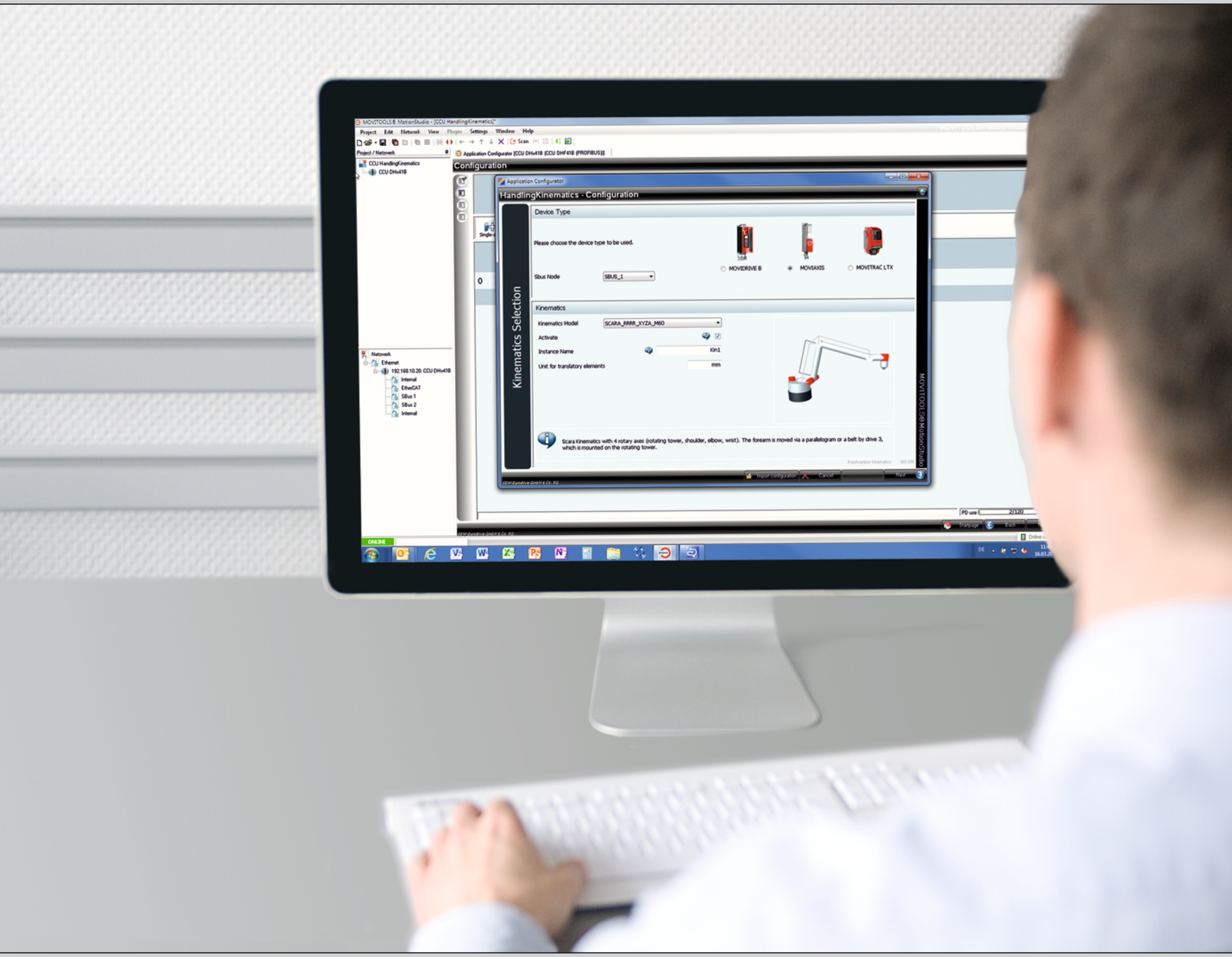
SEW EURODRIVE

Manual



CCU HandlingKinematics Application Module

Edition 12/2017

24796131/EN



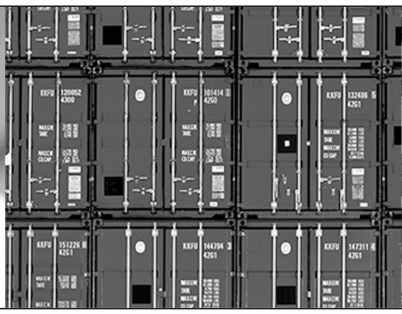



Table of contents

24796131/EN – 12/2017

24796131/EN – 12/2017

24796131/EN – 12/2017

24796131/EN – 12/2017

1 General information

1.1 About this documentation

The documentation is part of the product and contains important information. The documentation is for everyone who works with this product.

The documentation must be accessible and legible. Make sure that persons responsible for the system and its operation as well as persons who work independently with the software and the connected units of SEW-EURODRIVE have read through the manual carefully and understood it. If you are unclear about any of the information in this documentation or if you require further information, please contact SEW-EURODRIVE.

1.2 Structure of the safety notes

1.2.1 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 drive system or its environment.
INFORMATION	Useful information or tip: Simplifies handling of the drive system.	

1.2.2 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.

Meaning of the hazard symbols

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

Hazard symbol	Meaning
	General hazard

24796131/EN – 12/2017

Hazard symbol	Meaning
	Warning of dangerous electrical voltage
	Warning of hot surfaces
	Warning of risk of crushing
	Warning of suspended load
	Warning of automatic restart

1.2.3 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.

1.3 Right to claim under warranty

A requirement of fault-free operation and fulfillment of any rights to claim under limited warranty is that you adhere to the information in the documentation at hand. Therefore, read the documentation before you start working with the software and the connected units from SEW-EURODRIVE.

Make sure that the documentation is available to persons responsible for the machinery and its operation as well as to persons who work independently on the units. Also ensure that the documentation is legible.

1.4 Exclusion of liability

Please observe this documentation as well as the documentation for the software used and the SEW-EURODRIVE devices connected. This documentation must be observed to ensure that the devices operate safely and that the specified product properties and performance characteristics are achieved.

SEW-EURODRIVE assumes no liability for injury to persons or damage to equipment or property resulting from non-observance of the documentation. In such cases, SEW-EURODRIVE assumes no liability for defects.

1.5 Copyright notice

1.6 Product names and trademarks

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

1.7 Other applicable documentation

Observe the following applicable documentation for the controller and software:

- "Application Configurator for CCU" configuration software
- "Controller DHE21B / DHF21B / DHR21B (standard), DHE41B / DHF41B / DHR41B (advanced)" manual
- Communication manuals for the respective fieldbuses:
  - MOVI-PLC® advanced DHF41B Fieldbus Interfaces DeviceNet and PROFIBUS DP-V1
  - MOVI-PLC® advanced DHR41B Fieldbus Interfaces EtherNet/IP, Modbus/TCP and PROFINET IO

Also observe the following applicable documentation depending on the connected drive technology:

- "MOVIDRIVE® MDX Drive Inverter" operating instructions
- "MOVITRAC® LTX Frequency Inverter" operating Instructions
- "Multi-Axis Servo Inverter MOVIAXIS® MX" operating instructions
- "MOVIAXIS® Multi-Axis Servo Inverter – MXR Supply and Regenerative Module" manual

The HandlingKinematics application module uses the lower-level "AxisGroupControl Kinematics" technology module. The lower-level technology module is controlled automatically in the background. Although you can refer to the following manual to better understand the operating principle of the kinematic control and the fundamentals of robotics such as the operating modes, interpolation and properties, this manual is not required for use of the HandlingKinematics application module:

- "Kinematics Technology Module for MultiMotion/MultiMotion Light" manual (document no. 20211740 / EN).

24796131/EN – 12/2017

2 Safety notes

2.1 General

The following basic safety notes are intended to prevent injury to persons and damage to property. The user must ensure that the basic safety notes are read and observed.

Ensure that persons responsible for the machinery and its operation as well as persons who work 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, please contact SEW-EURODRIVE.

The following safety notes refer to the use of the software. Also observe the supplementary safety notes in this documentation and in the documentation for the connected units from SEW-EURODRIVE.

This document does not replace the detailed documentation for the connected units. This documentation assumes that the user has access to and is familiar with the documentation for all connected units from SEW-EURODRIVE.

Never install or operate damaged products. Report any damage to the shipping company immediately.

Depending on the degree of protection, units may have live, uninsulated, and sometimes moving or rotating parts, as well as hot surfaces during operation.

Removing required covers without authorization, improper use or incorrect installation and operation may result in severe injury to persons, or damage to machinery. Consult the documentation for further information.

2.2 Target group

Work with the software in this solution may only be performed by adequately qualified personnel. Qualified personnel in this context are persons who have the following qualifications:

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

All mechanical work on connected units is to be performed exclusively by adequately qualified personnel. Qualified personnel in the context of this documentation are persons familiar with the design, mechanical installation, troubleshooting and servicing of the product, who possess the following qualifications:

- Training in mechanical engineering, e.g. as a mechanic or mechatronics technician (final examinations must have been passed).
- Knowledge of this documentation and other applicable documentation.

All electrical work on connected units is to be performed exclusively by adequately qualified electricians. Qualified electricians in the context of this documentation are persons familiar with electrical installation, startup, troubleshooting and servicing of the product, who possess the following qualifications:

- Training in electrical engineering, e.g. as an electrician or mechatronics technician (final examinations must have been passed).
- Knowledge of this documentation and other applicable documentation.

24796131/EN – 12/2017

- Knowledge of the relevant safety regulations and laws.
- Knowledge of all other standards, directives and laws named in this documentation.

The above-mentioned persons must have the express authorization of the company to operate, program, configure, label and ground units, systems and circuits in accordance with the standards of safety technology.

All work in the areas of transportation, storage, operation and waste disposal must be carried out by persons who are trained appropriately.

2.3 Designated use

The HandlingKinematics application module is a multi-axis module for path control of kinematic models, as well as jogging and referencing of kinematic axes.

Use the unit-independent "Application Configurator" configuration software to start up and configure the axes and the kinematics for the application module and to download the complete configuration to the controller.

2.4 Bus systems

A bus system allows you to adapt the kinematic model, frequency inverters and/or motor starters to the particulars of the machinery.

NOTICE: Modification of the parameters, which is not visible from the outside, can result in unexpected, but controlled system behavior.

2.5 Short designations

The following short designations are used in this documentation:

Type designation	Short designation
HandlingKinematics application module	Application module, HandlingKinematics
Higher-level controller *(Programmable Logic Controller)*	PLC
Configurable Control Unit *(CCU)*	Controller

24796131/EN – 12/2017

3 Project planning notes

3.1 Requirements

Correct configuration and proper installation of the units are required for successfully starting up and operating the application modules with the Application Configurator.

You find detailed configuration information in the documentation of the respective units (see chapter "Other applicable documentation").

3.2 Hardware

Using the application module requires the following hardware:

- Controller in CCU design:
  - DHF41B (CCU advanced) or
  - DHR41B (CCU advanced)
- SD memory card (OMC41B) with sufficient technology points. The necessary number of technology points depends on the functionality required; see the "Technology points" (→ 🗎 15) chapter.

INFORMATION

If you want to use the functionality of HandlingKinematics on the **DHE41B** or **UHX71B** controller, use the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module instead of the HandlingKinematics application module because these controllers are not available in a purely parameterizable CCU version. You can find a description of the technology module's functions and startup in the corresponding manual, "HandlingKinematics for MultiMotion/MultiMotion Light technology module".

INFORMATION

The HandlingKinematics application module CANNOT be used on the **DHE21B**, **DHF21B** or **DHR21B** controllers because they do not have any technology points. However, both the HandlingKinematics application module and the "HandlingKinematics for MultiMotion/MultiMotion Light" technology module require technology points; see the "Technology points" (→ 🗎 15) chapter.

3.3 Software

Using the application module requires the following software:

- MOVITOOLS® MotionStudio 6.1 engineering software (service pack 1)
- "Application Configurator" configuration software (version V160R100 or later)
- "Kinematics technology module" Kinematics software package (version V160R100 or higher)

The complete software required is automatically included in the installation of MOVITOOLS® MotionStudio 6.1 (service pack 1) or higher.

Version	Description
V130R200	Among other things, this version only includes process data profile 1 with 32 process data words and no Touchprobe function.

24796131/EN – 12/2017

Version	Description
V140	Among other things, this version does NOT include *program 4: LINEARLY synchronized*.
V150	Among other things, this version does NOT include the HANDLING optimized blending profile.

You can download the software from the SEW-EURODRIVE homepage (www.sew-eurodrive.com).

3.4 Technology points

For unlimited functionality, the HandlingKinematics application module requires 5 technology points. If you use a CARTESIAN GANTRY kinematic model, 4 technology points are sufficient. In order to use the full version of the 3D simulation, you will need additional 10 technology points.

HandlingKinematics	3D simulation
5 technology points	10 technology points

The technology points required for your application are shown in the HandlingKinematics configuration wizard; see the "Selecting and setting the fieldbus interface" (→ 🗎 74) chapter.

INFORMATION

In order to adapt the kinematic model to the actual robot and be able to simulate and test the motion path, SEW-EURODRIVE recommends that **you always use the 3D simulation during configuration and startup**.

✓ **The 3D simulation is not possible when the simulation PC is connected via USB or fieldbus.** The 3D simulation only communicates via **Ethernet engineering interface (X37)**.

- For this purpose, establish an Ethernet connection between controller and simulation PC.

INFORMATION

Usually the full version is only required on the controller that is used for simulating and testing the motion and sequential program of the PLC in the office.

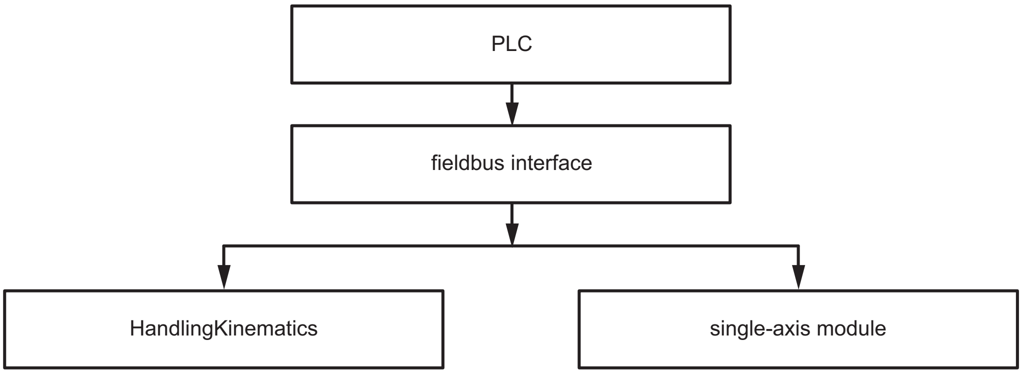
This application module consumes the necessary 4 or 5 technology points as soon as you start the controller. The 10 further technology points for the 3D simulation are only used 30 minutes after controller startup. During this time, the 3D simulation can be used without additional 10 technology points. When not enough technology points are available, the 3D simulation is deactivated after the test time of 30 minutes.

24796131/EN – 12/2017

4 System description

4.1 Areas of application

The *HandlingKinematics* application module is controlled by the higher-level controller via fieldbus.



13943506059

The application module allows for a path control for kinematics that cannot be realized in this quality with direct fieldbus control of the single axes via the PLC due to the worse and irregular response time to process information. The following tasks are taken from the PLC.

- Interpolation in space
- Transformation into motion profiles of axes
- Controlling the drives/inverters
- Sequence control of entire paths of motion

As the tasks are solved by the independent and reliable *HandlingKinematics* application module, programming errors are avoided.

During startup, the mechanical data and the motion parameters are entered once. During operation, only the positions of the path when the motion starts must be transmitted to the controller. The application module coordinates the required path movements in real time. If you define a wait point in the path, the controller continues the motion only if it is allowed. During palletizing and piling, the target position must often be adjusted in ongoing operation. The integrated Touchprobe function can do this in real time without intervention of the PLC.

4.2 Advantages

The application module offers the following advantages:

- Performance guarantee due to encapsulated continuous-path control that has been tried and tested and has proven itself time and time again.
- Quick and easy startup of the entire kinematic model with a graphical wizard that is intuitive to use and features a clear diagnostics and monitor function. No knowledge of a robot language is required because you only have to parameterize the application module.
- The clock-synchronous path control with customizable blending, LookAhead and bypassing of interfering edges without loss of contour accuracy significantly reduces the cycle time.

24796131/EN – 12/2017

4.3 Functions

The application module offers the following functions:

- The controller with the HandlingKinematics application module takes on control of the drives.
- The application module has a standardized fieldbus interface for control by the PLC, which coordinates the process sequence.
- The processes can be simulated and problems can be diagnosed early on without real machines.
- 4 different automatic programs:
  - Axis interpolation
  - Cartesian interpolation
  - Path interpolation with coordinated rotation
  - Path interpolation with synchronized rotation
- Support for mechanical components:
  - With up to 4 Cartesian degrees of freedom: Translation along the X, Y and Z-axes and rotation around the Z-axis.
  - With up to 4 drives in the standard version. When you use a kinematic model in which multiple drives are assigned to one Cartesian degree of freedom, the application module supports up to 6 drives.
- Reproducible path fidelity even after interference with BackToPath.
- Handling stationary objects.
- Can be combined with up to 8 other applications modules, e.g. for conveyor belts, lifting axes or grippers.
- Option of having wait points at each path point.
- Integration of a Touchprobe for position measurements and sensor-based positioning.

4.4 Operating principle

With the HandlingKinematics application module, a configurable kinematic model is controlled by the PLC via the fieldbus so that it automatically completes handling tasks.

Example: Material needs to be picked up at one location and deposited at another location in a defined position with a specific orientation.

The application module offers a wide range of kinematic models for this purpose that can be flexibly adapted to real robots (for example in terms of geometry).

Once you have configured the kinematic model and other settings in the application module, and started up the axes, the robot is ready for operation and can carry out movements.

Automatic operation requires control via the controller's fieldbus interface. The process data from the PLC must be assigned accordingly. The PLC gives the controller specifications as to which path should be traveled with which segment parameter sets.

The path consists of motion sequences (without stopping), which are specified by the PLC during runtime through path points and blending ranges. Various enables control when the robot performs which movements. Preconfigurable segment parameter sets define the speed and acceleration with which the robot travels these path segments.

The PLC specifies the number of the segment parameter set to be used for each path segment. For example, data sets for rapid speed, creep speed or gripping movements can be defined. For flexible handling tasks, you can also connect a Touchprobe to implement sensor-based positioning and measurement of gripping positions.

In addition to program mode, the controller with HandlingKinematics can reference the axes, jog axis-by-axis or in a Cartesian manner, and quickly decelerate with configurable ramps. Error handling with transmission of an error message to the PLC is integrated. Many useful diagnostics tools such as a 3D simulation are available for detailed error analysis and startup.

24796131/EN – 12/2017

5 Basics

5.1 Kinematic models

The kinematic models deviate from each other in the following aspects:

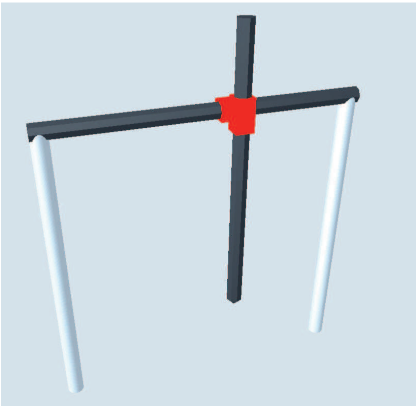
- Type and arrangement of the joints: Basic type (e.g. SCARA)
- Type and number of drives: A1 – A4, linear/rotary (e.g. LRRR)
- Cartesian degrees of freedom: X, Y, Z, rotation of A around the Z-axis (e.g. XYZA)
- Detailed differences: Arrangement of the drives (e.g. M10)

The kinematic models' identifiers are defined accordingly.

5.1.1 CARTESIAN GANTRY

CARTESIAN GANTRY is a kinematic model in which 2 or 3 linear axes are perpendicular to each other, generating a Cartesian work envelope.

CARTESIAN_GANTRY_LL_XY_M10

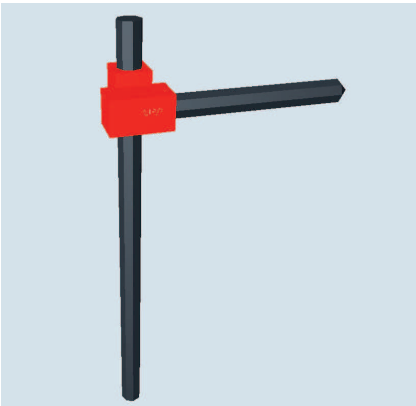


9308233355

CARTESIAN GANTRY (portal) with 2 linear axes:

- Axis 1: X direction
- Axis 2: Y direction

CARTESIAN_GANTRY_LL_ZXM10



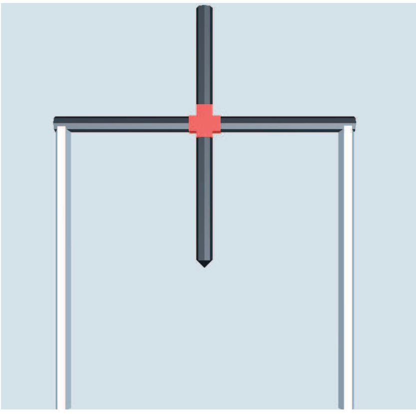
9308238987

CARTESIAN GANTRY (portal) with 2 linear axes:

- Axis 1: Z direction
- Axis 2: X direction

The angle between the Z and X directions can also be adjusted to values ≠ 90° using a parameter.

CARTESIAN_GANTRY_LL_XZ_M15

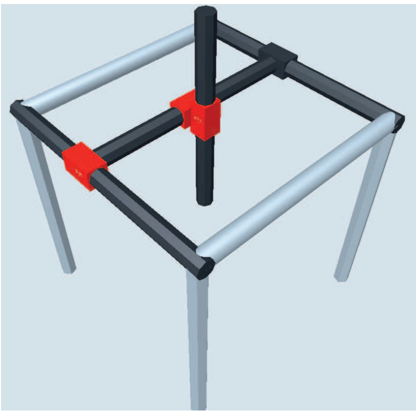


9308245003

CARTESIAN GANTRY (portal) with 2 linear axes:

- Axis 1: X direction
- Axis 2: Z direction

CARTESIAN_GANTRY_LLL_XYZ_M10

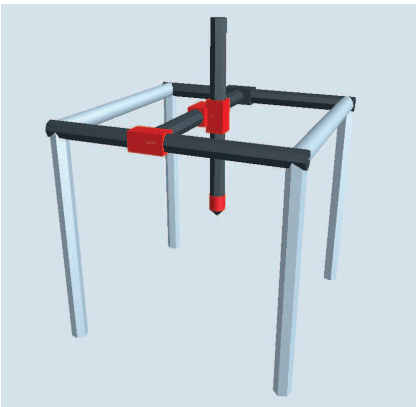


9308281483

CARTESIAN GANTRY (portal) with 3 linear axes:

- Axis 1: X direction
- Axis 2: Y direction
- Axis 3: Z direction

CARTESIAN_GANTRY_LLLR_XYZA_M10



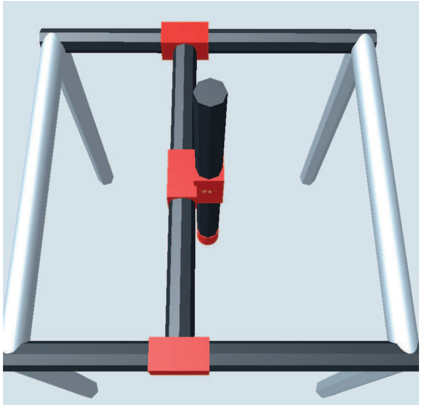
9308272907

CARTESIAN GANTRY (portal) with 3 linear axes and 1 rotary axis (rotation around Z):

- Axis 1: X direction
- Axis 2: Y direction
- Axis 3: Z direction
- Axis 4: A orientation

24796131/EN – 12/2017

CARTESIAN_GANTRY_LLLLR_XYZA_M10



9308251403

CARTESIAN GANTRY (portal) with 4 linear axes and 1 rotary axis (rotation around Z):

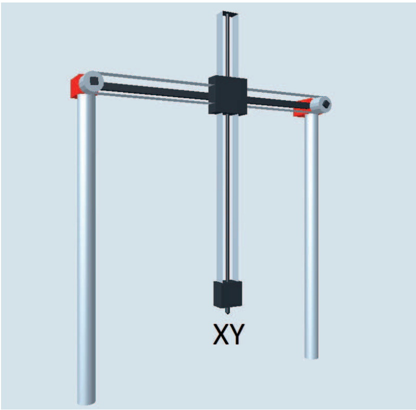
- Axes 1 and 2: X direction
- Axis 3: Y direction
- Axis 4: Z direction
- Axis 5: A orientation

INFORMATION: This is an overdetermined kinematic model; see the "Additional requirements for overdetermined kinematic models" (→ 🗎 102) chapter.

5.1.2 ROLLER GANTRY

ROLLER GANTRY is a kinematic model in which 2 translatory degrees of freedom are generally controlled by 2 stationary drives using a revolving toothed belt. Additional degrees of freedom can be added upstream or downstream of this assembly.

ROLLER_GANTRY_LL_XY_M10

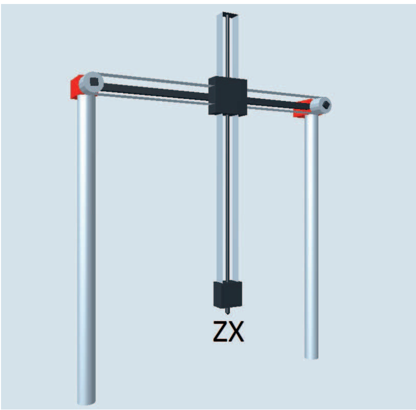


9310212747

ROLLER GANTRY with 2 axes:

- Axes 1 and 2: Allow movement on the XY plane via a toothed belt.

ROLLER_GANTRY_LL_ZX_M10



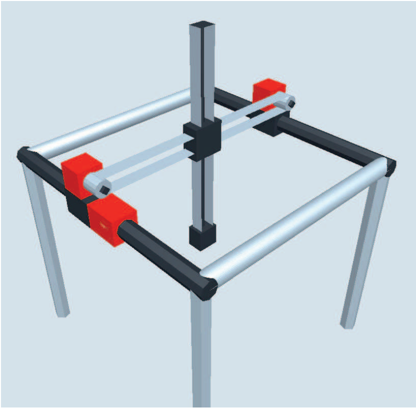
9310234123

ROLLER GANTRY with 2 axes:

- Axes 1 and 2: Allow movement on the ZX plane via a toothed belt.

24796131/EN – 12/2017

ROLLER_GANTRY_LLL_XYZ_M10

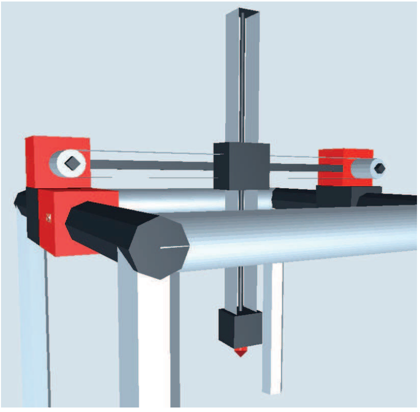


9310255883

ROLLER GANTRY with 3 axes:

- Axis 1: X direction
- Axes 2 and 3: Allow movement on the YZ plane via a toothed belt.

ROLLER_GANTRY_LLLR_XYZA_M10

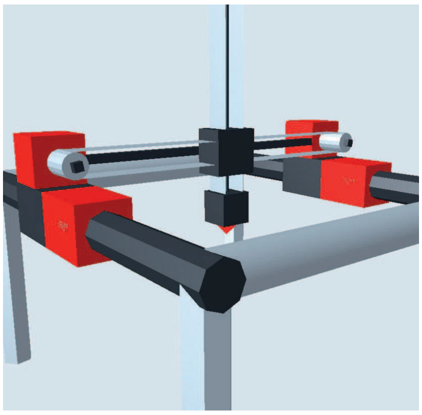


9310323467

ROLLER GANTRY with 4 axes:

- Axis 1: X direction
- Axes 2 and 3: Allow movement on the YZ plane via a toothed belt.
- Axis 4: Rotation A

ROLLER_GANTRY_LLLLR_XYZA_M10



9310278027

ROLLER GANTRY with 5 axes:

- Axes 1 and 2: X direction
- Axes 3 and 4: Allow movement on the YZ plane via a toothed belt.
- Axis 5: Rotation A

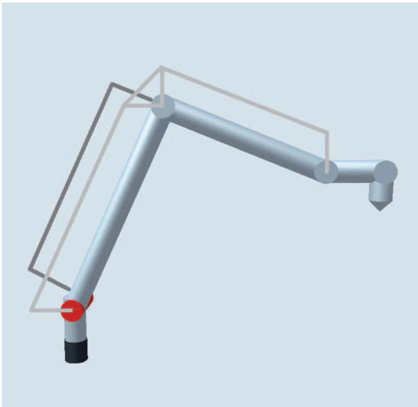
INFORMATION: This is an overdetermined kinematic model; see the "Additional requirements for overdetermined kinematic models" (→ 🗎 102) chapter.

24796131/EN – 12/2017

5.1.3 SCARA

SCARA stands for "Selective Compliance Assembly Robot Arm." SCARA is a kinematic chain in which 2 rotary axes are positioned parallel to each other. They are known as shoulder and elbow joints as in a human arm.

SCARA_RR_XY_M20



9310418955

SCARA with 2 rotary axes:

- Axis 1: Rotates the upper arm around the shoulder axis (parallel to the Z-axis).
- Axis 2: Rotates the lower arm around the elbow axis (parallel to the Z-axis) using a parallelogram or belt. The drive is stationary.

A parallelogram or belt is used to maintain a constant flange orientation, or the tool is simply something like a pen whose orientation is irrelevant.

SCARA_RRR_XYA_M20

13925772427

SCARA with 3 rotary axes:

- Axis 1: Rotates the upper arm around the shoulder axis (parallel to the Z-axis).
- Axis 2: Rotates the lower arm around the elbow axis (parallel to the Z-axis) using a parallelogram or belt. The drive is stationary.
- Axis 3: Rotates the tool around the wrist axis (parallel to the Z-axis).

SCARA_RRL_XYZ_M20

9310443787

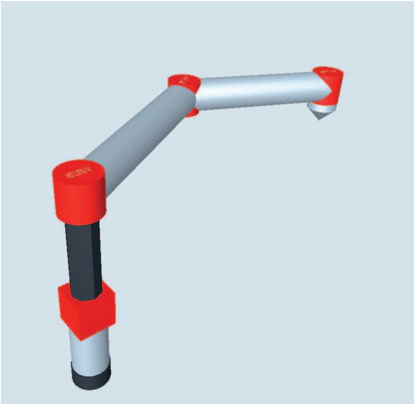
SCARA with 2 rotary axes and 1 linear axis:

- Axis 1: Rotates the upper arm around the shoulder axis (parallel to the Z-axis).
- Axis 2: Rotates the lower arm around the elbow axis (parallel to the Z-axis).
- Axis 3: Z direction

A parallelogram or belt is used to maintain a constant flange orientation, or the tool is simply something like a pen whose orientation is irrelevant.

24796131/EN – 12/2017

SCARA_LRRR_XYZA_M10

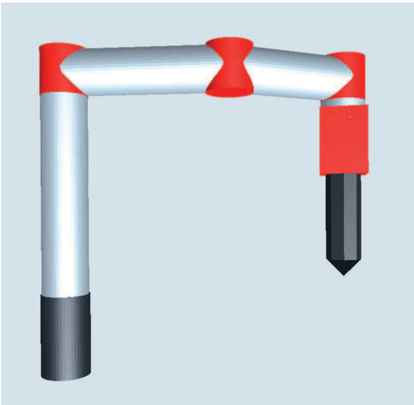


9310394507

SCARA with 1 linear axis and 3 rotary axes:

- Axis 1: Z direction
- Axis 2: Rotates the upper arm around the shoulder axis (parallel to the Z-axis).
- Axis 3: Rotates the lower arm around the elbow axis (parallel to the Z-axis).
- Axis 4: Rotates the tool around the wrist axis (parallel to the Z-axis).

SCARA_RRRL_XYZA_M10

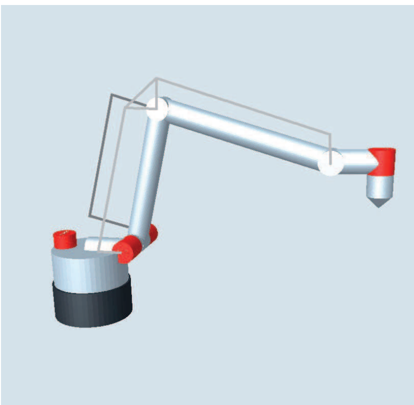


9310469003

SCARA with 3 rotary axes and 1 linear axis:

- Axis 1: Rotates the upper arm around the shoulder axis (parallel to the Z-axis).
- Axis 2: Rotates the lower arm around the elbow axis (parallel to the Z-axis).
- Axis 3: Rotates the tool around the wrist axis (parallel to the Z-axis).
- Axis 4: Z direction

SCARA_RRRR_XYZA_M60



9310494603

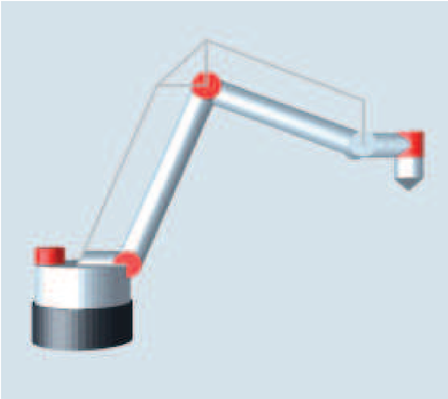
SCARA with 4 rotary axes:

- Axis 1: Rotates the turret around the turret axis (parallel to the Z-axis).
- Axis 2: Rotates the upper arm around the shoulder axis.
- Axis 3: Rotates the lower arm around the elbow axis using a parallelogram or belt. The drive is stationary on the rotary plate.
- Axis 4: Rotates the tool around the Z-axis.

A parallelogram or belt is used to maintain a constant angle for the tool.

24796131/EN – 12/2017

SCARA_RRRR_XYZA_M65



9637703179

SCARA with 4 rotary axes:

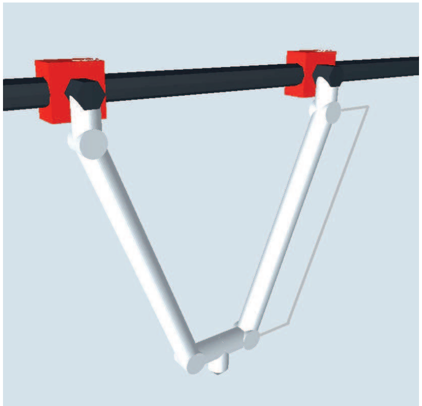
- Axis 1: Rotates the turret around the turret axis (parallel to the Z-axis).
- Axis 2: Rotates the upper arm around the shoulder axis.
- Axis 3: Rotates the lower arm around the elbow axis.
- Axis 4: Rotates the tool around the Z-axis.

A parallelogram or belt is used to maintain a constant angle for the tool.

5.1.4 DELTA

DELTA is a kinematic model in which 2 kinematic partial chains are connected in parallel in a triangular structure between the kinematic base and the tool flange.

DELTA_LL_XY_M10

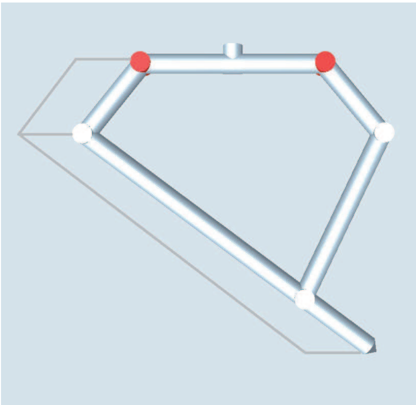


9308455051

DELTA with 2 linear axes:

- Axes 1 and 2: In combination, they allow translational movement of the tool on the XY plane. Movement of a drive cannot be clearly assigned to the movement of a Cartesian axis.

DELTA_RR_XY_M20



9308465675

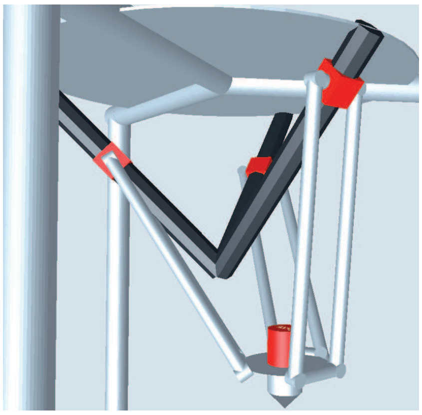
DELTA with 2 linear axes:

- Axes 1 and 2: In combination, they allow translational movement of the tool on the XY plane. Movement of a drive cannot be clearly assigned to the movement of a Cartesian axis.

5.1.5 TRIPOD

TRIPOD is a kinematic model that can be characterized as a tripod and consists of 3 parallel kinematic partial chains between the kinematic base and the tool flange.

TRIPOD_LLLR_XYZA_M10



12852397067

TRIPOD with 3 linear axes and 1 rotary axis:

- Axes 1 – 3 In combination, they allow translational movement of the tool (XYZ). Movement of a drive cannot be clearly assigned to the movement of a Cartesian axis.
- Axis 4: Orientation A

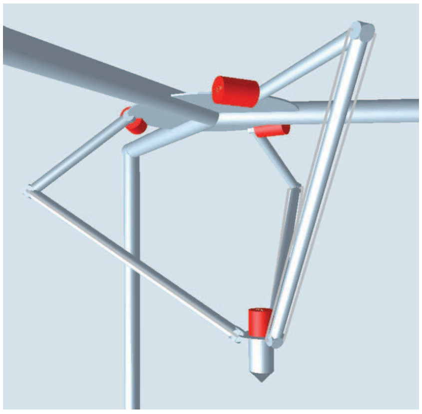
TRIPOD_RRR_XYZ_M10

9310520587

TRIPOD with 3 rotary axes:

- Axes 1 – 3 In combination, they allow translational movement of the tool (XYZ). Movement of a drive cannot be clearly assigned to the movement of a Cartesian axis.

TRIPOD_RRRR_XYZA_M10



9310546955

TRIPOD with 4 rotary axes:

- Axes 1 – 3 In combination, they allow translational movement of the tool (XYZ). Movement of a drive cannot be clearly assigned to the movement of a Cartesian axis.
- Axis 4: Orientation A

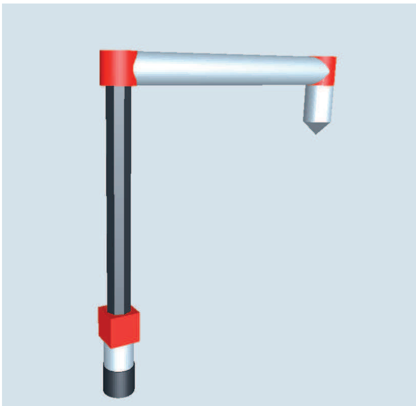
24796131/EN – 12/2017

5.1.6 MIXED

Kinematic models that do not clearly feature the characteristics of other kinematic models are called MIXED. In particular, they do not correspond to the following kinematic models:

- CARTESIAN GANTRY
- ROLLER GANTRY
- SCARA
- DELTA
- TRIPOD
- QUADROPOD
- HEXAPOD
- ARTICULATED

MIXED_LRR_ZXA_M10



9308658315

A MIXED kinematic model with 1 linear axis and 2 rotary axes that allows movement of the flange on the surface of a cylinder around the linear axis, and rotation of the flange (A). The X-axis runs horizontally on the circular arc of the cylinder around the linear axis:

- Axis 1: Z direction
- Axis 2: Rotates the arm around the shoulder axis (parallel to the Z-axis).
- Axis 3: Rotates the tool around the wrist axis (parallel to the Z-axis).

MIXED_RLLR_XYZA_M10

9308673931

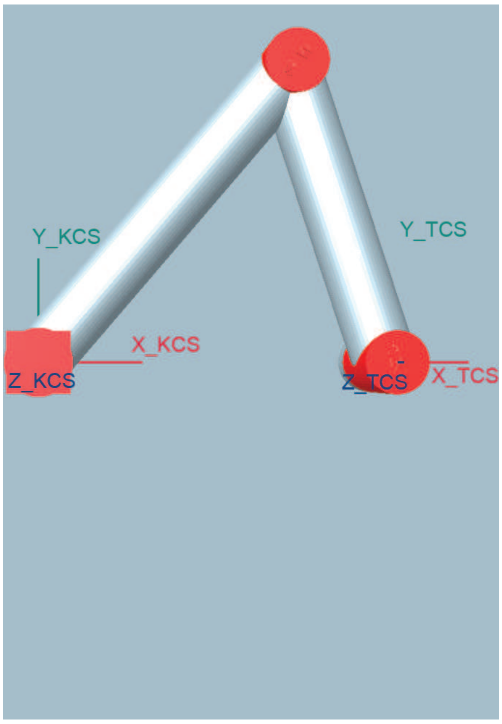
A MIXED kinematic model with 1 rotary axis in the socket, 2 linear axes and 1 rotational axis on the flange.

- Axis 1: Rotates the arm around the shoulder axis (parallel to the Z-axis).
- Axis 2: Z direction
- Axis 3: Linear axis along the X-axis in the home position.
- Axis 4: Rotates the tool around the wrist axis (parallel to the Z-axis).

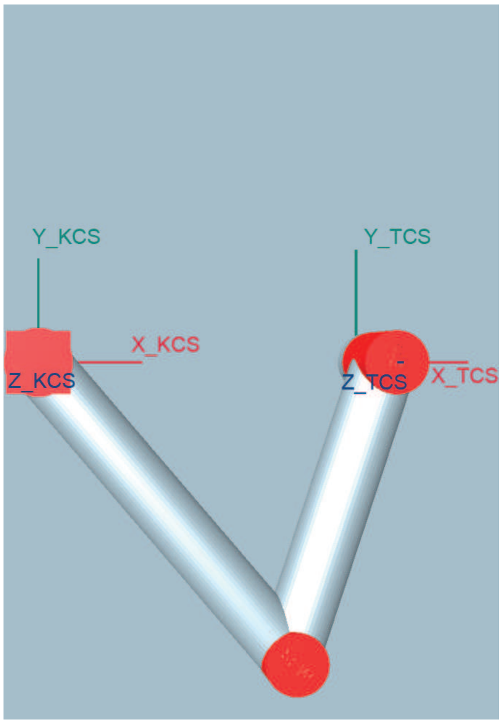
24796131/EN – 12/2017

5.1.7 Constellation

For certain kinematic models, the Cartesian pose is [1]. not enough to uniquely describe the axis position. A simple example is the SCARA kinematics, in which the same Cartesian pose can be reached with 2 different axis positions (see the figures below). The constellation describes the position of the robot in the pose. The constellation results from numbering the possible axis positions.



13935221643



13935224075

INFORMATION

The constellation is retained during travel in the following operating modes:

- In the *Program auto/Program step* program modes in programs 2, 3 and 4.
- In *Jog Cartesian* jog mode.

INFORMATION

The constellation can be changed in the following operating modes:

- In the *Program auto/Program step* program modes in program 1.
- In *Jog Axis* jog mode.

Detailed information is available in the "Programs in program mode: Program auto/ Program step" (→ 🗎 51) and "Constellations of the kinematic models" (→ 🗎 225) chapters.

1) Pose = "combination of position and orientation in space" (ISO 8373)

5.1.8 Limitation of the work envelope

Every kinematic model has a work envelope in which the kinematics can move. This envelope is the result of the kinematic model and the limitations from the software (software limitations). These parameters are set by a user with knowledge of the environment and mechanical properties of the real robot.

If a software limitation is exceeded, the system automatically triggers a rapid stop (see the "Rapid stop" (→ 🗎 33) chapter) and issues an error message. It is also possible to deliberately set the application module to temporarily ignore the software limitations. The limits are monitored in every operating mode.

INFORMATION

In programs 3 and 4, the work envelope is monitored by the "LookAhead" function; see the "LookAhead" (→ 🗎 60) chapter.

Software limitations

The following software limitations must be parameterized for the work envelope.

Axis limits

- Limit values for rotary and prismatic joints that are actively moved by a drive. They can result from factors such as the following conditions:
  - Restricted movement of the joint
  - Winding of cables
  - End positions of the linear guide

Cartesian limits

- Limit values for the Cartesian coordinates of the tool operating point and the tool orientation. The three translational limits spatially generate a cuboid in which the tool may move. The rotational limitation only restricts the permitted orientation of the tool.

Kinematic limitations

- Limits that result from the specific mechanical properties of the kinematic model and from special numerical cases (e.g. singularities). They depend on the kinematic model selected. For example, they can be used to limit the motion of a joint in parallel kinematics that is not directly moved by a drive.

5.1.9 Limiting the motor speeds

In addition to checking the work envelope, the system can also limit the motor speed. With motor speed limitations, the speeds of the drives for the planned movement must not exceed the configured maximum motor speeds. If they exceed these speeds, the system automatically triggers a rapid stop and issues an error message. It is also possible to deliberately set the application module to temporarily ignore the software limitation. The limits are monitored in every operating mode.

INFORMATION

In programs 3 and 4, the work envelope is monitored by the "LookAhead" function; see the "LookAhead" (→ 🗎 60) chapter.

24796131/EN – 12/2017

5.2 Motion control

5.2.1 Interpolation types

The kinematic control generates movement to the motor target positions for all the kinematics' operating states. The motion path depends on the interpolation type. The application module has the following basic interpolation types, which are used in the various operating modes:

Axis interpolation: *KIN_TARGET_AXIS*

- Each axis (A1 – A6) is positioned separately so that the target specified in Cartesian coordinates (X, Y, Z, A) is reached. The single axis movements can be synchronized.
- This interpolation type is used in program mode in *program 1: TARGET AXIS*.

Cartesian interpolation: *KIN_TARGET_CART*

- Each of the tool's Cartesian degrees of freedom (X, Y, Z, A) in the kinematic model is positioned separately. The individual Cartesian movements can be synchronized.
- This interpolation type is used in program mode in *program 2: TARGET CART* and in sensor-based positioning.

Path interpolation: *Continuous Path (CP)*

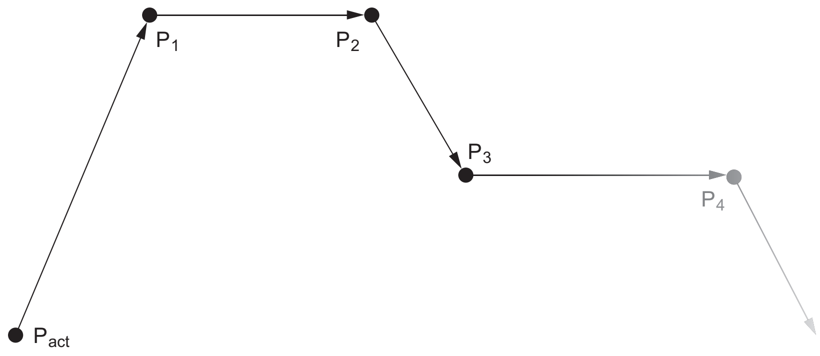
- The kinematic model's tool travels along a path defined in space consisting of straight sections and blending curves. Rounding off of the path (blending) can be defined at the transitions between 2 straight segments.
- Path interpolation can be of the following types:
  - Geometrically defined path of the translational degree of freedom. The rotational degree of freedom is coordinated with this definition. It is used in program mode in *program 3: LINEARLY coordinated*.
  - Geometrically defined path of the translational degrees of freedom, as well as the rotational degree of freedom A. It is used in program mode in *program 4: LINEARLY synchronized*.

The target is specified in Cartesian coordinates for all interpolation types.

24796131/EN – 12/2017

5.2.2 Path

The PLC defines the path of the tool for the kinematic model. The path is generated by a series of path points. A path that stops at each path point could look like the following example with 4 path points.



13929520267

$P_{current}$	Current path point
P_{1-4}	Path points 1 – 4

The movement section from one path point to the next is called the path segment.

INFORMATION

Whether the system travels exactly on the straight sections between the individual path points depends on the operating mode, the wait points and the blending.

5.2.3 Blending

Blending is rounding off the corners of a motion path. It ensures smooth transitions on the path and a constant path speed up to the next path segment. Blending protects the mechanical components and reduces the cycle time. Without blending, the kinematic model would stop at the *target pose* and then start movement to the next target point.

INFORMATION

If you want the kinematics to stop at a path point, you must activate a wait point there or end the motion sequence there; see the "Wait signals" (→ 🗎 56) and "End signals" (→ 🗎 56) chapters.

Blending starts as soon as the tool is close enough to the current *target pose*. The PLC specifies the distance from which the new path segment is blended for each path point (*blending distance*$_{PD}$).

However, the blending distance is limited to a configured percentage (*blending percentage*) of the segment length on which blending is performed. As a result, the actual distance (*blending distance*$_{effective}$) at which blending starts is based on the minimum of the two variables:

$$\textit{Blending distance}_{effective} =$$

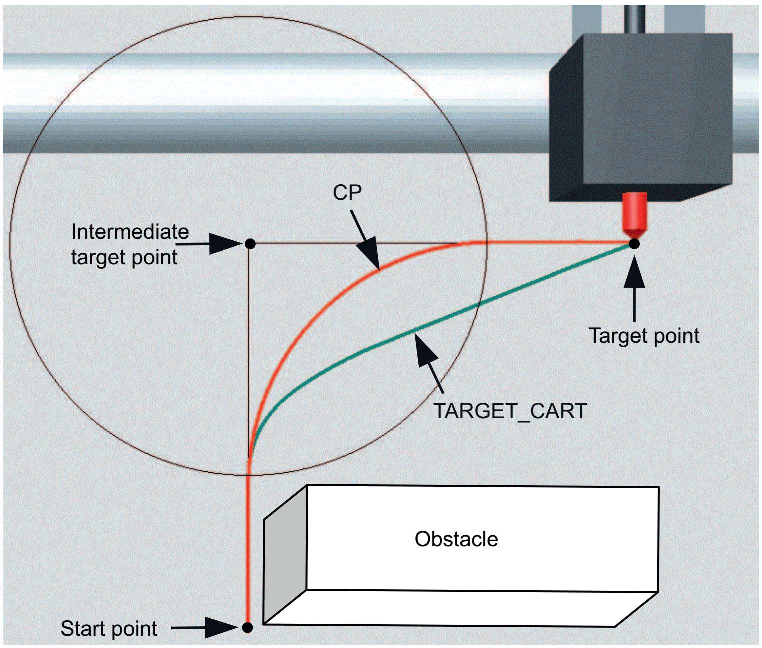
24796131/EN – 12/2017

Min. (*blending distance*$_{PD}$, *segment length* ▪ *blending percentage*)

INFORMATION

Blending is specified using the path segment or target point **to** which the blending curve leads.

The properties described above also apply to blending of the tool orientation. The rotational blending criterion is met when the angle difference to the target orientation is small enough. Blending starts as soon as the translational and rotational blending criteria are met.



13935228555

The exception is *program 4*: *LINEARLY synchronized*. In this program, only the translational blending criterion must be met. The orientation is automatically transferred to the defined blending range. Outside of the blending ranges, the kinematic model has path fidelity, meaning that the ratio between the path position and path orientation is always the same, regardless of the size of the blending ranges.

5.2.4 Movement enables

The following enable signals can be used to define the point in time at which a movement or a certain path segment will be executed.

Enable signal	Meaning
Controller inhibit	Not set: All inverter output stages are enabled.
Enable/rapid stop	Enables the inverter's interpolated position control.
Feed enable	Enables the kinematic model's motion.
Reference axis	Enables referencing of the axis.
Jog	Enables jog movement of the axis, Cartesian position or orientation of the tool.
Program start	Enables automated/incremental travel of the path.
Wait signals	Not set: Travel to the next path segment is enabled.

5.2.5 Rapid stop

Rapid stop uses the rapid stop ramps set for the current operating mode to brake the kinematic model to a standstill; see the "Rapid stop ramps" (→ 🗎 45) chapter. This function is used in the following situations:

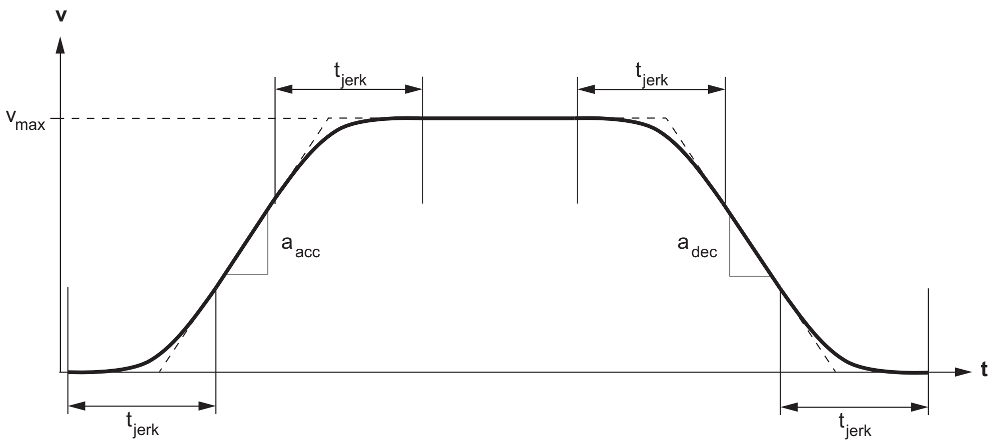
- Fault status of an application module
- Canceling the *Enable/rapid stop* enable signal: In this case, the enable for the single axes is automatically deactivated once they reach a standstill; this in turn activates a stop by the brakes or the inverter controller.

The rapid stop is not used if the inverter reports an error or the controller inhibit is set. In this case, every inverter brakes its drive separately with the rapid stop ramp configured in the inverter.

In contrast, deactivating the *feed enable* causes the kinematic model to stop with the currently set motion parameters, which depend on the operating mode.

5.2.6 Motion profile

A motion profile must be defined for movements along or around an axis. The following figure shows the variables used to define the motion profile when using this application module.



13935523339

v_{max}	Maximum permitted speed for the motion task.
a_{acc}	Maximum permitted acceleration during travel.
a_{dec}	Maximum permitted braking delay.
t_{jerk}	Jerk time (the time in which the acceleration or deceleration is performed).

v_{max} is not reached in the following cases (no constant travel):

- a_{dec} and a_{acc} are low relative to v_{max}.
- t_{jerk} is long.
- The distance to be traveled is short.

In this case, the travel diagram shown above is simplified to a jerk-limited triangular profile. Accordingly, it may also be the case that the acceleration or deceleration is not achieved (no section with constant acceleration).

INFORMATION

The jerk limitation increases the necessary time for these processes for the specified distance.

24796131/EN – 12/2017

In addition to these fundamental aspects, there are other factors and conditions that can affect the actual travel diagram. The travel diagram shown above is only intended to explain the principles of the 4 relevant motion parameters.

5.2.7 Segment parameter sets

The application module configures segment parameter sets for automatic operation. This way, the PLC programmer can simply assign the number of the appropriate segment parameter set to each path segment in the process data.

For example, data sets for rapid speed and creep speed can be configured in this process. Depending on the operating mode, certain motion parameters (speed, acceleration, deceleration) must be defined for every axis, the Cartesian degrees of freedom or the path segments. There are also other parameters such as *Constellation* and *Blending profile* that depend on the operating mode.

Detailed information is available in the "Overview of the segment parameter sets" (→ 🗎 89) chapter.

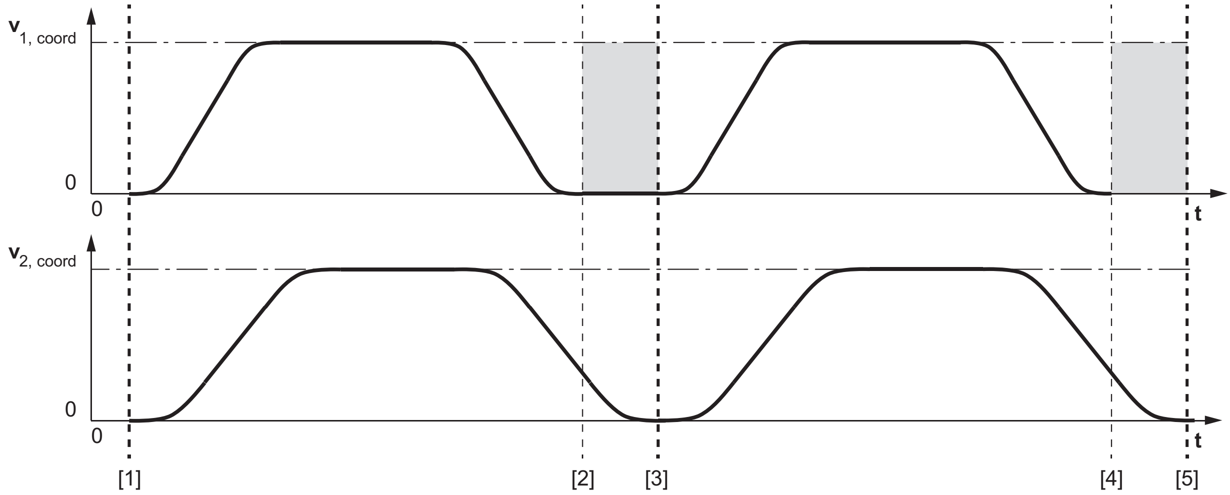
5.2.8 Scaling using override

In addition to the motion parameters in the segment parameter sets, the *Override* input value works in every operating mode except the *Homing* referencing mode. With this parameter, you can scale the speed as a percentage between 0 and 100% at any time. Override only affects speed, not acceleration, deceleration or jerk time.

5.2.9 Difference between coordination and synchronization

The following figures show the difference between coordinated and synchronized degrees of freedom/axes. The examples are shown in a simplified version without blending.

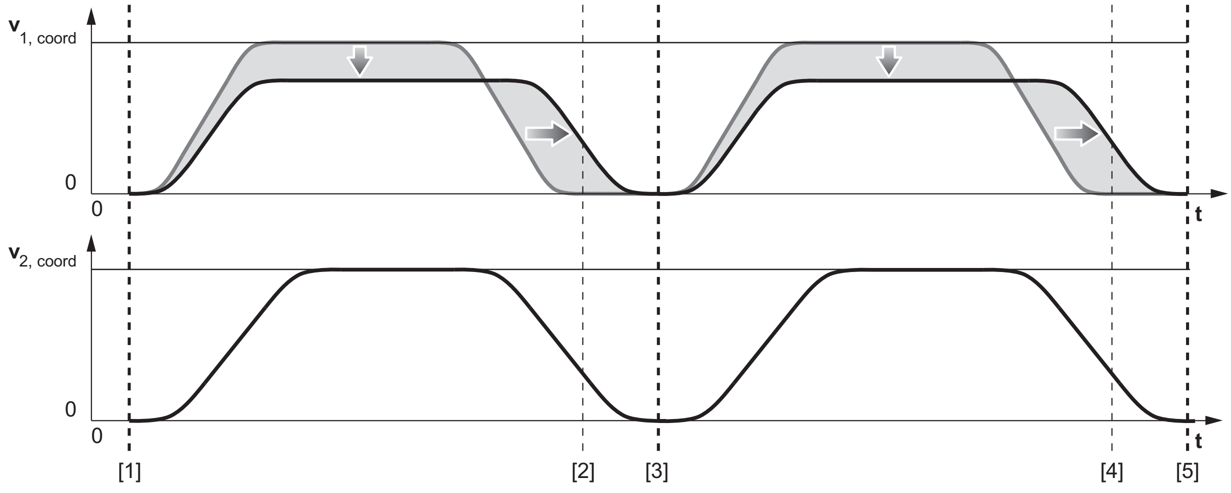
2 coordinated degrees of freedom



13936145163

$v_{1, coord}$	Maximum speed of the first degree of freedom with coordinated axes.
$v_{2, coord}$	Maximum speed of the second degree of freedom with coordinated axes.
[1]	Simultaneous start of movement for the two degrees of freedom.
[2]	End of movement for the first degree of freedom, staggered from the end of the second.
[3]	End of movement for the second degree of freedom, staggered from the end of the first. Immediately afterwards: Simultaneous start of movement for the two degrees of freedom.
[4]	End of movement for the first degree of freedom, staggered from the end of the second.
[5]	End of movement for the second degree of freedom, staggered from the end of the first.

2 synchronized degrees of freedom



13936147595

$v_{1, coord}$	Maximum speed of the first degree of freedom with coordinated axes.
$v_{2, coord}$	Maximum speed of the second degree of freedom with coordinated axes.
[1]	Simultaneous start of movement for the two degrees of freedom.
[2]	End of movement for the first degree of freedom if the movement were coordinated instead of synchronized (see the "2 coordinated degrees of freedom" diagram).
[3]	Simultaneous end of movement for the two degrees of freedom. Immediately afterwards: Simultaneous start of movement for the two degrees of freedom.
[4]	End of movement for the first degree of freedom if the movement were coordinated instead of synchronized (see the "2 coordinated degrees of freedom" diagram).
[5]	Simultaneous end of movement for the two degrees of freedom.

5.3 Communication and process data exchange

5.3.1 Communication and fieldbus

The application module's hardware topology includes at least the following system components that communicate with each other:

- PLC
- Controller with the application module
- Frequency/servo inverter

The controller with the application module communicates with the inverters via the CAN-based system bus. Every inverter on the system bus must have a different address in order to allow unique addressing of the messages and enable motion tasks for the individual drives. The controller's address is 0.

The controller communicates with the PLC via the fieldbus. Process data is exchanged for this purpose. Both systems must know how to interpret the data. For this reason, the process data profile, i.e. the length and contents of the telegrams, is precisely defined.

5.3.2 Overview of the process data profiles

The PLC must transmit the data required to travel a certain motion sequence to the controller. Because the controller's process data interface is limited to 120 process data words (PD), the possible number of path segments in a motion sequence is also limited.

4 process data profiles are defined in the application module. They differ in the number of required process data words. A larger process data profile means that more path segments can be traveled in a motion sequence without stopping.

Profile	Process data words	Path segments
1	32 PD	5 path segments
2	60 PD	10 path segments
3	88 PD	15 path segments
4	116 PD	20 path segments

These profiles build on each other; i.e. the first 32 process data words from profile 2 correspond to profile 1, and the first 60 process data words of profile 3 correspond to profile 2, etc. Refer to the "Process data assignment" (→ 🗎 135) chapter for detailed information on assigning process data.

The target positions of the path segments to be traveled are specified with the accuracy of the WORD data type.

5.3.3 Writing conventions and special features

The poses[1)] in the input and output areas of the process data interface are illustrated as a vector with the following 4 entries:

1. X position: coordinate on the X-axis
2. Y position: coordinate on the Y-axis
3. Z position: coordinate on the Z-axis
4. A orientation: rotation around the Z-axis

24796131/EN – 12/2017

INFORMATION

The process data numbering in this manual should be considered relative. It refers to the beginning of the process data used for the application module.

This is because other process data words can be created before the process data of the application module, and the number of these other process data words varies on a case-by-case basis. For this reason, all the numbers for process data in this manual feature a superscript "r" for "relative," e.g. $I8^r$.

