALSE
xStopped	BOOL	FALSE

9007219942187915

IN

Changes to these variables are applied immediately.

The dynamic parameters are scaled in user units.

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop If *xActivate* is set to "FALSE", the operating mode stops at the last setpoint position, and speed and acceleration go to zero.
xStart	Data type – BOOL
	• TRUE – Start function • FALSE – Stop function

Variable name	Description
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity) When the specified target speed has been reached, any change to *lrVelocity* takes immediate effect. The specified acceleration *lrAcceleration* or deceleration *lrDeceleration* is used to reach the newly specified speed.
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrDeceleration	Data type: LREAL – Floating-point number
	Deceleration
lrJerk	Data type: LREAL – Floating-point number
	Jerk
xRapidStop	Data type: BOOL
	• TRUE – Carry out a rapid stop. • FALSE – Do not carry out a rapid stop.
lrRapidStopDeceleration	Data type: LREAL – Floating-point number
	Deceleration applied during rapid stop

Config

Variable name	Description
eModuloMode	Data type: E_MODULOMODE
	Modulo travel strategy: • MODULO_OFF Modulo absolute • MODULO_SHORT Modulo absolute – shortest distance • MODULO_CW Modulo absolute – positive direction • MODULO_CCW Modulo absolute – negative direction • MODULO_RELATIVE Modulo relative The parameter is adopted with a rising edge at *xStart* but will take effect with a falling edge at *xStart* when the "Stop at position" function (*eStopMode*) is performed.

29179459/EN – 03/2020

Variable name	Description
ePresetMode	Data type: E_PRESETMODE
	Initialization behavior of profile generation: • ACTPOS_STANDSTILL Profile generation starts from standstill (e.g. during overlap). • ACTPOS_MOVING Profile generation starts with current values (e.g. during on-the-fly changeover). • PRESET_DATA Profile generation starts with the specified preset values.

stPresetValues

Values for initializing the profile generation.

The dynamic parameters are scaled in user units.

Variable name	Description
lrPosition	Data type: LREAL – Floating-point number
	Position
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration

stStopAtPosition

Behavior when setting the profile generator.

The dynamic parameters are scaled in user units.

Variable name	Description
eStopMode	Data type: E_STOPATPOSITIONMODE
	Behavior with a falling edge at *xStart*: • STOPMODE_OFF Profile generation stops using the specified deceleration. • STOPMODE_RELATIVE Profile generation stops at the position defined by *lrStopPosition* based on the falling edge of *xStart*. • STOPMODE_ABSOLUTE Profile generation stops at the position defined by *lrStopPosition*.
lrStopPosition	Data type: LREAL – Floating-point number
	Profile generation stop position

OUT

The dynamic parameters are scaled in user units.

Variable name	Description
lrSetpPosition	Data type: LREAL – Floating-point number
	Setpoint position
lrSetpPositionModulo	Data type: LREAL – Floating-point number
	Modulo setpoint position generated by profile generation
lrDeltaPosition	Data type: LREAL – Floating-point number
	Position difference to the last PLC cycle in PLC units
lrSetpVelocity	Data type: LREAL – Floating-point number
	Target speed
lrSetpAcceleration	Data type: LREAL – Floating-point number
	Setpoint acceleration
xInVelocity	Data type – BOOL
	• TRUE – Axis has reached the specified speed • FALSE – Axis has not reached the specified speed
xModeActive	Data type – BOOL
	• TRUE – Operating mode activated • FALSE – Operating mode not activated
xProfileActive	Data type – BOOL
	• TRUE – Profile generation active • FALSE – Profile generation inactive
xStopped	Data type – BOOL
	• TRUE – Profile generation stopped • FALSE – Profile generation active

29179459/EN – 03/2020

6.6.3 Absolute positioning (Positioning)

In *Positioning* operating mode, the positioning profiles generated are based on an absolute position.

The following control and status variables are available:

Interface in the IEC Editor

Positioning	SEW_UIInterpolModes.ModePositioning_UI	
In	SEW_IInterpolModes.ST_Positioning_In	
xActivate	BOOL	FALSE
xStart	BOOL	FALSE
lrPosition	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0
Config	SEW_IInterpolModes.ST_Positioning_Config	
eModuloMode	E_MODULOMODE	MODULO_OFF
stPresetValues	ST_ProfGen_PresetValues	
lrPosition	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
ePresetMode	E_PRESETMODE	ACTPOS_MOVING
xTargetPositionMonitoring	BOOL	FALSE
xWithoutReferencedEncoder	BOOL	FALSE
Out	SEW_IInterpolModes.ST_Positioning_Out	
lrSetpPosition	LREAL	0
lrSetpPositionModulo	LREAL	0
lrDeltaPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xInPosition	BOOL	FALSE
xModeActive	BOOL	FALSE
xProfileActive	BOOL	FALSE
xStopped	BOOL	FALSE

9007219942192651

IN

Changes to these variables are applied immediately.

The dynamic parameters are scaled in user units.

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop If *xActivate* is set to "FALSE", the operating mode stops at the last setpoint position, and speed and acceleration go to zero.
xStart	Data type – BOOL
	• TRUE – Start function • FALSE – Stop function

Variable name	Description
lrPosition	Data type: LREAL – Floating-point number
	Position When using a modulo axis, the specified target position must meet the following condition: *ModuloMin* ≤ *lrPosition* ≤ *ModuloMax*.
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity) When the specified target speed has been reached, any change to *lrVelocity* takes immediate effect. The specified acceleration *lrAcceleration* or deceleration *lrDeceleration* is used to reach the newly specified speed.
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrDeceleration	Data type: LREAL – Floating-point number
	Deceleration
lrJerk	Data type: LREAL – Floating-point number
	Jerk
xRapidStop	Data type: BOOL
	• TRUE – Carry out a rapid stop. • FALSE – Do not carry out a rapid stop.
lrRapidStopDeceleration	Data type: LREAL – Floating-point number
	Deceleration applied during rapid stop

29179459/EN – 03/2020

Config

Variable name	Description
eModuloMode	Data type: E_MODULOMODE
	Modulo travel strategy: • MODULO_OFF Modulo absolute • MODULO_SHORT Modulo absolute – shortest distance • MODULO_CW Modulo absolute – positive direction • MODULO_CCW Modulo absolute – negative direction • MODULO_RELATIVE Modulo relative The parameter is adopted with a rising edge at *xStart*. With absolute positioning, activating *MODULO_RELATIVE* has the same effect as *MODULO_SHORT*.
ePresetMode	Data type: E_PRESETMODE
	Initialization behavior of profile generation: • ACTPOS_STANDSTILL Profile generation starts from standstill (e.g. during overlap). • ACTPOS_MOVING Profile generation starts with current values (e.g. during on-the-fly changeover). • PRESET_DATA Profile generation starts with the specified preset values.
xTargetPositionMonitoring	Data type: BOOL
	• TRUE – Before starting the movement, the profile generation checks if the target position is outside the travel range defined by the software limit switches. If so, the profile generation does not start. • FALSE – Profile generation does not check if the target position is outside the defined travel range.
xWithoutReferencedEncoder	Data type: BOOL
	• TRUE – Allow positioning with a non-referenced inverter. • FALSE – Only allow positioning with a referenced inverter.

stPresetValues

Values for initializing the profile generation.

The dynamic parameters are scaled in user units.

29179459/EN – 03/2020

Variable name	Description
lrPosition	Data type: LREAL – Floating-point number
	Position
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration

OUT

The dynamic parameters are scaled in user units.

Variable name	Description
lrSetpPosition	Data type: LREAL – Floating-point number
	Setpoint position
lrSetpPositionModulo	Data type: LREAL – Floating-point number
	Modulo setpoint position generated by profile generation
lrDeltaPosition	Data type: LREAL – Floating-point number
	Position difference to the last PLC cycle in PLC units
lrSetpVelocity	Data type: LREAL – Floating-point number
	Target speed
lrSetpAcceleration	Data type: LREAL – Floating-point number
	Setpoint acceleration
xInPosition	Data type: BOOL
	• TRUE – Specified position is reached • FALSE – Specified position is not reached
xModeActive	Data type – BOOL
	• TRUE – Operating mode activated • FALSE – Operating mode not activated
xProfileActive	Data type – BOOL
	• TRUE – Profile generation active • FALSE – Profile generation inactive
xStopped	Data type – BOOL
	• TRUE – Profile generation stopped • FALSE – Profile generation active

29179459/EN – 03/2020

6.6.4 Relative positioning (PositioningRelative)

In *PositioningRelative* operating mode, the positioning profiles generated are based on relative positions.

In addition, the *PositioningRelative* operating mode also offers configuration options for changing the travel distance during relative positioning.

The following control and status variables are available:

Interface in the IEC Editor

PositioningRelative	SEW_UIInterpolModes.ModePositioningRelative_UI	
In	SEW_IInterpolModes.ST_PositioningRelative_In	
xActivate	BOOL	FALSE
xStart	BOOL	FALSE
xAcceptNewDistance	BOOL	FALSE
lrDistance	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0
Config	SEW_IInterpolModes.ST_PositioningRelative_Config	
eModuloMode	E_MODULOMODE	MODULO_OFF
stPresetValues	ST_ProfGen_PresetValues	
lrPosition	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
ePresetMode	E_PRESETMODE	ACTPOS_STANDSTILL
eMode	E_POSRELATIVEMODE	DistanceChangeOFF
xTargetPositionMonitoring	BOOL	FALSE
xContinueRelativeMove	BOOL	FALSE
Out	SEW_IInterpolModes.ST_PositioningRelative_Out2	
lrSetpPosition	LREAL	0
lrSetpPositionModulo	LREAL	0
lrDeltaPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xInPosition	BOOL	FALSE
xModeActive	BOOL	FALSE
xProfileActive	BOOL	FALSE
xStopped	BOOL	FALSE
xNewDistanceAccepted	BOOL	FALSE

9007219942197387

IN

Changes to these variables are applied immediately.

The dynamic parameters are scaled in user units.

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop If *xActivate* is set to "FALSE", the operating mode stops at the last setpoint position, and speed and acceleration go to zero.
xStart	Data type – BOOL
	• TRUE – Start function • FALSE – Stop function
xAcceptNewDistance	Data type – BOOL
	If *eMode* is set to *TargetPositionBased*, a change to *lrDistance* is applied with a rising edge. The new target position then results by adding *lrDistance* to the original target position.
lrDistance	Data type: LREAL – Floating-point number
	Travel distance Unlike in *Positioning* operating mode, the specified travel distance *lrDistance* is not subject to a restriction specified by the parameters *ModuloMin* or *ModuloMax*.
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrDeceleration	Data type: LREAL – Floating-point number
	Deceleration
lrJerk	Data type: LREAL – Floating-point number
	Jerk
xRapidStop	Data type: BOOL
	• TRUE – Carry out a rapid stop. • FALSE – Do not carry out a rapid stop.
lrRapidStopDeceleration	Data type: LREAL – Floating-point number
	Deceleration applied during rapid stop

29179459/EN – 03/2020

Config

Variable name	Description
eModuloMode	Data type: E_MODULOMODE
	Modulo travel strategy: • MODULO_OFF Modulo absolute • MODULO_SHORT Modulo absolute – shortest distance • MODULO_CW Modulo absolute – positive direction • MODULO_CCW Modulo absolute – negative direction • MODULO_RELATIVE Modulo relative The parameter is adopted with a rising edge at *xStart*. With relative positioning, activating *MODULO_SHORT* has the same effect as *MODULO_RELATIVE*.
ePresetMode	Data type: E_PRESETMODE
	Initialization behavior of profile generation: • ACTPOS_STANDSTILL Profile generation starts from standstill (e.g. during overlap). • ACTPOS_MOVING Profile generation starts with current values (e.g. during on-the-fly changeover). • PRESET_DATA Profile generation starts with the specified preset values.

29179459/EN – 03/2020

Variable name	Description
eMode	Data type: E_POSRELATIVEMODE
	Behavior of the profile generation when changing the *lrDistance* variable: • DistanceChangeOFF Any change during ongoing movement is ignored. The new value is applied with the rising edge at *xStart*. • StartPositionBased A change during ongoing movement is possible and is applied immediately. The new target position is calculated based on the original start position. • ActualPositionBased A change during ongoing movement is possible and is applied immediately. The new target position is calculated based on the current position of the profile generation. • TargetPositionBased A change during ongoing movement is possible and is applied with a rising edge at *xAcceptNewDistance*. The new target position is calculated based on the original target position.
xTargetPositionMonitoring	Data type: BOOL
	• TRUE – Before starting the movement, the profile generation checks if the target position is outside the travel range defined by the software limit switches. If so, the profile generation does not start. • FALSE – Profile generation does not check if the target position is outside the defined travel range.
xContinueRelativeMove	Data type: BOOL
	• TRUE – Continue an interrupted relative positioning if the axis has meanwhile left the interpolating operating mode, e.g. due to enable not being granted. • FALSE – Do not continue an initiated relative positioning.

stPresetValues

Values for initializing the profile generation.

The dynamic parameters are scaled in user units.

Variable name	Description
lrPosition	Data type: LREAL – Floating-point number
	Position
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration

29179459/EN – 03/2020

OUT

The dynamic parameters are scaled in user units.

Variable name	Description
lrSetpPosition	Data type: LREAL – Floating-point number
	Setpoint position
lrSetpPositionModulo	Data type: LREAL – Floating-point number
	Modulo setpoint position generated by profile generation
lrDeltaPosition	Data type: LREAL – Floating-point number
	Position difference to the last PLC cycle in PLC units
lrSetpVelocity	Data type: LREAL – Floating-point number
	Target speed
lrSetpAcceleration	Data type: LREAL – Floating-point number
	Setpoint acceleration
xInPosition	Data type: BOOL
	• TRUE – Specified position is reached • FALSE – Specified position is not reached
xModeActive	Data type – BOOL
	• TRUE – Operating mode activated • FALSE – Operating mode not activated
xProfileActive	Data type – BOOL
	• TRUE – Profile generation active • FALSE – Profile generation inactive
xStopped	Data type – BOOL
	• TRUE – Profile generation stopped • FALSE – Profile generation active
xNewDistanceAccepted	Data type: BOOL
	• TRUE – New distance is accepted • FALSE – New distance is not accepted

29179459/EN – 03/2020

6.6.5 Jog

The *Jog* operating mode generates a "jogging" profile using the specified configuration.

The following control and status variables are available:

Interface in the IEC Editor

Jog	SEW_UIInterpolModes.ModeJog_UI	
In	SEW_IInterpolModes.ST_Jog_In	
xActivate	BOOL	FALSE
xJogCW	BOOL	FALSE
xJogCCW	BOOL	FALSE
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0
Config	SEW_IInterpolModes.ST_Jog_Config	
stPresetValues	ST_ProfGen_PresetValues	
lrPosition	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
ePresetMode	E_PRESETMODE	ACTPOS_STANDSTILL
Out	SEW_IInterpolModes.ST_Jog_Out	
lrSetpPosition	LREAL	0
lrSetpPositionModulo	LREAL	0
lrDeltaPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xModeActive	BOOL	FALSE
xProfileActive	BOOL	FALSE
xStopped	BOOL	FALSE

9007219942419723

IN

Changes to these variables are applied immediately.

The dynamic parameters are scaled in user units.

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop If *xActivate* is set to "FALSE", the operating mode stops at the last setpoint position, and speed and acceleration go to zero.
xJogPositive	Data type – BOOL
	• TRUE – Start profile generation of the operating mode for moving in positive direction using the value of the specified speed *lrVelocity* The specified acceleration takes effect. • FALSE – Stop profile generation of the operating mode. The specified deceleration takes effect.

Variable name	Description
xJogNegative	Data type – BOOL
	• TRUE – Start profile generation of the operating mode for moving in a negative direction using the value of specified speed *lrVelocity*. The specified acceleration is effective. • FALSE – Stop profile generation of the operating mode The specified deceleration takes effect.
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrDeceleration	Data type: LREAL – Floating-point number
	Deceleration
lrJerk	Data type: LREAL – Floating-point number
	Jerk
xRapidStop	Data type: BOOL
	• TRUE – Carry out a rapid stop. • FALSE – Do not carry out a rapid stop.
lrRapidStopDeceleration	Data type: LREAL – Floating-point number
	Deceleration applied during rapid stop

Config

Variable name	Description
ePresetMode	Data type: E_PRESETMODE
	Initialization behavior of profile generation: • ACTPOS_STANDSTILL Profile generation starts from standstill (e.g. during overlap). • ACTPOS_MOVING Profile generation starts with current values (e.g. during on-the-fly changeover). • PRESET_DATA Profile generation starts with the specified preset values.

stPresetValues

Values for initializing profile generation.

The dynamic parameters are scaled in user units.

Variable name	Description
lrPosition	Data type: LREAL – Floating-point number
	Position

29179459/EN – 03/2020

Variable name	Description
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration

OUT

The dynamic parameters are scaled in user units.

Variable name	Description
lrSetpPosition	Data type: LREAL – Floating-point number
	Setpoint position
lrSetpPositionModulo	Data type: LREAL – Floating-point number
	Modulo setpoint position generated by profile generation
lrDeltaPosition	Data type: LREAL – Floating-point number
	Position difference to the last PLC cycle in PLC units
lrSetpVelocity	Data type: LREAL – Floating-point number
	Target speed
lrSetpAcceleration	Data type: LREAL – Floating-point number
	Setpoint acceleration
xModeActive	Data type – BOOL
	• TRUE – Operating mode activated • FALSE – Operating mode not activated
xProfileActive	Data type – BOOL
	• TRUE – Profile generation active • FALSE – Profile generation inactive
xStopped	Data type – BOOL
	• TRUE – Profile generation stopped • FALSE – Profile generation active

29179459/EN – 03/2020

6.6.6 Direct coupling (Tracking)

In *Tracking* operating mode, master-based motion profiles are generated that follow a master source. This operating mode has the function of an open and, therefore, fully user-controllable input to the profile generation.

This operating mode performs the following tasks in particular:

- Activate the setpoint profile of another axis

  In this case, an axis in tracking mode follows the setpoint profile of another axis, which can then be used to implement a simple synchronous operation. For this purpose, set the *ConfiguredAxis* as the source when configuring *Tracking* operating mode and select the corresponding axis. In this case, the setpoint profile and configuration data are automatically tapped at the axis connected as master. The *In.MasterUserProgram* is not processed with this setting.

- Activate a free, user-programmed profile

  This allows users to generate user-specific motion profiles that cannot be implemented using conventional profile generators. For this purpose, set the source to *user program* when configuring *tracking* mode. In this case, the configuration data must be made available in the *MasterUserProgram.Config* structure and the setpoint profile must be provided to the *MasterUserProgram.In* structure. The setpoint profile must be provided in the *TaskHighPrio* task.

Interface in the IEC Editor

The following control and status variables are available:

In	SEW_IInterpolModes.ST_Tracking_In2	
xActivate	BOOL	FALSE
xStart	BOOL	FALSE
diTrackingNumerator	DINT	1
diTrackingDenominator	DINT	1
Out	SEW_IInterpolModes.ST_Tracking_Out	
xProfileActive	BOOL	FALSE
xModeActive	BOOL	FALSE
lrSetpPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
MasterUserProgram	SEW_SyncMoUtil.MasterUserProgram	
In	SEW_IMast.ST_MasterSetpointValues	
lrMasterPosition	LREAL	0
lrMasterVelocity	LREAL	0
lrMasterAcceleration	LREAL	0
Config	SEW_IMast.ST_MasterConfig	
lrMasterModuloMax	LREAL	0
lrMasterModuloMin	LREAL	0
uiMasterResolution	UINT	0
stTimeBaseFactor	ST_TimeBaseFactor	
eVelocity	E_VELOCITYFACTOR	MINUTE
eAcceleration	E_ACCELERATIONFACTOR	MINUTE_MINUTE

18014419197165451

IN

Changes to these variables are applied immediately.

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop If *xActivate* is set to "FALSE", the operating mode stops at the last setpoint position, and speed and acceleration go to zero.
xStart	Data type – BOOL
	• TRUE – Start function • FALSE – Stop function
diTrackingNumerator	Data type: DINT
	Numerator of the proportional factor (user units of the slave axis)
diTrackingDenominator	Data type: DINT
	Denominator of the proportional factor (user units of the master axis)

Note:

In *Tracking* operating mode, the profile generation follows the connected master signal proportionally. The proportional factor results from:

diTrackingNumerator [user units slave axis]
diTrackingDenominator [user units master axis]

The two variables are initialized with the values from the "Tracking" configuration menu in MOVISUITE®. As with previously described operating modes, changes made to these variables take effect immediately. If this occurs during ongoing movement, a jump in speed occurs.

The range of values for *diNumerator* and *diDenominator* is limited:

$$-\frac{2^{30}}{10^{n_{Slave}}} < diTrackingNumerator < \frac{2^{30}}{10^{n_{Slave}}}$$

$$0 < diTrackingDenominator < \frac{2^{30}}{10^{n_{Master}}}$$

Here, n_{Master} is the number of master decimal places to be processed and n_{Slave} is the number of slave decimal places to be processed.

29179459/EN – 03/2020

OUT

The dynamic parameters are scaled in user units of the slave axis.

Variable name	Description
xProfileActive	Data type – BOOL
	• TRUE – Profile generation active • FALSE – Profile generation inactive
xModeActive	Data type – BOOL
	• TRUE – Operating mode activated • FALSE – Operating mode not activated
lrSetpPosition	Data type: LREAL – Floating-point number
	Setpoint position
lrSetpVelocity	Data type: LREAL – Floating-point number
	Target speed
lrSetpAcceleration	Data type: LREAL – Floating-point number
	Setpoint acceleration

MasterUserProgram

The *MasterUserProgram* structure is only processed if *UserProgram* is set as the master source. When the operating mode is activated and started, the setpoint profile of the axis follows the profile provided for *MasterUserProgram.In.lrMasterPosition*. It is the responsibility of the user to limit speeds and accelerations in such a way that the axis is able to follow the setpoints. Major jumps in the setpoint profile may lead to a lag error in the axis or to the speed monitoring triggering/responding. If speed values and acceleration values of the setpoint profile are to be processed further (e.g. transmitted to the inverter as precontrol values together with the process data), the corresponding values of the master signal must also be connected at this point.

In

The dynamic parameters are scaled in user units of the master axis.

Variable name	Description
lrMasterPosition	Data type: LREAL – Floating-point number
	Position of the connected master signal
lrMasterVelocity	Data type: LREAL – Floating-point number
	Speed (velocity) of the connected master signal
lrMasterAcceleration	Data type: LREAL – Floating-point number
	Acceleration of the connected master signal

Config

The *CONFIG* structure is only processed if *UserProgram* has been set as the master source. The parameters are initialized with the values configured in *Tracking* operating mode.

Changes to these variables are applied to *xActivate* with a rising edge.

The dynamic parameters are scaled in user units.

Variable name	Description
lrMasterModuloMax	Data type: LREAL – Floating-point number
	Modulo maximum of the connected master signal
lrMasterModuloMin	Data type: LREAL – Floating-point number
	Modulo minimum of the connected master signal
uiMasterResolution	Data type: UINT
	Number of decimal places processed within the position value of the connected master signal

stTimeBaseFactor

Time base of the connected master signal

Variable name	Description
eVelocity	Data type: E_VELOCITYFACTOR
	Speed (velocity) of the connected master signal: • MINUTE (e.g. m/min) • SECOND (e.g. m/s)
eAcceleration	Data type: E_ACCELERATIONFACTOR
	Acceleration of the connected master signal: • MINUTE_MINUTE (e.g. m/min^2) • MINUTE_SECOND (e.g. m/min/s) • SECOND_SECOND (e.g. m/s^2)

29179459/EN – 03/2020

6.6.7 Reference travel (Homing)

Unlike "Reference travel (Homing)" (→ 🗎 74) as described in the chapter "Basic axis functions" (referencing of the inverter), the *Homing* operating mode in the *ProfileGeneration* structure allows for referencing the profile generation. The same reference travel types as for the inverter are basically available for referencing the profile generation. Only the types "ZeroPulseNegDir", "ReferencingWithoutRefTravelRT8" and "FixedStopPos" as well as "FixedStopNeg" are not supported. As this is an interpolating operating mode, this mode can be activated simultaneously with the other interpolating operating modes. Simultaneous execution of the operating modes is shown in the figure in chapter "MultiMotion axis functions (ProfileGeneration)" (→ 🗎 94).

Operating principle

When referencing the profile generation in the *Homing* operating mode, the position of profile generation is shifted by an offset. This offset is then deduced again from the position transferred as setpoint to the axis. The setpoint profile towards the axis runs continuously.

The currently effective offset is displayed via the variables *lrActualMotionOffsetUserUnit* and *diActualMotionOffsetPLCUnit* in the "OUT" (→ 🗎 97) structure under *ProfileGeneration* as well as in the *_fbProfileGenerationControl* structure under *AxisName*.

NOTICE

When using the "Homing" operating mode, the setpoint position of profile generation and the setpoint position/actual position of the inverter are offset from each other by a position offset. If absolute positioning of profile generation does not take into account that the axis is moved to a position offset by the position offset, collisions may occur in axes with limited travel range or mechanical intervention.

Death, severe injuries, damage to property

- Do not use this operating mode for axes with finite travel range or with mechanical intervention. This operating mode is not critical when used with endlessly turning axes without mechanical intervention, such as simple conveyor belts or rotary tables.

29179459/EN – 03/2020

Interface in the IEC Editor

Homing	SEW_UIInterpolModes.ModeHoming_UI	
In	SEW_IInterpolModes.ST_Homing_In	
xActivate	BOOL	FALSE
xStart	BOOL	FALSE
xLimitSwitchNegative	BOOL	FALSE
xLimitSwitchPositive	BOOL	FALSE
xReferenceCam	BOOL	FALSE
Config	SEW_IInterpolModes.ST_Homing_Config	
eReferenceTravelType	E_REFSTRATEGY	Deactivated
lrReferenceOffset	LREAL	0
lrLimitSwitchDebouncingTime	LREAL	0
lrSearchVelocity	LREAL	0
lrClearVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0
xMoveToStartPosition	BOOL	FALSE
lrStartPosition	LREAL	0
lrStartPosVelocity	LREAL	0
Out	SEW_IInterpolModes.ST_Homing_Out	
lrSetpPosition	LREAL	0
lrDeltaPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xDone	BOOL	FALSE
xInPosition	BOOL	FALSE
xModeActive	BOOL	FALSE
xProfileActive	BOOL	FALSE
xStopped	BOOL	FALSE

18014419197182987

IN

Changes to these variables are applied immediately.

The dynamic parameters are scaled in user units.

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop If *xActivate* is set to "FALSE", the operating mode stops at the last setpoint position, and speed and acceleration go to zero.
xStart	Data type – BOOL
	• TRUE – Start function • FALSE – Stop function
xLimitSwitchNegative	Data type – BOOL
	Input for the negative hardware limit switch. For reference travel types that evaluate this input, the user will need to connect the input in an appropriate manner. • TRUE – Negative hardware limit switch is approached • FALSE – Negative hardware limit switch is not approached

29179459/EN – 03/2020

Variable name	Description
xLimitSwitchPositive	Data type – BOOL
	Input for the positive hardware limit switch. For reference travel types that evaluate this input, the user will need to connect the input in an appropriate manner. • TRUE – Positive hardware limit switch is approached • FALSE – Positive hardware limit switch is not approached
xReferenceCam	Data type – BOOL
	Input of the reference cam For reference travel types that evaluate this input, the user will need to connect the input in an appropriate manner. • TRUE – Approach limit switch • FALSE – Do not approach limit switch

29179459/EN – 03/2020

Config

The dynamic parameters are scaled in user units.

Variable name	Description
eReferenceTravelType	Data type: E_REFSTRATEGY
	Reference travel types: • **Deactivated** Reference travel deactivated • ZeroPulseNegDir (not supported) Zero pulse – negative direction • RefCamNegEnd Reference cam – negative end • RefCamPosEnd Reference cam – positive end • LimitSwitchPos Positive limit switch • LimitSwitchNeg Negative limit switch • ReferencingWithoutRefTravel Referencing without reference travel • RefCamFlushLimitSwitchPos Reference cam flush – limit switch positive • RefCamFlushLimitSwitchNeg Reference cam flush – limit switch negative • ReferencingWithoutRefTravelRT8 (not supported) Referencing without reference travel • FixedStopPos (not supported) Fixed stop positive • FixedStopNeg (not supported) Fixed stop negative
lrReferenceOffset	Data type: LREAL – Floating-point number
	Reference offset
lrLimitSwitch DebouncingTime	Data type: LREAL – Floating-point number
	Debouncing time for the limit switches in [s] **Default value: 0.2**
lrSearchVelocity	Data type: LREAL – Floating-point number
	Search speed for reference travel
lrClearVelocity	Data type: LREAL – Floating-point number
	Retraction speed for reference travel
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration

29179459/EN – 03/2020

Variable name	Description
lrDeceleration	Data type: LREAL – Floating-point number
	Deceleration
lrJerk	Data type: LREAL – Floating-point number
	Jerk
xMoveToStartPosition	Data type – BOOL
	• TRUE – Perform homing immediately after referencing • FALSE – Do not perform homing immediately after referencing
lrStartPosition	Data type: LREAL – Floating-point number
	Target position of homing
lrStartPosVelocity	Data type: LREAL – Floating-point number
	Homing speed

Out

The dynamic parameters are scaled in user units.

Variable name	Description
lrSetpPosition	Data type: LREAL – Floating-point number
	Setpoint position
lrDeltaPosition	Data type: LREAL – Floating-point number
	Position difference to the last PLC cycle in PLC units
lrSetpVelocity	Data type: LREAL – Floating-point number
	Target speed
lrSetpAcceleration	Data type: LREAL – Floating-point number
	Setpoint acceleration
xDone	Data type – BOOL
	• TRUE – Process is complete • FALSE – Process is not complete
xInPosition	Data type – BOOL
	• TRUE – Specified position reached The *xInPosition* variable can only be "TRUE" if homing has been carried out. • FALSE – Specified position not reached
xModeActive	Data type – BOOL
	• TRUE – Operating mode activated • FALSE – Operating mode not activated
xProfileActive	Data type – BOOL
	• TRUE – Profile generation active • FALSE – Profile generation inactive

29179459/EN – 03/2020

Variable name	Description
xStopped	Data type – BOOL
	• TRUE – Profile generation stopped • FALSE – Profile generation active

29179459/EN – 03/2020

6.6.8 Synchronous operation (Gearing)

INFORMATION

Only included when using MOVIKIT® MultiMotion Gearing.

In the "Gearing" operating mode, a master-based travel profile is generated in which the slave axis follows the master axis via adjustable numerator/denominator factors.

Interface in the IEC Editor

Gearing	SEW_Gear.ModeGearing_UI	
In	ST_Gearing_IN	
xActivate	BOOL	FALSE
xStart	BOOL	FALSE
diGearNumerator	DINT	1
diGearDenominator	DINT	1
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
Config	SEW_SyncMoUtil.ST_Gearing_CONFIG	
Start	SEW_ICam.ST_Start_Config	
lrGearInDistance	LREAL	0
lrGearInOffset	LREAL	0
lrGearInReferencePosition	LREAL	0
eStartMode	E_CAMMINGSTARTMODE	CW_DIRECT
eTransitionType	E_TRANSITIONTYPE	MASTERBASED
lrGearInTime	LREAL	0
Stop	SEW_ICam.ST_Stop_Config	
lrGearOutDistance	LREAL	0
lrGearOutStopPosition	LREAL	0
eStopMode	E_CAMMINGSTOPMODE	CW_DIRECT
eTransitionType	E_TRANSITIONTYPE	MASTERBASED
lrGearOutTime	LREAL	0
Out	ST_Gearing_OUT	
xModeActive	BOOL	FALSE
xProfileActive	BOOL	FALSE
eGearingState	E_GEARING_STATE	STOPPED
lrSetpPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xSetSlaveValueDone	BOOL	FALSE
XOffsetCorrection	SEW_SyncMoUtil.OffsetCorrection_UI	
YOffsetCorrection	SEW_SyncMoUtil.OffsetCorrection_UI	
Adjust	SEW_SyncMoUtil.AdjustProfilgenerator_UI	
MasterUserProgram	SEW_SyncMoUtil.MasterUserProgram	

31486936715

IN

The dynamic parameters are scaled in user units.

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop If *xActivate* is set to "FALSE", the operating mode stops at the last setpoint position, and speed and acceleration go to zero.
xStart	Data type – BOOL
	• TRUE – Start function • FALSE – Stop function
diGearNumerator	Data type - DINT
	Numerator of the proportional factor (user units of the slave axis)
diGearDenominator	Data type - DINT
	Denominator of the proportional factor (user units of the master axis)
xSetSlaveValue	Data type – BOOL
	This variable is only evaluated when the operating mode is disabled (*xActivate* must have the value "FALSE"). The internal slave position is set to the value of *lrSlave-Value* at a rising edge.
lrSlaveValue	Data type: LREAL – Floating-point number
	Internal slave position Is transferred from *xSetSlaveValue* with a rising edge.
xReset	Data type – BOOL
	This variable is only evaluated when the operating mode is disabled (*xActivate* must have the value "FALSE"). The state of the operating mode is reset at a rising edge. This sets *OUT.eGearingState* to *STOPPED*. This may be necessary, for example, if the master position stops during desynchronization.

29179459/EN – 03/2020

Config

The dynamic parameters are scaled in user units.

Start

Variable name	Description
lrGearInDistance	Data type: LREAL – Floating-point number
	Synchronization distance (in user units of the master axis) *lrGearInDistance = 0* means that the slave synchronizes without a transition function. Depending on the curve profile, the setpoint profile may be inconsistent. With *lrGearInDistance > 0*, a transition function is created automatically when synchronization starts. The parameter only takes effect if *eTransitionType* has the value "TIMEBASED". An automatic transition function is calculated so that the slave axis is synchronous after the synchronizing time.
lrGearInOffset	Data type: LREAL – Floating-point number
	Offset to the reference position *lrGearInReferencePosition* (master)
lrGearIn ReferencePosition	Data type: LREAL – Floating-point number
	Reference position (in user units of the master axis). This reference position takes effect only with the following settings for *eStartMode*: • *WITH_CW_REFERENCE_POSITION* • *WITH_CCW_REFERENCE_POSITION* • *WITH_CW_CCW_REFERENCE_POSITION*

29179459/EN – 03/2020

Variable name	Description
eStartMode	Data type: E_CAMMINGSTARTMODE
	Behavior when setting *xStart*: • CW_DIRECT Immediate synchronization with a rising edge of *xStart* with a positive direction of movement of the master • CCW_DIRECT Immediate synchronization with a rising edge of *xStart* with a negative direction of movement of the master • CW_CCW_DIRECT Immediate synchronization with a rising edge of *xStart* with a positive or negative direction of movement of the master • WITH_CW_REFERENCE_POSITION Synchronization based on the reference position defined in *lrGearInReferencePosition* with a positive direction of movement of the master • WITH_CCW_REFERENCE_POSITION Synchronization based on the reference position defined in *lrGearInReferencePosition* with a negative direction of movement of the master • WITH_CW_CCW_REFERENCE_POSITION Synchronization based on the reference position defined in *lrGearInReferencePosition* with a positive or negative direction of movement of the master • CONTINUE_SYNCHRONIZED Synchronizing referred to the last known synchronous position of master and slave axis.
eTransitionType	Data type – E_TRANSITIONTYPE
	Type of an automatically calculated transition function when synchronizing: • NONE No transition function is calculated when synchronizing. This setting can lead to a discontinuous setpoint profile (e.g. speed jump). • MASTER-BASED A master-based transition function is calculated for synchronizing. The duration of the transition function is defined by the *lrGearInDistance* parameter. If *lrGearInDistance* has the value 0, the behavior is the same as for the setting "NONE. • TIME-BASED A time-based transition function is calculated for synchronizing. The duration of the transition function is defined by the *lrGearInTime* parameter. • (PROFILE-BASED) Reserved for future functions

29179459/EN – 03/2020

Variable name	Description
lrGearInTime	Data type: LREAL – Floating-point number
	Synchronizing time [s] for time-based synchronization The parameter only takes effect if *eTransitionType* has the value "TIMEBASED". An automatic transition function is calculated so that the slave axis is synchronous after the synchronizing time.

Stop

Variable name	Description
lrGearOutDistance	Data type: LREAL – Floating-point number
	Desynchronization distance (in user units of the master axis) *lrGearOutDistance = 0* means that the slave synchronizes without a transition function. Depending on the curve profile, the setpoint profile may be inconsistent. With *lrGearOutDistance > 0*, a transition function is created automatically when desynchronization starts. The parameter only takes effect if *eTransitionType* has the value "TIMEBASED". An automatic transition function is calculated so that the slave axis is synchronous after the synchronizing time.
lrGearOut StopPosition	Data type: LREAL – Floating-point number
	Stop position (in user units of the master axis). This stop position takes effect only with the following settings for *eStopMode*: • WITH_CW_INTERNAL_POSITION • WITH_CCW_INTERNAL_POSITION • WITH_CW_CCW_INTERNAL_POSITION • WITH_CW_ EXTERNAL_POSITION • WITH_CCW_ EXTERNAL_POSITION • WITH_CW_CCW_EXTERNAL_POSITION

29179459/EN – 03/2020

Variable name	Description
eStopMode	Data type: E_CAMMINGSTOPMODE
	Behavior when resetting *xStart*: • CW_DIRECT Immediate desynchronization with a rising edge of *xStart* with a positive direction of movement of the master • CCW_DIRECT Immediate desynchronization with a rising edge of *xStart* with a negative direction of movement of the master • CW_CCW_DIRECT Immediate desynchronization with a rising edge of *xStart* with a positive or negative direction of movement of the master • WITH_CW_INTERNAL_POSITION WITH_CW_EXTERNAL_POSITION Desynchronization to the stop position of the external master defined by *lrGearOutStopPosition* with positive direction of movement • WITH_CCW_INTERNAL_POSITION Desynchronization to the stop position of the external master defined by *lrGearOutStopPosition* with negative direction of movement • WITH_CW_CCW_INTERNAL_POSITION Desynchronization to the stop position of the external master defined by *lrGearOutStopPosition* with positive or negative direction of movement • WITH_CCW_EXTERNAL_POSITION Desynchronization to the stop position of the external master defined by *lrGearOutStopPosition* with a negative direction of movement • WITH_CW_CCW_EXTERNAL_POSITION Desynchronization to the stop position of the external master defined by *lrGearOutStopPosition* with a positive or negative direction of movement

29179459/EN – 03/2020

Variable name	Description
eTransitionType	Data type – E_TRANSITIONTYPE
	Type of an automatically calculated transition function when desynchronizing: • NONE No transition function is calculated for desynchronizing. This setting can lead to a discontinuous setpoint profile (e.g. speed jump). • MASTER-BASED A master-based transition function is calculated for desynchronizing. The duration of the transition function is defined by the *lrGearOutDistance* parameter. If *lrGearOutDistance* has the value 0, the behavior is the same as for the setting "NONE". • TIME-BASED A time-based transition function is calculated for desynchronizing. The duration of the transition function is defined by the *lrGearOutTime* parameter. • (PROFILE-BASED) Reserved for future functions
lrGearOutTime	Data type: LREAL – Floating-point number
	Desynchronizing time [s] for time-based synchronization The parameter only takes effect if *eTransitionType* has the value "TIMEBASED". An automatic transition function is calculated in such a way that the slave axis is no longer synchronous after the desynchronizing time.

