SIEMENS

SIMOCRANE

Basic Technology

Operating Instructions

Valid for
SIMOCRANE Basic Technology V3.0

Hardware:
– SIMOTION D435-2 DP/PN as of version V4.3 SP1
– SINAMICS as of version V4.5

06/2012

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Ⓟ 08/2012 Technical data subject to change

Preface

This document is part of the SIMOCRANE Basic Technology package. It describes a standard application solution for cranes. This standard application solution has been available since mid 2007. It is suitable for both "ready-to-run" (parameterization only) and "ready-to-apply" (adapted by users) applications.

Prerequisite (area of validity)

This manual is valid for use with the following product versions:

Hardware:

- SIMOTION D435-2 DP/PN as of version V4.3 SP1
- SINAMICS as of version V4.5

Software:

- SIMOTION SCOUT as of version V4.3 SP1
- SIMOCRANE Basic Technology as of version V3.0
- Drive Control Chart (DCC) options package as of version V2.2 SP1

Additional information

Siemens product support

The latest information about SIMOTION products, product support, and FAQs can be found on the Internet here (http://support.automation.siemens.com/WW/view/en/10805436/130000).

The latest information about SINAMICS products, product support, and FAQs can be found on the Internet here (http://support.automation.siemens.com/WW/view/en/13305690/130000).

The latest information about SIMOCRANE products, product support, and FAQs can be found on the Internet here (http://support.automation.siemens.com/WW/view/de/10807397/130000).

Crane Application notes:

On the Internet here (http://support.automation.siemens.com/CN/view/zh/48342008/136000)
.

Product support for SIMOCRANE

The following addresses provide support for your SIMOCRANE products:

- Support request on the Internet:
  - http://support.automation.siemens.com
- Europe hotline
  - Tel.: +49 (0) 911 895 7 222
  - Fax: +49 (0) 911 895 7 223
  - E-mail: support.automation@siemens.com
- America hotline
  - Tel.: +1 423 262 5710
  - Fax: +1 423 262 2231
  - E-mail: support.america.automation@siemens.com
- Asia/Pacific hotline
  - Tel.: +86 10 6475 7575
  - Fax: +86 10 6474 7474
  - E-mail: support.asia.automation@siemens.com

Application support for SIMOCRANE

For additional customer-specific requirements and applications, please contact the following e-mail address:
applications.cranes.aud@siemens.com

Further assistance

We also offer introductory courses to help you familiarize yourself with SIMOCRANE Basic Technology. You can find more information here (www.siemens.nl/training/cranes).

If you have any additional questions please contact your local Siemens sales person.

Table of Contents

System overview 1

Introduction

The technology module T300 with MASTERDRIVES drive for crane-specific technology (basic technology) was used in the previous crane application solution. Since mid-2007, the new sector solution SIMOCRANE Basic Technology is available in the Siemens Cranes business area. The new platform based on SIMATIC + SIMOTION + SINAMICS is available for harbor and industrial cranes. All tasks are clearly segregated so that SIMOTION handles all crane technologies, e.g. basic technology, sway control, etc., which can be structured systematically on one another. The hardware configuration using an STS crane with new platform is shown in the following diagram as an example.