2 process data words for the controller are always created before the application module's process data. As a result, the numbering of the process data of the application module is always increased by at least 2. For example, in this manual, the process data word of *X target position 1* is designated as process input word $I8^r$, but is I10 or higher in the controller.

INFORMATION

Pose and speed values can only be transmitted as whole numbers via the fieldbus. For greater accuracy, you can set decimal positions in the application module; see the "Selecting and setting the fieldbus interface" (→ 🗎 74) chapter.

If the decimal positions set in the application module are set greater than 0, the input values in the controller are divided by a factor of 10 to the decimal positions before they are processed as floating-point numbers. On the other hand, the output values in the controller are multiplied by this factor before they are transferred to the PLC.

Examples

- If the decimal positions are set to 2, and a controller input word contains the value 1, the controller is operated with the value 0.01.
- If the decimal positions are set to 2, and a controller output word contains the value 1, this value must be interpreted as 0.01 in the PLC.

NOTICE

The following identification applies to all process data in this manual:

- The values identified with * are interpreted with the decimal positions set for poses in the controller and transmitted.
- The values identified with ** are interpreted with the decimal positions set for speed in the controller and transmitted.

1) Pose = "Combination of position and orientation in space" (ISO 8373)

5.3.4 Overview: profile 1 with 32 PD

This profile allows travel in a path starting at the current pose to a maximum of 5 target poses with parameterizable blending. For target poses 1 to 4, the following options can be set:

- Wait point using the wait signal; see the "Wait signals" (→ 🗎 56) chapter.
- End of the motion sequence using the end signal; see the "End signals" (→ 🗎 56) chapter.

The 32 process data words are used as follows:

- 6 process data words – for general control signals.
- 26 process data words – for the max. 5 path segments.

Process input data

The following table shows an overview of the process input data for profile 1 with 32 process data words.

PD grouping	PD	Process data: High byte	Process data: Low byte
Control signals	I1[r]	*Control word (application module)*	
	I2[r]	*Referencing*	*Operating mode*
	I3[r]	*Program no.*	*Telegram no.*
	I4[r]	*Control program*	
	I5[r]	*Jog*	
	I6[r]	*Touchprobe remaining travel [%]*	*Override*
	I7[r]	*Assignment of the segment parameter sets to the 5 segments*	
Segment 1	I8[r] – I11[r]	*Target pose1 (XYZA)**	
	I12[r]	*Reserved*	
Segment 2	I13[r] – I16[r]	*Target pose 2 (XYZA)* *	
	I17[r]	*Blending distance to segment 2*	
Segment 3	I18[r] – I21[r]	*Target pose 3 (XYZA)* *	
	I22[r]	*Blending distance to segment 3*	
Segment 4	I23[r] – I26[r]	*Target pose 4 (XYZA)* *	
	I27[r]	*Blending distance to segment 4*	
Segment 5	I28[r] – I31[r]	*Target pose 5 (XYZA)* *	
	I32[r]	*Blending distance to segment 5*	

[r] The process data numbering is relative. It refers to process data used for the application module; see the note in the "Overview of process data profiles" (→ 36) chapter.

* In the controller, the values are interpreted and transmitted with the decimal positions set for positions.

Process output data

The following table shows an overview of the process output data for profile 1 with 32 process data words.

PD grouping	PD	Process data: High byte	Process data: Low byte
Status signals	O1[r]	*Status word (application module)*	
	O2[r]	*Referencing*	*Operating mode*
	O3[r]	*Program no.*	*Telegram no.*
	O4[r]	*Program status*	
	O5[r] – O6[r]	*Error ID*	
Cartesian position KCS	O7[r] – O10[r]	*Current/measured pose (XYZA)* (can be switched with I1.8) *	
	O11[r]	*Constellation*	

24796131/EN – 12/2017

PD grouping	PD	Process data: High byte	Process data: Low byte
ACS axis values	O12r – O15r	*Axis values 1 – 4*	
Max./current motor speed	O16r – O19r	*Maximum/current motor speed 1 – 4* (can be switched with I1.7) **	
Status of axes 1 – 6	O20r – O25r	*Status of axes 1 – 6*	
Kinematic model	O26r	*Kinematic model status bits*	
	O27r	*Kinematic model operating mode*	
	O28r	*Translational speed (TCP)* **	
	O29r – O30r	Reserved	
	O31r	*CP path: Target distance**	
	O32r	*Current segment ID*	*Remaining CP segments*

r The process data numbering is relative. It refers to process data used for the application module; see the note in the "Overview of process data profiles" (→ 🗎 36) chapter.

* In the controller, the values are interpreted and transmitted with the decimal positions set for positions.

** In the controller, the values are interpreted and transmitted with the decimal positions set for speeds.

You can find a detailed list and description of the process data in the "Process data assignment" (→ 🗎 135) chapter.

5.3.5 Overview: Profile 2 with 60 PD

This profile allows travel in a path starting at the current pose to a maximum of 10 target poses with parameterizable blending. For target poses 1 to 9, the following options can be set using the fieldbus:

- Wait point using the wait signal; see the "Wait signals" (→ 🗎 56) chapter.
- End of the motion sequence using the end signal; see the "End signals" (→ 🗎 56) chapter.

The 60 process data words are used as follows:

- 6 process data words – for general control signals.
- 54 process data words – for the max. 10 path segments.

The first 32 process data words correspond to profile 1. Profile 2 includes an additional 28 process data words to allow you to specify 5 more path segments via the fieldbus interface.

Process input data

The following table shows an overview of the process input data for profile 2 with 60 process data words.

PD grouping	PD	Process data: High byte	Process data: Low byte
Profile 1	I1r – I32r	See profile 1	

24796131/EN – 12/2017

PD grouping	PD	Process data: High byte	Process data: Low byte
Control signals	I33[r]	Reserved	*Telegram no.*
	I34[r]	*Control program*	
	I35[r]	*Assignment of the segment parameter sets to the 5 segments*	
Segments 6 – 10	I36[r] – I60[r]	*Target poses 6 – 10 (XYZA) and the blending distance to each of these segments**	

[r] The process data numbering is relative. It refers to process data used for the application module; see the note in the "Overview of process data profiles" (→ 🗎 36) chapter.

***** In the controller, the values are interpreted and transmitted with the decimal positions set for positions.

Process output data

The following table shows an overview of the process output data for profile 2 with 60 process data words.

PD grouping	PD	Process data: High byte	Process data: Low byte
Profile 1	O1[r] – O32[r]	See profile 1	
Status signals	O33[r]	Reserved	
	O34[r]	*Program status*	
	O35[r]	Reserved	*Telegram no.*
Reserved	O36[r] – O60[r]	Reserved	

[r] The process data numbering is relative. It refers to process data used for the application module; see the note in the "Overview of process data profiles" (→ 🗎 36) chapter.

You can find a detailed list and description of the process data in the "Process data assignment" (→ 🗎 135) chapter.

5.3.6 Overview: Profile 3 with 88 PD

This profile allows travel in a path starting at the current pose to a maximum of 15 target poses with parameterizable blending. For target poses 1 to 14, the following options can be set using the fieldbus:

- Wait point using the wait signal; see the "Wait signals" (→ 🗎 56) chapter.
- End of the motion sequence using the end signal; see the "End signals" (→ 🗎 56) chapter.

The 88 process data words are used as follows:

- 6 process data words – for general control signals.
- 82 process data words – for the max. 15 path segments.

The first 60 process data words correspond to profile 2. Profile 3 includes an additional 28 process data words to allow you to specify 5 more path segments via the fieldbus interface.

24796131/EN – 12/2017

Process input data

The following table shows an overview of the process input data for profile 3 with 88 process data words.

PD grouping	PD	Process data: High byte	Process data: Low byte
Profile 2	I1[r] – I60[r]	See profile 2	
Control signals	I61[r]	Reserved	*Telegram no.*
	I62[r]	*Control program*	
	I63[r]	*Assignment of the segment parameter sets to the 5 segments*	
Segment 11 – 15	I64[r] – I88[r]	*Target poses 11– 15 (XYZA) and the blending distance to each of these segments**	

[r] The process data numbering is relative. It refers to process data used for the application module; see the note in the "Overview of process data profiles" (→ 🗎 36) chapter.

***** In the controller, the values are interpreted and transmitted with the decimal positions set for positions.

Process output data

The following table shows an overview of the process output data for profile 3 with 88 process data words.

PD grouping	PD	Process data: High byte	Process data: Low byte
Profile 2	O1[r] – O60[r]	See profile 2	
Status signals	O61[r]	Reserved	
	O62[r]	*Program status*	
	O63[r]	Reserved	*Telegram no.*
Reserved	O64[r] – O88[r]	Reserved	

[r] The process data numbering is relative. It refers to process data used for the application module; see the note in the "Overview of process data profiles" (→ 🗎 36) chapter.

You can find a detailed list and description of the process data in the "Process data assignment" (→ 🗎 135) chapter.

5.3.7 Overview: Profile 4 with 116 PD

This profile allows travel in a path starting at the current pose to a maximum of 20 target poses with parameterizable blending. For target poses 1 to 19, the following options can be set using the fieldbus:

- Wait point using the wait signal; see the "Wait signals" (→ 🗎 56) chapter.
- End of the motion sequence using the end signal; see the "End signals" (→ 🗎 56) chapter.

The 116 process data words are used as follows:

- 6 process data words – for general control signals.
- 110 process data words – for the max. 20 path segments.

24796131/EN – 12/2017

The first 88 process data words correspond to profile 3. Profile 4 includes an additional 28 process data words to allow you to specify 5 more path segments via the fieldbus interface.

Process input data

The following table shows an overview of the process input data for profile 4 with 116 process data words.

PD grouping	PD	Process data: High byte	Process data: Low byte
Profile 3	I1[r] – I88[r]	See profile 3	
Control signals	I89[r]	Reserved	*Telegram no.*
	I90[r]	*Control program*	
	I91[r]	*Assignment of the segment parameter sets to the 5 segments*	
Segment 16 – 20	I92[r] – I116[r]	*Target poses 16 – 20 (XYZA) and the blending distance to each of these segments**	

[r] The process data numbering is relative. It refers to process data used for the application module; see the note in the "Overview of process data profiles" (→ 36) chapter.

***** In the controller, the values are interpreted and transmitted with the decimal positions set for positions.

NOTICE

Segment 20 cannot be transmitted consistently; see the "Consistency blocks for profile 4 with 116 PD" (→ 105) chapter.

Process output data

The following table shows an overview of the process output data for profile 4 with 116 process data words.

PD grouping	PD	Process data: High byte	Process data: Low byte
Profile 3	O1[r] – O88[r]	See profile 3	
Status signals	O89[r]	Reserved	
	O90[r]	*Program status*	
	O91[r]	Reserved	*Telegram no.*
Reserved	O92[r] – O116[r]	Reserved	

[r] The process data numbering is relative. It refers to process data used for the application module; see the note in the "Overview of process data profiles" (→ 36) chapter.

You can find a detailed list and description of the process data in the "Process data assignment" (→ 135) chapter.

24796131/EN – 12/2017

6 Operating modes and functions

6.1 Overview

The following table provides an overview of the operating modes and their use. The operating mode is set as follows in input word I2^{r} in the low byte.

Operating mode	I2^{r}:Low	Description	Purpose
Operating mode inactive *Default*	Value 0	No operating mode is selected.	(Start status)
Axis-by-axis jog mode *Jog Axis*	**Bit 0** is TRUE; all other bits of the byte are FALSE.	Jogging the kinematics in the axis coordinates.	Accessibility check, movement in case of error, manual operation.
Cartesian jog mode *Jog Cartesian*	**Bit 1** is TRUE; all other bits of the byte are FALSE.	Jogging the kinematic model in the translation and orientation coordinates.	
Referencing mode *Homing*	**Bit 2** is TRUE; all other bits of the byte are FALSE.	Referencing the axes.	Defining the axis zero.
Auto program program mode	**Bit 3** is TRUE; all other bits of the byte are FALSE.	Executing programs in automatic mode (without confirming every program step).	Automatic movement of the kinematics (target state).
Program step program mode	**Bit 4** is TRUE; all other bits of the byte are FALSE.	Executing programs in step mode (with confirmation of every program step).	Diagnostics for automatic operation.

When you have successfully switched to the relevant operating mode, the mode is also displayed in the output area.

6.1.1 Requirements for cycle diagrams

The following chapters offer a typical cycle diagram and workflow description for each operating mode to help you better understand the operating principle. The requirements for the cycle diagram correspond to the general requirements for the movement; see the "General requirements for movement" (→ 🗎 101) chapter.

You can find a detailed list and description of the process data in the "Process data assignment" (→ 🗎 135) chapter.

6.2 Operating mode inactive: Default

If the *Default* operating mode is specified at the operating mode input, the application module is switched to the *Default* operating mode. The kinematics is decelerated up to standstill. The O1^{r}:3 *Application ready* signal is always FALSE in this operating mode.

NOTICE

If you switch to the *Default* operating mode during program processing, the program is no longer processed and the internal feed enable for the kinematics is set to *FALSE*. The program is reset once you return to program mode.

24796131/EN – 12/2017

6.3 Jog mode: Jog Axis / Jog Cartesian

These two operating modes allow you to implement simple manual operation. Jog mode is used for purposes such as checking whether the kinematic model can reach a certain pose.

Activating the corresponding signals in the I5r *Jog (control word)* process data word causes the kinematic model to move as follows:

- Along every kinematic axis (axis-by-axis jog mode: *Jog Axis*)
- Along the spatial axes (Cartesian jog mode: *Jog Cartesian*)

Jog mode	Description
Jog Axis	Jogging the axes in the axis coordinates.
Jog Cartesian	Jogging the kinematic model in the Cartesian translation and orientation coordinates.

The motion profiles for jog mode are defined during configuration of the axes and Cartesian motion parameters; see the "Jog Axis" (→ 🗎 82) and "Configuring the motion parameters" (→ 🗎 88) chapters.

NOTICE

When switching to jog mode during a movement (e.g. from a movement triggered by a program), the kinematic model is decelerated along the ramps defined for jogging.

When you return to program mode, any previously executed program is reset.

NOTICE

Jog mode is level controlled. A fault in jog mode results in deceleration of the kinematics.

If the fault is acknowledged and rectified, and the corresponding input bit is still applied, the kinematic model continues in the specified jog direction.

6.3.1 Rapid stop ramps in jog mode

The following table shows the rapid stop ramps in jog mode.

Rapid stop	Jog Axis	Jog Cartesian
Axes 1 – 4	Rapid stop ramps of the axes; see the "Axis-by-axis jogging" chapter.	-
XYZ	-	Rapid stop ramps of the Cartesian degrees of freedom XYZ; see the "Configuring the motion parameters" (→ 🗎 88) chapter.
A	-	Rapid stop ramps of the Cartesian degree of freedom A; see the "Configuring the motion parameters" (→ 🗎 88) chapter.

You can find more detailed information in the "Rapid stop" (→ 🗎 33) chapter.

6.4 Referencing mode: Homing

The objective of referencing is to define the axis zero.

The I2^r:8 – I2^r:11 *Start referencing axis 1 – 4* signals trigger referencing of the drives. The corresponding O2^r:8 – O2^r:11 *Axis 1 – 4 referenced* signal is set to *FALSE* for at least 2 seconds after the rising edge to the I2^r:8 – I2^r:11 *Start referencing axis 1 – 4* signal. When the inverter is referenced, the corresponding O2^r:8 – O2^r:11 *Axis 1 – 4 referenced* signal is set to *TRUE*. In this way, an edge change can be detected that indicates whether referencing has been carried out successfully.

The motion profile for referencing is defined by the reference travel parameters in the kinematic configuration; see the "Reference travel parameters" (→ 🗎 79) chapter.

NOTICE

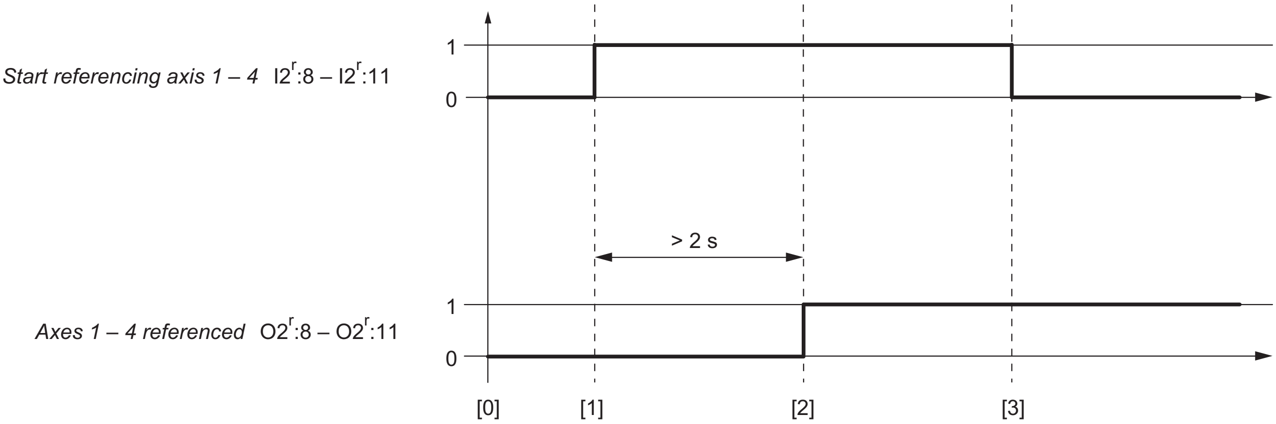
When switching to referencing mode during a movement (e.g. a movement triggered by a program), the kinematic model is decelerated with the rapid stop ramps defined in the configuration.

NOTICE

If one of the I2^r:8 – I2^r:11 *Start referencing axis 1 – 4* signals is set to *TRUE* during a switch to *Homing*, referencing is initiated for the corresponding axis. In this case, no rising edge of the *Start referencing* signal is required.

24796131/EN – 12/2017

6.4.1 Cycle diagram



9007213255418763

Process

No.	Description
[0]	Axes 1 to 4 are not referenced.
[1]	Rising edge of the I2^r:8 – I2^r:11 *Start referencing axis 1 – 4* start signal.
[2]	The O2^r:8 – O2^r:11 *Axis 1 – 4 referenced* signal is set to TRUE for at least 2 seconds after the axis is referenced successfully.
[3]	Resetting the I2^r:8 – I2^r:11 *Start referencing axis 1 – 4* start signal.

6.5 Program mode: Program auto/Program step

The *Program auto/Program step* program mode is used to travel to path points using predefined programs. The following edge-controlled signals are used for program control:

- I4^r:0 *Program init*
- I4^r:1 *Program start*
- I4^r:2 *Program stop*

The following operating modes are available in program mode.

Program mode	Description
Program Auto	If the I4^r:1 *Program start* signal has a rising edge, the program is executed until the end.
Program Step	If the I4^r:1 *Program start* signal has a rising edge, a program statement is executed.

The motion parameters for automatic mode are determined by specifying the number of the appropriate segment parameter set via the fieldbus (using the I7^{r}ff. *Assignment of the segment parameter sets to the segments* ... signals). The segment parameter sets are parameterized in the kinematic configuration; see the "Overview of the segment parameter sets" (→ 🗎 89) chapter.

NOTICE

In program mode, the I1ʳ:2 *Feed enable* signal is only transmitted to the kinematic model if the program is running. The kinematic model will be decelerated to standstill if the program is not running.

INFORMATION

The positions, blending distances and segment parameter sets are only applied when the program is started for the first time. If these values are changed after the first program start, it has no effect on program processing.

INFORMATION

Faults that are triggered by program processing can often only be acknowledged if the program was initialized first. This is because the data received via the fieldbus, such as the numbers of the segment parameter sets to use and the positions, are only applied when the program starts and are then permanently defined. The I4ʳ:0 *Program init* signal can be used to restart program initialization.

6.5.1 Program mode: Auto program

The *Program auto* program mode is used to continuously execute a program after it is started. The following edge-controlled signals are used for program control:

Signal	Function
I4ʳ:0 *Program init*	Resets the program.
I4ʳ:1 *Program start*	Starts program processing.
I4ʳ:2 *Program stop*	Stops the program (not the same as cancellation).

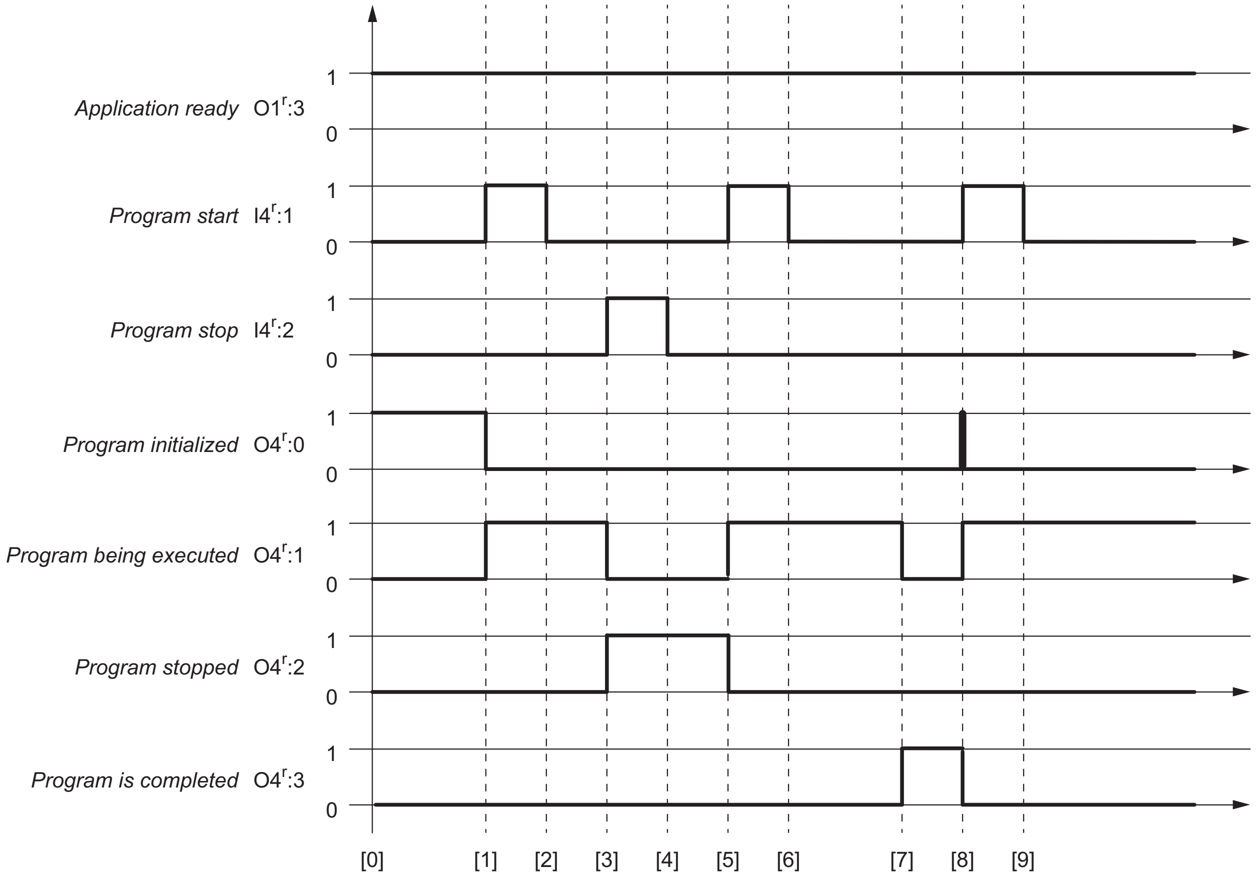
In the *Program auto* program mode, if there is a rising edge for the I4ʳ:1 *Program start* signal after the program ends, the program is reset and executed again (unlike in the *Program step* program mode).

INFORMATION

If the kinematic model does not move in the *Program auto* program mode despite a rising edge for the I4ʳ:1 *Program start* signal, the cause may be the following:

- I1ʳ:2 *Feed enable = FALSE.*
- I6ʳ:Low *Override* = 0.
- The target poses of all path segments are identical with the current pose; see the "Requirements for movement of the kinematic model" (→ 🗎 101) chapter.

Cycle diagram for temporary interruption of program execution and starting after end of program



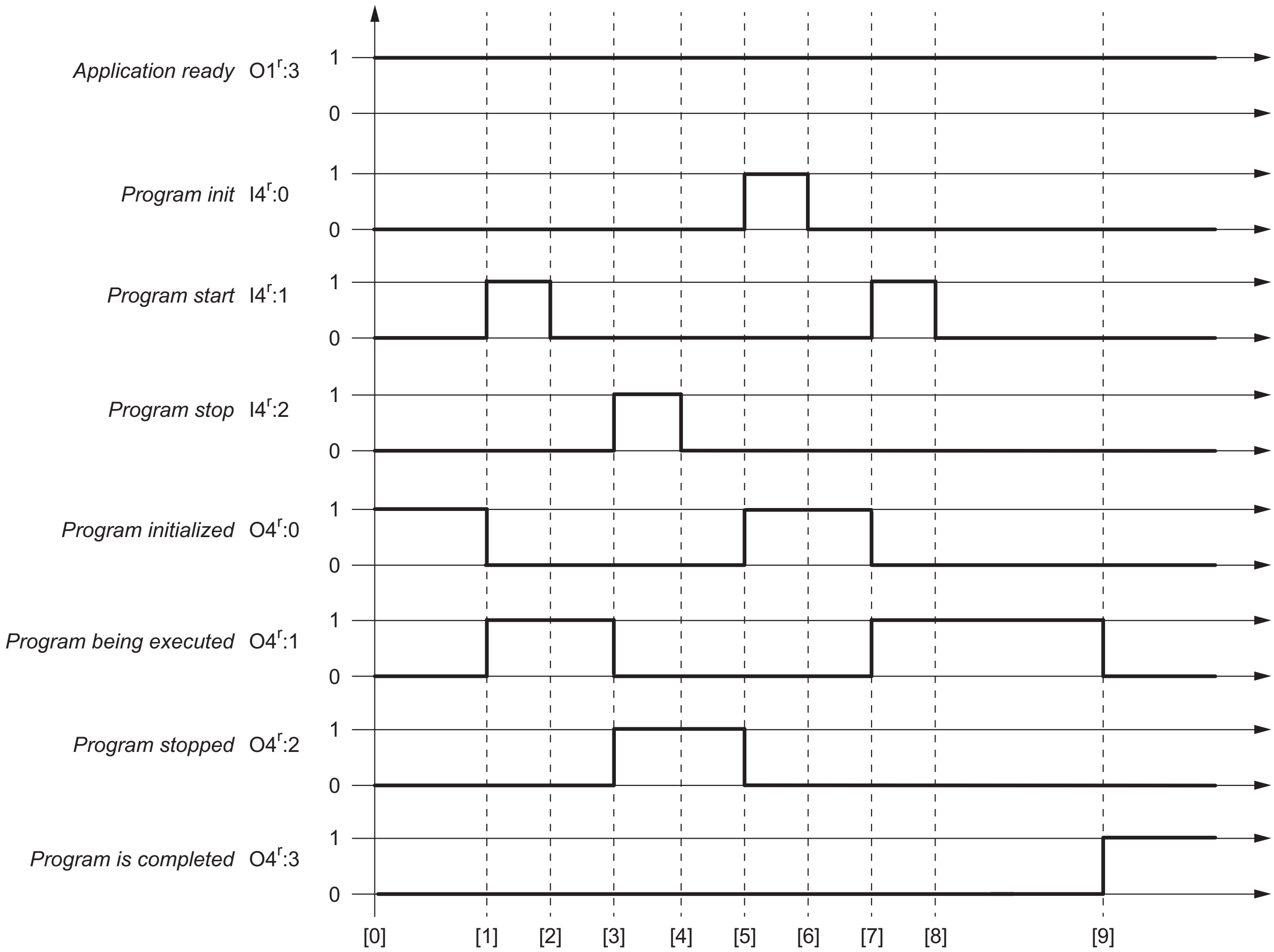
9007213255421835

Process

No.	Description
[0]	The program is initialized (O4^{r}:0 *Program initialized* = *TRUE*).
[1]	Activating the I4^{r}:1 *Program start* signal starts execution of the program.
[2]	Resetting the I4^{r}:1 *Program start* signal to *FALSE* (not until after fieldbus has returned the O4^{r}:1 *Program being executed* = *TRUE* signal).
[3]	Activating the I4^{r}:2 *Program stop* signal stops execution of the program. It is stopped with the motion parameter set currently in use.
[4]	Resetting the I4^{r}:2 *Program stop* signal to *FALSE* (not until after the O4^{r}:2 *Program stopped* signal = *TRUE* or O4^{r}:3 *Program is completed* = *TRUE* has been reported by the fieldbus).
[5]	Activating the I4^{r}:1 *Program start* signal continues execution of the program. The system travels the path segment where it stopped in step 2.
[6]	Resetting the I4^{r}:1 *Program start* signal to *FALSE* (not until after fieldbus has returned the O4^{r}:1 *Program being executed* = *TRUE* signal).
[7]	The program has been executed completely (O4^{r}:3 *Program ended* = *TRUE*).
[8]	After the end of the program, the program can be restarted from the beginning by activating the I4^{r}:1 *Program start* signal. It is not necessary to initialize it again.

No.	Description
[9]	Resetting the I4^r:1 *Program start* signal to *FALSE* (not until after fieldbus has returned the O4^r:1 *Program being executed* = *TRUE* signal).

Cycle diagram for canceling execution of the program and starting from the beginning of the program



9007213255424267

Process

No.	Description
[0]	The program is initialized (O4^r:0 *Program initialized* = *TRUE*).
[1]	Activating the I4^r:1 *Program start* signal starts execution of the program.
[2]	Resetting the I4^r:1 *Program start* signal to *FALSE* (not until after fieldbus has returned the O4^r:1 *Program being executed* = *TRUE* signal).
[3]	Activating the I4^r:2 *Program stop* signal stops execution of the program. It is stopped with the motion parameter set currently in use.
[4]	Resetting the I4^r:2 *Program stop* signal to *FALSE* (not until after the O4^r:2 *Program stopped* signal = *TRUE* or O4^r:3 *Program is completed* = *TRUE* has been reported by the fieldbus).
[5]	Activating the I4^r:0 *Program init* signal initializes the program again. As a result, the next program execution will start at the first path segment again.

24796131/EN – 12/2017

No.	Description
[6]	Resetting the $I4^{r}$:0 *Program init* signal to *FALSE* (not until after fieldbus has returned the $O4^{r}$:0 *The program initialized = TRUE* signal).
[7]	Activating the $I4^{r}$:1 *Program start* signal starts execution of the program. In this case, the program starts with the first path segment (can be identified by the $O4^{r}$:0 *Program initialized* signal = TRUE beforehand).
[8]	Resetting the $I4^{r}$:1 *Program start* signal to *FALSE* (not until after fieldbus has returned the $O4^{r}$:1 *Program being executed = TRUE* signal).
[9]	The program has been executed completely ($O4^{r}$:3 *Program is completed = TRUE*).

6.5.2 Program mode: Program step

In the *Program step* program mode, the instructions of the SRL[1] program in the SRL interpreter are performed step by step, but the next instruction is never performed until there is a rising edge for the $I4^{r}$:1 *Program start* signal (triggered step by step). The SRL program with the program and motion pointer is displayed on the diagnostic monitor; see the "Tab" (→ 🗎 123) chapter:" Program" (→ 🗎 123).

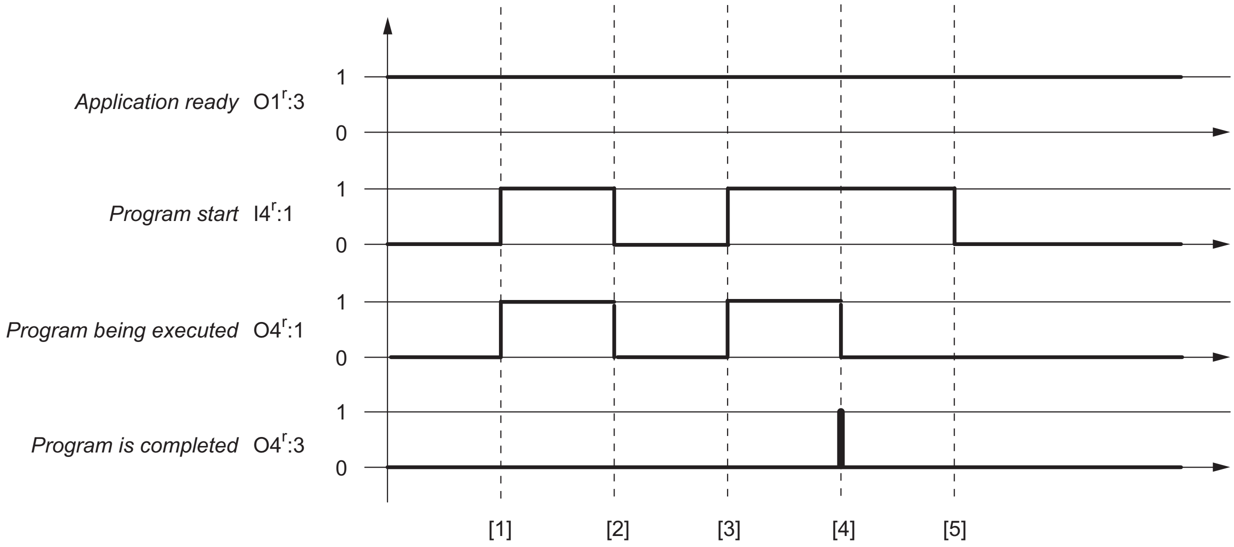
NOTICE

In the first step of diagnostics, the path must be traveled and analyzed by incrementally resetting the wait signals.

Only in cases where this does not provide the necessary information is the *Program step* program mode suitable for more detailed diagnostics. This mode does not perform the motion commands as triggered step by step, but rather the instructions in the SRL[1] program. For this reason, it only makes sense to use this operating mode if you are using the diagnostic monitor for debugging.

1) SRL = SEW ROBOT Language

Cycle diagram



9007213255426699

Process

No.	Description
[1]	Activating the I4ʳ:1 *Program start* signal starts a program command.
[2]	If the signal is deactivated again before completion of the program command, execution of the program command stops.
[3]	If it is reactivated, the program executes the program command again.
[4]	When the program command is ended, the O4ʳ:1 *Program being executed = FALSE* signal is issued.
[5]	The next program command can only be executed once the I4ʳ:1 *Program start* signal has been deactivated; continue with [1].

6.6 Programs in program mode

INFORMATION

For handling applications, we recommend the following:

- *Program 4: LINEARLY synchronized*
- HANDLING blending profile

The following table can help you decide which program to select for your application. It is based on the interpolation types described in the "Interpolation types" (→ 30) chapter.

	Program 1: TARGET AXIS	*Program 2: TARGET CART*	*Program 3: LINEARLYcoordinated*	*Program 4: LINEARLY synchronized*
Interpolation type	Axis interpolation	Cartesian interpolation	Translation: Path interpolation (CP) Rotation: Cartesian interpolation	Translation: Path interpolation (CP) Rotation: Path interpolation (CP)
Path fidelity	-	-	Translation only	✓
Synchronous rotation and translation	Depending on motion parameters	Depending on motion parameters	-	✓
BackToPath after interruption	-	-	✓	✓
LookAhead for checking all limitations (SWLS, speeds, etc.)	-	-	Only correct with unique assignment to exactly one rotary axis for orientation	✓
Change in orientation can be distributed to multiple segments	✓	✓	✓	-

24796131/EN – 12/2017

	Program 1: *TARGET AXIS*	*Program 2:* *TARGET CART*	*Program 3:* *LINEARLYcoordinated*	*Program 4:* *LINEARLY synchronized*
Cartesian motion parameters can be controlled explicitly	-	✓	-	-
Traveling through singularities	✓	-	-	-
Min. time required per path segment	> 30 ms	> 30 ms	> 300 ms	> 300 ms

INFORMATION

If you are using the HandlingKinematics technology module with the MOVIPLC® Power, the minimum time required per path segment is reduced by at least a factor of 3. This means that using MOVIPLC® Power allows considerably faster path movements.

INFORMATION

Working with HandlingKinematics does not generally require you to understand and work through the SRL program code.

For those who want to understand the program better, the program code run through in *program 1* is presented and explained in the appendix; see the "SRL commands and programs" (→ 🗎 208) chapter. The SRL language[1], which is explained in the appendix, is used in this process.

1) SRL = SEW Robot Language

6.6.1 Changing a program

A program can only be changed if the current program has been initialized or finished. If the program number is changed when the current program has already started but not yet finished, the following error message is issued.

Error	General meaning	Exact error message
FDB136	The program cannot be executed with the selected program number.	Program change is only possible if the Actual Program Number: 4 has not been executed yet.

If you want to abort a program and start another one, you will have to initialize the current program. Alternatively, you can switch to an operating mode other than *Auto program* or *Program step*. The program will be reset as a result.

6.6.2 Program number 1: TARGET AXIS

In *program 1*: *TARGET AXIS*, the kinematic model is moved axis-by-axis from the current pose to the next target pose. In this process, every axis moves to the axis position that is necessary in order for the tool (TCP)[1] to reach the Cartesian target pose. The motion profiles set and synchronization settings in the selected segment parameter set are used for axis interpolation; see the "Segment parameter sets for program 1 and BackToPath" (→ 🗎 91) chapter.

24796131/EN – 12/2017

In this process, the system travels to *target poses 1 to n*, where n is specified by the first end signal and/or the process data profile; see the "End signals" (→ 🗎 56) and "Communication and process data exchange" (→ 🗎 36) chapters. Travel to the relevant target pose continues until the values fall below both blending criteria:

- *Translational distance* < *Blending distance to following segment*
  - Specified via the fieldbus; see the "Blending" (→ 🗎 57) and "Overview" (→ 🗎 37) chapters:" Profile 1..4 with 32/60/88/116PD" (→ 🗎 37).
- *Rotational distance* < *Rotational blending*
  - Configured in the segment parameter set; see the "Segment parameter sets for program 1" (→ 🗎 91) chapter.

If both criteria are met, the system travels to the next *target pose*, at which it proceeds according to the same schema. Only the last *target pose n* is always approached precisely. Afterwards, the program ends.

1) TCP = tool center point

6.6.3 Program number 2: TARGET CART

In *program 2: TARGET CART*, the kinematic model is moved in a Cartesian manner from the *current pose* to the next *target pose*. In this process, every Cartesian degree of freedom (X, Y, Z and A) moves to the Cartesian *target pose*. The motion profiles set and synchronization settings in the selected segment parameter set are used for interpolation in the Cartesian space; see the "Segment parameter sets for program 2" (→ 🗎 92) chapter.

In this process, the system travels to *target poses 1 to n*, where n is specified by the first end signal and/or the process data profile; see the "End signals" (→ 🗎 56) and "Communication and process data exchange" (→ 🗎 36) chapters. Travel to the relevant target pose continues until the values fall below both blending criteria:

- *Translational distance* < *Blending distance to following segment*
  - Specified via the fieldbus; see the "Blending" (→ 🗎 57) and "Overview" (→ 🗎 37) chapters:" Profile 1..4 with 32/60/88/116PD" (→ 🗎 37).
- *Rotational distance* < *Rotational blending*
  - Configured in the segment parameter set; see the "Segment parameter sets for program 2" (→ 🗎 92) chapter.

If both criteria are met, the system travels to the next *target pose*, at which it proceeds according to the same schema. Only the last *target pose* is always approached precisely. Afterwards, the program ends.

6.6.4 Program number 3: LINEARLY coordinated

For *program 3*: *LINEARLY coordinated*, the kinematic model's tool travels on a translationally and geometrically defined path in space (Cartesian degrees of freedom X, Y and Z). The path is defined by a sequence of straight sections at whose intersections (i.e. the target poses) blending is performed; see the "Interpolation types" (→ 🗎 30) chapter. The straight sections are defined by the connection of the path points (start position and *target positions 1 to n*). At their intersections, an area defined by the blending distance is blended; see the "Blending" (→ 🗎 31) chapter.

In addition, the value must fall below the following blending criterion:

- *Rotational distance* < *Rotational blending*
  - Configured in the segment parameter set; see the "Segment parameter sets for programs 3 and 4" (→ 🗎 93) chapter.

If this criterion is met, the system travels to the next *target pose*, at which it proceeds according to the same schema. Only the last *target posen* is always approached precisely.

After travel to all path segments, the program ends. In this process, the last path segment is marked by the first end signal and/or specified by the process data profile.

The motion profiles and blending parameters in the selected segment parameter set are used for path interpolation; see the "Segment parameter sets for programs 3 and 4" (→ 🗎 93) chapter.

INFORMATION

In order to prevent collisions, the system must not reach an interfering contour when traveling within the sphere with the target pose as the center point and the blending distance as the radius. Outside of the blending range, the kinematic model has translational path fidelity. It always travels exactly on the defined straight sections. You must also ensure that no collisions can occur on these sections.

INFORMATION

In *program 3*: *LINEARLY coordinated*, the positioning can return to the geometrically defined path after an interruption (BackToPath). You can find more detailed information in the "BackToPath" (→ 🗎 58) chapter.

6.6.5 Program number 4: LINEARLY synchronized

For *program 4: LINEARLY synchronized*, the kinematic model's tool travels on a translationally and geometrically defined path in space (Cartesian degrees of freedom X, Y and Z). The orientation is interpolated synchronously to the translational movement.

The path is defined by a sequence of straight sections at whose intersections (i.e. the target poses) blending is performed. The straight sections are defined by the connection of the path points (start position and *target positions 1 to n*). At their intersections, an area defined by the blending distance is blended. Rotary blending criteria are not taken into account.

After travel to all path segments, the program ends. In this process, the last path segment is marked by the first end signal and/or specified by the process data profile.

INFORMATION

In *program 4: LINEARLY synchronized*, the positioning can return to the geometrically defined path after an interruption (BackToPath). You can find more detailed information in the "BackToPath" (→ 🗎 58) chapter.

6.6.6 Rapid stop ramps in program mode

The following table shows the rapid stop ramps in program mode.

Rapid stop	Program 1	Program 2	Program 3	Program 4
Axes 1 – 4	Rapid stop ramps of the axes; see the "System limits of the axes" (→ 🗎 81) chapter.			

24796131/EN – 12/2017

<table>
<tr><th>Rapid stop</th><th>Program 1</th><th>Program 2</th><th>Program 3</th><th>Program 4</th></tr>
<tr><td>XYZ</td><td></td><td>Rapid stop ramps of the Cartesian degrees of freedom XYZ; see the "Cartesian motion parameters" (→ 88) chapter.</td><td>Rapid stop ramp for path interpolation; see the "Continuous path motion parameters" (→ 89) chapter.</td><td rowspan="2">Rapid stop ramp for path interpolation; see the "Continuous path motion parameters" (→ 89) chapter.</td></tr>
<tr><td>A</td><td></td><td colspan="2">Rapid stop ramp of the Cartesian degree of freedom A; see the "Cartesian motion parameters" (→ 88) chapter.</td></tr>
</table>

6.6.7 Program sequence

Each of the programs follows the following general SRL program sequence:

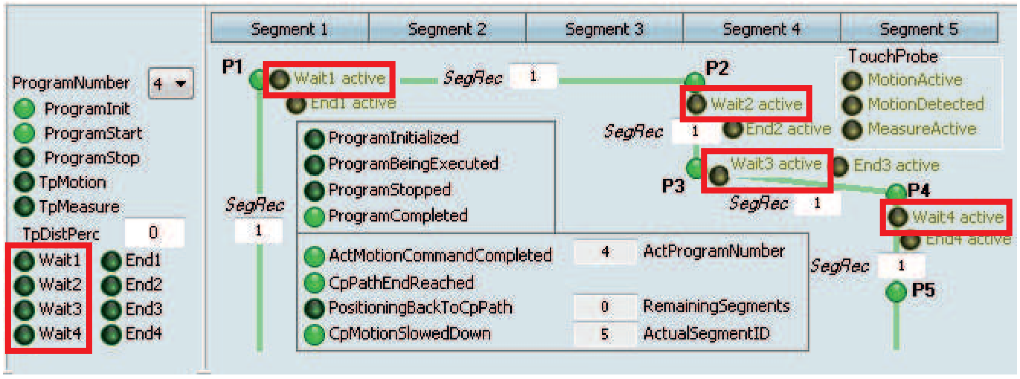
1. Preparing the motion sequence
   a. System settings using system variables.
   b. Preparing the Touchprobe function if it is activated.
2. Motion sequence
   a. Path segment 1
      i Movement to *target pose 1*.
      ii Execute *Touchprobe Measure* if this Touchprobe function is activated.
      iii Check *end signal 1*. If it is activated, skip to the end.
      iv Wait until *wait signal 1* is deactivated.
   b. Path segment 2
      i Blending from path segment 1 to path segment 2.
      ii Movement to *target pose 2*.
      iii Check *end signal 2*. If it is activated, skip to the end.
      iv Wait until *wait signal 2* is deactivated.
   c. Third to second-to-last path segments the same as path segment 2.
   d. Last path segment
      i Blending from second-to-last path segment to last path segment.
      ii Movement to last *target pose*.
3. End
   a. Execute *Touchprobe Motion* if this Touchprobe function is activated.
   b. Wait for it to stop.
   c. End of program.

24796131/EN – 12/2017

6.7 Functions in program mode

6.7.1 Wait signals

You can use wait signals to force stopping at the specified path points. This allows you to enable travel of path segments one step at a time.



14102276491

The number *n* of possible wait signals depends on how many path segments and which process data profile you use. The O4^r:8 *Wait 1 active* – *Wait n active* signals are activated with the I4^r:8 *Wait 1* – *Wait n* signals the first time the program is started. The I4^r:8 *Wait 1* – *Wait n* signals are evaluated when a program starts. Wait signals cannot be enabled once a program sequence has started. If the I4^r:8 *Wait 1* – *Wait n* signals are reset to *FALSE*, the program continues to be processed for each one (enabled to travel the next path segment).

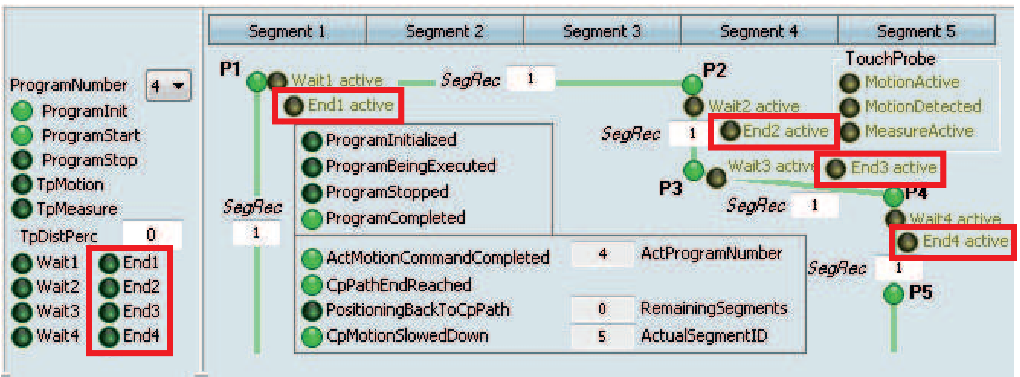
INFORMATION

You can query all the following output data to check whether the program reached the *target pose* to which the wait signal applies:

- O4^r:3 *Program is completed* = *FALSE*.
- O26^r:0 *Actual motion command completed* = *TRUE*.
- O32^r:High *Current segment ID* = wait signal ID.

6.7.2 End signals

The end signals allow you to move the end of the program to any point; i.e. the motion sequence is ended as soon as it reaches the point at which the end signal is *TRUE*. This means that the program can include 1 to 20 path segments.



14102279179

24796131/EN – 12/2017

The number *m* of possible end signals depends on which process data profile you use. The O4r:12 *End 1 active – End m active* signals are activated with the I4r:12 *End 1 – End m* signals the first time program is started. The I4r:12 *End 1 – End m* signals are evaluated when a program starts. Multiple end signals may be activated, but only the first active one in the motion sequence has an effect. If one of the end signals is *TRUE*, the program skips to the end after the relevant path segment. End signals cannot be enabled once a program sequence has started.

INFORMATION

Traveling directly to the same *target pose* multiple times in sequence takes time even though there is no movement. In order to indicate the end of a motion sequence without unnecessarily wasting cycle time, the end signal must be used instead of filling out the last path points with identical values.

6.7.3 Blending

The *blending distance* is indicated in the path segment with the *target pose* to which blending takes place. For example, the *Blending distance to segment 2* (I17r) is specified for movement to *target position 2* in *segment 2* (I13r – I15r). In spatial terms, this blending distance specifies the translational distance to *target position 1* (I8r – I10r) after which the movement is transferred to *target position 2* (I13r – I15r).

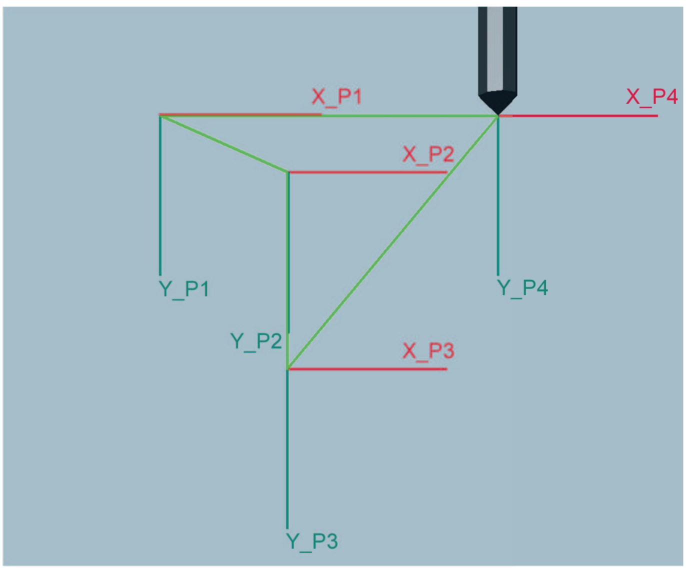
Please note the following:

INFORMATION

Set the *Blending Distance* parameter to a value of less than half the distance between two points. Otherwise, the program will switch directly to the next *target pose*.

- The following is critical for **programs 1, 2** and **3**: Overlapping blending ranges for two consecutive poses could prevent travel to a position or orientation.
- In **program 4**, blending is automatically limited to prevent this problem.

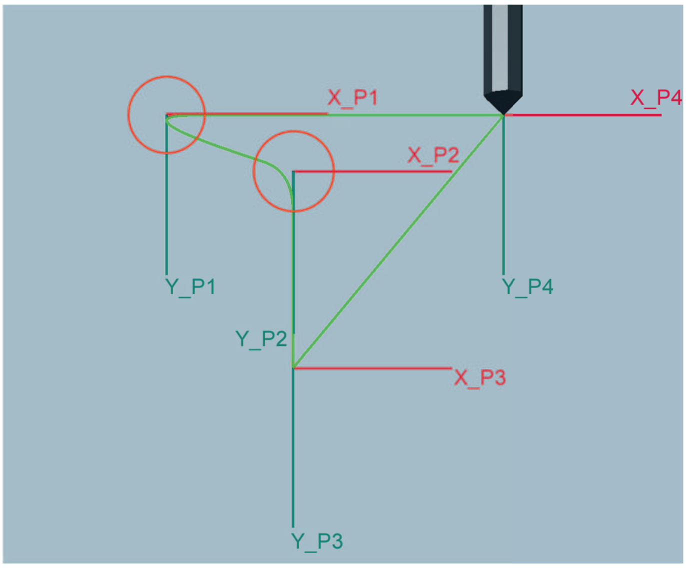
Example for program 1 or 2 without blending



14104044043

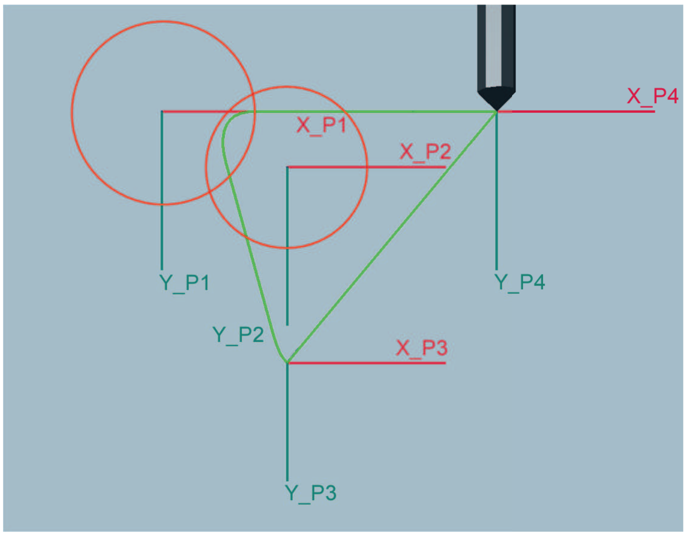
24796131/EN – 12/2017

Example for program 1 or 2 with a sufficiently short blending distance



14104046731

Example for program 1 or 2 with an excessive blending distance



14104049163

6.7.4 BackToPath

BackToPath is used in programs 3 and 4, in other words, the programs that use path interpolation (CP).

If the system leaves the specified path, repositioning onto the path must be initiated in order to continue the movement. The system leaves the path if the inverters lose their enable during program processing (e.g. left at MOVIAXIS® FCB 10).

The inverter's enable is deactivated in the following situations:

- Inverter error (e.g. lag error)
- Controller inhibit for the inverter
- Emergency stop/STO
- I1ʳ:1 *Enable/rapid stop* = *FALSE*, etc.

24796131/EN – 12/2017

If the inverters are subsequently enabled again, the O26^{r}:2 *Positioning back to CP path active TRUE* signal is set. Positioning returns to the path if the I4^{r}:1 *Program start* signal has a rising edge. The BackToPath movement can be stopped and continued (with level control) using the I4^{r}:1 *Program start* signal; i.e. the I4^{r}:1 *Program start* signal must remain set to allow repositioning. Repositioning is completed if the O26^{r}:2 *Positioning back to CP path active* signal is *FALSE* again. Now the program can be resumed with a rising edge for the I4^{r}:1 *Program start* signal.

Repositioning is performed in the axis interpolation that corresponds to axis-by-axis positioning. The speeds and accelerations used are 10% of the motion parameter values from the *Jog Axis* jog mode. The jerk time set for the axes in segment parameter set 1 is used for the jerk time.

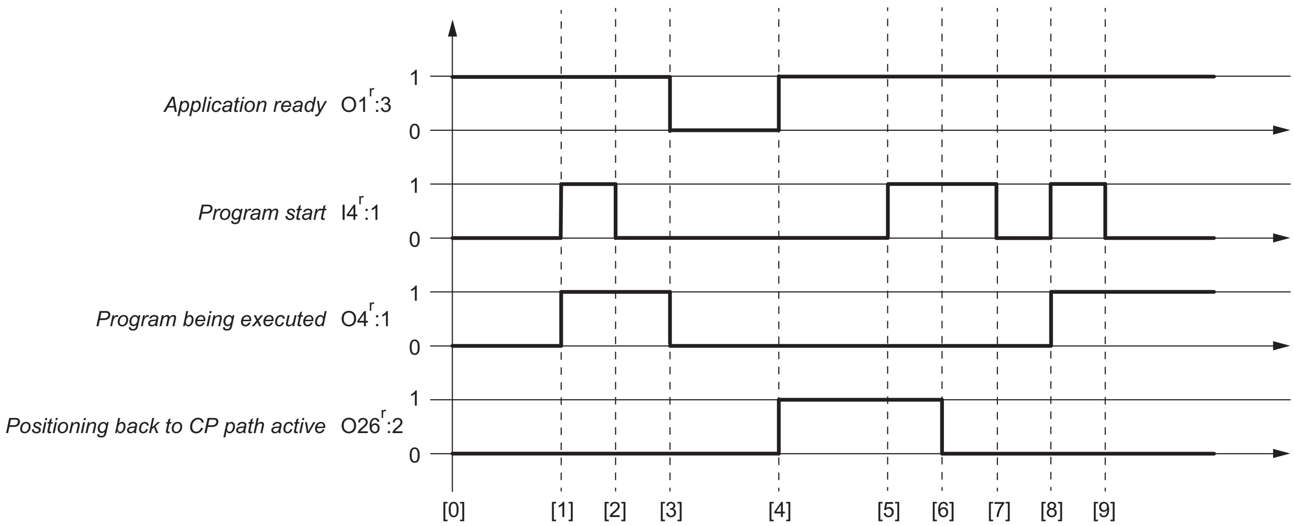
When is BackToPath necessary?

An interruption in program processing is detected by the O3^{r}:3 *Application ready = FALSE* signal. In order to continue program processing, you must wait until the signal changes back to O3^{r}:3 *Application ready = TRUE*. As soon as the O3^{r}:3 *Application ready* signal is *TRUE* again, it can be determined whether the program can be continued directly (because it has not left the path) or whether repositioning must be performed first. The O26^{r}:2 *Positioning back to CP path active* signal is evaluated for this purpose. If this signal is *TRUE*, repositioning starts with a rising edge for the I4^{r}:1 *Program start* signal.

INFORMATION

BackToPath can only be performed if the *Auto program* (I2^{r}:3) program mode remains selected. If you have switched to *Jog axis by axis/Cartesian* in signal I2^{r}:0/1 the meantime, repositioning to the path cannot be performed.

Cycle diagram



9007213255429131

Process

No.	Description
[0]	The program is initialized (O4^{r}:0 *Program initialized* = *TRUE*).
[1]	The program is started by a rising edge for the I4^{r}:1 *Program start* signal.

No.	Description
[2]	Resetting the I4^r:1 *Program start* signal to *FALSE* (not until after fieldbus has returned the O4^r:1 *Program being executed* = *TRUE* signal).
[3]	The system detects that processing of the program was interrupted (O3^r:3 *Application ready* = *FALSE*), e.g. because of an emergency stop or *Enable/rapid stop*. **INFORMATION:** Checking whether the system left the path requires continuous monitoring of the following variables: • O3^r:3 *Application ready* • O26^r:2 *Positioning back to CP path active*
[4]	As soon as the O3^r:3 *Application ready* signal changes back to *TRUE*, meaning that the reason for the interruption has been eliminated, the O26^r:2 *Positioning back to CP path active* signal indicates that repositioning to the path will be performed.
[5]	Repositioning is triggered by a rising edge for the I4^r:1 *Program start* signal. The kinematics returns to the path. The rising edge must remain set or be repeatedly set until repositioning is completed.
[6]	Repositioning is ended as soon as the O26^r:2 *Positioning back to CP path active* signal is *FALSE* again.
[7]	Resetting the I4^r:1 *Program start* signal to *FALSE* (not until after fieldbus has returned the O4^r:1 *Program being executed* = *TRUE* signal).
[8]	Now the actual program can be resumed with another edge of the I4^r:1 *Program start* signal.
[9]	Resetting the I4^r:1 *Program start* signal to *FALSE* (not until after fieldbus has returned the O4^r:1 *Program being executed* = *TRUE* signal).

6.7.5 LookAhead

In *program 4*, all limitations for the kinematic model are checked before a movement is made. If one of the limitations is violated, the movement is automatically adjusted accordingly. The following conditions are checked:

- **Work envelope** (Cartesian, kinematics and axis software limit switches)

  The kinematics stops as soon as a violation is detected and reports a fault.
- **Motor speeds and rapid stop accelerations** (axis-by-axis and Cartesian A)

  HandlingKinematics automatically adjusts the motion profiles so that the permitted rapid stop delays (for accelerating and decelerating) and motor speeds are not exceeded. This can slow down the movement.

You can check whether the movement was adjusted using the O26^r:3 *CP motion slowed down* signal. You can find detailed information in the MessageHandler. For this purpose, set the following variables in the advanced diagnostics; see the "Advanced diagnostics" (→ 🗎 132) chapter:

- *AxisGroupKin.Inst[1].In.Diag.Kin.MessageAtAutoSlowdown*
- *AxisGroupKin.Inst[1].In.Diag.Kin.CartRotation*
- *AxisGroupKin.Inst[1].In.Diag.Kin.Centrifugal*
- *AxisGroupKin.Inst[1].In.Diag.Kin.MotorAccDec*
- *AxisGroupKin.Inst[1].In.Diag.Kin.MotorSpeed*

24796131/EN – 12/2017

6.7.6 Touchprobe function: Overview

If you want a Touchprobe (Touchprobe sensor) in the kinematic model to determine the position for gripping or depositing a workpiece, you can use the Touchprobe function in HandlingKinematics. There are 2 typical cases:

- As soon as the Touchprobe is tripped, another remaining specified distance along the spatial direction of the last path segment needs to be traveled. For this purpose, HandlingKinematics offers the Touchprobe function for the "Touchprobe Motion" sensor-based positioning. To use this function, you must ensure that the Touchprobe is not covered, for example by the gripped workpiece.
- If the Touchprobe is covered by the gripped workpiece, the measurement of the next depositing position must already be taken after the previous workpiece is deposited and during travel away from the pile of deposited workpieces. HandlingKinematics offers the "Touchprobe Measure" Touchprobe function for this purpose. The PLC specifies the measured position as the *target position* during the next depositing process.

The Touchprobe functions are activated using the I4^r:5 *Touchprobe Motion* or I4^r:6 *Touchprobe Measure* signals. Both Touchprobe functions can also be used in combination in a single motion sequence. The O4^r:5 *Touchprobe Motion active* and/or O4^r:7 *Touchprobe Measure active* signals report if they have been activated. In addition, the "Touchprobe Motion" Touchprobe function issues the O4^r:6 *Touchprobe Motion Detected* signal as soon as the controller has received the signal from the Touchprobe.

6.7.7 Touchprobe function: Touchprobe Motion

With the I4^r:5 *Touchprobe Motion* signal, the "Touchprobe Motion" Touchprobe function is activated at the first rising edge of the I4^r:1 *Program start* signal for starting the motion sequence. This is indicated by the O4^r:5 *Touchprobe Motion active* = TRUE signal.

In the last path segment of the motion sequence, the Touchprobe function is enabled and waits for the Touchprobe signal. Once this signal is detected, the O4^r:6 *Touchprobe Motion detected* signal is set to *TRUE*. Sensor-based positioning starts. If no Touchprobe signal is received, the system travels to the last target point. In both cases, the program ends afterwards.

In this process, the last path segment can be determined by an end signal; see the "End signals" (→ 🗎 56) chapter. If no end signal is activated, the last path segment is the segment defined in the selected process data profile, see the "Communication and fieldbus" (→ 🗎 36) chapter. This means:

- If no end signal is TRUE, *Touchprobe Motion* is performed in the last path segment.
- If the *End 1* end signal is *TRUE*, *Touchprobe Motion* is performed in the first path segment.
- If the *End 2* end signal is *TRUE*, *Touchprobe Motion* is performed in the second path segment.
- etc.