Out

The dynamic parameters are scaled in user units of the slave axis.

Variable name	Description
xModeActive	Data type – BOOL
	• TRUE – Operating mode activated • FALSE – Operating mode not activated
xProfileActive	Data type – BOOL
	• TRUE – Profile generation active • FALSE – Profile generation inactive

29179459/EN – 03/2020

Variable name	Description
eGearingState	Data type: E_GEARING_STATE
	State of the axis: • STOPPED The slave axis is desynchronized and does not follow the master axis. • WAITING_FOR_STARTPOSITION Synchronization requested. The slave axis is waiting for a suitable master position. • GEAR_IN Synchronization is active. To synchronize with the master, the slave axis follows a transition function. • ACTIVE The slave axis is synchronized and follows the master axis. • WAITING_FOR_STOPPOSITION Desynchronization requested. The slave axis is waiting for a suitable master position. • GEAR_OUT Desynchronization is active. To reach the stop position, the slave axis follows a transition function.
lrSetpPosition	Data type: LREAL – Floating-point number
	Setpoint position
lrSetpVelocity	Data type: LREAL – Floating-point number
	Target speed
lrSetpAcceleration	Data type: LREAL – Floating-point number
	Setpoint acceleration
xSetSlaveValue-Done	Data type: BOOL
	• TRUE – The internal slave position has been set. • FALSE – The internal slave position has not been set.

29179459/EN – 03/2020

XOffsetCorrection

Using the *XOffsetCorrection* function, you can correct the phase position of the slave profile on the fly. The correction is carried out by superimposing a profile-based or master-based travel profile. With profile-based correction, the dynamic values are scaled in the user units of the master axis.

The following control and status variables are available:

Interface in the IEC Editor

XOffsetCorrection	OffsetCorrection_UI	
IN	SEW_ICam.ST_OffsetCorrection_IN	
xStart	BOOL	FALSE
lrCorrection	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0
OUT	SEW_ICam.ST_OffsetCorrectionUI_OUT	
xModeActive	BOOL	FALSE
xProfileActive	BOOL	FALSE
lrSetpPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xInPosition	BOOL	FALSE
xFBError	BOOL	FALSE
udiFBErrorID	UDINT	0

9007219942504331

IN

The dynamic parameters are scaled in user units of the master axis.

Variable name	Description
xStart	Data type – BOOL
	• TRUE – Start Set *xStart* to "TRUE" to create a travel profile that offsets the internal master to the external master by *lrCorrection*. The reference point remains constant, which means the original state is restored using *lrCorrection* equals zero (= 0). Set *eTransitionType* to "MASTERBASED" or "PROFILEBASED" for this purpose. • FALSE – Stop
lrCorrection	Data type: LREAL – Floating-point number
	Correction value
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrDeceleration	Data type: LREAL – Floating-point number
	Deceleration
lrJerk	Data type: LREAL – Floating-point number
	Jerk

Variable name	Description
xSetOffsetValue	Data type – BOOL
	With a rising edge, the correction value is set to the value defined in *lrOffsetValue*.
lrOffsetValue	Data type: LREAL – Floating-point number
	Correction value that is set with a rising edge at *xSetOffset-Value*.

Config

Variable name	Description
eOffsetCorrectionMode	Data type – E_OFFSETCORRECTIONMODE
	• ABSOLUTE Interpret the value specified in *lrCorrection* as absolute value. If the value is not changed, no correction will be made with an edge at *xStart*. The correction that can be made has an upper and a lower limit. The maximum correction is: $lrCorrection_{max} = (2^{31}-1) / 10^{Slave\ axis\ resolution}$ $lCorrection_{min} = -2^{31} / 10^{Slave\ axis\ resolution}$ The value "0" for *lrCorrection* results in the state without correction if no overflow has been exceeded. • RELATIVE Interpret the value specified in *lrCorrection* as a relative value. If the value is not changed, an edge at *xStart* results in a double correction. The correction that can be made is not limited. This is the reason why this setting is used, for example, to implement cut length control. **Note:** A change made to the value of this variable will take effect with a rising edge at *xStart*.
eTransitionType	Data type – E_TRANSITIONTYPE
	Type of travel profile: • (NONE) • MASTER-BASED A master-based travel profile is generated for offset correction. A 5th order rest-rest polynomial is used. With extensive corrections, this can lead to high speeds and accelerations that the slave axis might no longer be able to follow. • (TIME-BASED) Reserved for future functions • PROFILE-BASED For offset correction, a profile-based travel profile is generated with the specified dynamic settings.

29179459/EN – 03/2020

Variable name	Description
lrMasterDistance	Data type: LREAL – Floating-point number
	Distance of the master axis (in user units of the master axis) within which a master-based offset correction is performed.

Out

The dynamic parameters are scaled in user units of the master axis.

Variable name	Description
xProfileActive	Data type – BOOL
	• TRUE – Profile generation active • FALSE – Profile generation inactive
lrSetpPosition	Data type: LREAL – Floating-point number
	Setpoint position
lrSetpVelocity	Data type: LREAL – Floating-point number
	Target speed
lrSetpAcceleration	Data type: LREAL – Floating-point number
	Setpoint acceleration
lrDeltaPosition	Data type: LREAL – Floating-point number
	Position difference to the last PLC cycle in PLC units
xInPosition	Data type: BOOL
	• TRUE – Specified position is reached • FALSE – Specified position is not reached
xSetOffsetValueDone	Data type – BOOL
	• TRUE – Offset correction has been set • FALSE – Offset correction has not been set

29179459/EN – 03/2020

YOffsetCorrection

Using the *YOffsetCorrection* function, you can correct the amplitude position of the slave axis on the fly. The correction is carried out by superimposing a profile-based or master-based travel profile. With profile-based correction, the dynamic values are scaled in the user units of the slave axis.

The following control and status variables are available:

Interface in the IEC Editor

YOffsetCorrection	OffsetCorrection_UI	
IN	SEW_ICam.ST_OffsetCorrection_IN	
xStart	BOOL	FALSE
lrCorrection	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0
OUT	SEW_ICam.ST_OffsetCorrectionUI_OUT	
xModeActive	BOOL	FALSE
xProfileActive	BOOL	FALSE
lrSetpPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xInPosition	BOOL	FALSE
xFBError	BOOL	FALSE
udiFBErrorID	UDINT	0

9007219942577163

IN

The dynamic parameters are scaled in user units of the slave axis.

Variable name	Description
xStart	Data type – BOOL
	• TRUE – Start Set *xStart* to "TRUE" to create an offset profile that offsets the slave position to the curve position by *lrCorrection*. The reference point remains constant, which means the original state is restored using *lrCorrection* equals zero (= 0). Set *eTransitionType* to "MASTERBASED" or "PROFILE-BASED" for this purpose. • FALSE – Stop
lrCorrection	Data type: LREAL – Floating-point number
	Correction value
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrDeceleration	Data type: LREAL – Floating-point number
	Deceleration
lrJerk	Data type: LREAL – Floating-point number
	Jerk

29179459/EN – 03/2020

Variable name	Description
xSetOffsetValue	Data type – BOOL
	With a rising edge, the correction value is set to the value defined in *lrOffsetValue*.
lrOffsetValue	Data type: LREAL – Floating-point number
	Correction value that is set with a rising edge at *xSetOffset-Value*.

Config

Variable name	Description
eOffsetCorrectionMode	Data type – E_OFFSETCORRECTIONMODE
	• ABSOLUTE Interpret the value specified in *lrCorrection* as absolute value. If the value is not changed, no correction will be made with an edge at *xStart*. The correction that can be made has an upper and a lower limit. The maximum correction is: $lrCorrection_{max} = (2^{31}-1) / 10^{Slave\ axis\ resolution}$ $lCorrection_{min} = -2^{31} / 10^{Slave\ axis\ resolution}$ The value "0" for *lrCorrection* results in the state without correction if no overflow has been exceeded. • RELATIVE Interpret the value specified in *lrCorrection* as a relative value. If the value is not changed, an edge at *xStart* results in a double correction. The correction that can be made is not limited. This is the reason why this setting is used, for example, to implement cut length control. **Note:** A change made to the value of this variable will take effect with a rising edge at *xStart*.
eTransitionType	Data type – E_TRANSITIONTYPE
	Type of travel profile: • (NONE) • MASTER-BASED A master-based travel profile is generated for offset correction. A 5th order rest-rest polynomial is used. With extensive corrections, this can lead to high speeds and accelerations that the slave axis might no longer be able to follow. • (TIME-BASED) Reserved for future functions • PROFILE-BASED For offset correction, a profile-based travel profile is generated with the specified dynamic settings.

Variable name	Description
lrMasterDistance	Data type: LREAL – Floating-point number
	Distance of the master axis (in user units of the master axis) within which a master-based offset correction is performed.

Out

The dynamic parameters are scaled in user units of the slave axis.

Variable name	Description
xProfileActive	Data type – BOOL
	• TRUE – Profile generation active • FALSE – Profile generation inactive
lrSetpPosition	Data type: LREAL – Floating-point number
	Setpoint position
lrSetpVelocity	Data type: LREAL – Floating-point number
	Target speed
lrSetpAcceleration	Data type: LREAL – Floating-point number
	Setpoint acceleration
lrDeltaPosition	Data type: LREAL – Floating-point number
	Position difference to the last PLC cycle in PLC units
xInPosition	Data type: BOOL
	• TRUE – Specified position is reached • FALSE – Specified position is not reached
xSetOffsetValueDone	Data type – BOOL
	• TRUE – Offset correction has been set • FALSE – Offset correction has not been set

29179459/EN – 03/2020

Adjust

The *Adjust* function is used to align the axis to the position specified by the operating mode. The specified position is calculated at a rising edge at *xActivate*. Depending on *eModuloMode*, the *Adjust* function determines the difference between the current position of the slave axis and the position specified by the operating mode. The difference is output via *lrAdjustValue*. When adjusting, this difference is moved to 0 and the axis is in this way brought to the specified position.

The following control and status variables are available:

Interface in the IEC Editor

Adjust	FB_UI_AdjustProfilgenerator	
IN	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_IN	
xStart	BOOL	FALSE
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0
CONFIG	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_CONFIG	
eModuloMode	E_MODULOMODE	MODULO_OFF
OUT	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_OUT	
xAdjusted	BOOL	FALSE
lrAdjustValue	LREAL	0
eActiveModuloMode	E_MODULOMODE	MODULO_OFF

20687723275

The *Adjust* process can be controlled via *xStart* and via the dynamic parameters (*lrVelocity*, *lrAcceleration*, *lrDeceleration*, *lrJerk*). The dynamic parameters of the *Adjust* function are scaled in user units of the slave axis. The *Adjust* process is complete when *OUT.xAdjusted* is set to *"TRUE"* and *OUT.lrAdjustValue* is equal to *0*.

In

Changes to these variables are applied immediately.

The dynamic parameters are scaled in user units of the slave axis.

Variable name	Description
xStart	Data type – BOOL
	• TRUE – Start Set *xStart* to "TRUE" to create an offset profile that brings the slave position to the current curve position. • FALSE – Stop
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrDeceleration	Data type: LREAL – Floating-point number
	Deceleration
lrJerk	Data type: LREAL – Floating-point number
	Jerk

Config

Variable name	Description
eModuloMode	Data type: E_MODULOMODE
	Modulo travel strategy: • MODULO_OFF Modulo absolute • MODULO_SHORT Modulo absolute – shortest distance • MODULO_CW Modulo absolute – positive direction • MODULO_CCW Modulo absolute – negative direction • MODULO_RELATIVE Modulo relative The parameter is adopted with a rising edge at *xStart* but will take effect with a falling edge at *xStart* when the "Stop at position" function (*eStopMode*) is performed.

Out

Variable name	Description
xAdjusted	Data type: BOOL
	• TRUE – Slave axis is aligned with the current curve position (offset = 0). • FALSE – Slave axis is not aligned with the current curve position (offset ≠ 0).
lrAdjustValue	Data type: LREAL – Floating-point number
	The offset between the current position of the profile generation for the slave axis and the curve position (slave)
eActiveModuloMode	Data type: E_MODULOMODE
	Modulo mode for calculating the offset

29179459/EN – 03/2020

MasterUserProgram

The *MasterUserProgram* structure is only processed if *UserProgram* is set as the master source. When the operating mode is activated and started, the setpoint profile of the axis follows the profile provided for *MasterUserProgram.In.lrMasterPosition*. It is the responsibility of the user to limit speeds and accelerations in such a way that the axis is able to follow the setpoints. Major jumps in the setpoint profile may lead to a lag error in the axis or to the speed monitoring triggering/responding. If speed values and acceleration values of the setpoint profile are to be processed further (e.g. transmitted to the inverter as precontrol values together with the process data), the corresponding values of the master signal must also be connected at this point.

In

The dynamic parameters are scaled in user units of the master axis.

Variable name	Description
lrMasterPosition	Data type: LREAL – Floating-point number
	Position of the connected master signal
lrMasterVelocity	Data type: LREAL – Floating-point number
	Speed (velocity) of the connected master signal
lrMasterAcceleration	Data type: LREAL – Floating-point number
	Acceleration of the connected master signal

Config

The *CONFIG* structure is only processed if *UserProgram* has been set as the master source. The parameters are initialized with the values configured in *Tracking* operating mode.

Changes to these variables are applied to *xActivate* with a rising edge.

The dynamic parameters are scaled in user units.

Variable name	Description
lrMasterModuloMax	Data type: LREAL – Floating-point number
	Modulo maximum of the connected master signal
lrMasterModuloMin	Data type: LREAL – Floating-point number
	Modulo minimum of the connected master signal
uiMasterResolution	Data type: UINT
	Number of decimal places processed within the position value of the connected master signal

stTimeBaseFactor

Time base of the connected master signal

Variable name	Description
eVelocity	Data type: E_VELOCITYFACTOR
	Speed (velocity) of the connected master signal: • MINUTE (e.g. m/min) • SECOND (e.g. m/s)

Variable name	Description
eAcceleration	Data type: E_ACCELERATIONFACTOR
	Acceleration of the connected master signal: • MINUTE_MINUTE (e.g. m/min^2) • MINUTE_SECOND (e.g. m/min/s) • SECOND_SECOND (e.g. m/s^2)

29179459/EN – 03/2020

6.6.9 Electronic cam (Camming)

INFORMATION

Only visible when using MOVIKIT® MultiMotion Camming.

The term camming (electronic cam) refers to a definite assignment of positions between a master drive and a slave drive. In *Camming* operating mode, master-based profiles are generated where the position of the slave axis results from the position of the master axis based on a curve description.

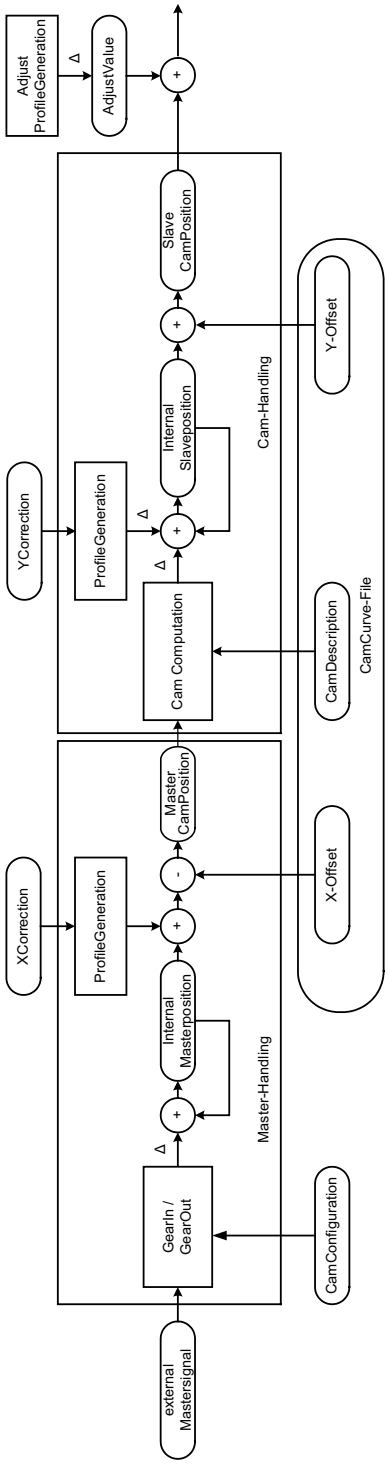
The following control and status variables are available:

Interface in the IEC Editor

Camming	SEW_Cam.ModeCamming_UI	
IN	SEW_ICam.ST_Camming_IN	
xActivate	BOOL	FALSE
xStart	BOOL	FALSE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
xGetNewCamDescription	BOOL	FALSE
CONFIG	SEW_ICam.ST_Camming_CONFIG	
Start	ST_CammingStart_CONFIG	
lrGearInDistance	LREAL	0
lrGearInOffset	LREAL	0
lrGearInReferencePosition	LREAL	0
eStartMode	E_CAMMINGSTARTMODE	CW_DIRECT
xStartWithLastXCorrection	BOOL	FALSE
Stop	ST_CammingStop_CONFIG	
lrGearOutDistance	LREAL	0
lrGearOutStopPosition	LREAL	0
eStopMode	E_CAMMINGSTOPMODE	CW_DIRECT
Change	ST_CammingChange_CONFIG	
lrChangeReferencePosition	LREAL	0
eChangeMode	E_CAMMINGCHANGEMODE	DIRECT
eTransitionMode	E_CAMMINGCHANGETRANSITION	DIRECT
FileHandling	CamFileHandler_UI	
XOffsetCorrection	OffsetCorrection_UI	
YOffsetCorrection	OffsetCorrection_UI	
Adjust	AdjustProfilgenerator_UI	
OUT	SEW_ICam.ST_Camming_OUT	
xModeActive	BOOL	FALSE
xProfileActive	BOOL	FALSE
eGearingState	E_GEARING_STATE	STOPPED
lrSetpPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
stSetpointMasterCam	SEW_IInterpolModes.ST_InterpolatedValues_UserUnits	
lrPosition	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
xSetMasterValueDone	BOOL	FALSE
xSetSlaveValueDone	BOOL	FALSE
xGetNewCamDescriptionDone	BOOL	FALSE

9007219965711115

Block diagram: Data flow in "Camming" mode



25804209291

- An internal master position is established from the external master signal by means of synchronizing and desynchronizing mechanisms. This internal master position passes through the range of the x-axis of the curve description. The synchronization and desynchronization mechanisms are configured using the "CONFIG" structure.
- The internal master position is passed to the curve evaluation once the x-offset of the curve description has been processed.
- Curve evaluation uses the "CamDescription" structure to determine the matching slave position.

- The slave position is passed to profile generation once the y-offset of the cam description has been processed.
- Dynamic offset corrections are possible both in x-direction (master) and y-direction (slave).
- The axis is adjusted to the curve position by means of the adjust function.

In

The dynamic parameters are scaled in user units.

Variable name	Description
xActivate	Data type – BOOL
	• TRUE – Activate • FALSE – Stop If *xActivate* is set to "FALSE", the operating mode stops at the last setpoint position, and speed and acceleration go to zero.
xStart	Data type – BOOL
	• TRUE – Start function • FALSE – Stop function
xSetMasterValue	Data type – BOOL
	This variable is only evaluated when *Camming* operating mode is disabled (*xActivate* must have the value "FALSE"). The internal master position is set to the value of *lrMaster-Value* at a rising edge.
lrMasterValue	Data type: LREAL – Floating-point number
	Internal master position Is transferred from *xSetMasterValue* with a rising edge.
xSetSlaveValue	Data type – BOOL
	This variable is only evaluated when the operating mode is disabled (*xActivate* must have the value "FALSE"). The internal slave position is set to the value of *lrSlave-Value* at a rising edge.
lrSlaveValue	Data type: LREAL – Floating-point number
	Internal slave position Is transferred from *xSetSlaveValue* with a rising edge.
xReset	Data type – BOOL
	This variable is only evaluated when the operating mode is disabled (*xActivate* must have the value "FALSE"). The state of the operating mode is reset at a rising edge. This sets *OUT.eGearingState* to *STOPPED*. This may be necessary, for example, if the master position stops during desynchronization.

Variable name	Description
xGetNewCamDescription	Data type: BOOL
	• TRUE – Change curves using the configured behavior. • FALSE – Do not change curves.

Config

The dynamic parameters are scaled in user units.

Variable name	Description
eCycleMode	Data type – E_CYCLEMODE
	Behavior after having passed through a curve profile: • ADJUST_SLAVE_REFERENCE_POSITION Place the reference point for the next curve onto the end point of the previous curve. • KEEP_SLAVE_REFERENCE_POSITION Interpret the curve profile as absolute position of the slave axis.

Note:

You find an application example that explains the *eCycleMode* variable in chapter "Using eCycleMode" (→ 🗎 226).

Start

Variable name	Description
lrGearInDistance	Data type: LREAL – Floating-point number
	Synchronization distance (in user units of the master axis) *lrGearInDistance = 0* means that the slave synchronizes without a transition function. Depending on the curve profile, the setpoint profile may be inconsistent. With *lrGearInDistance > 0*, a transition function is created automatically when synchronization starts. The parameter only takes effect if *eTransitionType* has the value "TIMEBASED". An automatic transition function is calculated so that the slave axis is synchronous after the synchronizing time.
lrGearInOffset	Data type: LREAL – Floating-point number
	Offset to the reference position *lrGearInReferencePosition* (master)
lrGearIn ReferencePosition	Data type: LREAL – Floating-point number
	Reference position (in user units of the master axis). This reference position takes effect only with the following settings for *eStartMode*: • *WITH_CW_REFERENCE_POSITION* • *WITH_CCW_REFERENCE_POSITION* • *WITH_CW_CCW_REFERENCE_POSITION*

29179459/EN – 03/2020

Variable name	Description
eStartMode	Data type: E_CAMMINGSTARTMODE
	Behavior when setting *xStart*: • CW_DIRECT Immediate synchronization with a rising edge of *xStart* with a positive direction of movement of the master • CCW_DIRECT Immediate synchronization with a rising edge of *xStart* with a negative direction of movement of the master • CW_CCW_DIRECT Immediate synchronization with a rising edge of *xStart* with a positive or negative direction of movement of the master • WITH_CW_REFERENCE_POSITION Synchronization based on the reference position defined in *lrGearInReferencePosition* with a positive direction of movement of the master • WITH_CCW_REFERENCE_POSITION Synchronization based on the reference position defined in *lrGearInReferencePosition* with a negative direction of movement of the master • WITH_CW_CCW_REFERENCE_POSITION Synchronization based on the reference position defined in *lrGearInReferencePosition* with a positive or negative direction of movement of the master
eTransitionType	Data type – E_TRANSITIONTYPE
	Type of an automatically calculated transition function when synchronizing: • NONE No transition function is calculated when synchronizing. This setting can lead to a discontinuous setpoint profile (e.g. speed jump). • MASTER-BASED A master-based transition function is calculated for synchronizing. The duration of the transition function is defined by the *lrGearInDistance* parameter. If *lrGearInDistance* has the value 0, the behavior is the same as for the setting "NONE. • TIME-BASED A time-based transition function is calculated for synchronizing. The duration of the transition function is defined by the *lrGearInTime* parameter. • (PROFILE-BASED) Reserved for future functions

29179459/EN – 03/2020

Variable name	Description
lrGearInTime	Data type: LREAL – Floating-point number
	Synchronizing time [s] for time-based synchronization The parameter only takes effect if *eTransitionType* has the value "TIMEBASED". An automatic transition function is calculated so that the slave axis is synchronous after the synchronizing time.

Stop

Variable name	Description
lrGearOutDistance	Data type: LREAL – Floating-point number
	Desynchronization distance (in user units of the master axis) *lrGearOutDistance = 0* means that the slave synchronizes without a transition function. Depending on the curve profile, the setpoint profile may be inconsistent. With *lrGearOutDistance > 0*, a transition function is created automatically when desynchronization starts. The parameter only takes effect if *eTransitionType* has the value "TIMEBASED". An automatic transition function is calculated so that the slave axis is synchronous after the synchronizing time.
lrGearOut StopPosition	Data type: LREAL – Floating-point number
	Stop position (in user units of the master axis). This stop position takes effect only with the following settings for *eStopMode*: • WITH_CW_INTERNAL_POSITION • WITH_CCW_INTERNAL_POSITION • WITH_CW_CCW_INTERNAL_POSITION • WITH_CW_ EXTERNAL_POSITION • WITH_CCW_ EXTERNAL_POSITION • WITH_CW_CCW_EXTERNAL_POSITION

29179459/EN – 03/2020

Variable name	Description
eStopMode	Data type: E_CAMMINGSTOPMODE
	Behavior when resetting *xStart*: • CW_DIRECT Immediate desynchronization with a rising edge of *xStart* with a positive direction of movement of the master • CCW_DIRECT Immediate desynchronization with a rising edge of *xStart* with a negative direction of movement of the master • CW_CCW_DIRECT Immediate desynchronization with a rising edge of *xStart* with a positive or negative direction of movement of the master • WITH_CW_INTERNAL_POSITION Desynchronization to the stop position of the internal master defined by *lrGearOutStopPosition* with a positive direction of movement • WITH_CCW_INTERNAL_POSITION Desynchronization to the stop position of the internal master defined by *lrGearOutStopPosition* with a negative direction of movement • WITH_CW_CCW_INTERNAL_POSITION Desynchronization to the stop position of the internal master defined by *lrGearOutStopPosition* with a positive or negative direction of movement • WITH_CW_EXTERNAL_POSITION Desynchronization to the stop position of the external master defined by *lrGearOutStopPosition* with a positive direction of movement • WITH_CCW_EXTERNAL_POSITION Desynchronization to the stop position of the external master defined by *lrGearOutStopPosition* with a negative direction of movement • WITH_CW_CCW_EXTERNAL_POSITION Desynchronization to the stop position of the external master defined by *lrGearOutStopPosition* with a positive or negative direction of movement

29179459/EN – 03/2020

Variable name	Description
eTransitionType	Data type – E_TRANSITIONTYPE
	Type of an automatically calculated transition function when desynchronizing: • NONE No transition function is calculated for desynchronizing. This setting can lead to a discontinuous setpoint profile (e.g. speed jump). • MASTER-BASED A master-based transition function is calculated for desynchronizing. The duration of the transition function is defined by the *lrGearOutDistance* parameter. If *lrGearOutDistance* has the value 0, the behavior is the same as for the setting "NONE". • TIME-BASED A time-based transition function is calculated for desynchronizing. The duration of the transition function is defined by the *lrGearOutTime* parameter. • (PROFILE-BASED) Reserved for future functions
lrGearOutTime	Data type: LREAL – Floating-point number
	Desynchronizing time [s] for time-based synchronization The parameter only takes effect if *eTransitionType* has the value "TIMEBASED". An automatic transition function is calculated in such a way that the slave axis is no longer synchronous after the desynchronizing time.

Change

Variable name	Description
lrChangeReference-Position	Data type: LREAL – Floating-point number
	Reference position of the master axis when changing curves (in user units of the master axis)
eChangeMode	Data type: E_CAMMINGCHANGEMODE
	Curve change trigger: • DIRECT • WITH_CW_INTERNAL_POSITION • WITH_CCW_INTERNAL_POSITION • WITH_CW_CCW_INTERNAL_POSITION • WITH_CW_EXTERNAL_POSITION • WITH_CCW_EXTERNAL_POSITION • WITH_CW_CCW_EXTERNAL_POSITION
eTransitionMode	Data type: E_CAMMINGTRANSITIONMODE
	Behavior on curve change: • DIRECT – Immediate change

29179459/EN – 03/2020

Out

The dynamic parameters are scaled in user units of the slave axis.

Variable name	Description
xModeActive	Data type – BOOL
	• TRUE – Operating mode activated • FALSE – Operating mode not activated
xProfileActive	Data type – BOOL
	• TRUE – Profile generation active • FALSE – Profile generation inactive
eGearingState	Data type: E_GEARING_STATE
	State of the axis: • STOPPED The slave axis is desynchronized and does not follow the master axis. • WAITING_FOR_STARTPOSITION Synchronization requested. The slave axis is waiting for a suitable master position. • GEAR_IN Synchronization is active. To synchronize with the master, the slave axis follows a transition function. • ACTIVE The slave axis is synchronized and follows the master axis. • WAITING_FOR_STOPPOSITION Desynchronization requested. The slave axis is waiting for a suitable master position. • GEAR_OUT Desynchronization is active. To reach the stop position, the slave axis follows a transition function.
lrSetpPosition	Data type: LREAL – Floating-point number
	Setpoint position
lrSetpVelocity	Data type: LREAL – Floating-point number
	Target speed
lrSetpAcceleration	Data type: LREAL – Floating-point number
	Setpoint acceleration
xSetMasterValue-Done	Data type: BOOL
	• TRUE – The internal master position has been set. • FALSE – The internal master position has not been set.
xSetSlaveValue-Done	Data type: BOOL
	• TRUE – The internal slave position has been set. • FALSE – The internal slave position has not been set.
xGetNewCamDe-scriptionDone	Data type: BOOL
	• TRUE – Curve change has been carried out. • FALSE – Curve change has not been carried out.

29179459/EN – 03/2020

stSetpointMasterCam

Setpoints of internal masters.

The dynamic parameters are scaled in user units of the master axis.

Variable name	Description
lrPosition	Data type: LREAL – Floating-point number
	Position
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration

29179459/EN – 03/2020

FileHandling

This structure offers all control and status information for the *Camming* operating mode in connection with file handling. This includes importing the curve list `CurveConfig.xml` and importing the curve files `*.camCurve`.

The following control and status variables are available:

Interface in the IEC Editor

FileHandling	CamFileHandler_UI	
IN	SEW_ICam.ST_CamFileHandler_IN	
xReadCurveFileList	BOOL	FALSE
xReadCurveFile	BOOL	FALSE
sFileName	STRING(80)	''
uiCurveNumber	UINT	0
CurveFileList	ST_CurveFileListUI	
stCurveFileList	SEW_ICam.ST_CurveFileList	
uiNumberOfLastCurveFile	UINT	0
sCurveFileName	ARRAY [1..gc_iMaxNumberOfCurveFiles] OF STRING	
OUT	SEW_ICam.ST_CamFileHandler_OUT	
xCurveFileListRead	BOOL	FALSE
xCurveFileRead	BOOL	FALSE
sLastReadCurveFile	STRING(80)	''
xFBError	BOOL	FALSE
udiFBErrorID	UDINT	0

9007219965716875

The import of curve list `CurveConfig.xml` does not start automatically when the program is initialized but must be triggered by setting *IN.xReadCurveFileList* to "TRUE". After successful import, *OUT.xCurveFileListRead* becomes "TRUE" and the curve list is displayed in the *CurveFileList* structure.

There are two options for importing the curve description:

- If the file is included in the curve list, simply pass the file number to *IN.uiCurveNumber* and set *IN.xReadCurveFile* to "TRUE".
- If the file is not included in the curve list, pass the file name to *IN.sFileName*. The name of the last imported file is displayed by *OUT.sLastReadCurveFile*. The imported curve description ends up in global variable *CamDescription*.

In

Changes to these variables are applied immediately.

Variable name	Description
xReadCurveFileList	Data type: BOOL
	• TRUE – Import the `CurveConfig.xml` file from the directory assigned to the axis The file contains a list of curve description files for the corresponding axis. The list is stored in the *CurveFileList* structure. • FALSE – Do not import a file.
xReadCurveFile	Data type: BOOL
	• TRUE – Import the curve description file The data included in the curve description file is saved in global structure *CamDescription* that is assigned to this axis. • FALSE – Do not import a curve description file.

Variable name	Description
sFileName	Data type: STRING(80)
	Name of the curve description file that is to be imported
uiCurveNumber	Data type: UINT
	Number of the curve description file that is to be imported from the *CurveFileList* structure

Out

Variable name	Description
xCurveFileListRead	Data type: BOOL
	• TRUE – The list of curve description files has been imported. • FALSE – The list of curve description files has not been imported.
xCurveFileRead	Data type: BOOL
	• TRUE – The curve description file specified has been imported. • FALSE – The curve description file specified has not been imported.
sLastReadCurveFile	Data type: STRING(80)
	Name of the curve description file that was imported last

29179459/EN – 03/2020

InterpolationHandler

INFORMATION

Only included if the function is activated in the "Basic settings" configuration menu under "Functions used".

The *Camming* mode offers the possibility to interpolate curve descriptions from curve point tables. The curve point table can be available in a source file or can be read in from the IEC program. See application example "Creating curve description from curve point table" (→ 🗎 238).

A defined segment in a curve description is configured for the interpolation of a curve point table. Subsequently, the user can use this curve description as well as the curve descriptions that are configured by importing a curve profile created with the Cam Editor. All functions of MOVIKIT® MultiMotion Camming can be used.

To operate the function, the *InterpolationHandler* structure is available in the *FileHandling* structure. Automatic code generation also creates the following instances for each axis that uses the function:

- *PointsList_AxisName of the type* PointsList

  The curve point table is stored in this structure if the interpolation method is "LINEAR" or "POLYNOM_3". The structure comprises up to 1024 pairs of curve points.

PointsList_Axis1	SEW_MK_MultiMotionCam.SEW_Cam.PointsList
lrNumberOfPoints	LREAL
astPoints	ARRAY [0..1024] OF SEW_PG_Interp.SEW_IPG_Base.ST_Point

31491274507

- *PointsListSpline_AxisName* of the type PointsListSpline

  The curve point table is stored in this structure if the interpolation method is "SPLINE". The structure comprises up to 64 pairs of curve points.

PointsListSpline_Axis1	SEW_MK_MultiMotionCam.SEW_Cam.PointsListSpline
lrNumberOfPoints	LREAL
astPoints	ARRAY [0..64] OF SEW_PG_Interp.SEW_IPG_Base.ST_Point

31491276939

Additionally, the following interface connections are made:

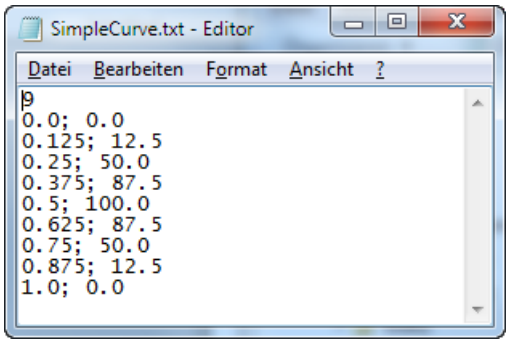
```
Axis1.itfTrackingProfile.linkMaster(Interface_Axis1.ProfileGeneration.Tracking);
Axis1._fbCammingProfile.linkMaster(VirtualAxis._fbProfileGenerationControl);
Axis1._fbCammingProfile.linkCamDescriptionScheduler(CamDescriptionScheduler_Axis1);
Axis1._fbCammingProfile.linkCamDescriptionSchedulerCyclic(CamDescriptionScheduler_Axis1);
Axis1._fbCammingProfile.linkPointsList(PointsList_Axis1);
Axis1._fbCammingProfile.linkPointsListSpline(PointsListSpline_Axis1);
```

31491272075

Curve point table

The file with the curve point table must be stored on the memory card of the MOVI-C® CONTROLLER in the same directory as the curve descriptions created with the MOVISUITE® Cam Editor, usually the directory `/DATA/Camming/Achsname`, where "Achsname" (axis name) is the identifier of the axis assigned in MOVISUITE®.

The curve point table must be stored in an ACII file in the following format:



31491136139

- In the first line there is a single number that defines the number n of pairs of curve points.
- This is followed by n lines, each containing a pair of numbers separated by a semicolon. The first number corresponds to the master position in user units of the master axis, the second number to the corresponding slave position in user units of the slave axis. It is not important whether the decimal point is a point (e.g. "0.125") or a comma (e.g. "0,125"). Both formats are processed.

Interpolation methods

The interpolation of curve point tables can be performed using one of the following interpolation methods:

- Linear (for up to 1024 curve points)

  Since a linear function is used to interpolate between the curve points, jumps in the speed profile can occur at the curve points. This method can therefore only be used meaningfully if the gradient of the interpolating straight lines and consequently the speed in the curve points changes only slightly.
- Polynom_3 (for up to 1024 curve points)

  Interpolation between the curve points is performed with a 3rd order polynomial; the speed value in a curve point is estimated using the adjacent curve points. The result is a process in which there are no jumps in the speed profile at the curve points, but there are still jumps in the acceleration profile.
- Spline (for up to 64 curve points)

  With this method, interpolation between the curve points is also performed with a 3rd order polynomial. However, the algorithm determines the speed in each curve point in such a way that there is a continuous speed and acceleration profile. Only a maximum of 64 curve points can be interpolated because this method is rather complex in mathematical terms.

In

Variable name	Description
xLoadPointsFrom-File	Data type – BOOL
	Reading of the curve point table is triggered at a rising edge and the configuration of a segment in the curve description is performed according to the set interpolation method.
sFileName	Data type: STRING(80)
	Name of the file with the curve point table to be read

Variable name	Description
eInterpolationSegmentMode	Data type – E_INTERPOLATIONMODE
	Set the interpolation type in use: • LINEAR • POLYNOM_3 • SPLINE

Out

Variable name	Description
xPointsFromFileLoaded	Data type – BOOL
	• TRUE – Curve point table read and corresponding segment in curve description configured • FALSE – Curve point table not read
sLastReadPointsFile	Data type: STRING(80)
	Name of the file from which the last curve point table was read.

CurveFileList

Information regarding the curve description file

Variable name	Description
uiNumberOfLastCurveFile	Data type: UINT
	Number of the last curve description file in the list
sCurveFileName	Data type: ARRAY OF STRING
	Contains the names of the configured curve description files

XOffsetCorrection

Using the *XOffsetCorrection* function, you can correct the phase position of the slave profile on the fly. The correction is carried out by superimposing a profile-based or master-based travel profile. With profile-based correction, the dynamic values are scaled in the user units of the master axis.