SIMATIC
SIMOTION
SINAMICS

Figure 1-1 The SIMOCRANE platform

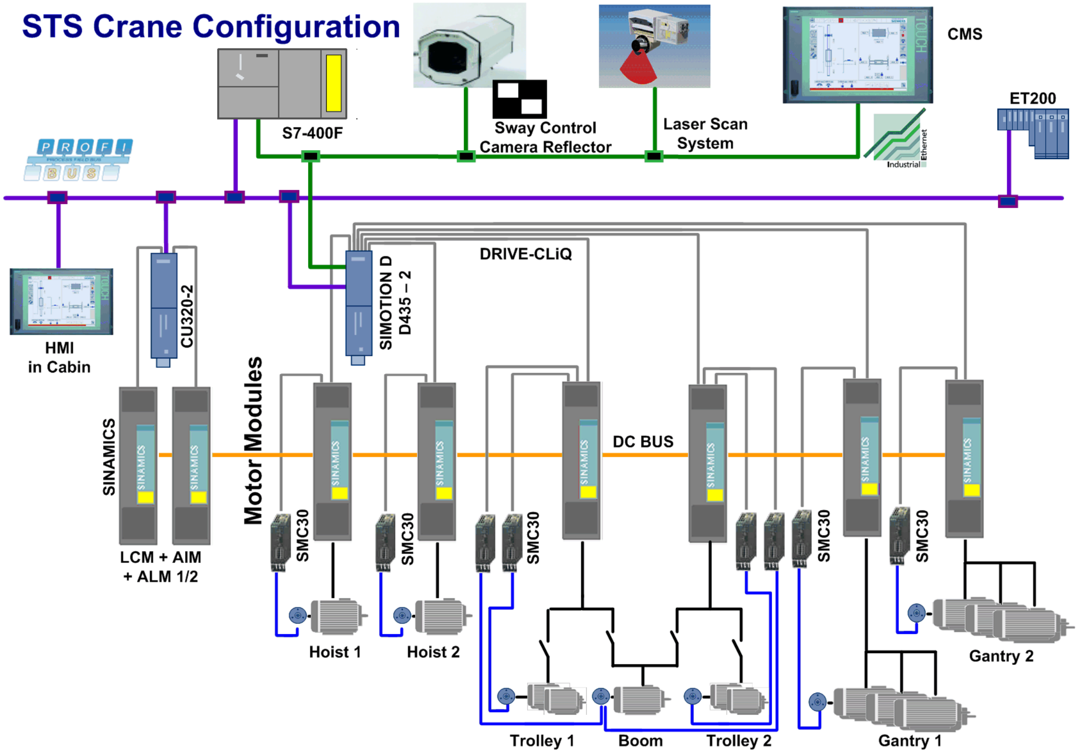


Figure 1-2 STS configuration with the new platform using SIMOTION D435-2 as example

In comparison to the solutions based on T300 and/or SIMATIC with MASTERDRIVES, the new solution on SIMOTION D includes the following features:

- The basic technology covers the motion control of all the main drives of a crane. It includes the functions of all six T300 modules (hoist, gantry, gantry-slave, slewing gear, holding gear and closing gear) and more (e.g. boom etc.); for information on the scope of the standard application, refer to Chapter STS crane (ship-to-shore) (Page 498).
- All of the functions proven in practice have been integrated into the new platform. All methods have been updated. The new requirements were taken into account.
- SIMOTION D offers a centralized solution. The internal data exchange so replaces the otherwise usual inter-axis communication.
- New closed-loop control concept for synchronous operation and positioning with position controller.
- Possibility of adapting to customer-specific requirements.
- A package permits both ready-to-run (only parameterization) as well as also ready-to-apply (adapted by the user) applications.

Special feature V3.0

The version V2.0 SP2 was released at end of 2010. Its focus was on the improved usability and the enhanced functionality.

As of June 2012, the SIMOTION D435 platform was enhanced with the innovation provided with SINAMICS CU320-2 and the new processor for SIMOTION. Thus, from the platform viewpoint, the software and hardware of the SIMOCRANE Basic Technology V3.0 package with SIMOTION D435-2 are no longer compatible with V2.0 SP2 that can run on the SIMOTION D435.

With the new version V3.0, the usability has been continuously improved, for example, with the introduction of the Web-based SIMOCRANE WebStart tool for commissioning and diagnostics as well as with the introduction of the Auto-Setting function for commissioning the load-dependent field weakening and the continuous load measurement function. The software and hardware interfaces are extended in V3.0, for example with the interface to the coming "ECO technology" innovation and the integrated interface to SINAMICS DC Master. In addition, a virtual interface has been implemented to create a PLC user program directly in SIMOTION without requiring a SIMATIC S7.

Scope of delivery

The previous SIMOCRANE Basic Technology V2.0 SP2 package with SIMOTION D435 (order number: 6AU1660-4AA10-0AA0) will continue to be marketed. In parallel, the SIMOCRANE Basic Technology V3.0 package with SIMOTION D435-2 (order number: 6AU1660-4AA20-0AA0) is supplied. This means, as customer you can choose between these two packages depending on the project and the system.