Sensor-based positioning

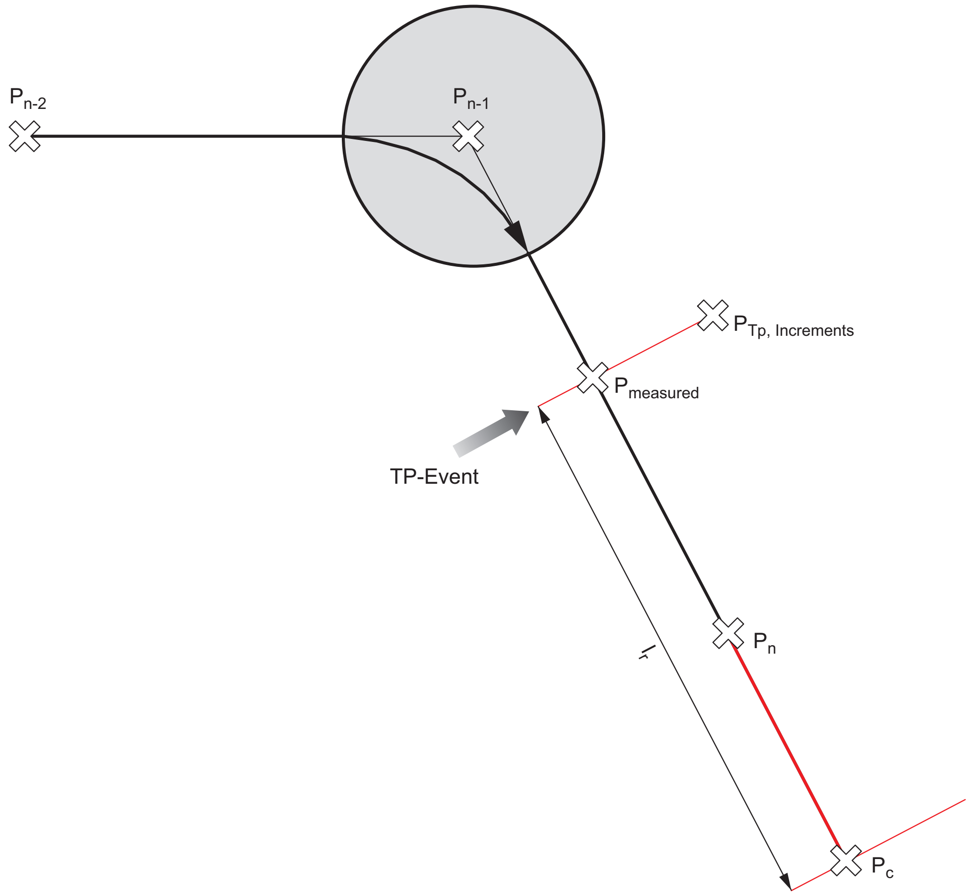
As soon as the Touchprobe signal is received, the system travels the remaining travel distance l_r to the new target point P_c. In this case, the direction $\bar{r}$ corresponds to that of the last path segment, i.e. the connecting line between P_{n-1} and P_n. The new target point P_c can also be further away than the original target point P_n.

The motion type for sensor-based positioning is always Cartesian interpolation with synchronization, regardless of the program number selected. The Cartesian motion parameters are always taken from the segment parameter set of the (last) segment in which the Touchprobe function was performed.

NOTICE

Sensor-based positioning is performed with the Cartesian motion parameters of the segment parameter set that was assigned to the last path segment. **PLEASE CONFIGURE CORRECTLY!**

Calculating the new target point



14104466827

Position	Description
P_{n-2}	Third-to-last *target pose*
P_{n-1}	Second-to-last *target pose*
P_n	Last *target pose*
$P_{Tp, Increments}$	Position of the tool in axis increments at the time of the Touchprobe event
$P_{measured}$	Cartesian position of the tool at the time of the Touchprobe event, which is projected onto the last path segment.
P_c	New *target pose*
l_r	Remaining travel distance
TP event	Touchprobe event

24796131/EN – 12/2017

The remaining travel distance l_r is calculated as follows:

$$l_r = l_{\text{Remaining travel segment}} \cdot \frac{f_{\text{PD-IN HighByte}}[\%]}{100}$$

13862892939

$l_{\text{Remaining travel segment}}$ — Remaining travel that can be configured in every segment parameter set, see the "Overview of segment parameters" (→ 89) chapter.

$f_{\text{PD-IN HighByte}}$ — Percentage for scaling the configured remaining travel (I6ʳ:High *TpRemainingDistPerc*)

The tool's Cartesian position P_{Tp} is calculated from the axis increments of the kinematic axes as the time of the Touchprobe event so that it is on the connecting line between the last *target position* P_n and the second-to-last *target position* of the path segment P_{n-1}.

The direction vector $\vec{r}$ for sensor-based positioning, which points along the connecting line from P_{n-1} to P_n, is calculated as follows (unit vector):

$$\vec{r} = \frac{(\vec{P}_n - \vec{P}_{n-1})}{\left|\vec{P}_n - \vec{P}_{n-1}\right|}$$

14070411403

The new *target position* to which the Cartesian interpolation is positioned is calculated as follows:

$$P_c = P_{measure} + l_r \cdot \vec{r}$$

14070406923

Process example 1: Effective remaining travel = 0

1. The kinematic model moves toward the last *target pose*.
2. A Touchprobe event occurs.
3. The new *target position* is calculated.
4. The kinematic model returns to the position at which the Touchprobe event occurred (and to the*target orientation n* of the last target point).

Process example 2: Effective remaining travel > 0

1. The kinematic model moves toward the last *target pose*.
2. A Touchprobe event occurs.
3. The new *target position* P_c is calculated.
4. Depending on the current position:
   - If the kinematic model has already traveled beyond the new *target position* P_c, it returns to this position.
   - If the kinematic model has not yet reached the new *target position* P_c, it continues to this position.

INFORMATION

If other Touchprobe events occur in the last path segment, they are discarded.

INFORMATION

The target orientation of the last path segment is applied as the target orientation for sensor-based positioning. Even for very short translational remaining travel (e.g. a fraction of the last segment), complete rotation to the specified target orientation of the last path segment is performed.

6.7.8 Touchprobe function: Touchprobe Measure

With the I4^{r}:5 *Touchprobe Measure* signal, the "Touchprobe Measure" Touchprobe function is activated at the first rising edge of the I4^{r}:1 *Program start* signal for starting the motion sequence. This is indicated by the O4^{r}:7 *Touchprobe Measure active = TRUE* signal.

The Touchprobe function is enabled in the first path segment of the motion sequence. If a Touchprobe signal is received in the first path segment, the Cartesian position of the TCP[1] at which the signal was output is transmitted to the PLC (O7^{r} – O9^{r} *Position / Touchprobe Measure XYZ* if I1^{r}:8 *Output Touchprobe Measure* is set). The PLC can then supply this position (in some cases after modifying it) as the target point for the next movement cycle. If the values fall below the *Blending distance to segment 2* (I17^{r}) when the Touchprobe signal is received, blending to segment 2 is only performed at this point in time. If *target pose 1* (I8^{r} – 11^{r}) is reached before the Touchprobe signal is received, the "FDB950" error is reported (O1^{r}:7 *Application error* and O5^{r}-O6^{r} *ErrorID*). The status of the I4^{r}:8 *Wait 1* signal has no effect on this process.

NOTICE

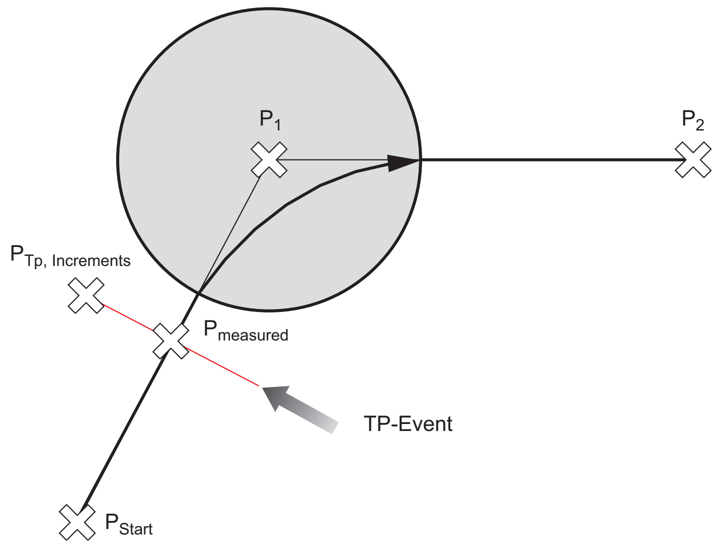
The PLC is only permitted to use the transmitted Touchprobe position as the *target position* of a path segment on the next path to be traveled if this position is valid.

The measured Touchprobe position (O7^{r} – O9^{r} *Position / Touchprobe Measure XYZ*) is valid if all of the following conditions are met:

- The Touchprobe function was activated before the start of the motion sequence: I1^{r}:8 *Output Touchprobe Measure = TRUE* for the first rising edge of the I4^{r}:1 *Program start* signal for starting the motion sequence.
- The program has been executed completely: O4^{r}:3 *Program finished = TRUE*.
- O1^{r}:7 *Application fault = FALSE*.

1) TCP = tool center point.

Calculating the measured position



14104464395

Position	Description
P_{Start}	Start pose/current pose
P_1	*Target pose 1* of the new motion sequence
P_2	*Target pose 2* of the new motion sequence
$P_{Tp, Increments}$	Position of the tool in axis increments at the time of the Touchprobe event
$P_{measured}$	Cartesian position of the tool at the time of the Touchprobe event, which is projected onto the first path segment.
TP event	Touchprobe event

$P_{measured}$ is calculated as follows:

The Cartesian position of the tool at the time of the $P_{measured}$ Touchprobe event is calculated using the kinematic axes' axis increments so that it is on the connecting line through the following two points:

- P_{Start}: *Start position* or *current position* ($O7^r$ – $O10^r$); this generally corresponds to the last *target position* of the last motion sequence.
- P_1: *Target pose* ($I8^r$ – $I11^r$) of the new motion sequence

If the $I1^r$:8 *Output Touchprobe Measure* signal is set, the X, Y and Z coordinates of $P_{Measure}$ are output in signals $O7^r$ – $O9^r$. If it is deactivated, the current position in the kinematics coordinate system (KCS)[1)] is issued there. In this process, observe the validity of the output value (see notice above).

INFORMATION

If other Touchprobe events occur in the first path segment, they are discarded and have no effect on the result of *Touchprobe Measure*.

1) KCS = kinematics coordinate system.

7 Startup

7.1 Requirements

For proper startup of HandlingKinematics, the requirements in the "Project planning notes" (→ 🗎 14) chapter must be met.

Check the inverter installation, the encoder connection and the controller installation based on the installation notes in the operating instructions, the fieldbus manuals and the appendix of this manual.

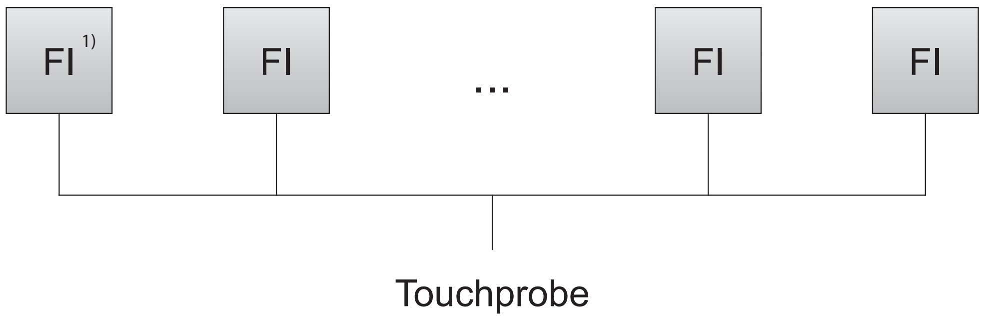
You can find installation notes in the documentation for the relevant units and software components, which are listed in the "Other applicable documentation" (→ 🗎 11) chapter.

7.1.1 Wiring for Touchprobe

To use the Touchprobe function, wire the Touchprobe to all of the kinematic model's inverters, see the "Touchprobe function" (→ 🗎 61) chapter:" Overview" (→ 🗎 61).

The following inputs on the inverter are used for wiring the Touchprobe.

Inverter	Inputs
MOVIDRIVE®	Input DI02 / terminal X13.3
MOVIAXIS®	Input DI02 / terminal X10.3
MOVITRAC® LTX	Input X14-3



9007213310106635

You can find more information on wiring, and particularly on supplying the Touchprobe with voltage, in the documentation for the inverter and the Touchprobe. For use of the Touchprobe function in HandlingKinematics, it is important that the Touchprobe's output is connected to all the kinematic model's inverters.

7.1.2 MOVITOOLS® MotionStudio engineering software

You need the MOVITOOLS® MotionStudio engineering software for startup. The scope of delivery includes the following required software modules:

- The **Drive Startup for MOVI-PLC®** is required to set up the inverters on the motor (motor startup) and establish communication with the controller.
- The **Application Configurator** is required to add and then set up the "HandlingKinematics" application module.

24796131/EN – 12/2017

INFORMATION

We strongly recommend using the following tools to aid configuration and startup:

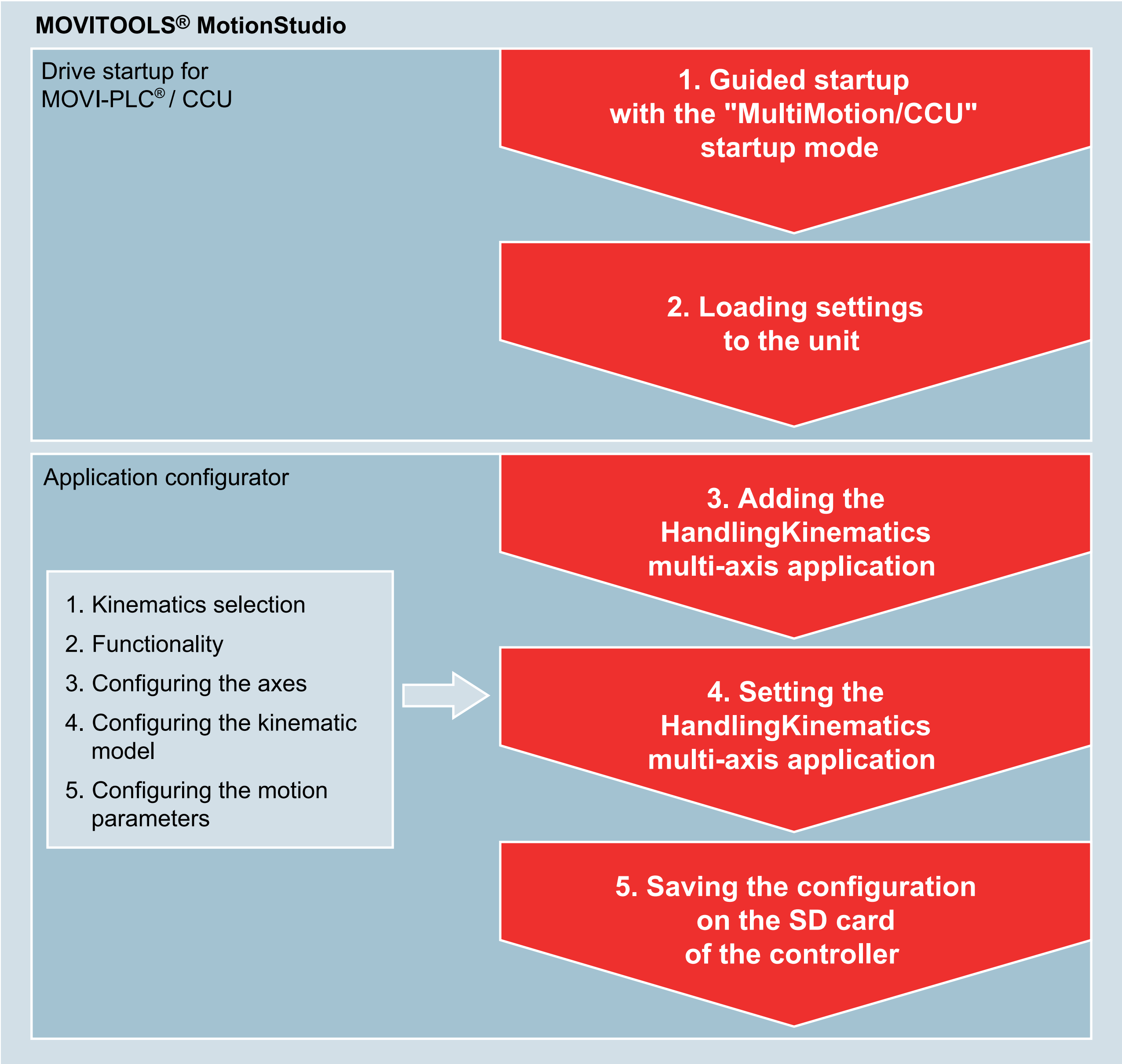
- 3D simulation
  - Adjustment of the configuration and real robot. The configuration is only correct when they match.
  - Start in the MOVITOOLS® MotionStudio under `[Menu] > [Settings] > [Extras]` or by using module diagnostics.
- MessageHandler
  - Detailed information in plain text, e.g. an error message and warnings.
  - Start in the MOVITOOLS® MotionStudio in the controller context menu under "Diagnostics."

7.1.3 Project/network view

The devices to be put into operation must be displayed in the MOVITOOLS® MotionStudio project and/or network view (controller and inverter). The "First steps" topic in the MOVITOOLS® MotionStudio online help provides a good introduction to this subject. It can be accessed under `[Help] > [Show help]`.

7.2 Startup procedure

The following figure shows the entire procedure step-by-step.



14208215435

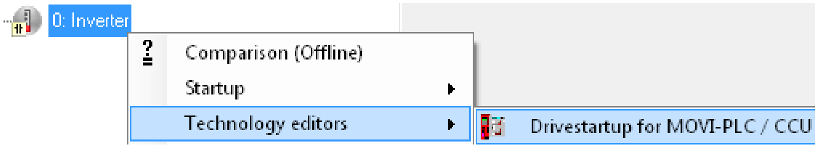
7.3 Starting the Drive Startup for MOVI-PLC®/CCU

1. Start the MOVITOOLS® MotionStudio engineering software.

   ⇨ The devices to be put into operation must be displayed in the MOVITOOLS® MotionStudio project view and/or network view.

24796131/EN – 12/2017

2. Select the inverter in the network view of MOVITOOLS® MotionStudio.



9007213329006475

3. Right-click to open the context menu of the inverter.
4. Select the menu command [Technology editors] > [Drive Startup for MOVI-PLC®/ CCU].

   ⇨ Drive Startup for MOVI-PLC®/CCU is started.

5. Follow the instructions of the wizard.

INFORMATION

A common configuration error is incorrect setting of baud rate and SBus addresses.

- The baud rate must be set identically for all connected devices.
- The SBus addresses must always differ for the various devices.

7.4 Starting the Application Configurator

NOTICE

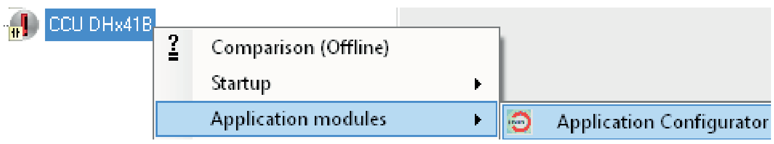
Beware of downloading the software/configuration to the controller when the plant is in operation.

This may result in injury and damage to the system.

- Bring the system into a safe condition.

✓ The devices to be put into operation must be displayed in the MOVITOOLS® MotionStudio project view and/or network view.

1. Select the controller in the network view of MOVITOOLS® MotionStudio.



14074270347

2. Right-click to open the context menu of the controller.
3. Select the [Application modules]/[Application Configurator] menu command.

⇨ The Application Configurator opens.

24796131/EN – 12/2017

The Application Configurator contains the following options for creating or modifying a configuration:

- **Creating a new configuration**

  You can create a completely new CCU configuration here. You can add the application module in the configuration overview page that opens.

- **Open controller configuration / Open configuration file**

  The Application Configurator configuration stored on the controller SD card or saved in a file is downloaded here and displayed in the Application Configurator overview page. An existing application module can then be revised. If the application module has not yet been added, it can be rectified here.

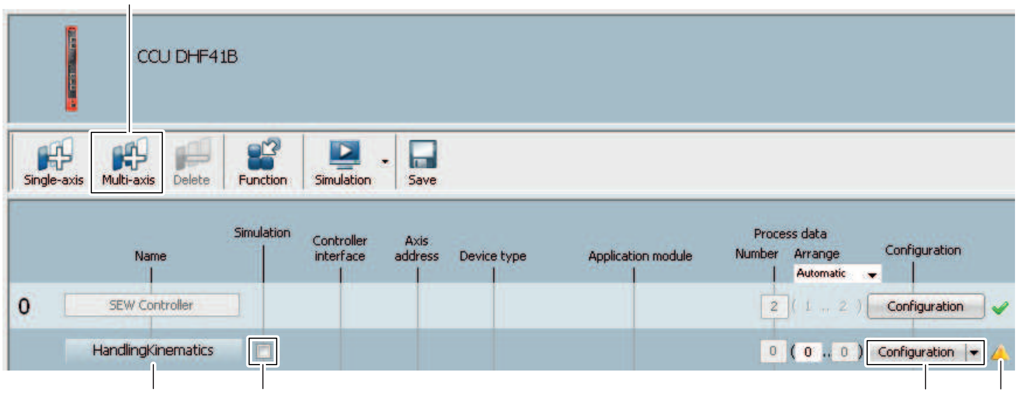
INFORMATION

For detailed information about the functions and operation of the Application Configurator and the other application modules; see the corresponding documentation, which is available under "Documentation" in the controller context menu in the MOVITOOLS® MotionStudio and elsewhere.

7.5 Adding the "HandlingKinematics" application module

Proceed as follows:

1. Click button [1] in the configuration interface of the Application Configurator.



14104584971

⇨ A dialog box for selecting the multi-axis application opens.

2. Select "HandlingKinematics" application module and click [Next].

   ⇨ The Application Configurator returns to the overview page.

   ⇨ The "HandlingKinematics" application module [5] has been added to the configuration.

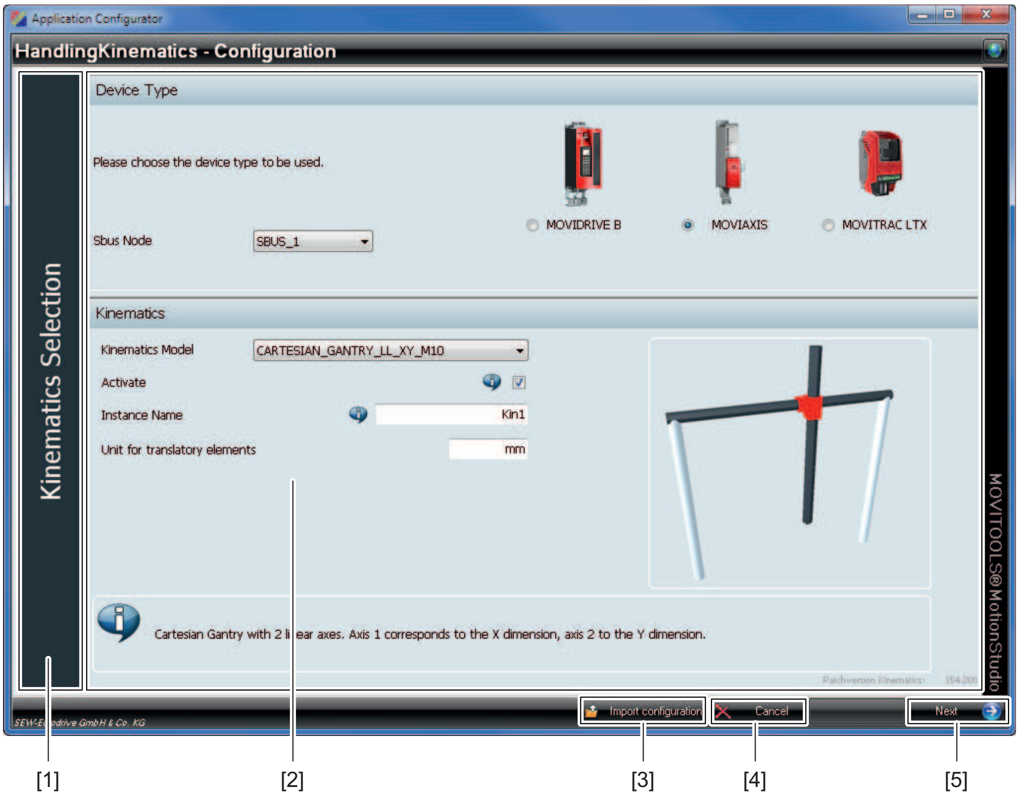
3. If the axes are still physically unavailable but you still want to perform diagnostics, you have the option of activating check box [4]. A virtual simulation of the kinematics is then performed.

24796131/EN – 12/2017

4. Click button [3].
   ⇨ A software wizard for setting the application module appears (Kinematics Configurator).
5. Follow the wizard instructions to perform the kinematic configuration.
   ⇨ Once the wizard is finished, the yellow warning triangle [2] turns into a green check mark.

7.6 Kinematics Configurator

The Kinematics Configurator is set up as follows.



14107223819

No.	Description
[1]	Configuration chapter
[2]	Kinematic configuration
[3]	You can use this button to import a kinematic configuration that was created and saved already at an earlier time. A kinematic configuration is not the same as an Application Configurator configuration. It contains only a kinematic instance and can also be used in the "HandlingKinematics" and "Kinematics" technology modules.
[4]	Click this button to cancel the configuration of the kinematics. The configuration wizard goes to the configuration overview page for the Application Configurator. In this case, the changed configuration values are lost. However, this must be confirmed beforehand.
[5]	Click this button to go to the next configuration page.

7.7 Kinematics selection

On this configuration page, select the device type and kinematic model.

14108792587

No.	Description
[1]	You set the device type and system bus connection in this group: • **Device type:** Choose the inverter type here. **INFORMATION:** All of the axes of the kinematic model must be of the same inverter type. In the SEW-Portfolio, only the selectable device types are suitable for HandlingKinematics. • **System bus connection type:** Here, you set the SBus interface from which the controller is to communicate with the inverters (see the information below). After a network scan in MOVITOOLS® MotionStudio online mode, the connected inverters are displayed in the network tree under the relevant controller interface: – SBUS_1: The first CAN-based SBus interface for the controller (X33). – SBUS_2: The second CAN-based SBus interface for the controller (X32). – NONE: There is no communication between the inverters and the controller.

24796131/EN – 12/2017

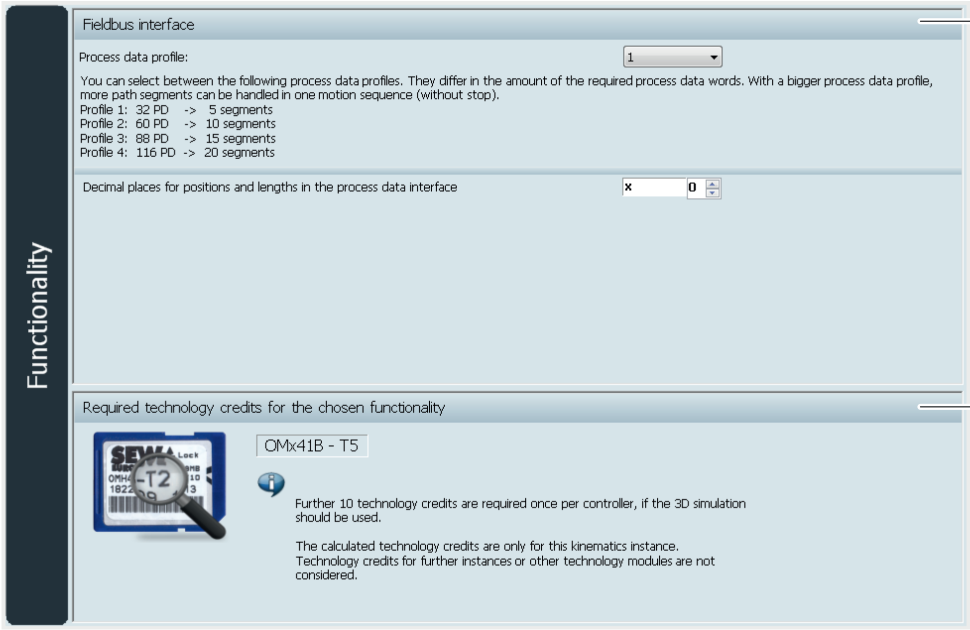
No.	Description
[2]	Set the kinematic model in this group: • **Kinematic model:** Here, select the correct kinematic model to ensure that the kinematic model functions correctly, see the "Kinematic models" (→ 🗎 19) chapter. • **Activated:** You can activate or deactivate the entire kinematic instance here. If you deactivate it, no calculations are performed and no technology points are consumed for this instance. The kinematic instance continues to reserve 8 axes. The advantage of this is that one configuration can be used for multiple machines and only the actual existing machines can be activated on a case-by-case basis. • **Instance name:** Enter the instance name here. The instance name is used to identify the kinematic model in the 3D simulation and in the variable structure. It is also used as a prefix for all of the kinematic axes. It must not be longer than 12 characters. • **Unit for translatory elements:** Here, you enter the unit for translatory elements. This string is used by the Application Configurator later when configuring the correct designation of variables with a track reference. This is primarily intended to help to avoid input errors due to incorrect units. **INFORMATION:** The unit of measure for rotary sizes is degree.

INFORMATION

To use the quicker EtherCat®-based SBus^PLUS system bus, you must switch to the MOVIPLC® with the HandlingKinematics technology module.

7.8 Functionality

On this configuration page, you set the fieldbus interface.



14108893195

No.	Description
[1]	In this group, you configure the fieldbus interface: • **Process data profile:** Select the required process data profile here. • **Decimal position for positions and lengths in the process data interface:** Positions and lengths with decimal positions are transferred to the process data without decimal separator. Chose how many decimal positions there are (see following section). For a detailed list of the process data profiles and process data, refer to chapter "Communication and process data exchange" (→ 🗎 36).
[2]	In this group, the required technology points for the current configuration are displayed, see chapter "Technology points" (→ 🗎 15).

INFORMATION

Only integral numbers can be transferred for position and speed values via fieldbus. For more accuracy, decimal positions can be set in the application module, see chapter "Writing conventions and special features" (→ 🗎 36).

If the decimal positions set in the application module are set greater than 0, the input values in the controller are divided by a factor of 10 to the decimal positions before they are processed as floating-point numbers. On the other hand, the output values in the controller are multiplied by this factor before they are transferred to the PLC.

Examples

- If the decimal positions are set to 2, and a controller input word contains the value 1, the controller is operated with the value 0.01.
- If the decimal positions are set to 2, and a controller output word contains the value 1, this value must be interpreted as 0.01 in the PLC.

Depending on the set number of decimal positions *N* and the user unit for the track measure *U*, the resolution *A* of the positioning changes:

$$A = 10^{-N} [U]$$

The maximum travel range [R_{min}, R_{max}] is restricted due to the data type WORD of the position:

$$R_{min} = -32768 \cdot A$$

$$R_{max} = 32767 \cdot A$$

The following table lists examples of the user unit for the track measure [*U*] = *mm*.

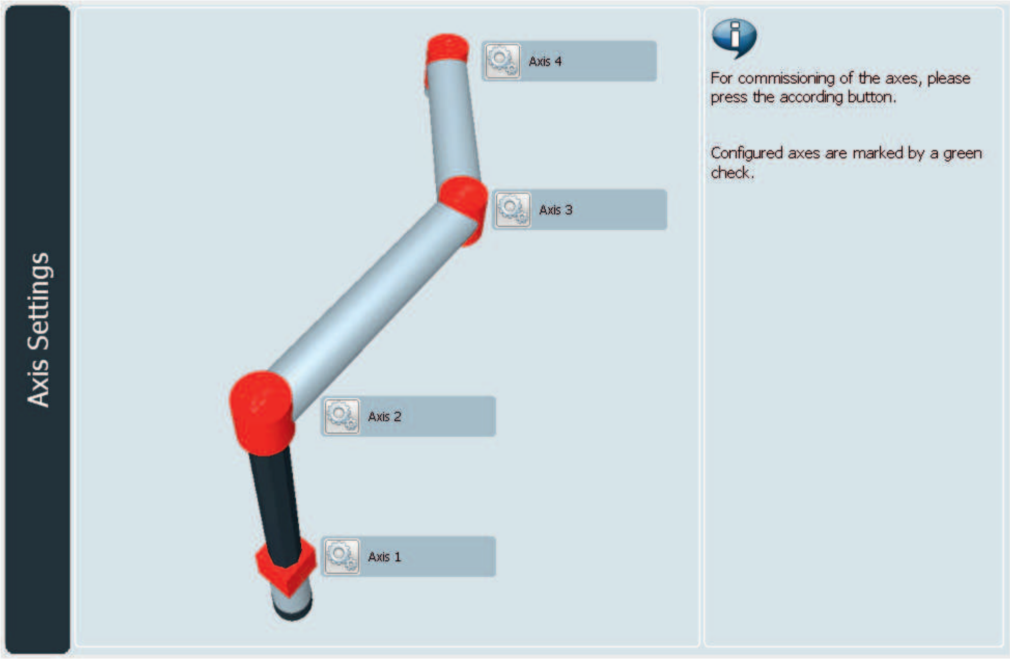
N	*A*	R_{min}, R_{max}	Note
etc.	etc.	etc.	Value *N* "can be set" (→ 96) in the extended configuration. *Config.FbusDecimalPlacesPos*
4	0.0001 mm	−3.2768 mm to 3.2767 mm	
3	0.001 mm	−32.768 mm to 32.767 mm	Value *N* can be set in the standard configuration.
2	0.01 mm	−327.68 mm to 327.67 mm	
1	0.1 mm	−3.2768 m to 3.2767 m	
0	1 mm	−32.768 m to 32.767 m	
−1	10 mm	−327.68 m to 327.67 m	Value *N* "can be set" (→ 96) in the extended configuration. *Config.FbusDecimalPlacesPos*
−2	100 mm	−3276.8 m to 3276.7 m	
etc.	etc.	etc.	

If the travel range is not sufficient with the required resolution and user unit for the track measure, you have to switch to the HandlingKinematics technology module (MOVI-PLC®) and change the position data type to DWORD.

24796131/EN – 12/2017

7.9 Configuring the axes

On this configuration page, you configure all of the axes used in the kinematic model.



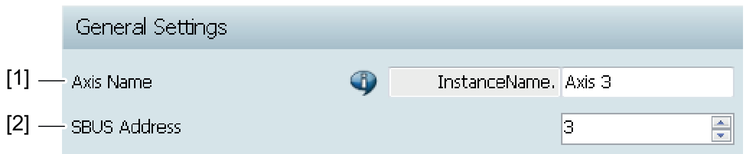
14111656459

To configure the kinematics correctly, you must configure each individual axis. Click the axis startup icon to go to the individual axis configuration, which is described in the subchapters below. After configuring an axis, you return to this configuration page. Fully configured axes are indicated with a green check mark.

If you want to choose [Next] to go the next configuration page before all of the axes are fully configured, the default values are retained in the axis settings.

7.9.1 General settings

In this window, you configure the following settings:



14112907531

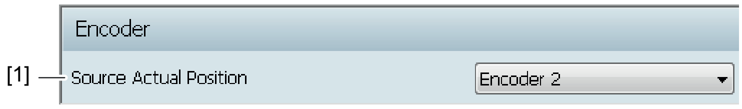
No.	Description
[1]	Enter the axis name here. The axis name is used to make it easier to identify the axis within the kinematic model in the variable structure. It must not be longer than 11 characters.
[2]	You set the SBus address (system bus address) here. The set SBus address must match the SBus address of the inverter to which the axis is to be assigned. In online mode in MOVITOOLS® MotionStudio, the SBus addresses of the inverters are displayed below the relevant controller system before the inverter name. You can find the address setting in the inverter manual.

24796131/EN – 12/2017

From the "Advanced" level onward or from the advanced configuration, you can change the inverter type here if it is different from the general inverter type of the kinematic model; see the "Kinematic selection" (→ 🗎 72) chapter.

7.9.2 Encoder

You can set the encoder in this window.



14112919435

No.	Description
[1]	Here, you select the source of the actual position, i.e. the inverter input at which the encoder signal arrives.

You have the following selection options: Motor encoder or distance encoder.

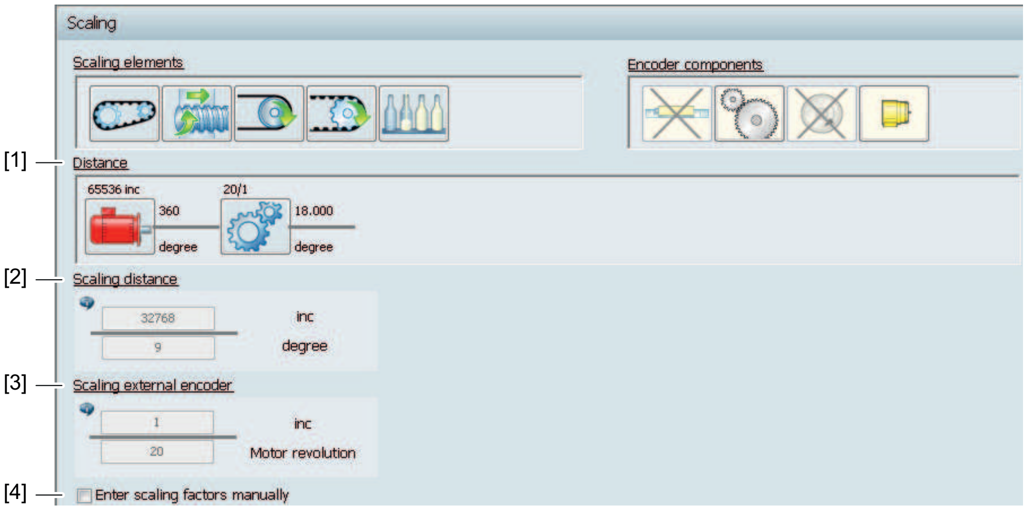
A distance encoder is managed directly from the axis. This means you have to specify the port/slot to which the encoder is connected.

Encoder	MOVIDRIVE® B	MOVIAXIS®	MOVITRAC® LTX
Encoder 1	Motor encoder (X15)	Encoder 1: Motor encoder (X13)	Motor encoder (X15)
Encoder 2	External encoder (X14)	Encoder 2: Encoder card in option card slot 2	
Encoder 3	SSI absolute encoder (X62) (via option card)	Encoder 3: Encoder card in option card slot 3	

24796131/EN – 12/2017

7.9.3 Scaling

In this window, you set the scaling for the path and, if required, for the external encoders.



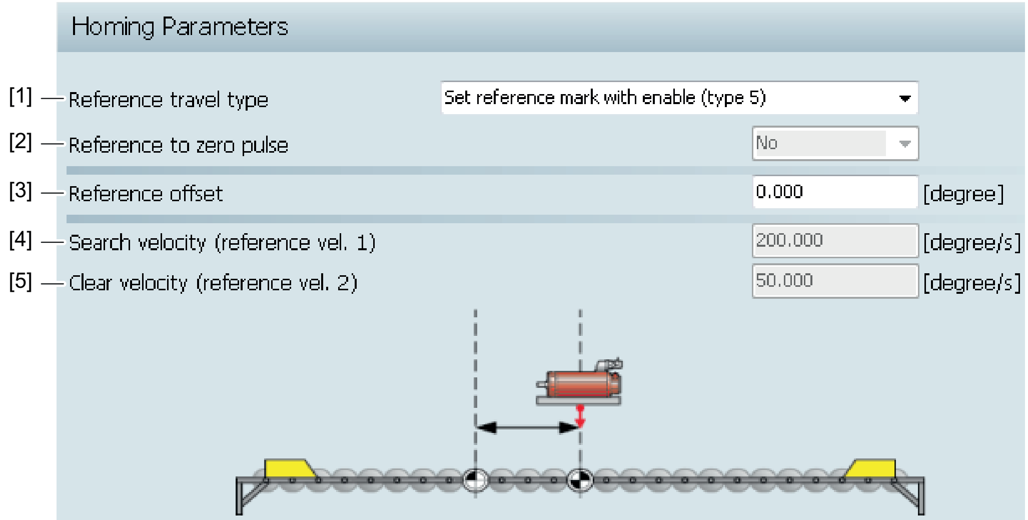
14112937099

No.	Description
[1]	In this area, you can combine the scaling and encoder components into a mechanically connected chain. This should form the actual constellation of the mechanics. The scalings for the path [2] and, if required, for the encoder [3] are calculated from the combined chain. 1. Move the required scaling and encoder components to this area one after the other in the correct sequence. 2. Note that only certain combinations are practical, which is why you cannot select individual elements. 3. You can then configure the gear ratio of the selected component in more detail by double clicking the relevant icon.
[2]	The scaling of the distance or angle is displayed here. The distance scaling states the number of increments by which the encoder is incremented when the axis is moved by a certain distance or rotated by a certain angle. This can be configured manually when [4] is activated.
[3]	The scaling for the external encoder is displayed here when the source of the actual position has been set appropriately (encoder 2/3). It states the number of increments by which the external encoder is incremented when the motor rotates by a certain number of revolutions. This can be configured manually when [4] is activated.
[4]	When you activate this check box, the scaling factors for [2] and [3] can be typed in directly and are no longer calculated from the mechanical chain configured in area [1].

24796131/EN – 12/2017

7.9.4 Reference travel parameters

You set the reference travel parameters in this window.



14112941963

No.	Description
[1]	You select the reference travel type here. To configure the axis reference travel, first select the reference travel type. Different reference travel types are available depending on the unit that is used. For information about the procedure, see the documentation for the relevant device. Depending on the selected reference travel type, the following options can be set.
[2]	Here, you choose whether the referencing takes place on the edge change of the reference cam or on the subsequent encoder zero pulse.
[3]	Enter the reference offset here. Depending on the reference point that was found by the reference travel, you can move the axis zero using the reference offset. **Example**: An axis reports 0 mm as the actual position after referencing without a reference offset. When a reference offset of 200 is set, the axis will report 200 mm as the actual position instead.
[4]	Enter the search speed (reference speed 1) here.
[5]	Enter the retraction speed (reference speed 2) here.

Its significance or the procedure is dependent on the reference travel type (see the inverter documentation).

Reference travel type		MOVIDRIVE®	MOVIAXIS®	MOVITRAC® LTX
Zero pulse	Positive direction	-	-	✓
	Negative direction	✓	✓	✓
Reference cam	Positive end	✓	✓	No zero pulse
	Negative end			

24796131/EN – 12/2017

Reference travel type		MOVIDRIVE®	MOVIAXIS®	MOVITRAC® LTX
Limit switch	Positive	✓	✓	-
	Negative			
Set reference point	With enable	(type 5)	(type 5)	✓
	Without enable	(type 8)	(type 8)	
Reference cam flush	Limit switch positive	✓	✓	-
	Limit switch negat-ive			
Fixed stop	Positive	-	✓	✓
	Negative			
Deactivated		-	✓	✓

7.9.5 Touchprobe function

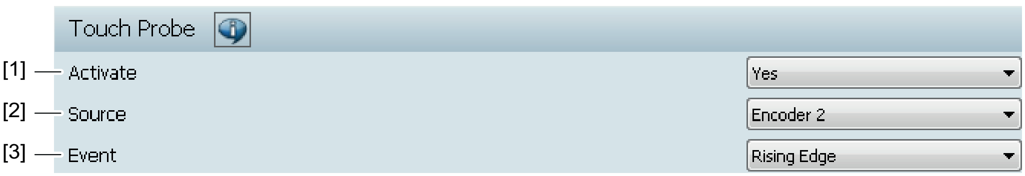
If the application is to use sensor-based positioning (Touchprobe Motion) or position measurement (Touchprobe Measure), activate the Touchprobe function for all of the axes of the kinematic model. You can view the wiring in the "Wiring for the Touchprobe" (→ 🗎 66) chapter.

INFORMATION

Two settings are permitted for the kinematics axes:

- The Touchprobe function is activated for **all** axes of the kinematic model.
- The Touchprobe function is activated for **none** of the axes of the kinematic model.

You can activate the Touchprobe function in this window.

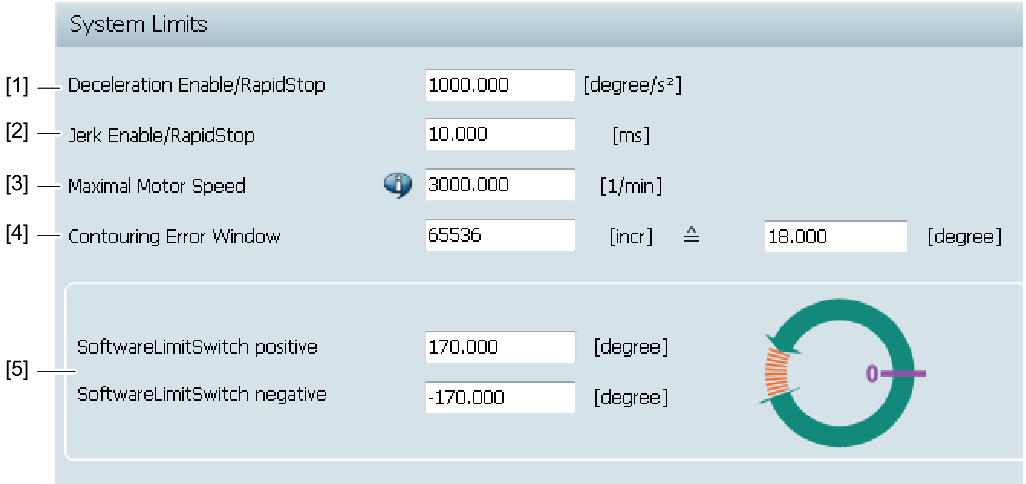


14113126027

No.	Description
[1]	You can activate or deactivate the Touchprobe function of the axis here. This setting must be the same for all of the axes of the kinematic model.
[2]	You can set the source for the axis position for the Touchprobe function here. For *Touchprobe Measure* and *Touchprobe Motion*, the *actual pose* at the time of the Touchprobe event must be determined. The actual position of the axes is required to do this. Generally, the same source as the *actual position source* can be selected here, see the "Encoder" (→ 🗎 77) chapter.
[3]	Here, you select whether you want the software to respond to a rising and/or falling signal edge for the detection of the Touchprobe event.

7.9.6 System limits of the axes

You set the system limits for the axes in this window.



14113130891

No.	Description
[1]	Here, you enter the deceleration for *Enable/rapid stop*[1)] that you want to use to decelerate the axis in the event of an axis-by-axis rapid stop. For information about the procedure, see the "Rapid stop" (→ 🗎 33) chapter.
[2]	Here, you enter the jerk time for *Enable/rapid stop*[1)], in which the full deceleration of the axis for an axis-by-axis rapid stop is structured. This is used for jerk limitation, i.e. to prevent impacts and damage to the mechanics.
[3]	You enter the maximum motor speed here. The maximum possible motor speed can be read on the drive nameplate. It is used by the kinematic control for motion planning. It has no effect on the actual maximum motor speed of the output stage and motor. **INFORMATION:** The input value must not exceed the maximum possible motor speed of the output stage and motor. Otherwise, the inverter may receive a fault status during the motion.
[4]	Here, you enter the lag error window for the positioning (that is, the dynamic deviation the target position can have from the actual position before an error is triggered). This setting is written to the inverter in the correct scaling and then monitored by the inverter. The error response is permanently set to *Emergency stop/waiting* for the MOVIAXIS®. For the MOVIDRIVE®, it is *Emergency stop/malfunction* by default but can be configured in the parameter tree.
[5]	Here, you set the permitted range for the axis position. If the axis travels outside of this range, the kinematic model moves to an error status; see the "Rapid stop" (→ 🗎 33) chapter.

1) Motion parameters for axis-by-axis rapid stop.

INFORMATION

The software limit switches are set with reference to the axis zero and the direction of movement of the real axis:

- The axis zero is the motor position with 0 increments.
- The direction of movement is positive when the motor increments are rising.

24796131/EN – 12/2017

7.9.7 Jog Axis

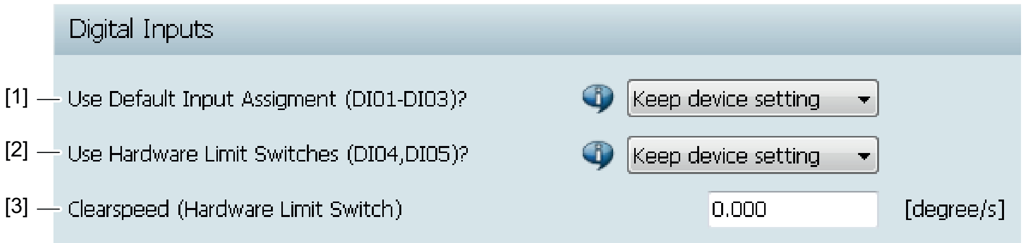
For the jerk-limited jog of the individual axes, the data about the motion profile of the axis is required. The motion parameters for the axis-by-axis jog mode *Jog axis* are configured in this window.

14113134603

No.	Description
[1]	You enter the maximum jog speed here. The jog speed can be scaled by a percentage at runtime using the fieldbus interface (0% to 100%) (jog percentage: I6^{r}:Low *Override*).
[2]	Enter the acceleration or deceleration here.

7.9.8 Digital inputs

In this window, you select how the digital inputs of the axis are to be handled. Each time that the controller starts, specific functions are automatically assigned to the digital inputs of the inverter based on the settings described below.



14113294219

No.	Description
[1]	Here, you select whether the standard input assignment (DI01 to DI03) is to be used. The digital inputs DI01 to DI03 are configured according to the following table.
[2]	Here, you select whether the hardware limit switch (DI04 to DI05) is to be used. The digital inputs DI04 to DI05 are configured according to the following table.
[3]	Here, you enter the speed (hardware limit switch) with which the device moves away from the hardware limit switch automatically in the event of the signal I1^{r}:6 *Error reset*. A speed greater than 0 must be entered, otherwise the controller or inverter cannot perform the move away automatically.

Use standard input assignment (DI01-DI03)?

Inputs	Keep device settings	Activate	Deactivate
DI01	The controller does not make any changes to the parameters in the inverter. The (manually) configured functions of the digital inputs are retained.	The parameterization of input DI01 is fixed at *Enable/rapid stop* by the controller.	The parameterization of inputs DI01 to DI03 is fixed at *No function* by the controller.
DI02		The parameterization of input DI02 is fixed at *Reset* by the controller.	
DI03		The parameterization of input DI03 is fixed as *Reference cam* by the controller.	

Use hardware limit switch (DI04, DI05)?

Inputs	Keep device settings	Activate	Deactivate
DI04	The controller does not make any changes to the parameters in the inverter. The (manually) configured functions of the digital inputs are retained.	The parameterization of input DI04 is fixed as */Limit switch positive* by the controller.	The parameterization of inputs DI04 to DI05 is fixed at *No function* by the controller.
DI05		The parameterization of input DI05 is fixed as */Limit switch negative* by the controller.	

The "/Limit switch" stands for:

- Signal 0 → Limit switch contacted
- Signal 1 → Limit switch not approved

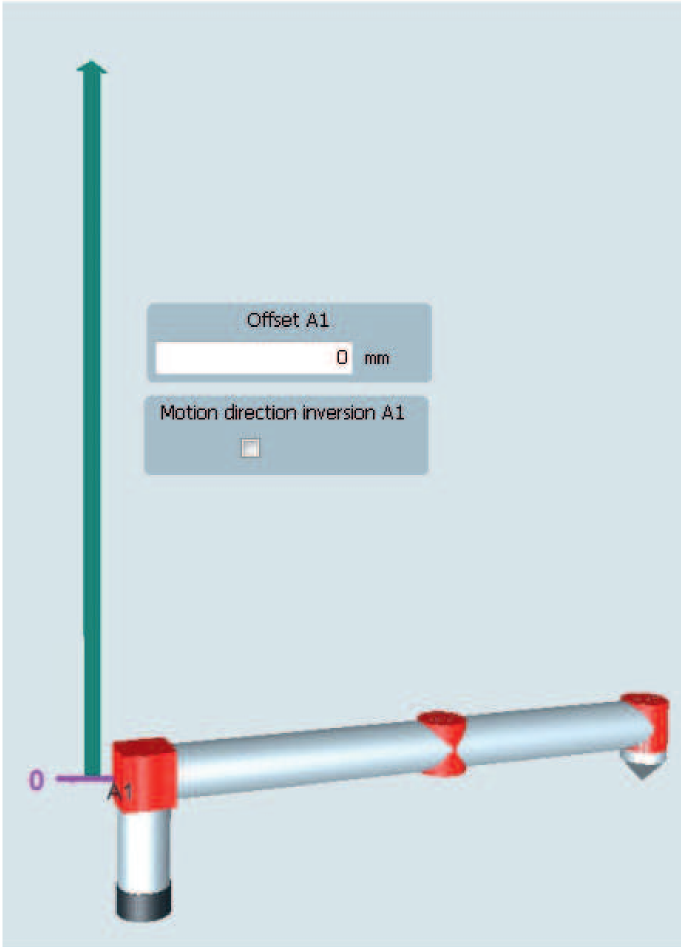
7.10 Configuration of the kinematic model

7.10.1 Kinematic model

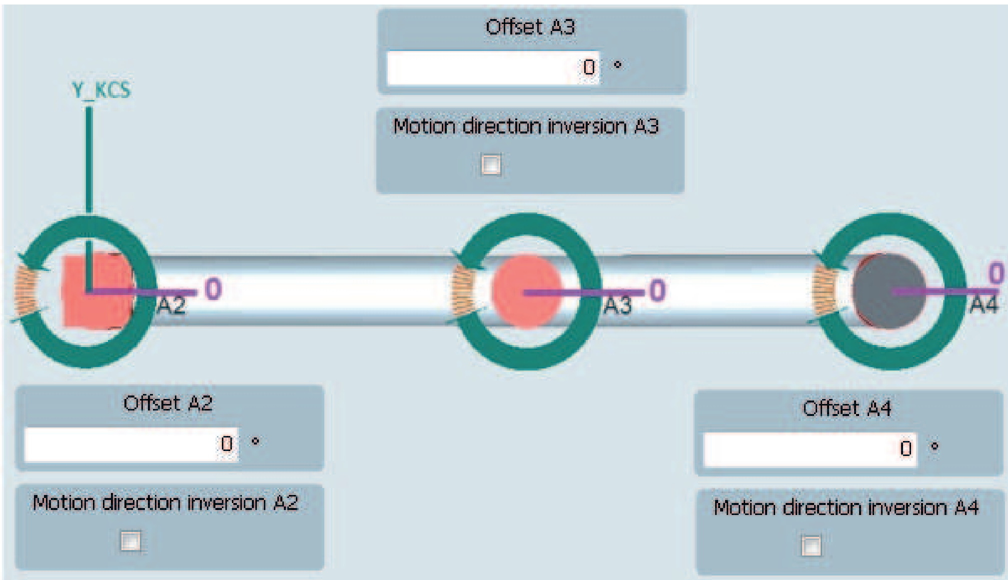
NOTICE

The kinematic model (displayed in the 3D simulation) must match the real robot. Otherwise, it is not possible to operate the kinematic control correctly.

24796131/EN – 12/2017



14121470347



9007213376867851

For the kinematic transformation to work correctly, you must adapt the kinematic model used in the controller to the real axes. The kinematic model is based on the zero points and directions of movement of the axes displayed in the figure. If the real axes that have been put into operation deviate from them, you must configure an offset that does not equal 0 or the direction reversal here:

- **Offset**

  If the kinematic zero point is different from the axis zero, enter an offset. The offset specifies where the kinematic zero point (see figures) is located in relation to the real axis zero in the real direction of movement.

  Real zero point + offset along the real direction of movement = kinematic zero point

  Or more simply: Wherever "0" is currently in purple would in theory be where the offset value set by you is.

- **Direction reversal**

  If the positive rotation of the model axis is different from the positive direction of movement of the real axis (increase in the motor increments), activate the direction of rotation reversal. When it is activated, the direction of the green arrow in the 3D simulation is reversed,

  This setting is not used to reverse the real axis direction of rotation. Instead, it is used only to adapt the model direction of rotation to the real axis direction of rotation. If you want to reverse the real axis direction of rotation, activate the direction of rotation reversal of the relevant inverter. Then, adapt the direction reversal of the HandlingKinematics configuration to the real direction of rotation.

NOTICE

After downloading the configuration, check that the setting is correct using the 3D simulation.

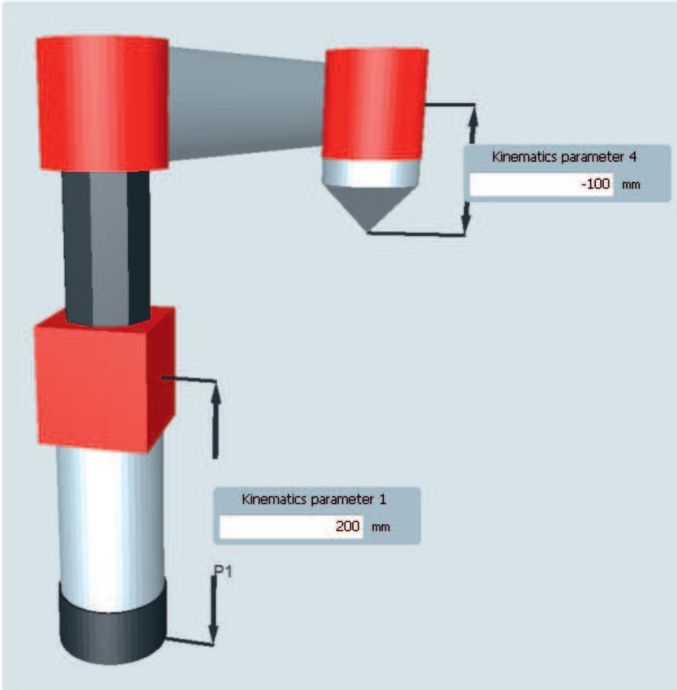
- To do this, activate the axis arrow in the control window of the 3D simulation under `[Objects] > [Display] > [Workspace] > [Axis n Workspace]`.
- For each axis, check whether the positive direction of movement (increasing motor increments) and the 0 position (0 motor increments = reference point) matches the real robot.

7.10.2 Kinematic parameters

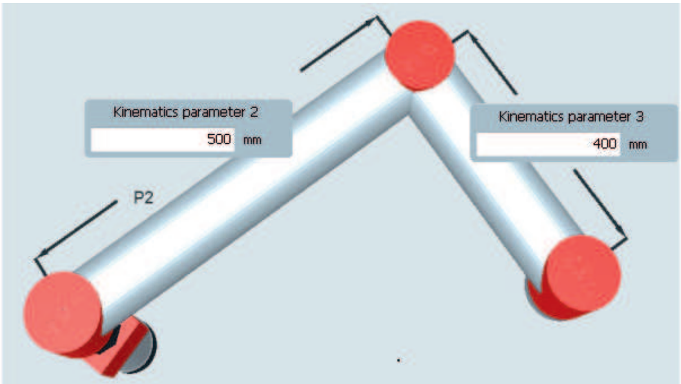
The kinematic parameters specify the geometry of the kinematics more precisely. They are transferred and displayed for each kinematic model in the 3D model. Among others, you can make the following settings:

- Distance/length
  - Between 2 joints
  - Between joint and a reference point (for instance, the origin of a coordinate system)
- Radius/diameter
- Angle
- Coupling factors between 2 axes
- Winding parameters (belt thickness, effective winding radius with the reference point)

24796131/EN – 12/2017



14122135819



14123061131

7.10.3 Kinematic limitations

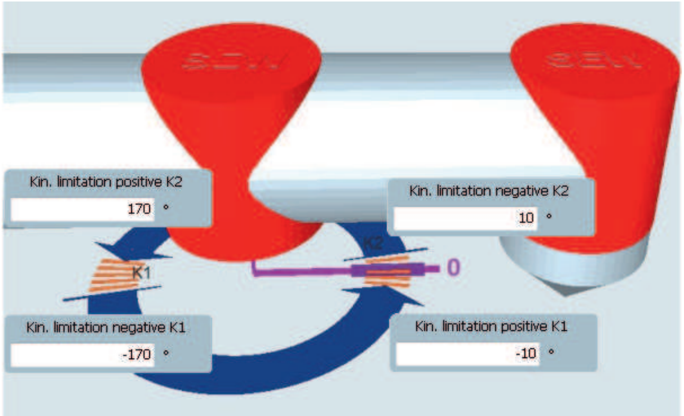
You can use the kinematic software limit switches to configure limitations including:

- Valid angle area of joints that are not directly assigned to a drive.
- Permitted distance from 2 linear drives that run on the same guide.

They are dependent on the kinematic model and displayed accordingly on the graph.

NOTICE

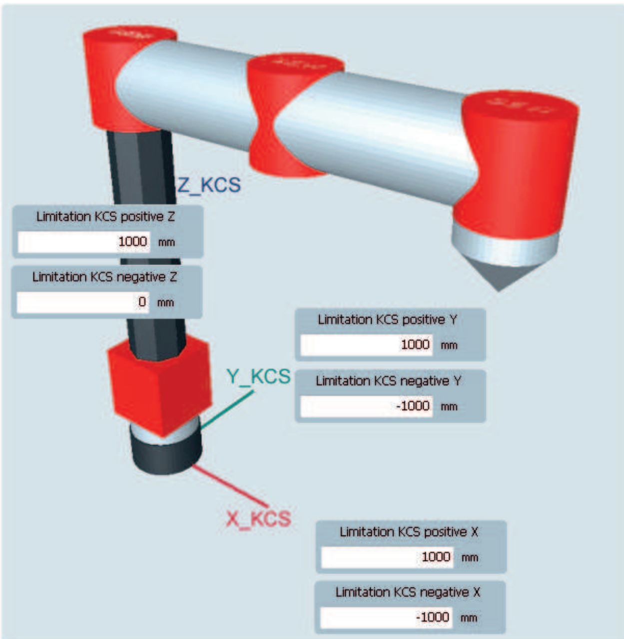
After downloading the configuration, check that the setting is correct using the 3D simulation. The blue arrow of the kinematic limitations are activated in the 3D control window under [Objects] > [Display] > [Workspace].



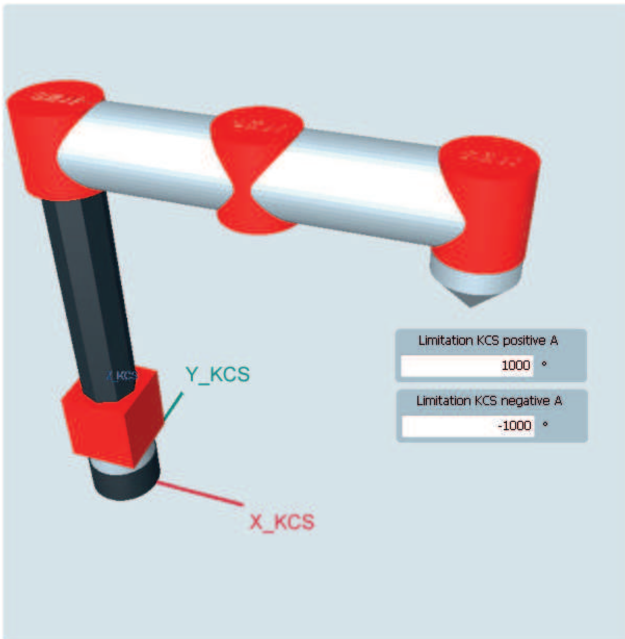
14123068939

7.10.4 Cartesian software limit switches

You can configure the Cartesian software limit switches here. They have an effect in each operating mode and limit the tool space coordinates in the kinematics coordinate system (KCS)[1] for the translational degrees of freedom X, Y, and Z and the rotary degree of freedom A. The individual axes are not limited here; instead, the 3D space coordinates of the tool are limited.