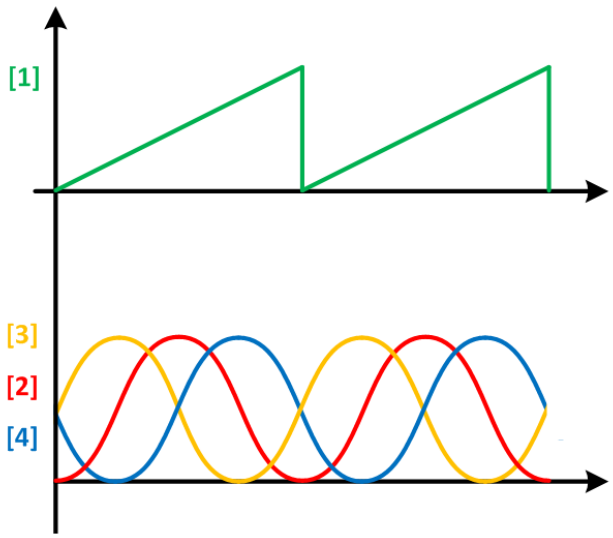
The following control and status variables are available:

Interface in the IEC Editor

XOffsetCorrection	OffsetCorrection_UI	
IN	SEW_ICam.ST_OffsetCorrection_IN	
xStart	BOOL	FALSE
lrCorrection	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0
OUT	SEW_ICam.ST_OffsetCorrectionUI_OUT	
xModeActive	BOOL	FALSE
xProfileActive	BOOL	FALSE
lrSetpPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xInPosition	BOOL	FALSE
xFBError	BOOL	FALSE
udiFBErrorID	UDINT	0

9007219942504331

Trace recording



24131018635

[1] Position of the master axis
[2] Curve profile of the slave axis with *lrCorrection = 0*
[3] Curve profile of the slave axis with *lrCorrection > 0*
[4] Curve profile of the slave axis with *lrCorrection < 0*

In

The dynamic parameters are scaled in user units of the master axis.

Variable name	Description
xStart	Data type – BOOL
	• TRUE – Start Set *xStart* to "TRUE" to create a travel profile that offsets the internal master to the external master by *lrCorrection*. The reference point remains constant, which means the original state is restored using *lrCorrection* equals zero (= 0). Set *eTransitionType* to "MASTERBASED" or "PROFILEBASED" for this purpose. • FALSE – Stop
lrCorrection	Data type: LREAL – Floating-point number
	Correction value
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrDeceleration	Data type: LREAL – Floating-point number
	Deceleration
lrJerk	Data type: LREAL – Floating-point number
	Jerk
xSetOffsetValue	Data type – BOOL
	With a rising edge, the correction value is set to the value defined in *lrOffsetValue*.
lrOffsetValue	Data type: LREAL – Floating-point number
	Correction value that is set with a rising edge at *xSetOffsetValue*.

29179459/EN – 03/2020

Config

Variable name	Description
eOffsetCorrectionMode	Data type – E_OFFSETCORRECTIONMODE
	• ABSOLUTE Interpret the value specified in *lrCorrection* as absolute value. If the value is not changed, no correction will be made with an edge at *xStart*. The correction that can be made has an upper and a lower limit. The maximum correction is: $lrCorrection_{max} = (2^{31}-1) / 10^{Slave\ axis\ resolution}$ $lCorrection_{min} = -2^{31} / 10^{Slave\ axis\ resolution}$ The value "0" for *lrCorrection* results in the state without correction if no overflow has been exceeded. • RELATIVE Interpret the value specified in *lrCorrection* as a relative value. If the value is not changed, an edge at *xStart* results in a double correction. The correction that can be made is not limited. This is the reason why this setting is used, for example, to implement cut length control. **Note:** A change made to the value of this variable will take effect with a rising edge at *xStart*.
eTransitionType	Data type – E_TRANSITIONTYPE
	Type of travel profile: • (NONE) • MASTER-BASED A master-based travel profile is generated for offset correction. A 5th order rest-rest polynomial is used. With extensive corrections, this can lead to high speeds and accelerations that the slave axis might no longer be able to follow. • (TIME-BASED) Reserved for future functions • PROFILE-BASED For offset correction, a profile-based travel profile is generated with the specified dynamic settings.
lrMasterDistance	Data type: LREAL – Floating-point number
	Distance of the master axis (in user units of the master axis) within which a master-based offset correction is performed.

29179459/EN – 03/2020

Out

The dynamic parameters are scaled in user units of the master axis.

Variable name	Description
xProfileActive	Data type – BOOL
	• TRUE – Profile generation active • FALSE – Profile generation inactive
lrSetpPosition	Data type: LREAL – Floating-point number
	Setpoint position
lrSetpVelocity	Data type: LREAL – Floating-point number
	Target speed
lrSetpAcceleration	Data type: LREAL – Floating-point number
	Setpoint acceleration
lrDeltaPosition	Data type: LREAL – Floating-point number
	Position difference to the last PLC cycle in PLC units
xInPosition	Data type: BOOL
	• TRUE – Specified position is reached • FALSE – Specified position is not reached
xSetOffsetValueDone	Data type – BOOL
	• TRUE – Offset correction has been set • FALSE – Offset correction has not been set

29179459/EN – 03/2020

YOffsetCorrection

Using the *YOffsetCorrection* function, you can correct the amplitude position of the slave axis on the fly. The correction is carried out by superimposing a profile-based or master-based travel profile. With profile-based correction, the dynamic values are scaled in the user units of the slave axis.

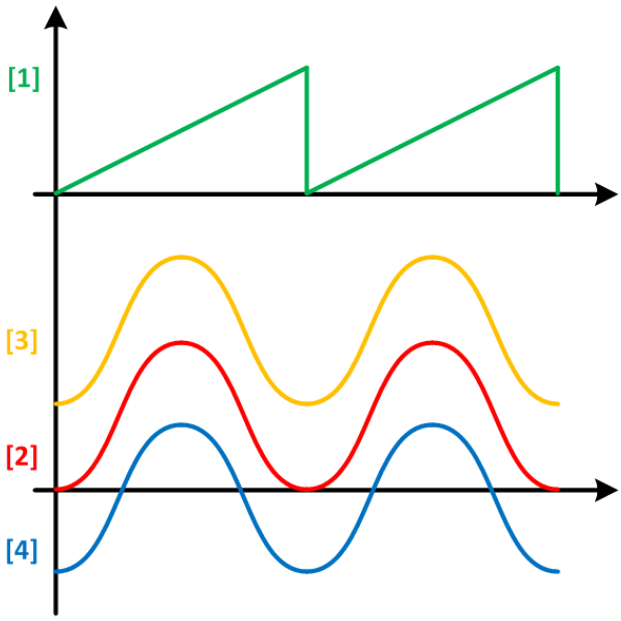
The following control and status variables are available:

Interface in the IEC Editor

YOffsetCorrection	OffsetCorrection_UI	
IN	SEW_ICam.ST_OffsetCorrection_IN	
xStart	BOOL	FALSE
lrCorrection	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0
OUT	SEW_ICam.ST_OffsetCorrectionUI_OUT	
xModeActive	BOOL	FALSE
xProfileActive	BOOL	FALSE
lrSetpPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xInPosition	BOOL	FALSE
xFBError	BOOL	FALSE
udiFBErrorID	UDINT	0

9007219942577163

Trace recording



24131021963

[1] Position of the master axis
[2] Curve profile of the slave axis with *lrCorrection = 0*
[3] Curve profile of the slave axis with *lrCorrection > 0*
[4] Curve profile of the slave axis with *lrCorrection < 0*

In

The dynamic parameters are scaled in user units of the slave axis.

Variable name	Description
xStart	Data type – BOOL
	• TRUE – Start Set *xStart* to "TRUE" to create an offset profile that offsets the slave position to the curve position by *lrCorrection*. The reference point remains constant, which means the original state is restored using *lrCorrection* equals zero (= 0). Set *eTransitionType* to "MASTERBASED" or "PROFILE-BASED" for this purpose. • FALSE – Stop
lrCorrection	Data type: LREAL – Floating-point number
	Correction value
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrDeceleration	Data type: LREAL – Floating-point number
	Deceleration
lrJerk	Data type: LREAL – Floating-point number
	Jerk
xSetOffsetValue	Data type – BOOL
	With a rising edge, the correction value is set to the value defined in *lrOffsetValue*.
lrOffsetValue	Data type: LREAL – Floating-point number
	Correction value that is set with a rising edge at *xSetOffset-Value*.

29179459/EN – 03/2020

Config

Variable name	Description
eOffsetCorrectionMode	Data type – E_OFFSETCORRECTIONMODE
	• ABSOLUTE Interpret the value specified in *lrCorrection* as absolute value. If the value is not changed, no correction will be made with an edge at *xStart*. The correction that can be made has an upper and a lower limit. The maximum correction is: $lrCorrection_{max} = (2^{31}-1) / 10^{Slave\ axis\ resolution}$ $lCorrection_{min} = -2^{31} / 10^{Slave\ axis\ resolution}$ The value "0" for *lrCorrection* results in the state without correction if no overflow has been exceeded. • RELATIVE Interpret the value specified in *lrCorrection* as a relative value. If the value is not changed, an edge at *xStart* results in a double correction. The correction that can be made is not limited. This is the reason why this setting is used, for example, to implement cut length control. **Note:** A change made to the value of this variable will take effect with a rising edge at *xStart*.
eTransitionType	Data type – E_TRANSITIONTYPE
	Type of travel profile: • (NONE) • MASTER-BASED A master-based travel profile is generated for offset correction. A 5th order rest-rest polynomial is used. With extensive corrections, this can lead to high speeds and accelerations that the slave axis might no longer be able to follow. • (TIME-BASED) Reserved for future functions • PROFILE-BASED For offset correction, a profile-based travel profile is generated with the specified dynamic settings.
lrMasterDistance	Data type: LREAL – Floating-point number
	Distance of the master axis (in user units of the master axis) within which a master-based offset correction is performed.

29179459/EN – 03/2020

Out

The dynamic parameters are scaled in user units of the slave axis.

Variable name	Description
xProfileActive	Data type – BOOL
	• TRUE – Profile generation active • FALSE – Profile generation inactive
lrSetpPosition	Data type: LREAL – Floating-point number
	Setpoint position
lrSetpVelocity	Data type: LREAL – Floating-point number
	Target speed
lrSetpAcceleration	Data type: LREAL – Floating-point number
	Setpoint acceleration
lrDeltaPosition	Data type: LREAL – Floating-point number
	Position difference to the last PLC cycle in PLC units
xInPosition	Data type: BOOL
	• TRUE – Specified position is reached • FALSE – Specified position is not reached
xSetOffsetValueDone	Data type – BOOL
	• TRUE – Offset correction has been set • FALSE – Offset correction has not been set

29179459/EN – 03/2020

Adjust

The *Adjust* function is used to align the axis to the position specified by the operating mode. The specified position is calculated at a rising edge at *xActivate*. Depending on *eModuloMode*, the *Adjust* function determines the difference between the current position of the slave axis and the position specified by the operating mode. The difference is output via *lrAdjustValue*. When adjusting, this difference is moved to 0 and the axis is in this way brought to the specified position.

The following control and status variables are available:

Interface in the IEC Editor

Adjust	FB_UI_AdjustProfilgenerator	
IN	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_IN	
xStart	BOOL	FALSE
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0
CONFIG	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_CONFIG	
eModuloMode	E_MODULOMODE	MODULO_OFF
OUT	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_OUT	
xAdjusted	BOOL	FALSE
lrAdjustValue	LREAL	0
eActiveModuloMode	E_MODULOMODE	MODULO_OFF

20687723275

The *Adjust* process can be controlled via *xStart* and via the dynamic parameters (*lrVelocity*, *lrAcceleration*, *lrDeceleration*, *lrJerk*). The dynamic parameters of the *Adjust* function are scaled in user units of the slave axis. The *Adjust* process is complete when *OUT.xAdjusted* is set to *"TRUE"* and *OUT.lrAdjustValue* is equal to *0*.

In

Changes to these variables are applied immediately.

The dynamic parameters are scaled in user units of the slave axis.

Variable name	Description
xStart	Data type – BOOL
	• TRUE – Start Set *xStart* to "TRUE" to create an offset profile that brings the slave position to the current curve position. • FALSE – Stop
lrVelocity	Data type: LREAL – Floating-point number
	Speed (velocity)
lrAcceleration	Data type: LREAL – Floating-point number
	Acceleration
lrDeceleration	Data type: LREAL – Floating-point number
	Deceleration
lrJerk	Data type: LREAL – Floating-point number
	Jerk

29179459/EN – 03/2020

Config

Variable name	Description
eModuloMode	Data type: E_MODULOMODE
	Modulo travel strategy: • MODULO_OFF Modulo absolute • MODULO_SHORT Modulo absolute – shortest distance • MODULO_CW Modulo absolute – positive direction • MODULO_CCW Modulo absolute – negative direction • MODULO_RELATIVE Modulo relative The parameter is adopted with a rising edge at *xStart* but will take effect with a falling edge at *xStart* when the "Stop at position" function (*eStopMode*) is performed.

Out

Variable name	Description
xAdjusted	Data type: BOOL
	• TRUE – Slave axis is aligned with the current curve position (offset = 0). • FALSE – Slave axis is not aligned with the current curve position (offset ≠ 0).
lrAdjustValue	Data type: LREAL – Floating-point number
	The offset between the current position of the profile generation for the slave axis and the curve position (slave)
eActiveModuloMode	Data type: E_MODULOMODE
	Modulo mode for calculating the offset

29179459/EN – 03/2020

MasterUserProgram

The *MasterUserProgram* structure is only processed if *UserProgram* is set as the master source. When the operating mode is activated and started, the setpoint profile of the axis follows the profile provided for *MasterUserProgram.In.lrMasterPosition*. It is the responsibility of the user to limit speeds and accelerations in such a way that the axis is able to follow the setpoints. Major jumps in the setpoint profile may lead to a lag error in the axis or to the speed monitoring triggering/responding. If speed values and acceleration values of the setpoint profile are to be processed further (e.g. transmitted to the inverter as precontrol values together with the process data), the corresponding values of the master signal must also be connected at this point.

In

The dynamic parameters are scaled in user units of the master axis.

Variable name	Description
lrMasterPosition	Data type: LREAL – Floating-point number
	Position of the connected master signal
lrMasterVelocity	Data type: LREAL – Floating-point number
	Speed (velocity) of the connected master signal
lrMasterAcceleration	Data type: LREAL – Floating-point number
	Acceleration of the connected master signal

Config

The *CONFIG* structure is only processed if *UserProgram* has been set as the master source. The parameters are initialized with the values configured in *Tracking* operating mode.

Changes to these variables are applied to *xActivate* with a rising edge.

The dynamic parameters are scaled in user units.

Variable name	Description
lrMasterModuloMax	Data type: LREAL – Floating-point number
	Modulo maximum of the connected master signal
lrMasterModuloMin	Data type: LREAL – Floating-point number
	Modulo minimum of the connected master signal
uiMasterResolution	Data type: UINT
	Number of decimal places processed within the position value of the connected master signal

stTimeBaseFactor

Time base of the connected master signal

Variable name	Description
eVelocity	Data type: E_VELOCITYFACTOR
	Speed (velocity) of the connected master signal: • MINUTE (e.g. m/min) • SECOND (e.g. m/s)

29179459/EN – 03/2020

Variable name	Description
eAcceleration	Data type: E_ACCELERATIONFACTOR
	Acceleration of the connected master signal: • MINUTE_MINUTE (e.g. m/min^2) • MINUTE_SECOND (e.g. m/min/s) • SECOND_SECOND (e.g. m/s^2)

29179459/EN – 03/2020

CamDescription

Curve profiles to be processed in the *Camming* operating mode are usually designed using the Cam Editor. Next, they are loaded to the memory card of the MOVI-C® CONTROLLER from where they are read for processing. Curve description can also be created in the IEC program. Using so-called "constructors" is recommended in this case. You find an application example in chapter "Creating a curve description using constructors" (→ 🗎 230).

For each axis to which MOVIKIT® MultiMotion Camming is assigned, automatic code generation creates an instance of the type *CamDescription*, which is defined in the *SEW MOS Camming* library.

Interface in the IEC Editor

CamDescription_Axis2	SEW_MK_MultiMotionCam.SEW_Cam.CamDescription	
_lrX_Offset	LREAL	0
_lrY_Offset	LREAL	0
aCamSegment	ARRAY [1..SEW_ICam.gc_iMaxNumberOfCamSegments] OF CamSegment	
aCamSegment[1]	CamSegment	
_eCamProfileType	E_CAMPROFILETYPE	POLYNOM_5
fbLeft	SegmentBorder	
stSegmentBorder	SEW_ICam.ST_SegmentBorder	
lrX	LREAL	0
lrY	LREAL	0
lrVy	LREAL	0
lrAy	LREAL	0
lrJy	LREAL	-48000
fbRight	SegmentBorder	
stSegmentBorder	SEW_ICam.ST_SegmentBorder	
lrX	LREAL	0.5
lrY	LREAL	-100
lrVy	LREAL	0
lrAy	LREAL	0
lrJy	LREAL	-48000
aCamSegment[...]	CamSegment	
aCamSegment[20]	CamSegment	
_uiNrOfLastDefinedCamSegment	UINT	0
_uiNrOfActiveCamSegment	UINT	0
uiNrOfLastActiveCamSegment	UINT	0
CamProfileHandler	CamProfileHandler	

29017116555

Variable name	Description
lrX_Offset	Data type: LREAL – Floating-point number
	X-offset of the curve description as set in the Cam Editor

Variable name	Description
lrY_Offset	Data type: LREAL – Floating-point number
	Y-offset of the curve description as set in the Cam Editor

Variable name	Description
_uiNrOfLastDefined-CamSegment	Data type: UINT
	Number of the curve segment defined last

Variable name	Description
_uiNrOfActiveCamSeg-ment	Data type: UINT
	Number of the currently active curve segment

Variable name	Description
uiNrOfLastActiveCam-Segment	Data type: UINT
	Number of the last active curve segment

aCamSegment

Structure for describing the curve segments.

An array consists of 20 curve segments.

aCamSegment[1..20]

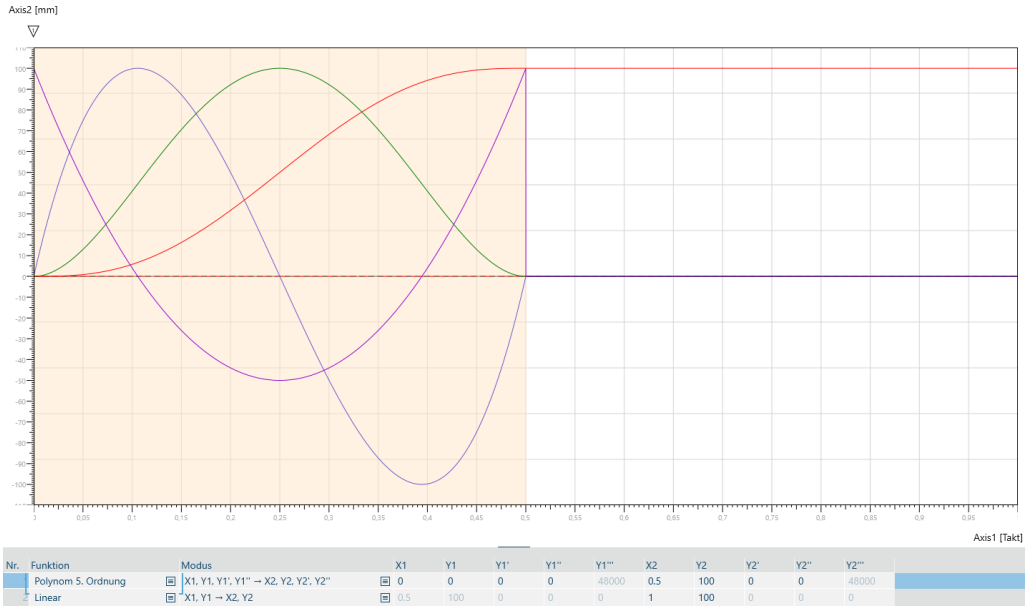
Structure for describing one of the curve segments.

A curve segment consists of several components.

Variable name	Description
_eCamProfileType	Data type – E_CAMPROFILETYPE
	Type of the mathematical curve function: • LINEAR • SINE • POLYNOM_3 • INCLINED_SINE • TRAPEZOID • POLYNOM_5 • MODIFIED_SINE • POLYNOM_7 • OPTIMIZED_TRAPEZOID • USER_DEFINED Can also be used with the "Interpolation" add-on: • LINEAR_INTERPOLATED • POLYNOM_3_INTERPOLATED • SPLINE-INTERPOLATED Can also be used with the "AntiSlosh" add-on: • ANTISLOSH

29179459/EN – 03/2020

The variables of the segment limits described in the subsequent chapters are shown in the Cam Editor in a table, as can be seen in the following diagram:



29017372427

fbLeft

Function values at the left segment border.

stSegmentBorder

Variable name	Description
lrX	Data type: LREAL – Floating-point number
	Position of the master axis
lrY	Data type: LREAL – Floating-point number
	Position of the slave axis
lrVy	Data type: LREAL – Floating-point number
	First derivative dY/dX
lrAy	Data type: LREAL – Floating-point number
	Second derivative d^2Y/dX^2
lrJy	Data type: LREAL – Floating-point number
	Third derivative d^3Y/dX^3

fbRight

Function values at the right segment border.

stSegmentBorder

Variable name	Description
lrX	Data type: LREAL – Floating-point number
	Position of the master axis

29179459/EN – 03/2020

Variable name	Description
lrY	Data type: LREAL – Floating-point number
	Position of the slave axis
lrVy	Data type: LREAL – Floating-point number
	First derivative dY/dX
lrAy	Data type: LREAL – Floating-point number
	Second derivative d^2Y/dX^2
lrJy	Data type: LREAL – Floating-point number
	Third derivative d^3Y/dX^3

CamProfileHandler

Structure where all available mathematical curve functions are gathered.

stActiveConfig (AntiSlosh)

INFORMATION

Only included if the function is activated in the "Basic settings" configuration menu under "Functions used".

INFORMATION

Each *CamDescription* may only contain one AntiSlosh segment.

Variable name	Description
lrSnapStart	Data type: LREAL – Floating-point number
	Derivation of the jerk (snap) at the starting point of the movement
lrSnapEnd	Data type: LREAL – Floating-point number
	Derivation of the jerk (snap) at the end point of the movement
lrDamping	Data type: LREAL – Floating-point number
	Damping of the "container-liquid" system
lrJerkMid	Data type: LREAL – Floating-point number
	Jerk in the middle of movement
lrNaturalFrequency	Data type: LREAL – Floating-point number
	Natural frequency of the "container-liquid" system
lrStepDistance	Data type: LREAL – Floating-point number
	Distance covered by the slave axis during the movement
lrVelocity	Data type: LREAL – Floating-point number
	Nominal speed of the application
lrCycleStart	Data type: LREAL – Floating-point number
	Position of the master at the beginning of the cycle
lrCycleEnd	Data type: LREAL – Floating-point number
	Position of the master at the end of the cycle
lrStartPos	Data type: LREAL – Floating-point number
	Position of the master at the beginning of the movement of the slave axis
lrDeltaX2	Data type: LREAL – Floating-point number
	Distance covered by the master during the movement of the slave axis

29179459/EN – 03/2020

7 Diagnostics

7.1 Monitor

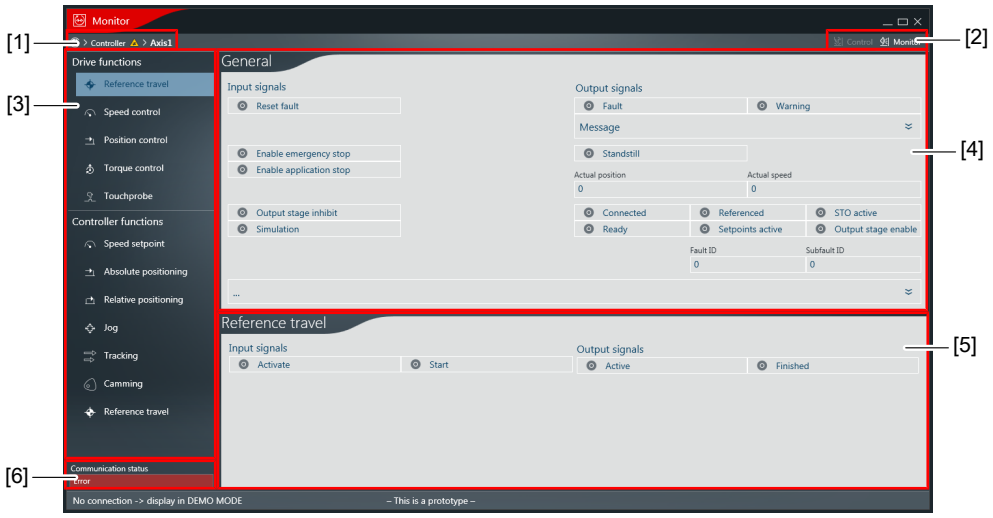
INFORMATION

Only available with "Advanced" permission level.

The MultiMotion monitor is a tool in the MOVISUITE® engineering software for monitoring and controlling axes or axis groups.

In MOVISUITE®, click [Monitor] in the Tools menu to open the tool from the context menu of a specific node.

The user interface of the MultiMotion monitor includes the following areas:



9007223554947083

No.	Area	Description
[1]	Device path and name	Path and name of the axis/axis group
[2]	Mode switching	• "Control" mode: Operating mode for setting control bits and control values. Control via the IEC program is ignored. • "Monitor" mode: Operating mode for monitoring the input and output values. In this mode, the monitor only has read-only rights and displays the values set by the controller.
[3]	Main menu	Main menu for opening the configuration menus of the functions available in each case, such as drive functions, controller functions, or functions of software modules.
[4]	General	Input and output signals of general settings
[5]	Functions	Depending on the function selected from the main menu, this area displays the corresponding input and output signals.
[6]	Communication status	Status of the communication link

8 Application examples

8.1 Speed control (Velocity)

8.1.1 Simple movement with modulo axis

The following application example illustrates how, in "Velocity" operating mode, you can use the user interface to easily move a modulo axis.

Control

Control the axis in the *Velocity* structure as follows:

- To activate the operating mode, set *xActivate* to "TRUE".
- Define the setpoints for *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk*.
- To start the operating mode, set *xStart* to "TRUE".

Interface in the IEC Editor

Velocity	SEW_MOS_UI_InterpolationModes.MC_UI_ModeVelocity	
IN	SEW_MOS_IInterpolationModes.ST_Velocity_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
lrVelocity	LREAL	60
lrAcceleration	LREAL	60
lrDeceleration	LREAL	60
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0

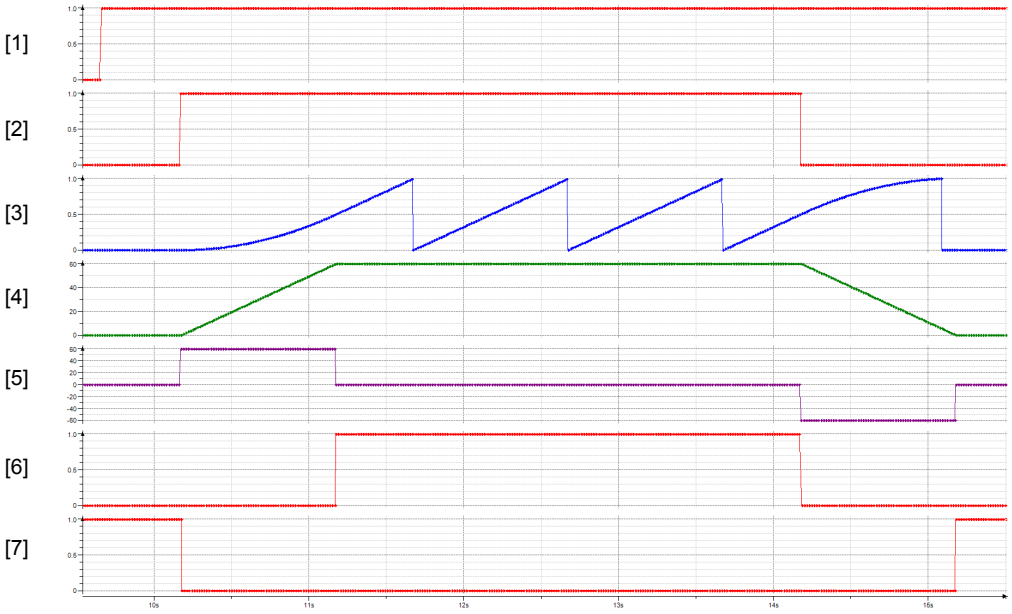
20729646091

Note:

When *xStart* has a value of "TRUE", the system accelerates with *lrAcceleration*. Once the target speed is reached, you receive the feedback message *xInVelocity* = *"TRUE"* in the *OUT* structure. When *xStart* has a value of "FALSE", the system decelerates with *lrDeceleration*. When the system is at standstill, the feedback message *xStopped* = "TRUE".

29179459/EN – 03/2020

Trace recording



25424762379

[1] SEW_GVL.Interface_Axis1.ProfileGeneration.Velocity.In.xActivate
[2] SEW_GVL.Interface_Axis1.ProfileGeneration.Velocity.In.xStart
[3] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[4] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[5] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[6] SEW_GVL.Interface_Axis1.ProfileGeneration.Velocity.Out.xInVelocity
[7] SEW_GVL.Interface_Axis1.ProfileGeneration.Velocity.Out.xStopped

8.1.2 Simple movement with linear axis

The following application example illustrates how, in "Velocity" operating mode, you can use the user interface to easily move a linear axis.

Control

Control the axis in the *Velocity* structure as follows:

- To activate the operating mode, set *xActivate* to "TRUE".
- Define the setpoints for *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk*.
- To start the operating mode, set *xStart* to "TRUE".

Interface in the IEC Editor

Velocity	SEW_MOS_UI_InterpolationModes.MC_UI_ModeVelocity	
IN	SEW_MOS_IInterpolationModes.ST_Velocity_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
lrVelocity	LREAL	60
lrAcceleration	LREAL	60
lrDeceleration	LREAL	60
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0

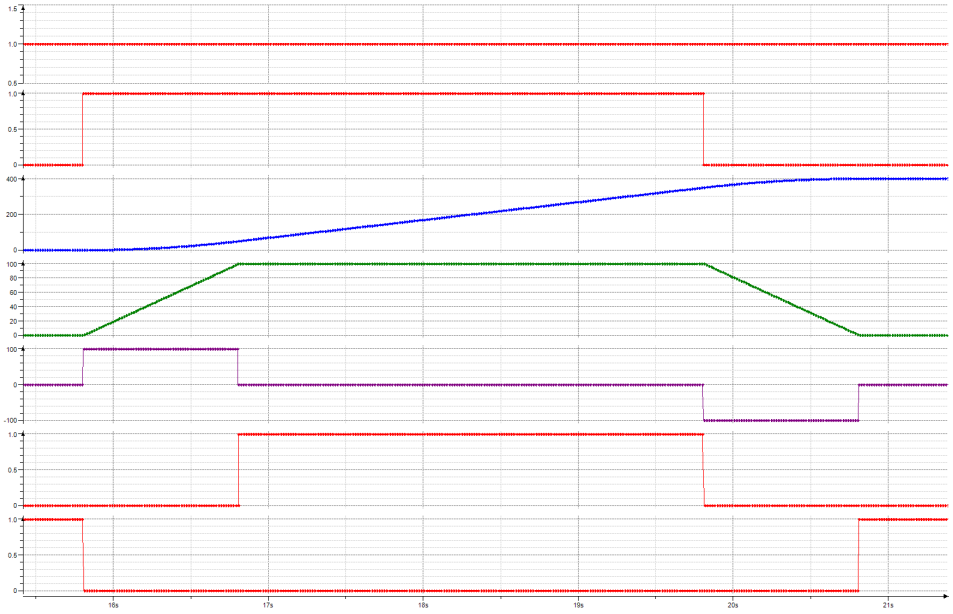
20848820363

Comment:

When *xStart* has a value of "TRUE", the system accelerates with *lrAcceleration*. When the target speed is reached, you will receive the feedback message *xInVelocity* = *"TRUE"* in the *OUT* structure. When *xStart* has a value of "FALSE", the system decelerates with *lrDeceleration*. When the system is in standstill, you will receive the feedback message *xStopped* = "TRUE".

29179459/EN – 03/2020

Trace recording



25424754699

[1] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.In.xActivate
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.In.xStart
[3] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[4] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[5] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[6] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.Out.xInVelocity
[7] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.Out.xStopped

29179459/EN – 03/2020

8.1.3 Stopping at an absolute position

The following application example illustrates how, in "Velocity" operating mode, you can control the profile generation such that the movement always stops at the same absolute position.

Control

Control the axis in the *Velocity* structure as follows:

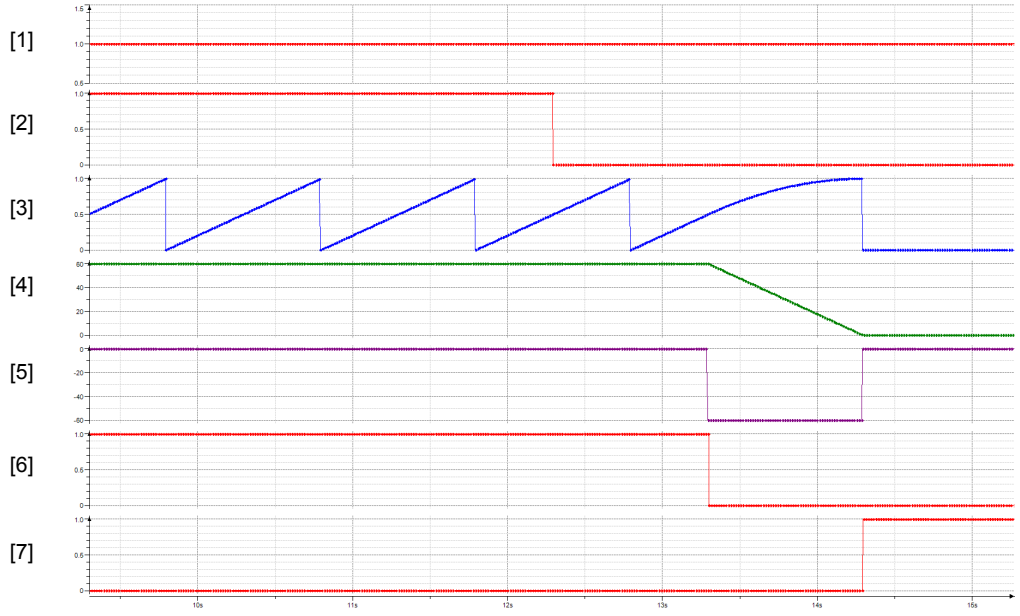
- Set *eModuloMode* to a modulo travel strategy of *MODULO_CW*.
- Set *eStopMode* to *STOPMODE_ABSOLUTE*.
- For *lrStopPosition*, enter the desired absolute stop position (in this example: 0).

Interface in the IEC Editor

Velocity	SEW_MOS_UI_InterpolationModes.MC_UI_ModeVelocity	
IN	SEW_MOS_IInterpolationModes.ST_Velocity_IN	
CONFIG	SEW_MOS_IInterpolationModes.ST_Velocity_Config	
eModuloMode	E_MODULOMODE	MODULO_CW
stPresetValues	ST_ProfGen_PresetValues	
ePresetMode	E_PRESETMODE	ACTPOS_MOVING
stStopAtPosition	ST_StopAtPosition	
eStopMode	E_STOPATPOSITIONMODE	STOPMODE_ABSOLUTE
lrStopPosition	LREAL	0

20730725643

Trace recording



25424770059

[1] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.In.xActivate
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.In.xStart
[3] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[4] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[5] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[6] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.Out.xInVelocity
[7] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.Out.xStopped

Comment:

The setpoint profile shows that the movement does not stop immediately with the falling edge at *xStart*. Deceleration is delayed so that the movement stops at the configured position, which in this case is the zero position.

8.1.4 Stopping at a relative position

The following application example illustrates how, in "Velocity" operating mode, you can control the profile generation such that the movement always stops at the same relative position based on the falling edge at *xStart*.

Control

Control the axis in the *Velocity* structure as follows:

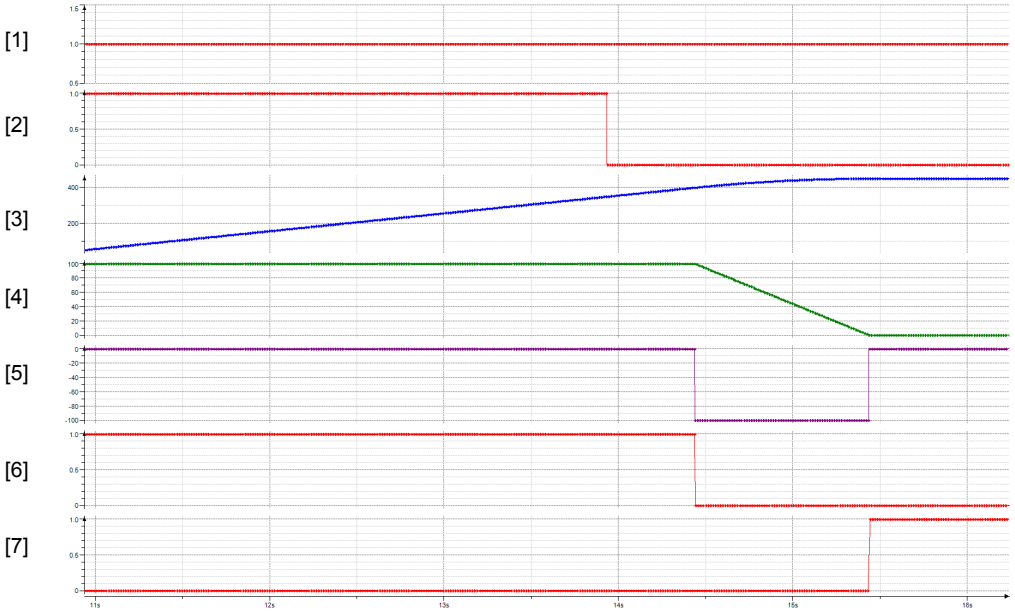
- Set *eModuloMode* to a modulo travel strategy of *MODULO_OFF*.
- Set *eStopMode* to *STOPMODE_RELATIVE*.
- For *lrStopPosition*, enter the desired relative stop position (in this example: 100 mm).

Interface in the IEC Editor

Velocity	SEW_MOS_UI_InterpolationModes.MC_UI_ModeVelocity	
IN	SEW_MOS_IInterpolationModes.ST_Velocity_IN	
CONFIG	SEW_MOS_IInterpolationModes.ST_Velocity_Config	
eModuloMode	E_MODULOMODE	MODULO_OFF
stPresetValues	ST_ProfGen_PresetValues	
ePresetMode	E_PRESETMODE	ACTPOS_MOVING
stStopAtPosition	ST_StopAtPosition	
eStopMode	E_STOPATPOSITIONMODE	STOPMODE_RELATIVE
lrStopPosition	LREAL	100

20731607819

Trace recording



25424777739

[1] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.In.xActivate
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.In.xStart
[3] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[4] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[5] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[6] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.Out.xInVelocity
[7] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.Out.xStopped

Comment:

The setpoint profile shows that the movement does not stop immediately with the falling edge at *xStart*. Deceleration is delayed so that the movement stops at the configured position; in this case, it is 100 mm after the falling edge at *xStart*.