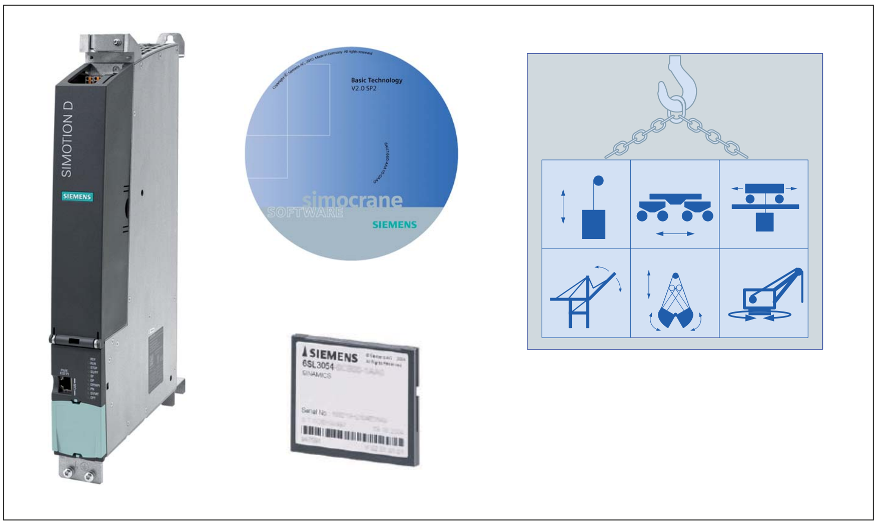


Figure 1-3 SIMOCRANE Basic Technology scope of supply

The SIMOCRANE Basic Technology package offers a control system for various crane applications. It includes:

SIMOCRANE Basic Technology V2.0 SP2

Hardware

- SIMOTION D435

Memory card (CF card)

- Latest firmware version for SIMOTION D4x5 and SINAMICS S120
- Licenses:
  - SIMOTION MultiAxes (for motion control)
  - SIMOTION IT (for diagnostics via Web server)
  - SIMOTION Crane Basic Technology (for functions in the crane DCC library)

Software

- CD with
  - Setup for installing the Crane DCC Library and online help
  - Standard application and the Crane FB Library
  - Documentation

SIMOCRANE Basic Technology V3.0

Hardware

- SIMOTION D435-2 DP/PN

Memory card (CF card)

- Latest firmware version for SIMOTION D4x5-2 and SINAMICS S120
- Licenses:
  - SIMOTION MultiAxes (for motion control)
  - SIMOTION Crane Basic Technology (for functions in the Crane DCC library)

Software

- CD with
  - Setup for installing the Crane DCC Library and online help
  - Standard application and the Crane FB library
  - SIMOTION D4x5-2 V4.3.1 and SINAMICS V4.5 firmware
  - Documentation

Product structure

The software package includes a Crane DCC library, a Crane FB library and several complete standard applications (e.g. for STS cranes).

The Crane DCC Library comprises a collection of blocks (e.g. load dependent field weakening) which are implemented as "Drive Control Charts" (DCC) blocks. DCC is a representation which supports graphic configuring and interconnecting. For detailed information about the scope of functions of the Cranes Library, refer to table 1-2.

The Crane DCC Library consists of a collection of blocks (e.g. OperationMode) which are implemented in "Structured Text" (ST) as function blocks. These blocks are called in the drive-oriented sequential control.

The standard applications comprise ready-to-use configured application software for different crane types, e.g. "Ship to shore crane" etc. These solutions are "Ready-to-Run" for users who only want to be concerned with parameterization. These can be considered being a "Ready-to-Apply" basis for large-scale adaptations and expansions, offering a high degree of expansibility and flexibility.

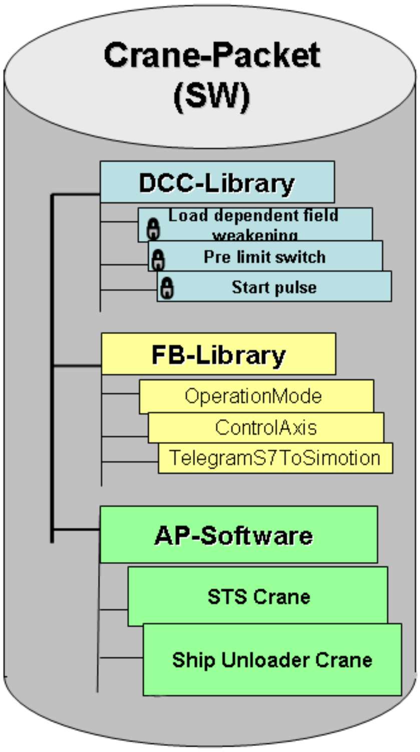


Figure 1-4 Product structure for Basic Technology applications

Functionality

The application software has a modular structure according to crane types. An overview of the function modules, their operating modes and technology function used are shown in the following tables.