14123098635



14123104907

1) KCS = kinematic coordinate system.

24796131/EN – 12/2017

7.11 Configuring the motion parameters

7.11.1 Cartesian motion parameters

Motion parameters in the Cartesian jog jog mode

For the jerk-limited jog of the Cartesian degrees of freedom (*Cartesian jog*), data about their motion profiles is required, see the "Jog mode: Axis jog/Cartesian jog" (→ 🗎 44) chapter."" (→ 🗎 44) For this, the maximum jog speed and jog acceleration or jog deceleration must be set. The speed can be scaled as a percentage (0% to 100%) from the fieldbus interface using the signal I6ʳ:Low *Override* at runtime.

For Cartesian degrees of freedom X/Y/Z/A, the 2 parameters must be configured in each case:

- *Jog speed at 100%*
- *Jog acceleration*

Motion parameters for the Cartesian rapid stop

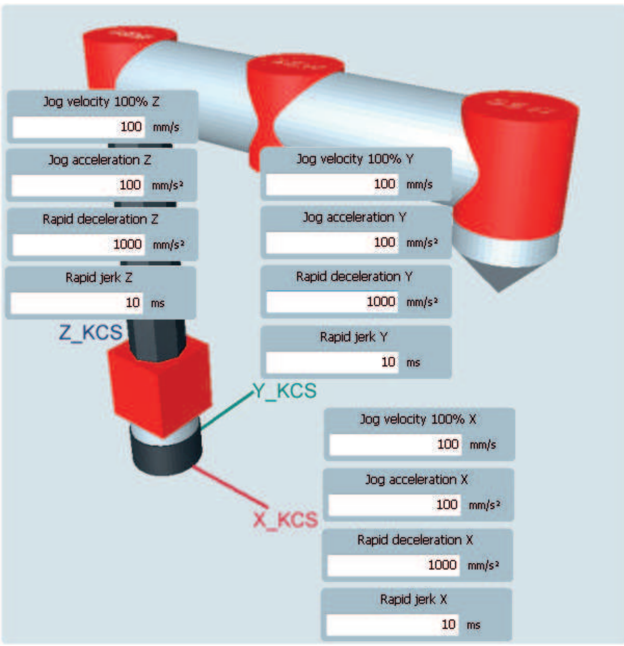
The Cartesian brake ramps must be configured for the rapid stop. You can view the procedure in the "Rapid stop" (→ 🗎 33) and "Rapid stop ramps in program mode" (→ 🗎 54) chapters.

For Cartesian degrees of freedom X/Y/Z/A, the 2 following parameters must be configured in each case:

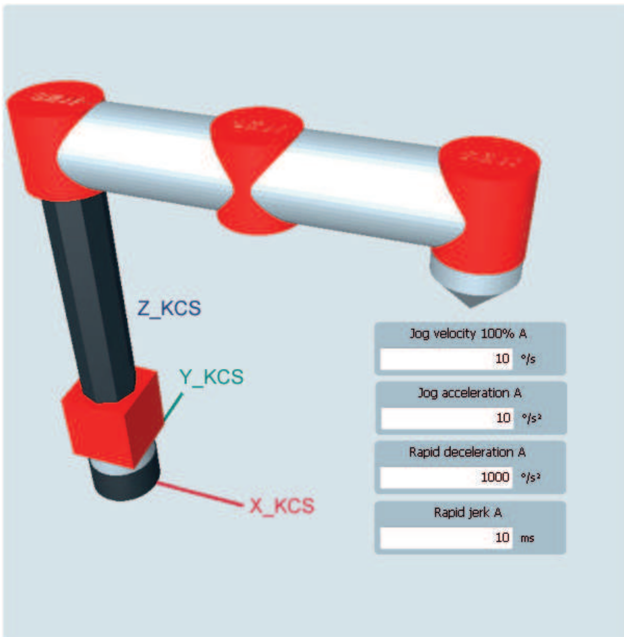
- **Rapid stop deceleration**

  The deceleration with which the Cartesian degree of freedom is to be decelerated.
- **Rapid stop jerk**

  The time in which the Cartesian degree of freedom is fully decelerated. This is used for jerk limitation (that is, to prevent impacts and damage to the mechanics).



14123193227



14123332235

24796131/EN – 12/2017

7.11.2 Continuous path motion parameters

Motion parameters for rapid stop on the path

The motion parameters must be defined for the rapid stop on the path. You can find the procedure in the "Rapid stop" (→ 🗎 33) and "Rapid stop ramps in program mode" (→ 🗎 54) chapters.

- **Rapid stop deceleration**

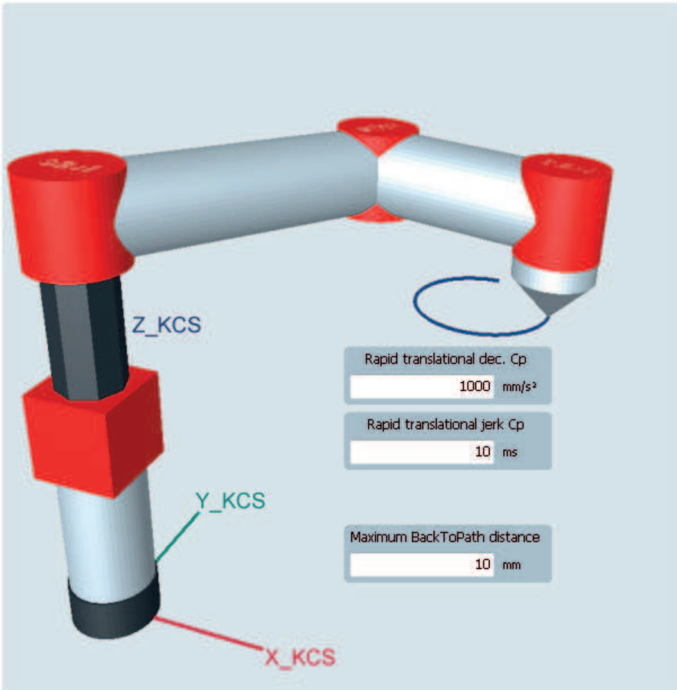
  The deceleration with which the path is to be decelerated.

- **Rapid stop jerk**

  The time in which full deceleration on the path is established. This is used for jerk limitation (that is, to prevent impacts and damage to the mechanics).

- **Maximum BackToPath distance**

  After an interruption, if the distance to the *CP*[1] path is smaller than the set maximum permitted distance to the path, repositioning to the path can be performed. When the distance from the path is too large, an error is issued and it is not possible to reposition and continue the program.



14123336715

1) CP = continuous path.

7.11.3 Overview of segment parameter sets

The segment parameter sets are data sets for describing the path segments more precisely.

The number of the segment parameter set for a specific path segment is predefined using the fieldbus interface. When the program is started, a data set is assigned to each path segment using the fieldbus interface. In this case, the same data set can also be used for multiple or all path segments. Depending on the application, it may be sufficient to configure segment parameter set 1 only and assign it to all path segments in the fieldbus interface.

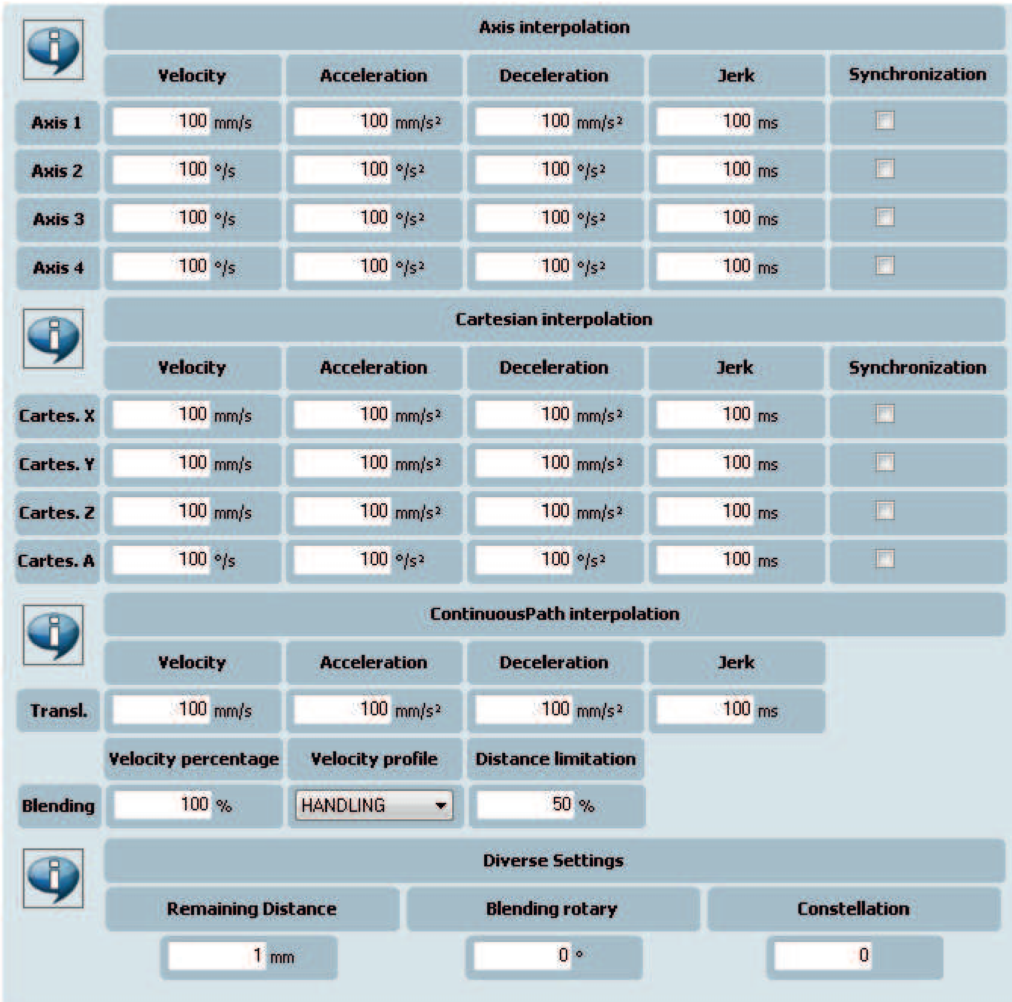
24796131/EN – 12/2017

This window provides an overview of the segment parameter sets, of which up to 8 can be parameterized.

14124696331

In all of the following screen shots, the maximum configuration of the parameters to be configured is displayed. If the kinematic model has fewer axes or fewer Cartesian degrees of freedom, only the relevant fields are displayed.

Click the startup icon next to the data set designation to go to the configuration of the relevant segment parameter set.



14123407371

After configuring a segment parameter set, you return to the overview page. Fully configured segment parameter sets are indicated with a green check mark. If you choose [Next] to go the next configuration page before all of the data sets are fully configured, the default values of the remaining segment parameter sets are retained.

In the HandlingKinematics application module, you can access 4 programs that can be selected using the fieldbus interface, see the "Programs in program mode" (→ 🗎 51) chapter. Some parameters are required by default for each program. A list of which parameters must be configured for which program is provided below. Some settings are used in multiple programs.

7.11.4 Segment parameter sets for program 1 and BackToPath

	Axis interpolation				
	Velocity	Acceleration	Deceleration	Jerk	Synchronization
Axis 1	100 mm/s	100 mm/s²	100 mm/s²	100 ms	☐
Axis 2	100 °/s	100 °/s²	100 °/s²	100 ms	☐
Axis 3	100 °/s	100 °/s²	100 °/s²	100 ms	☐
Axis 4	100 °/s	100 °/s²	100 °/s²	100 ms	☐

14123412235

For *Program 1: TARGET AXIS*, the axis interpolation is used. For this, configure the following parameters for axes 1 to 4:

- **Motion profile**, see the "Motion profile" (→ 🗎 33) chapter.
  - *Speed*
  - *Acceleration*
  - *Deceleration*
  - *Jerk (time)*
- **Synchronization**

  All axes for which the check box under "Synchronization" is activated are synchronized. All of the others are then traveled in coordination. You can view the difference in the "Difference between coordination and synchronization" (→ 🗎 34) chapter.

Various settings

The following parameters under "Diverse settings" are important:

Diverse Settings		
Remaining Distance	Blending rotary	Constellation
1 mm	0 °	0

14123454731

24796131/EN – 12/2017

- **Rotary blending**

  This parameter is visible and relevant only for kinematic models that support degree of freedom A. It specifies the angle from which blending to the next path segment is carried out.

  INFORMATION: When the rotary blending = 0°, the precise target orientation of the previous path segment must be reached so that blending to the next path can be performed.

- **Constellation**

  This parameter is visible and relevant only if the selected kinematic model has multiple constellations. It specifies the constellation with which the position is to be traveled to:

  – Constellation = 0 means that the current constellation is retained.

  – Constellation = 1, 2, and so on, means that the target point is traveled to using an explicit constellation. The significance of the different constellations depends on the selected kinematic model; see the "Constellation" (→ 🗎 28) chapter.

7.11.5 Segment parameter sets for program 2

	Cartesian interpolation				
	Velocity	Acceleration	Deceleration	Jerk	Synchronization
Cartes. X	100 mm/s	100 mm/s²	100 mm/s²	100 ms	☐
Cartes. Y	100 mm/s	100 mm/s²	100 mm/s²	100 ms	☐
Cartes. Z	100 mm/s	100 mm/s²	100 mm/s²	100 ms	☐
Cartes. A	100 °/s	100 °/s²	100 °/s²	100 ms	☐

14123460875

For *program 2: TARGET CART* and the (program-independent) *Touchprobe Motion*, the Cartesian interpolation is used. For this, configure the following parameters for Cartesian degrees of freedom X, Y, Z, and A:

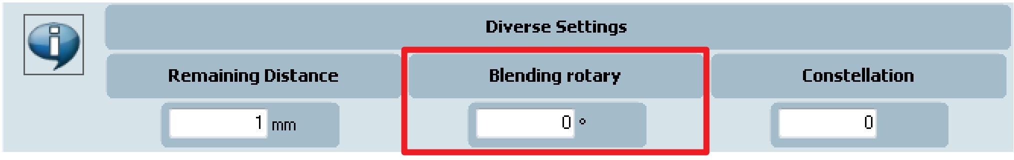
- **Motion profile**, see the "Motion profile" (→ 🗎 33) chapter.
  – *Speed*
  – *Acceleration*
  – *Deceleration*
  – *Jerk (time)*
- **Synchronization**

  All Cartesian degrees of freedom for which the check box under "Synchronization" is activated are synchronized. All of the others are then traveled in coordination. You can view the difference in the "Difference between coordination and synchronization" (→ 🗎 34) chapter.

24796131/EN – 12/2017

Various settings

The following parameters under "Diverse settings" are important:



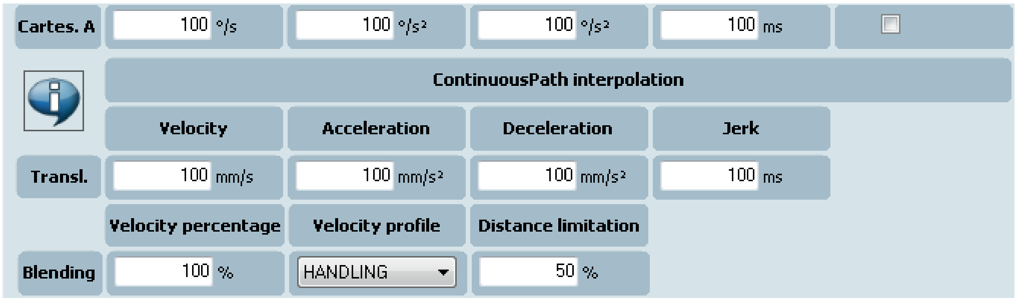
14123464971

- **Rotary blending**

  This parameter is visible and relevant only for kinematic models that support degree of freedom A. It specifies the angle from which blending to the next path segment is carried out.

  INFORMATION: When the rotary blending = 0°, the precise target orientation of the previous path segment must be reached so that blending to the next path can be performed.

7.11.6 Segment parameter sets for program 3 and 4:



14123483403

For *program 3: LINEAR coordinated* and *Program 4*: *LINEAR synchronized*, the path interpolation (CP) is used. For this, configure the following parameters for the translational path in the space:

- **Motion profile**, see the "Motion profile" (→ 🗎 33) chapter.
  - *Speed*
  - *Acceleration*
  - *Deceleration*
  - *Jerk (time)*
- **Parameters for blending**, see the "Blending" (→ 🗎 31) chapter.
  - *Speed percentage in the blending range*

    Percentage scaling of the speed resulting from the speed profile (1 to 100%).
  - *Distance limitation*

    When blending from path segment to the next, the blending distance can be predefined using the fieldbus interface. However, this is also limited to a percentage of the segment length by the blending distance limitation. The segment length must always be smaller than or equal to 50% so that the transmitted linear segment is always traveled on (no path fidelity in the blending curve!).
  - *Speed profile in the blending range*

    Here, you define the maximum path speed that is permitted to be reached in the blending curve. A circular arc is used for blending for speed profiles 1 to 6. In contrast, for speed profile 0 HANDLING, a jerk-limited profile is used.

24796131/EN – 12/2017

No.	Speed profile	Description
0	HANDLING	Preferred blending for handling. In contrast to all of the other settings, the blending is not circular for HANDLING, but is with a jerk-limited path. The speed is slowed down in a similar way as for the setting CENTRIFUGAL.
1	LOWER	The lower of the path speeds for the subsequent path segments.
2	UPPER	The higher of the path speeds for the subsequent path segments.
3	PRECEDENT	Path speed of the previous path segment.
4	SUBSEQUENT	Path speed for the path segment to which blending is performed.
5	AVERAGE	Average of the path speeds for the subsequent path segments.
6	CENTRIFUGAL	Limitation of the centrifugal acceleration that affects the workpiece. The speed in the blending curve is slowed down so that the centrifugal acceleration that affects the workpiece is not greater than the minimum deceleration of the previous segment and acceleration of the segment to which blending is performed. In addition, the path speed in the blending curve is limited to the average value of the path speeds of the subsequent path segments.

For Cartesian degree of freedom A (rotation), you must also configure the parameters of rotation A for the Cartesian interpolation:

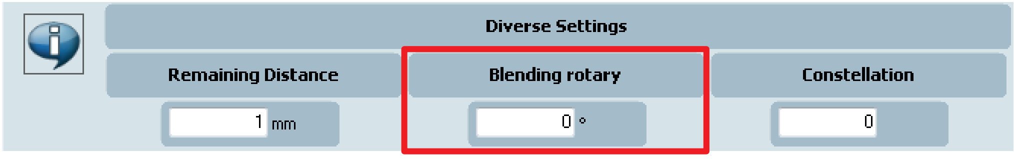
- **Motion profile**, see the "Motion profile" (→ 🗎 33) chapter.
  - *Speed*
  - *Acceleration*
  - *Deceleration*
  - *Jerk (time)*

  The synchronization bit of the Cartesian interpolation of A is not, however, important. It is already defined by the differentiation between *programs 3* and *4*.

24796131/EN – 12/2017

Various settings

Under "Diverse settings," the following parameters are important only for *Program 3: LINEAR coordinated* and NOT for *Program 4*:



14123464971

- **Rotary blending**

  This parameter is visible and relevant only for kinematic models that support degree of freedom A. It specifies the angle from which blending to the next path segment is carried out.

  INFORMATION: When the rotary blending = 0°, the precise target orientation of the previous path segment must be reached so that blending to the next path can be performed.

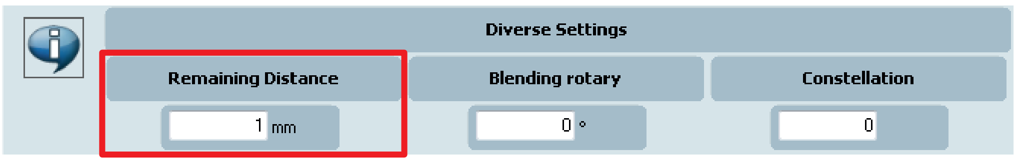
7.11.7 Segment parameter sets for Touchprobe Motion

For the *Touchprobe Motion* sensor-based positioning (program-independent), the Cartesian interpolation is used. For this, configure the following parameters for Cartesian degrees of freedom X, Y, Z, and A:

- **Motion profile**, see the "Motion profile" (→ 🗎 33) chapter.
  - *Speed*
  - *Acceleration*
  - *Deceleration*
  - *Jerk (time)*

Various settings

The following parameters under "Diverse settings" are important:



14123487499

- **Remaining travel**

  Here, you enter the remaining travel for the sensor-based positioning. It can be scaled as a [%] at runtime using the fieldbus interface (I6ʳ: High *multiplied*). In this case, the value set here equals 100%.

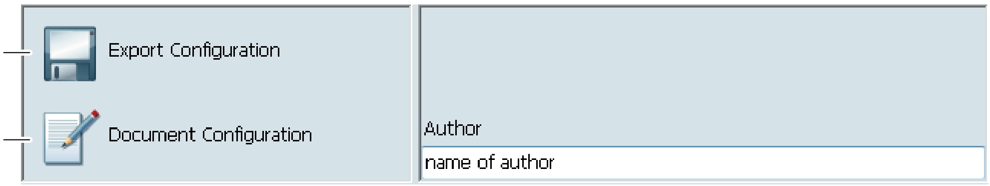
  Example: The unit for translational sizes is [mm]. The *Remaining travel* parameter in the segment parameter set is 20 mm. The percentage value in the signal I6ʳ:High *Remaining travel [%]* is 1. This means that the effective remaining travel is 0.2 mm.

24796131/EN – 12/2017

7.12 Finishing the kinematic configuration

In this window, you complete the kinematic configuration.

[1]
[2]



14123548683

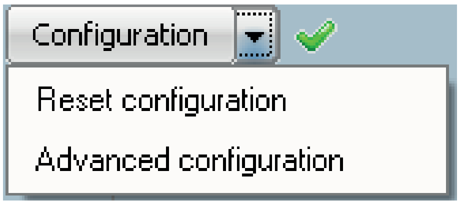
No.	Description
[1]	Use this button to export the configuration of the kinematic module. The values can then be used again in a different project. Here, only the configuration of the kinematic module is saved. This means that other individual axes that you have configured (e.g. the universal module) are not saved.
[2]	Use this button to create a report for the configuration as a PDF file that documents all of the settings.

Press the [Finish] button to leave the configuration of the kinematic module and return to the Application Configurator.

7.13 Advanced configuration or higher authorization level

Additional settings can be configured for the kinematics; however, their standard values must be suitable for the majority of application cases. The settings must be made only by trained personnel or by SEW-EURODRIVE. These additional settings are configured in the advanced configuration.

To go to the advanced configuration, select the advanced configuration in the "Configuration" drop-down list in the Application Configurator.



14123544587

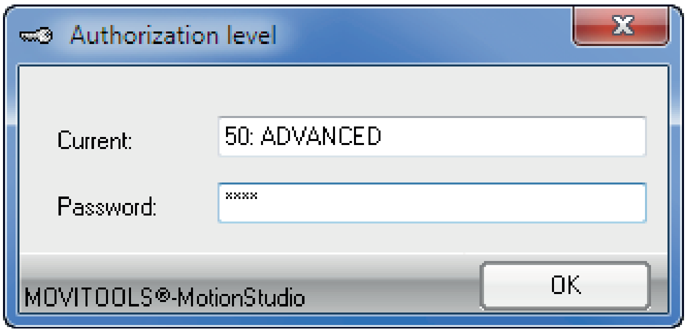
The advanced configuration is opened automatically if the MOVITOOLS® MotionStudio is in the "Advanced" authorization level or higher.

24796131/EN – 12/2017

Changing the authorization level in the MOVITOOLS® MotionStudio

Proceed as follows:

1. In the MOVITOOLS® MotionStudio, choose the [Settings] > [Authorization levels] menu command.
   ⇨ A window for entering the password for the required authorization level opens.



14123555851

2. In the second line, enter "5249."
3. Press [OK].
4. Restart the Application Configurator.

⇨ The authorization level is changed.

7.14 Adding application modules or functions

Up to 8 additional axes can be controlled using single-axis modules. For all of the application modules and functions, documentation is available in which you can read their purpose of use and operating principles.

7.15 Downloading the whole configuration

NOTICE

Beware of downloading the software/configuration to the controller when the plant is in operation.

Injury and damage to property.

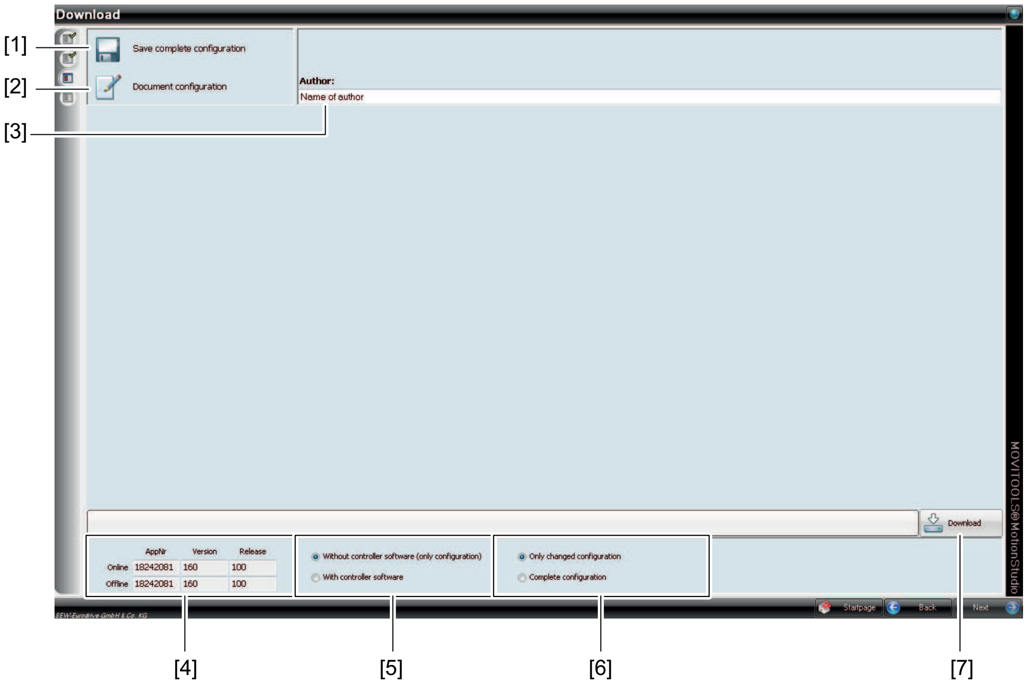
- Bring the system into a safe condition.

INFORMATION

For detailed information about the functions and operation of the Application Configurator and the other application modules, see the corresponding documentation, which is available under "Documentation" in the controller context menu in the MOVITOOLS® MotionStudio and elsewhere.

24796131/EN – 12/2017

To complete the configuration, click the [Next] button in the Application Configurator. If you have configured the "HandlingKinematics" application module, you can find the following information under "Download."



14130069259

No.	Description
[1]	Use this button to save the configurations to a file in the format *.AppConfig.ZIP. The values can then be used again during further startups.
[2]	Use this button to create a configuration report as a PDF file.
[3]	If you enter a name in this edit box, the name is displayed for identification in the report.
[4]	This group displays the information about the boot project installed online and offline (see the following table): • Application module part number • Version currently installed and downloaded • Release currently installed and downloaded
[5]	Use this radio button to select whether you want to download the configuration with or without controller software.
[6]	Use this radio button to select whether you want to download the changed or complete configuration.
[7]	Use this button to download the configuration.

1. Make the required settings.
2. To download, press the button [7].

⇨ The configuration is downloaded.

⇨ The controller is ready for operation.

⇨ You return to the configuration interface of the Application Configurator by choosing [Next].

Recommended download variant dependent on [4] (version/release and app no.)

App no.	Same version and same release	Older version and/or older release online	Newer version and/or newer release online
Same app no.	Download **without** controller software recommended.	Download **with** controller software recommended. As a result, new functions are enabled on the one hand and any errors are resolved on the other.	Update your MOVITOOLS® MotionStudio.
Other app no.	Download with controller software required. Otherwise, the configuration is not supported.		

8 Control via fieldbus

After you download the Application Configurator configuration, the controller is ready for operation, see the "Configuration" (→ 🗎 66) chapter. It can be controlled using the fieldbus of the PLC. To do this, proceed as follows:

1. Check that the HandlingKinematics controller are configured correctly using the diagnostics window and 3D simulation; see the "Module diagnostics with HandlingKinematicsMonitor" (→ 🗎 119) and "3D simulation" (→ 🗎 132) chapters.
2. Wire the controller to the PLC (fieldbus).
3. Familiarize yourself with the HandlingKinematics process data using the process data monitor (PD monitor), see the "PD monitor" (→ 🗎 130), "Segment parameter sets" (→ 🗎 34), and "Process data assignment" (→ 🗎 135) chapters.
4. Create the selected process data profile in the PLC; see the "Creating a process data profile in the PLC" (→ 🗎 106) chapter.
5. Fill in the process data interface with the required data:
   - ⇨ Program startup, see the "Startup sequence" (→ 🗎 110) chapter.
   - ⇨ Process sequence, see chapter "Process sequence" (→ 🗎 112).
   - ⇨ Errors, see the "Handling errors" (→ 🗎 114) chapter.

8.1 Important status signals

8.1.1 Cartesian position valid

The following conditions must be fulfilled for the signal O26^{r}:4 *Cartesian position valid* to be TRUE:

- O1^{r}:2 *All drives referenced = TRUE*
- If O2^{r}:Low-Byte *Operating mode* ≠ Operating mode *Default*, the following must also apply:
  - I1^{r}:0 *Controller inhibit = FALSE*
  - I1^{r}:1 *Enable/rapid stop = TRUE*
  - O1^{r}:10 *All output stages enabled = TRUE*

If these conditions are met and the signal O26^{r}:4 *Cartesian position valid* is still not TRUE, the cause is a current fault (O1^{r}:7 *Application fault = TRUE*). Refer to chapter "Handling errors" (→ 🗎 114) for further procedures.

8.1.2 Application error

If there are application errors (O1^{r}:7 *Application error = TRUE*), correct the cause of error. The error messages displayed in the MessageHandler or HandlingKinematicsMonitor can be used to determine the cause of error.

Correct the error status by increasing the edge of signal I1^{r}:6 *Error reset*.

8.1.3 Application ready

The following conditions must be fulfilled for the signal O1^{r}:3 *Application ready* to be *TRUE*:

- SRL is initialized.
- O1^{r}:7 *Application fault = FALSE*

- One of the following conditions:
  - I2^{r}:2 or O2^{r}:2 *Referencing* = *TRUE*
  - I2^{r}:0/1/3/4 and O2^{r}:0/1/3/4 *operating mode: Axis-by-axis jog / Cartesian jog / Program auto / Program step = TRUE*
    - → I1^{r}:0 *Controller inhibit = FALSE*
    - → I1^{r}:1 *Enable/rapid stop = TRUE*
    - → O1^{r}:0 *At least one motor is turning= FALSE* (**only for changing to** O1^{r}:3 *Application ready*!)
    - → O1^{r}:2 *All drives referenced = TRUE* (**except for axis-by-axis jog!**)
    - → O1^{r}:10 *All output stages enabled = TRUE*
    - → O20^{r} – 23^{r}:5 *FI connected = TRUE*

8.2 Requirements for movement of the kinematic model

8.2.1 General requirements for movement

- O1^{r}:3 *Application ready*
- I1^{r}:2 *Feed enable* (except for in referencing mode)
- I6^{r}:Low *Override* > 0 (except for in referencing mode)
- Operating mode-dependent requirements

8.2.2 Additional requirements for movement in referencing mode

- I2^{r}:2 *Referencing = TRUE*
- I2^{r}:8 to 11 *Start referencing axis 1 to 4* (the relevant bit) = *TRUE*

8.2.3 Additional requirements for movement in jog mode

- I2^{r}:0/1 *Cartesian/Axis-by-axis jog* (exactly one of the bits) = *TRUE*
- I5^{r}:0 to 3, I5^{r}:8 to 11 *Positive/Negative jog* (at least one of the bits) = *TRUE*

8.2.4 Additional requirements for movement in program mode

- I2^{r}:3/4 *Auto program/Program step = TRUE*
- I5^{r}:High *Program number* between 1 and 4
- Enter segments:
  - I8^{r}ff. *Target poses* (**different values!**)
  - Optional:
    → I7^{r}ff. *Assignment of the segment parameter sets to the ... segments*
    → I19^{r}ff. Blending distance to segment …
- Control program sequence; signal sequence:
  1. I4^{r}:0 *Program init* = positive edge
     → O4^{r}:0 *Program initialized = TRUE*
  2. I4^{r}:2 *Program stop = FALSE*

24796131/EN – 12/2017

3. I4^r:1 *Program start* = positive edge (then *TRUE*)
   → O4^r:1 *Program being executed* = *TRUE*
   → O4^r:1 *Program complete* = *TRUE*

8.2.5 Additional requirements for overdetermined kinematic models

Overdetermined kinematic models have multiple drives for a Cartesian degree of freedom. Among others, this affects the following kinematic models; see the "Kinematic models" (→ 🗎 19) chapter:

- CARTESIAN_GANTRY_LLLLR_XYZA_M10
- ROLLER_GANTRY_LLLLR_XYZA_M10

These kinematic models have 5 drives (A1 to A5) but have only 4 degrees of freedom of movement (X, Y, Z, and A). For these 2 kinematic models, a linear axis L is controlled by the 2 drives A1 and A2. During an initial startup, a restart of the machine, or an emergency stop among others, the 2 axes may be at different positions even though they are referenced correctly.

Therefore, for the automated kinematic model movement (*Program auto/Program step*), *Program 1: TARGET AXIS* must be used before each switch to a Cartesian operating mode (that is, *Cartesian jog* or *Program auto/Program step* with *Program 2 to 4*) in *Program auto* to move to O7^r to O10^r *Current position and orientation*. The axes are then aligned by the kinematic control so that Cartesian travel is then enabled. Otherwise, a lag error may occur.

8.3 Consistent data transmission

NOTICE

For consistent data transmission from the PLC to the controller, process data must be configured accordingly in the PLC.

8.3.1 Consistency blocks: Profile 1 with 32 PD

NOTICE

In the fieldbus master, the following consistency blocks must be selected:

- A 2 PD block (for the general controller signals)
- A 32 PD block (for the HandlingKinematics signals)

→ In this case, the sequence from the table displayed below must be maintained.

The 32 PD block must be transmitted consistently between the controller and the PLC using the fieldbus. In this profile, the *telegram number* must not be described, in contrast to other profiles.

To ensure that the 32 PD block is transmitted consistently, a 32 PD block must also be created in the fieldbus master.

Se-quence	Consist-ency blocks	PD	Description	Telegram number position
-	2 PD	-	*SEW controller* process data	-
-	Any	Up to 86 PD	Process data for other application modules	-
1	32 PD	I1^r to I32^r	*HandlingKinematics* process data	I3^r:Low (optional)
-	Any	Up to 86 PD	Process data for other application modules	-

8.3.2 Consistency blocks: Profile 2 with 60 PD

NOTICE

In the fieldbus master, the following consistency blocks must be selected:

- A 2 PD block (for the general controller signals)
- 2 x 32 PD blocks (for the HandlingKinematics signals)

→ In this case, the sequence from the table displayed below must be maintained.

A maximum of 32 process data words can be transmitted consistently. To still achieve consistency across the 60 process data words of the profile, both 32 PD blocks have *telegram numbers* that must be changed by the PLC in cycles (e.g. incremented) and assigned to both of these blocks.

NOTICE

The process input data is transmitted in the controller only when both telegram numbers match.

The telegram numbers in the first and second consistency block must match and be changed in cycles (e.g. incremented).

If the telegram numbers are different for 100 ms, an error is issued.

When the process data is changed, the telegram number must be changed in the PLC. This is achieved through a cyclical change. In the controller, it is ensured that the process input data is transferred only if the consistency blocks have arrived in the controller.

<table>
<tr><th>Se-quence</th><th>Consist-ency blocks</th><th>PD</th><th>Description</th><th>Telegram number position</th></tr>
<tr><td>-</td><td>2 PD</td><td>-</td><td>SEW controller process data</td><td>-</td></tr>
<tr><td>-</td><td>Any</td><td>Up to 54 PD</td><td>Process data for other application modules</td><td>-</td></tr>
<tr><td>1</td><td>32 PD</td><td>I1^r to I32^r</td><td rowspan="2">Process data for HandlingKinematics</td><td>I3^r:Low (optional)</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">32 PD</td><td>I33^r to I60^r</td><td rowspan="2">I33^r:Low (optional)</td></tr>
<tr><td>I61^r to I64^r</td><td>Not assigned</td></tr>
</table>

Se-quence	Consist-ency blocks	PD	Description	Telegram number position
-	Any	Up to 54 PD	Process data for other application modules	-

NOTICE

The telegram numbers must not be in the same consistency block.

The assignment of process data words to the consistency blocks must not be changed. Otherwise, you cannot ensure that the telegram number is included precisely once in each of the consistency blocks.

The sequence must correspond to the specification in the table above.

8.3.3 Consistency blocks: Profile 3 with 88 PD

NOTICE

In the fieldbus master, the following consistency blocks must be selected:

- A 2 PD block (for the general controller signals)
- 2 x 32 PD blocks and 1 x 24 PD block (for the HandlingKinematics signals)

→ In this case, the sequence from the table displayed below must be maintained.

A maximum of 32 process data words can be transmitted consistently. To still achieve consistency across the 88 process data words of the profile, both 32 PD blocks and the 24 PD block have *telegram numbers* that must be changed by the PLC in cycles (e.g. incremented) and assigned to these 3 blocks.

NOTICE

The process input data is transmitted in the controller only when the telegram numbers match.

The telegram numbers in the first, second, and third consistency block must match and be changed in cycles (e.g. incremented).

If the telegram numbers are different for 100 ms, an error is issued.

When the process data is changed, the telegram number must be changed in the PLC. This is achieved through a cyclical change. In the controller, it is ensured that the process input data is transferred only if the consistency blocks have arrived in the controller.

Se-quence	Consist-ency blocks	PD	Description	Telegram number position
-	2 PD	-	*SEW controller* process data	-
-	Any	Up to 30 PD	Process data for other application modules	-
1	32 PD	I1^r to I32^r	Process data for *HandlingKinematics*	I3^r:Low
2	24 PD	I33^r to I56^r		I33^r:Low
3	32 PD	I57^r to I88^r		I61^r:Low

Se-quence	Consist-ency blocks	PD	Description	Telegram number position
-	Any	Up to 30 PD	Process data for other application modules	-

NOTICE

The telegram numbers must not be in the same consistency block.

The assignment of process data words to the consistency blocks must not be changed. Otherwise, you cannot ensure that the telegram number is included precisely once in each of the consistency blocks.

The sequence must correspond to the specification in the table above.

8.3.4 Consistency blocks: Profile 4 with 116 PD

NOTICE

In the fieldbus master, the following consistency blocks must be selected:

- A 2 PD block (for the general controller signals)
- 2 x 32 PD blocks, 2 x 24 PD blocks, and 1 x 4 PD block (for the HandlingKinematics signals)

→ In this case, the sequence from the table displayed below must be maintained.

A maximum of 32 process data words can be transmitted consistently. To still achieve consistency across the 116 process data words of the profile, both 32 PD blocks and both 24 PD blocks have *telegram numbers* that must be changed by the PLC in cycles (e.g. incremented) and assigned to these 4 blocks. An exception is the 4 PD block, which cannot be transmitted consistently (see the following information).

NOTICE

Segment 20 cannot be transmitted consistently because the process data interface, which has a maximum size of 120 process data words, does not accommodate another 32 PD block after the last 32 PD block.

Segment 20 can be used only if there is enough wait time before the *Program start* signal is set (around 2 x fieldbus cycle time + 20 ms) after setting the values of segment 20.

NOTICE

The process input data is transmitted in the controller only when the telegram numbers match.

The telegram numbers in the first, second, third, and fourth consistency block must match and be changed in cycles (e.g. incremented).

If the telegram numbers are different for 100 ms, an error is issued.

24796131/EN – 12/2017

When the process data is changed, the telegram number must be changed in the PLC. This is achieved through a cyclical change. In the controller, it is ensured that the process input data is transferred only if the consistency blocks have arrived in the controller.

Sequence	Consistency blocks	PD	Description	Telegram number position
-	2 PD	-	*SEW controller* process data	-
-	Any	Up to 2 PD	Process data for other application modules	-
1	32 PD	I1^r to I32^r	Process data for *HandlingKinematics*	I3^r:Low
2	24 PD	I33^r to I56^r		I33^r:Low
3	32 PD	I57^r to I88^r		I61^r:Low
4	24 PD	I89^r to I112^r		I89^r:Low
5	4 PD	I113^r to I116^r		**Not possible**
-	Any	Up to 2 PD	Process data for other application modules	-

NOTICE

The telegram numbers must not be in the same consistency block.

The assignment of process data words to the consistency blocks must not be changed. Otherwise, you cannot ensure that the telegram number is included precisely once in each of the consistency blocks.

The sequence must correspond to the specification in the table above.

8.4 Sample program for SIMATIC S7

On the SEW homepage (www.sew-eurodrive.de) under `[Online Support] > [Data & documents] > [Software]`, you can find the "HandlingKinematics function block" sample program under the `[PLC, fieldbus and device description files] > [SPS Sample Projects]` software category. This is a free service that provides only a non-binding sample of the basic procedure for creating a PLC program. SEW therefore does not accept any liability for the content of the sample program.

8.4.1 Storing a process data profile in the PLC

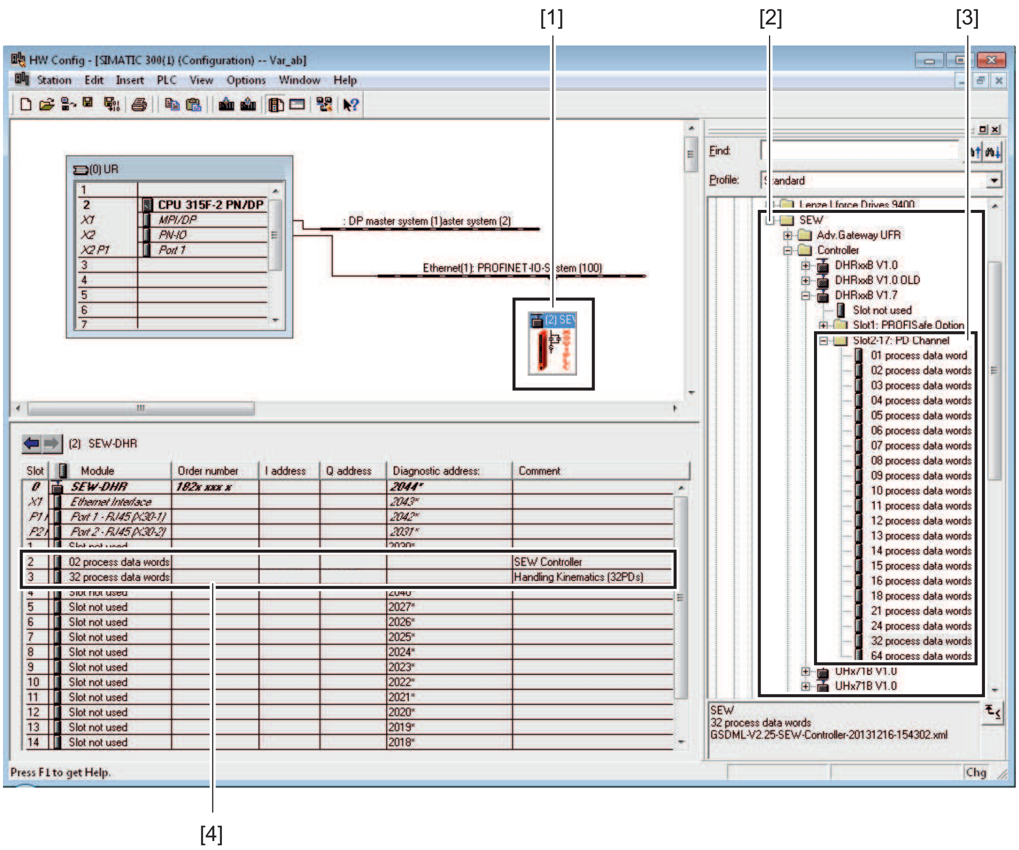
For the selected process data profile, a specific setting must be configured in the hardware configuration of the SIMATIC S7 (HW Config). It is required to ensure that the process data is transmitted consistently.

The first 2 process data words are provided for the SEW controller and must be added to the hardware configuration. You can find a description of the controller process data in the "Configuration Software – Application Configurator for CCU" manual.

8.4.2 Hardware configuration: Profile 1 with 32 PD

In profile 1, the following process data must be added to the slots displayed below:

- **02 process data words**
  - Required to map the controller process data (*SEW controller*).
- **32 process data words**
  - Entry for HandlingKinematics profile 1, see the "Profile 1 with 32 PD" (→ 🗎 37) chapter.



14183547787

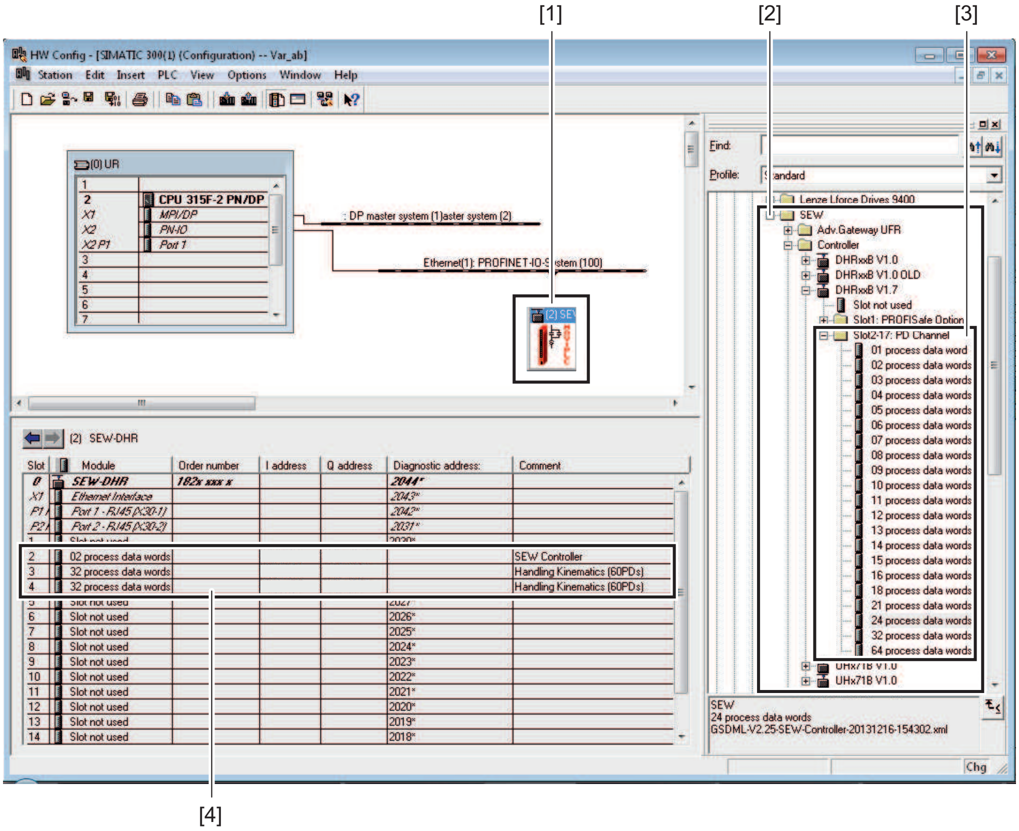
No.	Description
[1]	MOVIPLC® DHRxxB V1.7: PROFINET Slave
[2]	SEW PROFINET slave
[3]	Potential PD blocks for DHRxxB V1.7
[4]	PD blocks for profile 1

8.4.3 Hardware configuration: Profile 2 with 60 PD

In profile 2, the following process data must be added to the slots displayed below; see the "Profile 2 with 60 PD" (→ 🗎 39) chapter:

- **02 process data words**
  - Required to map the controller process data (*SEW controller*).
- **32 process data words**

  – First entry for HandlingKinematics profile 2.
- **32 process data words**
  – Second entry for HandlingKinematics profile 2.



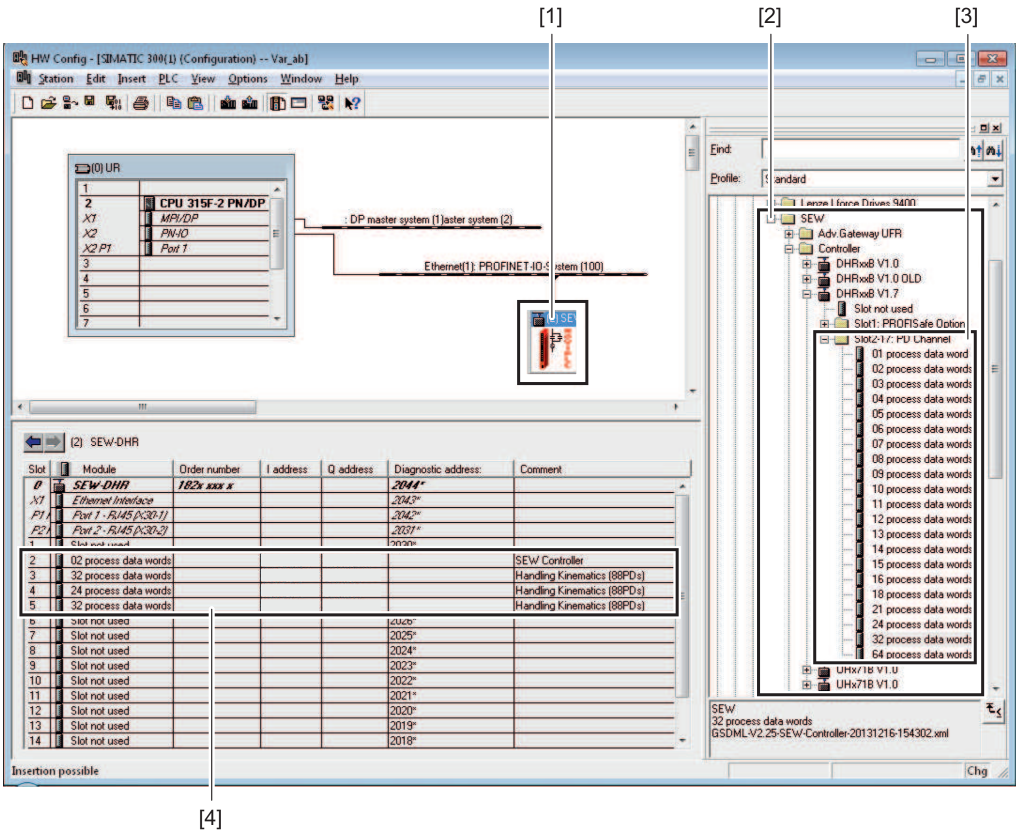
14183550219

No.	Description
[1]	MOVIPLC® DHRxxB V1.7: PROFINET Slave
[2]	SEW PROFINET slave
[3]	Potential PD blocks for DHRxxB V1.7
[4]	PD blocks for profile 2

8.4.4 Hardware configuration: Profile 3 with 88 PD

In profile 3, the following process data must be added to the slots displayed below; see the "Profile 3 with 88 PD" (→ 🗎 40) chapter:

- **02 process data words**
  – Required to map the controller process data (*SEW controller*).
- **32 process data words**
  – First entry for HandlingKinematics profile 3.
- **24 process data words**
  – Second entry for HandlingKinematics profile 3.
- **32 process data words**
  – Third entry for HandlingKinematics profile 3.



14183680651

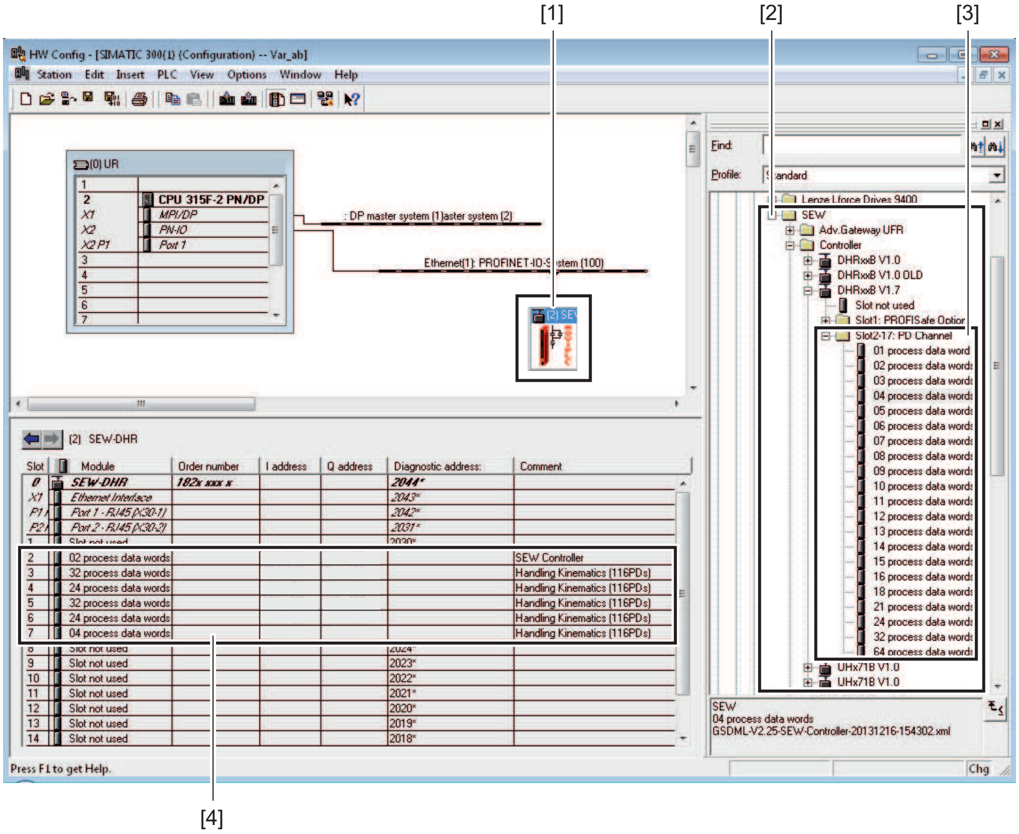
No.	Description
[1]	MOVIPLC® DHRxxB V1.7: PROFINET Slave
[2]	SEW PROFINET slave
[3]	Potential PD blocks for DHRxxB V1.7
[4]	PD blocks for profile 3

8.4.5 Hardware configuration: Profile 4 with 116 PD

In profile 4, the following process data must be added to the slots displayed below; see the "Profile 4 with 116 PD" (→ 🗎 41) chapter:

- **02 process data words**
  - Required to map the controller process data (*SEW controller*).
- **32 process data words**
  - First entry for HandlingKinematics profile 4.
- **24 process data words**
  - Second entry for HandlingKinematics profile 4.
- **32 process data words**
  - Third entry for HandlingKinematics profile 4.
- **24 process data words**
  - Fourth entry for HandlingKinematics profile 4.
- **4 process data words**

– Fifth entry for HandlingKinematics profile 4.



14183683083

No.	Description
[1]	MOVIPLC® DHRxxB V1.7: PROFINET Slave
[2]	SEW PROFINET slave
[3]	Potential PD blocks for DHRxxB V1.7
[4]	PD blocks for profile 4

NOTICE

Segment 20 cannot be transmitted consistently because the process data interface, which has a maximum size of 120 process data words, does not accommodate another 32 PD block after the last 32 PD block.

Segment 20 can be used only if there is enough wait time before the *Program start* signal is set (around 2 x fieldbus cycle time + 20 ms) after setting the values of segment 20.

8.5 Startup sequence

The program startup in the PLC after switching on the system may appear as follows. There may be other queries for the application in this case.

Action | Start the program

	↓	
Transition	Fieldbus is running (PLC ↔ controller)	
	↓	
	Enable axes	I1^r:0 *Controller inhibit = FALSE* I1^r:1 *Enable/rapid stop = TRUE*
	↓	
	Axes are enabled	O1^r:10 *All output stages enabled = TRUE?*
	↓	
	Select an operating mode	I2^r *Operating mode*
	↓	
	The selected operating mode was adopted.	O2^r *Operating mode* correct?
	HandlingKinematics reports the signal O1^r:3 *Application ready*.	O1^r:3 *Application ready = TRUE?*
	↓	
	Further program sequence ...	

8.6 Process sequence

Two alternating programs can be started as follows:

- First, the path points are transmitted with orientation, the blending is defined, and the segment parameter sets are assigned to the path segments.
- The program can then be started.
- Once it is detected that the program is complete, a new path and motion parameters can be specified and a new edge of the signal I4^r:1 *Program start* can be used for the start.
- For each program sequence, the wait and end signals can also be used or the *Touchprobe Motion* and *Touchprobe Measure* Touchprobe functions can also be activated.

The following flow diagram shows how the continuous execution of programs can be implemented. Before the displayed sequence can be executed, the requirements for a movement in the *Program auto* program mode must be fulfilled, see the "Requirements for movement of the kinematic model" (→ 🗎 101) chapter.

The "BackToPath" (→ 🗎 58) chapter describes how an interruption to the program sequence can be responded to.

Preparing the "Pick" program	
• Describe the path segments:	For example, I8^r to I11^r *Target pose for segment 1* (XYZA)
– Pose + blending	For example, I17^r *Blending distance to segment 2*
• Assign segment parameter sets to the path segments	For example, I7^r *Assignment of the segment parameter sets to the first 5 segments*
• Optional:	
– Set the wait signal, end signal	For example, I4^r:8 *Wait 1*, I4^r:15 *End 4*
– Activate the *Touchprobe Measure*	I4^r:6 *Touchprobe Measure*
– Activate the *Touchprobe Motion* and define the *Remaining travel*	I4^r:6 *Touchprobe Motion* and I6^r:High *TpRemaining DistPerc*
Set the program start	I4^r:1 *Program start = TRUE*
↓	
The program is being executed	O4^r:1 *Program being executed = TRUE*?
↓	
Reset the program start	I4^r:1 *Program start = FALSE*
↓	
Program fully executed	O4^r:3 *Program finished = TRUE*?
↓	
If *Touchprobe Measure* is used:	
• Read and save the measured position	O7^r – O9^r *Touchprobe Measure → X, Y, Z*
Preparing the "Place" program	
• Describe the path segments:	For example, I8^r to I11^r *Target pose for segment 1* (XYZA)
– Pose + blending	For example, I17^r *Blending distance to segment 2*
• Assign segment parameter sets to the path segments	For example, I7^r *Assignment of the segment parameter sets to the first 5 segments*

• Optional:	
– Set the wait signal, end signal	For example, I4^{r}:8 *Wait 1*, I4^{r}:15 *End 4*
– Activate the *Touchprobe Measure*	I4^{r}:6 *Touchprobe Measure*
– Activate the *Touchprobe Motion* and define the *Remaining travel*	I4^{r}:6 *Touchprobe Motion* and I6^{r}:High *TpRemaining DistPerc*
Set the program start	I4^{r}:1 *Program start = TRUE*
↓	
The program is being executed	O4^{r}:1 *Program being executed = TRUE*?
↓	
Reset the program start	I4^{r}:1 *Program start = FALSE*
↓	
Program fully executed	O4^{r}:3 *Program finished = TRUE*?

8.7 Handling errors

You can detect whether the HandlingKinematics application module has an error status from the signal O1^{r}:7 *Application error*. In this case, the error may be triggered by the controller or the assigned kinematics inverters may report errors. You can detect whether there is an inverter error from the signal O1^{r}:5 *At least one FI error*. The inverter that is affected can be viewed in the area O20^{r}ff. *Bit 6 FI error*. In the event of an error, the signal O1^{r}:3 *Application ready* is set to *FALSE*.

If the application module has an error status, deceleration is performed automatically using the configured rapid stop ramps. It can proceed further only once the cause of error is corrected and the error is reset. The error is reset using the signal I1^{r}:6 *Error reset*.

To find the cause of an error, a unique error number is issued in the signal O5^{r}-O6^{r} *ErrorID*. You can find a list of the error numbers attached. In the event of an inverter error, you can find more information about the error number in the documentation of the relevant device.

INFORMATION

For an overview display of all pending and archived error, warning, and notice messages, we recommend the MessageHandler, which can be started under "Diagnostics" in the controller context menu; see the "MessageHandler" (→ 🗎 134) chapter.

Current pending errors and messages are displayed in the MessageHandler. Past errors are also displayed there (in the archive) with additional information such as the "time stamp".

The MessageHandler can be used from the HandlingKinematics module diagnostics (see the "Tab: Messages" (→ 🗎 129) chapter) or as a standalone plug-in. You can start the plug-in in the controller context menu under [Diagnostics] > [MessageHandler].

24796131/EN – 12/2017

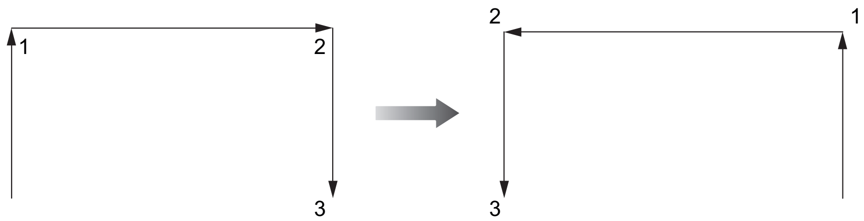
8.8 Notes on the layout of the path

- If a path segment is pointing in the opposite direction to the previous one, a wait point or the end of program must be in front.

Incorrect solution: No program end or wait point after the third path segment.

14320329483

Correct solution: Program end or wait point after the third path segment.



14320331915

- If you do not want blending, select *Blending distance* = 0 AND add a wait point there; see the information section in the "Wait signal" (→ 🗎 56) chapter. Without the wait point, the path point is generally not reached precisely.

9 Diagnostics

You can access the following Application Configurator functions to test the application module functions and diagnose errors:

- Diagnostics monitor: HandlingKinematics monitor
- 3D simulation
- Process data monitor: PD monitor
- Trace
- Extended configuration

The listed functions are opened by clicking button [1] on the Application Configurator initial screen.