29179459/EN – 03/2020

8.2 Absolute positioning (Positioning)

8.2.1 Simple positioning with linear axis

The following application example illustrates how, in "Positioning" operating mode, you can use the user interface to easily position a linear axis.

Control

Control the axis in the *Positioning* structure as follows:

- To activate the operating mode, set *xActivate* to "TRUE".
- Define the target position *lrPosition*.
- Define the setpoints for *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk*.
- To start the operating mode, set *xStart* to "TRUE".

Interface in the IEC Editor

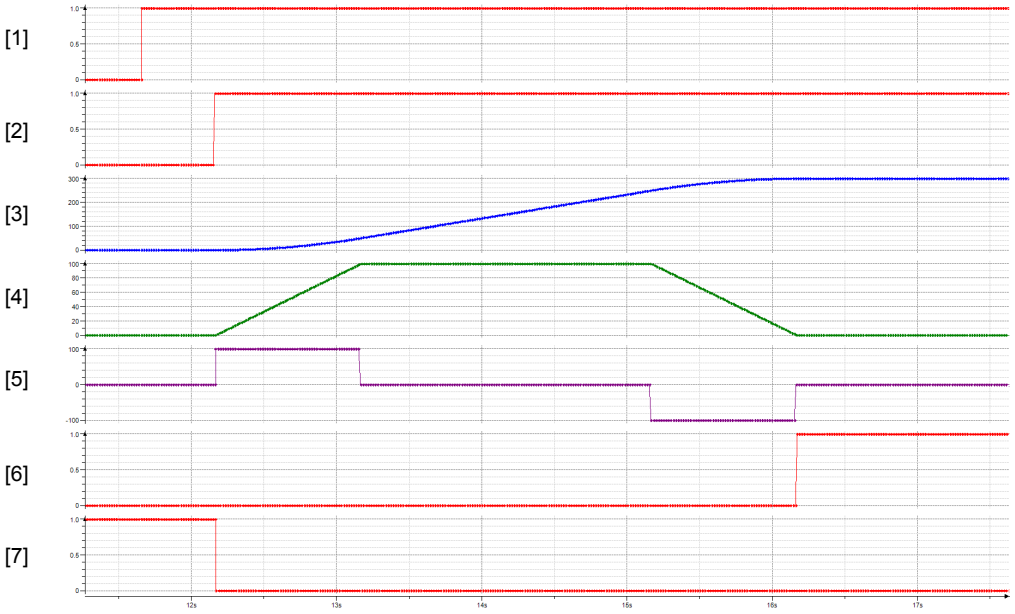
Positioning	SEW_MOS_UI_InterpolationModes.MC_UI_ModePositioning	
IN	SEW_MOS_IInterpolationModes.ST_Positioning_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
lrPosition	LREAL	500
lrVelocity	LREAL	100
lrAcceleration	LREAL	100
lrDeceleration	LREAL	100
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0

20735752971

Comment:

As soon as the setpoint profile has reached the target position, you will receive the feedback message *xInPosition = "TRUE"* in the *OUT* structure.

Trace recording



25424644619

[1] SEW_GVL.Interface_Axis2.ProfileGeneration.Positioning.In.xActivate
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Positioning.In.xStart
[3] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[4] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[5] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[6] SEW_GVL.Interface_Axis2.ProfileGeneration.Positioning.Out.xInPosition
[7] SEW_GVL.Interface_Axis2.ProfileGeneration.Positioning.Out.xStopped

8.2.2 Simple positioning with modulo axis

The following application example illustrates how, in "Positioning" operating mode, you can use the user interface to easily position a modulo axis.

Control

Control the axis in the *Positioning* structure as follows:

- To activate the operating mode, set *xActivate* to "TRUE".
- Define the setpoints for*lrPosition*, *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk*.
- Set *eModuloMode* to *MODULO_CW* in the *CONFIG* structure.
- To start the operating mode, set *xStart* to "TRUE".

29179459/EN – 03/2020

Interface in the IEC Editor

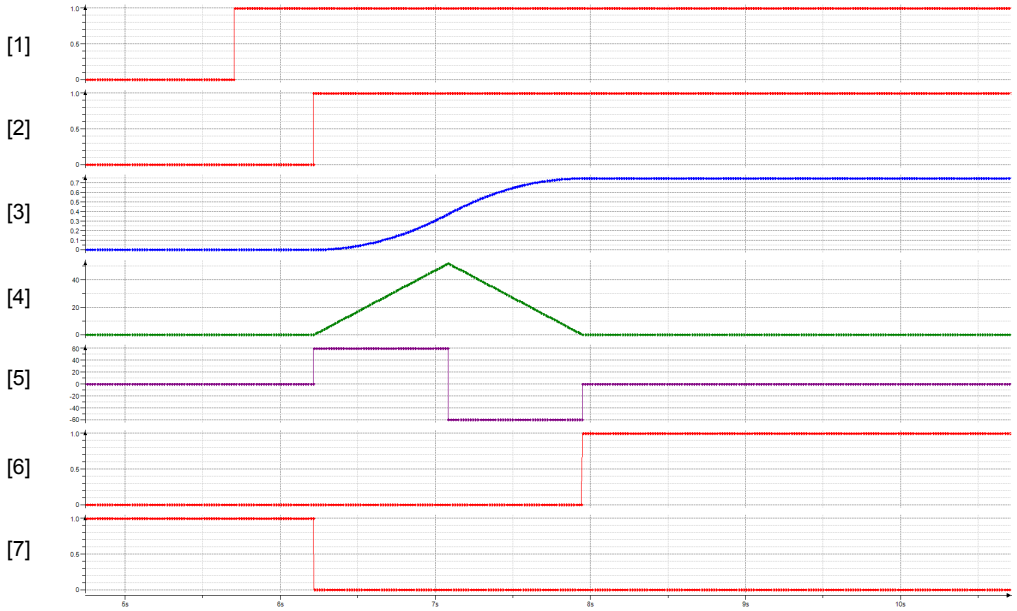
Positioning	SEW_MOS_UI_InterpolationModes.MC_UI_ModePositioning	
IN	SEW_MOS_IInterpolationModes.ST_Positioning_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
lrPosition	LREAL	1
lrVelocity	LREAL	60
lrAcceleration	LREAL	60
lrDeceleration	LREAL	60
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0
CONFIG	SEW_MOS_IInterpolationModes.ST_Positioning_Config	
eModuloMode	E_MODULOMODE	MODULO_CW
stPresetValues	ST_ProfGen_PresetValues	
ePresetMode	E_PRESETMODE	ACTPOS_MOVING
xTargetPositionMonitoring	BOOL	FALSE
xWithoutReferencedEncoder	BOOL	FALSE

20735776267

Comment:

As soon as the setpoint profile has reached the target position, you will receive the feedback message *xInPosition* = *"TRUE"* in the *OUT* structure.

Trace recording



25424665099

[1] SEW_GVL.Interface_Axis1.ProfileGeneration.Positioning.In.xActivate
[2] SEW_GVL.Interface_Axis1.ProfileGeneration.Positioning.In.xStart
[3] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[4] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[5] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[6] SEW_GVL.Interface_Axis1.ProfileGeneration.Positioning.Out.xInPosition
[7] SEW_GVL.Interface_Axis1.ProfileGeneration.Positioning.Out.xStopped

Comment:

The modulo mode is applied with a rising edge at *xStart*, i.e. not during ongoing movement.

8.3 Relative positioning (PositioningRelative)

8.3.1 Relative positioning in operating mode DistanceChangeOFF

The following application example illustrates how, in "DistanceChangeOFF" operating mode, you can use the user interface for relative positioning of the axis.

In "DistanceChangeOFF" operating mode, a change to *lrDistance* is ignored. The target position results from the sum of the start position and *lrDistance* at the time of the rising edge at *xStart*.

Control

Control the axis in the *PositioningRelative* structure as follows:

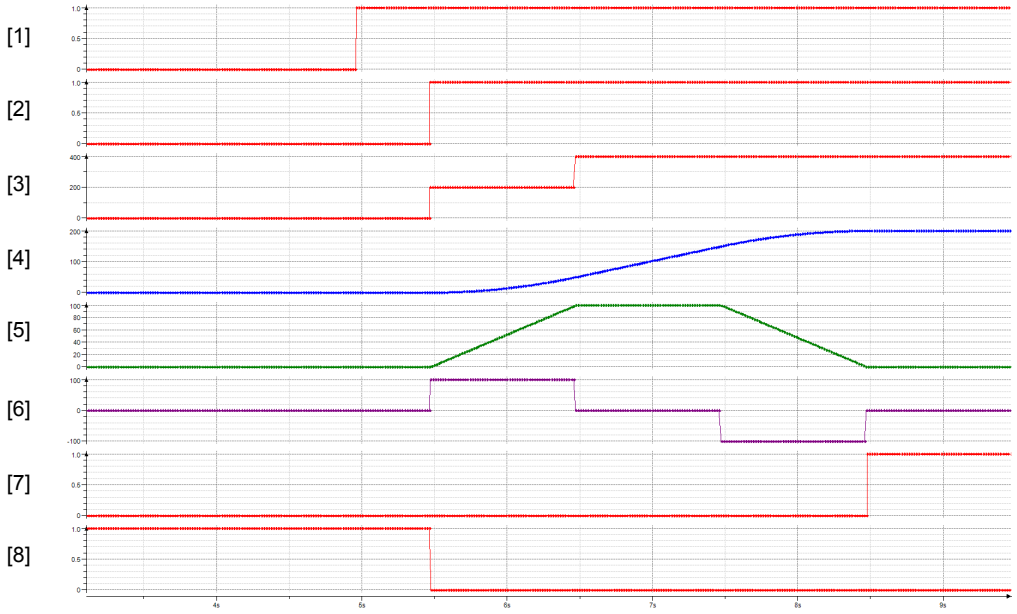
- To activate the operating mode, set *xActivate* to "TRUE".
- Define the setpoints for *lrDistance*, *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk*.
- Set *eMode* to *DistanceChangeOFF* in the *CONFIG* structure.
- To start the operating mode, set *xStart* to "TRUE".

Interface in the IEC Editor

PositioningRelative	SEW_MOS_UI_Inter...	
IN	SEW_MOS_IInterpol...	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
xAcceptNewDistance	BOOL	FALSE
lrDistance	LREAL	100
lrVelocity	LREAL	100
lrAcceleration	LREAL	100
lrDeceleration	LREAL	100
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0
CONFIG	SEW_MOS_IInterpol...	
eModuloMode	E_MODULOMODE	MODULO_OFF
stPresetValues	ST_ProfGen_Preset...	
ePresetMode	E_PRESETMODE	ACTPOS_STANDSTILL
eMode	E_POSRELATIVEMODE	DistanceChangeOFF
xTargetPositionMonitoring	BOOL	FALSE
xContinueRelativeMove	BOOL	FALSE

20853338763

Trace recording



25424695819

[1] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.xActivate
[2] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.xStart
[3] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.lrDistance
[4] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[5] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[6] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[7] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.Out.xInPosition
[8] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.Out.xStopped

Comment:

During ongoing movement, *lrDistance* is changed from 200.0 to 400.0. The change is ignored, and the movement stops at 200.0.

8.3.2 Relative positioning in operating mode StartPositionBased

The following application example illustrates how, in "StartPositionBased" operating mode, you can use the user interface for relative positioning of an axis.

A change to *lrDistance* is applied. The new target position results from the start position plus *lrDistance*. A change to *lrDistance* is applied as long as *xStart* is present.

Control

Control the axis in the *PositioningRelative* structure as follows:

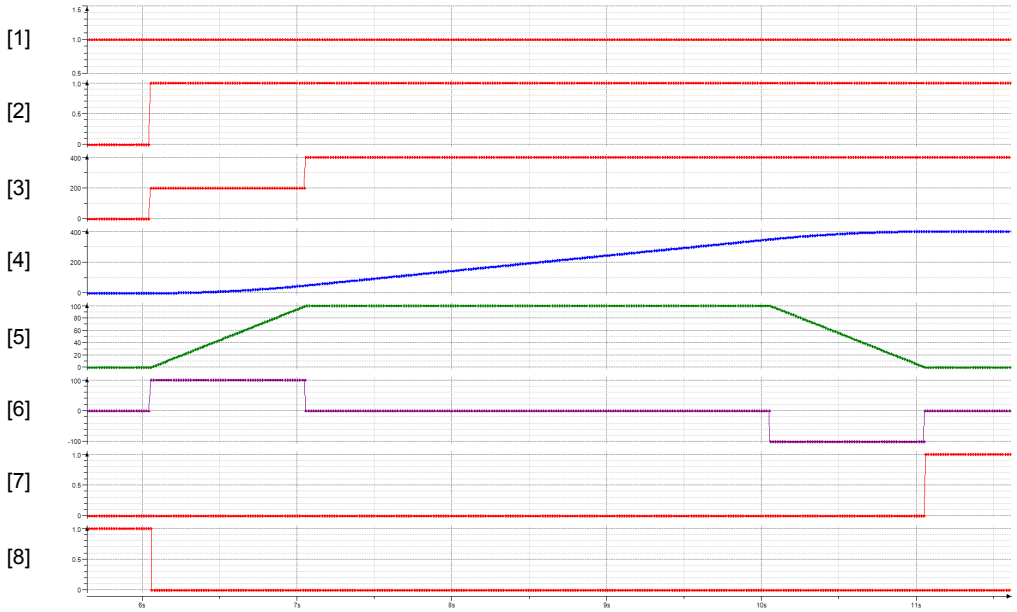
- To activate the operating mode, set *xActivate* to "TRUE".
- Define the setpoints for *lrDistance*, *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk*.
- Set *eMode* to *StartPositionBased* in the *CONFIG* structure.
- To start the operating mode, set *xStart* to "TRUE".

Interface in the IEC Editor

PositioningRelative	SEW_MOS_UI_Inter...		
IN	SEW_MOS_IInterpol...		
xActivate	BOOL	TRUE	
xStart	BOOL	TRUE	
xAcceptNewDistance	BOOL	FALSE	
lrDistance	LREAL	100	150
lrVelocity	LREAL	100	
lrAcceleration	LREAL	100	
lrDeceleration	LREAL	100	
lrJerk	LREAL	0	
xRapidStop	BOOL	FALSE	
lrRapidStopDeceleration	LREAL	0	
CONFIG	SEW_MOS_IInterpol...		
eModuloMode	E_MODULOMODE	MODULO_OFF	
stPresetValues	ST_ProfGen_Preset...		
ePresetMode	E_PRESETMODE	ACTPOS_STANDSTILL	
eMode	E_POSRELATIVEMODE	StartPositionBased	
xTargetPositionMonitoring	BOOL	FALSE	
xContinueRelativeMove	BOOL	FALSE	

20853341963

Trace recording



25424703499

[1] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.xActivate
[2] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.xStart
[3] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.lrDistance
[4] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[5] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[6] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[7] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.Out.xInPosition
[8] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.Out.xStopped

Comment:

During ongoing movement, *lrDistance* is changed from 200.0 to 400.0. The change is in relation to the start position; the movement stops at 400.0.

8.3.3 Relative positioning in operating mode ActualPositionBased

The following application example illustrates how, in "ActualPositionBased" operating mode, you can use the user interface for relative positioning of an axis.

A change to *lrDistance* is applied. The new target position is calculated based on the current position of the profile generation at the time of the change plus *lrDistance*.

Control

Control the axis in the *PositioningRelative* structure as follows:

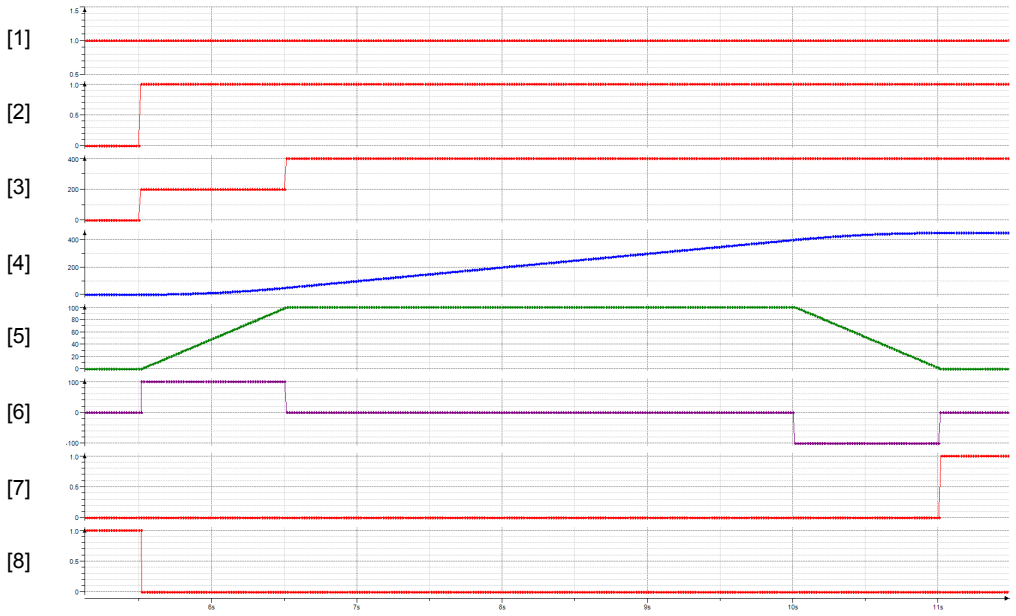
- To activate the operating mode, set *xActivate* to "TRUE".
- Define the setpoints for *lrDistance*, *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk*.
- Set *eMode* to *ActualPositionBased* in the *CONFIG* structure.
- To start the operating mode, set *xStart* to "TRUE".

Interface in the IEC Editor

PositioningRelative	SEW_MOS_UI_Inter...		
IN	SEW_MOS_IInterpol...		
xActivate	BOOL	TRUE	
xStart	BOOL	TRUE	
xAcceptNewDistance	BOOL	FALSE	
lrDistance	LREAL	100	150
lrVelocity	LREAL	100	
lrAcceleration	LREAL	100	
lrDeceleration	LREAL	100	
lrJerk	LREAL	0	
xRapidStop	BOOL	FALSE	
lrRapidStopDeceleration	LREAL	0	
CONFIG	SEW_MOS_IInterpol...		
eModuloMode	E_MODULOMODE	MODULO_OFF	
stPresetValues	ST_ProfGen_Preset...		
ePresetMode	E_PRESETMODE	ACTPOS_STANDSTILL	
eMode	E_POSRELATIVEMODE	ActualPositionBased	
xTargetPositionMonitoring	BOOL	FALSE	
xContinueRelativeMove	BOOL	FALSE	

20853348363

Trace recording



25424688139

[1] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.xActivate
[2] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.xStart
[3] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.lrDistance
[4] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[5] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[6] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[7] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.Out.xInPosition
[8] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.Out.xStopped

Comment: During ongoing movement, *lrDistance* is changed from 200.0 to 400.0. The change is in relation to the position at the time of the change (in this example: 50.0); the movement stops at 450.0.

8.3.4 Relative positioning in operating mode TargetPositionBased

The following application example illustrates how, in "TargetPositionBased" operating mode, you can use the user interface for relative positioning of an axis.

lrDistance is added to the original target position with the rising edge at *xAcceptNewDistance*. The advantage of this operating mode is that the target position can be changed without changing *lrDistance*.

Control

Control the axis in the *PositioningRelative* structure as follows:

- To activate the operating mode, set *xActivate* to "TRUE".
- Define the setpoints for *lrDistance*, *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk*.
- Set *eMode* to *TargetPositionBased* in the *CONFIG* structure.
- To start the operating mode, set *xStart* to "TRUE".

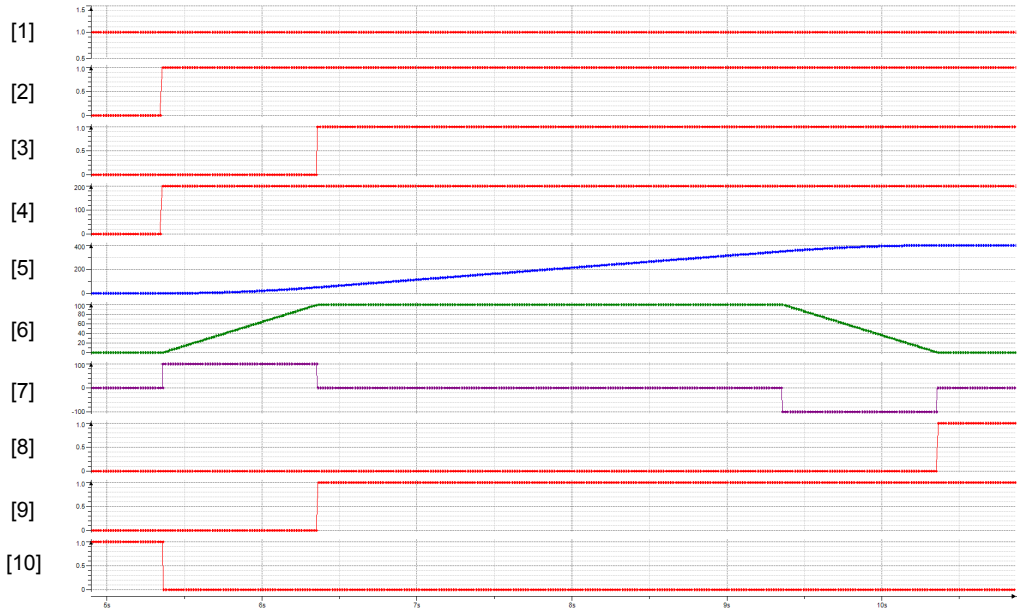
29179459/EN – 03/2020

Interface in the IEC Editor

PositioningRelative	SEW_MOS_UI_Inter...		
IN	SEW_MOS_IInterpol...		
xActivate	BOOL	TRUE	
xStart	BOOL	TRUE	
xAcceptNewDistance	BOOL	FALSE	TRUE
lrDistance	LREAL	100	30
lrVelocity	LREAL	100	
lrAcceleration	LREAL	100	
lrDeceleration	LREAL	100	
lrJerk	LREAL	0	
xRapidStop	BOOL	FALSE	
lrRapidStopDeceleration	LREAL	0	
CONFIG	SEW_MOS_IInterpol...		
eModuloMode	E_MODULOMODE	MODULO_OFF	
stPresetValues	ST_ProfGen_Preset...		
ePresetMode	E_PRESETMODE	ACTPOS_STANDSTILL	
eMode	E_POSRELATIVEMODE	TargetPositionBased	
xTargetPositionMonitoring	BOOL	FALSE	
xContinueRelativeMove	BOOL	FALSE	

20853345163

Trace recording



25424711179

[1] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.xActivate
[2] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.xStart
[3] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.xAccept-NewDistance
[4] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.In.lrDistance
[5] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpola-tion.stSetpointsUserUnits.lrPosition
[6] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpola-tion.stSetpointsUserUnits.lrVelocity
[7] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpola-tion.stSetpointsUserUnits.lrAcceleration
[8] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.Out.xInPosi-tion
[9] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.Out.xNewDis-tanceAccepted
[10] SEW_GVL.Interface_Axis1.ProfileGeneration.PositioningRelative.Out.xStopped

Comment:

During ongoing movement and with the rising edge of *xAcceptNewDistance*, the value of *lrDistance* (in this example: "200.0") is added to the original target position (in this example: "200.0"). The movement stops at 400.0. *xNewDistanceAcccepted* = "TRUE" indicates that the new target position has been applied.

29179459/EN – 03/2020

8.4 Reference travel (Homing)

8.4.1 Referencing with homing

The following application example illustrates how you can configure referencing with homing.

Control

Control the axis in the *Homing* structure as follows:

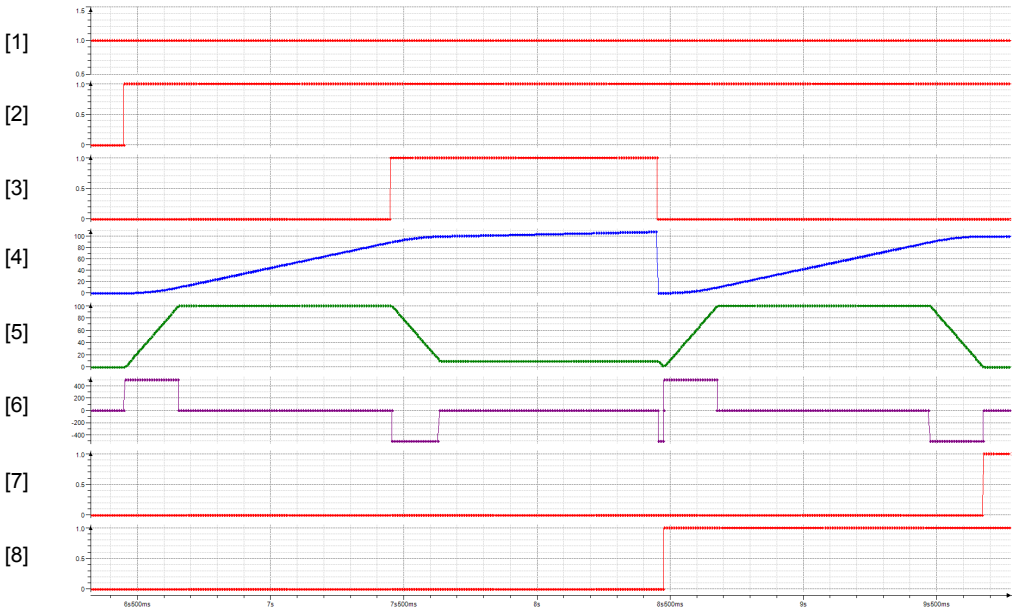
- To activate the operating mode, set *xActivate* to "TRUE".
- Define search speed *lrSearchVelocity* in the *CONFIG* structure.
- In the *stStartPositionMove* structure, define acceleration *lrAcceleration* to which the search speed is accelerated.
- Define retraction speed *lrClearVelocity* in the *CONFIG* structure.
- In the *stStartPositionMove* structure, define deceleration *lrDeceleration* with which the search speed is decelerated to the retraction speed.
- To start the operating mode, set *xStart* to "TRUE".

Interface in the IEC Editor

Homing	SEW_MOS_UI_InterpolationModes.MC_UI_ModeHoming	
IN	SEW_MOS_IInterpolationModes.ST_Homing_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
lrReferenceOffset	LREAL	0
xLimitSwitchLeft	BOOL	FALSE
xLimitSwitchRight	BOOL	TRUE
xReferenceCam	BOOL	TRUE
stStartPositionMove	ST_StartPositionMove	
xMoveToStartPosition	BOOL	FALSE
lrStartPosition	LREAL	0
lrVelocity	LREAL	0
lrAcceleration	LREAL	100
lrDeceleration	LREAL	100
lrJerk	LREAL	0
CONFIG	SEW_MOS_IInterpolationModes.ST_Homing_Config	
eReferenceTravelType	E_REFSTRATEGY	Deactivated
lrLimitSwitchDebouncingTime	LREAL	0
lrSearchVelocity	LREAL	100
lrClearVelocity	LREAL	500

20853377163

Trace recording



25429411467

[1] SEW_GVL.Interface_Axis1.ProfileGeneration.Homing.In.xActivate
[2] SEW_GVL.Interface_Axis1.ProfileGeneration.Homing.In.xStart
[3] SEW_GVL.Interface_Axis1.ProfileGeneration.Homing.In.xReferenceCam
[4] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[5] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[6] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[7] SEW_GVL.Interface_Axis1.ProfileGeneration. Homing.Out.xInPosition
[8] SEW_GVL.Interface_Axis1. Homing.PositioningRelative.Out.xDone

Comment:

The acceleration set under *stStartPositionMove.lrAcceleration* is used for accelerating the system to the search speed defined under *CONFIG.lrSearchVelocity*.

When the cam is detected at *IN.xReferenceCam*, the movement decelerates as specified under *stStartPositionMove.lrDeceleration* until retraction speed *CONFIG.lrClearVelocity* is reached.

The movement stops when the inverter has moved clear of the cam.

Homing then follows on from standstill.

INFORMATION

A modulo mode is not available for homing.

29179459/EN – 03/2020

8.4.2 Referencing on the fly

The following application example illustrates how you can perform referencing during ongoing operation. Referencing on the fly can be useful for endlessly turning axes without mechanical intervention, such as simple conveyor belts or rotary tables. In both cases it is useful to define a position zero point with reference to a product. The reference point could be the front edge of a product or a label on the product, for example.

Control

For example, control the axis in the *Homing* structure as follows:

Interface in the IEC Editor

ProfileGeneration	SEW_MuMoCamAx....	
In	SEW_IInterpolMode...	
Config	SEW_IInterpolMode...	
Out	SEW_IInterpolMode...	
Velocity	SEW_UIInterpolMod...	
In	SEW_IInterpolMode...	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
lrVelocity	LREAL	100
lrAcceleration	LREAL	100
lrDeceleration	LREAL	100
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0
Config	SEW_IInterpolMode...	
Out	SEW_IInterpolMode...	
Positioning	SEW_UIInterpolMod...	
PositioningRelative	SEW_UIInterpolMod...	
Jog	SEW_UIInterpolMod...	
Tracking	SEW_UIInterpolMod...	
Homing	SEW_UIInterpolMod...	
In	SEW_IInterpolMode...	
xActivate	BOOL	TRUE
xStart	BOOL	FALSE
xLimitSwitchNegative	BOOL	FALSE
xLimitSwitchPositive	BOOL	FALSE
xReferenceCam	BOOL	FALSE
Config	SEW_IInterpolMode...	
eReferenceTravelType	E_REFSTRATEGY	ReferencingWithoutRefTravel
lrReferenceOffset	LREAL	0

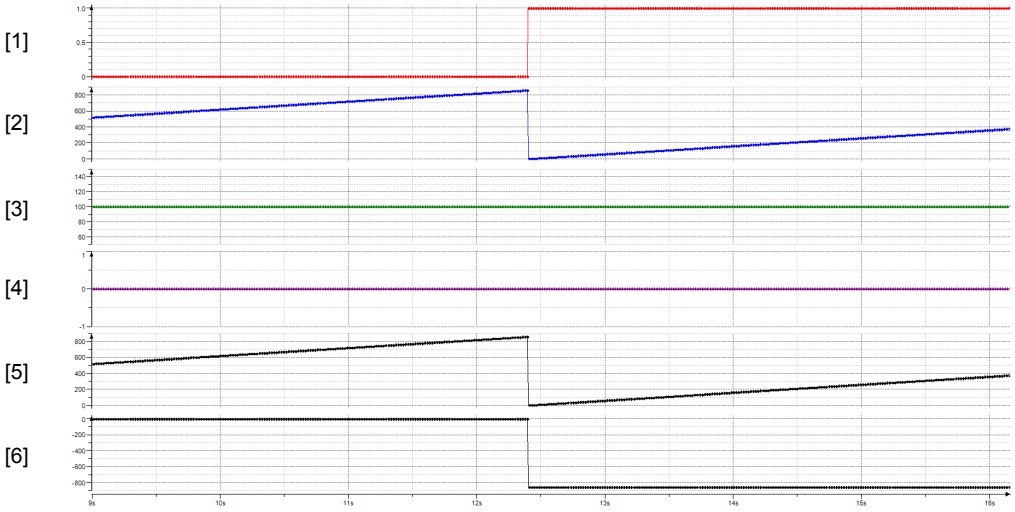
29002464139

Explanation:

The axis is enabled and moves at the specified speed in *Velocity* operating mode. The *Homing* operating mode is active and the required reference travel type and reference offset have been set (in this example: 0). With the respective event, such as the front edge of a product in a light barrier, *xStart* is set in the *Homing* operating mode.

The following trace recording shows this behavior: With the rising edge at *xStart*, the current position value is deducted from the position offset. The setpoint position of profile generation jumps to the value 0 without a change in the speed and acceleration profile. The position of profile generation now indicates the distance of the front edge of the product and the light barrier.

Trace recording



29006590603

[1] SEW_GVL.Interface_Axis1.ProfileGeneration.Homing.In.xStart
[2] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[3] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[4] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[5] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.lrActualMotionPositionUserUnit
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.lrActualMotionOffsetUserUnit

29179459/EN – 03/2020

8.5 Direct coupling (Tracking)

8.5.1 Tracking with configured axis as master source

The following application example illustrates how you can configure the "Tracking" operating mode using a configured axis as the master source. In this example, axis 2 follows axis 1. In "Velocity" operating mode, axis 1 moves at 60 cycles per minute. The proportional factor is set to 100 mm/cycle.

Control

Control the axis in the *Velocity* structure as follows:

- To activate the operating mode, set *xActivate* to "TRUE".
- Define a speed of 60 cycles per minute by setting *lrVelocity*, *lrAcceleration*, and*lrDeceleration* to *60*.
- To start the operating mode, set *xStart* to "TRUE".

Control the axis in the *Tracking* structure as follows:

- To activate the operating mode, set *xActivate* to "TRUE".
- Define a proportional factor of 100 mm/cycle by setting *diTrackingNumerator* to *100* and *diTrackingDenominator* to *1*.
- To start the operating mode, set *xStart* to "TRUE".

Interface in the IEC Editor

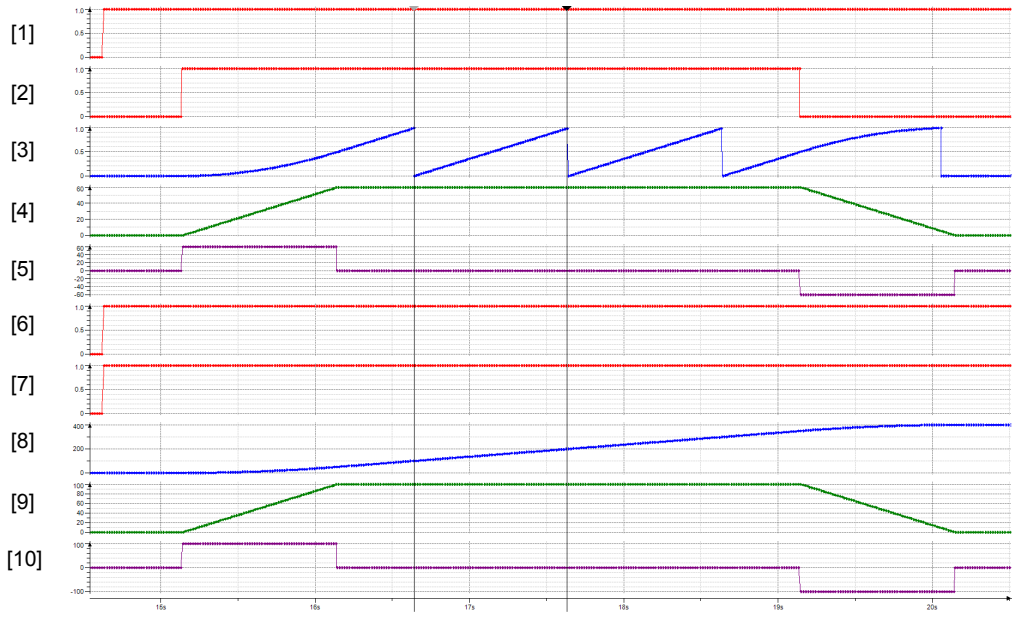
Velocity	SEW_MOS_UI_InterpolationModes.MC_UI_ModeVelocity	
IN	SEW_MOS_IInterpolationModes.ST_Velocity_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
lrVelocity	LREAL	60
lrAcceleration	LREAL	60
lrDeceleration	LREAL	60
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0

Control for axis 1 (master axis) *20738516491*

Tracking	SEW_MOS_UI_InterpolationModes.MC_UI_ModeTracking	
IN	SEW_MOS_IInterpolationModes.ST_Tracking_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
stMasterSetpoints	SEW_MOS_Master.ST_MasterSetpointValues	
diTrackingNumerator	DINT	100
diTrackingDenominator	DINT	1

Control for axis 2 (slave axis) *20738519563*

Trace recording



25424726539

[1] SEW_GVL.Interface_Axis1.ProfileGeneration.Velocity.In.xActivate
[2] SEW_GVL.Interface_Axis1.ProfileGeneration.Velocity.In.xStart
[3] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[4] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[5] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[6] SEW_GVL.Interface_Axis2.ProfileGeneration.Tracking.In.xActivate
[7] SEW_GVL.Interface_Axis2.ProfileGeneration.Tracking.In.xStart
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[9] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[10] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration

Comment:

- The measurement with Cursor1 and Cursor2 shows that the slave axis moves exactly 100 mm, while the master axis moves 1 cycle.
- The master axis moves at 60 cycles per minute. The slave axis therefore moves at 100 mm/s.
- The master axis accelerates at a rate of 60 cycles per minute per second. It reaches the speed after 1 second; therefore, the slave axis accelerates at a rate of 100 mm/s/s.

29179459/EN – 03/2020

8.6 Electronic cam (Camming)

To execute this application example, you need the MOVIKIT® MultiMotion Camming license.

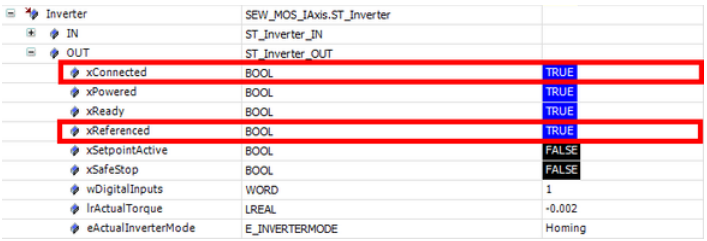
8.6.1 Basic procedure

The following step-by-step instructions describe the basic procedure for using "Camming" operating mode:

✓ The desired curve descriptions have been created using the CamEditor and transferred to the memory card (directory "\DATA\Camming\AxisName") of the MOVI-C® CONTROLLER (CurveConfig.xml – list of all curve files; *.camCurve – curve files in which curve profiles are defined per segment).

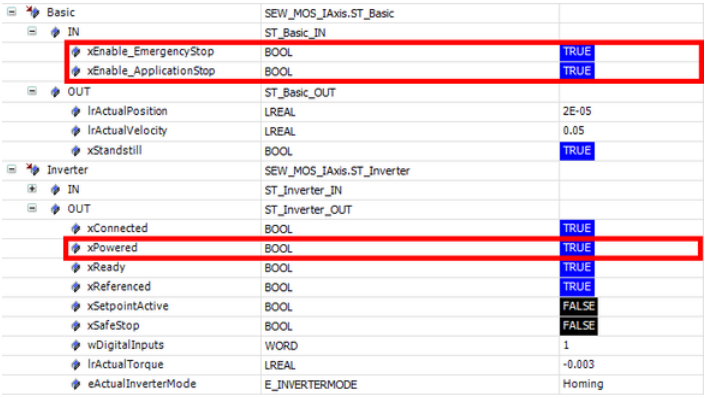
1. Check if the axes are connected, referenced, and without error.