Table 1- 1 Overview, function modules, axes and operating modes

Function module	Number of axes	Control mode	Modes of operation
Hoist	4	• Speed-controlled • Positioning • Master-slave operation • Synchronous operation	• AUTOMATIC • MANUAL • SPEED_CONTROLLED (jogging) • SWAYCONTROL • EASY_POSITIONING
Trolley (Cross Travel)	2	• Speed-controlled • Positioning • Master-slave operation • Synchronous operation	• AUTOMATIC • MANUAL • SPEED_CONTROLLED (jogging) • SENSORLESS EMERGENCY • SWAYCONTROL • EASY_POSITIONING
Gantry (Long Travel)	2	• Speed-controlled • Positioning • Master-slave operation • Synchronous operation	• AUTOMATIC • MANUAL • SPEED_CONTROLLED (jogging) • SENSORLESS EMERGENCY • EASY_POSITIONING
Boom (Luffing Gear)	1	• Speed-controlled	• SPEED_CONTROLLED (jogging)
Holding and Closing Gear	4	• Each separately speed-controlled with one axis (Holding Gear or Closing Gear) • Synchronous operation between both axes	• SPEED_CONTROLLED (jogging) • SWAYCONTROL • EASY_POSITIONING
Slewing Gear	1	• Speed-controlled • Positioning	• AUTOMATIC • MANUAL • SPEED_CONTROLLED (jogging) • SENSORLESS EMERGENCY • EASY_POSITIONING

Table 1- 2 Overview, crane-specific technology functions

No.	Function/block	Brief description
1	Load-dependent field weakening/ DCC_LoadDependField Weak_1/ DCC_LoadDependingField Weak	Using this DCC block, a supplementary speed setpoint is calculated dependent on the load. This speed increase for partial loads above the rated speed is required for cranes to increase the handling capacity.
2	Prelimit switch (selectable limiting)/ DCC_PreLimitSwitch	The velocity of the drive can be influenced using this DCC block when a predefined pre-limit switch is reached.
3	Start pulse / DCC_StartPulse_1/ DCC_StartPulse	Using this DCC block, "load sag" when starting hoisting gear with a suspended load is either prevented or reduced.
4	Changeover of the ramp-function generator in the field-weakening range and when selecting heavy duty operation DCC_ChangeOverHDFW	Using this DCC block, the acceleration and deceleration times are modified in heavy duty operation or in field weakening.
5	Current distribution monitoring/ DCC_CurrentDistribution	This DCC block is used to compare the current setpoint or the actual values of the master and slave. A message is generated if a specified difference is exceeded.
6	Slack rope controller/ DCC_SlackRopeControl_1/ DCC_SlackRopeControl	This function helps to avoid the formation of slack ropes when the grab starts handling the cargo. This function also ensures that the grab can bury itself into the material – therefore ensuring the maximum filling level.
7	Current equalization control with orange-peel bucket mode / DCC_CurrentEqualControl_1/ DCC_CurrentEqualControl	When raising and lowering the closed grab, the tension levels in the holding and closing ropes should be approximately the same. This means that the hoisting power is optimally distributed across the two motors.
8	Slewing velocity dependent on working radius/ DCC_VelocityChangeSlewGear	The speed of the slewing gear is adapted depending on the boom working radius in order to keep the circumferential velocity constant.
9	Ramp-up and ramp-down time dependent on working radius + velocity/ DCC_AccelChangeSlew Gear	For cranes with luffing gear, with increasing working radius, the load torque for the slewing gear increases while accelerating. In order to avoid the current limits being reached, the ramp-up and ramp-down times are suitably adapted as a function of the working radius and/or the angular velocity.
10	Master switch / DCC_MasterSwitch_1/ DCC_MasterSwitch	Using this DCC block, the drive can be moved with a fine sensitivity for manual positioning using a master switch, which is directly connected.
11	Anti-slip control/ DCC_TractionControl	The velocity between the motor encoder and the external encoder is monitored using this DCC block. If an excessively high velocity deviation occurs, the velocity of the acceleration is adapted.
12	Heavy duty or constant field weakening/ DCC_HeavyDuty	With this DCC block, the drive becomes capable of heavy duty operation (HeavyDuty) or operation with constant field weakening (FieldWeak) through variation in the velocity.
13	Overspeed monitoring/ DCC_OverSpeed	For hoisting gear applications, using this DCC block, an overspeed condition is monitored or a setpoint-actual value deviation is detected (this is not a fail-safe function).