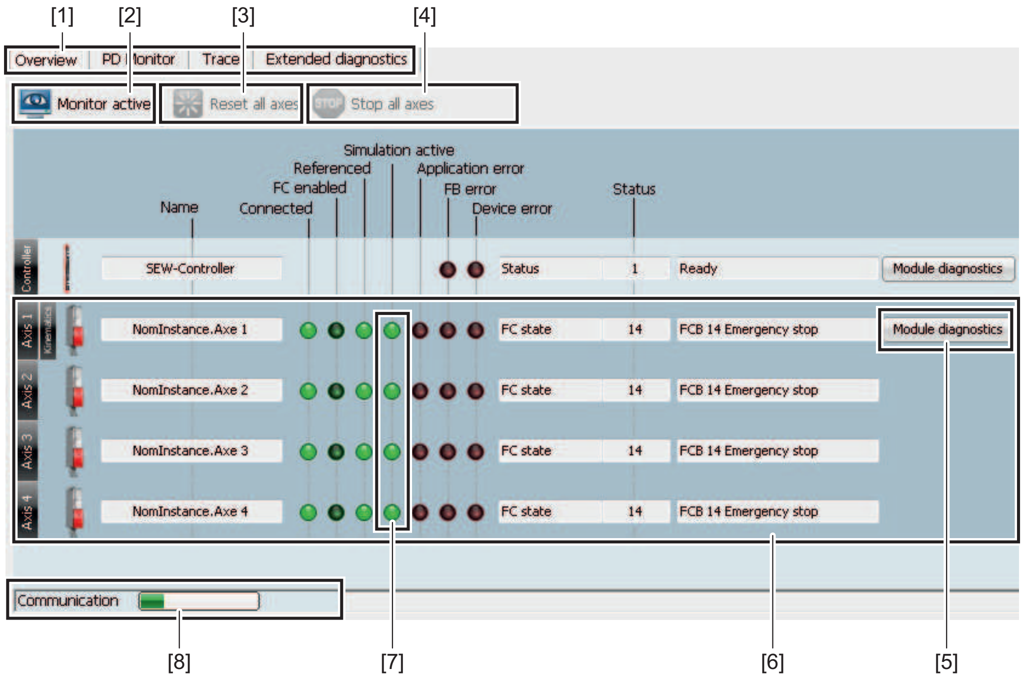
9007213353390219

The individual diagnostics methods are described in the chapters below. For detailed information about the individual Application Configurator functions, see the documentation for the "Application Configurator for CCU" configuration software.

24796131/EN – 12/2017

9.1 Application Configurator diagnostics view

When changing to the diagnostics of the Application Configurator, the following diagnostics view opens. Here you can open detailed diagnostics of the various application modules. In this case, the displayed information comes directly from the controller and is fieldbus-independent.



14179401099

No.	Description
[1]	Use the button in this group to access the following functions: • Overview (initial screen of the diagnostics view) • PD monitor • Trace • Extended diagnostics These functions are described in the following chapters.
[2]	Use this button to switch between monitor mode (display of the input and output data of the application modules only) and control mode (input data of the application modules can be controlled using the user interface). Only one tool can be in control mode at any stage. For example, if the HandlingKinematicsMonitor is in control mode, the PD monitor cannot be in control mode at the same time.
[3]	Use this button to acknowledge the faults of all axes.
[4]	Use this button to stop all of the technology module axes (for example, in the event of an emergency). Deceleration is carried out via emergency stop ramps.
[5]	Use this button to open the HandlingKinematics module diagnostics.
[6]	This section displays the individual axes. In this case, HandlingKinematics is displayed as an axis group. All of the kinematics axes are highlighted in the same color, the first axis is also highlighted by kinematics, and there is only one module diagnostics for all of the axes.

No.	Description
[7]	This signal indicates whether the axes are simulated.
[8]	This section shows the communication status of the controller. For successful diagnostics and control of the kinematic model, the status "Online" must be reported and the green communication bar must finish.

⚠ DANGER

Unexpected movement of the machine.

Severe or fatal injuries.

Unexpected movement of the machine is possible in the following situations:

- While switching from monitor mode to control mode or the other way around.
- After clearing the fieldbus input data.

→ Make sure that an automatic restart or stop of the machine represents no danger to people or equipment.

→ Make sure that the machine is in a safe state.

INFORMATION

The following requirements must be met for the kinematic model to be ready for operation:

- Diagnostics monitor communication must report the status "Online."
- All of the kinematics axes must report the following status:

→ Connected = *TRUE* (green)

→ Application error = *FALSE* (dark red)

→ Fieldbus error = *FALSE* (dark red)

→ Device error = *FALSE* (dark red)

If one of the signals is not available as required, proceed as follows:

- **Connected**
  - See the "System bus CAN 1/CAN 2" (→ 🗎 165) chapter.
- **Application error**
  - HandlingKinematics error, see the error message in the module diagnostics or MessageHandler. For detailed information, see the "Additional error codes" (→ 🗎 186) chapter and the appendix.
- **Fieldbus error**
  - Control software error: The error name is displayed on the right in the display field. You can find more detailed information in the appendix.
- **Device fault**
  - The inverter reports an error and its description, cause, and corrective measures can be read in the inverter system manual.

9.2 Module diagnostics using the HandlingKinematicsMonitor

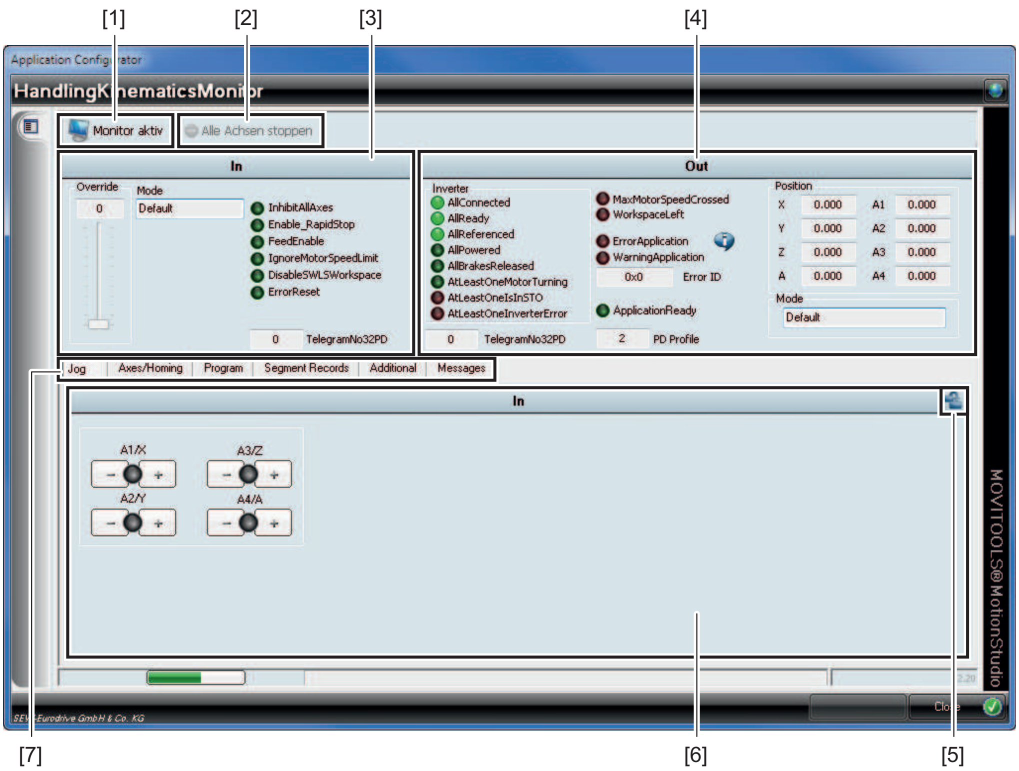
You can use the HandlingKinematics module diagnostics, the HandlingKinematicsMonitor, to view module-specific diagnostics information about the process data interface, the operating states, and errors.

To open module diagnostics, click the [Module diagnostics] button in the Application Configurator diagnostics view, see the "Diagnostics view" (→ 🗎 117) chapter.

All input and output data of the process data interface serve as basis. The data is grouped by themes and presented in graphs. In contrast to the PD monitor, a direct mapping of the fieldbus data is not displayed in this case. Instead, the actual HandlingKinematics variables that are available in the controller are visualized.

24796131/EN – 12/2017

To help understand the relationship, the relevant process data is added before the variable names below. Additional internal HandlingKinematics variables that should simplify the diagnostics are added to the variables.

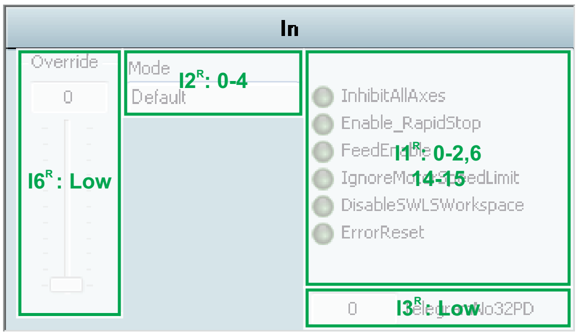


14175000587

No.	Description
[1]	Use this button to switch between monitor mode (display of the variable values only) and control mode (input variables can be controlled using the user interface). **INFORMATION:** Control mode cannot be activated at the same time as the control mode of a different tool. For example, if the HandlingKinematicsMonitor is in control mode, the PD monitor cannot be in control mode at the same time.
[2]	Use this button to stop all of the technology module axes (for example, in the event of an emergency). Deceleration is carried out via emergency stop ramps.
[3]	This area displays the general input data for controlling the kinematics. The input data displayed here is operating mode-independent and function-independent. The data can be changed from the user interface once control mode is selected.
[4]	This area displays the general output data. The data displays the current operating mode-independent and function-independent state and cannot be changed directly by the user. In addition, this area continues to display the configured PD profile.
[5]	You can use this icon to detach the current tab from the monitor. All of the tabs can then be displayed simultaneously on the screen. **INFORMATION:** This icon does not apply to the "Segment Records" tab.
[6]	This area displays the specific input and output data.
[7]	You can use this button to switch between the tabs for specific input and output data.

24796131/EN – 12/2017

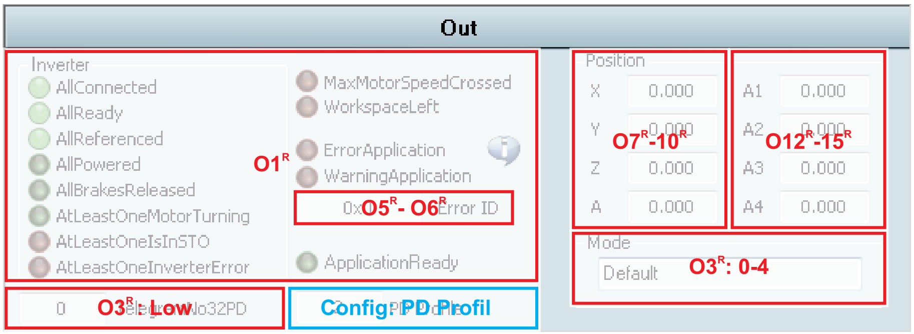
Input data



14175005963

Signal	Meaning
I6^r:Low	*Override*
I2^r:0 – 4	*Operating mode*
I1^r:0 – 2, 6, 14 – 15	Part of the *control word*
I3^r:Low	*Telegram number*

Output data



9007213429744011

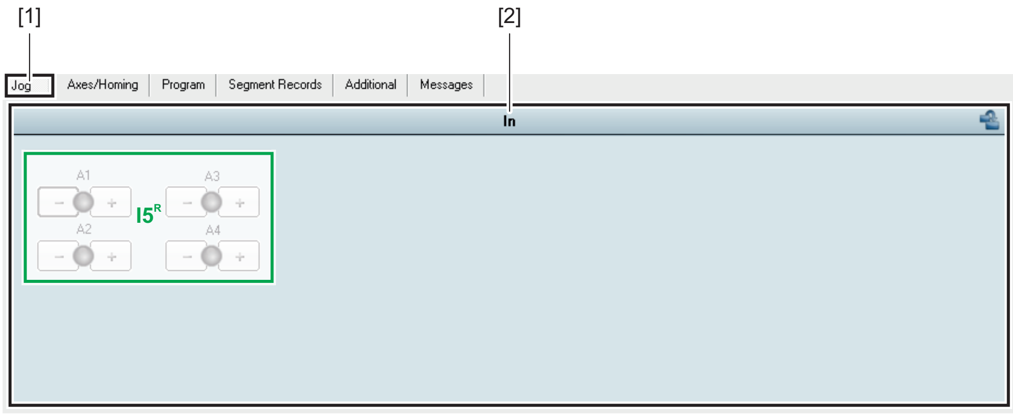
Signal	Meaning
O1^r	*Status word*
O3^r:Low	*Telegram number*
O5^r – O6^r	*ErrorID*
O7^r – O10^r	*Current pose*
O12^r – O15^r	*Axis values 1 to 4*
O3^r:0 – 4	*Operating mode*

9.2.1 Tab: Jog

This tab displays the input variables for the jog.

INFORMATION

The buttons on this page click into place. That is, when you click the button, the variable remains set if you release the mouse button (→ movement). When you click the button again, the variable is reset (→ stop the movement). The pushbuttons do not return to their prior state when you release the mouse button.



14175009803

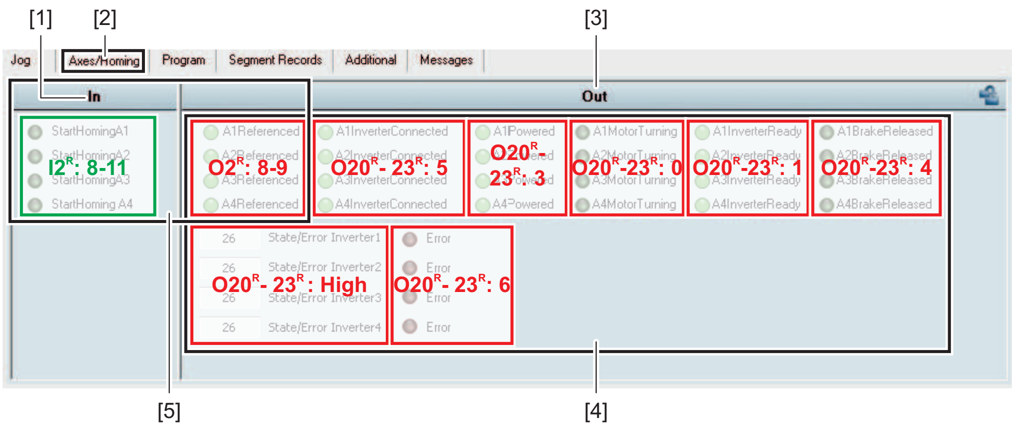
No.	Description
[1]	The "Jog" tab is selected.
[2]	This area displays the input data.

Input data

Signal	Meaning
$I5^{r}$	*Jog*

9.2.2 Tab: Axes/Homing

This tab displays the following information.



14175012491

No.	Description
[1]	This area displays the input data of the *Homing* referencing mode.
[2]	The "Axes/Homing" tab is selected.
[3]	This area displays the output data of the *Homing* referencing mode.
[4]	This area displays the output data for the status of the individual axes ($O20^{r}$ to 23^{r}).
[5]	This area displays the input and output data of the *Homing* referencing mode.

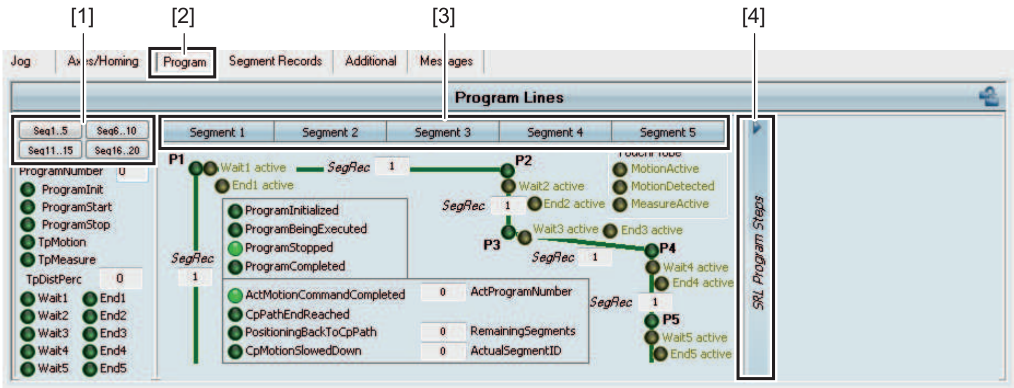
24796131/EN – 12/2017

Input and output data

Signal	Meaning
I2^{r}:8 to 11	*Start referencing*
O2^{r}:8 to 9	*Axes referenced*
O20^{r} to O23^{r}:5	*Inverter connected*
O20^{r} to O23^{r}:3	*Output stage enabled*
O20^{r} to O23^{r}:0	*Motor is turning*
O20^{r} to O23^{r}:1	*Inverter ready*
O20^{r} to O23^{r}:4	*Brake released*
O20^{r} to O23^{r}:High	*Inverter status or error number*
O20^{r} to O23^{r}:6	*Inverter error status*

9.2.3 Tab: Program

This tab is explained using the overview of path segments 1 to 5 from profile 4. The tab displays the following information.

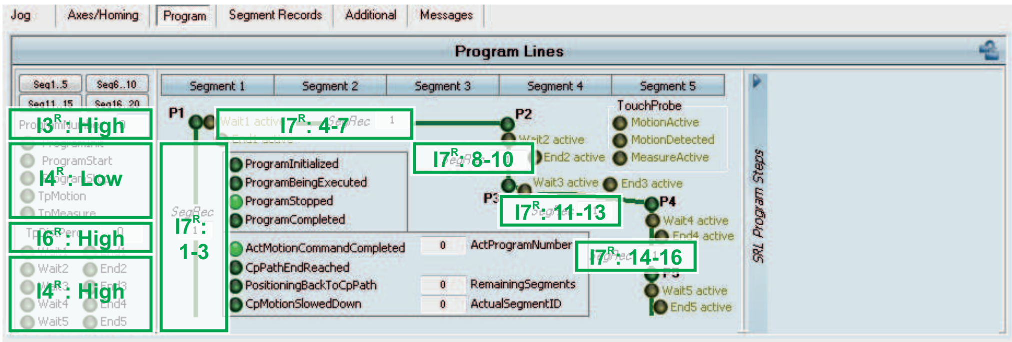


14175045643

No.	Description
[1]	Use this button to switch between the groups of the path segments.
[2]	The "Program" tab is selected.
[3]	Use this button to open the settings and information for the individual path segments (see the "Path segments" section).
[4]	This button is used to open the SRL[1] program and display the program progress. Above all, displaying the program is helpful in the *Program step* program mode (see the "SRL Program steps" section).

1) SEW Robot Language

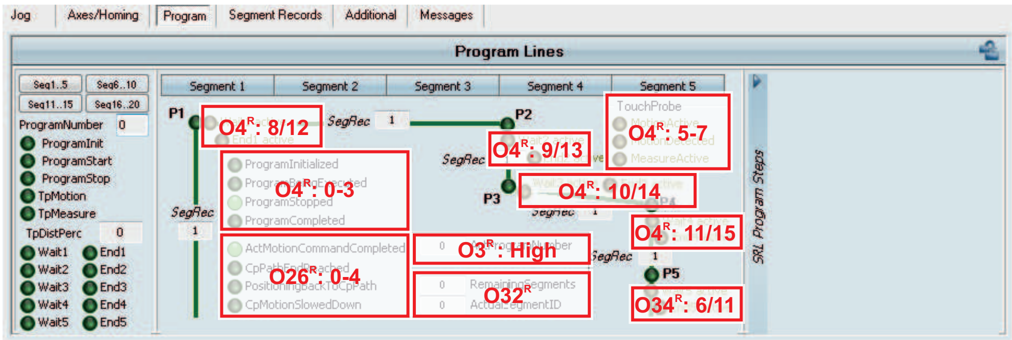
Input data



14175060747

Signal	Meaning
$I3^r$:High	*Program number*
$I4^r$:Low	*Control program: General*
$I6^r$:High	*Touchprobe: Remaining Distance Percentage*
$I4^r$:High	*Control program: Wait/End*
$I7^r$:1 – 3	*Segment parameter sets for segment 1*
$I7^r$:4 – 7	*Segment parameter sets for segment 2*
$I7^r$:8 – 10	*Segment parameter sets for segment 3*
$I7^r$:11 – 13	*Segment parameter sets for segment 4*
$I7^r$:14 – 16	*Segment parameter sets for segment 5*

Output data

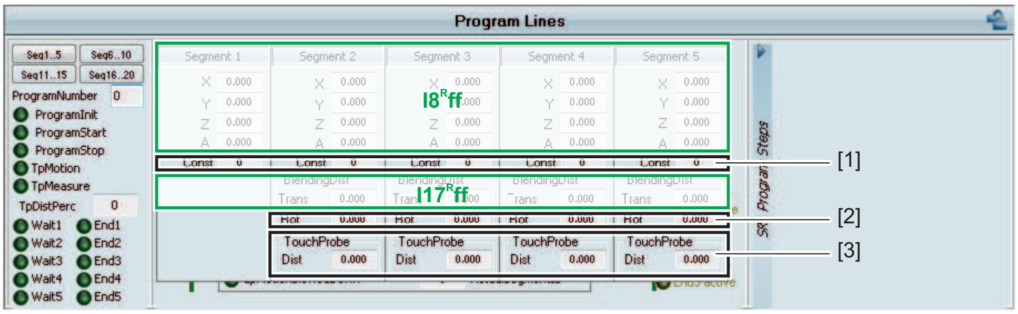


9007213429784203

Signal	Meaning
$O4^r$:8/12	*Wait/End 1*
$O4^r$:0 – 3	*Status program: General*
$O26^r$:0 – 4	*Program status*
$O4^r$:9/13	*Wait/End 2*
$O4^r$:5 – 7	*Status program: Touchprobe*
$O4^r$:10/14	*Wait/End 3*
$O3^r$:High	*Program number*
$O32^r$	*Current segment ID/Remaining CP segments*
$O4^r$:11/15	*Wait/End 4*
$O34^r$:6/11	*Wait/End 5*

24796131/EN – 12/2017

Path segments



14177160843

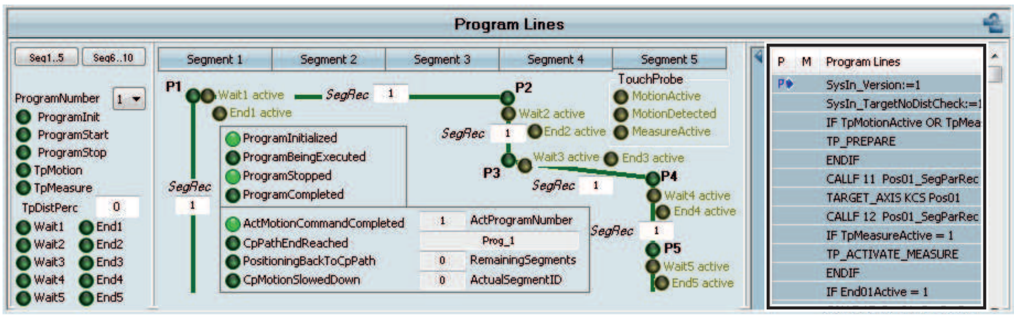
No.	Description
[1]	*Constellation* configuration parameter for the selected segment parameter set.
[2]	*Rotary blending* configuration parameter for the selected segment parameter set, see the "Configuring the motion parameters" (→ 🗎 88) chapter.
[3]	Calculated remaining travel length l_r, see the "Touchprobe Motion" (→ 🗎 61) chapter.

Input data

Signal	Meaning
I8^{r}ff.	*Segment target pose*
I17^{r}ff.	*Blending distance to the next segment*

SRL Program Steps

To open the SRL[1)] program, click the [SRL Program Steps] button. The program progress is displayed.



14175193867

The blue program indicator [P→] ("P →" for program) is used to flag which command is to be processed now or to be processed next.

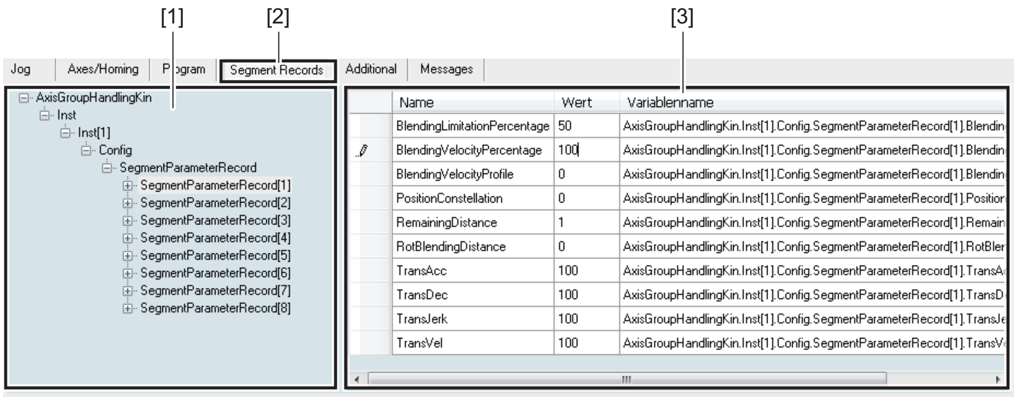
The orange motion pointer [M→] ("M →" for motion) is used to flag which motion command the kinematic control is processing, i.e. that is, which path segment the kinematic model is currently moving from.

The program indicator may deviate from the motion pointer because, for example, in the case of a *CP* motion type (e.g. LIN), the *CP* command only affects entries to the list of path segments to be traveled (queue). This does not necessarily mean that the movement to this position has already started. The commands of the SRL language and SRL programs are listed in the appendix.

1) SRL = SEW Robot Language.

9.2.4 Tab: Segment Records

On this tab, the *AxisGroupHandlingKin* structure tree is displayed and the parameters of the segment parameter sets are listed individually. This allows the values in the segment parameter sets to be changed and optimized in control mode without having to switch to the application module configuration (e.g. setting a higher path speed, increasing jerk times, and so on).



14175199627

No.	Description
[1]	This area displays the *AxisGroupHandlingKin* structure tree together with the available segment parameter sets.
[2]	The "Segment Records" tab is selected.
[3]	This area displays the parameters of the segment parameter set that is selected in the structure tree.

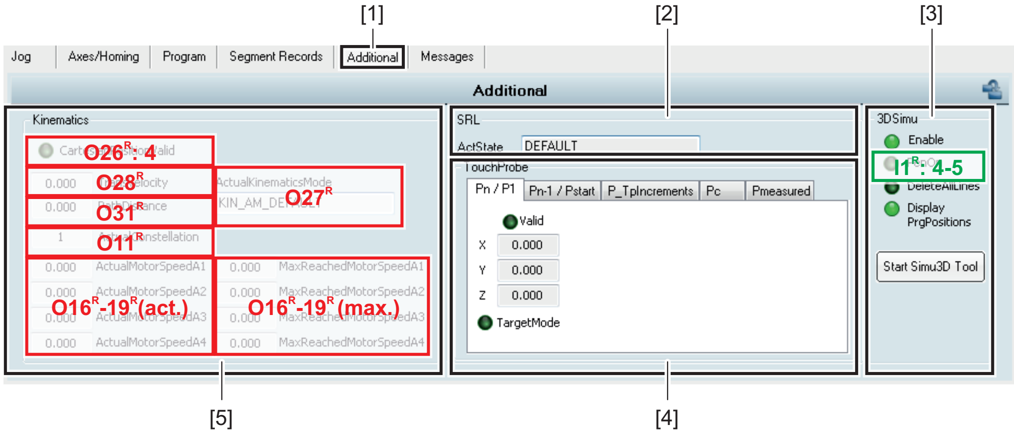
The segment parameter sets correspond to the configured values and can be changed either by reconfiguring them or as volatile by overwriting the assigned value in the "Value" column (if control is active, confirm with the enter key). When overwriting using the HandlingKinematicsMonitor, the data overwritten is volatile, i.e. it is lost after the controller is restarted. To write the data to the configuration permanently, click the "Write to MemoryCard" button. It is displayed with the ADVANCED permission level or higher; see the "Advanced configuration or higher permission level" (→ 🗎 96) chapter.

For more information about the meaning of the segment parameter sets, see the "Overview of segment parameter sets" (→ 🗎 89) chapter.

24796131/EN – 12/2017

9.2.5 Tab: Additional

This tab displays the following information.



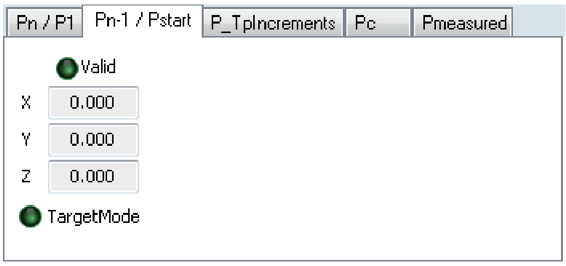
14175203723

No.	Description
[1]	The "Additional" tab is selected.
[2]	This area displays the current SRL status.
[3]	You can control the 3D simulation in this area. • The 3D simulation can be started by choosing *Start Simu3D Tool* and activated by choosing *Enable*. • The pen for drawing the motion path can be activated by choosing *PenOn* and reset by choosing *DeleteAllLines*. • The *target pose* for program mode (I8^{r}ff.) can be displayed by choosing *Display PrgPositions*.
[4]	This area displays the information about *Touchprobe Motion* and *Touchprobe Measure* (see the "Touchprobe tabs" section).
[5]	This area displays the state information of the kinematic model.

Input and output data

Signal	Meaning
O26^r:4	*Cartesian position valid*
O28^r	*Translational speed*
O31^r	*CP path: Target distance*
O11^r	*Constellation*
O16^r – O19^r (act.)	*Current speed (motor 1 to 4)*
O27^r	*Kinematic model operating mode*
O16^r – O19^r (max.)	*Maximum speed (motor 1 to 4)* can be reset with the signal I1^r:6 *Error reset*
I1^r:4 – 5	*Delete 3D simulation pen and lines*

Touchprobe tabs



14179391755

Tabs	Description
P_n / P_1	Information about the last *target position* for *Touchprobe Motion* or about the first *target position* for *Touchprobe Measure*.
P_{n-1} / P_{start}	Information about the second last *target position* for *Touchprobe Motion* or about the *start position* for *Touchprobe Measure*.
$P_{TpIncrements}$	Information about the position at which the Touchprobe signal was triggered.
P_c	Calculated *target pose* for the *Touchprobe Motion* sensor-based positioning, see the "Touchprobe Motion" (→ 🗎 61) chapter.
$P_{measured}$	$O7^r$ to $O9^r$ *Position for Touchprobe Measure* (XYZ), see the "Touchprobe Measure" (→ 🗎 64) chapter.

The tabs display the information about the following issues that are relevant to the Touchprobe functions:

- Positions P_n / P_1 or P_{n-1} / P_{start}

  The information about the last 2 *target positions* or the first *target position* and the *start position* of the motion sequence, between which the *Touchprobe Motion* or *Touchprobe Measure* is activated and with which P_c is calculated from $P_{TpIncrements}$.

  – **X/Y/Z**

  The last/first/second last *target position* or *start position* of the motion sequence.

  – **Valid**

  The position is valid. If it is not valid, a position has not been entered yet. It can be used to detect whether values that are still initial or are obsolete are entered in XYZ.

  – **TargetMode**

  The movement to the position is a *TARGET* motion task.

- $P_{TpIncrements}$

  Measured Touchprobe position in motor increments.

  – **Increments**

  > **A1/A2/A3/A4**:

  The increments reported by the inverters after the Touchprobe signal was triggered there.

  – > **Valid**:

The entries for the increments reported by the inverters are valid. This can be used to detect whether values that are still initial, are obsolete, or are incorrect are entered in A1/A2/A3/A4.

– **Touchprobe Counter**

  > **A1/A2/A3/A4**

  Counter for counting how many Touchprobe events occurred on the respective axis inverters. If they are not the same, an error is issued.

  > **Detected Touchprobe Counter**

  Maximum value of the Touchprobe counters for the axes.

- P_c

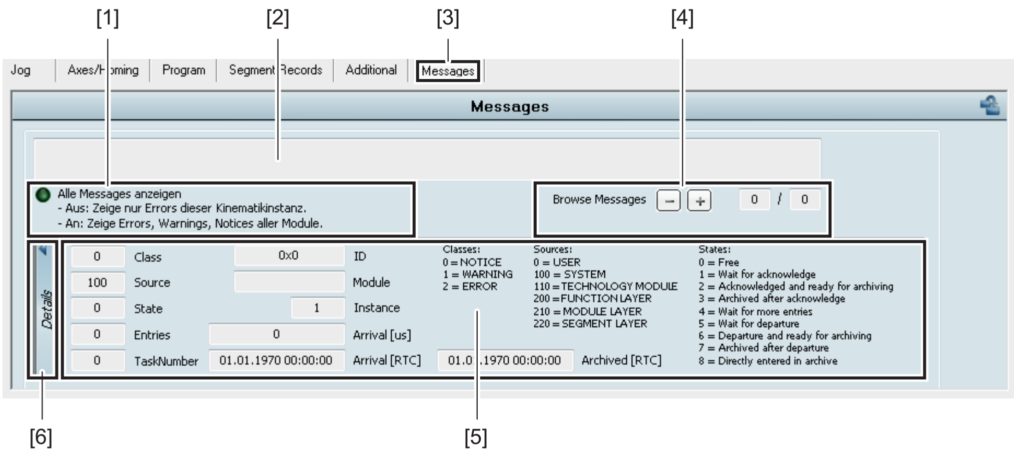
  Calculated *target pose* for the sensor-based positioning of the *Touchprobe Motion*.

- $P_{measured}$

  Measured position of the *Touchprobe Measure*.

9.2.6 Tab: Messages

You can find a selection of MessageHandler messages on this tab. In this case, only the kinematic instance errors are displayed by default. If you also want to display warnings, notices, and errors for all modules, activate the "Show all messages" setting [1].



14179396747

No.	Description
[1]	You can activate or deactivate the "Show all messages" setting in this area.
[2]	The text of the selected MessageHandler message is displayed in this area.
[3]	The "Messages" tab is selected.
[4]	In this area, you can select the messages to be displayed and use the "Browse Messages" arrows to navigate through the messages.
[5]	In this area, you display the details about the selected message by clicking the [Details] button.
[6]	You can use this button to display the details about the message that is currently selected. The explanations for the "Class," "Source," and "State" values are displayed on the right.

You can find a detailed list of all of the possible errors in the appendix.

INFORMATION

For an overview display of all pending and archived error, warning, and notice messages, we recommend the MessageHandler, which can be started under "Diagnostics" in the controller context menu; see the "MessageHandler" (→ 🗎 134) chapter.

9.3 PD monitor

The process data monitor (PD monitor) is used for diagnostics and to learn about the fieldbus interface. The content of the PD monitor consists of the data of the SEW controller and all of the configured application modules. The PD monitor accesses the fieldbus interface data only.

⚠ DANGER

Unexpected movement of the machine.

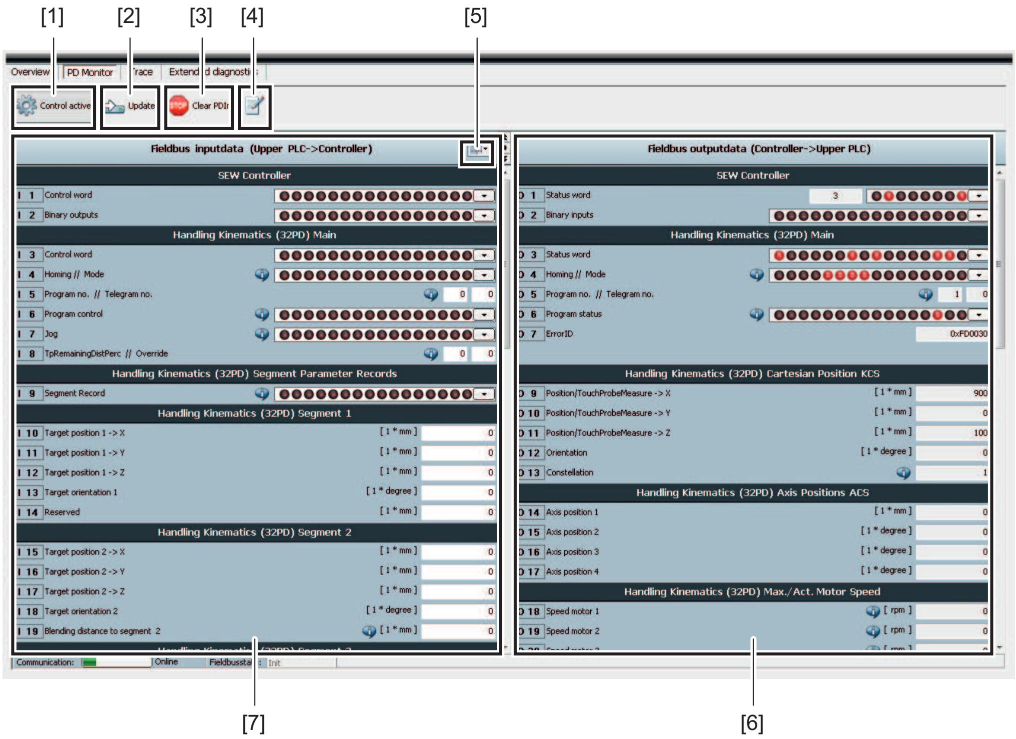
Severe or fatal injuries.

Unexpected movement of the machine is possible in the following situations:

- While switching from monitor mode to control mode or the other way around.
- After clearing the fieldbus input data.

→ Make sure that an automatic restart or stop of the machine represents no danger to people or equipment.

→ Make sure that the machine is in a safe state.



14182215819

No.	Description
[1]	Use this button to switch between monitor mode (display of the input and output data only) and control mode (input data can be controlled using the user interface). **INFORMATION:** Control mode cannot be activated at the same time as the control mode of an application module.
[2]	This button is used to transmit all of the control signals and setpoints (fieldbus input data) to the controller. The button is only enabled in control mode.
[3]	Use this button to delete all of the fieldbus input data. In most applications, the deletion of fieldbus input data causes all of the axes to perform a rapid stop once the data is transmitted with [2].
[4]	Use this button to create a PDF file with the process data (e.g. for the programmer of the PLC).
[5]	You can use this icon to save or download the current assignment of the fieldbus input data for later use. Click the icon and select the desired option.
[6]	This area displays the fieldbus output data. **INFORMATION:** The fieldbus output data is used only for display and cannot be changed directly from the user interface.
[7]	This area is used to display the fieldbus input data. **INFORMATION:** You can change the fieldbus input data in control mode.

The most important part of the PD monitor is the fieldbus input data and output data that is exchanged between the controller and the PLC. This contains the fieldbus data of all of the configured application modules.

The output data cannot be changed directly. The input data can be changed in control mode from the user interface.

Its operation depends on the data type:

- **Bit:** Activate/deactivate by clicking the relevant LED or the check box for the bit after opening the byte/word drop-down list.
- **Byte or word:** Enter the setpoint in the relevant edit box and confirm by pressing the enter key.

For detailed information about the individual HandlingKinematics fieldbus data, refer to the corresponding information field or the "Communication and process data exchange" (→ 🗎 36) and "Process data assignment" (→ 🗎 135) chapters.

9.4 Trace

The trace can be used to trace various process signals (speeds, position of the axes, and so on) for the individual axes. In this case, up to 4 channels can be traced at the same time. For detailed information about this, refer to the Application Configurator documentation.

9.5 Extended diagnostics

The extended diagnostics tool is for expert diagnostics. You can find the variable structures of all of the global variables in the controller here. In this case, the important variables for HandlingKinematics include:

- *AxisGroupHandlingKin*
- *AxisGroupHandlingKinProfile*
- *AxisGroupKin*

To be able to describe variables in the advanced diagnostics, the "Advanced" MotionStudio authorization level is required at minimum. In addition, you must select the "Write" entry in the "Mode" choice box.

9.6 3D simulation

The integrated 3D simulation in MOVITOOLS® MotionStudio allows the automatic simulation of the kinematics. The model automatically adapts to the specified configurations.

3D simulation is very useful as feedback for the following tasks:

- Configuring the kinematic model and adaptation to the real robot
- Learning how to operate the kinematic control
- Checking motion sequences

3D simulation can be carried out both offline and online:

- Offline means that no real axes are connected or moved.
- Online means that the simulation is carried out in parallel with the real robot motion.
- In both cases, controller hardware (CCU DHR41B/DHF41B) on which the "HandlingKinematics" application module is executed is required.

Online 3D simulation is very useful for visualizing the tool motion of the robot. Appraising the tool path in terms of the path type and blending behavior is generally quite difficult for the human eye.

24796131/EN – 12/2017

9.6.1 Requirements

- Note that you require an additional 10 technology points for the full version (unlimited runtime) of the 3D simulation. For detailed information about this, see the "Technology points" (→ 🗎 15) chapter.
- If the engineering PC with the 3D simulation is in the same Ethernet subnetwork, the IP address necessary for the 3D simulation is automatically determined. In the advanced configuration or on the "Advanced" MotionStudio permission level or higher, the IP address of the engineering PC can also be set explicitly in the configuration wizard.
- If you do not have access to real axes, activate the simulation of the axes in the Application Configurator; see the "Adding the HandlingKinematics application module" (→ 🗎 70).
- Establish an Ethernet connection between the controller and engineering PC. Communication with the 3D simulation takes place exclusively via the Ethernet engineering interface (X37).

INFORMATION

3D simulation is not possible via USB or fieldbus.

The 3D simulation must be activated in order to perform the 3D simulation. It is activated by default. However, the default setting can be changed in the expert configuration on the "Advanced" MotionStudio permission level or higher or in the advanced configuration.

Activating or deactivating the 3D simulation

Proceed as follows:

1. Go to the Application Configurator diagnostics view.
2. Click the [Module diagnostics] button in the HandlingKinematics application module.
   ⇨ The HandlingKinematicsMonitor is displayed.
3. Open the "Additional" tab.
4. In the "3DSimu" area, you can use the *Enable* setting to activate or deactivate the 3D simulation at runtime; see the "Tab: Additional" (→ 🗎 127) chapter.

9.6.2 Start

If the simulation is still activated in the expert configuration, the 3D simulation starts automatically when the HandlingKinematicsMonitor is opened.

You can also start the 3D simulation explicitly as follows:

- In the MOVITOOLS® MotionStudio menu bar, choose the [Settings] > [Extras] > [3D simulation] menu command.
- In the HandlingKinematicsMonitor, select the "Start Simu3D Tool" setting in the "3DSimu" area on the "Additional" tab, see the "Tab: Additional" (→ 🗎 127) chapter.

24796131/EN – 12/2017

9.6.3 Adjusting the 3D simulation

You can use the following settings to adapt the appearance of the 3D simulation on the tabs of the control window that starts automatically with the 3D simulation:

- **Pencils:** You can set the color of the lines, for example. The pencils cannot be activated/deactivated here.
- **Coordinate systems:** You can display the various coordinate systems here to check the transformations that are currently valid.
- **Workspace:** Here, you can display the arrows for displaying the zero points and directions of the valid travel ranges for the axes, the Cartesian dimensions, and the kinematic limitations.
- **Parameter arrows:** You can display useful information using arrows here (e.g. for identifying the kinematic parameters such as arm lengths and offsets).
- **Bodies:** You can show/hide selected bodies.

9.6.4 Setting options using the fieldbus

You can configure the following settings for the 3D simulation using the fieldbus:

- I1^{r}:4 *3D simulation pen*

  Activate the pen for drawing the path of the tool in the simulation.
- I1^{r}:5 *3D Delete 3D simulation lines*

  Delete the tool path drawn by the pen.

9.7 MessageHandler

The MessageHandler is a diagnostics tool that displays information on errors in plain text and in a clear manner. This information supplements the information that you receive about the error number (O5^{r}-O6^{r} *ErrorID*) and makes troubleshooting easier.

INFORMATION

If you want to use the MessageHandler, we recommend establishing the connection to the controller via the Ethernet engineering interface (X37). Diagnostics is also possible via USB and fieldbus, but the transmission of the messages is slower.

Start MessageHandler

Proceed as follows:

1. Select the controller in the network view of MOVITOOLS® MotionStudio.
2. Right-click to open the context menu of the controller.
3. Select the [Diagnostics] > [MessageHandler] menu command.

⇨ The MessageHandler starts.

Current error messages are located in the top area. Messages that have already been archived are displayed below that.

10 Process data assignment

10.1 Fieldbus input data: Profile 1 with 32 PD

The fieldbus input data must be transmitted consistently; see the "Overview: profile 1 with 32 PD" (→ 🗎 37) chapter.

PD	Name	Consistency blocks
--	*SEW-EURODRIVE controller*	**2 PD**
I1^r	*Control word (kinematics)*	**32 PD**
I2^r	*Referencing/operating mode*	
I3^r	*Program no. /telegram no.*	
I4^r	*Control program*	
I5^r	*Jog (control word)*	
I6^r	*Touchprobe / override*	
I7^r	*Assignment of the segment parameter sets to the 5 segments*	
I8^r – I11^r	*Target pose 1 → XYZA*	
I12^r	--	
I13^r – I16^r	*Target pose 2 → XYZA*	
I17^r	*Blending distance to segment 2*	
I18^r – I21^r	*Target pose 3 → XYZA*	
I22^r	*Blending distance to segment 3*	
I23^r – I26^r	*Target pose 4 → XYZA*	
I27^r	*Blending distance to segment 4*	
I28^r – I31^r	*Target pose 5 → XYZA*	
I32^r	*Blending distance to segment 5*	

INFORMATION

Below, the abbreviation RTRIG (rising trigger) means that there is a rising edge (FALSE→ TRUE) of the bit.

The following tables show the fieldbus input data from the PLC to the kinematic model for fieldbus control with profile 1 with 32 process data words.

I1^r *control word*				
Bit	**Name**	**Description**		**See chapter**
0	*Controller inhibit*	TRUE:	Activates the controller inhibit for all axes.	(→ 🗎 32) (→ 🗎 33)
1	*Enable/Rapid stop*	TRUE:	Enables the interpolated position control of the kinematic axes.	
		FALSE:	The kinematic model performs a rapid stop.	

24796131/EN – 12/2017

I1^{r} *control word*			
Bit	**Name**	**Description**	**See chapter**
2	*Feed enable*	TRUE: Enables the movement (of the feed) of the kinematic model. FALSE: The kinematic model stops on the path.	(→ 🗎 32) (→ 🗎 33)
3	--		–
4	*3D simulation pen*	TRUE: Activates the pen in the 3D simulation.	(→ 🗎 134)
5	*Delete 3D simulation lines*	RTRIG: Deletes the lines in the 3D simulation.	
6	*Error reset*	RTRIG: The following sizes are reset: • O1^{r}:7 *Application fault* • O16^{r} – O19^{r} *Rotational speed motor* Highest speed last reached.	(→ 🗎 100)
7	*Actual motor speed output*	TRUE: The signal O16^{r} – O19^{r} *Motor Speed* displays the current motor speed. FALSE: The signal O16^{r} – O19^{r} *Motor Speed* displays the highest motor speed reached last.	–
8	*Output Touchprobe Measure*	TRUE: In O7^{r} to O10^{r} *Position/Touchprobe Measure*, the measured pose (XYZA) for the kinematics coordinate system (KCS) that was last determined in the first *Touchprobe Measure* path segment is output. FALSE: In O7^{r} – O10^{r} *Position/Touchprobe Measure*, the current pose (X, Y, Z, A) for the kinematics coordinate system (KCS) is output.	(→ 🗎 61)
9 – 13	--		–
14	*Ignore overspeed*	TRUE: Motor speed limitation is ignored. It is still evaluated and output in the signal O1^{r}:14 *Speed limit exceeded*.	(→ 🗎 33)
15	*Ignore work envelope monitoring*	TRUE: Kinematic, Cartesian, and axis limits are ignored. They are still evaluated and output in the signal O1^{r}:15 Leave work envelope.	

I2^{r} *Referencing / operating mode*			
Bit	**Name**	**Description**	**See chapter**
0	*Axis-by-axis jog*	TRUE: Selects the *Axis-by-axis jog* operating mode.	(→ 🗎 44)
1	*Cartesian jog*	TRUE: Selects the *Cartesian jog* operating mode.	
2	*Referencing*	TRUE: Selects the *Referencing* operating mode.	(→ 🗎 45)
3	*Program Auto*	TRUE: Selects the *Program auto* operating mode.	(→ 🗎 47)

I2ʳ *Referencing / operating mode*			
Bit	**Name**	**Description**	**See chapter**
4	*Program Step*	TRUE: Selects the *Program step* operating mode.	(→ 🗎 50)
5 – 7	--		–
8	*Start referencing axis 1*	TRUE: Starts referencing the respective axis in referencing mode.	(→ 🗎 45)
9	*Start referencing axis 2*		
10	*Start referencing axis 3*		
11	*Start referencing axis 4*		
12	*Start referencing axis 5*		
13	*Start referencing axis 6*		
14	*Start referencing axis 7*		
15	*Start referencing axis 8*		

I3ʳ *Program number / telegram number*			
Byte	**Name**	**Description**	**See chapter**
High	*Program number*	Program to be executed in *Program auto / step* operating mode.	(→ 🗎 51)
Low	*Telegram number*	Telegram number that ensures the consistency of the message: Only option for profile 1, required for profiles 2 to 4.	(→ 🗎 102)

I4ʳ *Program control*			
Bit	**Name**	**Description**	**See chapter**
0	*Program init*	RTRIG: Initializes the program in *Program Auto / step*.	(→ 🗎 46) (→ 🗎 101)
1	*Program start*	RTRIG: Executes the program in *Program Auto / step*.	
2	*Program stop*	TRUE: Stops the program in *Program Auto / step*.	
3 – 4	--		–
5	*Touchprobe Motion*	TRUE: Activates *Touchprobe Motion* (sensor-based positioning).	(→ 🗎 61)
6	*Touchprobe Measure*	TRUE: Activates the *Touchprobe Measure* (positioning measurement).	
7	--		–

I4^{r} *Program control*			
Bit	**Name**	**Description**	**See chapter**
8	*Wait 1*	TRUE: Forced stop at *Target pose 1 – 4*. FALSE: Approval to travel the next path segments 2 to 5 is granted if the path was not already canceled by the signal I4^{r}:12ff. End.	(→ 🗎 56)
9	*Wait 2*		
10	*Wait 3*		
11	*Wait 4*		
12	*End 1*	TRUE: Ends the program at *target poses 1 to 4*, where the first active end signal comes into effect.	(→ 🗎 56)
13	*End 2*		
14	*End 3*		
15	*End 4*		

I5^{r} *Jog mode*			
Bit	**Name**	**Description**	**See chapter**
0	*Jog positive for axis 1 / Position X*	TRUE: For *Axis-by-axis jog*: • Jog in positive A1, A2, A3, or A4 direction For *Cartesian jog*: • Jog in positive X, Y, Z, or A direction	(→ 🗎 44)
1	*Jog positive for axis 2 / Position Y*		
2	*Jog positive for axis 3 / Position Z*		
3	*Jog positive for axis 4 / Orientation*		
4	*Jog positive for axis 5*	TRUE: For *Axis-by-axis jog*: • Jog in positive A5, A6, A7, and A8 direction (if configured)	
5	*Jog positive for axis 6*		
6	*Jog positive for axis 7*		
7	*Jog positive for axis 8*		
8	*Jog negative for axis 1 / Position X*	TRUE: For *Axis-by-axis jog*: • Jog in negative A1, A2, A3, or A4 direction For *Cartesian jog*: • Jog in negative X, Y, Z, or A direction	
9	*Jog negative for axis 2 / Position Y*		
10	*Jog negative for axis 3 / Position Z*		
11	*Jog negative for axis 4 / Orientation*		
12	*Jog negative for axis 5*	TRUE: For *Axis-by-axis jog*: • Jog in negative A5, A6, A7, and A8 direction (if configured)	(→ 🗎 44)
13	*Jog negative for axis 6*		
14	*Jog negative for axis 7*		
15	*Jog negative for axis 8*		

I6ʳ *Touchprobe / Override*			
Byte	**Name**	**Description**	**See chapter**
High	*Touchprobe: Remaining Distance Percentage*	Scaling of the remaining path length parameterized in the segment parameter sets in percent (*Touchprobe Motion*).	(→ 🗎 61)
Low	*Override*	Percentage scaling of the speed for all operating modes apart from *Referencing*.	(→ 🗎 34)

I7ʳ *Assignment of the segment parameter sets to the 5 segments*			
Bit	**Name**	**Description**	**See chapter**
0	*Segment 1 bit 1/3*	Selection of the segment parameter sets for path segments 1 to 5 using the relevant 3 bits. In this case, the following conversion from decimal (minus 1) to binary applies: 1 = 000, 2 = 001, 3 = 010, 4 = 011, 5 = 100, 6 = 101, 7 = 110, 8 = 111.	(→ 🗎 34)
1	*Segment 1 bit 2/3*		
2	*Segment 1 bit 3/3*		
3	*Seg. 2 bit 1/3*		
4	*Seg. 2 bit 2/3*		
5	*Seg. 2 bit 3/3*		
6	*Segment 3 bit 1/3*		
7	*Segment 3 bit 2/3*		
8	*Segment 3 bit 3/3*		
9	*Seg. 4 bit 1/3*		
10	*Seg. 4 bit 2/3*		
11	*Seg. 4 bit 3/3*		
12	*Segment 5 bit 1/3*		
13	*Segment 5 bit 2/3*		
14	*Segment 5 bit 3/3*		
15	--		–

I8ʳ – I12ʳ *Segment 1*				
PD		**Name**	**Description**	**See chapter**
I8ʳ	INT	*Target position 1 → X*	Default *target pose* for path segment 1.	(→ 🗎 31)
I9ʳ	INT	*Target position 1 → Y*		
I10ʳ	INT	*Target position 1 → Z*		
I11ʳ	INT	*Target orientation 1*		
I12ʳ	Word	--		–

I13^r – I17^r *Segment 2*

PD		Name	Description	See chapter
I13^r	INT	*Target position 2 → X*	Default *target pose* for path segment 2.	(→ 🗎 31)
I14^r	INT	*Target position 2 → Y*		
I15^r	INT	*Target position 2 → Z*		
I16^r	INT	*Target orientation 2*		
I17^r	Word	*Blending distance to segment 2*	Distance to *target position 1* from which blending to path segment 2 occurs.	(→ 🗎 31)

I18^r – I22^r *segment 3*

PD		Name	Description	See chapter
I18^r	INT	*Target position 3 → X*	Default *target pose* for path segment 3.	(→ 🗎 31)
I19^r	INT	*Target position 3 → Y*		
I20^r	INT	*Target position 3 → Z*		
I21^r	INT	*Target orientation 3*		
I22^r	Word	*Blending distance to segment 3*	Distance to *target position 2* from which blending to path segment 3 occurs.	(→ 🗎 31)

I23^r – I27^r *segment 4*

PD		Name	Description	See chapter
I23^r	INT	*Target position 4 → X*	Default *target pose* for path segment 4.	(→ 🗎 31)
I24^r	INT	*Target position 4 → Y*		
I25^r	INT	*Target position 4 → Z*		
I26^r	INT	*Target orientation 4*		
I27^r	Word	*Blending distance to segment 4*	Distance to *target position 3* from which blending to path segment 4 occurs.	(→ 🗎 31)

I28^{r} – I32^{r} *segment 5*				
PD		**Name**	**Description**	**See chapter**
I28^{r}	INT	*Target position 5 → X*	Default *target pose* for path segment 5.	(→ 🗎 31)
I29^{r}	INT	*Target position 5 → Y*		
I30^{r}	INT	*Target position 5 → Z*		
I31^{r}	INT	*Target orientation 5*		
I32^{r}	Word	*Blending distance to segment 5*	Distance to *target position 4* from which blending to path segment 5 occurs.	(→ 🗎 31)

10.2 Fieldbus output data: Profile 1 with 32 PD

The following tables show the fieldbus output data from the controller to the PLC for field control with profile 1 with 32 process data words.

O1^{r} *Status word*				
Bit	**Name**	**Description**		**See chapter**
0	*At least one motor is turning*	TRUE:	At least one motor is turning (speed ≠ 0).	Manual for the "MultiMotion" program module[1]
		See:	O20^{r} – 23^{r}:0 *Motor is turning*	
1	*All frequency inverters are ready*	TRUE:	The inverters are ready for interpolation/ for control by the controller.	
		See:	O20^{r} – 23^{r}:1 *FI ready for operation*	
2	*All drives are referenced*	TRUE:	All drives are referenced (signal is set to *FALSE* for at least 2 seconds after the rising edge of the signals I2^{r}:8 to I2^{r}:11 *Start referencing axis 1 – 4*).	
		See:	O20^{r} – 23^{r}:2 *Drive referenced*	
3	*Application ready*	TRUE:	The application is ready for operation.	(→ 🗎 100)
4	*All brakes are released*	TRUE:	The brakes of all drives are released.	Manual for the "MultiMotion" program module[1]
		See:	O20^{r} – 23^{r}:4 *Brake open*	
5	*At least one FI error*	TRUE:	At least one inverter has an error status.	
		See:	O20^{r} – 23^{r}:6 *FI error*	
6	*Application warning*	TRUE:	An application warning is generated if the motor speed is close to the maximum motor speed (90% to 100%).	(→ 🗎 29)
		See:	O5^{r} - O6^{r} *ErrorID*	
7	*Application fault*	TRUE:	HandlingKinematics has an error status.	(→ 🗎 100)
		See:	O5^{r} - O6^{r} *ErrorID*	

24796131/EN – 12/2017

O1^r *Status word*			
Bit	**Name**	**Description**	**See chapter**
8	*At least one FI is in "STO" state*	TRUE: At least one inverter is in "STO" state = "SafeTorqueOff" (*InverterData.SafeStop1/SafeStop2* on the *AxisInterface*).	Manual for the "MultiMotion" program module[1]
9	*All FIs are connected*	TRUE: The communication between controller and inverter has been established. See: O20^r – 23^r:5 *FI connected*	
10	*All output stages are enabled*	TRUE: The inverter output stages are enabled and deliver output voltage. See: O20^r – 23^r:3 *Output stage enabled*	
11 – 13	--		–
14	*Speed limit exceeded*	TRUE: The motor speed limit was exceeded.	(→ 🗎 29)
15	*Work envelope left*	TRUE: The work envelope of the kinematic model was left.	(→ 🗎 29)

1) "MultiMotion Program Module Manual – Universal, Parameterizable Software Platform for MOVI-PLC."

O2^r *Referencing / operating mode*			
Bit	**Name**	**Description**	**See chapter**
0	*Axis-by-axis jog*	TRUE: The active operating mode[1] is *Axis-by-axis jog*.	(→ 🗎 44)
1	*Cartesian jog*	TRUE: The active operating mode[1] is *Cartesian jog*.	
2	*Referencing*	TRUE: The active operating mode[1] is *Referencing*.	(→ 🗎 45)
3	*Program Auto*	TRUE: The active operating mode[1] is *Program auto*.	(→ 🗎 46)
4	*Program Step*	TRUE: The active operating mode[1] is *Program step*.	
5 – 7	--		–
8	*Axis 1 referenced*	TRUE: The respective axis is referenced. See: O20^r – 23^r:3 *Drive referenced*	(→ 🗎 45)
9	*Axis 2 referenced*		
10	*Axis 3 referenced*		
11	*Axis 4 referenced*		
12	*Axis 5 referenced*		
13	*Axis 6 referenced*		
14	*Axis 7 referenced*		
15	*Axis 8 referenced*		

1) If none of the bits are TRUE, HandlingKinematics is inactive/in "Default" operating mode.

O3r *Program number / telegram number*			
Byte	**Name**	**Description**	**See chapter**
High	*Program number*	Program selected in *Program auto / Program step* operating mode.	(→ 🗎 51)
Low	*Telegram number*	Transmitted telegram number that can be used to ensure that an input signal was transmitted in the same process data telegram, e.g. I16r to output the speed.	(→ 🗎 102)

O4r *Program status*				
Bit	**Name**	**Description**		**See chapter**
0	*The program is initialized*	TRUE:	The program is initialized.	(→ 🗎 46) (→ 🗎 101)
1	*The program is being executed*	TRUE:	The program was restarted and is now run.	
2	*Program stopped*	TRUE:	The program was stopped while it was run.	
3	*Program completed*	TRUE:	The program was run completely.	
4	--			–
5	*Touchprobe Motion active*	TRUE:	*Touchprobe Motion* has been successfully activated for the current motion sequence.	(→ 🗎 61)
6	*Touchprobe Motion detected*	TRUE:	The Touchprobe signal for *Touchprobe Motion* was successfully detected in the last path segment.	
7	*Touchprobe Measure active*	TRUE:	*Touchprobe Measure* has been successfully activated for the current motion sequence.	
8	*Wait 1 active*	TRUE:	The wait point on I8rff. *Target pose 1 to 4* is active. To keep moving, the signal I4r:8ff. *Wait 1 to 4* must be set to *FALSE*.	(→ 🗎 56)
9	*Wait 2 active*			
10	*Wait 3 active*			
11	*Wait 4 active*			
12	*End 1 active*	TRUE:	The end of the motion sequence was brought forward by the signal I4r:12ff. *End 1 – 4* to the I8rff. *Target pose 1 – 4*.	(→ 🗎 56)
13	*End 2 active*			
14	*End 3 active*			
15	*End 4 active*			

O5r - O6r *ErrorID*				
PD		**Name**	**Description**	**See chapter**
O5r	Word	*ErrorID*	If an application error occurs (O1r:7 *Application error*), the number of the first error that occurred is issued. In contrast, if there is only an application warning (O1r:5 *Application warning*), the warning number is transmitted.	Down (→ 🗎 165)
O6r	Word			

24796131/EN – 12/2017

$O7^{r}$ – $O11^{r}$ *Cartesian position of KCS*				
PD		**Name**	**Description**	**See chapter**
$O7^{r}$	INT	*Position / Touchprobe Measure → X*	Current position or Cartesian position measured with the *Touchprobe Measure*[1] (X, Y, Z coordinates) in the kinematics coordinate system (KCS). You can switch between the two using the $I1^{r}$:8 *Output Touchprobe Measure* signal.	(→ 🗎 19) (→ 🗎 61)
$O8^{r}$	INT	*Position / Touchprobe Measure → Y*		
$O9^{r}$	INT	*Position / Touchprobe Measure → Z*		
$O10^{r}$	INT	*Orientation*	The current Cartesian orientation[1] A in the kinematics coordinate system (KCS) is 0 if the $I1^{r}$:8 *Output Touchprobe Measure* signal is activated.	(→ 🗎 19)
$O11^{r}$	Word	*Constellation*	Current constellation of the kinematic model[1].	(→ 🗎 28)

1) Valid only if the signal O26r:4 "Cartesian position valid" is TRUE. Otherwise, the position, orientation, and constellation must not be used in the PLC.