9007219942623883



9007224181537675

2. Enable the axes.



20687885963

3. Import the `CurveConfig.xml` file using *xReadCurveFileList* = *"TRUE"*.
   ⇨ *xCurveFileListRead* reports back that the file has been imported successfully.
   ⇨ The curve description files listed in the `CurveConfig.xml` file are displayed in the *CurveFileList* structure. This is also where the *uiNumberOfLastCurveFile* variable can be found that indicates how many curve description files have been configured.

FileHandling	FB_UI_CamFileHandler	
IN	SEW_MOS_ICamming.ST_CamFileHandler_IN	
xReadCurveFileList	BOOL	TRUE
xReadCurveFile	BOOL	FALSE
sFileName	STRING(80)	''
uiCurveNumber	UINT	0
CurveFileList	FB_CurveFileList	
stCurveFileList	SEW_MOS_ICamming.ST_CurveFileList	
uiNumberOfLastCurveFile	UINT	3
sCurveFileName	ARRAY [1..gc_iMaxNumberOfCurveFiles] OF STRING	
sCurveFileName[1]	STRING	'DoublePolynom5.camCurve'
sCurveFileName[2]	STRING	'Cosinus.camCurve'
sCurveFileName[3]	STRING	'Sinus.camCurve'
sCurveFileName[4]	STRING	''
sCurveFileName[20]	STRING	''
OUT	SEW_MOS_ICamming.ST_CamFileHandler_OUT	
xCurveFileListRead	BOOL	TRUE

20739422091

4. Select curve description file *uiCurveNumber* from the list and import the file using *xReadCurveFile = "TRUE"*.
   ⇨ *xCurveFileRead* reports back that the list has been imported successfully.
   ⇨ The name of the last imported file is displayed under *sLastReadCurveFile*.

FileHandling	FB_UI_CamFileHandler	
IN	SEW_MOS_ICamming.ST_CamFileHandler_IN	
xReadCurveFileList	BOOL	TRUE
xReadCurveFile	BOOL	TRUE
sFileName	STRING(80)	''
uiCurveNumber	UINT	1
CurveFileList	FB_CurveFileList	
stCurveFileList	SEW_MOS_ICamming.ST_CurveFileList	
uiNumberOfLastCurveFile	UINT	3
sCurveFileName	ARRAY [1..gc_iMaxNumberOfCurveFiles] OF STRING	
OUT	SEW_MOS_ICamming.ST_CamFileHandler_OUT	
xCurveFileListRead	BOOL	TRUE
xCurveFileRead	BOOL	TRUE
sLastReadCurveFile	STRING(80)	'Config/Axis2/DoublePolynom5.camCurve
xFBError	BOOL	FALSE
udiFBErrorID	UDINT	0

20739591563

5. Using *IN.xActivate = "TRUE"*, activate the "Camming" operating mode.

   ⇨ The offset calculated between the current position of the slave axis and the curve position is displayed under *Adjust.OUT.lrAdjustValue*.

Camming	SEW_MOS_Camming.MC_UI_ModeCamming	
IN	SEW_MOS_ICamming.ST_Camming_IN	
xActivate	BOOL	TRUE
xStart	BOOL	FALSE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
CONFIG	SEW_MOS_ICamming.ST_Camming_CONFIG	
FileHandling	FB_UI_CamFileHandler	
XOffsetCorrection	FB_UI_OffsetCorrection	
YOffsetCorrection	FB_UI_OffsetCorrection	
Adjust	FB_UI_AdjustProfilgenerator	
IN	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_IN	
xStart	BOOL	FALSE
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0
CONFIG	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_CONFIG	
eModuloMode	E_MODULOMODE	MODULO_OFF
OUT	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_OUT	
xAdjusted	BOOL	FALSE
lrAdjustValue	LREAL	388.5
eActiveModuloMode	E_MODULOMODE	MODULO_OFF

20741640203

6. Specify speed *Adjust.IN.lrVelocity* and acceleration *Adjust.IN.lrAcceleration*.

Adjust	FB_UI_AdjustProfilgenerator	
IN	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_IN	
xStart	BOOL	FALSE
lrVelocity	LREAL	0
lrAcceleration	LREAL	0
lrDeceleration	LREAL	0
lrJerk	LREAL	0

20880407435

7. Set *Adjust.IN.xStart* to "TRUE".

   ⇨ The axis is aligned with the curve position.

   ⇨ When *lrAdjustValue* has moved to *0*, the feedback message *xAdjusted = "TRUE"* is displayed.

Adjust	FB_UI_AdjustProfilgenerator	
IN	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_IN	
xStart	BOOL	TRUE
lrVelocity	LREAL	100
lrAcceleration	LREAL	100
lrDeceleration	LREAL	100
lrJerk	LREAL	0
CONFIG	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_CONFIG	
eModuloMode	E_MODULOMODE	MODULO_OFF
OUT	SEW_MOS_ICamming.ST_AdjustProfilgeneratorUI_OUT	
xAdjusted	BOOL	TRUE
lrAdjustValue	LREAL	0
eActiveModuloMode	E_MODULOMODE	MODULO_OFF

20741643275

8. Start "Camming" operating mode.

Camming	SEW_MOS_Camming.MC_UI_ModeCamming	
IN	SEW_MOS_ICamming.ST_Camming_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
CONFIG	SEW_MOS_ICamming.ST_Camming_CONFIG	
FileHandling	FB_UI_CamFileHandler	
XOffsetCorrection	FB_UI_OffsetCorrection	
YOffsetCorrection	FB_UI_OffsetCorrection	
Adjust	FB_UI_AdjustProfilgenerator	
OUT	SEW_MOS_ICamming.ST_Camming_OUT	
xModeActive	BOOL	TRUE
xProfileActive	BOOL	TRUE
eGearingState	E_GEARING_STATE	ACTIVE
lrSetpPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xFBError	BOOL	FALSE
udiFBErrorID	UDINT	0
stSetpointMasterCam	SEW_MOS_IInterpolationModes.ST_InterpolatedValues_UserUnits	
xSetMasterValueDone	BOOL	FALSE
xSetSlaveValueDone	BOOL	FALSE

20688221323

⇨ When starting/stopping, the parameters set in the *CONFIG* structure for synchronization/desynchronization take effect and offer a wide variety of different options. These options are discussed in more detail below.

8.6.2 Synchronizing/desynchronizing with stopped master axis

Synchronizing the slave axis

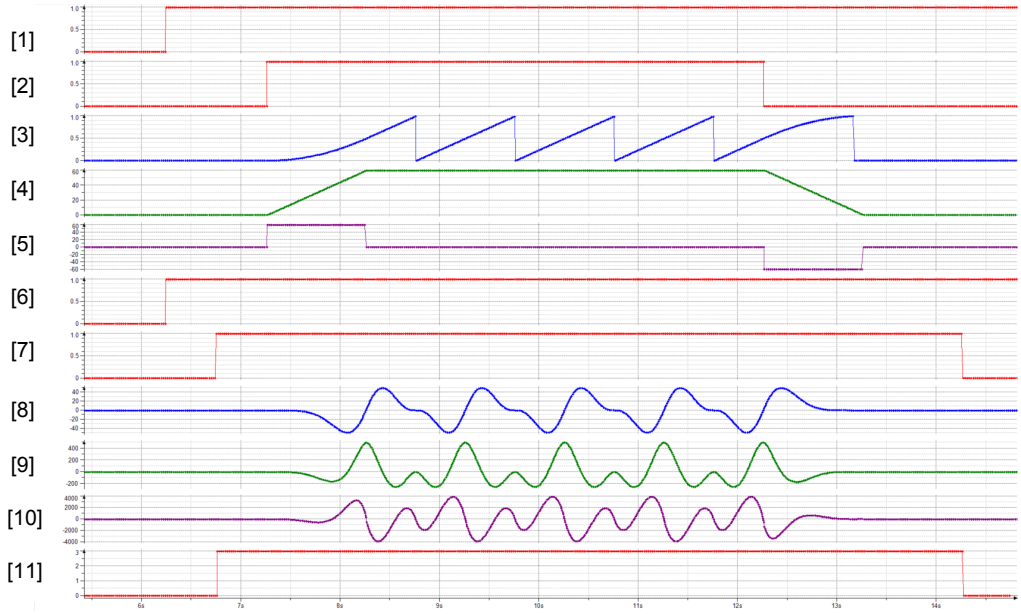
1. Set *IN.xStart* to "TRUE".
   - ⇨ If an appropriate setting (e.g. *eStartMode* = *CW_CCW_DIRECT*) is configured in the *CONFIG* structure, the slave axis is synchronized immediately.
   - ⇨ *xProfileActive* reports back "TRUE".
   - ⇨ *eGearingState* reports back "ACTIVE".

Camming	SEW_Cam.ModeCamming_UI	
IN	SEW_ICam.ST_Camming_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
xGetNewCamDescription	BOOL	FALSE
CONFIG	SEW_ICam.ST_Camming_CONFIG2	
Start	ST_CammingStart_CONFIG	
lrGearInDistance	LREAL	0
lrGearInOffset	LREAL	0
lrGearInReferencePosition	LREAL	0
eStartMode	E_CAMMINGSTARTMODE	CW_DIRECT
xStartWithLastXCorrection	BOOL	FALSE
Stop	ST_CammingStop_CONFIG	
Change	ST_CammingChange_CONFIG	
eCycleMode	E_CYCLEMODE	ADJUST_SLAVE_REFERENCE_POSITION
FileHandling	CamFileHandler_UI	
XOffsetCorrection	OffsetCorrection_UI	
YOffsetCorrection	OffsetCorrection_UI	
Adjust	AdjustProfilgenerator_UI	
OUT	SEW_ICam.ST_Camming_OUT	
xModeActive	BOOL	TRUE
xProfileActive	BOOL	TRUE
eGearingState	E_GEARING_STATE	ACTIVE
lrSetpPosition	LREAL	0
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
stSetpointMasterCam	SEW_IInterpolModes.ST_InterpolatedValues_UserUnits	
xSetMasterValueDone	BOOL	FALSE
xSetSlaveValueDone	BOOL	FALSE
xGetNewCamDescriptionDone	BOOL	FALSE

9007219996588555

Starting/stopping the master axis

1. Move the master axis using "Velocity" operating mode. For this purpose, activate this operating mode using *xActivate* = "TRUE". Next, specify speed and acceleration (e.g. *lrVelocity* = 60.0, etc.) and start the movement using *xStart* = "TRUE".
   - ⇨ Following *xStart* = "TRUE" in "Camming" operating mode of the slave axis (Axis2), you will receive the feedback message *eGearingState* = "3" (ACTIVE). After that, you can start the master axis using *xStart* = "TRUE" in "Velocity" operating mode, and stop it using *xStart* = "FALSE". When *xStart* = "FALSE" in "Camming" operating mode, *eGearingState* = "0" (STOPPED).
   - ⇨ The slave axis follows the master axis according to the curve profile defined in the CAM Editor.

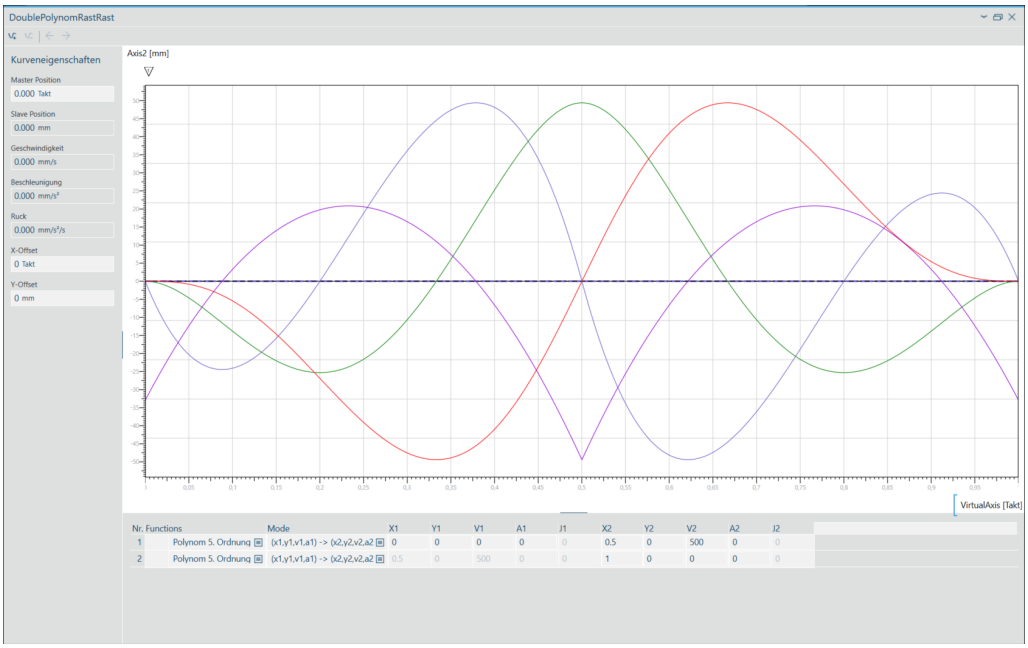


25429403019

[1] SEW_GVL.Interface_Axis1.ProfileGeneration.Velocity.In.xActivate
[2] SEW_GVL.Interface_Axis1.ProfileGeneration.Velocity.In.xStart
[3] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[4] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[5] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[6] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.In.xActivate
[7] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.In.xStart
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[9] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[10] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[11] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.Out.eGearingState

29179459/EN – 03/2020

The CAM Editor displays the curve profile as follows:



9007219998529675

Desynchronizing the slave axis

1. To desynchronize the axis, set *xStart* to "FALSE".
   ⇨ If an appropriate setting (e.g. *eStopMode* = "CW_DIRECT") is configured in the *CONFIG* structure, the slave axis is desynchronized immediately.
   ⇨ *xProfileActive* reports back "FALSE".
   ⇨ *eGearingState* reports back "STOPPED".

Camming	SEW_Cam.ModeCamming_UI	
IN	SEW_ICam.ST_Camming_IN	
xActivate	BOOL	TRUE
xStart	BOOL	FALSE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
xGetNewCamDescription	BOOL	FALSE
CONFIG	SEW_ICam.ST_Camming_CONFIG2	
Start	ST_CammingStart_CONFIG	
Stop	ST_CammingStop_CONFIG	
lrGearOutDistance	LREAL	0
lrGearOutStopPosition	LREAL	0
eStopMode	E_CAMMINGSTOPMODE	CW_DIRECT
Change	ST_CammingChange_CONFIG	
eCycleMode	E_CYCLEMODE	ADJUST_SLAVE_REFERENCE_POSITION
FileHandling	CamFileHandler_UI	
XOffsetCorrection	OffsetCorrection_UI	
YOffsetCorrection	OffsetCorrection_UI	
Adjust	AdjustProfilgenerator_UI	
OUT	SEW_ICam.ST_Camming_OUT	
xModeActive	BOOL	TRUE
xProfileActive	BOOL	FALSE
eGearingState	E_GEARING_STATE	STOPP
lrSetpPosition	LREAL	0
lrSetpVelocity	LREAL	ACTIVE
lrSetpAcceleration	LREAL	0
stSetpointMasterCam	SEW_IInterpolModes.ST_InterpolatedValues_UserUnits	
xSetMasterValueDone	BOOL	FALSE
xSetSlaveValueDone	BOOL	FALSE
xGetNewCamDescriptionDone	BOOL	FALSE

9007219998289163

29179459/EN – 03/2020

8.6.3 Synchronizing/desynchronizing with moving master axis

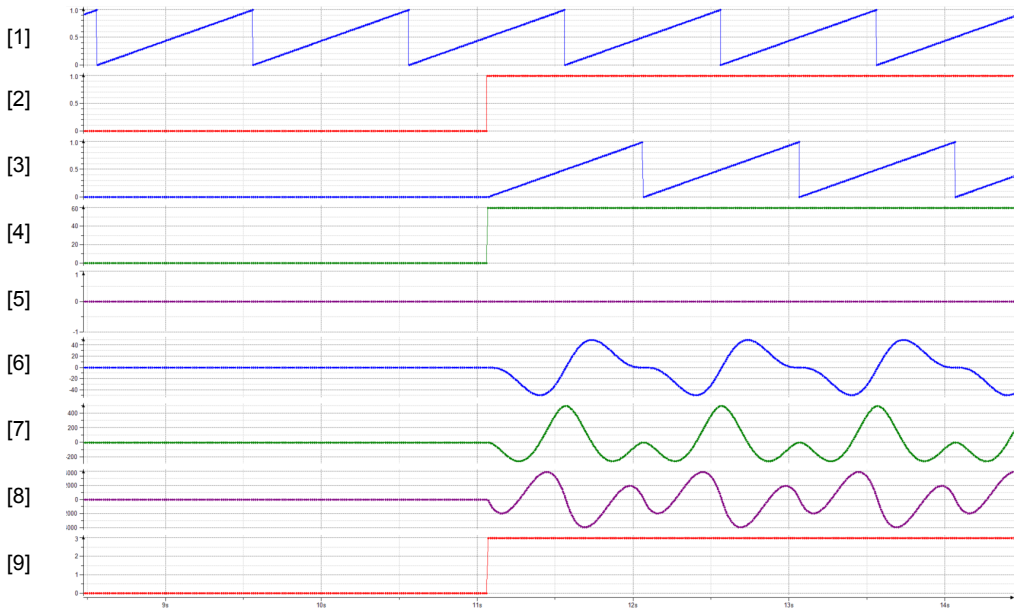
Direct synchronization without transition function

1. To synchronize the axis, set *xStart* to "TRUE".
   - ⇨ If an appropriate setting (e.g. *eStartMode* = "CW_DIRECT") is configured in the *CONFIG* structure, the axis is synchronized immediately.
   - ⇨ *xProfileActive* reports back "TRUE".
   - ⇨ *eGearingState* reports back "ACTIVE".

Camming	SEW_Cam.ModeCamming_UI	
IN	SEW_ICam.ST_Camming_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
xGetNewCamDescription	BOOL	FALSE
CONFIG	SEW_ICam.ST_Camming_CONFIG2	
Start	ST_CammingStart_CONFIG	
lrGearInDistance	LREAL	0
lrGearInOffset	LREAL	0
lrGearInReferencePosition	LREAL	0
eStartMode	E_CAMMINGSTARTMODE	CW_DIRECT
xStartWithLastXCorrection	BOOL	FALSE
Stop	ST_CammingStop_CONFIG	
Change	ST_CammingChange_CONFIG	
eCycleMode	E_CYCLEMODE	ADJUST_SLAVE_REFERENCE_POSITION

9007219998833675

   - ⇨ The internal master position starts movement when *xStart* is set.
   - ⇨ Setting *xStart* results in a jump in speed in the profile of the internal master position. Jumps in the speed and acceleration profile may also occur in the setpoint profile of the slave axis.
   - ⇨ In this phase, the internal master position is offset relative to the external master position.



25424613899

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.In.xStart
[3] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrPosition
[4] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrVelocity
[5] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrAcceleration
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[9] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.Out.eGearingState

29179459/EN – 03/2020

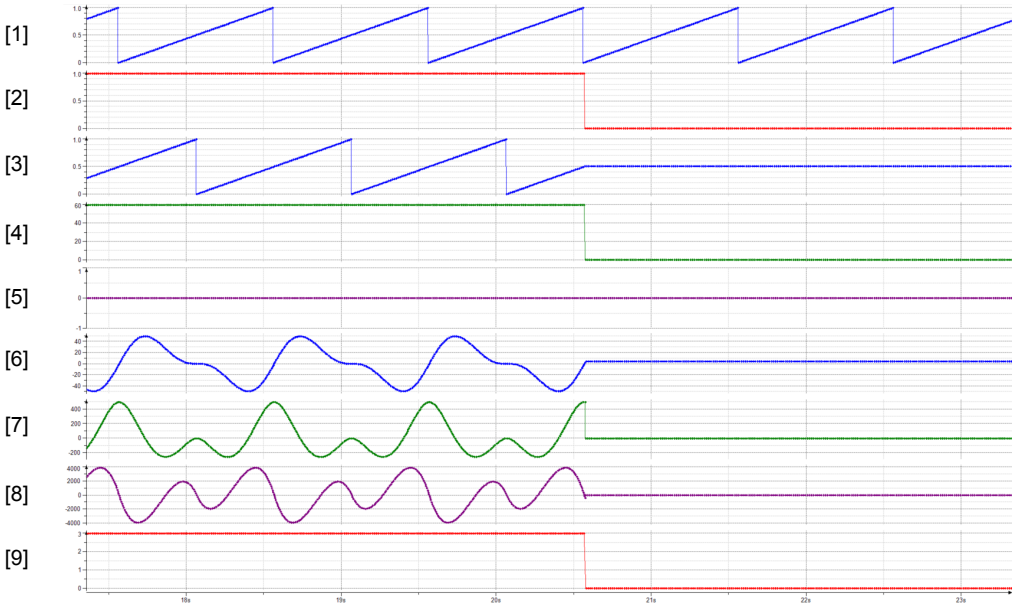
Direct desynchronization without transition function

1. To desynchronize the axis, set *xStart* to "FALSE".
   ⇨ If an appropriate setting (e.g. *eStopMode* = "CW_DIRECT") is configured in the *CONFIG* structure, the axis is desynchronized immediately.
   ⇨ *xProfileActive* reports back "FALSE".
   ⇨ *eGearingState* reports back "STOPPED".

Camming	SEW_Cam.ModeCamming_UI	
IN	SEW_ICam.ST_Camming_IN	
xActivate	BOOL	TRUE
xStart	BOOL	FALSE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
xGetNewCamDescription	BOOL	FALSE
CONFIG	SEW_ICam.ST_Camming_CONFIG2	
Start	ST_CammingStart_CONFIG	
Stop	ST_CammingStop_CONFIG	
lrGearOutDistance	LREAL	0
lrGearOutStopPosition	LREAL	0
eStopMode	E_CAMMINGSTOPMODE	CW_DIRECT
Change	ST_CammingChange_CONFIG	
eCycleMode	E_CYCLEMODE	ADJUST_SLAVE_REFERENCE_POSITION
FileHandling	CamFileHandler_UI	
XOffsetCorrection	OffsetCorrection_UI	
YOffsetCorrection	OffsetCorrection_UI	
Adjust	AdjustProfilgenerator_UI	
OUT	SEW_ICam.ST_Camming_OUT	
xModeActive	BOOL	TRUE
xProfileActive	BOOL	FALSE
eGearingState	E_GEARING_STATE	STOPP
lrSetpPosition	LREAL	0
lrSetpVelocity	LREAL	ACTIVE
lrSetpAcceleration	LREAL	0
stSetpointMasterCam	SEW_IInterpolModes.ST_InterpolatedValues_UserUnits	
xSetMasterValueDone	BOOL	FALSE
xSetSlaveValueDone	BOOL	FALSE
xGetNewCamDescriptionDone	BOOL	FALSE

9007219999367307

   ⇨ If *xStart* is set to "FALSE", the profile freezes and the internal master position stops and remains at its last value.
   ⇨ This also means that the slave position stops.
   ⇨ Jumps in the speed and acceleration profile may occur as a result.



25424586123

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.In.xStart
[3] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrPosition
[4] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrVelocity
[5] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrAcceleration
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[9] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.Out.eGearingState

29179459/EN – 03/2020

Direct synchronization with transition function

To avoid jumps in the setpoint profile during synchronization, use a master-based transition function. For this purpose, set *lrGearInDistance* > "0".

1. Set *lrGearInDistance* to "0.5".

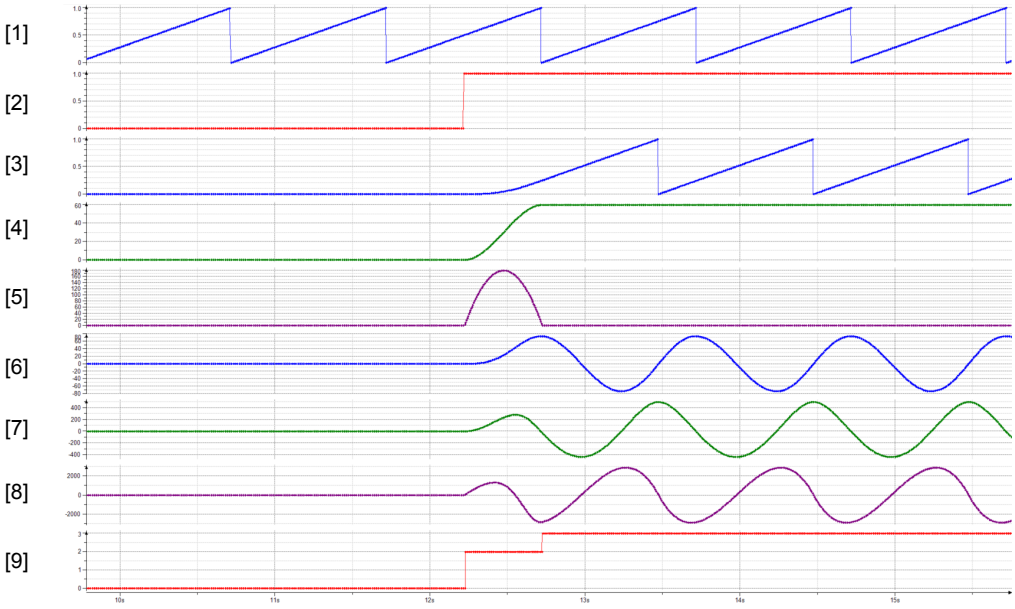
Camming	SEW_Cam.ModeCamming_UI	
IN	SEW_ICam.ST_Camming_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
xGetNewCamDescription	BOOL	FALSE
CONFIG	SEW_ICam.ST_Camming_CONFIG2	
Start	ST_CammingStart_CONFIG	
lrGearInDistance	LREAL	0.5
lrGearInOffset	LREAL	0
lrGearInReferencePosition	LREAL	0
eStartMode	E_CAMMINGSTARTMODE	CW_DIRECT
xStartWithLastXCorrection	BOOL	FALSE
Stop	ST_CammingStop_CONFIG	
Change	ST_CammingChange_CONFIG	
eCycleMode	E_CYCLEMODE	ADJUST_SLAVE_REFERENCE_POSITION

9007219999439499

⇨ The internal master position starts movement when *xStart* is set.

⇨ The setpoints of the internal master position follow a transition function smoothly without jumps.

⇨ The phase of the internal master position is offset to the external master position.



25424593419

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.In.xStart
[3] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrPosition
[4] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrVelocity
[5] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrAcceleration
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[9] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.Out.eGearingState

Direct desynchronization with transition function

To avoid jumps in the setpoint profile during desynchronization, use a master-based transition function. For this purpose, set *lrGearOutDistance* > "0".

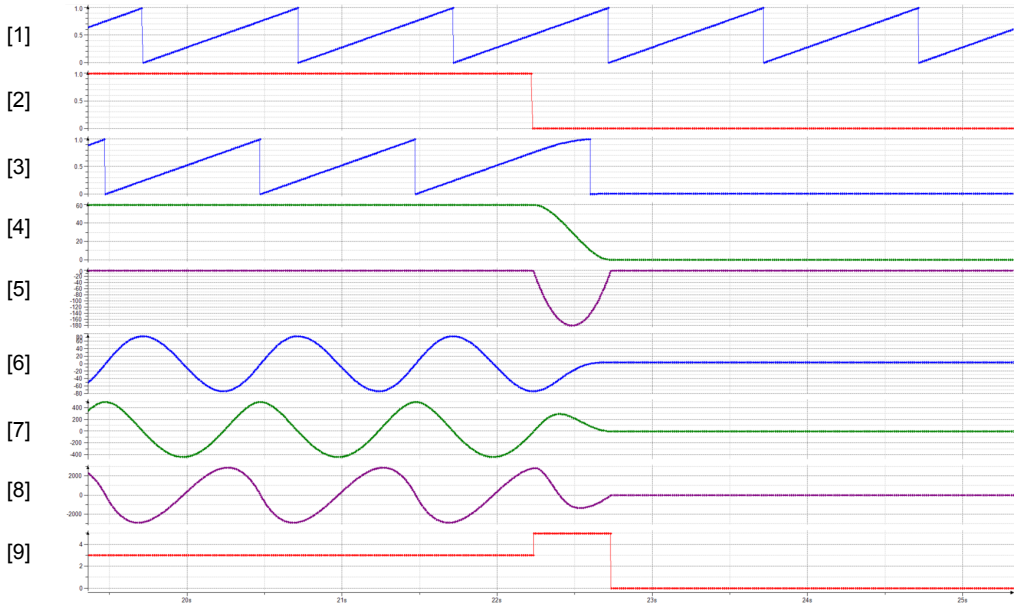
1. Set *lrGearOutDistance* to "0.5".

Camming	SEW_Cam.ModeCamming_UI	
IN	SEW_ICam.ST_Camming_IN	
xActivate	BOOL	TRUE
xStart	BOOL	FALSE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
xGetNewCamDescription	BOOL	FALSE
CONFIG	SEW_ICam.ST_Camming_CONFIG2	
Start	ST_CammingStart_CONFIG	
Stop	ST_CammingStop_CONFIG	
lrGearOutDistance	LREAL	0.5
lrGearOutStopPosition	LREAL	0
eStopMode	E_CAMMINGSTOPMODE	CW_DIRECT
Change	ST_CammingChange_CONFIG	
eCycleMode	E_CYCLEMODE	ADJUST_SLAVE_REFERENCE_POSITION

9007219999447691

⇨ If *xStart* is set to "FALSE", the setpoint profile of the internal master follows a transition function smoothly, without jumps, to standstill.

⇨ The setpoint profile of the slave axis follows the profile of the internal master position.



25424578955

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.In.xStart
[3] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrPosition
[4] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrVelocity
[5] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrAcceleration
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[9] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.Out.eGearingState

Synchronizing to reference position without transition function

To configure the phase position between internal and external master, take into account a *lrGearInReferencePosition* reference position when starting.

1. Define a *lrGearInReferencePosition* reference position.
2. Using *eStartMode*, define the behavior when *xStart* is set.

Camming	SEW_Cam.ModeCamming_UI	
IN	SEW_ICam.ST_Camming_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
xGetNewCamDescription	BOOL	FALSE
CONFIG	SEW_ICam.ST_Camming_CONFIG2	
Start	ST_CammingStart_CONFIG	
lrGearInDistance	LREAL	0
lrGearInOffset	LREAL	0
lrGearInReferencePosition	LREAL	0
eStartMode	E_CAMMINGSTARTMODE	WITH_CW_REFERENCE_POSITION
xStartWithLastXCorrection	BOOL	FALSE
Stop	ST_CammingStop_CONFIG	
Change	ST_CammingChange_CONFIG	
eCycleMode	E_CYCLEMODE	ADJUST_SLAVE_REFERENCE_POSITION

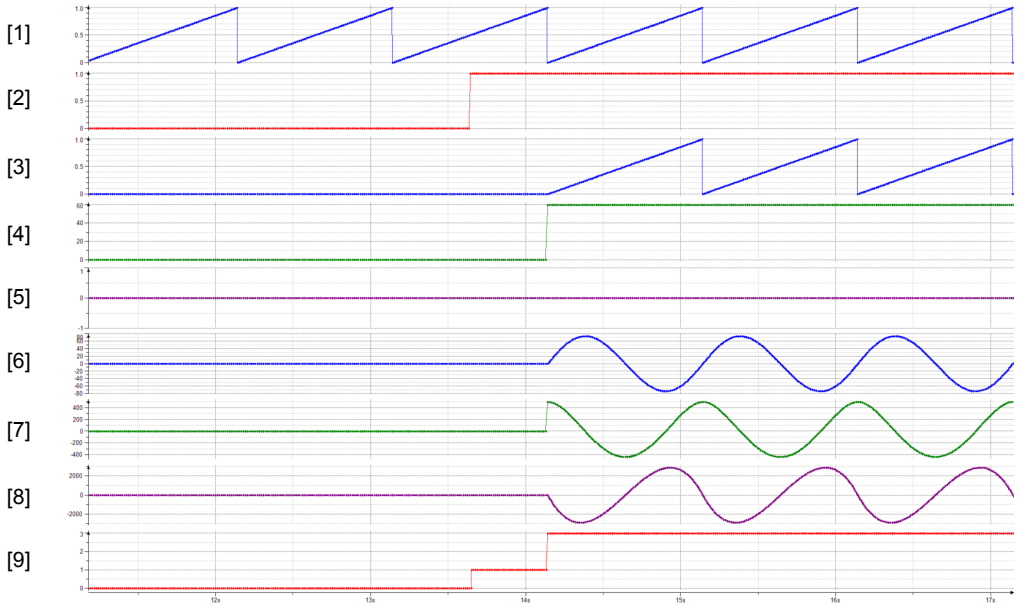
9007219999481611

⇨ When the position defined in *lrGearInReferencePosition* has been reached, the internal master position will start its movement, once *xStart* has been set to "TRUE".

⇨ A jump in speed occurs in the profile of the internal master position. Jumps in the speed and acceleration profile may also occur in the setpoint profile of the slave axis.

⇨ The phase position of the internal master position depends on *lrGearInReferencePosition*.

29179459/EN – 03/2020



25424629259

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.In.xStart
[3] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrPosition
[4] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrVelocity
[5] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrAcceleration
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[9] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.Out.eGearingState

29179459/EN – 03/2020

Desynchronizing to stop position without transition function

Using *lrGearOutStopPosition*, you can control the position at which the setpoint profile stops after desynchronization.

1. Set *lrGearOutStopPosition* to "0".
2. Set *eStopMode* to "WITH_CW_EXTERNAL_POSITION".

Camming	SEW_Cam.ModeCamming_UI	
IN	SEW_ICam.ST_Camming_IN	
xActivate	BOOL	TRUE
xStart	BOOL	FALSE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
xGetNewCamDescription	BOOL	FALSE
CONFIG	SEW_ICam.ST_Camming_CONFIG2	
Start	ST_CammingStart_CONFIG	
Stop	ST_CammingStop_CONFIG	
lrGearOutDistance	LREAL	0
lrGearOutStopPosition	LREAL	0
eStopMode	E_CAMMINGSTOPMODE	WITH_CW_EXTERNAL_POSITION
Change	ST_CammingChange_CONFIG	
eCycleMode	E_CYCLEMODE	ADJUST_SLAVE_REFERENCE_POSITION

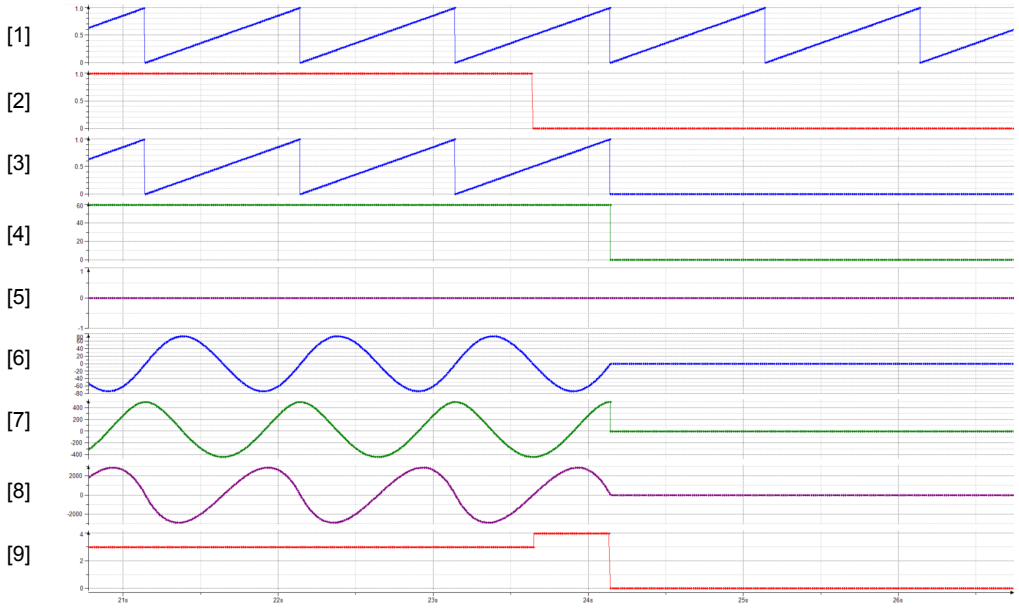
9007219999771403

⇨ If *xStart* is set to "TRUE", the internal master position stops and remains at the value defined in *lrGearOutStopPosition*.

⇨ This also means that the slave position stops.

⇨ This results in a jump in the speed profile of the internal master position.

29179459/EN – 03/2020



25424572427

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.In.xStart
[3] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrPosition
[4] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrVelocity
[5] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrAcceleration
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[9] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.Out.eGearingState

29179459/EN – 03/2020

Synchronizing to reference position with transition function

You can also use a transition function when synchronizing to a reference position by setting lrGearInDistance > "0".

1. Set *lrGearInDistance* to "0.5".
2. Set *lrGearInReferencePosition* to "0".
3. Set *eStartMode* to "WITH_CW_REFERENCE_POSITION".

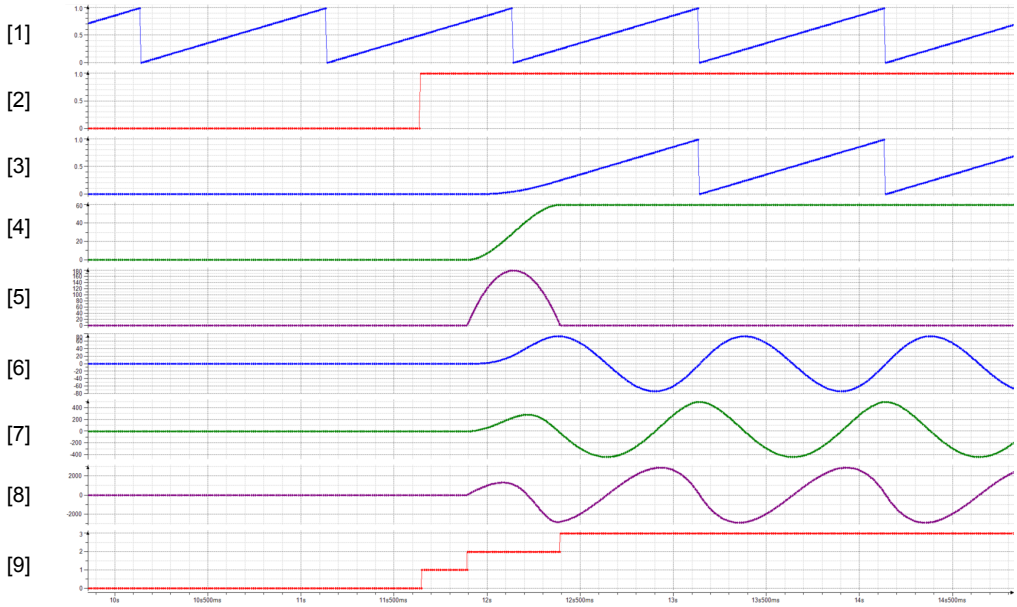
Camming	SEW_Cam.ModeCamming_UI	
IN	SEW_ICam.ST_Camming_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
xGetNewCamDescription	BOOL	FALSE
CONFIG	SEW_ICam.ST_Camming_CONFIG2	
Start	ST_CammingStart_CONFIG	
lrGearInDistance	LREAL	0.5
lrGearInOffset	LREAL	0
lrGearInReferencePosition	LREAL	0
eStartMode	E_CAMMINGSTARTMODE	WITH_CW_REFERENCE_POSITION
xStartWithLastXCorrection	BOOL	FALSE
Stop	ST_CammingStop_CONFIG	
Change	ST_CammingChange_CONFIG	
eCycleMode	E_CYCLEMODE	ADJUST_SLAVE_REFERENCE_POSITION

9007219999779595

⇨ The phase position is defined using *lrGearInReferencePosition*.