No.	Function/block	Brief description
14	Setpoint monitoring/ DCC_Monitoring	This DCC block is used to monitor whether the velocity, acceleration or deceleration have been reduced. Further, it is monitored whether the drive is in field weakening.
15	Continuous load measurement/ DCC_ContLoadMeasurement_1/ DCC_ContLoadMeasurement	This DCC block measures the load continuously to guide the crane driver when he/she cannot see the grab. The message "Grab touchdown" is also displayed.
16	Grab monitoring/ DCC_GrabMonitor	This DCC block can be used to detect bulky load material in the closing gear.
17	Time-optimized positioning for a single axis	Using this system function, the drive can be moved to the target position as quickly as possible with the specified maximum velocity and acceleration/deceleration.
18	Master-slave closed-loop torque control	Master-slave operation is used if two motors are connected to a common shaft. The master operates either closed-loop position controlled or closed-loop speed controlled depending on the operating mode. The slave only operates closed-loop torque controlled. The master sends the torque as torque setpoint to the slave.
19	Synchronous operation	Synchronous operation control is used if two motors are connected to a common load. Depending on the operating mode, the master and slave operate either closed-loop position controlled or closed-loop speed controlled. The slave receives a speed or position setpoint depending on the operating mode from the master via a gear (gear ratio 1 : 1). The functional scope has been expanded with the implementation of flying homing, offset compensatory control and offset mode.
20	Tandem mode	The tandem mode is an extension of the synchronous operation control mode. Synchronous operation motion control takes place between two groups. In each group, two drives can be coupled in master-slave closed-loop torque control or also in synchronous operation. The function is suitable for applications with harbor cranes (for example double-spreader container cranes or large ship unloaders with four-drum grabs) as well as for applications involving industrial cranes with several hoisting gear and trolleys.
21	Cornering movement	Using this function, cornering movement for the crane long travel (gantry) can be executed in closed-loop speed controlled operation.
22	Brake test	The mechanical brake function (e.g. hoisting gear) should be regularly checked using this function. To do this, the axis moves against the closed brake with a certain torque setpoint in order to check the braking capability of the brake.
23	Basic positioner	This is a positioning that does not use a position controller of the axis but rather FB_OperationMode is calculated in the FB library; it is suitable for systems that tend to be subject to mechanical vibrations, such as a rope-drawn trolley on STS cranes.

For example, the standard application for the STS crane includes the four function modules: Hoist, Trolley, Gantry and Boom. For the actual scope of these function modules, their functions, operating modes and technologies, refer to Standard applications (Page 497).

Hardware interfaces 2

2.1 Handling a CompactFlash card of a D4x5-2

The CompactFlash card is supplied bootable with the latest SIMOTION Kernel, the latest drive software and the licenses required for the crane applications. The SIMOTION D4xx cannot be operated without a CompactFlash card! The licensing is associated with the serial number of the CompactFlash card.

NOTICE
The SIMOTION CompactFlash card must not be formatted. The crane application software cannot run without licenses.

As of SIMOCRANE Basic Technology V3.0, the firmware versions for SIMOTION and SINAMICS with which the associated version of the Basic Technology was released are contained on the product CD-ROM. This means, should the firmware version of SIMOTION or SINAMICS not be compatible, you can use the version stored on the CD.