$O12^{r}$ – $O15^{r}$ *ACS axis values*				
PD		**Name**	**Description**	**See chapter**
$O12^{r}$	INT	*Axis value 1*	Actual positions of axes A1/A2/A3/A4 in user units.	(→ 🗎 19)
$O13^{r}$	INT	*Axis value 2*		
$O14^{r}$	INT	*Axis value 3*		
$O15^{r}$	INT	*Axis value 4*		

$O16^{r}$ – $O19^{r}$ *Max./Act. Motor speed*				
PD		**Name**	**Description**	**See chapter**
$O16^{r}$	INT	*Rotational speed of motor 1*	Maximum or actual rotational speed of the motors in min^{-1}. You can switch between the two using the $I1^{r}$:9 *Output actual motor speed* signal. The maximum motor speed (when $I1^{r}$:9 = *FALSE*) is reset to 0 when switching to a kinematic mode or when the edge of signal $I1^{r}$:6 *Error reset* increases.	(→ 🗎 29)
$O17^{r}$	INT	*Rotational speed of motor 2*		
$O18^{r}$	INT	*Rotational speed of motor 3*		
$O19^{r}$	INT	*Rotational speed of motor 4*		

O20^{r} *Status axis 1*			
Bit	**Name**	**Description**	**See chapter**
0	*Motor running*	TRUE: The motor is turning (motor speed ≠ 0), with the condition that the inverter is connected (O20^{r}ff. – O20^{r}:5 *FI connected*). (*InverterData.MotorStandstill* of *AxisInterface*)	Manual for the "MultiMotion" program module[1]
1	*Frequency inverter ready for operation*	TRUE: The inverter is ready for interpolation/for control by the controller. (*InverterData.InverterReady* and no *FBError* on the *AxisInterface*)	
2	*Drive referenced*	TRUE: The drive is now referenced. The signal is reset to FALSE for at least 2 seconds after the rising edge of the signal I2^{r}:8 to I2^{r}:11 *Start referencing axis 1 - 4*. (*InverterData.Referenced* on the *AxisInterface*)	(→ 🗎 45) and Manual for the "MultiMotion" program module[1]
3	*Output stage enabled*	TRUE: The output stage of the inverter is enabled and is supplying output voltage. (*Inverter.Powered* on the *AxisInterface*)	Manual for the "MultiMotion" program module[1]
4	*Brake released*	TRUE: The drive/axis brake is released. (*Inverter.BrakeReleased* on the *AxisInterface*).	
5	*FI connected*	TRUE: Communication between the controller and inverter is established. (*Connected* on the *AxisInterface*)	
6	*Frequency inverter error*	TRUE: Fault on the inverter. See: O20^{r}ff:High *FI status*	
7	--		–
High byte	*Frequency inverter status*	Number of the inverter status or error. (*InverterData.FaulStatus/InverterStatus* on the *AxisInterface*)	Manual for the "MultiMotion" program module[1]

1) "MultiMotion Program Module Manual – Universal, Parameterizable Software Platform for MOVI-PLC."

24796131/EN – 12/2017

O21[r] *Axis status 2*			
Bit	**Name**	**Description**	**See chapter**
0	*Motor running*	Equivalent to O20[r] *Axis status 1*, only for axis 2.	= O20[r]
1	*Frequency inverter ready for operation*		
2	*Drive referenced*		
3	*Output stage enabled*		
4	*Brake released*		
5	*FI connected*		
6	*Frequency inverter error*		
7	--		
High byte	*Frequency inverter status*		

O22[r] *Axis status 3*			
Bit	**Name**	**Description**	**See chapter**
0	*Motor running*	Equivalent to O20[r] *Axis status 1*, only for axis 3.	= O20[r]
1	*Frequency inverter ready for operation*		
2	*Drive referenced*		
3	*Output stage enabled*		
4	*Brake released*		
5	*FI connected*		
6	*Frequency inverter error*		
7	--		
High byte	*Frequency inverter status*		

O23[r] *Axis status 4*			
Bit	**Name**	**Description**	**See chapter**
0	*Motor running*	Equivalent to O20[r] *Axis status 1*, only for axis 4.	= O20[r]
1	*Frequency inverter ready for operation*		
2	*Drive referenced*		
3	*Output stage enabled*		
4	*Brake released*		
5	*FI connected*		
6	*Frequency inverter error*		
7	--		
High byte	*Frequency inverter status*		

24796131/EN – 12/2017

O24^r *Axis status 5*			
Bit	**Name**	**Description**	**See chapter**
0	*Motor running*	Equivalent to O20^r *Axis status 1*, only for axis 5. It is active only for overdetermined kinematic models.	= O20^r
1	*Frequency inverter ready for operation*		
2	*Drive referenced*		
3	*Output stage enabled*		
4	*Brake released*		
5	*FI connected*		
6	*Frequency inverter error*		
7	--		
High byte	*Frequency inverter status*		

O25^r *Axis status 6*			
Bit	**Name**	**Description**	**See chapter**
0	*Motor running*	Equivalent to O20^r *Axis status 1*, only for axis 6. It is active only for overdetermined kinematic models.	= O20^r
1	*Frequency inverter ready for operation*		
2	*Drive referenced*		
3	*Output stage enabled*		
4	*Brake released*		
5	*FI connected*		
6	*Frequency inverter error*		
7	--		
High byte	*Frequency inverter status*		

O26^r *Kinematic model*			
Bit	**Name**	**Description**	**See chapter**
0	*Current motion task executed*	TRUE: The current motion command is terminated. That is, the current path segment has been traveled. This may be due to the two following causes: • When O4^r:3 *Program complete* = *FALSE*: The kinematic model has stopped at a wait point. • When O4^r:3 *Program complete* = *TRUE*: The kinematic model has traveled the motion sequence fully and the SRL program is finished.	(→ 🗎 56)
1	*CP path end reached*	Only for program 3 and 4: TRUE: The path end of the path interpolation (CP) is reached.	(→ 🗎 56)

O26ʳ *Kinematic model*			
Bit	**Name**	**Description**	**See chapter**
2	*Repositioning for CP path active*	Only for program 3 and 4: TRUE: The path has been left and a repositioning is to be carried out or is being carried out (BackToPath). When the repositioning is completed, the signal changes back to FALSE.	(→ 🗎 58)
3	*Predefined CP movement slowed down*	Only for program 3 and 4: TRUE: The look ahead function has automatically decelerated the specified movement, to not breach any speed or acceleration limits. The signal is reset to FALSE when a motion path is set that must not be decelerated.	(→ 🗎 60)
4	*Cartesian position is valid*	TRUE: The output *position, orientation*, and *constellation* in O7ʳ to 11ʳ are valid. The kinematics configuration is complete and correctly calculates the transformations. When this bit is set to *FALSE*, position, orientation, and constellation must not be used in the PLC.	(→ 🗎 100)
5 – 15	--		–

24796131/EN – 12/2017

O27^{r} – O32^{r} *Kinematic model*

PD		Name	Description	See chapter
O27^{r}	Word	*Kinematic model operating mode*	Operating mode for all of the axes of the lower-level Kinematics technology module. Depending on the selected HandlingKinematics mode: • **Default**: 0 = KIN_AM_DEFAULT • **Axis-by-axis jog mode**: 110 = KIN_JOG_AXIS • **Cartesian jog mode**: 111 = KIN_JOG_CART • **Referencing**: 2 = KIN_AM_HOMING • **Program auto/step** Program 1: 120 = KIN_TARGET_AXIS • **Program auto/step** Program 2: 121=KIN_TARGET_CART • **Program auto/step** Program 3 and 4: – 213 = KIN_LIN_3D – 300 = KIN_CP_BLENDING – KIN_TARGET_AXIS (only for BackToPath) • **Program auto/step** TouchprobeMotion: – 121 = KIN_TARGET_CART Other operating modes exist, but are not relevant for HandlingKinematics. You can find detailed information in the "Interpolating operating modes" chapter of the "Kinematics Technology Module for MultiMotion/MultiMotion Light" manual.	(→ 🗎 30)
O28^{r}	Word	*Translational speed*	Effective translational speed of the tool center point (TCP) that contains the speed in direction X, Y, and Z.	–
O29	Word	--		–
O30	Word	--		–
O31^{r}	Word	*CP path: Target distance*	Only for program 3 and 4: Remaining segment on the path up to the *target pose* at which the next wait point is active or the path ends.	(→ 🗎 30) (→ 🗎 53) (→ 🗎 56)
O32^{r}	High byte	*Current segment ID*	Current path segment being traveled by the kinematic model.	
	Low byte	*Remaining CP segments*	Only for program 3 and 4: Number of CP path segments until the *target pose* at which the next wait point is active or the path ends.	(→ 🗎 31)

10.3 Fieldbus input data: Profile 2 with 60 PD

The fieldbus input data must be transmitted consistently; see the "Overview: Profile 2 with 60 PD" (→ 🗎 39) chapter.

PD	Name	Consistency blocks
--	*SEW-EURODRIVE controller*	**2 PD**
$I1^{r}$	*Control word (kinematics)*	
$I2^{r}$	*Referencing/operating mode*	
$I3^{r}$	*Program no. /telegram no.*	
$I4^{r}$	*Control program*	
$I5^{r}$	*Jog (control word)*	
$I6^{r}$	*Touchprobe / override*	
$I7^{r}$	*Assignment of the segment parameter sets to the 5 segments*	
$I8^{r}$ – $I11^{r}$	*Target pose 1 → XYZA*	**32 PD**
$I12^{r}$	--	
$I13^{r}$ – $I16^{r}$	*Target pose 2 → XYZA*	
$I17^{r}$	*Blending distance to segment 2*	
$I18^{r}$ – $I21^{r}$	*Target pose 3 → XYZA*	
$I22^{r}$	*Blending distance to segment 3*	
$I23^{r}$ – $I26^{r}$	*Target pose 4 → XYZA*	
$I27^{r}$	*Blending distance to segment 4*	
$I28^{r}$ – $I31^{r}$	*Target pose 5 → XYZA*	
$I32^{r}$	*Blending distance to segment 5*	
$I33^{r}$	-- / *Telegram number*	
$I34^{r}$	*Program control*	
$I35^{r}$	*Assignment of the segment parameter sets to the 5 segments*	**32 PD**
$I36^{r}$ – $I60^{r}$	*Target poses 6 to 10 + blending distance*	
$I61^{r}$ – $I64^{r}$	--	

The following tables show the fieldbus input data from the PLC to the kinematics for field control with profile 2 with 60 process data words.

The assignment of process data words 1 to 32 is identical to profile 1. Only 28 process data words are added.

$I33^{r}$ *Telegram number*			
Byte	**Name**	**Description**	**See chapter**
High	--		–
Low	*Telegram number*	Telegram number required to ensure the consistency of the message. It must be the same as $I3^{r}$:Low *Telegram number*.	(→ 🗎 102)

24796131/EN – 12/2017

I34r *Program control*			
Bit	**Name**	**Description**	**See chapter**
0 – 5	--		–
6	*Wait 5*	TRUE: Stopping at *target pose 5 to 9* is forced. FALSE: Approval to travel the next path segments 6 to 10 is granted if the path was not already canceled by the signal I34r:11ff. *End* .	(→ 🗎 56)
7	*Wait 6*		
8	*Wait 7*		
9	*Wait 8*		
10	*Wait 9*		
11	*End 5*	TRUE: Program ends at *target poses 5 to 9*, where the first active end signal comes into effect.	(→ 🗎 56)
12	*End 6*		
13	*End 7*		
14	*End 8*		
15	*End 9*		

I35r *Assignment of the segment parameter sets to the 5 segments*			
Bit	**Name**	**Description**	**See chapter**
0	*Segment 6 bit 1/3*	Selection of the segment parameter sets for path segments 6 to 10 using the relevant 3 bits. In this case, the following conversion from decimal (minus 1) to binary applies: 1 = 000, 2 = 001, 3 = 010, 4 = 011, 5 = 100, 6 = 101, 7 = 110, 8 = 111.	(→ 🗎 34)
1	*Segment 6 bit 2/3*		
2	*Segment 6 bit 3/3*		
3	*Seg. 7 bit 1/3*		
4	*Seg. 7 bit 2/3*		
5	*Seg. 7 bit 3/3*		
6	*Segment 8 bit 1/3*		
7	*Segment 8 bit 2/3*		
8	*Segment 8 bit 3/3*		
9	*Seg. 9 bit 1/3*		
10	*Seg. 9 bit 2/3*		
11	*Seg. 9 bit 3/3*		
12	*Segment 10 bit 1/3*		
13	*Segment 10 bit 2/3*		
14	*Segment 10 bit 3/3*		
15	--		–

24796131/EN – 12/2017

$I36^r$ – $I40^r$ *segment 6*				
PD		**Name**	**Description**	**See chapter**
$I36^r$	INT	*Target position 6 → X*	Default *target pose* for path segment 6.	(→ 🗎 31)
$I37^r$	INT	*Target position 6 → Y*		
$I38^r$	INT	*Target position 6 → Z*		
$I39^r$	INT	*Target orientation 6*		
$I40^r$	Word	*Blending distance to segment 6*	Distance to *target position 5* from which blending to path segment 6 occurs.	(→ 🗎 31)

$I41^r$ – $I45^r$ *Segment 7*				
PD		**Name**	**Description**	**See chapter**
$I41^r$	INT	*Target position 7 → X*	Default *target pose* for path segment 7.	(→ 🗎 31)
$I42^r$	INT	*Target position 7 → Y*		
$I43^r$	INT	*Target position 7 → Z*		
$I44^r$	INT	*Target orientation 7*		
$I45^r$	Word	*Blending distance to segment 7*	Distance to *target position 6* from which blending to path segment 7 occurs.	(→ 🗎 31)

$I46^r$ – $I50^r$ *Segment 8*				
PD		**Name**	**Description**	**See chapter**
$I46^r$	INT	*Target position 8 → X*	Default *target pose* for path segment 8.	(→ 🗎 31)
$I47^r$	INT	*Target position 8 → Y*		
$I48^r$	INT	*Target position 8 → Z*		
$I49^r$	INT	*Target orientation 8*		
$I50^r$	Word	*Blending distance to segment 8*	Distance to *target position 7* from which blending to path segment 8 occurs.	(→ 🗎 31)

I51^r – I55^r *Segment 9*				
PD		**Name**	**Description**	**See chapter**
I51^r	INT	*Target position 9 → X*	Default *target pose* for path segment 9.	(→ 🗎 31)
I52^r	INT	*Target position 9 → Y*		
I53^r	INT	*Target position 9 → Z*		
I54^r	INT	*Target orientation 9*		
I55^r	Word	*Blending distance to segment 9*	Distance to *target position 8* from which blending to path segment 9 occurs.	(→ 🗎 31)

I56^r – I60^r *Segment 10*				
PD		**Name**	**Description**	**See chapter**
I56^r	INT	*Target position 10 → X*	Default *target pose* for path segment 10.	(→ 🗎 31)
I57^r	INT	*Target position 10 → Y*		
I58^r	INT	*Target position 10 → Z*		
I59^r	INT	*Target orientation 10*		
I60^r	Word	*Blending distance to segment 10*	Distance to *target position 9* from which blending to path segment 10 occurs.	(→ 🗎 31)

10.4 Fieldbus output data: Profile 2 with 60 PD

The following tables show the fieldbus output data from the controller to the PLC for fieldbus control with profile 2 with 60 process data words.

The assignment of process data words 1 to 32 is identical to profile 1. The only difference is that 28 process data words are added.

O33^r *Telegram number*			
Byte	**Name**	**Description**	**See chapter**
High	--		
Low	*Telegram number*	Telegram number required to ensure the consistency of the message. It must be the same as O3^r:Low *Telegram number*.	(→ 🗎 102)

O34^r *Program status*			
Bit	**Name**	**Description**	**See chapter**
0 – 5	--		–

24796131/EN – 12/2017

O34^r *Program status*			
Bit	**Name**	**Description**	**See chapter**
6	*Wait 5 active*	TRUE: The wait point on I36^{r}ff. *Target pose 5 to 9* is active. To keep moving, the signal I34^r:8ff. *Wait 5 to 9* must be set to *FALSE*.	(→ 🗎 56)
7	*Wait 6 active*		
8	*Wait 7 active*		
9	*Wait 8 active*		
10	*Wait 9 active*		
11	*End 5 active*	TRUE: The end of the motion sequence was brought forward by the signal I34^r:12ff. *End 5 – 9* to the I36^{r}ff. *Target pose 5 – 9*.	(→ 🗎 56)
12	*End 6 active*		
13	*End 7 active*		
14	*End 8 active*		
15	*End 9 active*		

O35^r – 60^r *Reserved*			
PD	**Name**	**Description**	**See chapter**
O36^r – O60^r	Word	--	–

10.5 Fieldbus input data: Profile 3 with 88 PD

The fieldbus input data must be transmitted consistently; see the "Overview: Profile 3 with 88 PD" (→ 🗎 40) chapter.

PD	Name	Consistency blocks
--	*SEW-EURODRIVE controller*	**2 PD**
$I1^r$	*Control word (kinematics)*	
$I2^r$	*Referencing/operating mode*	
$I3^r$	*Program no. /telegram no.*	
$I4^r$	*Control program*	
$I5^r$	*Jog (control word)*	
$I6^r$	*Touchprobe / override*	
$I7^r$	*Assignment of the segment parameter sets to the 5 segments*	
$I8^r$ – $I11^r$	*Target pose 1 → XYZA*	
$I12^r$	--	**32 PD**
$I13^r$ – $I16^r$	*Target pose 2 → XYZA*	
$I17^r$	*Blending distance to segment 2*	
$I18^r$ – $I21^r$	*Target pose 3 → XYZA*	
$I22^r$	*Blending distance to segment 3*	
$I23^r$ – $I26^r$	*Target pose 4 → XYZA*	
$I27^r$	*Blending distance to segment 4*	
$I28^r$ – $I31^r$	*Target pose 5 → XYZA*	
$I32^r$	*Blending distance to segment 5*	
$I33^r$	*-- / Telegram number*	
$I34^r$	*Program control*	
$I35^r$	*Assignment of the segment parameter sets to the 5 segments*	**24 PD**
$I36^r$ – $I60^r$	*Target poses 6 to 10 + blending distance*	
$I61^r$	*-- / Telegram number*	
$I62^r$	*Program control*	**32 PD**
$I63^r$	*Assignment of the segment parameter sets to the 5 segments*	
$I64^r$ – $I88^r$	*Target poses 11 to 15 + blending distance*	

The following tables show the fieldbus input data from the PLC to the kinematics for field control with profile 3 with 88 process data words.

The assignment of process data words 1 to 32 is identical to profile 1.

24796131/EN – 12/2017

The assignment of process data words 33 to 60 is identical to profile 2.

Only 28 process data words are added.

I61ʳ *Telegram number*			
Byte	**Name**	**Description**	**See chapter**
High	--		
Low	*Telegram number*	Telegram number required to ensure the consistency of the message. It must be the same as I33ʳ:Low *Telegram number*.	(→ 🗎 102)

I62ʳ *Program control*			
Bit	**Name**	**Description**	**See chapter**
0 – 5	--		–
6	*Wait 10*	TRUE: Forced stop at *Target pose 10 – 14*. FALSE: Approval to travel the next path segments 11 to 15 is granted if the path was not already canceled by the signal I62ʳ:11ff. *End* 10 to 14.	(→ 🗎 56)
7	*Wait 11*		
8	*Wait 12*		
9	*Wait 13*		
10	*Wait 14*		
11	*End 10*	TRUE: Program ends at *target poses 10 to 14*, where the first active end signal comes into effect.	(→ 🗎 56)
12	*End 11*		
13	*End 12*		
14	*End 13*		
15	*End 14*		

I63ʳ *Assignment of the segment parameter sets to the 5 segments*			
Bit	**Name**	**Description**	**See chapter**
0	*Segment 11 bit 1/3*	Selection of the segment parameter sets for path segments 11 to 15 using the relevant 3 bits. In this case, the following conversion from decimal (minus 1) to binary applies: 1 = 000, 2 = 001, 3 = 010, 4 = 011, 5 = 100, 6 = 101, 7 = 110, 8 = 111.	(→ 🗎 34)
1	*Segment 11 bit 2/3*		
2	*Segment 11 bit 3/3*		
3	*Seg. 12 bit 1/3*		
4	*Seg. 12 bit 2/3*		
5	*Seg. 12 bit 3/3*		
6	*Segment 13 bit 1/3*		
7	*Segment 13 bit 2/3*		
8	*Segment 13 bit 3/3*		
9	*Seg. 14 bit 1/3*		
10	*Seg. 14 bit 2/3*		
11	*Seg. 14 bit 3/3*		
12	*Segment 15 bit 1/3*		
13	*Segment 15 bit 2/3*		
14	*Segment 15 bit 3/3*		

24796131/EN – 12/2017

I63^{r} *Assignment of the segment parameter sets to the 5 segments*			
Bit	**Name**	**Description**	**See chapter**
15	--		–

I64^{r} – I68^{r} segment 11				
PD		**Name**	**Description**	**See chapter**
I64^{r}	INT	*Target position 11 → X*	Default *target pose* for path segment 11.	(→ 🗎 31)
I65^{r}	INT	*Target position 11 → Y*		
I66^{r}	INT	*Target position 11 → Z*		
I67^{r}	INT	*Target orientation 11*		
I68^{r}	Word	*Blending distance to segment 11*	Distance to *target position 10* from which blending to path segment 11 occurs.	(→ 🗎 31)

I69^{r} – I73^{r} segment 12				
PD		**Name**	**Description**	**See chapter**
I69^{r}	INT	*Target position 12 → X*	Default *target pose* for path segment 12.	(→ 🗎 31)
I70^{r}	INT	*Target position 12 → Y*		
I71^{r}	INT	*Target position 12 → Z*		
I72^{r}	INT	*Target orientation 12*		
I73^{r}	Word	*Blending distance to segment 12*	Distance to *target position 11* from which blending to path segment 12 occurs.	(→ 🗎 31)

I74^{r} – I78^{r} segment 13				
PD		**Name**	**Description**	**See chapter**
I74^{r}	INT	*Target position 13 → X*	Default *target pose* for path segment 13.	(→ 🗎 31)
I75^{r}	INT	*Target position 13 → Y*		
I76^{r}	INT	*Target position 13 → Z*		
I77^{r}	INT	*Target orientation 13*		
I78^{r}	Word	*Blending distance to segment 13*	Distance to *target position 12* from which blending to path segment 13 occurs.	(→ 🗎 31)

24796131/EN – 12/2017

I79^{r} – I83^{r} segment 14				
PD		Name	Description	See chapter
I79^{r}	INT	*Target position 14 → X*	Default *target pose* for path segment 14.	(→ 🗎 31)
I80^{r}	INT	*Target position 14 → Y*		
I81^{r}	INT	*Target position 14 → Z*		
I82^{r}	INT	*Target orientation 14*		
I83^{r}	Word	*Blending distance to segment 14*	Distance to *target position 13* from which blending to path segment 14 occurs.	(→ 🗎 31)

I84^{r} – I88^{r} segment 15				
PD		Name	Description	See chapter
I84^{r}	INT	*Target position 15 → X*	Default *target pose* for path segment 15.	(→ 🗎 31)
I85^{r}	INT	*Target position 15 → Y*		
I86^{r}	INT	*Target position 15 → Z*		
I87^{r}	INT	*Target orientation 15*		
I88^{r}	Word	*Blending distance to segment 15*	Distance to *target position 14* from which blending to path segment 15 occurs.	(→ 🗎 31)

10.6 Fieldbus output data: Profile 3 with 88 PD

The following tables show the fieldbus output data from the controller to the PLC for fieldbus control with profile 3 with 88 process data words.

The assignment of process data words 1 to 32 is identical to profile 1.

The assignment of process data words 33 to 60 is identical to profile 2.

Only 28 process data words are added.

O61^{r} *Telegram number*			
Byte	Name	Description	See chapter
High	--		–
Low	*Telegram number*	Telegram number required to ensure the consistency of the message. It must be the same as O33^{r}:Low *Telegram number*.	(→ 🗎 102)

O62^{r} *Program status*			
Bit	Name	Description	See chapter
0 – 5	--		–

O62^r *Program status*			
Bit	**Name**	**Description**	**See chapter**
6	*Wait 10 active*	TRUE: The wait point on the I64^{r}ff. *Target pose 10 to 14* is active. To keep moving, the signal I62^r:8ff. *Wait 10 to 14* must be set to *FALSE*.	(→ 🗎 56)
7	*Wait 11 active*		
8	*Wait 12 active*		
9	*Wait 13 active*		
10	*Wait 14 active*		
11	*End 10 active*	TRUE: The end of the motion sequence was brought forward by the signal I62^r:12ff. *End 10 – 14* to the I64^{r}ff. *Target pose 10 – 14*.	(→ 🗎 56)
12	*End 11 active*		
13	*End 12 active*		
14	*End 13 active*		
15	*End 14 active*		

O63^r – O88^r *Reserved*			
PD	**Name**	**Description**	**See chapter**
O63^r – O88^r	Word	--	–

10.7 Fieldbus input data: Profile 4 with 116 PD

The fieldbus input data must be transmitted consistently; see the "Overview: Profile 4 with 116 PD" (→ 🗎 41) chapter.

PD	Name	Consistency blocks
--	*SEW-EURODRIVE controller*	**2 PD**

PD	Name	Consistency blocks
I1^r	*Control word (kinematics)*	32 PD
I2^r	*Referencing/operating mode*	
I3^r	*Program no. /telegram no.*	
I4^r	*Control program*	
I5^r	*Jog (control word)*	
I6^r	*Touchprobe / override*	
I7^r	*Assignment of the segment parameter sets to the 5 segments*	
I8^r – I11^r	*Target pose 1 → XYZA*	
I12^r	--	
I13^r – I16^r	*Target pose 2 → XYZA*	
I17^r	*Blending distance to segment 2*	
I18^r – I21^r	*Target pose 3 → XYZA*	
I22^r	*Blending distance to segment 3*	
I23^r – I26^r	*Target pose 4 → XYZA*	
I27^r	*Blending distance to segment 4*	
I28^r – I31^r	*Target pose 5 → XYZA*	
I32^r	*Blending distance to segment 5*	
I33^r	*-- / Telegram number*	24 PD
I34^r	*Program control*	
I35^r	*Assignment of the segment parameter sets to the 5 segments*	
I36^r – I60^r	*Target poses 6 to 10 + blending distance*	
I61^r	*-- / Telegram number*	32 PD
I62^r	*Program control*	
I63^r	*Assignment of the segment parameter sets to the 5 segments*	
I64^r – I88^r	*Target poses 11 to 15 + blending distance*	
I89^r	*-- / Telegram number*	24 PD
I90^r	*Program control*	
I91^r	*Assignment of the segment parameter sets to the 5 segments*	
I92^r – I112^r	*Target poses 16 to 20 + blending distance*	
I113^r – I116^r		4 PD

24796131/EN – 12/2017

The following tables show the fieldbus input data from the PLC to the kinematics for field control with profile 3 with 88 process data words.

The assignment of process data words 1 to 32 is identical to profile 1.

The assignment of process data words 33 to 60 is identical to profile 2.

The assignment of process data words 61 to 88 is identical to profile 3.

Only 28 process data words are added.

$I89^{r}$ -- / telegram number			
Byte	**Name**	**Description**	**See chapter**
High	--		
Low	*Telegram number*	Telegram number required to ensure the consistency of the message. It must be the same as $I61^{r}$:Low *Telegram number*.	(→ 🗎 102)

$I90^{r}$ *Program control*				
Bit	**Name**	**Description**		**See chapter**
0 – 5	--			–
6	*Wait 15*	TRUE:	Stopping at *target poses 15 to 19* is forced.	(→ 🗎 56)
7	*Wait 16*			
8	*Wait 17*	FALSE:	Approval to travel the next path segments 16 to 20 is granted if the path was not already ended by the signal $I90^{r}$:11 to 15 *End*.	
9	*Wait 18*			
10	*Wait 19*			
11	*End 15*	TRUE:	Program ends at *target poses 15 to 19*, where the first active end signal comes into effect.	(→ 🗎 56)
12	*End 16*			
13	*End 17*			
14	*End 18*			
15	*End 19*			

24796131/EN – 12/2017

I91^r Assignment of the segment parameter sets to the 5 segments			
Bit	**Name**	**Description**	**See chapter**
0	*Segment 16 Bit 1/3*	Selection of the segment parameter sets for path segments 16 to 20 using the relevant 3 bits. In this case, the following conversion from decimal (minus 1) to binary applies: 1 = 000, 2 = 001, 3 = 010, 4 = 011, 5 = 100, 6 = 101, 7 = 110, 8 = 111.	(→ 🗎 34)
1	*Segment 16 bit 2/3*		
2	*Segment 16 bit 3/3*		
3	*Seg. 17 bit 1/3*		
4	*Seg. 17 bit 2/3*		
5	*Seg. 17 bit 3/3*		
6	*Segment 18 bit 1/3*		
7	*Segment 18 bit 2/3*		
8	*Segment 18 bit 3/3*		
9	*Seg. 19 bit 1/3*		
10	*Seg. 19 bit 2/3*		
11	*Seg. 19 bit 3/3*		
12	*Segment 20 bit 1/3*		
13	*Segment 20 bit 2/3*		
14	*Segment 20 bit 3/3*		
15	--		–

I92^r – I96^r segment 16				
PD		**Name**	**Description**	**See chapter**
I92^r	INT	*Target position 16 → X*	Default *target pose* for path segment 16.	(→ 🗎 31)
I93^r	INT	*Target position 16 → Y*		
I94^r	INT	*Target position 16 → Z*		
I95^r	INT	*Target orientation 16*		
I96^r	Word	*Blending distance to segment 16*	Distance to *target position 15* from which blending to path segment 16 occurs.	(→ 🗎 31)

I97^r – I101^r segment 17				
PD		**Name**	**Description**	**See chapter**
I97^r	INT	*Target position 17 → X*	Default *target pose* for path segment 17.	(→ 🗎 31)
I98^r	INT	*Target position 17 → Y*		
I99^r	INT	*Target position 17 → Z*		
I100^r	INT	*Target orientation 17*		

I97^{r} – I101^{r} segment 17				
PD		**Name**	**Description**	**See chapter**
I101^{r}	Word	*Blending distance to segment 17*	Distance to *target position 16* from which blending to path segment 17 occurs.	(→ 🗎 31)

I102^{r} – I106^{r} segment 18				
PD		**Name**	**Description**	**See chapter**
I102^{r}	INT	*Target position 18 → X*	Default *target pose* for path segment 18.	(→ 🗎 31)
I103^{r}	INT	*Target position 18 → Y*		
I104^{r}	INT	*Target position 18 → Z*		
I105^{r}	INT	*Target orientation 18*		
I106^{r}	Word	*Blending distance to segment 18*	Distance to *target position 17* from which blending to path segment 18 occurs.	(→ 🗎 31)

I107^{r} – I111^{r} segment 19				
PD		**Name**	**Description**	**See chapter**
I107^{r}	INT	*Target position 19 → X*	Default *target pose* for path segment 19.	(→ 🗎 31)
I108^{r}	INT	*Target position 19 → Y*		
I109^{r}	INT	*Target position 19 → Z*		
I110^{r}	INT	*Target orientation 19*		
I111^{r}	Word	*Blending distance to segment 19*	Distance to *target position 18* from which blending to path segment 19 occurs.	(→ 🗎 31)

I112^{r} – I116^{r} segment 20				
PD		**Name**	**Description**	**See chapter**
I112^{r}	INT	*Target position 20 → X*	Default *target pose* for path segment 20.	(→ 🗎 31)
I113^{r}	INT	*Target position 20 → Y*		
I114^{r}	INT	*Target position 20 → Z*		
I115^{r}	INT	*Target orientation 20*		
I116^{r}	Word	*Blending distance to segment 20*	Distance to *target position 19* from which blending to path segment 20 occurs.	(→ 🗎 31)

24796131/EN – 12/2017

10.8 Fieldbus output data: Profile 4 with 116 PD

The following tables show the fieldbus output data from the controller to the PLC for fieldbus control with profile 3 with 88 process data words.

The assignment of process data words 1 to 32 is identical to profile 1.

The assignment of process data words 33 to 60 is identical to profile 2.

The assignment of process data words 61 to 88 is identical to profile 3.

Only 28 process data words are added.

O89^{r} *Telegram number*			
Byte	**Name**	**Description**	**See chapter**
High	--		
Low	*Telegram number*	Telegram number required to ensure the consistency of the message. It must be the same as O61^{r}:Low *Telegram number*.	(→ 🗎 102)

O90^{r} *Program status*				
Bit	**Name**	**Description**		**See chapter**
0 – 5	--			–
6	*Wait 15 active*	TRUE:	The wait point on I92^{r}ff. *Target pose 15 to 19* is active. To keep moving, the signal I90^{r}:8ff. *Wait 15 to 19* must be set to *FALSE*.	(→ 🗎 56)
7	*Wait 16 active*			
8	*Wait 17 active*			
9	*Wait 18 active*			
10	*Wait 19 active*			
11	*End 15 active*	TRUE:	The end of the motion sequence was brought forward by the signal I90^{r}:12ff. *End 15 – 19* to the I92^{r}ff. *Target pose 15 – 19*.	(→ 🗎 56)
12	*End 16 active*			
13	*End 17 active*			
14	*End 18 active*			
15	*End 19 active*			

O91^{r} – O116^{r} *Reserved*			
PD		**Description**	**See chapter**
O91^{r} – O116^{r}	Word	--	–

11 Problem management

11.1 System bus CAN 1/CAN 2

Problem

Communication via system bus CAN 1 or CAN 2 is not working.

Remedy

Perform the following steps one after the other until the problem is corrected.

Initial status

- System bus CAN 1 or CAN 2 is connected correctly.
- Communication via the CAN 1 or CAN 2 system bus was started up correctly for the controller in the Application Configurator and for the axis in the drive startup for MOVI-PLC®/CCU.

↓

Bus connector connected? — No → **[A]**

Yes

↓

How does the LED of the CAN 1 or CAN 2 state respond? — Orange → **[B]**; Green → **[C]**

Lights up or flashes red?

↓

The system bus CAN 1 or CAN 2 is off or the system bus communication is faulty.

↓

[A]

↓

Check the set baud rate.

↓

Baud rate OK? — No → **[D]**

Yes

↓

Ensure that terminating resistors are connected correctly.

[A]	Check the bus cabling.
[B]	System bus CAN 1 or CAN 2 is being initialized.
[C]	Bus communication is OK.

↓

Check the configuration of the axes in the Application Configurator.

[D]	Correct the baud rate.

24796131/EN – 12/2017

11.2 3D simulation

11.2.1 Problem: No connection established

Problem

You cannot establish a connection. A red "0" is displayed in the bottom left corner of the simulation window.

Remedy

Carry out the following steps one after the other until the problem is solved.

Check the communication settings:

- Establish an Ethernet connection between the controller and the simulation PC. Communication with the 3D simulation takes place exclusively via the Ethernet engineering interface (X37).

INFORMATION

3D simulation is not possible via USB or fieldbus.

- Ensure that HandlingKinematics determines the IP address of the connected simulation PC automatically and that this function has not been deactivated.
  – For this purpose, open the extended configuration, see chapter "Extended configuration or higher permission level" (→ 🗎 96).
  – Click the [Next] button until you reach the "Various settings" window with the 3D simulation settings.
  – Find the check box "Determine IP automatically" which is activated as standard for a new configuration.
  – Activate the check box "Determine IP automatically". The IP address is determined automatically with this setting.
- Make sure that no other PC running the 3D simulation is in the same subnetwork. Otherwise, it cannot be ensured that the controller connects with the correct PC. If it is not possible that 3D simulation is only open on your PC, reset to manual configuration of the IP address.
  – For this purpose, open the extended configuration, see chapter "Extended configuration or higher permission level" (→ 🗎 96).
  – Click the [Next] button until you reach the "Various settings" window with the 3D simulation settings.
  – Deactivate the check box "Determine IP automatically".
  – Enter the required IP address in the edit box.
- If an entered IP address is already used, check if the address or the IP address corresponds to the Ethernet interface of the simulation PC.

24796131/EN – 12/2017

INFORMATION

You can read the IP address set for the network adapter used in MOVITOOLS® MotionStudio:

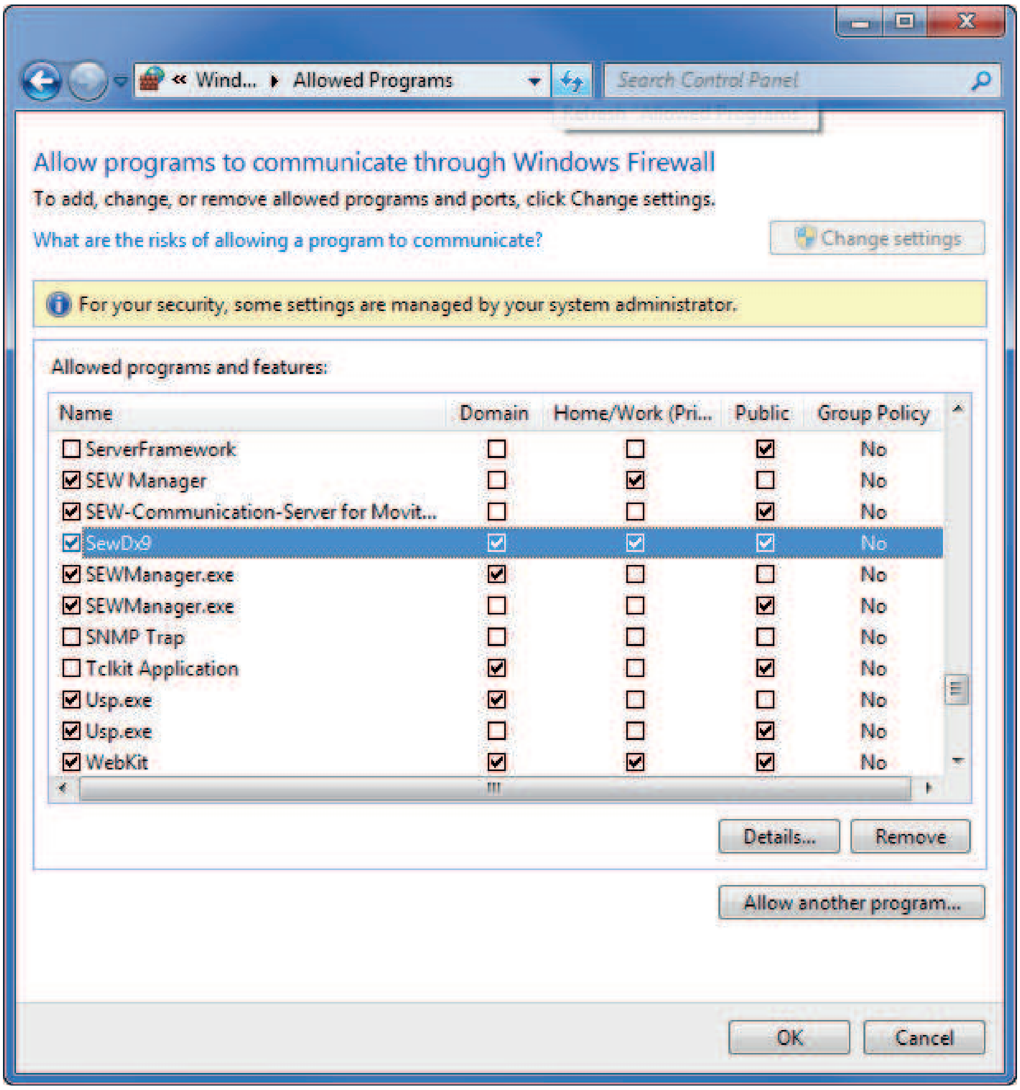
- From the menu, choose [Network] > [Configure communication connections].
- Choose "Ethernet", click the [Edit] button and then [Network adapter]. A window opens displaying the properties (IP address etc.) of the available network adapters.

Checking the firewall settings

The firewall of the PC can be another reason why there is no communication between the controller and the 3D simulation. There are 2 possibilities:

- Deactivate the firewall completely.
- Create an exception for the 3D simulation in the firewall:
  - Click to open the Windows start menu.
  - Click [Control Panel]. The initial screen of the control panel opens.
  - Select the "System and Security" section.
  - Under "Windows Firewall", select the menu command "Allow a program through Windows Firewall".
  - Choose "SEWDx9" from the list.
  - Activate the check boxes "Domain", "Home/Work (Private)", and "Public".

24796131/EN – 12/2017



14319257227

INFORMATION

If the entry "SewDx9" does not appear in the list:

- Choose the exe file from the following path: `C:\Program Files (x86)\SEW\MotionStudio\SewDx9.exe`
- Activate the check boxes "Domain", "Home/Work (Private)", and "Public".

INFORMATION

If the entry “SewDx9” appears multiple times in the list, the check boxes for all entries should be activated.

11.2.2 Problem: TEST TIME EXPIRED!

Problem

In the 3D simulation, the text "TEST TIME EXPIRED!" is displayed in red.

24796131/EN – 12/2017

Remedy

The technology points of the controller are not sufficient.

- Make sure that there are enough technology points on the memory card. For the number of technology points necessary for the configured functionality, refer to chapter "Technology points" (→ 🗎 15) or the Kinematics Configurator, see chapter "Functionality" (→ 🗎 74).

11.2.3 Problem: No model

Problem

You can establish a connection (indicated by a green number in the bottom left corner of the simulation window) but no model appears.

Remedy

The technology level of the controller is not sufficient.

- Make sure that there are enough technology points on the memory card. For the number of technology points necessary for the configured functionality, refer to chapter "Technology points" (→ 🗎 15) or the Kinematics Configurator, see chapter "Functionality" (→ 🗎 74).
- Ensure that the kinematic axes are referenced and that the kinematic configuration can be successfully executed. Possible error messages with information on the causes of error appear in the MessageHandler (in the context menu of the controller in the device tree).

11.3 Load precontrol

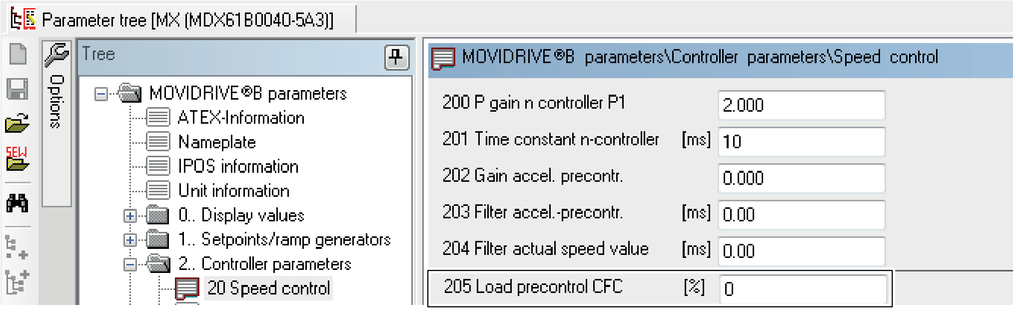
Problem

When the inverter is enabled after an emergency stop or inverter controller inhibit, the axes sink by one piece.

Remedy

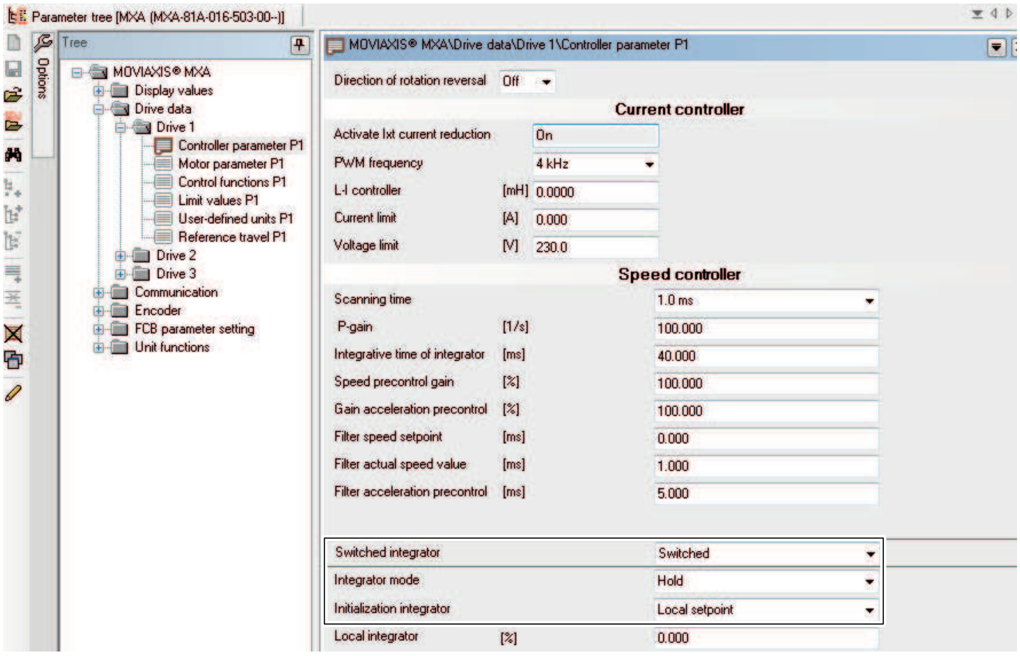
To prevent an axis with a load from "sinking" after a controller inhibit or Safe Torque Off (STO) on the inverter, you can configure a load precontrol for the inverter.

- Parameterize the load precontrol in the MOVITOOLS® MotionStudio parameter tree for the relevant device.
  - MOVIDRIVE®: In the parameter *P205*, set a proportion of the nominal current to be used for the precontrol. You can find detailed information about this in the description of the parameters in the MOVIDRIVE® MDX60B/61B system manual or in the MOVITOOLS® MotionStudio online help (select a parameter and press the <F1> key).



14216243723

– MOVIAXIS®: Set the parameters *10058, 9994*, and *9995*. You can find detailed information about this in the description of the parameters in the MOVIAXIS® manual or in the MOVITOOLS® MotionStudio online help (select a parameter and press the <F1> key).



14216249739

11.4 Cycle time

Problem

The cycle time to be reached in the application is not achieved or is irregular.

Remedy

The following steps allow you to reduce the cycle time and make the cycle time easier to reproduce:

- Activate the wait signal or end signal in a path point if the next path segment points in the opposite direction to the last. This also ensures that this path point is reached; see the "Information about the path design" (→ 🗎 115) chapter.
- Use the end signal to signal the end of a motion sequence instead of assigning the last path points with identical poses; see the "End signals" (→ 🗎 56) chapter.

24796131/EN – 12/2017

- Decrease the jerk time or increase the acceleration/deceleration in the configuration. You should also do this for parameters that are assumed not to be used in the current operating mode because your setting may affect the resulting motion parameters due to the synchronization and look-ahead; see the "Configuring the motion parameters" (→ 🗎 88) chapter.
- Use signal O26r:3 *Predefined CP motion slowed down* to check whether the path movement was adjusted by the look-ahead function. You can find more detailed information in the MessageHandler based on the relevant setting; see the "Look-ahead" (→ 🗎 60) and "MessageHandler" (→ 🗎 134) chapters.
- Increase the blending distance if the interfering contours permit it; see the "Blending" (→ 🗎 57) chapter.
- Check whether *program 1*: *TARGET_AXIS* or *program 2: TARGET_CART* can be used and whether they are executed more quickly with the specific selected parameterization than *program 3*: *LINEAR coordinated* or *program 4: LINEAR synchronized*; see the "Programs in program mode" (→ 🗎 51) chapter.
- If all of the steps listed above are still insufficient, you can increase the setting of *In.Cp.Path.FidelityPercentage* in the expert configuration up to the maximum value of 100. This setting can be accessed with the "Advanced" MotionStudio permission level or higher or in the advanced configuration; see the "Advanced configuration or higher permission level chapter" (→ 🗎 96). However, the increase in the *FidelityPercentage* leads to increased jerk.
- If all of the steps listed above are still insufficient, you can reduce the minimum amount of time required for each path segment by a factor of 3 at minimum by using the HandlingKinematics technology module with the MOVIPLC® power; see the "Programs in program mode" (→ 🗎 51) chapter.

11.5 Functional limit

11.5.1 Problem: Additional functions required (HandlingKinematics application module)

Problem

The function scope of the HandlingKinematics application module is insufficient for your application. Some of the situations in which this may arise include:

- The level of precision for the position specification (WORD) is insufficient. The application requires DWORD, for example.
- The 4 Cartesian degrees of freedom XYZA are insufficient. The application requires a kinematic model with rotatory degrees of freedom B and C.
- Additional process data must be transmitted to the PLC via the fieldbus.
- Management of recipe data and path planning on the MOVIPLC®.
- Connection of additional sensors such as a camera and actuators or pneumatic axes to the MOVIPLC®.
- Use of the SBus^{PLUS} system bus, for example, due to a long cable length between the controller and inverters.
- The application requires the "MultiMotion" program module for additional axes instead of "MultiMotion Light".
- Control of up to 64 axes or multiple kinematic instances by one controller.
- etc.

Remedy

Go to the HandlingKinematics technology module for the programmable MOVIPLC®.

11.5.2 Problem: Additional functions required (HandlingKinematics technology module)

Problem

The function scope of the "HandlingKinematics" technology module for the programmable MOVI-PLC® is insufficient for your application. Some of the situations in which this may arise include:

- Moved workpieces
- Circular interpolation
- Use of various coordinate systems and transformations
- Master-slave relations, for example, for moving a kinematic model along a CAM profile
- Basic G-code import for motion control along the CAD contour
- etc.

Remedy

Go to the "Kinematics" technology module for the programmable MOVI-PLC®.

24796131/EN – 12/2017

12 Kinematics error codes

12.1 Error: HandlingKinematics, general

Code	Error	Meaning
FDC000	E_HANDLINGKIN_MODE_NOT_VALID	The selected operating mode is not valid.
FDC100	E_HANDLINGKIN_SEG_PAR_RE-CORD_NUMBER_NOT_VALID	The selected segment parameter set is not valid.
FDC110	E_HANDLINGKIN_WRITING_OF_APPLICA-TION_MODULE_VERSION_NOT_POSSIBLE	The application module version could not be written to.
FDC120	E_HANDLINGKIN_PROGRAMNUM-BER_NOT_VALID	The selected *program number* is not valid.
FDC130	E_HANDLINGKIN_TELEGRAM_NUM-BER_NOT_EQUAL	The *telegram number* of the consistent blocks is not equal.
FDC140	E_HANDLINGKIN_PATH_SWLS	During the LookAhead function for the path, a violation of the work envelope was detected. The software limit switch (SWLS) is violated.

12.2 Error: Kinematics, general

Code	Error	Meaning
FD0000	E_KIN_GENERAL_INTERNAL_ERROR	Internal error, consult SEW Service.
FD0008	E_KIN_GENERAL_LICENSE_MODE	No license points consumed for the execution of the requested operating mode were con-sumed during configuration.
FD0010	E_KIN_GENERAL_MODE	The requested operating mode is not permitted in this case.
FD0020	E_KIN_GENERAL_DIRKIN	Error while executing the direct kinematic trans-formation.
FD0030	E_KIN_GENERAL_INVKIN	Error while executing the inverse kinematic transformation.
FD0038	E_KIN_GENERAL_AXIS_COUPLING	Error while coupling several kinematics axis values to motor positions.
FD0040	E_KIN_GENERAL_WORKSPACE_NO_AUTO-GOON	Leaving the approved work envelope; details in MessageHandler as well as in OUT variables.
FD0050	E_KIN_GENERAL_MOTOR-SPEED_LIMIT_NO_AUTOGOON	Exceeding the permissible motor speed, details in MessageHandler as well as in OUT variables
FD0060	E_KIN_GENERAL_AXISINCRE-MENTS_TOOLARGE_NEG	Axis increments near the negative 32-bit limit value.
FD0070	E_KIN_GENERAL_AXISINCRE-MENTS_TOOLARGE_POS	Axis increments near the positive 32-bit limit value.
FD0080	E_KIN_GENERAL_MESSAGEBUFFER_COR-RUPT	MessageBuffer damaged due to unauthorized overwriting.
FD0090	E_KIN_GENERAL_MESSAGEBUFFER_FULL	MessageBuffer is completely filled by the cur-rent error series.

24796131/EN – 12/2017

Code	Error	Meaning
FD00A0	E_KIN_GENERAL_ABC_STEP_AT_COORD-SYS_CHANGE	Change of the *ABC orientation values* while switching between coordinate systems due to an ABC singularity in the position.
FD00B0	E_KIN_GEN-ERAL_ABC_STEP_AT_JOG_CART	Changed *ABC orientation values* in the KIN_JOG_CART operating mode due to ABC singularity in the position.
FD00C0	E_KIN_GENERAL_ABC_STEP_AT_TAR-GET_CART	Change of the *ABC orientation values* in KIN_TARGET_CART operating mode due to an ABC singularity in the position.
FD0100	E_KIN_GENERAL_TEST_ENABLE	Function is not permitted without test activation.
FD0110	E_KIN_GENERAL_AXISCARTBIJEC-TION_OUT_OF_AXIS_SWLS	The unique assignment of axis values and Cartesian coordinates is violated because the *Axis.SWLS* was left. Move the axes to the per-mitted work envelope using KIN_JOG/TAR-GET_AXIS.
FD0120	E_KIN_GENERAL_AXISCARTBIJEC-TION_TRANSFORM	The unique assignment of axis values and Cartesian coordinates is violated due to an un-authorized transformation.

12.3 Error: Configuration

Code	Error	Meaning
FD1000	E_KIN_CONFIG_INTERNAL_ERROR	Internal error, consult SEW Service.
FD1008	E_KIN_CONFIG_NO_CONFIG	The read kinematic configuration is incorrect.
FD1010	E_KIN_CONFIG_IN_PROCESS	A reconfiguration trigger is not permitted while a kinematic configuration is being adopted.
FD1020	E_KIN_CONFIG_CYCLE_TIME	The configured cycle time is ≤ 0 ms.
FD1030	E_KIN_CONFIG_LICENSE_GET	The number of license points corresponding to the configured functions cannot be consumed because the memory card does not have enough license points, or because another technology function has already consumed li-cense points so there are not enough remain-ing license points.
FD1040	E_KIN_CONFIG_LICENSE_CORRUPT	The license check is damaged.
FD1050	E_KIN_CONFIG_LICENSE_KINTYPE	The licensing of the selected kinematics has not been requested in the configuration.
FD1060	E_KIN_CONFIG_LICENSE_COORDSYS	The licensing of the use of WCS, PCS1, and PCS2 has not been requested in the configura-tion.
FD1070	E_KIN_CONFIG_KINTYPE	The configured kinematics is not supported.
FD1080	E_KIN_CONFIG_DIRKIN	Error while executing the direct kinematic trans-formation while the configuration was being ad-opted.

24796131/EN – 12/2017

Code	Error	Meaning
FD1088	E_KIN_CONFIG_AXIS_DECOUPLING	Error while decoupling several motor positions for the kinematics axis values.
FD1090	E_KIN_CONFIG_KINPAR_USE	Incorrect configuration of the use of parameters, e.g. KIN_USE_DEGR for a linear axis.
FD10A0	E_KIN_CONFIG_KINPAR_OUT_OF_RANGE	Incorrect parameter values, e.g. negative arm length.
FD10B0	E_KIN_CONFIG_KINLIMIT_USE	Incorrect setting for the use of a kinematic limitation, e.g. KIN_USE_NONE for an existing limitation in the selected kinematic type.
FD10C0	E_KIN_CONFIG_KINLIMIT_MINMAX	Incorrect setting for the software limit switch of the kinematic limitations, e.g. SWLS_Neg > SWLS_Pos.
FD10F0	E_KIN_CONFIG_AXIS_USE	Incorrect configuration of the use of an axis, e.g. AxisUse = KIN_USE_NONE for an existing axis in the selected kinematic type.
FD1100	E_KIN_CONFIG_AXISPOS_MINMAX	Incorrect setting for the software limit switch for an axis, e.g. SWLS_Neg > SWLS_Pos.
FD1130	E_KIN_CONFIG_AXIS_JOGVEL100	Incorrect setting for AxisJogVel100Percent ≤ 0
FD1138	E_KIN_CONFIG_AXIS_JOGACCDEC	Incorrect setting for AxisJogAccDec ≤ 0.
FD1140	E_KIN_CONFIG_AXIS_RAPIDDEC	Incorrect setting for AxisRapidDeceleration ≤ 0.
FD1150	E_KIN_CONFIG_AXIS_RAPIDJERK	Incorrect setting for AxisRapidJerkTime ≤ 0.
FD1160	E_KIN_CONFIG_NUMERATOR	Incorrect setting for Numerator = 0.
FD1170	E_KIN_CONFIG_DENOMINATOR	Incorrect setting for Denominator = 0.
FD1180	E_KIN_CONFIG_MOTORSPEED_CONVERSION	Incorrect setting for MotorSpeedConversionFactor ≤ 0.
FD1190	E_KIN_CONFIG_MOTORSPEED_MAX	Incorrect setting for MotorSpeedMaxLimit ≤ 0.
FD11A0	E_KIN_CONFIG_MOTORSPEED_WARN_PERC	Incorrect setting for MotorSpeedWarningPercentage < 1.
FD11B0	E_KIN_CONFIG_MOTORSPEED_LIMIT_PERC	Incorrect setting for MotorSpeedLimitPercentage < MotorSpeedWarningPercentage.
FD11C0	E_KIN_CONFIG_CARTUNIT_USE	Incorrect setting for CartUnitUse, e.g. KIN_USE_DEGR for a translation dimension.
FD11D0	E_KIN_CONFIG_CART_JOGLEVEL100	Incorrect setting for CartJogVel100Percent ≤ 0.
FD11D8	E_KIN_CONFIG_CART_JOGACCDEC	Incorrect setting for CartJogAccDec ≤ 0
FD11E0	E_KIN_CONFIG_CART_RAPIDDEC	Incorrect setting for CartRapidDeceleration ≤ 0.
FD11F0	E_KIN_CONFIG_CART_RAPIDJERK	Reserved
FD1200	E_KIN_CONFIG_CARTPOS_MINMAX	Incorrect setting for the software limit switches for the Cartesian work envelope in KCS, e.g. SWLS_Neg > SWLS_Pos.
FD1230	E_KIN_CONFIG_CART_MODULO_USE	Incorrect setting for CartModuloUse, e.g. *TRUE* for a translation dimension.

Code	Error	Meaning
FD1240	E_KIN_CONFIG_CART_MODULO_RANGE	Incorrect setting for CartModuloUnderfollow/Overflow; the range does not represent an integer multiple of 2π or 360°.
FD1250	E_KIN_CONFIG_CP_RAPIDTRANSDEC	Incorrect setting for CpRapidTransDeceleration ≤ 0.
FD1260	E_KIN_CONFIG_CP_RAPIDTRANSJERK	Reserved
FD1270	E_KIN_CONFIG_AXISCARTBIJECTION_NOT_WITH_MAPABC	AxisGroupKin.Inst[..].Config.General.AxisCartBijection and MapABCToStartupValuesAtConfig cannot be *TRUE* at the same time.
FD1280	E_KIN_CONFIG_AXISCARTBIJECTION_MODEL_NOT_SUPPORTED	AxisGroupKin.Inst[..].Config.General.AxisCartBijection is not supported for the configured kinematic model.
FD1290	E_KIN_CONFIG_AXISCARTBIJECTION_AXIS_SWLS_TOO_LARGE	AxisGroupKin.Inst[..].Config.General.AxisCartBijection is not supported because the ranges that are generated by the software limit switches of the axes are too large.
FD12A0	E_KIN_CONFIG_AXISCARTBIJECTION_CART_OFFSET_DIRREV	AxisGroupKin.Inst[..].Config.General.AxisCartBijection is not possible because Cart.Offset_KCS or DirectionReversal_KCS <> 0.
FD1238	E_KIN_CONFIG_CART_MODULO_NOT_WITH_AXISCARTBIJECTION	Cart.ModuloUse must be *FALSE* because General.AxisCartBijection is *TRUE*.

12.4 Error: General parameters

Code	Error	Meaning
FD2000	E_KIN_GENPAR_INTERNAL_ERROR	Internal error, consult SEW Service.
FD2010	E_KIN_GENPAR_OVERRIDE_CHANGE	Change of *override* during standstill.ActCoordSys = *FALSE*. The new *override* is adopted when standstill is reached.
FD2018	E_KIN_GENPAR_OVERRIDE_SMALL	Override = 0 not permitted.
FD2020	E_KIN_GENPAR_OVERRIDE_LARGE	AxisGroupKin.Inst[..].In.General.Kin.OverridePercentage > AxisGroupKin.Inst[..].Config.General.MaxOverridePercentage not permitted.
FD2030	E_KIN_GENPAR_JOGPERC_LARGE	AxisGroupKin.Inst[..].In.Jog.VelocityPercentage > AxisGroupKin.Inst[..].Config.General.MaxJogVelocityPercentage not permitted.
FD2040	E_KIN_GENPAR_COORDSYS	A change of the coordinate system is not permitted and not executed in this situation.
FD2050	E_KIN_GENPAR_LICENSE_COORDSYS	The licensing of the use of WCS, PCS1, PCS2 has not been requested in the configuration.
FD2060	E_KIN_GENPAR_MODULO_MODE	AxisGroupKin.Inst[..].In.Target.ModuloMode = KIN_DIRECTION_NIL is not permitted for a Cartesian dimension in modulo mode.

12.5 Error: Target parameters

Code	Error	Meaning
FD3000	E_KIN_TARGPAR_INTERNAL_ERROR	Internal error, consult SEW Service.
FD3008	E_KIN_TARGPAR_VEL	Incorrect setting for AxisGroupKin.Inst[..].In.Target.Axis / Cart.Velocity < 0.
FD3010	E_KIN_TARGPAR_ACC	Incorrect setting for AxisGroupKin.Inst[..].In.Target.Axis / Cart.Acceleration ≤ 0.
FD3020	E_KIN_TARGPAR_DEC_SMALL	Incorrect setting for AxisGroupKin.Inst[..].In.Target.Axis / Cart.Deceleration ≤ 0.
FD3030	E_KIN_TARGPAR_DEC_LARGE	Incorrect setting for AxisGroupKin.Inst[..].In.Target.Axis / Cart.Deceleration > AxisGroupKin.Inst[..].Config.Axis / Cart.RapidDeceleration, based on the current operating mode KIN_TARGET_AXIS / CART.
FD3040	E_KIN_TARGPAR_JERK	Incorrect setting for AxisGroupKin.Inst[..].In.Target.Axis / Cart.Jerk < AxisGroupKin.Inst[..].Config.Axis / Cart.RapidJerk, based on the current operating mode KIN_TARGET_AXIS / CART.
FD3050	E_KIN_TARGPAR_MODULO_OUT_OF_RANGE	Incorrect setting for AxisGroupKin.Inst[..].In.Target.Position > Cart.ModuloUnderflow or AxisGroupKin.Inst[..].In.Target.Axis / Cart.Velocity > CartModulo.Overflow.
FD3060	E_KIN_TARGPAR_OUT_OF_AXIS_SWLS	TARGET_AXIS to Cartesian *target position* not possible because the axis values corresponding to the *target position* lie outside the axis SWLS.
FD3070	E_KIN_TARGPAR_AXISCARTBIJECTION_NOT_WITH_MAPABC	AxisGroupKin.Inst[..].Config.General.AxisCartBijection and AxisGroupKin.Inst[..].In.Target.ABCMapping cannot both be set to *TRUE*.