⇨ The transition function ensures that the setpoint profiles are smooth (continuous) and without any jumps.

29179459/EN – 03/2020



25424621579

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.In.xStart
[3] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrPosition
[4] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrVelocity
[5] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrAcceleration
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[9] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.Out.eGearingState

29179459/EN – 03/2020

Desynchronizing to stop position with transition function

You can also use a transition function when desynchronizing to a defined stop position by setting *lrGearOutDistance* > "0".

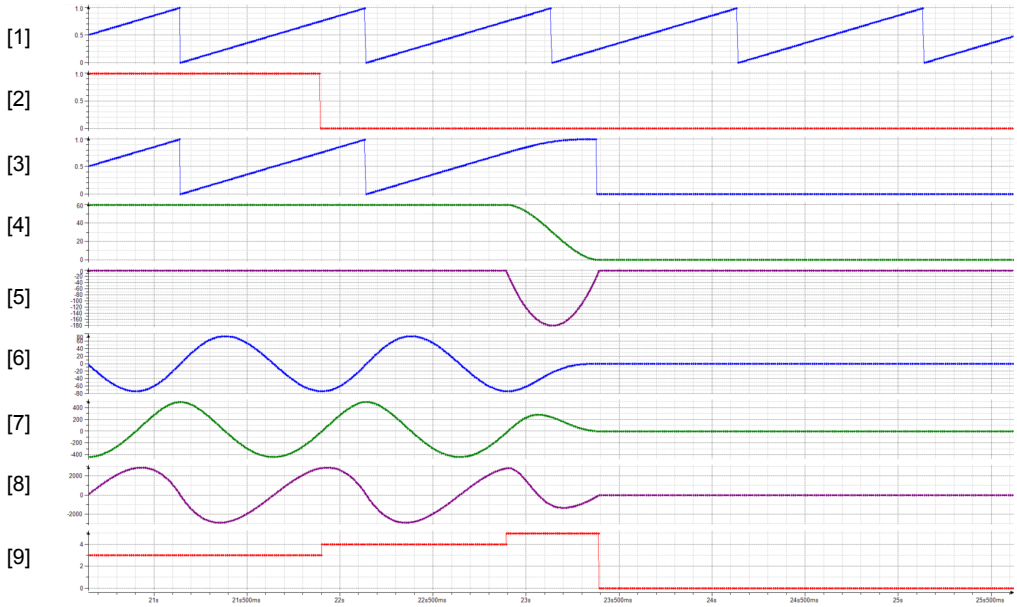
1. Set *lrGearOutDistance* to "0.5".
2. Set *lrGearOutStopPosition* to "0".
3. Set *eStartMode* to "WITH_CW_EXTERNAL_POSITION".

Camming	SEW_Cam.ModeCamming_UI	
IN	SEW_ICam.ST_Camming_IN	
xActivate	BOOL	TRUE
xStart	BOOL	FALSE
xSetMasterValue	BOOL	FALSE
lrMasterValue	LREAL	0
xSetSlaveValue	BOOL	FALSE
lrSlaveValue	LREAL	0
xReset	BOOL	FALSE
xGetNewCamDescription	BOOL	FALSE
CONFIG	SEW_ICam.ST_Camming_CONFIG2	
Start	ST_CammingStart_CONFIG	
Stop	ST_CammingStop_CONFIG	
lrGearOutDistance	LREAL	0.5
lrGearOutStopPosition	LREAL	0
eStopMode	E_CAMMINGSTOPMODE	WITH_CW_EXTERNAL_POSITION
Change	ST_CammingChange_CONFIG	
eCycleMode	E_CYCLEMODE	ADJUST_SLAVE_REFERENCE_POSITION

9007219999851787

⇨ The stop position is defined using *lrGearOutStopPosition*.

⇨ The transition function ensures that the setpoint profiles are smooth (continuous) and without any jumps.



25424553483

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.In.xStart
[3] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrPosition
[4] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrVelocity
[5] SEW_GVL_Internal.Axis2._fbCammingProfile.stMasterHandlerOut.lrAcceleration
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[9] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.Out.eGearingState

8.6.4 Offsetting the phase position

You can use the *XOffsetCorrection* structure to control the phase shift. In the example below, the phase position is offset by half a cycle.

1. Specify the dynamic parameters speed (*IN.lrVelocity*), acceleration (*IN.lrAcceleration*) and deceleration (*IN.Deceleration*).

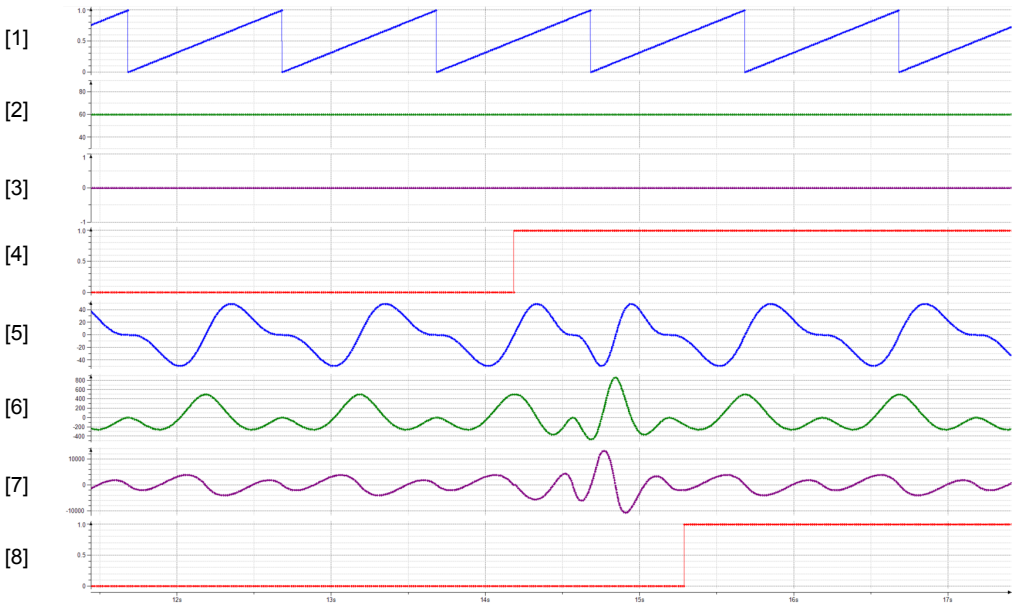
29179459/EN – 03/2020

2. Set *IN.lrCorrection* to "0.5".
3. Set *IN.xStart* to "TRUE".

XOffsetCorrection	FB_UI_OffsetCorrection	
IN	SEW_MOS_ICamming.ST_OffsetCorrection_IN	
xStart	BOOL	TRUE
lrCorrection	LREAL	0.5
lrVelocity	LREAL	100
lrAcceleration	LREAL	100
lrDeceleration	LREAL	100
lrJerk	LREAL	0
OUT	SEW_MOS_ICamming.ST_OffsetCorrectionUI_OUT	
xModeActive	BOOL	TRUE
xProfileActive	BOOL	FALSE
lrSetpPosition	LREAL	0.5
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xInPosition	BOOL	TRUE
xFBError	BOOL	FALSE
udiFBErrorID	UDINT	0

20741853067

⇨ The phase position is offset by half a cycle:



25424636939

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[3] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[4] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.XOffsetCorrection.In.xStart
[5] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[8] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.XOffsetCorrection.Out.xInPosition

Note:

Following *xStart* = "TRUE" at *XOffsetCorrection*, the phase position is offset by *lrCorrection* (in this example: 0.5). Once the phase shift is set, *xInPosition* becomes "TRUE".

8.6.5 Offsetting the amplitude

You can use the *YOffsetCorrection* structure to control the amplitude shift. In the example below, the amplitude is offset by 50 mm.

1. Specify the dynamic parameters speed (*IN.lrVelocity*), acceleration (*IN.lrAcceleration*) and deceleration (*IN.Deceleration*).

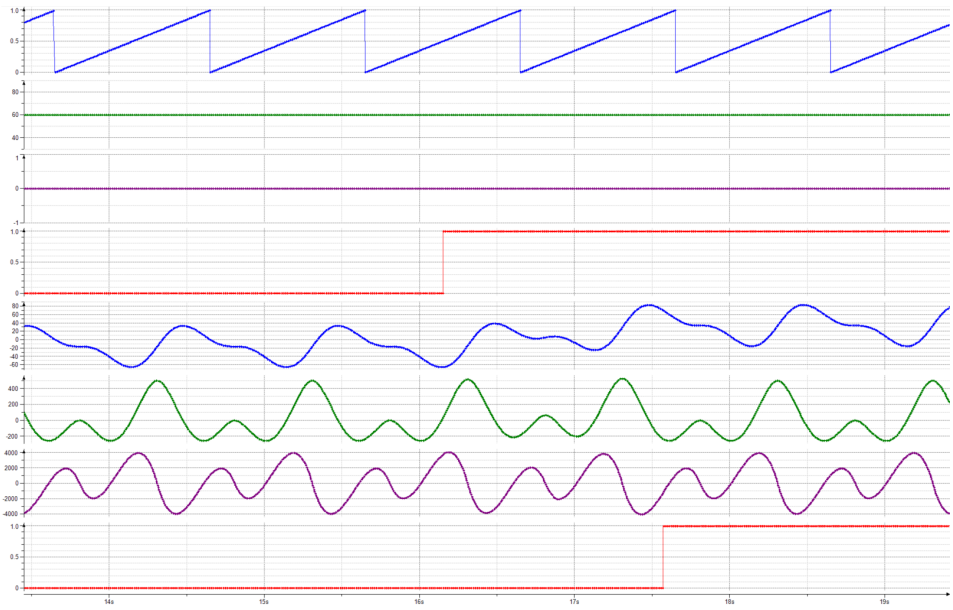
2. Set *IN.lrCorrection* to "50".
3. Set *IN.xStart* to "TRUE".

YOffsetCorrection	FB_UI_OffsetCorrection	
IN	SEW_MOS_ICamming.ST_OffsetCorrection_IN	
xStart	BOOL	TRUE
lrCorrection	LREAL	50
lrVelocity	LREAL	100
lrAcceleration	LREAL	100
lrDeceleration	LREAL	100
lrJerk	LREAL	0
OUT	SEW_MOS_ICamming.ST_OffsetCorrectionUI_OUT	
xModeActive	BOOL	TRUE
xProfileActive	BOOL	FALSE
lrSetpPosition	LREAL	50
lrSetpVelocity	LREAL	0
lrSetpAcceleration	LREAL	0
xInPosition	BOOL	TRUE
xFBError	BOOL	FALSE
udiFBErrorID	UDINT	0

20742821259

⇨ The amplitude is offset by 50 mm:

[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]



25424547723

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[3] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[4] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.YOffsetCorrection.In.xStart
[5] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[8] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.YOffsetCorrection.Out.xInPosition

29179459/EN – 03/2020

Note:

Following *xStart* = "TRUE" at *YOffsetCorrection*, the amplitude is offset by *IrCorrection* (in this example: 50.0). Once amplitude shift is set, *xInPosition* becomes "TRUE".

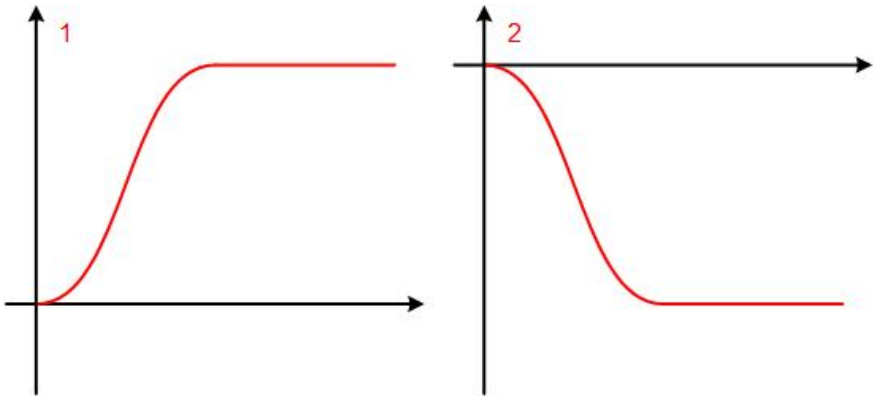
29179459/EN – 03/2020

8.6.6 Using eCycleMode

The eCycleMode variable offers the following setting options:

Setting: ADJUST_SLAVE_REFERENCE_POSITION

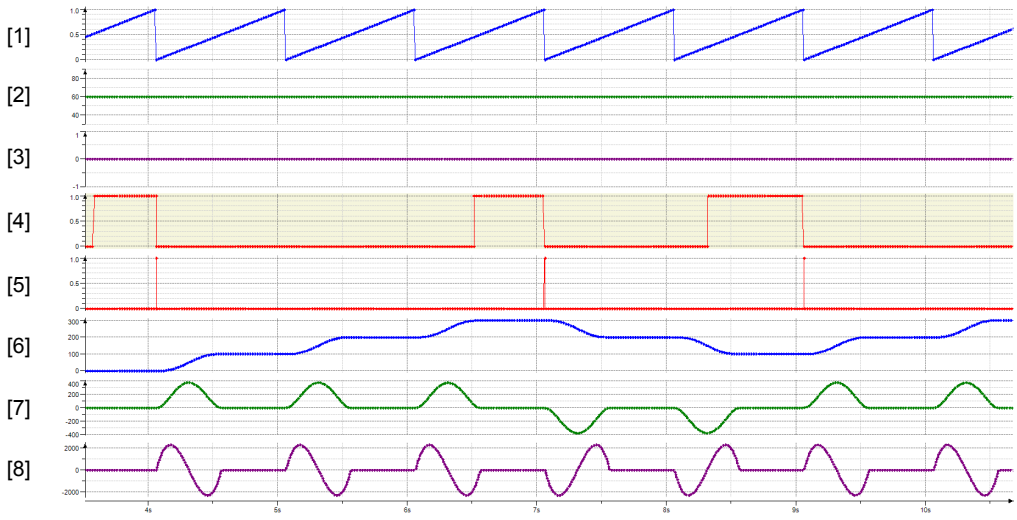
In this setting, the reference point for the next curve is placed onto the end point of the previous curve once a cam profile has been passed through. In this way, curves beginning with slave position 0 can be arranged in a continuous row. In the example, a change is made between the following curves so they are passed through several times one after the other.



29109273355

29179459/EN – 03/2020

The following trace recording shows the profile in which first curve 1 is passed through three times, next curve 2 two times, and finally curve 1 again:



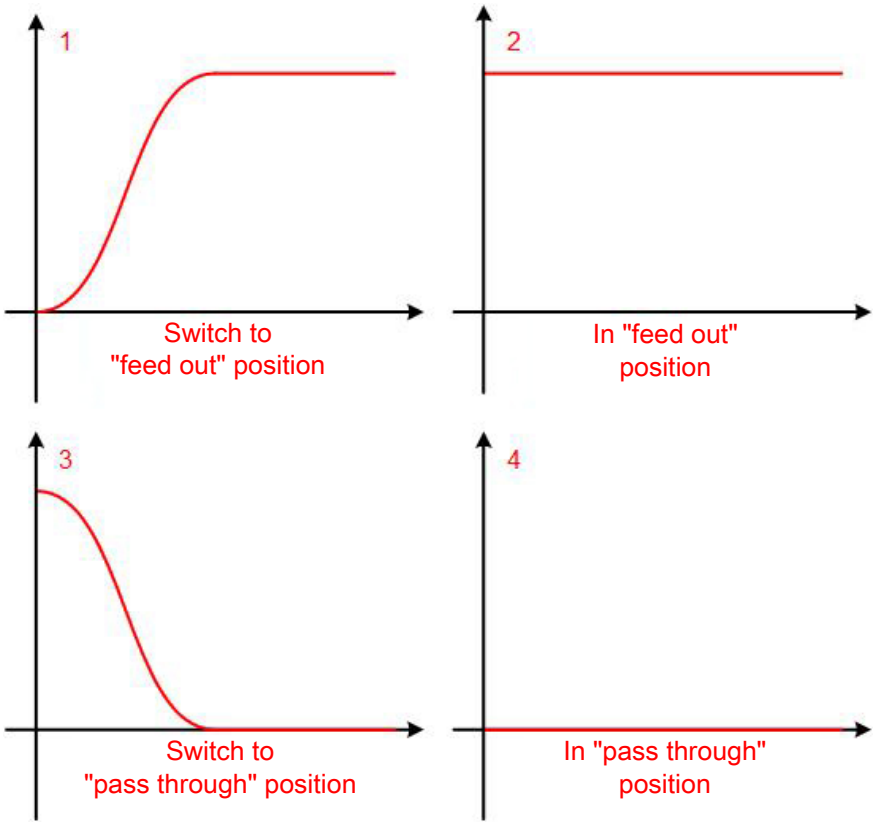
29006111755

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[3] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[4] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.IN.xGetNewCamDescription
[5] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.OUT. xGetNewCamDescriptionDone
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration

Setting: KEEP_SLAVE_REFERENCE_POSITION

This setting is useful, for example, when users want to switch between several curves and want to interpret the position of the various curve profiles as the absolute position of the slave axis.

A simple application could be to control the feeding out of goods. It is controlled by switching between the following four curves:



9007228364018315

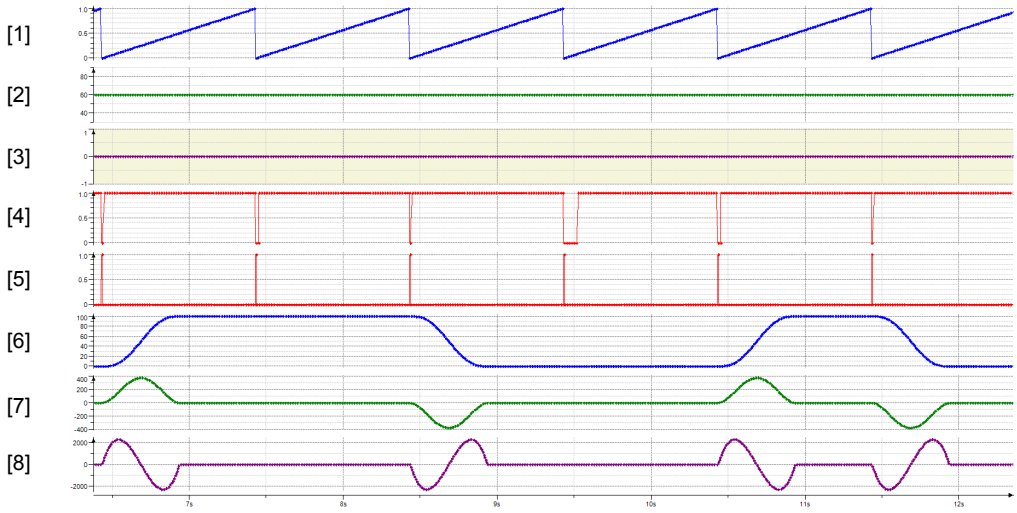
In all curves, the position of the curve profile corresponds to the absolute position of the slave axis. This means the reference point is the same in all curves, i.e. the 0 line. In this case, the end point of curve 1 must not be set automatically as the new reference point for curve 2 when switching from curve 1 to curve 2 because this would cause an abrupt setpoint change. The *eCycleMode* parameter must therefore be set to the value "KEEP_SLAVE_REFERENCE_POSITION".

When this setting is selected, the user is responsible for implementing the control in such a way that no abrupt setpoint changes occur. In this example, the user must ensure that once curve 1 has been passed through, switching takes place either to curve 2 or to curve 3. In the same way, curve 3 must be followed by curve 4 or by curve 1 again.

The following trace recording shows an example of a profile that could be resulting from switching between the various curves. The following sequence of curves is shown:

29179459/EN – 03/2020

Curve 1 => curve 2 => curve 3 => curve 4 => curve 1 => curve 3



29006570635

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[3] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[4] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.IN.xGetNewCamDescription
[5] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.OUT. xGetNewCamDescriptionDone
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration

29179459/EN – 03/2020

8.6.7 Creating a curve description using constructors

For creating a curve description from the IEC program, we advise against writing into the corresponding data structure for the following reasons:

- The data might be inconsistent.

  No checking is performed as to whether the values set at the limits can be implemented using the selected mathematical function. For example, a linear function could be configured with two different speeds at the left and the right segment border.

- Double assignments are required.

  The values at the left limit of a segment are usually identical with those at the right limit of the previous segment. In the case of direct access, the user would have to ensure that a value is always assigned correctly at both limits.

- It is the user who has to take account of mathematical dependencies.

  By using constructors, an optimized symmetrical acceleration profile with a 5th degree polynomial is obtained if there are certain interrelations between the values at the segment limits. With direct access to the data structure, users have to calculate these interrelations themselves in order to determine the values accordingly.

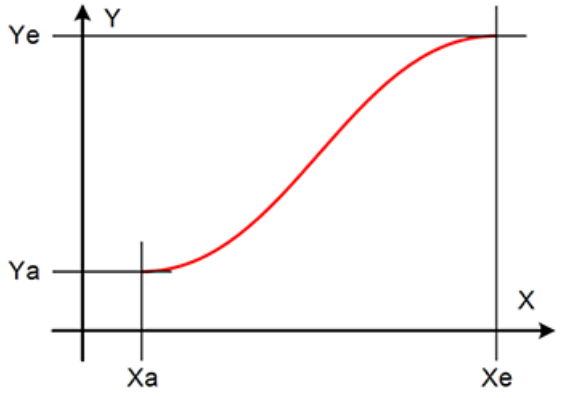
We recommend using so-called "constructors" instead of direct access to the data structure for the following reasons:

- Constructors provide an amount of parameters that matches the mathematical function. All the other parameters at the segment limits are calculated consistently.
- The "Append" constructor is available for adding a curve segment to an existing segment. This constructor calculates the values at the right limit of the previous segment automatically.
- Constructors take account of mathematical dependencies so that users no longer have to worry about them.

Each mathematical function provides the matching set of constructors. The following types of constructors are basically distinguished:

- "RestRest"

  Defines a rest-rest profile with (Xa, Ya, Xe, Ye); Va = Ve = Aa = Ae = 0

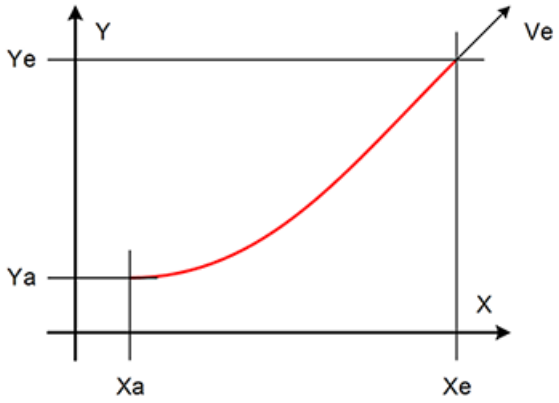


29017381515

- "RestVelocity"

  Defines a rest-speed profile with (Xa, Ya, Xe, Ye, Ve); Va = Aa = Ae = 0

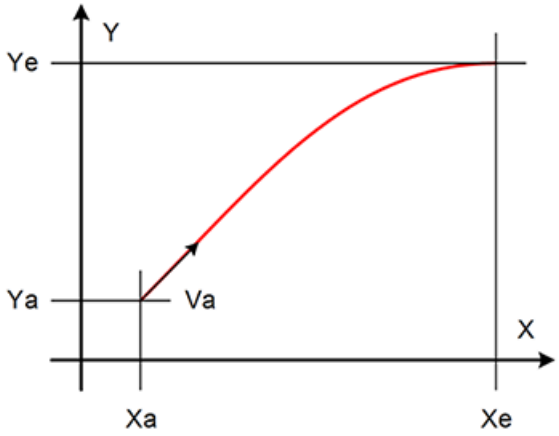
29179459/EN – 03/2020



29017550987

- "VelocityRest"

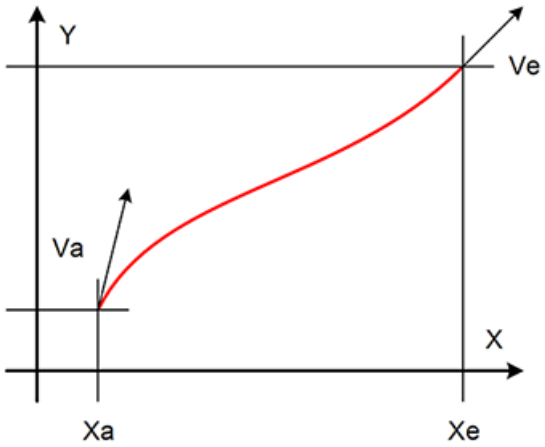
  Defines a speed-rest profile with (Xa, Ya, Va, Xe, Ye); Ve = Aa = Ae = 0



29017554059

- "Common"

  Defines a general profile matching the mathematical function. The number of specified parameters depends on the mathematical function.

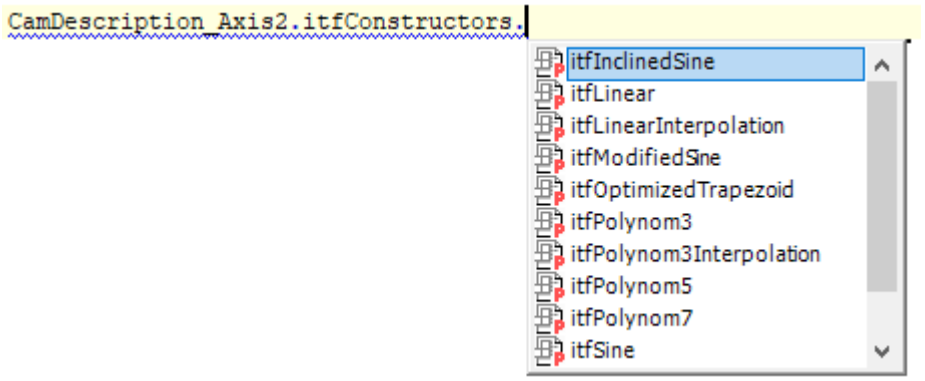


29017557131

Two more types of constructors are distinguished:

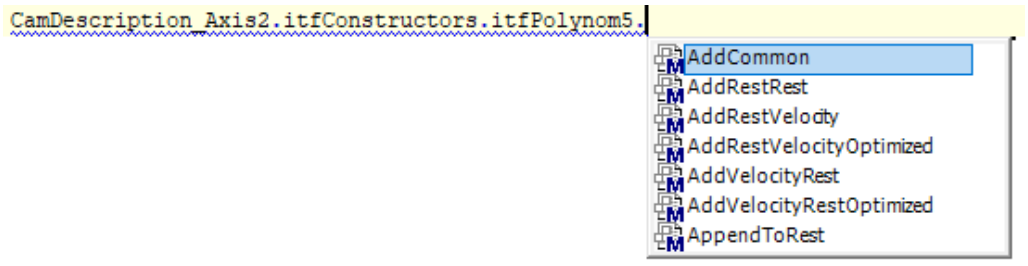
- "Add" constructors add a segment to the curve description with the segment limits resulting from the parameters passed to the constructor.
- "Append" constructors add a segment to the curve description with the values at the left limit resulting from the values at the right limit of the previous segment.

Access is made via the *itfConstructors* property at a variable of the type "CamDescription". This is where the required mathematical function is selected:



29017560715

Next, select the required constructor and define the required values using the specified variables:

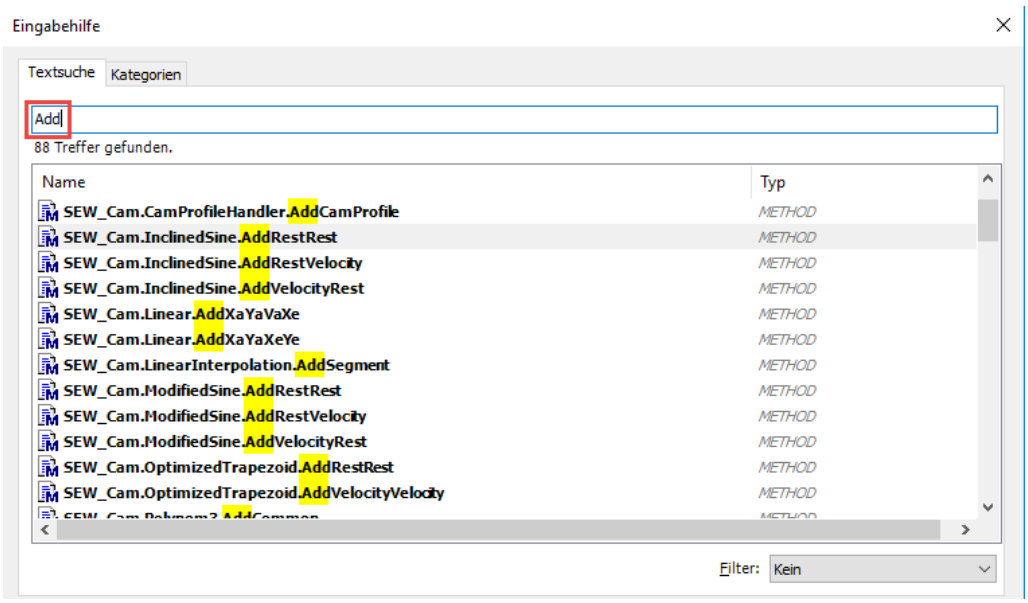


29017576715

```
CamDescription_Axis2.itfConstructors.itfPolynom5.AddRestRest    (
                                                                lrXa :=
                                                                lrYa :=
                                                                lrXe :=
                                                                lrYe :=
                                                                )
```

29017579915

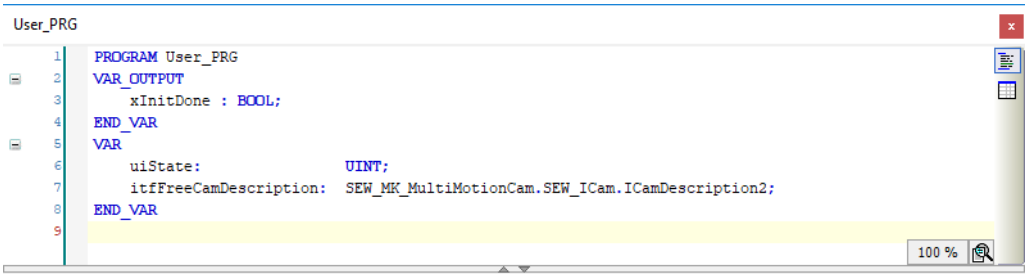
If the name of a constructor is indicated in the text search (e.g. "Add"), you can choose the input help ("F2") instead.



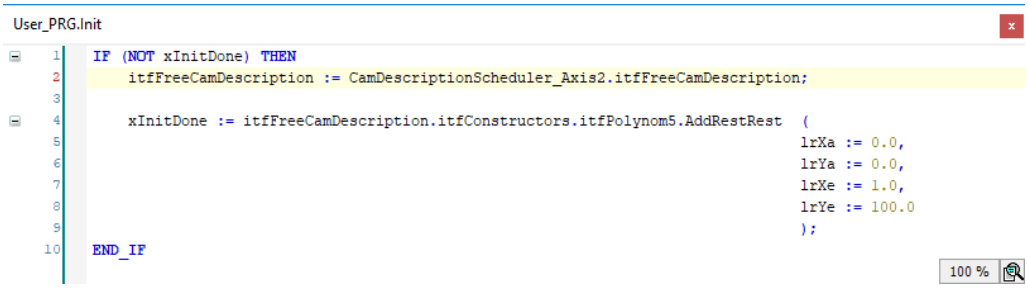
29017582987

Automatic code generation generates a variable of the type "CamDescriptionScheduler" for each axis that uses MOVIKIT® MultiMotion Camming. It contains two curve descriptions of the type "CamDescription" between which users can switch on the fly. In this case, the curve descriptions of the "CamDescriptionScheduler" are managed, which means one of the two curve descriptions might currently be active and should not be configured.

Users first have to be granted access to the curve description that is currently not in use and save it to a local variable of the type "ICamDescription2". Only then can users access the required constructor via the locally saved variable. Constructors have a return value of the type "BOOL" indicating whether the configuration has been successful.



29017610379



29017613835

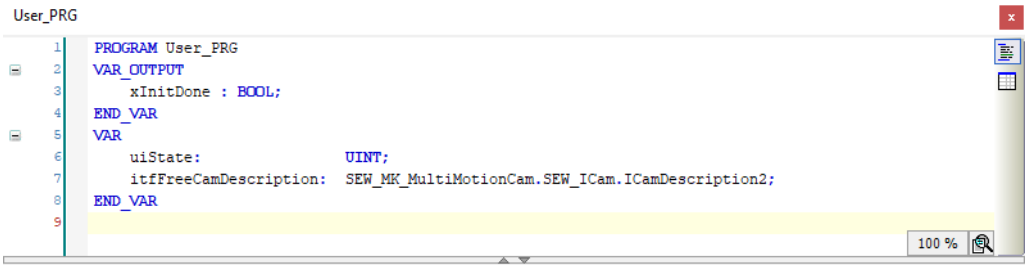
INFORMATION

A constructor adds a segment each time the curve description has been called successfully. A successful call is processed within a PLC cycle. This is why it is important that a constructor is not called in several subsequent cycles.

Following a sample program code for configuring the curve description for a simple flying saw:

User_PRG

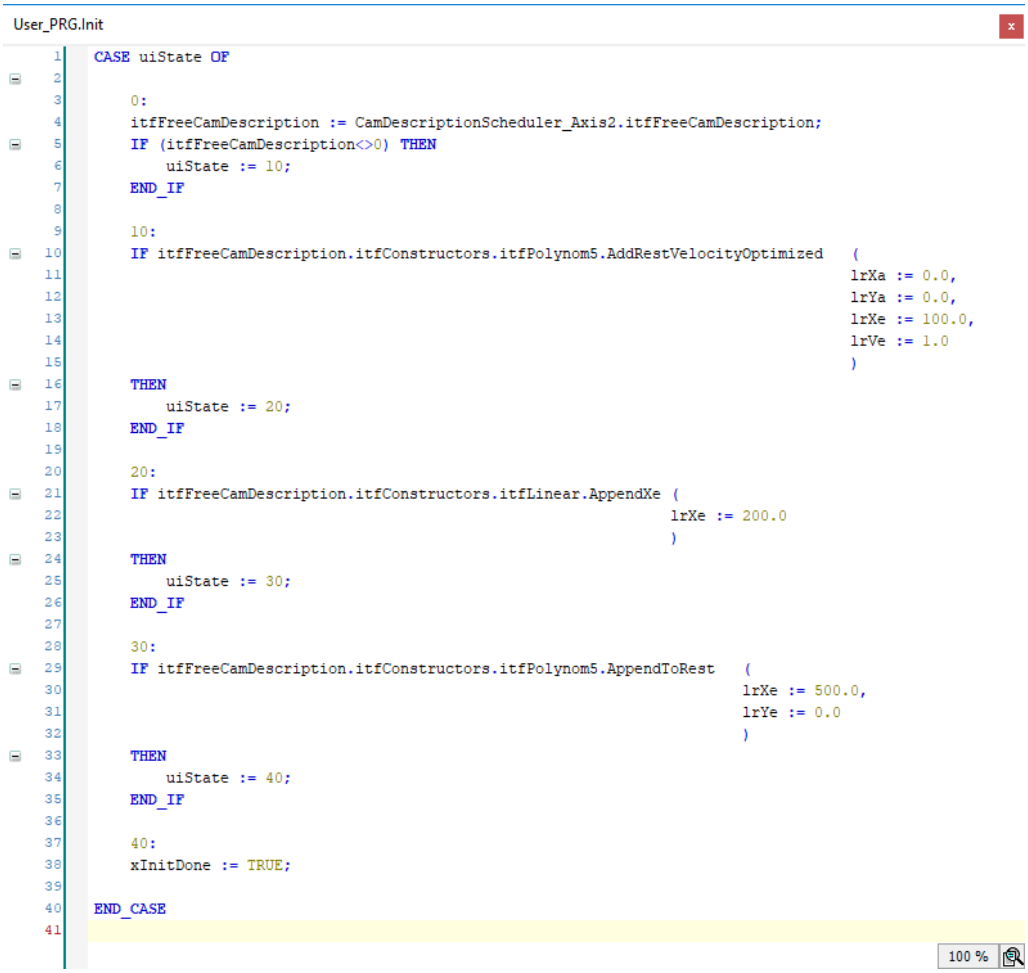
```
PROGRAM User_PRG
VAR_OUTPUT
    xInitDone : BOOL;
END_VAR
VAR
    uiState:                UINT;
    itfFreeCamDescription:  SEW_MK_MultiMotionCam.SEW_ICam.ICamDescription2;
END_VAR
```



29017706891

User_PRG.Init

```
CASE uiState OF

    0:
    itfFreeCamDescription := CamDescriptionScheduler_Axis2.itfFreeCamDescription;
    IF (itfFreeCamDescription<>0) THEN
        uiState := 10;
    END_IF

    10:
    IF itfFreeCamDescription.itfConstructors.itfPolynom5.AddRestVelocityOptimized   (
                                                                                     lrXa := 0.0,
                                                                                     lrYa := 0.0,
                                                                                     lrXe := 100.0,
                                                                                     lrVe := 1.0
                                                                                     )
    THEN
        uiState := 20;
    END_IF

    20:
    IF itfFreeCamDescription.itfConstructors.itfLinear.AppendXe (
                                                                 lrXe := 200.0
                                                                 )
    THEN
        uiState := 30;
    END_IF

    30:
    IF itfFreeCamDescription.itfConstructors.itfPolynom5.AppendToRest   (
                                                                        lrXe := 500.0,
                                                                        lrYe := 0.0
                                                                        )
    THEN
        uiState := 40;
    END_IF

    40:
    xInitDone := TRUE;

END_CASE
```

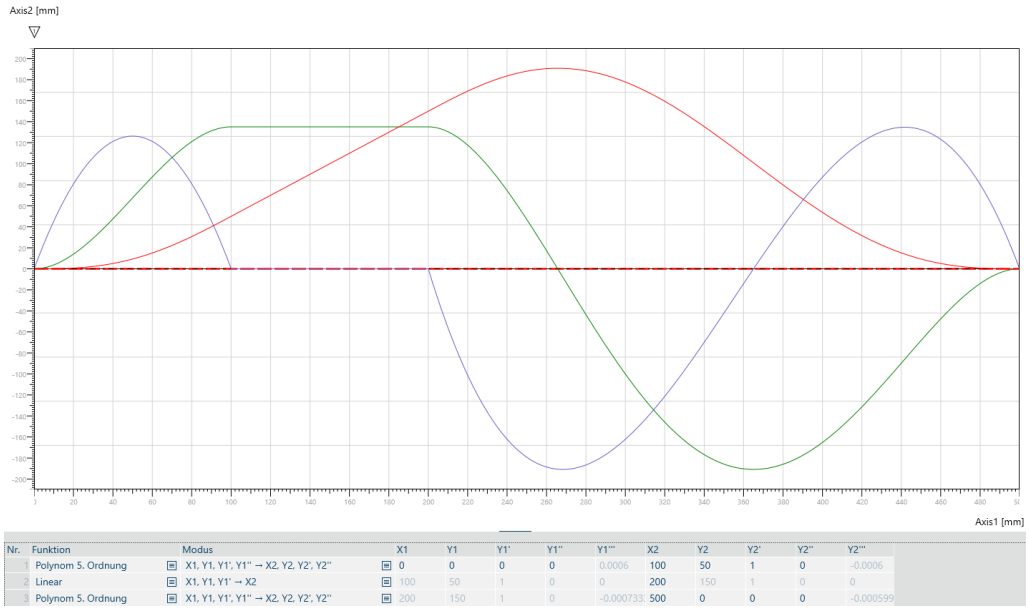


29017710347

The curve consists of the following three segments:

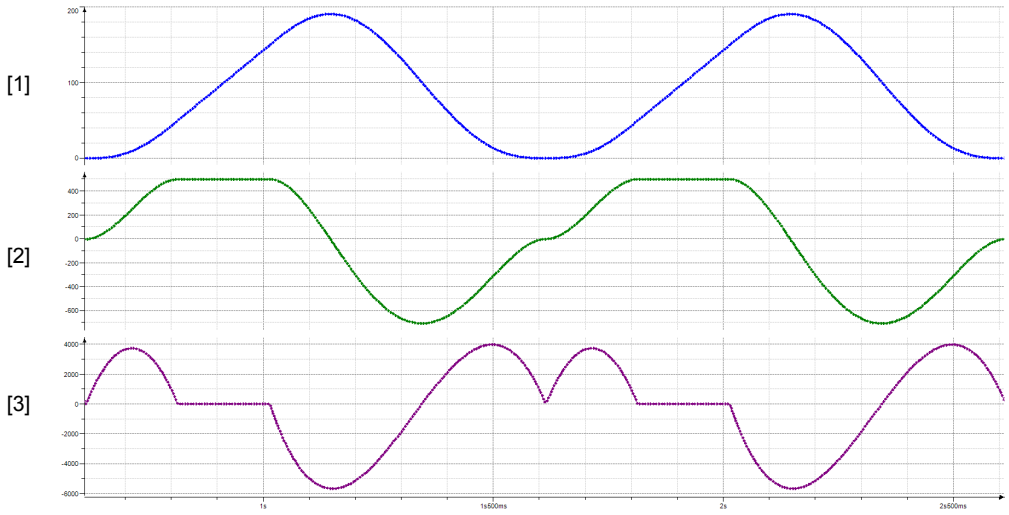
- Polynom 5 "rest speed"
- Straight line with constant speed
- Polynom 5 "speed-rest"

The curve configured with the constructors is represented as follows in the Cam Editor:



29017713803

The profiles look as follows in the trace recording:



29017749003

[1] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition

[2] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity

[3] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration

8.6.8 Switching curve description on the fly

INFORMATION

The user is responsible to ensure that no abrupt setpoint changes occur when another curve description is applied. In the example, position, speed, and acceleration are all "0" at the time of changeover.