12.6 Error: Continuous path parameters

Code	Error	Meaning
FD4000	E_KIN_CPPAR_INTERNAL_ERROR	Internal error; please contact the SEW-EURODRIVE service.
FD4010	E_KIN_CPPAR_LICENSE_PLUSTARGCART	The licensing of the use of KIN_TARGET_CART for the reorientation of the TCP during the execution of a translation via the path interpolation (CP) has not been requested in the configuration.
FD4018	E_KIN_CPPAR_UNCLEAR_PLUSTARGCART_OR_INCLUDINGABC	No clear selection of whether ABC should be interpolated during continuous path in synchronized mode or as a parallel process using TARGET_CART

Code	Error	Meaning
FD4028	E_KIN_CPPAR_BACKTOPATH_AXIS_COUPLING	Error while coupling several kinematics axis values to motor positions during a CP BackToPath motion.
FD4029	E_KIN_CPPAR_BACKTOPATH_AXIS_DECOUPLING	Error while decoupling several motor positions to kinematics axis values during a CP BackToPath motion.
FD4030	E_KIN_CPPAR_BACKTOPATH_MODE	No repositioning to the CP path because the path interpolation (CP) is deactivated.
FD4040	E_KIN_CPPAR_BACKTOPATH_DIST	No repositioning to the CP path because AxisGroupKin.Inst[..].In.Cp.BackToPath.MaxInitialDistance < distance to the start of repositioning.
FD4050	E_KIN_CPPAR_PATH_FIDELITY	Incorrect setting for AxisGroupKin.Inst[..].In.Cp.Path.FidelityPercentage < 90 or > 100.
FD4060	E_KIN_CPPAR_TRANSPERC_LARGE	In.Cp.Path.VelocityPercentage > Config.Cp.MaxPathVelocityPercentage.
FD4070	E_KIN_CPPAR_MASTERPOS	The values at the AxisGroupKin.Inst[..].In.MasterPosition.Cp input must not decline.
FD4080	E_KIN_CPPAR_DIRKIN	Continuous path segment has not been applied due to an error during the execution of the direct kinematic transformation.
FD4090	E_KIN_CPPAR_TRANSVEL	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Translation.Velocity ≤ 0.
FD40A0	E_KIN_CPPAR_TRANSACC	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Translation.Acceleration ≤ 0.
FD40B0	E_KIN_CPPAR_TRANSDEC_SMALL	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Translation.Deceleration ≤ 0.
FD40C0	E_KIN_CPPAR_TRANSDEC_LARGE	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Translation.Deceleration > Config.CpRapidTransDeceleration.
FD40D0	E_KIN_CPPAR_TRANSJERK	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Translation.Jerk < Config.Cp.RapidTransJerk.
FD40E0	E_KIN_CPPAR_QUEUE_SIZE	Continuous path segment has not been applied because the internal list of path segments to be traveled is completely filled and the corresponding path has not been traveled yet.
FD40E8	E_KIN_CPPAR_QUEUE_SIZE_REVERSE	When traveling back on the path, the limitation for the list of path segments to be traveled was reached. A rapid stop is performed.

24796131/EN – 12/2017

Code	Error	Meaning
FD40F0	E_KIN_CPPAR_QUEUE_NOT_FILL_AT_ERROR	Continuous path segment has not been applied due to the current kinematics error.
FD4100	E_KIN_CPPAR_PATH_LENGTH	Continuous path segment has not been applied because the total length of the traveled and planned path since the queue of path segments to travel was last emptied exceeds the maximum permitted value.
FD4110	E_KIN_CPPAR_LIN_DIST	The straight segment is not applied because the length is 0.
FD4120	E_KIN_CPPAR_LIN_BLENDING_DIST	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Translation.Deceleration ≤ 0.
FD4130	E_KIN_CPPAR_LIN_BLENDING_MAXPERC	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Blending.LimitationPercentage < 1 or < 99.
FD4140	E_KIN_CPPAR_CIRC_MODE	Circle segment (CIRC) has not been applied due to an AxisGroupKin.Inst[..].In.Cp.Segment.CircMode that is not permitted in this case.
FD4150	E_KIN_CPPAR_CIRC_PLANE	Circle segment (CIRC) has not been applied because the end tangent of the previous continuous path segment is not in the circular plane.
FD4160	E_KIN_CPPAR_CIRC_ANGLE	Circle segment (CIRC) has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.CircAngle ≤ 0.
FD4170	E_KIN_CPPAR_CIRC_RADIUS	Circle segment (CIRC) has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Radius ≤ 0 is not permitted for the set CIRC operating mode.
FD4180	E_KIN_CPPAR_CIRC_CENTER_CORRECTION	Reserved
FD4190	E_KIN_CPPAR_CIRC_DIRECTION	Circle segment (CIRC) has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.CircDirection = KIN_DIRECTION_NIL / SHORT.
FD41A0	E_KIN_CPPAR_CIRC_DISTORTION	Circle segment (CIRC) has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Circ.EllipticDistortion ≤ 0.
FD4320	E_KIN_CPPAR_ROTATION_VEL	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Rotation.Velocity[j] ≤ 0, j = 4 to 6.
FD4330	E_KIN_CPPAR_ROTATION_ACC	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Cart.Acceleration[j] ≤ 0, j = 4 to 6.

24796131/EN – 12/2017

Code	Error	Meaning
FD4340	E_KIN_CPPAR_ROTATION_DEC_SMALL	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Cart.Deceleration[j] ≤ 0, j = 4 to 6.
FD4350	E_KIN_CPPAR_ROTATION_DEC_LARGE	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Cart.Deceleration[j] > AxisGroupKin.Inst[..].Config.Cart.RapidDeceleration[j], j = 4 to 6.
FD4360	E_KIN_CPPAR_ROTATION_JERK	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Cart.Jerk < AxisGroupKin.Inst[..].Config.Cart.RapidJerk[j], j = 4 to 6.
FD43B0	E_KIN_CPPAR_AXIS_VEL	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Axis.Velocity[j] ≤ 0.
FD43C0	E_KIN_CPPAR_AXIS_ACC	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Axis.Acceleration[j] ≤ 0.
FD43D0	E_KIN_CPPAR_AXIS_DEC_SMALL	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Axis.Deceleration[j] ≤ 0.
FD43E0	E_KIN_CPPAR_AXIS_DEC_LARGE	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Axis.Deceleration[j] > AxisGroupKin.Inst[..].Config.Axis.RapidDeceleration[j].
FD43F0	E_KIN_CPPAR_AXIS_JERK	Continuous path segment has not been applied because AxisGroupKin.Inst[..].In.Cp.Segment.Axis.Jerk < AxisGroupKin.Inst[..].Config.Axis.RapidJerk[j].

12.7 Error: Profile generator

Code	Error	Meaning
FD5000	E_KIN_PROFGEN_INTERNAL_ERROR	Internal error, consult SEW Service.
FD5010	E_KIN_PROFGEN_INIT_DELTAT	Incorrect setting for KinProfGen.Init.DeltaT < 0.
FD5020	E_KIN_PROFGEN_INIT_MODULO_RANGE	Incorrect setting for KinProfGen.Init.ModuloUnderflow ≤ Overflow.
FD5030	E_KIN_PROFGEN_INIT_OUT_OF_MODULO_RANGE	Incorrect setting for KinProfGen.Init.InitPos outside the range of KinProfGenInit.ModuloUnderflow (Overflow).
FD5040	E_KIN_PROFGEN_MODULO_TARGET_OUT_OF_RANGE	Incorrect setting for KinProfGen.TargPos outside the range of KinProfgenInit.ModuloUnderflow (Overflow).
FD5050	E_KIN_PROFGEN_MODULO_MODE	Incorrect assignment of KinProfGen.ModuloMode = KIN_DIRECTION_NIL.

24796131/EN – 12/2017

Code	Error	Meaning
FD5060	E_KIN_PROFGEN_MOD-ULO_TOO_FAST_POSITIVE	More than the complete modulo range would have to be passed in positive direction in one cycle.
FD5070	E_KIN_PROFGEN_MOD-ULO_TOO_FAST_NEGATIVE	More than the complete modulo range would have to be passed in negative direction in one cycle.
FD5080	E_KIN_PROFGEN_VEL_MAX	Incorrect assignment of KinProfGen.Vmax < 0.
FD5090	E_KIN_PROFGEN_ACC_MAX	Incorrect assignment of KinProfGen.Amax ≤ 0.
FD50A0	E_KIN_PROFGEN_DEC_MAX	Incorrect assignment of KinProfGen.Dmax ≤ 0.
FD50B0	E_KIN_PROFGEN_DEC_RAPID	Incorrect assignment of KinProfGen.D_Rapid ≤ 0.
FD50C0	E_KIN_PROFGEN_JERK	Incorrect assignment of KinProfGen.JerkTime < 0.
FD50C0	E_KIN_PROFGEN_JERK_RAPID	Incorrect assignment of KinProf-Gen.JerkTime_Rapid < 0.

12.8 Error: SRL

Code	Error	Meaning
FDB100	E_KIN_SRL_CONTROL_STRUC-TURE_NESTING_DEPTH_OVERFLOW	The nesting depth of the control structure has been exceeded.
FDB101	E_KIN_SRL_END_CONTROLINSTRUC-TION_EXPECTED	The end identifier of a control structure is miss-ing.
FDB102	E_KIN_SRL_VERSION_EXPECTED	Version expected.
FDB105	E_KIN_SRL_INSTANCENUM-BER_NOT_VALID	The instance number is not valid.
FDB110	E_KIN_SRL_OPERAND_1_OF_EXPRES-SION_NOT_VALID	The first operand of an expression is not valid.
FDB111	E_KIN_SRL_OPERAND_2_OF_EXPRES-SION_NOT_VALID	The second operand of an expression is not valid.
FDB112	E_KIN_SRL_EXPRESSION_NUM-BER_NOT_VALID	The identifier of an expression is not valid.
FDB113	E_KIN_SRL_SUBSCRIPTED_AC-CESS_NOT_POSSIBLE_VARIABLE_IN-DEX_OUT_OF_RANGE	Subscripted access is not possible, access viol-ation.
FDB114	E_KIN_SRL_INCOMPATIBLE_OPER-AND_FOR_EXPRESSION	The operands of an expression are not compat-ible.
FDB115	E_KIN_SRL_SUBSCRIPTED_AC-CESS_NOT_POSSIBLE	Subscripted is not possible because one of the following conditions is violated: • Operand 1 is a variable. • The type for operand 2 is REAL or LREAL. • The operation result type is compatible with the assignment target type.

Code	Error	Meaning
FDB116	E_KIN_SRL_INCOMPATIBLE_TARGET-TYPE_FOR_EXPRESSION	The expression result type is not compatible with the assignment target type.
FDB117	E_KIN_SRL_INCOMPATIBLE_TYPES_IN_EX-PRESSION	The operand types of the expression are not compatible.
FDB120	E_KIN_SRL_CONTROLSTRUC-TURE_END_TOKEN_NOT_ALLOWED	The end identifier of a control structure is not permitted.
FDB121	E_KIN_SRL_CONTROLSTRUC-TURE_NOT_IMPLEMENTED	The control structure has not been implemen-ted.
FDB122	E_KIN_SRL_START_CONTROLINSTRUC-TION_EXPECTED	A start identifier of a control structure is expec-ted.
FDB130	E_KIN_SRL_CALL_NEST-ING_DEPTH_OVERFLOW	The call nesting depth has been exceeded.
FDB131	E_KIN_SRL_NOT_IMPLEMENTED	The command has not been implemented.
FDB132	E_KIN_SRL_PROGRAM-POINTER_NOT_VALID	The program indicator is not valid.
FDB133	E_KIN_SRL_SUBTOKEN_NOT_ALLOWED	The subtoken of a command is not valid.
FDB134	E_KIN_SRL_TOKEN_NOT_ALLOWED	The token of a command is not valid.
FDB135	E_KIN_SRL_PROGRAMNUM-BER_NOT_VALID	The *Program number* is not valid.
FDB136	E_KIN_SRL_PROGRAMNUM-BER_NOT_POSSIBLE_TO_EXECUTE	The program with the selected *Program num-ber* cannot be executed.
FDB137	E_KIN_SRL_SYS_PARAMETER_NOT_VALID	An SRL system parameter is not valid.
FDB140	E_KIN_SRL_KIN_TOKEN_NOT_ALLOWED	The token of a kinematics command is not valid.
FDB150	E_KIN_SRL_KIN_SUBTOKEN_NOT_AL-LOWED	The subtoken of a kinematics command is not valid.
FDB160	E_KIN_SRL_MODE_NOT_ALLOWED	The selected SRL operating mode is not valid.
FDB170	E_KIN_SRL_EVENT_NUMBER_OVERFLOW	The number of registered events has been ex-ceeded.
FDB171	E_KIN_SRL_EVENT_BUFFER_ERROR	Error in the event buffer.
FDB300	E_KIN_SRL_EDIT_INSTRUCTION_OPER-AND_NOT_POSSIBLE	It is not possible to edit an operand.
FDB301	E_KIN_SRL_EDIT_INSTRUCTION_OPER-ATOR_NOT_POSSIBLE	It is not possible to edit an operator.
FDB302	E_KIN_SRL_COMBINATION_OPER-AND1_AND_OPERAND2_NOT_VALID	The combination of operand 1 and 2 is not valid.
FDB400	E_KIN_SRL_GO_EXACT_NOT_POS-SIBLE_PLEASE_SELECT_NON_RELAT-IVE_MOTIONINSTRUCTION	Cannot select a relative motion command.
FDB500	E_KIN_SRL_PARAMETER-VALUE_NOT_VALID	A parameter value is not valid.
FDB501	E_KIN_SRL_VARTYPE_NOT_VALID	A variable type is not valid.

24796131/EN – 12/2017

Code	Error	Meaning
FDB510	E_KIN_SRL_GET_VARIABLE-VALUE_OF_VARTYPE_NOT_POSSIBLE	The value of a variable could not be determined.
FDB520	E_KIN_SRL_READ_FROM_VARINDEX_NOT_POSSIBLE	The index of a variable is not valid.
FDB521	E_KIN_SRL_READ_VARTYPE_NOT_POSSIBLE	The variable type is not valid.
FDB530	E_KIN_SRL_PARAMETER-VALUE_NOT_POSSIBLE	The parameter value is not in the permitted range.
FDB531	E_KIN_SRL_SET_DEREF_VARIABLE_OF_VARTYPE_NOT_POSSIBLE	Cannot dereference a variable.
FDB540	E_KIN_SRL_SET_VARIABLE_OF_VARTYPE_NOT_POSSIBLE	Cannot describe a variable.
FDB550	E_KIN_SRL_POSITIONINDEX_NOT_VALID	The position index is not valid.
FDB560	E_KIN_SRL_WRITE_TO_VARIABLE_INDEX_NOT_POSSIBLE	Cannot write to a variable index.
FDB561	E_KIN_SRL_WRITE_VARIABLE_ONLY_POSSIBLE_IF_READING_DATA_WAS_SUCCESSFUL	A variable can be written only if reading was successful.
FDB562	E_KIN_SRL_WRITE_VARIABLE-TYPE_NOT_POSSIBLE	The variable type could not be changed.
FDB600	E_KIN_SRL_TYPE_DATA_READ_NOT_VALID	Error when reading from data medium. The data types are not active.
FDB610	E_KIN_SRL_TYPE_DATA_WRITE_NOT_VALID	Error when writing to data medium. The data types are not active.
FDB611	E_KIN_SRL_WRITING_FILE_NOT_POSSIBLE	Could not write to the file on the data medium.
FDB620	E_KIN_SRL_TYPE_DATA_FILE_NOT_VALID	The file type is not valid.
FDB621	E_KIN_SRL_NUMBER_OF_DATA_FILE_NOT_VALID	The number of files is not valid.
FDB700	E_KIN_SRL_FUNCTION_NUMBER_NOT_VALID	The number of the user function is not valid.
FDB900	E_KIN_SRL_TOUCH_PROBE_CALCULATION_POSITION_FROM_ENCODER	TP_ActivateMeasure/Motion: The calculation of the Cartesian position from the encoder values of the inverters (increments) failed.
FDB910	E_KIN_SRL_TOUCH_PROBE_POSITION_FOR_CALCULATION_NOT_VALID	TP_ActivateMeasure/Motion: The position P_n or P_{n-1} is not valid. P_n and P_{n-1} must be valid for calculating the Touchprobe.
FDB920	E_KIN_SRL_TOUCH_PROBE_TP_COUNTERS_OF_INVERTERS_NOT_EQUAL	TP_ActivateMeasure/Motion: The Touchprobe counters of the inverters are not the same.
FDB925	E_KIN_SRL_TOUCH_PROBE_TP_COUNTERS_OF_INVERTERS_NOT_ZERO	TP_Prepare: At least one touch probe counter was not set to 0.
FDB930	E_KIN_SRL_TOUCH_PROBE_PROJECT_POINT_TO_LINE	TP_ActivateMeasure/Motion: The projection of the Touchprobe position to the connection line from P_{n-1} to P_n failed.

Code	Error	Meaning
FDB940	E_KIN_SRL_TOUCH_PROBE_NOT_ACTIVE	TP_Prepare: The Touchprobe of at least one inverter is not active.
FDB950	E_KIN_SRL_TOUCH_PROBE_NO_TP_EVENT_OCCURRED	TP_ActivateMeasure: No Touchprobe result occurred.
FDB960	E_KIN_SRL_TOUCH_PROBE_NO_VALID_TP_EVENT_OCCURRED	TP_ActivateMeasure/Motion: No valid Touchprobe result occurred.
FDBA00	E_KIN_SRL_AGK_CPSETTINGS_NOT_TAKEN_OVER : UDINT :=16#00FDBA00	A CP setting such as IncludingABC, IncludingAxis7/8, or PlusTargetCartABC was not applied.

12.9 Error: AxisGroupControl Kinematics

Code	Error	Meaning
FDF000	E_KIN_AXISGROUPKIN_INTERNAL_ERROR	Internal error, consult SEW Service.
FDF010	E_KIN_AXISGROUPKIN_READING_AXIS_VALUES_NOT_POSSIBLE	Reading of the axis increments was unsuccessful. Check whether the inverter is ready.
FDF020	E_KIN_AXISGROUPKIN_WRITING_AXIS_VALUES_NOT_POSSIBLE	Unable to successfully write the initial values for the MC_KinControl module.
FDF030	E_KIN_AXISGROUPKIN_CONFIGURATION_PRG_DONE_NOT_FALSE	Handshake to the configuration program was unsuccessful.
FDF040	E_KIN_AXISGROUPKIN_CONFIGURATION_PRG_DONE_NOT_TRUE	
FDF041	E_KIN_AXISGROUPKIN_READ_CONFIG_FROM_SD_CARD_NOT_DONE	Reading of the configuration from the memory card was not executed.
FDF050	E_KIN_AXISGROUPKIN_CONFIG_IS_NOT_TAKEN_OVER	Kinematics configuration not applied because the configuration is incorrect or the cyclical task is at standstill, for example.
FDF060	E_KIN_AXISGROUPKIN_CLEAR_LAG_FALSE_NOT_POSSIBLE	Handshake to the configuration program was unsuccessful.
FDF070	E_KIN_AXISGROUPKIN_LAG_IS_NOT_CLEARED	
FDF080	E_KIN_AXISGROUPKIN_NOT_ALL_AXES_IN_INTERPOLATION_MODE	One or more axes could not be switched to interpolation operating mode (see MultiMotion Axis Interface) after switching to an interpolating operating mode.
FDF090	E_KIN_AXISGROUPKIN_BACKTOPATH_NOT_ACTIVE	Faulty handshake to MC_KinControl.
FDF0A0	E_KIN_AXISGROUPKIN_BACKTOPATH_CLEAR_LAG_FALSE_NOT_POSSIBLE	Handshake to the configuration program was unsuccessful.
FDF0B0	E_KIN_AXISGROUPKIN_BACKTOPATH_LAG_NOT_CLEARED	

Code	Error	Meaning
FDF0C0	E_KIN_AXISGROUPKIN_BACKTO-PATH_NOT_IN_INTERPOLATION	One or more axes could not be switched to interpolation operating mode (see MultiMotion AxisInterface), after the kinematic model left the path (BackToPath).
FDF0D0	E_KIN_AXISGROUPKIN_NUM-BER_OF_KIN_AXES_NOT_VALID	Incorrect setting for the number of kinematic axes ≤ 0 or > 6.
FDF0E0	E_KIN_AXISGROUPKIN_NUM-BER_OF_AUX_AXES_NOT_VALID	Incorrect setting for the number of auxiliary axes < 0 or > 2.
FDF0F0	E_KIN_AXISGROUPKIN_MODE_NOT_VALID	Selected operating mode is not supported by AxisGroupControl Kinematics.
FDF100	E_KIN_AXISGROUPKIN_IN-STANCEID_NOT_VALID	The InstanceID set in AxisGroup Configuration is not valid. The InstanceID must not be smaller than 1 and not greater than the maximum number of instances (3 to 12).
FDF110	E_KIN_AXISGROUPKIN_INSTANCEID_DU-PLICATE	The InstanceID set in the AxisGroup Configuration was assigned for at least one other instance.
FDF120	E_KIN_AXISGROUPKIN_INSTANCE-NAME_DUPLICATE	The InstanceName set in the AxisGroup Configuration was assigned for at least one other instance.
FDF130	E_KIN_AXISGROUPKIN_NUMBER_OF_IN-STANCES_NOT_VALID	NumberOfInstances is greater than MaxNumberOfAGKInstances.

24796131/EN – 12/2017

13 Additional error codes

13.1 Error: NV memory

SEW ErrorCode 25: E_WRITE_NV_MEM

An error was detected during access to non-volatile parameter memory.

Code	Error	Meaning
190001	E_NV_ADDRESS_ERROR	NV memory address access.
190003	E_NV_IMPORT_ERROR	NV memory import error.
190004	E_NV_SETUP_ERROR	NV memory setup error.
190005	E_NV_INVALID_DATA	NV memory data error.
190006	E_NV_INCOMPATIBLE_DATA	Data incompatible with NV memory.
190007	E_NV_INITIALISATION_ERROR	NV memory initialization error.
190008	E_NV_INTERNAL_ERROR	NV memory internal error.
1900C8	E_NV_BUSY	Communication with NV memory is already active.
1900C9	E_NV_MEM_OVERFLOW	Memory limits reached.
1900CA	E_NV_READ_INIT	Initialization error while reading data.
1900CB	E_NV_WRITE_INIT	Initialization error while writing data.
1900CC	E_NV_READ_INIT_EXT	Initialization error while reading from the extended memory area.
1900CD	E_NV_WRITE_INIT_EXT	Initialization error while writing to the extended memory area.
1900CE	E_NV_NO_WRITE_ACCESS	No write access to the memory area.
1900DC	E_NV_NO_ACK	No receipt acknowledgment from the hardware.
1900DD	E_NV_PAGE_OVERFLOW	EEPROM page overflow.
1900DE	E_NV_NO_DATA_TRANSFER_FLAG	Data transfer flag is missing.
1900DF	E_NV_MEMORY_SIZE_SMALL	Incorrect memory size.

13.2 Error: Unit configuration data

SEW ErrorCode 94: E_CONFIG_ERROR

An error has occurred in the unit configuration data block during testing in reset phase.

Code	Error	Meaning
5E0001	E_CONFIG_CHECKSUM	Unit configuration data: CRC checksum error

24796131/EN – 12/2017

13.3 Error: PLC configuration

SEW ErrorCode 201: E_PLCINIT

Code	Error	Meaning
C90001	E_CONFIG_FATAL_ERROR	PLC configuration: A severe firmware error occurred.
C90002	E_CONFIG_UNKNOWN_MODUL_ID	PLC configuration: Unknown module detected.
C90003	E_CONFIG_UNKNOWN_PARAM_ID	PLC configuration: Unknown parameter detected.
C90004	E_CONFIG_UNKNOWN_CHANNEL_ID	PLC configuration: Unknown channel detected.
C90005	E_CONFIG_INIT_IEC_POINTER_FAILED	PLC configuration: Initialization of an IO pointer failed.
C90006	E_CONFIG_TOO_MANY_SUBMODULES	PLC configuration: Maximum number of sub modules reached.
C90007	E_CONFIG_NGDPM_CHANNEL_REQ_FAILED	PLC configuration: Channel setup failed.
C90008	E_CONFIG_CAN_DRV_INIT_FAILED	PLC configuration: CAN driver initialization failed.
C90009	E_CONFIG_INVALID_WORD_ADDRESS	PLC configuration: Invalid WORD address.
C9000A	E_CONFIG_DIO_INCOMPATIBLE	PLC configuration: MOVI-PLC inputs/outputs incompatible.

13.4 Error: CPU trap

SEW ErrorCode 202: E_CPU_TRAP

Code	Error	Meaning
CA0001	E_CPU_TRAP_NMI	CPU trap: Non-maskable interrupt
CA0002	E_CPU_TRAP_STACK_OVERFLOW	CPU trap: Stack overflow
CA0003	E_CPU_TRAP_STACK_UNDERFLOW	CPU trap: Stack underflow
CA0004	E_CPU_TRAP_UNDEF_OPCODE	CPU trap: Undefined opcode
CA0005	E_CPU_TRAP_SW_BREAK	CPU trap: Software break
CA0006	E_CPU_TRAP_PMI_ACCESS_ERROR	CPU trap: Program memory access error
CA0007	E_CPU_TRAP_PROTECTED_INSTRUCTION_FAULT	CPU trap: Protection fault
CA0008	E_CPU_TRAP_ILLEGAL_WORD_OPERAND_ACCESS	CPU trap: Illegal word operand access

13.5 Error: Floating point trap

SEW ErrorCode 203 E_FP_TRAP

Code	Error	Meaning
CB0001	E_FP_TRAP_OVERFLOW	Floating point trap: Overflow

24796131/EN – 12/2017

Code	Error	Meaning
CB0002	E_FP_TRAP_UNDERFLOW	Floating point trap: Underflow
CB0003	E_FP_TRAP_DIV_ZERO	Floating point trap: Division by zero
CB0004	E_FP_TRAP_INVALID_OP	Floating point trap: Invalid OP code
CB0005	E_FP_TRAP_CONVERSION_ERROR	Floating point trap: Converting error
CB0006	E_FP_TRAP_STK_UNDERFLOW	Floating point trap: Stack overflow
CB0007	E_FP_TRAP_STK_OVERFLOW	Floating point trap: Stack underflow
CB0008	E_FP_TRAP_INT_OVERFLOW	Floating point trap: Integer overflow
CB0009	E_TRAP_UNKNOWN	Floating point trap: Unknown trap

13.6 Error: Memory

SEW ErrorCode 204 E_MEMORY: Error in the memory

Code	Error	Meaning
CC0001	E_CRC_ERROR_EXT_CODE	Memory error: memory code check error

13.7 Error: Error handling

SEW ErrorCode 240 E_ERRORHANDLING

Code	Error	Meaning
F00001	E_ERRORBUFFER_OVERFLOW	Error handling: Buffer overflow
F00002	E_ERRORID_NOT_IN_REACTION_TABLE	Error handling: Combination of ErrorCode and SubErrorCode not found in error reaction table
F0000A	E_FW_EC_OUT_OF_RANGE	Error handling: ErrorCode not found within firmware error codes
F00014	E_LIB_EC_OUT_OF_RANGE	Error handling: ErrorCode not found within IEC error codes
F0001E	E_USER_EC_OUT_OF_RANGE	Error handling: ErrorCode not found within user error codes

13.8 Error: MOVILINK®

SEW ErrorCode 242 E_MOVILINK

Code	Error	Meaning
F20001	E_ML_ERROR	MOVILINK error : General MOVILINK error. See MOVILINK return code for further information

24796131/EN – 12/2017

13.9 Error: Communication function block

SEW ErrorCode 243 E_COMMUNICATION_FB: Error within communication function blocks

Code	Error	Meaning
F30001	E_INWORD_OUT_OF_RANGE	Communication function block: Value out of range
F30002	E_OUTWORD_OUT_OF_RANGE	Communication function block: Value not in OUT memory
F30003	E_INVALID_INPUT_BUFFER_LOCATION	Communication function block: Input object does not fit to input buffer
F30004	E_INVALID_OUTPUT_BUFFER_LOCATION	Communication function block: Output object does not fit to output buffer
F30005	E_INVALID_MERKER_BUFFER_LOCATION	Communication function block: Memory object does not fit to Memory buffer
F30006	E_PRM_MASTER_ALL_BUFFERS_IN_USE	Parameter master: Too many parameter master requests on a channel active
F30007	E_PLC_PMSTR_INVALID_SERVICE	Parameter master: Invalid service
F30008	E_PLC_PMSTR_SERVICE_NOT_IMPLEMENTED	Parameter master: Service not supported
F30064	E_CAN_UNKNOWN_ERROR	Communication function block CAN: Unknown error
F30065	E_CAN_INVALID_NODE	Communication function block CAN: Invalid CAN node
F30066	E_CAN_INVALID_ADDRESS	Communication function block CAN: Invalid CAN address
F30067	E_CAN_INVALID_BAUDRATE	Communication function block CAN: Invalid baud rate
F30068	E_CAN_INVALID_SERVICE	Communication function block CAN: Invalid service
F30069	E_CAN_INVALID_CHANNEL	Communication function block CAN: Invalid channel
F3006A	E_CAN_CH_BUSY	Communication function block CAN: Communication channel is busy
F3006B	E_CAN_SCOM_ALREADY_IN_STATE	Communication function block CAN: SCOM control state is already reached
F3006C	E_CAN_SCOM_INIT_ERROR	Communication function block CAN: Error in SCOM init
F3006D	E_CAN_SCOM_INIT_WHILE_RUN	Communication function block CAN: SCOM initialization not allowed while CAN is on
F3006E	E_CAN_INVALID_SCOM_ID	Communication function block CAN: Invalid CAN ID
F3006F	E_CAN_INVALID_SCOM_OBJECT	Communication function block CAN: Invalid SCOM object
F30070	E_CAN_SCOM_INVALID_LENGTH	Communication function block CAN: Invalid data length

24796131/EN – 12/2017

Code	Error	Meaning
F30071	E_CAN_MAX_SCOM_OBJECTS	Communication function block CAN: Maximum number of SCOM objects reached
F30072	E_CAN_SCOM_ACYCLIC_TIMEOUT	Communication function block CAN: SCOM timeout
F30073	E_CAN_NO_SCOM_ON	Communication function block CAN: SCOM is not in state ON
F30074	E_CAN_BY_INIT_LOCKED	Communication function block CAN: CAN init is locked by another instance
F30075	E_CAN_INVALID_COM_TYPE	Communication function block CAN: Unknown SCOM type
F30076	E_CAN_INVALID_CYCLETIME	Communication function block CAN: Invalid cycle time
F30077	E_CAN_CH_ERROR_LEN	Communication function block CAN: Invalid data length
F30078	E_SCOM_NOT_SUPPORTED	Communication function block CAN: SCOM remote not supported
F300C8	E_IO_UNKNOWN_ERROR	Communication function block DIO: Unknown error
F300C9	E_IO_INVALID_PORT	Communication function block DIO: Invalid port
F300CA	E_IO_INVALID_INTERRUPT_MODE	Communication function block DIO: Invalid interrupt mode
F3012C	E_RS485_UNKNOWN_ERROR	Communication function block COM: Unknown error
F3012D	E_RS485_INVALID_TIMEOUT	Communication function block COM: Invalid timeout value
F3012E	E_RS485_INVALID_ADDRESS	Communication function block COM: Invalid address
F3012F	E_RS485_INVALID_BAUDRATE	Communication function block COM: Invalid baud rate
F30130	E_RS485_INVALID_MODE	Communication function block COM: Invalid mode
F30131	E_RS485_INVALID_PARITY	Communication function block COM: Invalid parity
F30132	E_RS485_INVALID_STOPBIT	Communication function block COM: Invalid stop bit
F30133	E_RS485_INVALID_SERVICE	Communication function block COM: Invalid service
F30134	E_RS485_BY_INIT_LOCKED	Communication function block COM: RS485 is already in use
F30135	E_RS485_INVALID_PDU_TYPE	Communication function block COM: Invalid number of process data
F30136	E_RS485_INVALID_NODE	Communication function block COM: Invalid COM node

Code	Error	Meaning
F30137	E_RS485_TOO_MANY_SLAVES	Communication function block COM: Maximum number of slaves reached
F30138	E_RS485_SLAVE_ALREADY_EXISTS	Communication function block COM: Slave already exists
F30139	E_RS485_INVALID_CONTROL_MODE	Communication function block COM: Invalid control mode
F3013A	E_RS485_RESPONSE_TIMEOUT	Communication function block COM: Parameter timeout
F3013B	E_RS485_WRONG_DRIVER	Communication function block COM: Wrong driver
F3013C	E_RS485_CONVERSION_ERROR	Communication function block COM: Conversion error
F3013D	E_RS485_PARAMETER_ERROR	Communication function block COM: Parameter error
F3013E	E_RS485_SUBIDX_NOT_SUPPORTED	Communication function block COM: Subindex not supported
F3013F	E_RS485_DRIVER_NOT_CONNECTED	Communication function block COM: Internal driver error
F30140	E_RS485_INVALID_IO_ADDRESS	Communication function block COM: Invalid IO addresses
F30141	E_RS485_DHW_ALLREADYSENDING	Communication function block COM Ex: Send buffer full
F30142	E_RS485_DHW_TOOMANYBYTESTOWRITE	Communication function block COM Ex: Too much data to send
F30143	E_RS485_DHW_CONVERSIONERROR	Communication function block COM Ex: Conversion error
F30144	E_RS485_FAILURE	Communication function block COM Ex: General error
F30145	E_RS485_INVALID_POINTER	Communication function block COM Ex: Invalid pointer
F30146	E_RS485_MAX_PD_REPETITIONS	Communication function block COM: Maximum PdRepetitions reached
F30190	E_PB_TIMEOUT	Communication function block fieldbus: Timeout
F30191	E_PB_INVALID_OPTION	Communication function block fieldbus: Invalid option
F30192	E_PB_TIMEOUT_DPRAM	Communication function block fieldbus: DpRam timeout
F30193	E_PB_DPRAM_NOT_RUN	Communication function block fieldbus: DpRam not in state run
F30194	E_PB_WRONG_PD_TYPE	Communication function block fieldbus: DpRam wrong PD type
F30195	E_PB_NO_PD_DATA	Communication function block fieldbus: DpRam no valid PD data

Code	Error	Meaning
F301F3	E_PB_UNKNOWN_ERROR	Communication function block fieldbus: DpRam unknown error
F301F4	E_NG_DPM_TOO_MANY_PARAMETER_REQUESTS	Communication function block NGDPRAM: Too many parameter master requests
F301F5	E_NG_DPM_PARAMETER_REQUEST_FAILED	Communication function block NGDPRAM: Parameter request failed
F301F6	E_NG_DPM_PARAMETER_FB_TIMEOUT	Communication function block NGDPRAM: Parameter timeout
F301F7	E_NG_DPM_UNKOWN_ERROR	Communication function block NGDPRAM: Unknown error
F301F8	E_NG_DPM_INVALID_SERVICE	Communication function block NGDPRAM: Invalid service
F301F9	E_NG_DPM_INVALID_REQUEST_BUFFER_REF	Communication function block NGDPRAM: Invalid pointer to buffer
F301FA	E_NG_DPM_INVALID_PRIORITY	Communication function block NGDPRAM: Invalid priority
F301FB	E_NG_DPM_CHANNEL_REQUEST_FAILED	Communication function block NGDPRAM: Channel request failed
F301FC	E_NG_DPM_VARDATA_INVALID_DATA_LENGTH	Communication function block NGDPRAM: Invalid data length
F301FD	E_NG_DPM_PARAM_INVALID_DATA_LENGTH	Communication function block NGDPRAM: Invalid data length
F301FE	E_NG_DPM_INVALID_SYNC_INTERVAL	Communication function block NGDPRAM: Invalid sync interval
F302BC	E_VDFC02_WRONG_DATATYPE	VarData FC02 header: Invalid data type received
F302BD	E_VDFC02_WRONG_GRANULARITY	VarData FC02 header: Invalid granularity received
F302BE	E_VDFC02_WRONG_STARTADDRESS	VarData FC02 header: Wrong start address
F302BF	E_VDFC02_WRONG_SERVICE	VarData FC02 protocol: Invalid service
F302C0	E_VDFC02_WRONG_FUNCTION	VarData FC02 protocol: Invalid function code
F302C1	E_VDFC02_SLAVE_BUSY	VarData FC02 protocol: Slave not ready
F302C2	E_VDFC02_INVALID_POINTER	VarData FC02 protocol: Invalid pointer
F302C6	E_CAMDATA_INVALID_SERVICE	Cam data: Invalid service
F302C7	E_CAMDATA_INVALID_CAM_NUMBER	Cam data: Invalid cam number
F302C8	E_CAMDATA_INVALID_CAM_LENGTH	Cam data: Max number of cam points reached
F302C9	E_CAMDATA_NO_PARAM_BUFFER_FREE	Cam data: No data buffer available
F302D0	E_IPOSCODE_INVALID_SERVICE	IPOS code download: Invalid service
F302D1	E_IPOSCODE_INVALID_DATA_LENGTH	IPOS code download: Invalid data length
F302D2	E_IPOSCODE_NO_PARAM_BUFFER_FREE	IPOS code download: No data buffer available
F302D3	E_IPOSCODE_INVALID_VERSION	IPOS code download: Invalid version
F302DA	E_DEVICEDATA_INVALID_SERVICE	Device data download: Invalid service

24796131/EN – 12/2017

Code	Error	Meaning
F302DB	E_DEVICEDATA_NO_PARAM_BUF-FER_FREE	Device data download: No data buffer available
F302DC	E_DEVICEDATA_INVALID_COMP_MODE	Device data download: Invalid compatibility mode
F302E4	E_USERMEM_INVALID_SERVICE	UserMem: Unknown service selected
F302E5	E_USERMEM_INVALID_DATA_LENGTH	UserMem: Invalid data length
F302E6	E_USERMEM_INVALID_ADDRESS	UserMem: Invalid address
F302E7	E_USERMEM_INVALID_MEM_AREA	UserMem: Invalid memory area
F302E8	E_USERMEM_ABORTED	UserMem: Communication was terminated
F302E9	E_USERMEM_INVALID_POINTER	UserMem: Invalid pointer
F30320	E_COMLIB_INVALID_INTERFACE	Com lib: Invalid interface
F30321	E_COMLIB_INVALID_ROUTINGINFO	Com lib: No routing parameter found for the selected parameter
F30384	E_ROUTING_INVALID_SERVICE	Routing invalid service
F30385	E_ROUTING_INVALID_ADDRESS	Routing invalid address
F30386	E_ROUTING_SENDREQUEST_FAILED	Routing: Send request failed
F30387	E_ROUTING_INVALID_INST	Routing: Error while establishing an instance for this function block
F303E8	E_ETC_BY_INIT_LOCKED	ETC: EtherCAT is locked by another instance
F303E9	E_ETC_INVALID_SERVICE	ETC: Invalid service
F303EA	E_ETC_CH_BUSY	ETC: Communication channel is busy
F303EB	E_ETC_INVALID_ADDRESS	ETC: Invalid EtherCAT slave address
F303EC	E_ETC_INVALID_STATE	ETC: Invalid EtherCAT master/slave state
F303ED	E_ETC_INVALID_DATA_LENGTH	ETC: Invalid data length
F303EE	E_ETC_STATE_CHANGE_IN_PROGRESS	ETC: State change in progress
F303EF	E_ETC_MASTER_NOT_SUPPORTED	ETC: SNI master is activated
F303F0	E_ETC_SDO_NO_RESPONSE	ETC: ETC SDO timeout
F303F1	E_ETC_SDO_INVALID_INDEX	ETC: ETC SDO invalid Index
F303F2	E_ETC_SDO_READ_ONLY_INDEX	ETC: ETC SDO read only Index
F303F3	E_ETC_SDO_INVALID_VALUE	ETC: ETC SDO invalid value
F303F4	E_ETC_SDO_VALUE_TOO_LARGE	ETC: ETC SDO value too large
F303F5	E_ETC_SDO_VALUE_TOO_SMALL	ETC: ETC SDO value too small
F303F6	E_ETC_SDO_INTERNAL_ERROR	ETC: ETC SDO internal error
F3044B	E_ETC_UNKNOWN_ERROR	ETC: Unknown error
F3044C	E_SNI_BY_INIT_LOCKED	SNI: SNI is locked by another instance
F3044D	E_SNI_INVALID_SERVICE	SNI: Invalid service
F3044E	E_SNI_CH_BUSY	SNI: Communication channel is busy
F3044F	E_SNI_INVALID_ADDRESS	SNI: Invalid SNI slave address

24796131/EN – 12/2017

Code	Error	Meaning
F30450	E_SNI_INVALID_STATE	SNI: Invalid SNI master/slave state
F30451	E_SNI_INVALID_DATA_LENGTH	SNI: Invalid data length
F30452	E_SNI_STATE_CHANGE_IN_PROGRESS	SNI: State change in progress
F30453	E_SNI_MASTER_NOT_SUPPORTED	SNI: ETC master is activated
F304AF	E_SNI_UNKNOWN_ERROR	SNI: Unknown error
F304B0	E_FBUS_DRIVER_INIT_ERROR	FBUS: Init failed
F304B1	E_FBUS_RECURSIVE_CALL	FBUS: FB called more than once
F304B2	E_FBUS_TIMEOUT	FBUS: Timeout on fieldbus
F304B3	E_FBUS_WRONG_PD_TYPE	FBUS: Invalid configuration from master
F304B4	E_FBUS_WATCHDOG	FBUS: Timeout in internal communication
F304B5	E_FBUS_QUEUE_ERROR	FBUS: Error in internal queues
F30513	E_FBUS_UNKOWN_ERROR	FBUS: Unknown error

13.10 Error: System function block

SEW ErrorCode 244 E_SYSTEM_FB: Error within system function blocks

Code	Error	Meaning
F4000A	E_SYSFCTS_UNKOWN_ERROR	System function: Unknown error
F4000B	E_SYSFCTS_INVALID_POINTER	System function: Invalid pointer
F4000C	E_SYSFCTS_FUNCTION_NOT_IMPLEMEN-TED	System function: Function is not implemented
F40014	E_TASKSYS_UNKOWN_ERROR	Task system: Unknown error
F40015	E_TASKSYS_INVALID_TASK_ID	Task system: Invalid task ID
F40016	E_TASKSYS_NO_TASK_VIOLATION	Task system: No task violation
F40017	E_TASKSYS_INVALID_DUTYCYCLE	System function: Invalid duty cycle
F40018	E_TASKSYS_INVALID_CONTROL	Task system: Invalid task control
F40019	E_TASKSYS_INVALID_WDT_TRIGGER	System function: Invalid flex task trigger
F4001E	E_NO_VALID_LICENCE_INFORMA-TION_FOUND	System function: No valid license found on SD card or DHP11B. Contact SEW service
F40028	E_TIMER_INVALID_SCALEFACTOR	System function: Invalid scaling factor
F40064	E_IEC_MAINTENANCE_ARCHIVE_HAND-LING_ERROR	Maintenance: Error while opening/closing archive or while extracting or deflating
F40065	E_IEC_MAINTENANCE_SERVICE_RE-QUIRES_REBOOT	Maintenance: Reboot necessary for finishing the execution
F40066	E_IEC_MAINTENANCE_WRONG_DATA-FORMAT	Maintenance: Wrong data format
F4006E	E_SYSTEM_FIFO_IS_EMPTY_ERROR	FIFO: Is empty error
F4006F	E_SYSTEM_FIFO_IS_FULL_ERROR	FIFO: Is full error
F40070	E_SYSTEM_FIFO_INVALID_POINTER	FIFO: Invalid pointer

24796131/EN – 12/2017

Code	Error	Meaning
F40071	E_SYSTEM_FIFO_NOT_INITIALIZED	FIFO: Not initialized
F40072	E_SYSTEM_FIFO_INTERNAL_ERROR	FIFO: Internal error
F40073	E_SYS-TEM_FIFO_LENGTH_NOT_POWER_OF_TWO	FIFO: Value of max elements must be power of two
F400C8	E_SYS_DATA_BUF_INVALID_BUFFER	DATA_BUFFER: Invalid buffer
F400C9	E_SYS_DATA_BUF_INVALID_SUB_BUFFER	DATA_BUFFER: Invalid subbuffer
F400CA	E_SYS_DATA_BUF_INVALID_POINTER	DATA_BUFFER: Invalid pointer
F400CB	E_SYS_DATA_BUF_INVALID_BUF_SIZE	DATA_BUFFER: Invalid buffer size

13.11 Error: Data exchange

SEW ErrorCode 245 E_FW_LIB_DATAEXCHANGE

Code	Error	Meaning
F50064	E_TOO_MANY_BYTES_TO_COPY	Memcopy: Too many bytes to copy
F50065	E_CONFDATAEX_IOWORD_OUT_OF_BUF-FER	CONFDATAEX: IO word out of buffer range
F50066	E_CONFDATAEX_PARAMSET_SIZE_MIS-MATCH	CONFDATAEX: No match for parameter set size
F50067	E_CONFDATAEX_INVALID_PARAMSET_RE-FNUM	CONFDATAEX: Invalid reference number
F50068	E_CONFDATAEX_EMPTY_PARAMSET_BUF-FER	CONFDATAEX: Parameter set buffer is empty
F50069	E_CONF-DATAEX_NO_MATCH_FOR_PARAMSET	CONFDATAEX: The combination of SBUS ad-dress and node is not present in the PLC con-figuration
F5006A	E_CONFDATAEX_INVALID_BUFFER_TYPE	CONFDATAEX: Invalid buffer type
F5006B	E_CONFDATAEX_INVALID_POINTER	CONFDATAEX: Invalid pointer

13.12 Error: Firmware motion blocks

SEW ErrorCode 246 E_MOTION_FB: Error within firmware motion blocks

Code	Error	Meaning
F60001	E_MODULO_POS_X_GNUM_OR_LZ	ProfGen Modulo: In "Modulo positive absolute" mode, target pos is greater numerator or less than zero
F60002	E_MODULO_POS_DX_LZ	ProfGen Modulo: In "Modulo positive relative" mode, DeltaX has to be positive
F60003	E_MODULO_NEG_X_GNUM_OR_LZ	ProfGen Modulo: In "Modulo negative absolute" mode, target pos is greater numerator or less than zero

Code	Error	Meaning
F60004	E_MODULO_NEG_DX_GZ	ProfGen Modulo: In "Modulo positive relative" mode, DeltaX has to be negative
F60005	E_MODULO_SHORT_X_GNUM_OR_LZ	ProfGen Modulo: In "Modulo short absolute" mode, target pos greater numerator or less than zero
F60006	E_JERKTIME_GT2000MS	ProfGen Modulo: Invalid jerk time (value too big)
F60007	E_ERROR_SAMPLETIME	ProfGen Modulo: Sample time out of range
F60008	E_CAM_LENGTH	Cam: Cam length out of range
F60009	E_CAM_RANGE_X1_X2	Cam: X position out of range
F6000A	E_CAM_X1_X2	Cam: X position has to be strictly uniform
F6000B	E_CAM_V1	Cam: V1 has to be different from zero
F6000C	E_CAM_V2	Cam: V2 has to be different from zero
F6000D	E_CAM_Y1_Y2	Cam: Y1 has to be unequal to Y2
F6000E	E_CAM_LGS	Cam: Equation for polynomial 3 not solvable
F6000F	E_FILTERTAPS	Cam: Ratio between filter time and sample time is too big
F60010	E_MODULO_INVALID_MIN_MAX	FB: Modulo min/max value out of range
F60011	E_INVALID_ENCODER_RESOLUTION	FB: Encoder resolution out of range
F60012	E_INVALID_MODE	FB: Invalid mode
F60013	E_VENCODER_RAMP_PAR	VirtualEncoder: The parameters n32AmaxIn and n32DmaxIn have to be greater than zero
F60014	E_VENCODER_NMAX_PAR	VirtualEncoder: The parameters n32Nmax-PosIn and n32NmaxNegIn have to be greater than or equal to zero
F60015	E_VENCODER_DESTPOSIN	VirtualEncoder: Target position DestPosIn is not within n32ModuloMin ≤ DestPosIn < n32-ModuloMax
F60016	E_VENCODER_POSITIONINIIN	VirtualEncoder: Initialization position Posi-tionIniIn is not within the valid modulo range n32ModuloMin ≤ PositionIniIn < n32ModuloMax
F60017	E_INVALID_POINTER	FB: Invalid pointer
F60018	E_VENCODER_JERK_PAR	VirtualEncoder: The parameter jerk has to be greater than zero
F60019	E_VENCODER_CALC_ERROR1	VirtualEncoder: Error while calculating the stop-ping distance
F6001A	E_VENCODER_CALC_ERROR2	VirtualEncoder: Error while calculating the speed profile
F6001B	E_VENCODER_CALC_ERROR3	VirtualEncoder: Error while calculating the posi-tion profile
F6001C	E_VENCODER_CALC_ERROR4	VirtualEncoder: Error while calculating the stop-ping profile
F60064	E_SPLINE_RANGE	Spline: Value out of range or not uniform

Code	Error	Meaning
F60065	E_SPLINE_ILLEGAL_MODE	Spline: Invalid mode
F60066	E_SPLINE_ILLEGAL_INTERNAL_STATE	Spline: Invalid internal state
F60067	E_SPLINE_POINT_ERROR	Spline: Invalid point
F60068	E_SPLINE_INVALID_POINTER	Spline: Pointer is null or at limit of memory
F60069	E_SPLINE_INVALID_POINTLENGTH	Spline: Invalid number of points
F6006A	E_SPLINE_COEFF_INVALID	Spline: Data in structure invalid
F6006B	E_SPLINE_CALC_DIAG_ERROR	Spline: Error while solving the system of equations
F600C8	E_CAM_TRACK_MEM_ERROR	Cam track: Error while establishing instance
F600C9	E_CAM_TRACK_NO_FREE_ELEMENT	Cam track: Linking of cam to track failed
F600CA	E_CAM_TRACK_NO_DATA_SOURCE	Cam track: No data source set
F600CB	E_CAM_TRACK_INVALID_VALUE	Cam track: Invalid value
F600CC	E_CAM_TRACK_IS_COMPARING	Cam track: Action not possible during comparison
F600CD	E_CAM_TRACK_IS_INITIALIZING	Cam track: Action not possible during initialization
F6012C	E_PROFGENRR_JPG_CALC_A	ProfGenRR: Profile not possible
F6012D	E_PROFGENRR_XE_INVALID	ProfGenRR: X_e is null
F6012E	E_PROFGENRR_VX_INVALID	ProfGenRR: V_{xmax} is null
F6012F	E_PROFGENRR_TA_INVALID	ProfGenRR: Sample time ≤ null
F60130	E_PROFGENRR_MODE_INVALID	ProfGenRR: Invalid mode
F60131	E_PROFGENRR_VYMAX_INVALID	ProfGenRR: Vy_{max} ≤ null
F60132	E_PROFGENRR_VYMIN_INVALID	ProfGenRR: Vy_{min} ≤ null
F60133	E_PROFGENRR_AYMAX_INVALID	ProfGenRR: Ay_{max} ≤ null
F60134	E_PROFGENRR_DYMAX_INVALID	ProfGenRR: Dy_{max} ≤ null
F60135	E_PROFGENRR_JAY_INVALID	ProfGenRR: Jay ≤ null
F60136	E_PROFGENRR_JDY_INVALID	ProfGenRR: Jdy ≤ null
F60137	E_PROFGENRR_YE_INVALID	ProfGenRR: Y_e is null
F60138	E_PROFGENRR_MODULO_INVALID	ProfGenRR: Modulo min/max value out of range
F60139	E_PROFGENRR_DX_INVALID	ProfGenRR: $X_e < X_a$
F60140	E_POLYNOM5_XE_INVALID	Polynom5: X_e is null
F60141	E_POLYNOM5_MODULO_INVALID	Polynom5: Modulo min/max value out of range
F6014A	E_PROFGENXC_VX_INVALID	ProfGenXC: V_x ≤ null
F6014B	E_PROFGENXC_MODE_INVALID	ProfGenXC: Invalid mode
F6014C	E_PROFGENXC_YE_INVALID	ProfGenXC: Y_e is null
F6014D	E_PROFGENXC_XE_INVALID	ProfGenXC: X_e is null
F6014E	E_PROFGENXC_VYMAX_INVALID	ProfGenXC: Vymax ≤ null

Code	Error	Meaning
F6014F	E_PROFGENXC_VYMIN_INVALID	ProfGenXC: Vymin ≤ null
F60150	E_PROFGENXC_AYMAX_INVALID	ProfGenXC: Aymax ≤ null
F60151	E_PROFGENXC_DYMAX_INVALID	ProfGenXC: Dymax ≤ null
F60152	E_PROFGENXC_JYMAX_INVALID	ProfGenXC: Jymax ≤ null
F60153	E_PROFGENXC_VYINI_INVALID	ProfGenXC: Vyini out of range
F60154	E_PROFGENXC_PROFILE_INVALID	ProfGenXC: Profile not possible
F60155	E_PROFGENXC_MODULO_INVALID	ProfGenXC: Modulo min/max value out of range
F60156	E_PROFGENXC_DX_INVALID	ProfGenXC: $X_e \leq X_a$
F60157	E_PROFGENXC_VA_VE_INVALID	ProfGenXC: V_a and V_e <> null
F60158	E_PROFGENXC_VA_VE_RANGE	ProfGenXC: V_a or V_e out of range
F60159	E_PROFGENXC_VYLIMIT_PAR	ProfGenXC: Vymax ≤ Vymin
F60168	E_CAMFCT_DX_INVALID	CamFct: $X_e \leq X_a$
F60169	E_CAMFCT_VA_VE_INVALID	CamFct V_a and V_e <> null
F6016A	E_CAMFCT_POINTLENGTH_INVALID	CamFct: Point length out of range
F6016B	E_CAMFCT_X_INVALID	CamFct: Points are not strictly increasing uniformly
F6016C	E_CAMFCT_MODE_INVALID	CamFct: Invalid mode
F6016D	E_CAMFCT_VYMAX_INVALID	CamFct: Vymax ≤ null
F6016E	E_CAMFCT_VYMIN_INVALID	CamFct: Vymin ≤ null
F6016F	E_CAMFCT_AYMAX_INVALID	CamFct: Aymax ≤ null
F6017C	E_LPG_SIM_RAMP_PAR	LinProfGenSim: A_{max} or D_{max} ≤ zero
F6017D	E_LPG_SIM_NMAX_PAR	LinProfGenSim: V_{max} or V_{min} ≤ zero
F6017E	E_LPG_SIM_CALC_ERROR	LinProfGenSim: Error while calculating the position profile
F6017F	E_LPG_SIM_CALC_ERROR2	LinProfGenSim: Error while calculating the speed profile
F60180	E_LPG_SIM_MODE_INVALID	LinProfGenSim: Invalid mode
F60181	E_LPG_SIM_XDEST_INVALID	LinProfGenSim: X_{Dest} out of range
F60182	E_LPG_SIM_XINI_INVALID	LinProfGenSim: X_{Ini} out of range
F60190	E_PROFGENXYOPT_VLIMIT_PAR	ProfGenXYOpt: $V_{limit} \geq V_{max}$ or $V_{limit} \leq V_{min}$
F601A4	E_LINMOD_MODULO_INVALID	LinModulo: Modulo min/max value out of range
F601A5	E_LINMOD_POSINI_INVALID	LinModulo: Initialization position is not within the valid modulo range
F601A6	E_LINMOD_DENOM_INVALID	LinModulo: Denominator is ≤ 0
F601B8	E_PROFGENSPEEDCAM_V_1	ProfGenSpeedCam: Speed parameter of a cam ≤ 0
F601B9	E_PROFGENSPEEDCAM_V_2	ProfGenSpeedCam: Speed values of two cams are invalid

24796131/EN – 12/2017

Code	Error	Meaning
F601BA	E_PROFGENSPEEDCAM_POS	ProfGenSpeedCam: Cam positions are invalid
F67530	E_MOTION_FB_UNKNOWN_ERROR	Cam track: Unknown error

13.13 Error: Communication driver

SEW ErrorCode 247 E_COMMUNICATION_DRIVER

Code	Error	Meaning
F70001	E_RS485_DRV_UNKNOWN_RETCODE	RS485 driver: Unknown return value
F70002	E_RS485_DRV_ITF_NOT_SUPPORTED	RS485 driver: Interface does not exist
F70003	E_RS485_DRV_ILLEGAL_ADR	RS485 driver: Invalid address
F70004	E_RS485_DRV_MODE_FAULT	RS485 driver: Invalid mode
F70005	E_RS485_DRV_ILLEGAL_PDU	RS485 driver: Invalid process data length
F70006	E_RS485_DRV_SLAVE_EXIST	RS485 driver: Slave already exists
F70007	E_RS485_DRV_WRONG_BAUDRATE	RS485 driver: Invalid baud rate
F70008	E_RS485_DRV_SRV_IN_PROCESS	RS485 driver: Service still busy
F70009	E_RS485_DRV_NO_RESPONSE	RS485 driver: No response
F7000A	E_RS485_DRV_PRM_ERROR	RS485 driver: MOVILINK error. See MOVILINK return code for further information
F7000B	E_RS485_DRV_MAX_SLAVES	RS485 driver: Max number of slaves reached
F70064	E_SBUS_DRV_SCOM_CTRL_FAILED	SBUS driver: SCOM control failed
F70065	E_SBUS_1_DRV_PDO_TMO	SBUS driver: SBUS1 process data timeout
F70066	E_SBUS_2_DRV_PDO_TMO	SBUS driver: SBUS2 process data timeout
F70067	E_CAN_CH_RECURSIVE	SBUS driver: Recursion call

13.14 Error: Process image

SEW ErrorCode 248 E_COMMUNICATION_FW

Code	Error	Meaning
F80001	E_PDOTYPE_NOT_SUPPORTED	Process image: The selected PDO type is not supported by this interface
F80002	E_COMBUF_INVALID_CONTROL_OBJECT	Process image: Invalid control object
F80003	E_COMBUF_TOO_MANY_PARAMSETS	Process image: Max number of parameter sets reached
F80004	E_COMBUF_PARAMSET_ALLOCA-TION_FAILED	Process image: Memory allocation failed
F80005	E_REG_CYC_PD_DEST_FAILED	Process image: Link to process image failed

24796131/EN – 12/2017

13.15 Error: IEC, general

SEW ErrorCode 250 E_IEC_GENERAL

Code	Error	Meaning
FA0001	E_IEC_GENERAL_MAX_NUMBER_OF_AXIS	Max. number of axis reached
FA0002	E_IEC_GENERAL_INTERNAL_ERROR	Internal error (Contact SEW service)
FA0003	E_IEC_GENERAL_COM_NOT_READY	COM not ready
FA0004	E_IEC_GENERAL_INVALID_COM_NODE	COM node not valid
FA0005	E_IEC_GENERAL_INVALID_COM_ADR	COM address not valid
FA0006	E_IEC_GENERAL_SIMULA-TION_NOT_AVAILABLE	Simulation is not supported for this device
FA0007	E_IEC_GENERAL_INVERTER_NOT_REFER-ENCED	General: Execution of the function block is not allowed. The axis is not referenced!
FA0008	E_IEC_GENERAL_USE_OF_FB_NOT_AL-LOWED	General: The use of the function block is not al-lowed in the present state!
FA0009	E_IEC_GENERAL_CTRLINHIBIT_REQUIRED	General: Set control inhibit to use this function block
FA0010	E_IEC_GEN-ERAL_WRONG_MOVI_PLC_FIRMWARE	General: MOVI-PLC firmware does not fit the library version. Change MOVI-PLC firmware!
FA0011	E_IEC_GENERAL_SYNC_ALREADY_ESTAB-LISHED	General: Sync telegram is already established at this CAN node!
FA0012	E_IEC_GENERAL_INVALID_TECHNOLO-GIE_OPTION	General: The function cannot be used with the present MOVI-PLC technology version. Update MOVI-PLC technology!
FA0014	E_IEC_GENERAL_INVALID_TEC_ED-ITOR_CONFIGURATION	General: Invalid TecEditor configuration
FA0015	E_IEC_GENERAL_DIFFERENT_PLC_SERI-ANO_REQUIRED	General: Different MOVI-PLC is required. An-other device serial number required
FA0016	E_IEC_GEN-ERAL_WRONG_PLC_DEVICE_FAMILY	General: Wrong MOVI-PLC device family
FA0017	E_IEC_GENERAL_WRONG_SD_CARD	General: Wrong SD card in MOVI-PLC detec-ted
FA0018	E_IEC_GENERAL_INVALID_NODE	General: Invalid node
FA0019	E_IEC_GENERAL_TIMEOUT	General: Timeout during communication ser-vice
FA0020	E_IEC_INVALID_IO_CONFIG_IN_PLCCON-FIGURATION	General: The IO configuration in PLC configur-ation is invalid
FA0030	E_IEC_GENERAL_INVALID_REFER-ENCE_TYPE	General: Reference type is not supported
FA0031	E_IEC_GENERAL_FB_ENCODER_SET-TINGS_NOT_CALLED	General: The function block "EncoderSettings" is not called
FA0032	E_IEC_GENERAL_ENCODER_SET-TINGS_OUT_OF_RANGE	General: The encoder settings are out of range
FA0070	E_IEC_PARA-METER_VALUE_OUT_OF_RANGE	Parameters: Value out of Range

Code	Error	Meaning
FA0071	E_IEC_PARAMETER_INVALID_SELECTION	Parameters: Invalid selection at an input of the function block
FA0072	E_IEC_PARAMETER_INVALID_SERVICE	Parameters: Invalid service
FA0080	E_IEC_FILE_DOES_NOT_EXIST	File handling: File does not exist
FA0081	E_IEC_FILE_COULD_NOT_READ	File handling: No read access to the file
FA0082	E_IEC_FILE_COULD_NOT_WRITE	File handling: No write access to the file
FA0083	E_IEC_FILE_COULD_NOT_CLOSE	File handling: File could not be closed
FA0084	E_IEC_FILE_COULD_NOT_OPEN	File handling: File could not be opened
FA0085	E_IEC_FILE_WRONG_FILENAME	File handling: Invalid file name
FA0086	E_IEC_SDCARD_OR_FILE_IS_READONLY	File handling: SD card or file is read only
FA0087	E_IEC_FILE_ER-ROR_NOT_ENOUGH_FREE_SPACE	File handling: Not enough free space on file system
FA0088	E_IEC_GENERAL_SOFTWARELIMIT-SWITCH_RIGHT	General: SoftwareLimitSwitch right reached
FA0089	E_IEC_GENERAL_SOFTWARELIMIT-SWITCH_LEFT	General: SoftwareLimitSwitch left reached
FA008A	E_IEC_GENERAL_FB_NOT_SUPPOR-TED_WITH_THIS_TARGET	General: The selected function block is not supported with this MOVI-PLC target
FA008B	E_IEC_GENERAL_HARDWARELIMIT-SWITCH_RIGHT	General: HardwareLimitSwitch right reached
FA008C	E_IEC_GENERAL_HARDWARELIMIT-SWITCH_LEFT	General: HardwareLimitSwitch left reached
FA008D	E_IEC_FILE_TOO_LARGE_FOR_READOUT	File handling: The file size is too large, the ar-ray for save the data is limited to 8 kByte
FA0200	E_TEC_GENERAL_MULTIPLE_TECLINKS	Technology: Only one instance of the function block MC_LinkTec... allowed per axis
FA0201	E_TEC_GENERAL_INVALID_LINKSTATE	Technology: Function block may not be called from present LinkState (output of MC_Linlk-Tec...)
FA0202	E_TEC_GENERAL_NOT_LINKED	Technology: The function block may not be called before MC_LinkTec sets the output Done to TRUE
FA0203	E_TEC_GENERAL_NOT_INITIALIZED	Technology: The function block may not be called before the technology function is initial-ized (e.g. call MC_SetGearConfig_... first)
FA0204	E_TEC_GENERAL_SERVICE_NOT_IMPLE-MENTED	Technology: The selected service is not sup-ported
FA0220	E_TEC_VIRTUAL_IDENTIFIER_NOT_VALID	Technology virtual encoder: Send ID out of range
FA0221	E_TEC_VIRTUAL_MAX_NUMBER_AXIS	Technology virtual encoder: Max number of vir-tual axis reached
FA0222	E_TEC_VIRTUAL_INVALID_STATE	Technology virtual encoder: Function block must not be called from actual PLCOpenState (Output of MC_ConnectAxis_...)