Some situations require that the curve profile of an axis is changed during ongoing operation. In this case, an instance of the class *CamDescriptionScheduler* must be used. Automatic code generation generates an instance of this class in the global variable list *SEW_GVL* for each axis that uses MOVIKIT® MultiMotion Camming.

SEW_GVL

```
1  VAR_GLOBAL
2  Interface_Axis1 : SEW_MK_MultiMotionCam.MultiMotionVirtualCammingAxis_UI;
3  CamDescriptionScheduler_Axis1 : SEW_MK_MultiMotionCam.CamDescriptionScheduler;
4  Interface_Axis2 : SEW_MK_MultiMotionCam.MultiMotionVirtualCammingAxis_UI;
5  CamDescriptionScheduler_Axis2 : SEW_MK_MultiMotionCam.CamDescriptionScheduler;
6  END_VAR
```

100 %

29006029451

The *CamDescriptionScheduler* contains two curve descriptions of the type *CamDescription*. It calculates the curve descriptions in such a way that one is active during ongoing operation and the other one is free for generating a new curve profile. The currently active curve description is displayed in the *uiCamDescriptionActive* variable:

CamDescriptionScheduler_Axis2	SEW_MK_MultiMotionCam.CamDescriptionScheduler	
aCamDescription	ARRAY [1..2] OF CamDescription	
aCamDescription[1]	CamDescription	
aCamDescription[2]	CamDescription	
uiCamDescriptionActive	UINT	0

29006034443

When reading a curve file, its curve description is stored in *CamDescription[1]* as long as *Camming* operating mode is not active. The curve description is marked as active as soon as the *Camming* operating mode is activated. If another curve file is read, its curve description will be stored in the *CamDescription[2]* structure.

For changeover on the fly, some parameters have to be set in the configuration. Changeover usually does not take place in the middle of a cycle but at the beginning of the next cycle. The following settings have to be made for this purpose:

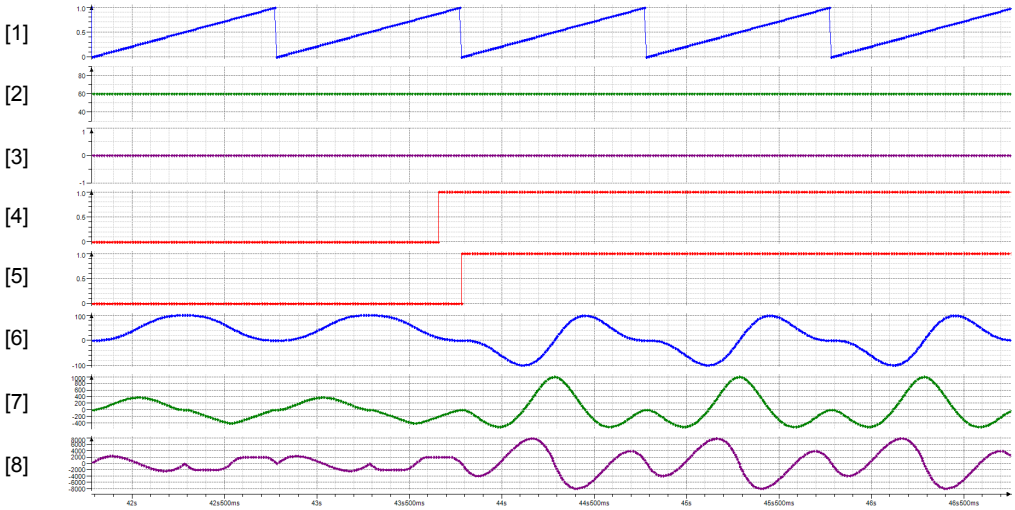
Change	ST_CammingChange_CONFIG	
lrChangeReferencePosition	LREAL	0
eChangeMode	E_CAMMINGCHANGEMODE	WITH_CW_INTERNAL_POSITION
eTransitionMode	E_CAMMINGCHANGETRANSITION	DIRECT

29006037515

This setting configures that the curve change is to take place with reference position 0 (which means at the beginning of the next cycle) and in positive direction of movement of the master. Changeover is triggered by setting the *xGetNewCamDescription* signal. *xGetNewCamDescriptionDone* returns the value "TRUE" as feedback.

29179459/EN – 03/2020

The following trace recording diagram illustrates this configuration. Once *xGetNewCamDescription* is set to "TRUE", the curve profile is changed when the next cycle begins, and *xGetNewCamDescriptionDone* returns the value "TRUE".



29006580619

[1] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[2] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[3] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[4] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.IN.xGetNewCamDescription
[5] SEW_GVL.Interface_Axis2.ProfileGeneration.Camming.OUT. xGetNewCamDescriptionDone
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[8] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration

8.6.9 Creating curve description from curve point table

This application example shows the interpolation of a curve description from a curve point table. For further information, refer to chapter "InterpolationHandler" (→ 🗎 155).

The application example requires a MOVISUITE® project in which MOVIKIT® MultiMotion Camming is assigned to an axis and the "Interpolation" add-on is activated.

Preparing the curve point table

Reading curve point table from file

Loading the source file onto the MOVI-C® CONTROLLER

First load the source file with the curve point table into the corresponding directory on the memory card of the MOVI-C® CONTROLLER. To do so, use the file management of the MOVI-C® CONTROLLER in the IEC Editor.

Proceed as follows to upload the source file:

✓ There is a connection between engineering PC and MOVI-C® CONTROLLER.

✓ A source file with a suitable curve point table is available. See chapter "Curve point table" (→ 🗎 155).

1. Under "Devices", open the configuration of the MOVI-C® CONTROLLER.
2. In the configuration of the MOVI-C® CONTROLLER, open the "Files" tab.
3. In the "Runtime system" system, click the [Update] button.
   ⇨ The file system on the MOVI-C® CONTROLLER is displayed.
4. Navigate in the file system on the MOVI-C® CONTROLLER to the required target folder: `DATA/Camming/Achsname`.
5. In the "Host" section, navigate to the directory that contains your source file.
6. Select your source file in the "Host" section.
7. Click the [>>] button between the "Host" and "Runtime system" sections.

⇨ The source file has now been uploaded to the MOVI-C® CONTROLLER.

Reading the source file into the IEC program

Proceed as follows to read the source file into the IEC program:

✓ The source file is stored on the MOVI-C® CONTROLLER.

1. Open the IEC project.
2. Assign the file name of your source file to the input variable *sFileName* (`ProfileGeneration>Camming>FileHandling>InterpolationHandler>IN`).
3. Assign the required interpolation method (LINEAR, POLYNOM_3, SPLINE) to the input variable *eInterpolationSegmentMode* (`ProfileGeneration>Camming>FileHandling>InterpolationHandler>IN`).
4. Set the input variable *xLoadPointsFromFile* (`ProfileGeneration>Camming>FileHandling>InterpolationHandler>IN`) to "TRUE".

⇨ In the structure OUT (`ProfileGeneration>Camming>FileHandling>InterpolationHandler`), the output variable *xPointsFromFileLoaded* returns "TRUE" and the output variable *sLastReadPointsFile* provides the file name of the source file.

29179459/EN – 03/2020

⇨ The values of the curve points are stored in the global variable list *PointsList_AxisName*.

⇨ A segment with linear interpolation is configured in the curve description in the global variable list *CamDescriptionScheduler_AxisName*.

Writing a curve point table from the IEC program

Curve point tables can also be written from the IEC program by means of pointer access. Both types of curve point tables ("PointsList" and "SplinePointsList") provide a pointer to the internal array in which the curve point pairs are stored.

Proceed as follows to read the curve point table from the IEC program using this pointer:

1. Declare a local variable *pastPoints* of the type POINTER TO ARRAY [0..1024] (for writing a "PointsList") or of the type POINTER TO ARRAY [0..64] (for writing a "PointsListSpline"):
   ⇨ `POINTER TO ARRAY [0..1024] OF SEW_MK_MultiMotion-Cam.SEW_ICam.SEW_IPG_Base.ST_Point`
   ⇨ `POINTER TO ARRAY [0..64] OF SEW_MK_MultiMotion-Cam.SEW_ICam.SEW_IPG_Base.ST_Point`
2. Assign the corresponding property of the "PointsList" or "PointsListSpline" to the local variable.
   ⇨ `pastPoints:=PointsList_Achsname.pastPoints;`
3. Use the local variable to write curve point pairs in the "PointsList" or "PointsListSpline".
   ⇨
   ```
   pastPoints[0].lrX:=…
   pastPoints[0].lrY:=…
   pastPoints[1].lrX:=…
   pastPoints[1].lrY:=…
   ```
4. Specify the number of curve points.
   ⇨ `PointsList_Achsname.uiNumberOfPoints:=…`
5. Use the constructor of the required interpolation method to configure the appropriate segment in the curve description. There is a separate constructor for each interpolation method:
   ⇨ `itfConstructors4.itfLinearInterpolation.AddSegment(itf-PointsList)`
   ⇨ `itfConstructors4.itfPolynom3Interpolation.AddSegment(itf-PointsList)`
   ⇨ `itfConstructors4.itfSplineInterpolation.AddSegment(itf-PointsListSpline)`

Sample program:

The following is a program example in which curve points are assigned to a curve point table of the type "PointsList" and a linear interpolation is configured with the help of the corresponding constructor.

29179459/EN – 03/2020

The declaration part contains the following declarations:

User_PRG

```
PROGRAM User_PRG
VAR_OUTPUT
    xInitDone : BOOL;
END_VAR
VAR
    uiState:                UINT;
    pastPoints:             POINTER TO ARRAY [0..1024] OF SEW_MK_MultiMotionCam.SEW_ICam.SEW_IPG_Base.ST_Point;
    itfFreeCamDescription:  SEW_MK_MultiMotionCam.SEW_ICam.ICamDescription4;
END_VAR
```

31536711691

The following sequence is programmed in the "Main" action:

User_PRG.Main

```

CASE uiState OF

    0:
    IF Interface_Axis1.xInitDone THEN
        uiState := 10;
    END_IF

    10:
    pastPoints := PointsList_Axis1.pastPoints;

    pastPoints^[0].lrX := 0.0;
    pastPoints^[0].lrY := 0.0;
    pastPoints^[1].lrX := 0.125;
    pastPoints^[1].lrY := 12.5;
    pastPoints^[2].lrX := 0.25;
    pastPoints^[2].lrY := 50.0;
    pastPoints^[3].lrX := 0.375;
    pastPoints^[3].lrY := 87.5;
    pastPoints^[4].lrX := 0.5;
    pastPoints^[4].lrY := 100.0;
    pastPoints^[5].lrX := 0.625;
    pastPoints^[5].lrY := 87.5;
    pastPoints^[6].lrX := 0.75;
    pastPoints^[6].lrY := 50.0;
    pastPoints^[7].lrX := 0.875;
    pastPoints^[7].lrY := 12.5;
    pastPoints^[8].lrX := 1.0;
    pastPoints^[8].lrY := 0.0;

    PointsList_Axis1.uiNumberOfPoints := 9;

    itfFreeCamDescription := CamDescriptionScheduler_Axis1.itfFreeCamDescription4;
    IF (itfFreeCamDescription<>0) THEN
        uiState := 20;
    END_IF

    20:
    IF itfFreeCamDescription.itfConstructors4.itfLinearInterpolation.AddSegment (itfPointsList := PointsList_Axis1) THEN
        uiState := 30;
    END_IF

    30:
    ;

END_CASE
```

31536752523

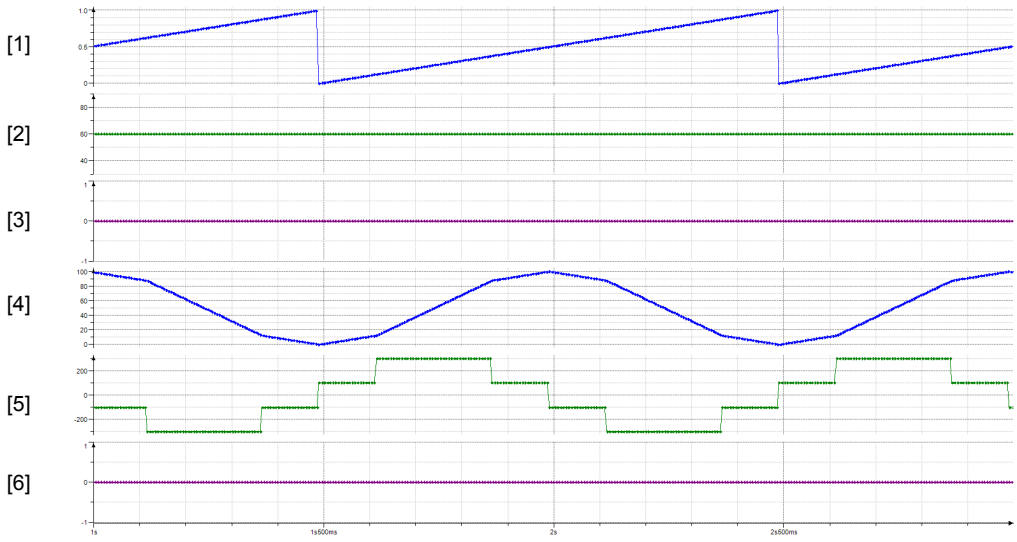
Move axis

Follow the steps described in chapter "Basic procedure" (→ 🗎 199) to move the axis with the curve profile interpolated from the curve point table.

Linear interpolation

If the interpolation method "LINEAR" is used to interpolate the curve point table, the following profile can be seen in the trace recording:

Trace recording



31495042699

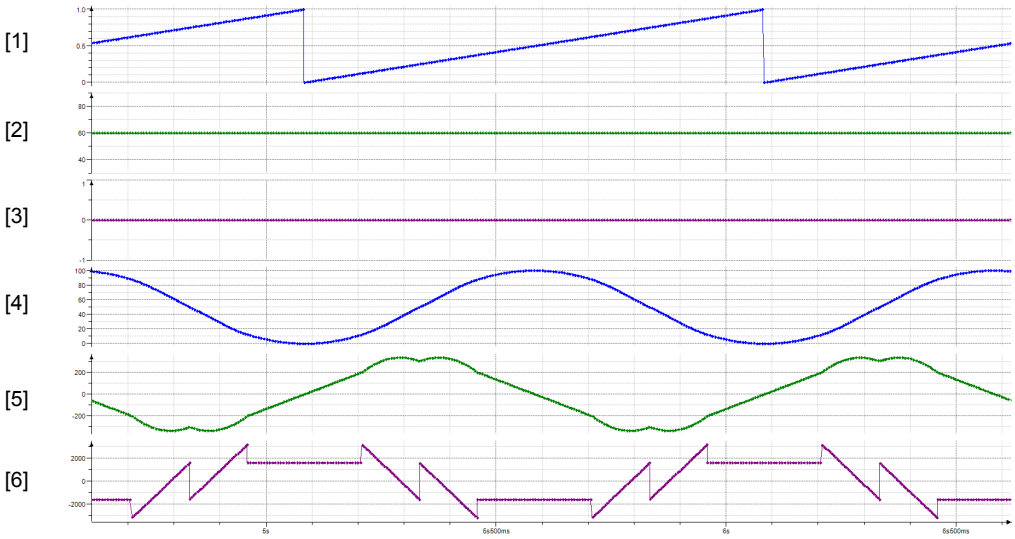
[1] Position of the master axis
[2] Speed of the master axis
[3] Acceleration of the master axis
[4] Position of the slave axis
[5] Speed of the slave axis
[6] Acceleration of the slave axis

29179459/EN – 03/2020

Polynom_3 interpolation

If the interpolation method "POLYNOM_3" is used to interpolate the curve point table, the following profile can be seen in the trace recording:

Trace recording



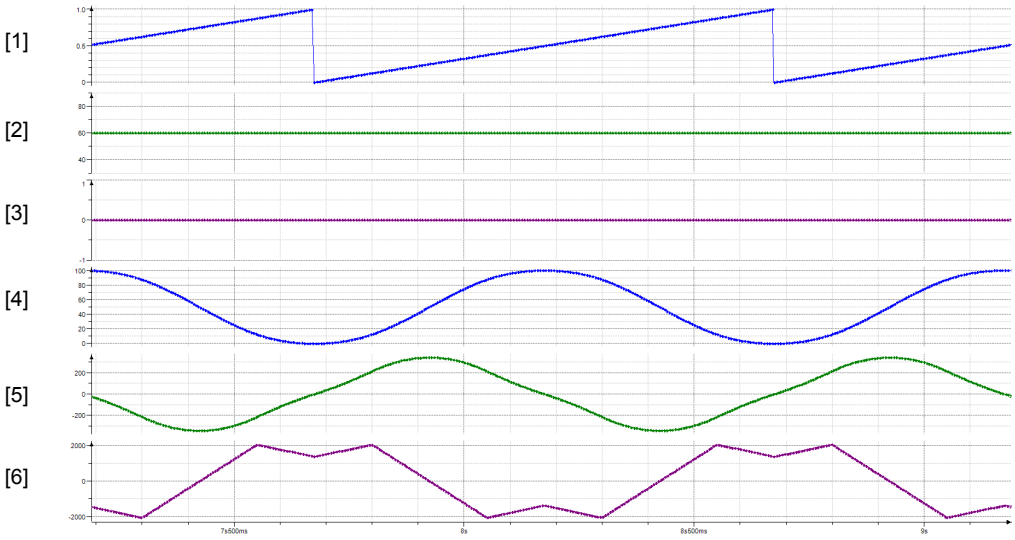
31495066763

[1] Position of the master axis
[2] Speed of the master axis
[3] Acceleration of the master axis
[4] Position of the slave axis
[5] Speed of the slave axis
[6] Acceleration of the slave axis

Spline interpolation

If the interpolation method "SPLINE" is used to interpolate the curve point table, the following profile can be seen in the trace recording:

Trace recording



31495076747

[1] Position of the master axis
[2] Speed of the master axis
[3] Acceleration of the master axis
[4] Position of the slave axis
[5] Speed of the slave axis
[6] Acceleration of the slave axis

29179459/EN – 03/2020

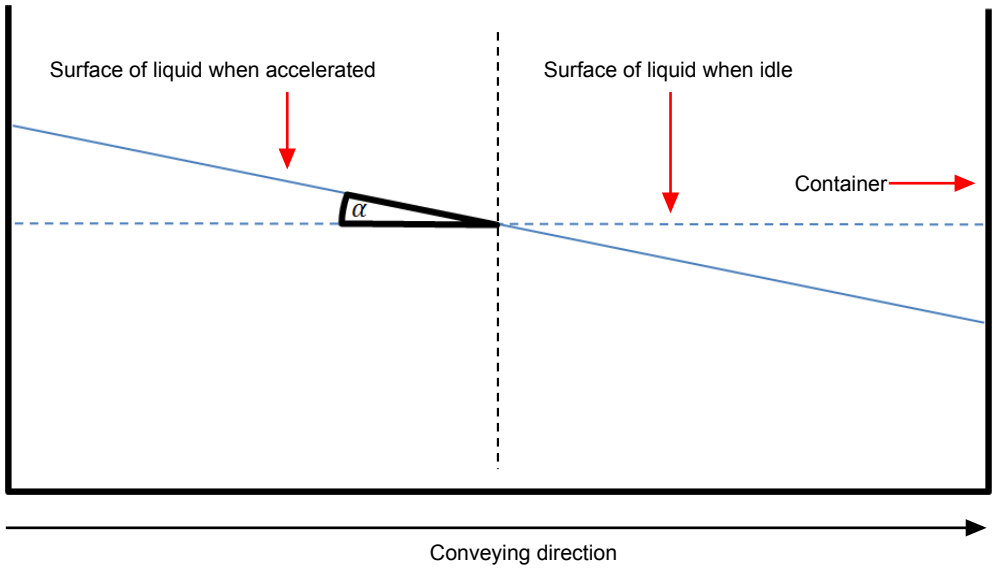
8.6.10 Reducing the oscillation of liquids

The following application example shows how you can use the "AntiSlosh" add-on to generate travel profiles for reducing oscillations during cyclical movements of liquids. See also "AntiSlosh" (→ 🗎 48).

Determining model parameters

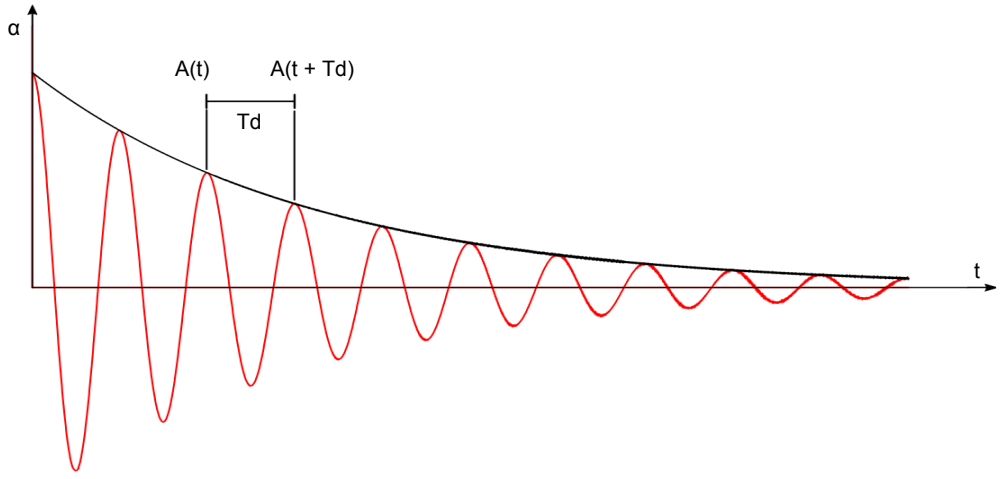
To configure the "AntiSlosh" add-on, you first have to determine the model parameters "Undamped natural frequency f_0", "Damping D", and "Nominal speed of the application". Model parameters are determined as follows:

1. Determine the parameters "Natural frequency f_0" and "Damping D" using the impulse response of the "container-liquid" system. To do this, measure the "Angle of inclination α" of the liquid surface.



31503056395

2. Based on the measured impulse response, determine the oscillation period T_d of the damped system. The first maxima belonging to the settling process are not included in the evaluation. T_d is given directly by the distance between two maxima.



31503060235

3. Using the "Logarithmic decrement Λ", calculate the "Decay constant δ" and the "Damping factor D" using the following formulae:

$$\Lambda = \ln \frac{A(t)}{A(t + T_d)}$$

31503148939

$$\delta = \frac{\Lambda}{T_d}$$

31503144075

$$D = \frac{1}{\sqrt{1 + \left(\frac{2\pi}{\Lambda}\right)^2}}$$

31503146507

4. Use the following formula to calculate the "Natural frequency of the undamped system f_0":

$$f_0 = \frac{\delta}{2\pi D}$$

31503432971

5. Use the following formula to calculate the "Nominal application speed" (the speed of the master axis v_{Master} must be configured in user units per second):

$$\frac{v_{Master}}{ZykluslängeMaster} \times \frac{60s}{min} = \frac{v_{Master}}{ModuloMax - ModuloMin} \times \frac{60s}{min}$$

31503435403

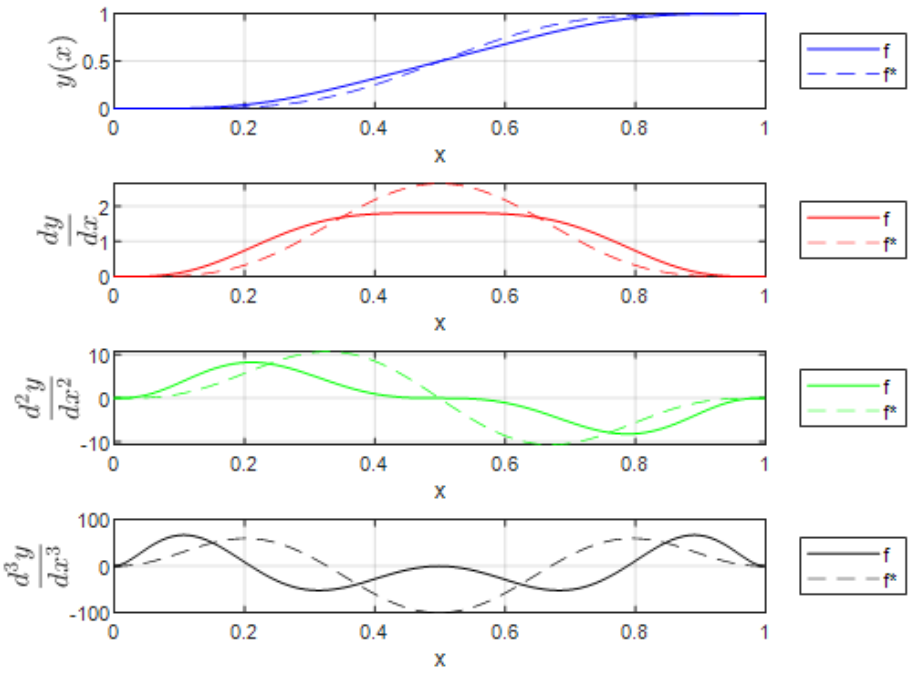
Activating and configuring the AntiSlosh function

1. In the configuration of MOVIKIT® MultiMotion Camming, open the "Basic settings" configuration menu and activate the "AntiSlosh" add-on under "Functions used".
2. In the configuration of MOVIKIT® MultiMotion Camming, open the "AntiSlosh" (→ 🗎 48) configuration menu.
3. Enter your model parameters in the "General parameters" section. See also "Determining model parameters" (→ 🗎 243).
4. You specify the required profile using the parameters in the "Travel profile parameters" section. The parameters "End of cycle", "Starting point of movement", "Distance of movement [ΔX2]" and "Stroke of movement [ΔY]" result from the boundary conditions of the application.

29179459/EN – 03/2020

5. To optimize the AntiSlosh function, adjust the travel profile using the normalized parameters "Jerk in the middle of movement", "Snap at the starting point of movement", and "Snap at the end point of movement". Note the following:

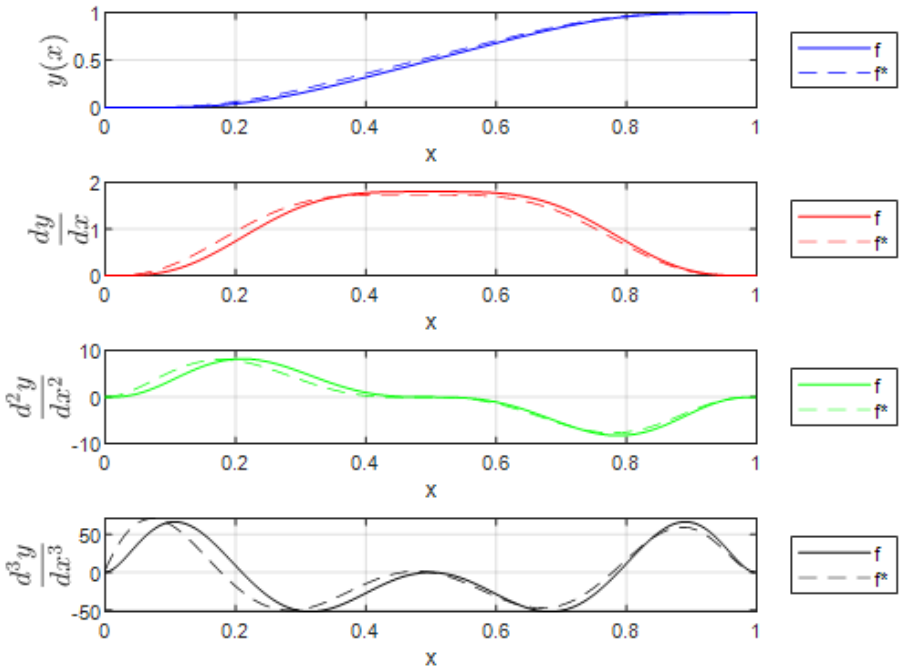
   ⇨ If "Snap at the starting point of movement" and "Snap at the end point of movement" are 0, favorable values for "Jerk in the middle of movement" are between 0 and -100.



31503485835

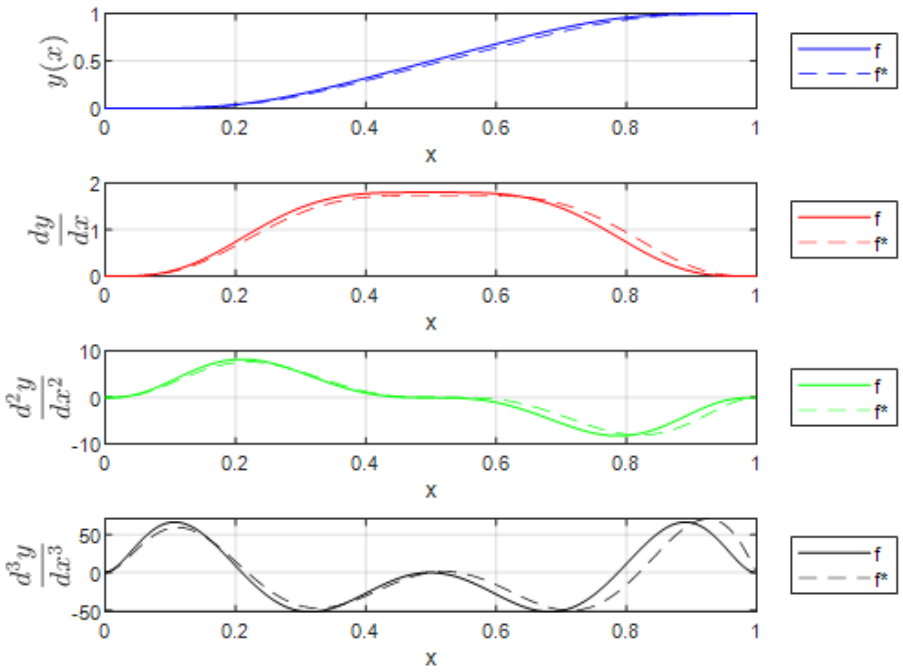
   ⇨ If "Snap at the starting point of movement", "Snap at the end point of movement" and "Jerk in the middle of movement " are 0, the acceleration in the middle of the cycle is 0.

   ⇨ The favorable value range for "Snap at the starting point of movement" is between 0 and 2000.



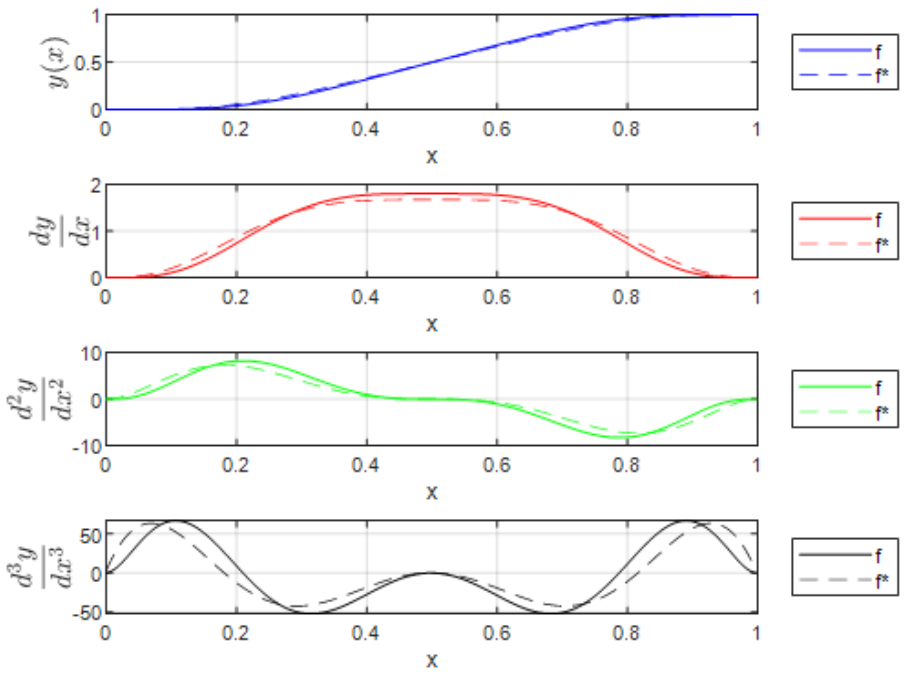
31503488267

⇨ The favorable value range for "Snap at the end point of movement" is between -2000 and 0.



31503493131

⇨ If for "Snap at the end point of movement" = - "Snap at the starting point of movement" = -2000 is selected, the following profile results.



31503490699

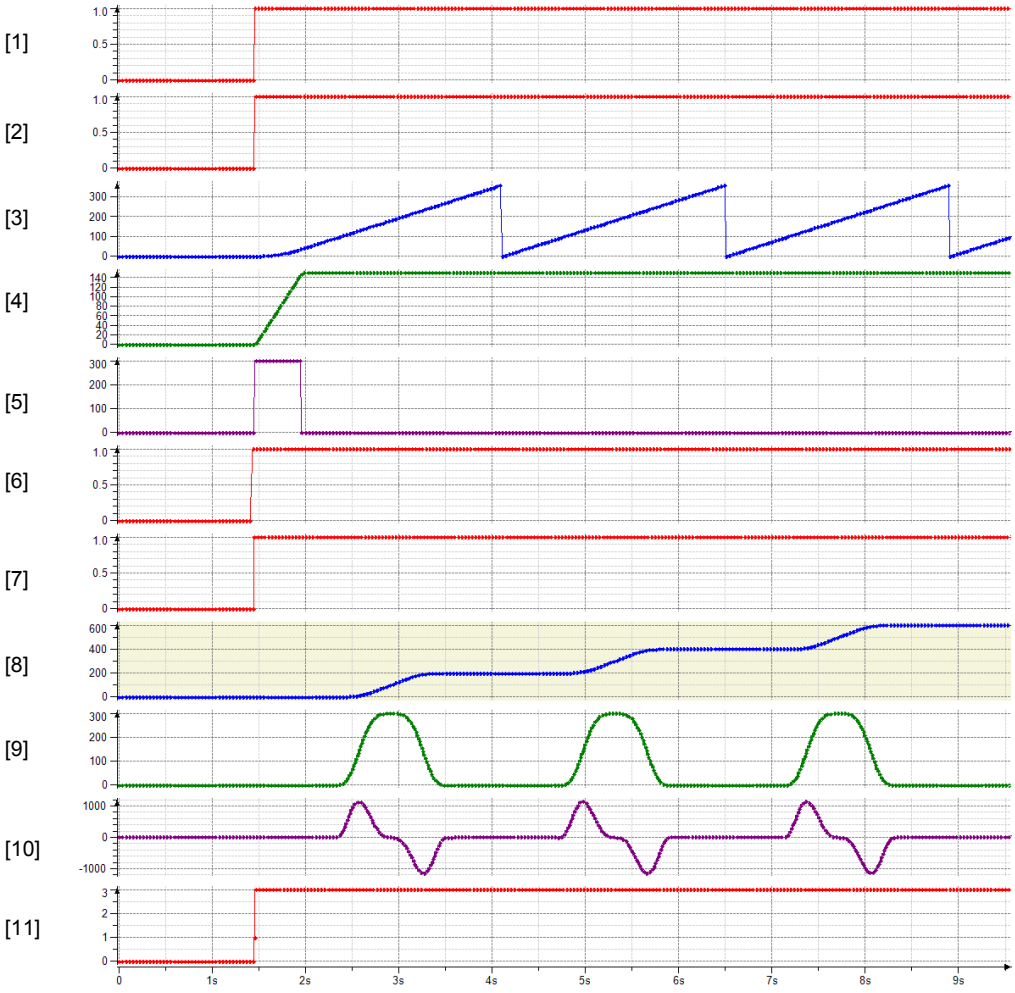
6. Update the configuration data and the IEC project on the memory card of the MOVI-C® CONTROLLER. See "Generating an IEC project" (→ 🗎 55).
   ⇨ The code generation creates the necessary code to initialize the function.
   ⇨ The *CamDescription[1]* of the *CamDescriptionScheduler* of the corresponding axis then contains the configured AntiSlosh profile as *CamSegment*.

Move axis

Follow the steps described in chapter "Basic procedure" (→ 🗎 199) to move the axis with the adjusted curve profile.

The slave axis follows the master axis with the configured AntiSlosh profile. The trace recording shows the following:

Trace recording



31504244491

[1] SEW_GVL.Interface_MasterAxis.ProfileGeneration.Velocity.In.xActivate
[2] SEW_GVL.Interface_MasterAxis.ProfileGeneration.Velocity.In.xStart
[3] SEW_GVL_Internal.MasterAxis._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[4] SEW_GVL_Internal.MasterAxis._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[5] SEW_GVL_Internal.MasterAxis._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[6] SEW_GVL.Interface_SlaveAxis.ProfileGeneration.Camming.IN.xActivate
[7] SEW_GVL.Interface_SlaveAxis.ProfileGeneration.Camming.IN.xStart
[8] SEW_GVL_Internal.SlaveAxis._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[9] SEW_GVL_Internal.SlaveAxis._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[10] SEW_GVL_Internal.SlaveAxis._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[11] SEW_GVL.Interface_SlaveAxis.ProfileGeneration.Camming.OUT.eGearingState

29179459/EN – 03/2020

Changing the AntiSlosh profile on the fly

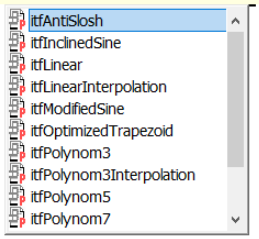
Like other curve profiles, the AntiSlosh profile can be changed on the fly during runtime. To create a new profile, the constructor mechanism known from the *Camming* operating mode is used. See "Creating a curve description using constructors" (→ 🗎 230). The changeover is performed as described in chapter "Switching curve description on the fly" (→ 🗎 236).

INFORMATION

Changeover via the variable *xGetNewCamDescription* may only be performed once the constructor has returned the value "TRUE".

Selection of the mathematical function:

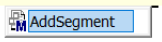
```
CamDescriptionScheduler_SlaveAxis.itfFreeCamDescription4.itfConstructors4.
```



31504376331

Selection of the constructor:

```
CamDescriptionScheduler_SlaveAxis.itfFreeCamDescription4.itfConstructors4.itfAntiSlosh.
```



31504373899

The required structure is provided by the Camming Interfaces library and includes all the parameters necessary for configuration.

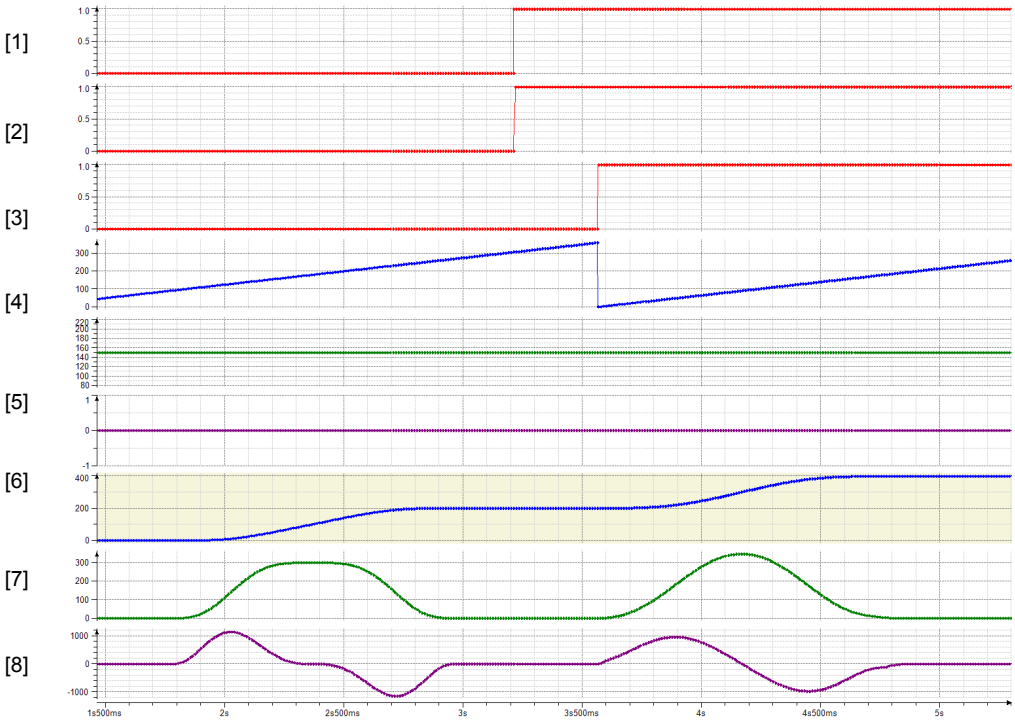
```
stMyAntiSloshConfig : SEW_MK_MultiMotionCam.SEW_ICam.ST_AntiSloshConfig2 := (
                                                                            lrSnapStart := 600.0,
                                                                            lrSnapEnd := -600.0,
                                                                            lrDamping := 0.3,
                                                                            lrJerkMid := -47.0,
                                                                            lrNaturalFrequency := 3.333,
                                                                            lrStepDistance := 200.0,
                                                                            lrVelocity := 25.0,
                                                                            lrCycleStart := 0.0,
                                                                            lrCycleEnd := 360.0,
                                                                            lrDeltaX2 := 180.0
                                                                            );
```

31505634187

29179459/EN – 03/2020

Trace recording

The following profile results from trace recording:



31505640075

[1] Controller.Application.User_PRG.MAIN.xConfigOK
[2] Controller.Application.User_PRG.MAIN.Interface_SlaveAxis.ProfileGeneration.Camming.IN.xGetNewCamDescription
[3] Controller.Application.User_PRG.MAIN.Interface_SlaveAxis.ProfileGeneration.Camming.OUT.xGetNewCamDescriptionDone
[4] SEW_GVL_Internal.MasterAxis._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[5] SEW_GVL_Internal.MasterAxis._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[6] SEW_GVL_Internal.SlaveAxis._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[7] SEW_GVL_Internal.SlaveAxis._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[8] SEW_GVL_Internal.SlaveAxis._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration

29179459/EN – 03/2020

8.7 Anti-sway control

The following application example illustrates how to use the add-on "Anti-sway control" (→ 🗎 15) (MOVIKIT® Motion add-on AntiSway). You activate the add-on in the configuration menu "Basic settings" of the software module in the "Functions used" section. When the add-on is activated, the corresponding configuration menus are displayed in MOVISUITE® and the corresponding structures are created when generating an IEC project.

8.7.1 Defining the application type

1. In the configuration of the software module, open the "Anti-sway control" (→ 🗎 50) configuration menu.
2. Select the required "Application type" from the "Anti-sway control" section.
   ⇨ The setting fields for configuring the selected application type are displayed.
3. Specify a "Conversion factor for user units in meters". This step is necessary because the function works in physical units. If you have already selected meters as the user unit, you can deactivate conversion by setting the value "0".

29179459/EN – 03/2020

8.7.2 Configuring the application type

Determining stiffness

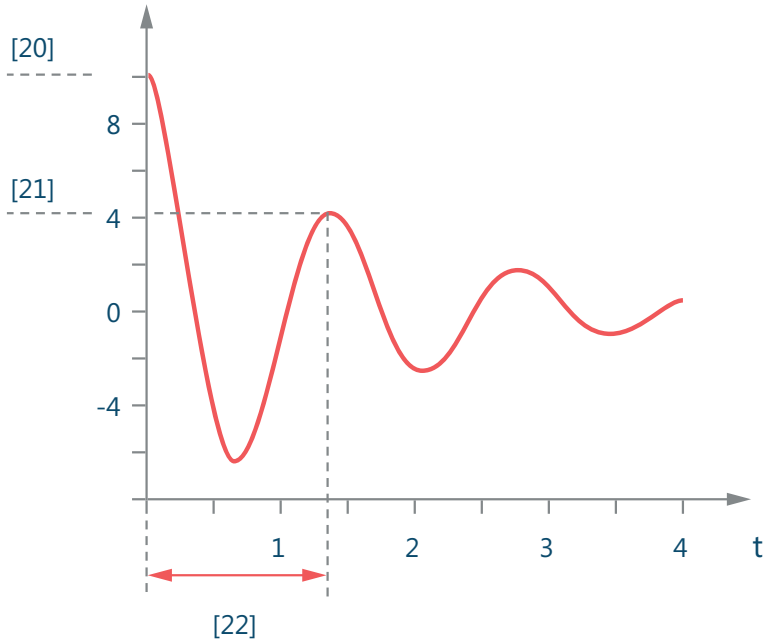
INFORMATION

The calculations in the configuration menu "Support for parameter determination" (→ 🗎 52) are based, among others, on the values entered in the "Anti-sway control" (→ 🗎 50) configuration menu.

The configuration of some application types includes the definition of the parameter "Spring stiffness between tower and trolley". If an application type is used that does not require this parameter, you can skip this step.

If you do not know these values, they can be determined from the signal waveform of the oscillating system by entering certain measured values in the configuration menu "Support for parameter determination" (→ 🗎 52).

1. Record the amplitudes of the oscillating system e.g. by means of a video recording at the point of greatest oscillation. If there is enough torque back to the motor from the oscillating mass, a scope recording of the torque can also be used.



No.	Field in the configuration menu
[20]	Oscillation amplitude (1st peak)
[21]	Oscillation amplitude (2nd peak)
[22]	Oscillation period (time between 1st and 2nd oscillation amplitude)

2. Enter the determined measured values (see diagram) into the corresponding setting fields in the configuration menu "Support for parameter determination".
   ⇨ The values "Damping ratio between tower and trolley" and "Spring stiffness between tower and trolley" are calculated and used directly for anti-sway control.

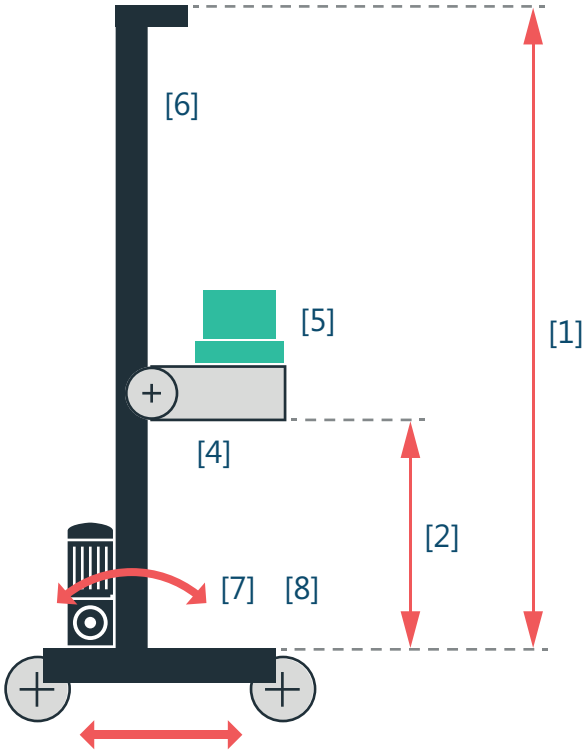
29179459/EN – 03/2020

⇨ The natural frequency, resonance frequency and "Deflection at lifting height" are determined for plausibility checks against reality and design calculations.

3. Carry out a plausibility check of the calculated values together with the data provider.

TowerSway

Configure your application in the "Anti-sway control" configuration menu by entering values for the parameters shown in the following diagram. For more information on the parameters, refer to the chapters "Anti-sway control" (→ 🗎 50) and "IEC programming" (→ 🗎 89).



31521023755

No.	Field in the configuration menu	IEC name
[1]	Height of the mast	IrHeightTower
[2]	Distance between lifting and traveling trolley	IrDistanceHoistToCar
[4]	Mass of the trolley	IrMassHoist
[5]	Mass of the payload	IrMassPayload
[6]	Mass of the mast	IrMassTower
[7]	Spring stiffness between tower and trolley **Information:** To determine this parameter, refer to chapter "Determining stiffness" (→ 🗎 252).	IrSpringTowerToCar
[8]	Degree of damping between tower and trolley **Information:** To determine this parameter, refer to chapter "Determining stiffness" (→ 🗎 252).	IrDampTowerToCar

8.7.3 Transferring the configuration

Use automatic code generation to generate an IEC project and load the project and the modified configuration to the MOVI-C® CONTROLLER. See chapter "Generating an IEC project" (→ 🗎 55).

8.7.4 Controlling/monitoring a function

The setting fields in the configuration menus in MOVISUITE® or variables in the IEC project explained in the following chapters are available for controlling, adapting and monitoring anti-sway control.

TowerSway

- Switch on/off the function via "Selection of setpoint correction" (IEC: *eSelector*)

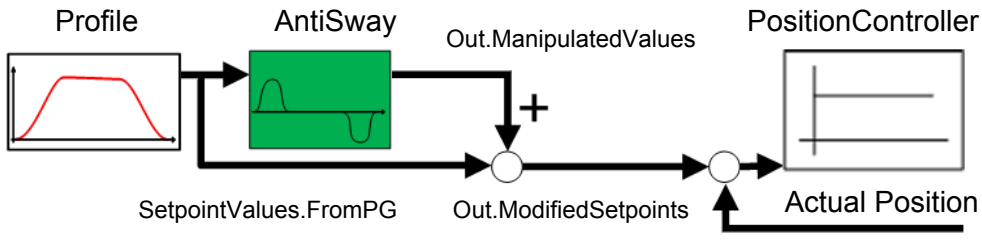
  Switch the function on or off during runtime at standstill.
- Adjust the lifting height via "Distance between lifting and traveling trolley" (IEC: *lrDistanceHoistToCar*)

  If the lifting axis has a considerable influence on the oscillation behavior of the travel axis, it is recommended to specify the height of the lifting axis continuously. It is important that this happens cyclically in the*HighPrio* task. The following guide values can be assumed to have a considerable influence: "Mass of lifting trolley" (*lrMassHoist*) > ⅓ × "Mass of master" (*lrMassTower*) Variance *lrDistanceHoistToCar* > ¼ "Height of tower" (*lrHeightTower*)
- Adjust the payload via "Mass of payload" (IEC: *lrMassPayload*)

  If the payload has a considerable influence on the oscillation behavior, it is recommended to set the payload as a static value prior to each positioning process. The following guide values can be assumed to have a considerable influence: Variance *lrMassPayload* > ⅓ mass of lifting gear (*lrMassHoist*)
- Terminate travel order via IEC: *xBusy*

  To continue the control chain, *xBusy* must report "FALSE". *xBusy* takes into account all changes triggered by anti-sway control compared to the original travel profile.

The variables illustrated in the following diagram are particularly suitable for monitoring the function (e.g. using trace recording):



29179459/EN – 03/2020

8.8 Operating mode examples

8.8.1 On-the-fly changeover from velocity to positioning with linear axis

The following application example illustrates how you can implement an on-the-fly changeover of the operating mode, where one operating mode replaces all others. The original operation mode is deactivated at the same time as the new operation mode is activated. For an on-the-fly changeover within the same cycle, set control signals *xActivate* and *xStart* in operating mode "Velocity" to "FALSE" and set control signals *xActivate* and *xStart* in operating mode "Positioning" to "TRUE".

Control

Control the axis in the *Velocity* structure as follows:

- To activate the operating mode, set *xActivate* to "TRUE".
- Define the setpoints for *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk*.
- To start the operating mode, set *xStart* to "TRUE".

Control the axis in the *Positioning* structure as follows:

- Define the setpoints for *lrPosition*, *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk* in the *Positioning* structure.
- Set *ePresetMode* to "ACTPOS_MOVING" in the *CONFIG* structure.

Control the axis for an on-the-fly changeover in the same cycle as follows:

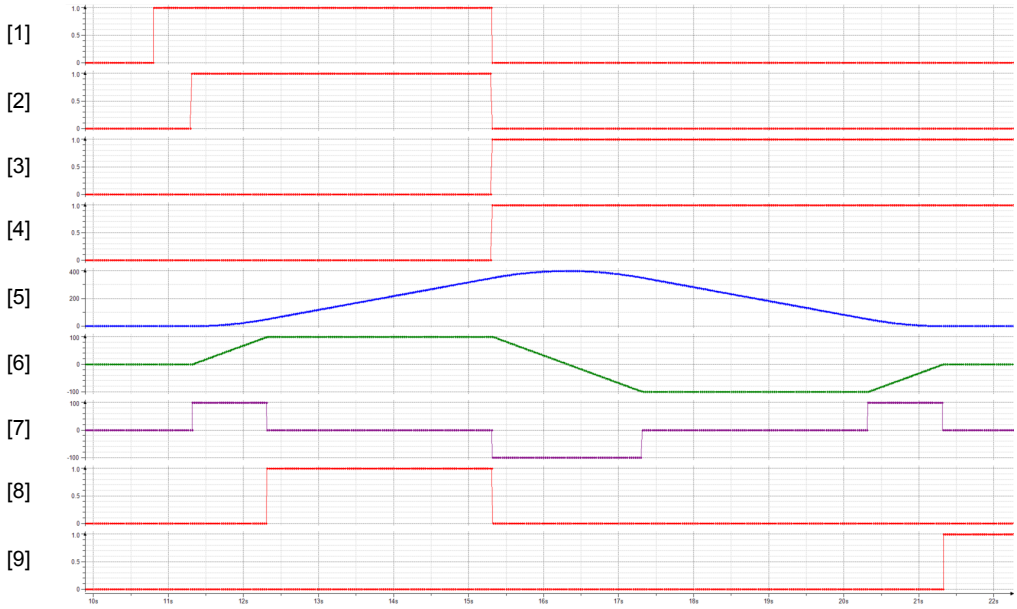
- Set *xActivate* and *xStart* to "FALSE" in the *Velocity* structure, and set *xActivate* and *xStart* to "TRUE" in the *Positioning* structure.

Interface in the IEC Editor

Velocity	SEW_MOS_UI_InterpolationModes.MC_UI_ModeVelocity	
IN	SEW_MOS_IInterpolationModes.ST_Velocity_IN	
xActivate	BOOL	FALSE
xStart	BOOL	FALSE
lrVelocity	LREAL	100
lrAcceleration	LREAL	100
lrDeceleration	LREAL	100
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0
CONFIG	SEW_MOS_IInterpolationModes.ST_Velocity_Config	
OUT	SEW_MOS_IInterpolationModes.ST_Velocity_OUT	
Positioning	SEW_MOS_UI_InterpolationModes.MC_UI_ModePositioning	
IN	SEW_MOS_IInterpolationModes.ST_Positioning_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
lrPosition	LREAL	0
lrVelocity	LREAL	100
lrAcceleration	LREAL	100
lrDeceleration	LREAL	100
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0
CONFIG	SEW_MOS_IInterpolationModes.ST_Positioning_Config	
eModuloMode	E_MODULOMODE	MODULO_OFF
stPresetValues	ST_ProfGen_PresetValues	
ePresetMode	E_PRESETMODE	ACTPOS_MOVING
xTargetPositionMonitoring	BOOL	FALSE
xWithoutReferencedEncoder	BOOL	FALSE

20735786379

Trace recording



25424672779

[1] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.In.xActivate
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.In.xStart
[3] SEW_GVL.Interface_Axis2.ProfileGeneration.Positioning.In.xActivate
[4] SEW_GVL.Interface_Axis2.ProfileGeneration.Positioning.In.xStart
[5] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[8] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.Out.xInVelocity
[9] SEW_GVL.Interface_Axis2.ProfileGeneration.Positioning.Out.xInPosition

Comment:

For an on-the-fly changeover, it is important that the change to the control signals is detected by a PLC cycle of "TaskHighPrio". When controlling via the *Interface_AxisName* structure, make sure to integrate the mapping methods in "TaskHighPrio".

8.8.2 On-the-fly changeover from velocity to positioning with modulo axis

The following application example illustrates how you can implement an on-the-fly changeover of the operating mode, where one operating mode replaces all others. The original operating mode is deactivated at the same time as the new operating mode is activated. For an on-the-fly changeover within the same cycle, set control signals *xActivate* and *xStart* in operating mode "Velocity" to "FALSE" and set control signals *xActivate* and *xStart* in operating mode "Positioning" to "TRUE".

Control

Control the axis in the *Velocity* structure as follows:

- To activate the operating mode, set *xActivate* to "TRUE".
- Define the setpoints for *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk*.
- To start the operating mode, set *xStart* to "TRUE".

Control the axis in the *Positioning* structure as follows:

- Define the setpoints for *lrPosition*, *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk* in the *Positioning* structure.
- Set *eModuloMode* to "MODULO_CW" in the *CONFIG* structure.
- Set *ePresetMode* to "ACTPOS_MOVING" in the *CONFIG* structure.

Control the axis for an on-the-fly changeover in the same cycle as follows:

- Set *xActivate* and *xStart* to "FALSE" in the *Velocity* structure, and set *xActivate* and *xStart* to "TRUE" in the *Positioning* structure.

Interface in the IEC Editor

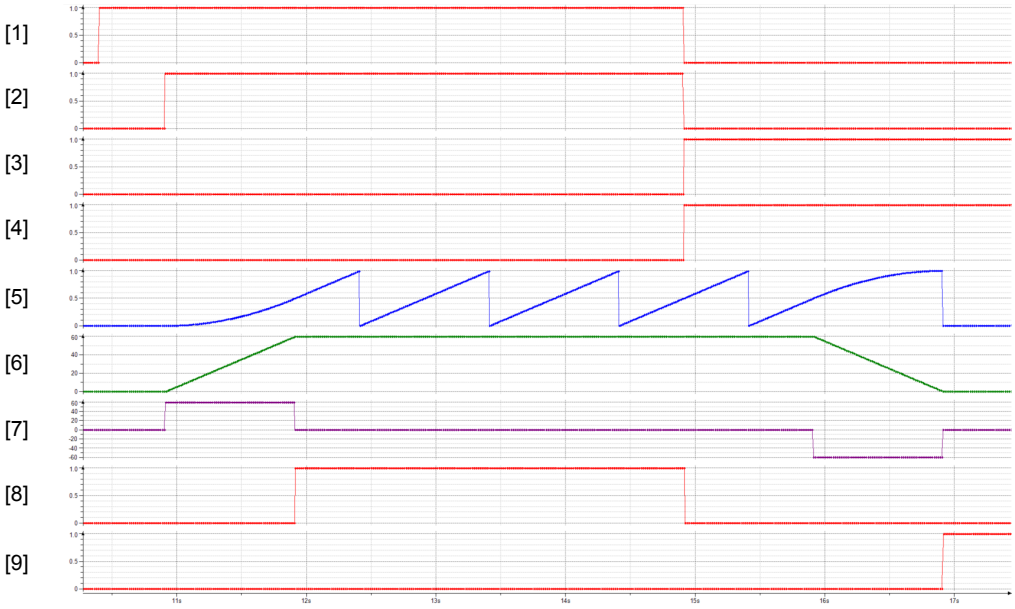
Velocity	SEW_MOS_UI_InterpolationModes.MC_UI_ModeVelocity	
IN	SEW_MOS_IInterpolationModes.ST_Velocity_IN	
xActivate	BOOL	FALSE
xStart	BOOL	FALSE
lrVelocity	LREAL	60
lrAcceleration	LREAL	60
lrDeceleration	LREAL	60
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0
CONFIG	SEW_MOS_IInterpolationModes.ST_Velocity_Config	
OUT	SEW_MOS_IInterpolationModes.ST_Velocity_OUT	
Positioning	SEW_MOS_UI_InterpolationModes.MC_UI_ModePositioning	
IN	SEW_MOS_IInterpolationModes.ST_Positioning_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
lrPosition	LREAL	0
lrVelocity	LREAL	60
lrAcceleration	LREAL	60
lrDeceleration	LREAL	60
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0
CONFIG	SEW_MOS_IInterpolationModes.ST_Positioning_Config	
eModuloMode	E_MODULOMODE	MODULO_CW
stPresetValues	ST_ProfGen_PresetValues	
ePresetMode	E_PRESETMODE	ACTPOS_MOVING
xTargetPositionMonitoring	BOOL	FALSE
xWithoutReferencedEncoder	BOOL	FALSE

20735835275

Comment:

Unlike with linear axes, the behavior in this case depends on the modulo mode that has been set (in this example: "MODULO_CW"):

Trace recording



25424680459

[1] SEW_GVL.Interface_Axis1.ProfileGeneration.Velocity.In.xActivate
[2] SEW_GVL.Interface_Axis1.ProfileGeneration.Velocity.In.xStart
[3] SEW_GVL.Interface_Axis1.ProfileGeneration.Positioning.In.xActivate
[4] SEW_GVL.Interface_Axis1.ProfileGeneration.Positioning.In.xStart
[5] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[6] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[7] SEW_GVL_Internal.Axis1._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[8] SEW_GVL.Interface_Axis1.ProfileGeneration.Velocity.Out.xInVelocity
[9] SEW_GVL.Interface_Axis1.ProfileGeneration.Positioning.Out.xInPosition

Comment:

The movement stops at the specified position (in this case at the zero position), similar to *StopAtPosition*.

29179459/EN – 03/2020

8.8.3 Overlapping of the Velocity and PositioningRelative operating modes

The following application example illustrates how you can activate multiple operating modes simultaneously (operating mode "Velocity" and operating mode "Positioning Relative"). This allows you to overlap motion profiles.

Control

Control the axis in the *Velocity* structure as follows:

- To activate the operating mode, set *xActivate* to "TRUE".
- Define the setpoints for *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk*.
- To start the operating mode, set *xStart* to "TRUE".

Control the axis in the *PositioningRelative* structure as follows:

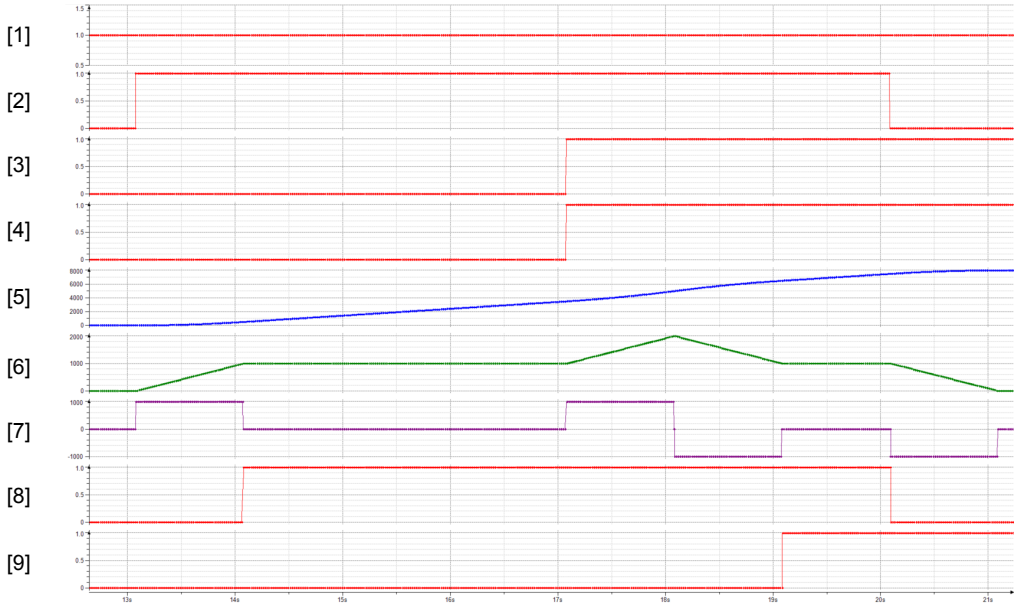
- To activate the operating mode, set *xActivate* to "TRUE".
- Define the setpoints for *lrDistance*, *lrVelocity*, *lrAcceleration*, *lrDeceleration*, and *lrJerk*.
- To start the operating mode, set *xStart* to "TRUE".

Interface in the IEC Editor

Velocity	SEW_MOS_UI_InterpolationModes.MC_UI_ModeVelocity	
IN	SEW_MOS_IInterpolationModes.ST_Velocity_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
lrVelocity	LREAL	1000
lrAcceleration	LREAL	1000
lrDeceleration	LREAL	1000
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0
CONFIG	SEW_MOS_IInterpolationModes.ST_Velocity_Config	
OUT	SEW_MOS_IInterpolationModes.ST_Velocity_OUT	
Positioning	SEW_MOS_UI_InterpolationModes.MC_UI_ModePositioning	
PositioningRelative	SEW_MOS_UI_InterpolationModes.MC_UI_ModePositioningRelative	
IN	SEW_MOS_IInterpolationModes.ST_PositioningRelative_IN	
xActivate	BOOL	TRUE
xStart	BOOL	TRUE
xAcceptNewDistance	BOOL	FALSE
lrDistance	LREAL	1000
lrVelocity	LREAL	1000
lrAcceleration	LREAL	1000
lrDeceleration	LREAL	1000
lrJerk	LREAL	0
xRapidStop	BOOL	FALSE
lrRapidStopDeceleration	LREAL	0

20735954315

Trace recording



25424718859

[1] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.In.xActivate
[2] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.In.xStart
[3] SEW_GVL.Interface_Axis2.ProfileGeneration.PositioningRelative.In.xActivate
[4] SEW_GVL.Interface_Axis2.ProfileGeneration.PositioningRelative.In.xStart
[5] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrPosition
[6] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrVelocity
[7] SEW_GVL_Internal.Axis2._fbProfileGenerationControl._stOUTInterpolation.stSetpointsUserUnits.lrAcceleration
[8] SEW_GVL.Interface_Axis2.ProfileGeneration.Velocity.Out.xInVelocity
[9] SEW_GVL.Interface_Axis2.ProfileGeneration.PositioningRelative.Out.xInPosition

Comment:

As can be seen in the speed profile, in particular, the "PositioningRelative" operating mode overlaps with the "Velocity" operating mode in the area between Cursor1 and Cursor2.

29179459/EN – 03/2020

9 Fault management

9.1 Fault codes

9.1.1 Camming

Code	Meaning
16#7400	One or more parameters of the curve segments could not be found.
16#7401	Import of the curve description caused an error; see ConfigHandling for details.
16#7402	The number of imported curve segments is outside the permitted range (1 – 20).
16#7403	The interface to ICurveFileList is not connected.
16#7404	The interface to ICamSegment is not connected.
16#7405	The interface to IAxisConfig is not connected.
16#7406	The interface to ICammingProfile is not connected.
16#7407	The interface to CamDescription is not connected.
16#7408	The interface to IUnitCalculations is not connected.
16#7409	The interface to IConfigDataHandler is not connected.
16#740A	The interface to IInterpolationModeSelection is not connected.
16#740B	The interface to the master axis is not connected.
16#740C	The interface to CamDescriptionScheduler is not connected.
16#740D	The interface to CamDescription and to CamDescriptionScheduler is not connected.

9.1.2 ConfigDataHandling

Code	Meaning
16#6600	The configuration file with the specified name could not be found.
16#6601	The configuration file with the specified name could not be opened.
16#6602	The configuration file could not be closed again. File access is not complete.
16#6603	The configuration data could not be read from the file.
16#6604	One or more configuration parameters could not be found.
16#6605	The configuration parameter has no data or data length is too large.

29179459/EN – 03/2020

9.1.3 DeviceAdapter

Fault

Code	Meaning
16#6A00	Inverter error: The error number can be found in the "Inverter.Out" structure.

Warnings

Code	Meaning
16#195F0	One of the dynamic values transferred is too large (cannot be mapped using 16 bits).
16#195F1	One of the dynamic values transferred is too small (cannot be mapped using 16 bits).

9.1.4 ProfileGeneration

Code	Meaning
16#7600	The software limit switches (in PLC units) are outside the DINT range of numbers.
16#7601	The modulo limits (in PLC units) are outside the DINT range of numbers.
16#7602	The preset position (in PLC units) is outside the DINT range of numbers.
16#7603	The reference offset (in PLC units) is outside the DINT range of numbers.
16#7604	The reference offset is outside the modulo limits of a modulo axis.
16#7605	The homing target position (in PLC units) is outside the DINT range of numbers.
16#7606	The selected ModuloMode is not allowed for this axis type or is not supported by the offset profile.
16#7607	The activated offset profile requires a referenced axis.
16#7608	The target position (in PLC units) is outside the DINT range of numbers.
16#7609	The travel distance (in PLC units) is outside the DINT range of numbers.
16#760A	The target position is outside the software limit switches.
16#760B	The position to be approached when the velocity profile stops (in PLC units) is outside the DINT range of numbers.
16#760C	Tracking master resolution is outside the permitted limits.
16#760D	The tracking master modulo limits (in PLC units) are outside the DINT range of numbers.
16#760E	The tracking slave modulo limits (in PLC units) are outside the DINT range of numbers.
16#760F	Tracking numerator or denominator is outside the permitted limits.

29179459/EN – 03/2020

Code	Meaning
16#7610	Tracking master position is outside the permitted limits.
16#7611	Tracking master time base is outside the permitted limits.
16#7612	Tracking slave time base is outside the permitted limits.
16#7613	Positive software limit switch approached.
16#7614	Negative software limit switch approached.
16#7615	The deceleration is greater than the application limit.
16#7616	The acceleration is greater than the application limit.
16#7617	The speed (velocity) is greater than the positive application limit.
16#7618	The speed (velocity) is greater than the negative application limit.
16#7619	Missing interface connection
16#761A	System error during profile generation. For further information on enumeration, refer to the "SEW ProfGen JLimit Extern" library.
16#761B	Error importing configuration data
16#761C	Missing or invalid license

9.1.5 AntiSlosh

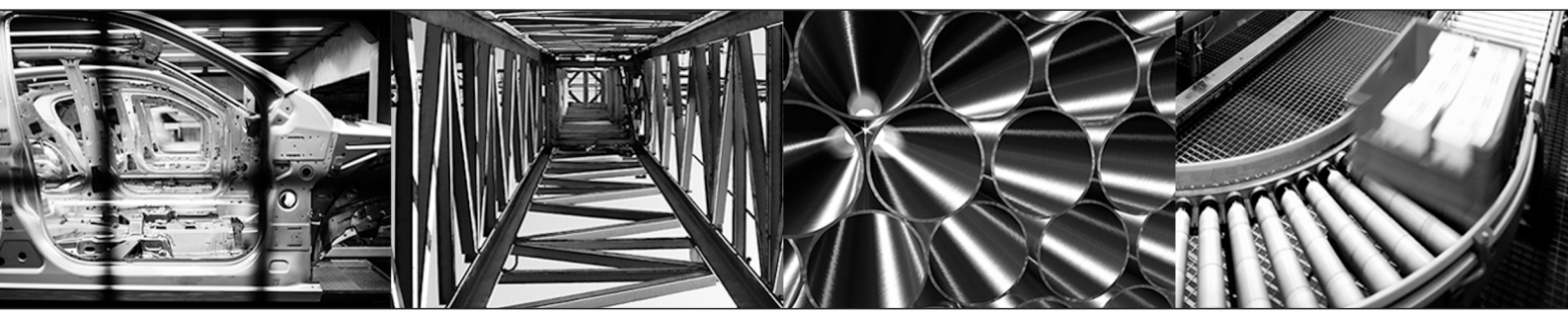
Code	Meaning
16#7700	A necessary interface is not connected.
16#7701	An invalid input parameter was transferred.
16#7702	Reading the configuration file has failed.
16#771F	An undefined error has occurred.

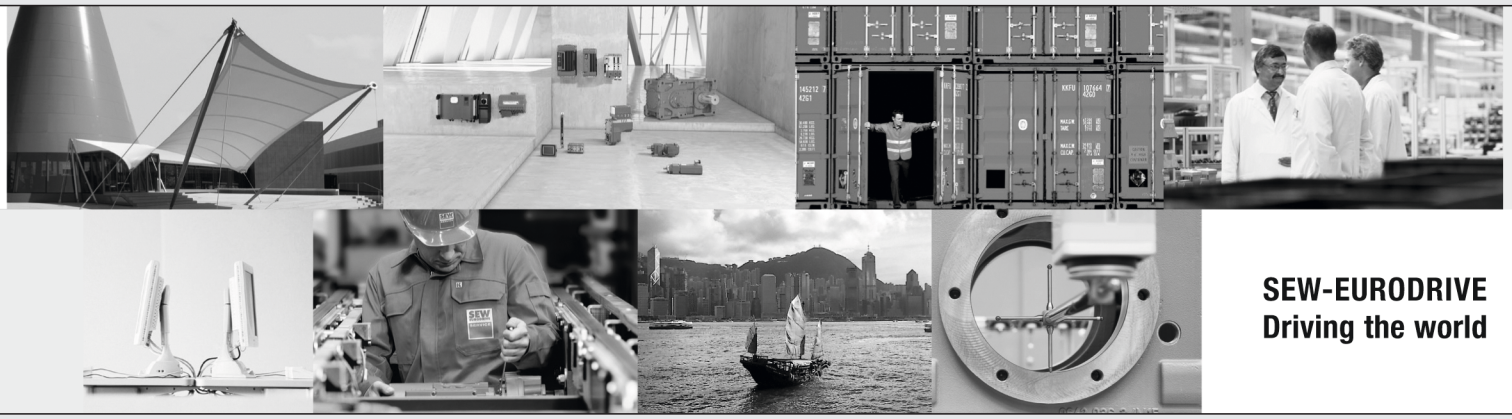
29179459/EN – 03/2020

Index

29179459/EN – 03/20

29179459/EN – 03/20





SEW-EURODRIVE
Driving the world

SEW-EURODRIVE GmbH & Co KG
Ernst-Blickle-Str. 42
76646 BRUCHSAL
GERMANY
Tel. +49 7251 75-0
Fax +49 7251 75-1970
sew@sew-eurodrive.com
➞ **www.sew-eurodrive.com**