Code	Error	Meaning
FA0223	E_TEC_VIRTUAL_LOG_ADR_NOT_INITIAL-IZED	Technology virtual encoder: VirtualAxis.LogAdr is not valid. Wait for MC_LinkTec_... Done set to TRUE
FA0230	E_TEC_CAM_INVALID_CAM_MODE	Technology cam control: Function block may not be called from present CamState (Output of MC_LinkTecCam_...)
FA0231	E_TEC_CAM_INVALID_CAM_TABLESCAL-ING	Technology cam control: Error during scaling of the cam table
FA0232	E_TEC_CAM_INVALID_SERVICE_REQUEST	Technology cam control: Selected service is not implemented
FA0233	E_TEC_CAM_MX_CURVE_AKTIV	Technology cam control: Call of function block is not allowed while a curve is active
FA0234	E_TEC_CAM_MX_SLAVESHIFT_TO_LOW	Technology cam control: The calculated Slave-shift is lower than the global Slaveshift
FA0240	E_TEC_GEAR_CLEAR_LAG_FAILED	Technology gear: Clear lag not done success-fully. Retrigger MC_GearClearLag_... !
FA0241	E_TEC_GEAR_INVALID_GEAR_MODE	Technology gear: Function block may not be called from present GearState (Output of MC_LinkTecGear_...)
FA0300	E_ETH_SOCKET_INITIALIZATION_FAILED	Ethernet: Socket initialization failed
FA0301	E_ETH_PORT_NOT_AVAILABLE	Ethernet: Port is not available
FA0302	E_ETH_NO_SERVER_FOUND	Ethernet: Server not found
FA0303	E_ETH_SOCKET_CLOSING_FAILED	Ethernet: Socket closing failed
FA0304	E_ETH_TRANSMISSION_FAILED	Ethernet: Transmission failed
FA0305	E_ETH_RECEPTION_FAILED	Ethernet: Reception failed
FA0306	E_ETH_DATA_LENGTH_OUT_OF_RANGE	Ethernet: Data length out of range
FA0307	E_ETH_SOCKET_NOT_IN_LISTEN-ING_MODE	Ethernet: Socket did not activate listening mode
FA0308	E_ETH_CONNECTION_DOES_NOT_EXIST	Ethernet: Connection does not exist
FA0309	E_ETH_SOCKET_TIMEOUT	Ethernet: Socket connection timeout
FA030A	E_ETH_CONNECTION_CLOSED	Ethernet: Connection lost
FA030B	E_ETH_NUMBER_OF_CLIENTS_NOT_SUP-PORTED	Ethernet: Function does not support number of clients
FA030C	E_ETH_ML_ERROR	Ethernet: MOVILINK error Additional info in "MvlReturnCode"
FA030D	E_ETH_ML_TIMEOUT	Ethernet: MOVILINK error Parameter timeout
FA030E	E_ETH_ML_GENERAL_ERROR	Ethernet: MOVILINK error Reply does not match request
FA030F	E_ETH_SET_SOCKET_OPTION_FAILED	Socket option not valid
FA0310	E_ETH_TRANSMIT_BUFFER_FULL	Transmit failed, transmit buffer is full
FA0400	E_SBUS_INVALID_NODE	SBUS: Data length out of range
FA0401	E_SBUS_TIMEOUT	SBUS: Timeout during SBUS Service

Code	Error	Meaning
FA0402	E_SBUS_COMMUNICA-TION_TASK_NOT_CALLED	SBUS: MC_CommunicationTask_xxx has to be called in a Task
FA0403	E_SBUS_SELECTED_SBUS_NOT_SUPPOR-TED	SBUS: The function block does not support the selected SBUS
FA0404	E_SBUS_INVALID_PROCESSDATA_UP-DATE_MODE	SBUS: Invalid process data update mode
FA0405	E_SBUS_SLAVE_CONFIGURATION_FAILED	SBUS: The slave could not be configured cor-rectly
FA0406	E_SBUS_MUXLIST_OVERFLOW	Overflow in multiplexer. Function block cannot be used.
FA0407	E_SBUS_SLAVES_IN_ERROR	Some slaves are in error state. The configura-tion in the PLC configuration has to match the hardware configuration
FA0408	E_SBUS_DATABUFFER_OVERFLOW	Overflow in multiplexer. Function block cannot be used.
FA0409	E_SBUS_PD_MASTER_NOT_ENABLED	The PD master function is not enabled in the PLC configuration. Check CAN module para-meters in the PLC configuration
FA0410	E_SBUS_INVALID_GATEWAY_ADDRESS	The SBUS gateway address does not match any configured SBUS gateway
FA0500	E_MSGHANDLER_INTERNAL_ERROR	MsgHandler: Internal error, please contact SEW service
FA0501	E_MSGHANDLER_NOT_LOWESTPRIORITY	MsgHandler: Prg MC_MsgHandler must be ex-ecuted in the task with the lowest priority (= highest priority value)
FA0502	E_MSGHANDLER_TASKBUFFER_OVER-FLOW	MsgHandler: Task buffer overflow, please reset errors and/or generate less messages
FA0503	E_MSGHANDLER_ARCHIVE_SIZE	MsgHandler: The MSGHAND-LER_ARCHIVE_SIZE is smaller than the "num-ber of tasks in use" x MSGHAND-LER_TASKBUFFER_SIZE
FA1000	E_IEC_GENERAL_PARAMETER_RW_ER-ROR	General: Warning, parameter request failed! Function block is possibly not executed cor-rectly
FA1001	E_IEC_GENERAL_SECOND_CALL_DUR-ING_ACTIVE	General: Warning: Call not allowed as long as the function block is still active
FA1001	E_IEC_DM_SECOND_CALL_DURING_ACT-IVE	Data management: Call not allowed as long as the function block is still active
FA1002	E_IEC_DM_DATASET_DOES_NOT_EXIST	Data management: Dataset does not exist in A box
FA1003	E_IEC_DM_WRONG_FIRMWARE_VERSION	Data management: Firmware not allowed
FA1004	E_IEC_DM_NO_PARTNUMBER_ASSIGNED	Data management: Part number must be as-signed
FA1005	E_IEC_DM_ABOX_HEADER_NOT_VALID	Data management: ABox header is not valid

24796131/EN – 12/2017

Code	Error	Meaning
FA1006	E_IEC_DM_ABOX_NO_CELLS_LEFT	Data management: No cells left in A box memory
FA1007	E_IEC_DM_ADDRESS_TO_HIGH	Data management: Requested address is too high
FA1008	E_IEC_DM_DATABLOCK_TO_LONG	Data management: Requested data block is too long
FA1009	E_IEC_DM_DATAB-LOCK_OUT_OF_MEMORY_AREA	Data management: Requested data block is out of memory area
FA1010	E_IEC_DM_DEVICE_NOT_SUPPORTED	Connected device is not supported
FA1011	E_IEC_DM_NUMBER_OF_DEVICE_EXCEED	Number of connected device exceed

13.16 Error: Motion function blocks

SEW ErrorCode 251 E_MOTION_LIB

Code	Error	Meaning
FB0030	E_MDX_CONNECTAXIS_NO_IN-VERTER_CONNECTED	Motion function block: No inverter connected, check SBUS connection, SBUS address and baud rate, go through DriveStartup again.
FB0031	E_MDX_CONNECTAXIS_CAN_ID_ERROR	Motion function block: A required CAN ID is already in use. Check if additional SCOM transmit IDs are greater than 2000
FB0032	E_MDX_CONNECTAXIS_CYCLIC_COMMU-NICATION	Motion function block: Cyclic communication interrupted between MOVI-PLC and inverter
FB0033	E_MDX_CONNECTAXIS_IPOS_DOWN-LOAD_ERROR	Motion function block: Error during communication driver (IPOS) download
FB0034	E_MDX_CON-NECTAXIS_WRONG_DEVICE_CONNECTED	Motion function block: Wrong inverter connected. Settings do not match an entry in the PLC configuration
FB0036	E_MDX_CON-NECTAXIS_WRONG_MDX_FIRM-WARE_VERSION	Motion function block: Use of this library is not possible with the current inverter firmware. Check library documentation
FB0060	E_MDX_POWER_INVERTER_NOT_READY	Motion function block: Inverter is not ready to be switched on. Inverter is in 24V operation or safety stop mode
FB0061	E_MDX_POWER_INVERTER_FAULT_STATE	Motion function block: Inverter is in fault state. Switch on is not possible
FB0070	E_MDX_MOTIONBLOCK_IN-VALID_DATA_PROFIL	Motion function block: Motion block cannot be executed with the selected data profile in the PLC configuration
FB0071	E_MDX_MOTION-BLOCK_LOG_ADR_NOT_INITIALIZED	Motion function block: Motion block called with invalid logical address (AXIS_REF)
FB0072	E_MDX_MOTIONBLOCK_IN-VALID_LOG_ADR	Motion function block: Motion block called with invalid logical address (AXIS_REF)

24796131/EN – 12/2017

Code	Error	Meaning
FB0073	E_MDX_MOTIONBLOCK_INVALID_STATE	Motion function block: Motion block must not be called from actual PLCOpen state
FB0074	E_MDX_MOTIONBLOCK_INVALID_OPERATING_MODE	Motion function block: The startup of the inverter does not support the current operating mode of the inverter
FB0075	E_MDX_MOTIONBLOCK_INVALID_INVERTER_STATUS	Motion function block: Motion block must not be called from actual inverter status
FB0076	E_MDX_MOTIONBLOCK_INVALID_VELOCITY	Motion function block: The value at the input velocity is invalid
FB0077	E_MDX_MOTIONBLOCK_INVALID_RAMP_TYPE	Motion function block: Motion block is not executable with the current ramp type
FB0090	E_MDX_PARAMCHANNEL_SEND_BUFFER_OVERFLOW	Parameter channel: Send buffer overflow
FB1000	E_MM_CONNECTAXIS_MULTIPLE_COM_ADR	Motion function block: Multiple com address
FB1002	E_MM_CONNECTAXIS_WRONG_INVERTER_TYPE	-
FB1003	E_MM_CONNECTAXIS_NO_INVERTER_CONNECTED	-
FB1004	E_MM_WRONG_MOVIMOT_TYPE	-
FB1005	E_MG_WRONG_PD_MODE	MG: For library use, DIP switch S2.1/S2.2 = ON
FB2000	E_CAM_INVALID_FILESIZE	-
FB2001	E_CAM_INCORRECT_NUMBER_OF_POINTS	-
FB2002	E_CAM_INVALID_CAM_CONFIGURATION	-
FB2003	E_CAM_CONFIGURATION_ACCESSED	-
FB2004	E_MODULO_PARAMETER_OVERFLOW	-
FB2005	E_CAM_SEGMENT_OUT_OF_RANGE	-
FB2006	E_CAM_CURVEFILE_NUMBER_OUT_OF_RANGE	Too many CurveFiles specified for this axis in MuMoCurveEditor
FB3000	E_ELVCD_POSITION_OUT_OF_VALID_ENCODER_RANGE	-
FB3001	E_ELVCD_SERIAL_NUMBER_CHANGED	-
FB3002	E_ELVCD_PARAMETER_CONFIG_CHANGED	-

24796131/EN – 12/2017

13.17 Error: External components

SEW ErrorCode 252 E_EXTERNAL_COMPONENTS

Code	Error	Meaning
FC0001	E_CAN_IO_MODULE_TIME_OUT_SBUS_INIT	CAN IO: Timeout during SBUS init

Code	Error	Meaning
FC0003	E_CAN_IO_MODULE_NO_CONNEC-TION_DURING_INIT_SEQ	CAN IO: No connection during init sequence
FC0004	E_CAN_IO_MODULE_WATCH_DOG_ERROR	CAN IO: Watch dog error. No connection to CAN IO module
FC0005	E_CAN_IO_MODULE_ILLEGAL_CONFIGUR-ATION	CAN IO: The CAN IO hardware does not match the PLC configuration settings
FC0006	E_CAN_IO_MODULE_CAN_ID_ERROR	CAN IO: CAN ID is already in use
FC0007	E_CAN_IO_MODULE_IN-VALID_BYTES_NUMBER	CAN IO: The value at the "Bytes" input is in-valid. Value has to be ≤ 4
FC0008	E_CAN_IO_MODULE_INVALID_PARAMET-ERS	CAN IO: Some parameters (Index, SubIndex, Bytes or WriteData) are not valid. Error code sent from the bus coupler is in the input Read-Data
FC0009	E_CAN_IO_MODULE_TIME_OUT_SBUS	CAN IO: Timeout. No connection to the CAN IO module
FC0010	E_CAN_IO_MODULE_NOT_IN_OPERA-TIONAL	CAN IO: Module not in operational mode
FC0100	E_GTW_SBUS_TIME_OUT	CAN IO: Timeout. No connection to the SEW Gateway
FC0101	E_GTW_TIMEOUT_SBUS_INIT	CAN IO: Timeout during SBUS init
FC0102	E_GTW_WAIT_FOR_AUTOSETUP	CAN IO: Wait for auto setup
FC0603	E_ETC_IO_MODULE_NO_CONNEC-TION_DURING_INIT_SEQ	ETC IO: No connection during init sequence
FC0605	E_ETC_IO_MODULE_ILLEGAL_CONFIGUR-ATION	ETC IO: The ETC IO hardware does not match the PLC configuration settings
FC0608	E_ETC_IO_MODULE_INVALID_PARAMET-ERS	ETC IO: Some parameters (Index, SubIndex, Bytes or WriteData) are not valid. Error code sent from the bus coupler is in the input Read-Data
FC0609	E_ETC_IO_MODULE_TIME_OUT_SBUS	ETC IO: Timeout. No connection to the IO module
FC0610	E_ETC_IO_MODULE_NOT_IN_OPERA-TIONAL	ETC IO: Module not in operational mode

13.18 Error: Kinematics

You can find these errors in the "Kinematics error codes" (→ 🗎 173) chapter.

24796131/EN – 12/2017

13.19 Error: Other application modules

SEW ErrorCode 254 E_APPLICATION_MODULES

Code	Error	Meaning
FE0001	E_APPMOD_GENERAL_INVALID_CONFIG-URATION	Wrong parameters in general configuration of the application module
FE0002	E_APPMOD_GENERAL_INVALID_COMPON-ENT_NUMBER	Component number (Axis, CamTrack, etc.) out of range
FE0003	E_APPMOD_GENERAL_IN-VALID_AXIS_HANDLER_TYPE	AxisHandlerLight has found an XML file for Ax-isHandlerMultiMotion
FE0010	E_APPMOD_LIMITS_MAX_SPEED	Max speed limit error
FE0011	E_APPMOD_LIMITS_MAX_ACCELERATION	Max acceleration limit error
FE0012	E_APPMOD_LIMITS_MAX_DECELERATION	Max deceleration limit error
FE0013	E_APPMOD_LIMITS_MAX_TRAVEL_DIS-TANCE	Max travel distance limit error (max 2147483647 [incr])
FE0014	E_APPMOD_LIMITS_MAX_TRAVEL_AREA	Max travel area limit error (target position must be between -2147483648 and 2147483647 [incr])
FE0015	E_APPMOD_TARGETPOS_OUT_OF_RANGE	Target position is outside the software limit switch area
FE0020	E_APPMOD_HOMING_NOT_AL-LOWED_WITH_ENABLE	Homing with the actual ReferenceTravelType is only possible when the axis is NOT enabled
FE0021	E_APPMOD_HOMING_NOT_AL-LOWED_WITHOUT_ENABLE	Homing with the actual ReferenceTravelType is only possible when the axis is enabled
FE0022	E_APPMOD_HOMING_NOT_ACTIV-ATED_REFERENCE_TRAVEL_TYPE_0	Homing is not activated. (ReferenceTravelType = 0)

24796131/EN – 12/2017

14 Appendix

14.1 SRL commands

This chapter is intended to provide information about the internal operating principle of the application module.

The various programs that are selected using program numbers 1 to 4 are implemented using SEW Robot Language (SRL). This also allows the programs to be displayed and simulated on a CCU controller. However, the user cannot change the programs. This chapter is intended only to aid transparency and understanding.

Normally, you do not need to know the SRL commands or to understand the SRL programs line by line. It is sufficient to use the data available in the fieldbus interface.

14.1.1 SRL control structures, general

The following underlying operators are available:

- **Disjunction:**
  OR link for truth statements A and B.

  Statement A **OR** statement B

- **Conditional statement:**
  The statement(s) is/are only executed if the condition is fulfilled.

  IF Condition

  Statement(s)

  ENDIF

- **Loop:**
  Executes the statement(s) until the condition is no longer fulfilled.

  WHILE Condition

  Statement(s)

  WEND

- **Wait condition:**
  The WAIT command leads to a pause until the condition is fulfilled.

  WAIT Condition

- **Goto statement:**
  The program goes to the next line that starts with "Jump flag:". Users can replace the "Jump flag" as they choose.

  GOTO Jump flag

  ...

  Jump flag: Instruction

- **Error message:**
  Generates an error message containing the "Text."

  ERROR Text

- **End of program:**
  Ends the program.

  END

24796131/EN – 12/2017

14.1.2 SRL motion commands

The following 3 travel commands that differ in terms of the interpolation type are available for moving to a position; see the "Interpolation types" (→ 🗎 30) chapter:

- Axis interpolation: **TARGET_AXIS** Coordinate system for the *target pose*
- Cartesian interpolation: **TARGET_CART** Coordinate system for the *target pose*
- Path interpolation (CP): **LIN** Coordinate system for the *target pose*

These travel commands each transmit the O8'ff. *target pose* in relation to a "coordinate system". In the case of the HandlingKinematics application module, this is always the kinematics coordinate system (KCS).

14.1.3 SRL commands for the segment parameter sets

The SRL command CALLF (call function = function call) is suitable for writing to the kinematics motion parameters with the correct and appropriate values from the configured segment parameter set. In this case, they are written to a buffer and not used again until the SRL motion command is called again.

CALLF `Nummer Pos**_SegParRec`

In this case, the CALLF number is differentiated based on the subgroup of the segment parameter sets. The numbering corresponds to the enumeration of the subgroups for the segment parameter sets:

- 11: Segment parameter set for axis interpolation (*TARGET_AXIS*)
  - Motion profiles for axes 1 to 6 (speed, acceleration, deceleration, and jerk time).
  - Synchronization of axes 1 to 6.
- 12: Segment parameter set for Cartesian interpolation (*TARGET_CART*)
  - Motion profiles for the Cartesian degrees of freedom X, Y, Z, and A (speed, acceleration, deceleration, and jerk time).
  - Synchronization of Cartesian degrees of freedom X, Y, Z, and A.
- 13: Segment parameter set for path interpolation (CP)
  - Motion profiles for the path (speed, acceleration, deceleration, and jerk time).
  - Speed profile, percentage, and distance limitation for blending.
- 14: Segment parameter set for rotary blending
  - Rotary blending
- 15: Segment parameter set for (Cartesian) sensor-based positioning (Touchprobe motion) (*TARGET_CART*)
  - Motion profiles for the Cartesian degrees of freedom X, Y, Z, and A (speed, acceleration, deceleration, and jerk time).
  - Remaining travel

The individual parameters are defined during configuration; see the "Overview of segment parameter sets" (→ 🗎 89) chapter.

14.1.4 SRL commands for the Touchprobe

To activate the inverter Touchprobe function, the **TP_PREPARE** command must first be executed. However, the Touchprobe signals are only interpreted and handled as such when one of the **TP_ACTIVATE_MEASURE** or **TP_ACTIVATE_MOTION** system functions is executed.

14.1.5 Variables

System input variables	Description
SysIn_Version	Version of the SRL interpreter
SysIn_TargetNoDistCheck	No check of the target distance within the *TARGET_AXIS/CART* travel command.
SysIn_RotBlendDist	Rotary blending distance
SysIn_CpBlendDist	CP blending distance
SysIn_CpPlusTargetCart	Path interpolation (CP) is executed with coordinated rotation
SysIn_CpIncludeABC	Path interpolation (CP) is executed with synchronized rotation
SysIn_TargetPos.A	*TARGET* travel command for the assigned target orientation A

System output variables	Description
SysOut_RotDist	Angle difference between the actual tool orientation and target orientation.
SysOut_TransDist	Translational distance to the target
SysOut_KinDone	HandlingKinematics has executed all of the travel commands and is at a standstill.
SysOut_KoordSys	Current reference coordinate system for the travel commands.
SysOut_QueueSize	Number of path segments still to be executed that have been reported to the kinematic model

Fieldbus variable	Description
TpMotionActive	O4[r]:5 *Touchprobe Motion active*
TpMeasureActive	O4[r]:5 *Touchprobe Measure active*
*End**Active*	O4[r]:8 – 11, … *End***
*Wait**Active*	O4[r]:12 – 15, … *Wait***
*Pos***	I8[r]ff. *Target position and target orientation***
*Pos**_TransBlendDist*	I17[r]ff. *Blending distance***
*Pos**.A*	I11[r]ff. *Target orientation***
Pos**_SegParRec	See the "SRL commands for the segment parameter sets" (→ 🗎 209) chapter.

14.2 SRL programs

14.2.1 Program 1: TARGET AXIS

Preparation

System settings		
Line	**Command**	**Description**
1	*SysIn_Version* := 1	Version of the SRL interpreter
2	*SysIn_TargetNoDistCheck* := 1	The check of the target distance (for blending) for the *target pose* within the **TARGET_AXIS** travel command is deactivated because it occurred outside of the command in the SRL program.

Touchprobe preparation		
Line	**Command**	**Description**
3	**IF** *TpMotionActive* **OR** *TpMeasureActive*	If at least one of the 2 Touchprobe functions is activated (O4^{r}:5 *Touchprobe Motion active* = *TRUE* or O4^{r}:7 *Touchprobe Measure active* = *TRUE*), the **TP_PREPARE** system function is executed, which activates the inverter Touchprobe functions as preparation for the *Touchprobe Measure* and/or *Touchprobe Motion* (row 10 and 232).
4	**TP_PREPARE**	
5	**ENDIF**	

Segment 1

Segment 1		
Line	**Command**	**Description**
6	**CALLF** 11 *Pos*01*_SegParRec*	Loading the subgroup of the segment parameter set of I8^{r} to 11^{r} *Target pose 1* for axis interpolation. It is used as a motion parameter for the subsequent travel command (subsequent row).
7	**TARGET_AXIS** KCS *Pos*01	Travel command for I8^{r} – 11^{r} *Target pose 1* in the kinematics coordinate system (KCS) when using the axis interpolation.
8	**CALLF** 12 *Pos*01*_SegParRec*	Loading the subgroup of the segment parameter set of I8^{r} to 11^{r} *Target pose 1* for Cartesian interpolation. It is not used subsequently.

Touchprobe Measure		
Line	**Command**	**Description**
9	**IF** *TpMeasureActive* = 1	If the signal O4^{r}:7 *Touchprobe Measure active* is*TRUE*, the **TP_ACTIVATE_MEASURE** system function is executed. The system waits for the next Touchprobe signal and calculates the current position on path segment 1 for it. If no Touchprobe signal was reported when the target point was reached, an error is issued; see the "Touchprobe Measure" (→ 🗎 64) chapter.
10	**TP_ACTIVATE_MEASURE**	
11	**ENDIF**	

24796131/EN – 12/2017

End 1		
Line	**Command**	**Description**
12	**IF** *End*01*Active* = 1	If the signal O4^r:8 *End 1 active* is set, the subgroup of the segment parameter set for I8^r to 11^r *Target pose 1* for the Touchprobe is loaded as a motion parameter for *Touchprobe Motion*. The program then jumps to the end of the program or to the *End* jump flag in row 231.
13	**CALLF** 15 *Pos*01*_SegParRec*	
14	**GOTO** End	
15	**ENDIF**	

Wait 1		
Line	**Command**	**Description**
16	**WAIT** *Wait*01*Active* = 0	The program pauses in the row until the signal O4^r:12 *Wait 1 active* is deactivated.

Segment 2

Blending 1 → 2		
Line	**Command**	**Description**
17	**CALLF** 14 *Pos*02*_SegParRec*	Loading the subgroup of the segment parameter set of I13^r to 16^r *Target pose 2* for the rotational blending. This subgroup is used for the rotary blending criterion in the subsequent row as a limit value for the angle difference (*SysIn_RotBlendDist*).
18	**WAIT** *SysOut_RotDist* ≤ *SysIn_RotBlendDist*	The program pauses in the row until the rotary blending criterion is fulfilled; that is, the angle difference to the *target orientation* of the tool is less than the limit value of the angle difference.
19	**WAIT** *SysOut_TransDist* ≤ *Pos*02*_TransBlendDist*	The program pauses in the row until the translational blending criterion is fulfilled; that is, the distance of the tool to the *target position* is less than I17^r *Blending distance to segment 2*.

Segment 2		
Line	**Command**	**Description**
20	**CALLF** 11 *Pos*02*_SegParRec*	Loading the subgroup of the segment parameter set of I13^r to 16^r *Target pose 2* for axis interpolation. It is used as a motion parameter for the subsequent travel command (subsequent row).
21	**TARGET_AXIS** KCS *Pos*02	Travel command for I13^r – 16^r *Target pose 2* in the kinematics coordinate system (KCS) when using the axis interpolation.
22	**CALLF** 12 *Pos*02*_SegParRec*	Loading the subgroup of the segment parameter set of I13^r to 16^r *Target pose 2* for Cartesian interpolation. It is not used subsequently.

End 2		
Line	**Command**	**Description**
23	**IF** *End*02*Active* = 1	If O4^r:9 *End 2 active* is set, the subgroup of the segment parameter set for I13^r to 17^r *Target pose 2* for the Touchprobe is loaded as a motion parameter for *Touchprobe Motion*. The program then jumps to the end of the program or to the *End* jump flag in row 231.
24	**CALLF 15** *Pos*02*_SegParRec*	
25	**GOTO** End	
26	**ENDIF**	

Wait 2		
Line	**Command**	**Description**
27	**WAIT** *Wait*02*Active* = 0	The program pauses in the row until the signal O4^r:13 *Wait 2 active* is deactivated.

Segment 3 to N-1

Rows 17 to 27 are repeated with the corresponding indexes for positions 03 to N−1... .

Last segment N

Blending N − 1 → N		
Line	**Command**	**Description**
224	**CALLF** 14 *Pos*N*_SegParRec*	Loading the subgroup of the segment parameter set for *Target pose N* for the rotary blending. This subgroup is used for the rotary blending criterion in the subsequent row as a limit value for the angle difference (*SysIn_RotBlendDist*).
225	**WAIT** *SysOut_RotDist* ≤ *SysIn_RotBlendDist*	The program pauses in the row until the rotary blending criterion is fulfilled; that is, the angle difference to the *target orientation* of the tool is less than the limit value of the angle difference.
226	**WAIT** *SysOut_TransDist* ≤ *Pos*N*_TransBlendDist*	The program pauses in the row until the translational blending criterion is fulfilled; that is, the distance of the tool to the *Target position* is smaller than the *Blending distance to segment N*.

Segment N		
Line	**Command**	**Description**
227	**CALLF** 11 *Pos*N*_SegParRec*	Loading the subgroup of the segment parameter set for *Target pose N* for axis interpolation. It is used as a motion parameter for the subsequent travel command (subsequent row).
228	**TARGET_AXIS** KCS *Pos*N	Travel command for *Target pose N* in the kinematics coordinate system (KCS) when using axis interpolation.
229	**CALLF** 12 *Pos*N*_SegParRec*	Loading the subgroup of the segment parameter set for *target pose N* for Cartesian interpolation. It is not used subsequently.
230	**CALLF** 15 *Pos*N*_SegParRec*	Loading the subgroup of the segment parameter set for *Target pose N* for the Touchprobe as a motion parameter for *Touchprobe Motion*.

24796131/EN – 12/2017

End

Touchprobe Motion		
Line	**Command**	**Description**
231	End: **IF** *TpMotionActive* = 1	This row is jumped to if one of the end signals is active. It is also executed if the program indicator has reached this point without a goto statement (End). If the signal is O4^r:5 *Touchprobe Motion active*, sensor-based positioning (*Touchprobe Motion* = **TP_ACTIVATE_MOTION**) is performed. In this case, the subgroup of the segment parameter set assigned in advance (in row 13, 24, … or 230) for the Touchprobe is used as a motion parameter. If no Touchprobe signal was reported when the last target point was reached, an error is issued; see the "Touchprobe Motion" (→ 🗎 61) chapter.
232	**TP_ACTIVATE_MOTION**	
233	**ENDIF**	

Stop		
Line	**Command**	**Description**
234	**Wait** *SysOut_RotDist* ≤ 0	The program pauses in the row until O10^r *Current orientation* corresponds to the *target orientation N*.
235	**Wait** *SysOut_TransDist* ≤ 0	The program pauses in the row until O7^r to 9^r *Current position* corresponds to *target position N*.
236	**Wait** *SysOut_KinDone* = 1	The program pauses in the row until all travel commands have been executed and the kinematic model is at a standstill.
237	**END**	The program is ended (O4^r:3 *Program complete* = *TRUE*).

14.2.2 Program 2: TARGET CART

Preparation

System settings		
Line	**Command**	**Description**
1	*SysIn_Version* := 1	Version of the SRL interpreter
2	*SysIn_TargetNoDistCheck* := 1	The check of the target distance (for blending) for the *target pose* within the **TARGET_CART** travel command is deactivated because it occurred outside of the command in the SRL program.

Touchprobe preparation		
Line	**Command**	**Description**
3	**IF** *TpMotionActive* **OR** *TpMeasureActive*	If at least one of the 2 Touchprobe functions is activated (O4^r:5 *Touchprobe Motion active* = *TRUE* or O4^r:7 *Touchprobe Measure active* = *TRUE*), the **TP_PREPARE** system function is executed, which activates the inverter Touchprobe functions as preparation for the *Touchprobe Measure* and/or *Touchprobe Motion* (row 9 and 212).
4	**TP_PREPARE**	
5	**ENDIF**	

Segment 1

Segment 1		
Line	**Command**	**Description**
6	**CALLF** 12 *Pos*01_*SegParRec*	Loading the subgroup of the segment parameter set of I8^r to 11^r *Target pose 1* for Cartesian interpolation. It is used as a motion parameter for the subsequent travel command (subsequent row).
7	**TARGET_CART** KCS *Pos*01	Travel command for I8^r – 11^r *Target pose 1* in the kinematics coordinate system (KCS) when using Cartesian interpolation.

Touchprobe Measure		
Line	**Command**	**Description**
8	**IF** *TpMeasureActive* = 1	If the signal O4^r:7 *Touchprobe Measure active* is*TRUE*, the **TP_ACTIVATE_MEASURE** system function is executed. The system waits for the next Touchprobe signal and calculates the *current position* on path segment 1 for it. If no Touchprobe signal was reported when the target point was reached, an error is issued; see the "Touchprobe Measure" (→ 🗎 64) chapter.
9	**TP_ACTIVATE_MEASURE**	
10	**ENDIF**	

End 1		
Line	**Command**	**Description**
11	**IF** *End*01*Active* = 1	If the signal O4^r:8 *End 1 active* is set, the segment parameter set subgroup for the Touchprobe is loaded as a motion parameter for *Touchprobe Motion*. The program then jumps to the end of the program or to the *End* jump flag in row 211.
12	**CALLF** 15 *Pos*01_*SegParRec*	
13	**GOTO** End	
14	**ENDIF**	

Wait 1		
Line	**Command**	**Description**
15	**WAIT** *Wait*01*Active* = 0	The program pauses in the row until the signal O4^r:12 *Wait 1 active* is deactivated.

Segment 2

Blending 1 → 2		
Line	**Command**	**Description**
16	**CALLF** 14 *Pos*02_*SegParRec*	Loading the subgroup of the segment parameter set of I13^r to 16^r *Target pose 2* for the rotational blending. This subgroup is used for the rotary blending criterion in the subsequent row as a limit value for the angle difference (*SysIn_RotBlendDist*).
17	**WAIT** *SysOut_RotDist* ≤ *SysIn_RotBlendDist*	The program pauses in the row until the rotary blending criterion is fulfilled; that is, the angle difference to the *target orientation* of the tool is less than the limit value of the angle difference.

24796131/EN – 12/2017

Blending 1 → 2		
Line	**Command**	**Description**
18	**WAIT** *SysOut_TransDist* ≤ *Pos*02*_TransBlendDist*	The program pauses in the row until the translational blending criterion is fulfilled; that is, the distance of the tool to the *target position* is less than I17^r *Blending distance to segment 2*.

Segment 2		
Line	**Command**	**Description**
19	**CALLF** 12 *Pos*02*_SegParRec*	Loading the subgroup of the segment parameter set of the signals I13^r to 16^r *Target pose 2* for Cartesian interpolation. It is used as a motion parameter for the subsequent travel command (subsequent row).
20	**TARGET_CART** KCS *Pos*02	Travel command for I13^r – 16^r *Target pose 2* in the kinematics coordinate system (KCS) when using the Cartesian interpolation.

End 2		
Line	**Command**	**Description**
21	**IF** *End*02*Active* = 1	If the signal O4^r:9 *End 2 active* is set, the segment parameter set subgroup for the Touchprobe is loaded as a motion parameter for *Touchprobe Motion*. The program then jumps to the end of the program or to the *End* jump flag in row 211.
22	**CALLF 15** *Pos*02*_SegParRec*	
23	**GOTO** End	
24	**ENDIF**	

Wait 2		
Line	**Command**	**Description**
25	**WAIT** *Wait*02*Active* = 0	The program pauses in the row until the signal O4^r:13 *Wait 2 active* is deactivated.

Segment 3 to N-1

Rows 16 to 25 are repeated with the corresponding indexes for positions 03 to N−1... .

Last segment N

Blending N − 1 → N		
Line	**Command**	**Description**
205	**CALLF** 14 *Pos*(N − 1)*_SegParRec*	Loading the subgroup of the segment parameter set for *Target pose N* for the rotary blending. This subgroup is used for the rotary blending criterion in the subsequent row as a limit value for the angle difference (*SysIn_RotBlendDist*).
206	**WAIT** *SysOut_RotDist* ≤ *SysIn_RotBlendDist*	The program pauses in the row until the rotary blending criterion is fulfilled; that is, the angle difference to the *target orientation* of the tool is less than the limit value of the angle difference.
207	**WAIT** *SysOut_TransDist* ≤ *Pos*(N)*_TransBlendDist*	The program pauses in the row until the translational blending criterion is fulfilled; that is, the distance of the tool to the *target position* is smaller than the *blending distance to segment N*.

24796131/EN – 12/2017

Segment N		
Line	**Command**	**Description**
208	**CALLF** 12 *Pos*(N)*_SegParRec*	Loading the subgroup of the segment parameter set for *target pose N* for Cartesian interpolation. It is used as a motion parameter for the subsequent travel command (subsequent row).
209	**TARGET_CART** KCS *Pos*(N)	Travel command for *target pose N* in the kinematics coordinate system (KCS) when using Cartesian interpolation.
210	**CALLF** 15 *Pos*(N)*_SegParRec*	Loading the subgroup of the segment parameter set for *target pose N* for the Touchprobe as a motion parameter for *Touchprobe Motion*.

End

Touchprobe Motion		
Line	**Command**	**Description**
211	End: **IF** *TpMotionActive* = 1	This row is jumped to if one of the end signals is active. It is also executed if the program indicator has reached this point without a goto statement (End). If the signal is O4^r:5 *Touchprobe Motion active*, sensor-based positioning (*Touchprobe Motion* = **TP_ACTIVATE_MOTION**) is performed. In this case, the subgroup of the segment parameter set assigned in advance (in row 12, 22, … or 210) for the Touchprobe is used as a motion parameter. If no Touchprobe signal was reported when the last target point was reached, an error is issued; see the "Touchprobe Motion" (→ 🗎 61) chapter.
212	**TP_ACTIVATE_MOTION**	
213	**ENDIF**	

Stop		
Line	**Command**	**Description**
214	**Wait** *SysOut_RotDist* ≤ 0	The program pauses in the row until O10^r *Current orientation* corresponds to the *target orientation N*.
215	**Wait** *SysOut_TransDist* ≤ 0	The program pauses in the row until O7^r to 9^r *Current position* corresponds to *target position N*.
216	**Wait** *SysOut_KinDone* = 1	The program pauses in the row until all travel commands have been executed and the kinematic model is at a standstill.
217	**END**	The program is ended (O4^r:3 *Program complete* = *TRUE*).

14.2.3 Program 3: LINEARLY coordinated

Preparation

System settings		
Line	**Command**	**Description**
1	*SysIn_Version* := 1	Version of the SRL interpreter
2	*SysIn_TargetNoDistCheck* := 1	Included for program 1 and 2 for the purpose of completeness and consistency only.

System settings		
Line	**Command**	**Description**
3	*SysIn_CpPlusTargetCart* := 1	The coordination of rotary degree of freedom A (Cartesian interpolation) for the translational path interpolation (CP) is activated.
4	*SysIn_CpIncludeABC*:= 0	The synchronization of rotary degree of freedom A (and B and C, which are not used) for the translational path interpolation (CP) is deactivated.

Touchprobe preparation		
Line	**Command**	**Description**
5	**IF** *TpMotionActive* OR *TpMeasureActive*	If at least one of the 2 Touchprobe functions is activated (O4^r:5 *Touchprobe Motion active* = *TRUE* or O4^r:7 *Touchprobe Measure active* = *TRUE*), the **TP_PREPARE** system function is executed, which activates the inverter Touchprobe function as preparation for the *Touchprobe Measure* or *Touchprobe Motion* (row 16 and 276).
6	**TP_PREPARE**	
7	**ENDIF**	

Coordinate system		
Line	**Command**	**Description**
8	**WHILE** *SysOut_KinCoordSys* <> 0	It is not possible to switch the coordinate system in the path interpolation (CP). It must therefore be checked whether the kinematics coordinate system (KCS) is – for whatever reason – not selected. In this case, an error is issued and the program is not processed further.
9	**ERROR** *Wrong coordinate system: use KCS*	
10	**WEND**	

Segment 1

Segment 1		
Line	**Command**	**Description**
11	**CALLF** 12 *Pos*01*_SegParRec*	Loading the subgroup of the segment parameter set of I8^r to 11^r *Target pose 1* for Cartesian interpolation. It is used as a motion parameter for orientation A in the subsequent travel command (row 12).
12	*SysIn_TargetPos.A*:= *Pos*01.*A*	I11^r *Target orientation 1* specification for the Cartesian interpolation of A. It is moved to immediately.
13	**CALLF** 13 *Pos*01*_SegParRec*	Loading the subgroup of the segment parameter set of I8^r to 11^r *Target pose 1* for path interpolation (CP). It is used as a motion parameter for the translational path in the subsequent travel command (row 14).
14	**LIN** KCS *Pos*01	Travel command for I8^r to 11^r *Target position 1* in the kinematics coordinate system (KCS) when using the path interpolation (CP) with the coordinated rotation of A.

Touchprobe Measure		
Line	**Command**	**Description**
15	**IF** *TpMeasureActive* = 1	If the signal O4^r:7 *Touchprobe Measure active* is*TRUE*, the **TP_ACTIVATE_MEASURE** system function is executed. The system waits for the next Touchprobe signal and calculates the *current position* on path segment 1 for it. If no Touchprobe signal was reported when the target point was reached, an error is issued; see the "Touchprobe Measure" (→ 🗎 64) chapter.
16	**TP_ACTIVATE_MEASURE**	
17	**ENDIF**	

End 1		
Line	**Command**	**Description**
18	**IF** *End*01*Active* = 1	If the signal O4^r: 8 *End 1 active* is set, the segment parameter set subgroup for the Touchprobe is loaded as a motion parameter for *Touchprobe Motion*. The program then jumps to the end of the program or to the *End* jump flag in row 275.
19	**CALLF** 15 *Pos*01*_SegParRec*	
20	**GOTO** End	
21	**ENDIF**	

Wait 1		
Line	**Command**	**Description**
22	**WAIT** *Wait*01*Active* = 0	The program pauses in the row until the signal O4^r:12 *Wait 1 active* is deactivated.

Segment 2

Blending 1 → 2		
Line	**Command**	**Description**
23	**CALLF** 14 *Pos*02*_SegParRec*	Loading the subgroup of the segment parameter set of I13^r to 16^r *Target pose 2* for the rotational blending. This subgroup is used for the rotary blending criterion in the subsequent row as a limit value for the angle difference (*SysIn_RotBlendDist*).
24	**WAIT** *SysOut_RotDist* ≤ *SysIn_RotBlendDist*	The program pauses in the row until the rotary blending criterion is fulfilled; that is, the angle difference to the *target orientation* of the tool is less than the limit value of the angle difference.
25	*SysIn_CpBlendDist* := *Pos*02*_TransBlendDist*	I17^r *Blending distance to segment 2* is transferred to the path interpolation (CP) for the translational blending criterion. The path is adjusted accordingly; see the "Blending" (→ 🗎 31) chapter.

Segment 2		
Line	**Command**	**Description**
26	**CALLF** 13 *Pos*02*_SegParRec*	Loading the subgroup of the segment parameter set of I13^r to 16^r *Target pose 2* for Cartesian interpolation (CP). It is used as a motion parameter for the translational path in the subsequent travel command (row 27).

Segment 2		
Line	**Command**	**Description**
27	**LIN** KCS *Pos*02	Travel command for I13^r to 16^r *Target position 2* in the kinematics coordinate system (KCS) when using the path interpolation (CP) with the coordinated rotation of A.
28	**WAIT** *SystOut_QueueSize* ≤ 1	The program pauses in the row until the movement is in the path segment for *target position 2*; that is, when blending begins.
29	**CALLF** 12 *Pos*02*_SegParRec*	Loading the subgroup of the segment parameter set of I13^r to 16^r *Target pose 2* for Cartesian interpolation. It is used as a motion parameter for orientation A in the subsequent travel command (row 30).
30	*SysIn_TargetPos.A* := *Pos*02*.A*	I16^r *Target orientation 2* A specification for the **TARGET_CART** Cartesian interpolation of A. It is moved to immediately.

End 2		
Line	**Command**	**Description**
31	**IF** *End*02*Active* = 1	If the signal O4^r:9 *End 2 active* is set, the segment parameter set subgroup for the Touchprobe is loaded as a motion parameter for *Touchprobe Motion*. The program then jumps to the end of the program or to the *End* jump flag in row 275.
32	**CALLF 15** *Pos*02*_SegParRec*	
33	**GOTO** End	
34	**ENDIF**	

Wait 2		
Line	**Command**	**Description**
35	**WAIT** *Wait*02*Active* = 0	The program pauses in the row until the signal O4^r:13 *Wait 2 active* is deactivated.

Segment 3 to N-1

Rows 23 to 35 are repeated with the corresponding indexes for positions 03 to N−1...

Last segment N

Blending N − 1 → N		
Line	**Command**	**Description**
266	**CALLF** 14 *Pos*(N)*_SegParRec*	Loading the subgroup of the segment parameter set for *Target pose N* for the rotary blending. This subgroup is used for the rotary blending criterion in the subsequent row as a limit value for the angle difference (*SysIn_RotBlendDist*).
267	**WAIT** *SysOut_RotDist* ≤ *SysIn_RotBlendDist*	The program pauses in the row until the rotary blending criterion is fulfilled; that is, the angle difference to the *target orientation* of the tool is less than the limit value of the angle difference.
268	*SysIn_CpBlendDist* := *Pos*(N)*_TransBlendDist*	The *Blending distance to segment N* is transferred to the path interpolation (CP) for the translational blending criterion. The path is adjusted accordingly; see the "Blending" (→ 🗎 31) chapter.

Segment N		
Line	**Command**	**Description**
269	**CALLF** 13 *Pos*(N)*_SegParRec*	Loading the subgroup of the segment parameter set for *target pose N* for the path interpolation (CP). It is used as a motion parameter for the translational path in the subsequent travel command (row 270).
270	**LIN** KCS *Pos*(N)	Travel command for *target position N* in the kinematics coordinate system (KCS) when using the path interpolation (CP) with the coordinated rotation of A.
271	**WAIT** *SystOut_QueueSize* ≤ 1	The program pauses in the row until the movement is in the path segment for *target position N*; that is, when blending begins.
272	**CALLF** 12 *Pos*(N)*_SegParRec*	Loading the subgroup of the segment parameter set for *target pose N* for Cartesian interpolation. It is used as a motion parameter for orientation A in the subsequent travel command (row 273).
273	*SysIn_TargetPos.A* := *Pos*(N).*A*	I16^r *Target orientation N* A specification for the Cartesian interpolation of A. It is moved to immediately.
274	**CALLF** 15 *Pos*(N)*_SegParRec*	Loading the subgroup of the segment parameter set for *target pose N* for the Touchprobe as a motion parameter for *Touchprobe Motion*.

End

Touchprobe Motion		
Line	**Command**	**Description**
275	End: **IF** *TpMotionActive* = 1	This row is jumped to if one of the end signals is active. It is also executed if the program indicator has reached this point without a goto statement (End). If the signal is O4^r:5 *Touchprobe Motion active*, sensor-based positioning (*Touchprobe Motion* = **TP_ACTIVATE_MOTION**) is performed. In this case, the segment parameter set subgroup assigned in advance (in row 19, 32, … or 274) for the Touchprobe is used as a motion parameter. If no Touchprobe signal was reported when the last target point was reached, an error is issued; see the "Touchprobe Motion" (→ 🗎 61) chapter.
276	**TP_ACTIVATE_MOTION**	
277	**ENDIF**	

Stop		
Line	**Command**	**Description**
278	**Wait** *SysOut_RotDist* ≤ 0	The program pauses in the row until O10^r *Current orientation* corresponds to the *target orientation N*.
279	**Wait** *SysOut_KinDone* = 1	The program pauses in the row until all travel commands have been executed and the kinematic model is at a standstill.
280	**END**	The program is ended (O4^r:3 *Program complete* = *TRUE*).

14.2.4 Program 4: LINEARLY synchronized

Preparation

System settings		
Line	**Command**	**Description**
1	*SysIn_Version* := 1	Version of the SRL interpreter
2	*SysIn_TargetNoDistCheck* := 0	Included for program 1 and 2 for the purpose of completeness and consistency only.
3	*SysIn_CpPlusTargetCart* := 0	The coordination of rotary degree of freedom A (Cartesian interpolation) for the translational path interpolation (CP) is deactivated.
4	*SysIn_CpIncludeABC* := 1	The synchronization of rotary degree of freedom A (and B and C, which are not used) for the translational path interpolation (CP) is activated.

Touchprobe preparation		
Line	**Command**	**Description**
5	**IF** *TpMotionActive* OR *TpMeasureActive*	If at least one of the 2 Touchprobe functions is activated ($O4^r$:5 *Touchprobe Motion active* = *TRUE* or $O4^r$:7 *Touchprobe Measure active* = *TRUE*), the **TP_PREPARE** system function is executed, which activates the inverter Touchprobe function as preparation for the *Touchprobe Measure* and/or *Touchprobe Motion* (row 15 and 199).
6	**TP_PREPARE**	
7	**ENDIF**	

Coordinate system		
Line	**Command**	**Description**
8	**WHILE** *SysOut_KinCoordSys* <> 0	It is not possible to switch the coordinate system in the path interpolation (CP). It must therefore be checked whether the kinematics coordinate system (KCS) is – for whatever reason – not selected. In this case, an error is issued and the program is not processed further.
9	**ERROR** *Wrong coordinate system: use KCS*	
10	**WEND**	

Segment 1

Segment 1		
Line	**Command**	**Description**
11	**CALLF** 12 *Pos*01*_SegParRec*	Loading the subgroup of the segment parameter set of $I8^r$ to 11^r *Target pose 1* for Cartesian interpolation. It is used as a motion parameter for orientation A in the subsequent travel command (row 14).
12	**CALLF** 13 *Pos*01*_SegParRec*	Loading the subgroup of the segment parameter set of $I8^r$ to 11^r *Target pose 1* for path interpolation (CP). It is used as a motion parameter for the translational path in the subsequent travel command (row 13).
13	**LIN** KCS *Pos*01	Travel command for $I8^r$ to 11^r *Target position 1* in the kinematics coordinate system (KCS) when using the path interpolation (CP) with the synchronized rotation of A.

Touchprobe Measure		
Line	**Command**	**Description**
14	**IF** *TpMeasureActive* = 1	If the signal O4^{r}:7 *Touchprobe Measure active* is*TRUE*, the **TP_ACTIVATE_MEASURE** system function is executed. The system waits for the next Touchprobe signal and calculates the *current position* on path segment 1 for it. If no Touchprobe signal was reported when the target point was reached, an error is issued; see the "Touchprobe Measure" (→ 🗎 64) chapter.
15	**TP_ACTIVATE_MEASURE**	
16	**ENDIF**	

End 1		
Line	**Command**	**Description**
17	**IF** *End*01*Active* = 1	If the signal O4^{r}: 8 *End 1 active* is set, the segment parameter set subgroup for the Touchprobe is loaded as a motion parameter for *Touchprobe Motion*. The program then jumps to the end of the program or to the *End* jump flag in row 198.
18	**CALLF** 15 *Pos*01*_SegParRec*	
19	**GOTO** End	
20	**ENDIF**	

Wait 1		
Line	**Command**	**Description**
21	**WAIT** *Wait*01*Active* = 0	The program pauses in the row until the signal O4^{r}:12 *Wait 1 active* is deactivated.

Segment 2

Blending 1 → 2		
Line	**Command**	**Description**
22	*SysIn_CpBlendDist* := *Pos*02*_TransBlendDist*	I17^{r} *Blending distance to segment 2* is transferred to the path interpolation (CP) for the translational blending criterion. The path is adjusted accordingly; see the "Blending" (→ 🗎 31) chapter. There is no rotary blending criterion for program 4.

Segment 2		
Line	**Command**	**Description**
23	**CALLF** 12 *Pos*02*_SegParRec*	Loading the subgroup of the segment parameter set of I13^{r} to 16^{r} *Target pose 2* for Cartesian interpolation. It is used as a motion parameter for orientation A in the subsequent travel command (row 25).
24	**CALLF** 13 *Pos*02*_SegParRec*	Loading the subgroup of the segment parameter set of I13^{r} to 16^{r} *Target pose 2* for Cartesian interpolation (CP). It is used as a motion parameter for the translational path in the subsequent travel command (row 25).
25	**LIN** KCS *Pos*02	Travel command for I13^{r} to 16^{r} *Target pose 2* in the kinematics coordinate system (KCS) when using the path interpolation (CP) with the synchronized rotation of A.

24796131/EN – 12/2017

End 2		
Line	**Command**	**Description**
26	**IF** *End*02*Active* = 1	If the signal O4^{r}:9 *End 2 active* is set, the segment parameter set subgroup for the Touchprobe is loaded as a motion parameter for *Touchprobe Motion*. The program then jumps to the end of the program or to the *End* jump flag in row 198.
27	**CALLF 15** *Pos*02*_SegParRec*	
28	**GOTO** End	
29	**ENDIF**	

Wait 2		
Line	**Command**	**Description**
30	**WAIT** *Wait*02*Active* = 0	The program pauses in the row until the signal O4^{r}:13 *Wait 2 active* is deactivated.

Segment 3 to N-1

Rows 22 to 30 are repeated with the corresponding indexes for positions 03 to N−1... .

Last segment N

Blending N − 1 → N		
Line	**Command**	**Description**
31	*SysIn_CpBlendDist* := *Pos*(N)*_TransBlendDist*	The *Blending distance to segment N* is transferred to the path interpolation (CP) for the translational blending criterion. The path is adjusted accordingly; see the "Blending" (→ 🗎 31) chapter.

Segment N		
Line	**Command**	**Description**
194	**CALLF** 12 *Pos*(N)*_SegParRec*	Loading the subgroup of the segment parameter set for *target pose N* for Cartesian interpolation. It is used as a motion parameter for orientation A in the subsequent travel command (row 196).
195	**CALLF** 13 *Pos*(N)*_SegParRec*	Loading the subgroup of the segment parameter set for *target pose N* for the path interpolation (CP). It is used as a motion parameter for the translational path in the subsequent travel command (row 196).
196	**LIN** KCS *Pos*(N)	Travel command for *target pose N* in the kinematics coordinate system (KCS) when using the path interpolation (CP) with the synchronized rotation of A.
197	**CALLF** 15 *Pos*(N)*_SegParRec*	Loading the subgroup of the segment parameter set for *target pose N* for the Touchprobe as a motion parameter for *Touchprobe Motion*.

24796131/EN – 12/2017

End

Touchprobe Motion		
Line	**Command**	**Description**
198	End: **IF** *TpMotionActive* = 1	This row is jumped to if one of the end signals is active. It is also executed if the program indicator has reached this point without a goto statement (End). If the signal is O4^{r}:5 *Touchprobe Motion active*, sensor-based positioning (*Touchprobe Motion* = **TP_ACTIVATE_MOTION**) is performed. In this case, the subgroup of the segment parameter set assigned in advance (in row 18, 27, … or 197) for the Touchprobe is used as a motion parameter. If no Touchprobe signal was reported when the last target point was reached, an error is issued; see the "Touchprobe Motion" (→ 🗎 61) chapter.
199	**TP_ACTIVATE_MOTION**	
200	**ENDIF**	

Stop		
Line	**Command**	**Description**
201	**Wait** *SysOut_KinDone* = 1	The program pauses in the row until all travel commands have been executed and the kinematic model is at a standstill.
202	**END**	The program is ended (O4^{r}:3 *Program complete* = *TRUE*).

14.3 Terminal assignment

The following table shows the terminal assignment of each unit when the default assignment and hardware limit switches are activated.

Input terminal	MOVIDRIVE® B / MOVIAXIS® / MOVITRAC® LTX
DI00	/Controller inhibit
DI01	Enable/stop
DI02	(Error reset; also possible via process data)
DI03	Reference cam
DI04	/Hardware limit switch positive
DI05	/Hardware limit switch negative

14.4 Constellations of the kinematic models

- KIN_SCARA_RR_XY_M20: Constellations 1, 2
  - The elbow is folded through to one side or the other.
- KIN_SCARA_RRR_XYA_M20: Constellations 1, 2
  - The elbow is folded through to one side or the other.
- KIN_SCARA_RRL_XYZ_M20: Constellations 1, 2
  - The elbow is folded through to one side or the other.
- KIN_SCARA_LRRR_XYZA_M10: Constellations 1, 2
  - The elbow is folded through to one side or the other.

24796131/EN – 12/2017

- KIN_SCARA_RRRL_XYZA_M10: Constellations 1, 2
  - The elbow is folded through to one side or the other.
- KIN_SCARA_RRRR_XYZA_M60: Constellations 1, 2, 3, 4
  - The elbow is folded through to one side or the other.
  - In constellations 3 and 4, the kinematic model is also in the "upside-down position" (TCP is on the opposite side of the rotary axis from A1). Each elbow is folded through to one side or the other.
- KIN_SCARA_RRRR_XYZA_M65: Constellations 1, 2, 3, 4
  - See KIN_SCARA_RRRR_XYZA_M60
- KIN_DELTA_LL_XY_M10: Constellations 1, 2
  - The two arms are folded through to the inside/outside.
- KIN_DELTA_RR_XY_M20: Constellations 1, 2, 3, 4
  - 1: The elbows of the two arms are folded through to the outside; see the figure in the "DELTA" (→ 🗎 25) chapter.
  - 2 and 3: One elbow is folded through to the outside and one to the inside.
  - 4: The elbows of both arms are folded through to the inside.
- KIN_TRIPOD_LLLR_XYZA_M10: Constellations 1, 2, 3, 4, 5, 6, 7, 8
  - Generally, only constellation 1 is used for tripods. In this constellation, all three arms are pointing downwards; see the figure in the "TRIPOD" (→ 🗎 26) chapter.
  - In constellations 2 – 8, one or several arms are directed upwards.
- KIN_TRIPOD_RRR_XYZ_M10: Constellations 1, 2, 3, 4, 5, 6, 7, 8
  - For tripods, only constellation 1 is common. In this constellation, all three elbows are folded through to the outside.
  - In constellations 2 – 8, one or more elbows are folded through to the inside.
- KIN_TRIPOD_RRRR_XYZA_M10: Constellations 1, 2, 3, 4, 5, 6, 7, 8
  - See KIN_TRIPOD_RRR_XYZ_M10
- KIN_MIXED_RLLR_XYZA_M10: Constellations 1, 2
  - The flange (FCS) lies along the second linear axis (A3) on one of the sides of the first rotary axis (A1).
- All other standard kinematic models are always in constellation 1.

24796131/EN – 12/2017

Glossary

ACS

Abbreviation for axis coordinate system.

Articulated kinematic model

Kinematic model of an articulated arm robot. This model has 5 to 6 degrees of freedom and multiple rotary joints.

Axis arrow

Arrow in the 3D simulation that indicates the work area (SWLS), the axis zero (0 motor increments), and the direction in which the motor increments increase (with one-to-one assignment of a motor to an axis).

Axis coordinate system

A coordinate system in which the position of the kinematics and the auxiliary axes is described by the axis coordinates.

Axis zero

Position of an axis with 0 motor increments (with clear assignment of a motor to an axis).

Blending

Smooth transition of the motion path to the next path segment.

Cartesian degree of freedom

Extent to which the position and orientation can be changed in a Cartesian coordinate system; defined as follows in robotics: Extent to which the position and orientation of the tool center point (TCP), expressed with the translational coordinates X, Y and Z and the rotational coordinates A, B and C, can be changed.

Cartesian Gantry

Kinematic model with at least 2 or 3 linear axes that are positioned along the axes of a Cartesian coordinate system.

CP

Abbreviation for continuous path; see path interpolation.

Delta kinematics

Parallel kinematics characterized by kinematic partial chains in a triangular structure.

Hexapod kinematics

Parallel kinematics characterized by 6 parallel kinematic chains.

Interpolation

Defining a certain motion path between two points.

KCS

Abbreviation for kinematics coordinate system.

Kinematic chain

Series of parts connected by joints.

Kinematic model

A mathematical model of a robot's kinematics. It defines the movement options of the bodies, joints and flange using equations of motion.

Kinematic model

Kinematics describes how a robots mechanical components move in space. It is described with the following variables: Position, speed, acceleration, orientation, angular velocity and angular acceleration of the bodies, joints and tool.

Kinematics coordinate system

Coordinate system that is generally permanently connected to the socket of the kinematic model. KCS is defined with reference to the world coordinate system (WCS).

Mixed kinematic model

Kinematic models that does not clearly feature the characteristics of any other kinematic models.

Motion path

The continuous sequence of poses of a robot's tool with reference to time, i.e. at which pose the tool is at which point of time. The motion path can be described by: Time, position, velocity, acceleration, orientation, angular velocity and angular acceleration of the tool.

Motion sequence

A part of a motion path with the start and end poses at a standstill.

Path

The continuous sequence of poses for a robot's tool without reference to time.

Path interpolation

Interpolation type in which the path is described geometrically in space (e.g. line/circle segments and geometrically defined blending ranges).

Path point

A point on a path specified by the pose of a robot's tool at this point.

Path segment

The movement section from one path point to the next.

PD

Abbreviation for process data words

Quadropod kinematics

Parallel kinematics characterized by 4 parallel kinematic chains.

Roller Gantry

Kinematic model in which two translational degrees of freedom are controlled by two generally stationary drives using a revolving toothed belt. Additional degrees of freedom can be added upstream or downstream of this assembly.

SCARA kinematics

Abbreviation for Selective Compliance Assembly Robot Arm. Kinematic chain that, like a human arm, is characterized by shoulder and elbow joints which, in robotics, are two rotary joints with parallel rotary axes.

Software limitation

A limitation that is monitored by the controller's software. Violation of this limit generally results in the error status. Software limitations are intended to prevent violation of the actual limits of the hardware and/or mechanical components (e.g. hardware limit switches). One example is limitation of the working space.

SRL

Abbreviation for SEW Robot Language.

TCP

Abbreviation for tool center point.

Tool

The last element in a kinematic chain, which is attached to the robot flange and which the robot uses to perform its task.

Tool operating point

Point on the robot's tool that is moved to the target coordinates by the controller. It corresponds to the origin of the tool coordinate system in the kinematic model.

Touchprobe

A switch that is activated by a sensor and issues a Boolean statement as to whether a pose has been reached.

Touchprobe function

Recording the position of the axes at the time of the Touchprobe signal. The kinematic control uses these axis positions to calculate the associated Cartesian pose of the tool.

Tripod kinematics

Parallel kinematics characterized by a tripod.

Wait point

A path point at which the robot stops until it is released to continue its movement.

WCS

Abbreviation for World Coordinate System.

Work envelope

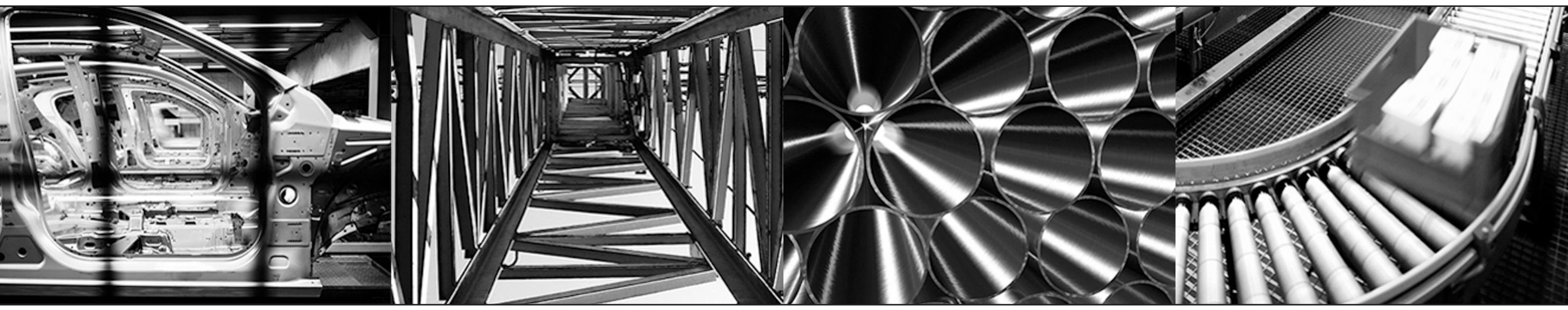
Configured area in which the kinematics can move. It is limited by the kinematic model, kinematics limitations, axis limits and Cartesian limits.

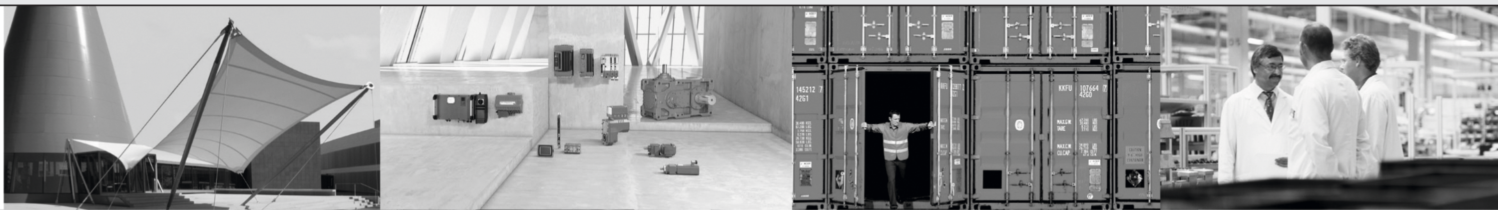
World coordinate system

Stationary coordinate system using the world as a reference.

Index

24796131/EN – 12/17





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