2.2 Interfaces of the SIMOTION D435-2 DP/PN

Available interfaces

Table 2- 1 Overview of the available interfaces

Interface	Designation	Connector type
DRIVE-CLiQ interface	X100	DRIVE-CLiQ socket
DRIVE-CLiQ interface	X101	DRIVE-CLiQ socket
DRIVE-CLiQ interface	X102	DRIVE-CLiQ socket
DRIVE-CLiQ interface	X103	DRIVE-CLiQ socket
DRIVE-CLiQ interface (not for SIMOTION D425-2)	X104	DRIVE-CLiQ socket
DRIVE-CLiQ interface (not for SIMOTION D425-2)	X105	DRIVE-CLiQ socket
Ethernet interface PN/IE	X127 P1	RJ45 socket
Ethernet interface PN/IE-NET	X130 P1	RJ45 socket
PROFINET PN IO interface (only for SIMOTION D4x5-2 DP/PN)	X150 (P1, P2, P3) **Note:** X150 (P3) corresponds to X120 PN/IE-OP.	RJ45 socket

Interface	Designation	Connector type
Digital I/Os	X122, X132, X142	Mini Combicon, 3.5 mm 3x14-pin
Power supply connector	X124	Combicon, 4-pin
PROFIBUS DP interface	X126	9-pin Sub-D socket
PROFIBUS DP/MPI interface	X136	9-pin Sub-D socket
Measuring sockets (T0, T1, T2 and M)	X141	4-pin, socket
SIMOTION CF slot	X109	CF card connector
Fan/battery module interface	X190/X191	Fan/battery module
1. USB port	X125	USB socket
2. USB port	X135	USB socket
Option slot		Sockets

Non-usable interfaces

Table 2- 2 Overview of interfaces that cannot be used for SIMOTION D

Interface designation	Interface	Connector type
RS 232 interface	X140	9-pin Sub-D, pins

2.2.1 Ethernet ports

Interfaces for connection to Industrial Ethernet

Industrial Ethernet is a communication network with a transmission rate of 10/100/1000 Mbit/s.
SIMOTION D4x5-2 offers the following functions via Ethernet interfaces:

- Communication with STEP 7 and SIMOTION SCOUT
- Communication between SIMOTION and SIMATIC NET OPC

  The following software must be installed on the PG/PC for this function:
  "SIMATIC NET SOFTNET-S7 (S7-OPC server)"
- Connection of HMI systems
- Communication with other devices using TCP/IP or UDP communication
- IT communication (via SIMOTION IT DIAG, SIMOTION IT OPC XML-DA, SIMOTION IT Virtual Machine).

Position of the connectors for SIMOTION D4x5-2 DP/PN

The following figure shows the position of the Ethernet interfaces on the D4x5-2 DP/PN and their displays.

Display	Status		Meaning
LINK	orange	🟧	Transfer rate 1000 Mbit/s
	green	🟩	Transfer rate 10 or 100 Mbit/s
	Off	□	No or faulty connection
ACT	yellow	🟨	Data exchange
	Off	□	No data exchange

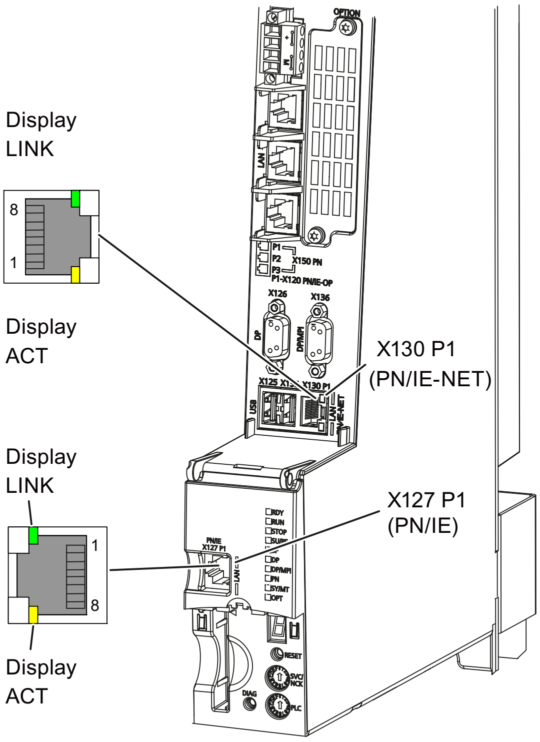


Figure 2-1 The position of the Ethernet interfaces and their displays (example of SIMOTION D445-2 DP/PN)

Note

Because, as of V4.3, the two Ethernet interfaces support PROFINET basic services, enter the designation PN/IE-NET or PN/IE.

These PROFINET basic services (e.g. DCP, LLDP, SNMP) provide uniform functions for the address assignment and diagnostics, but do not provide PROFINET IO communication for the connection of drives, I/O modules, etc.

Additional references

Detailed information about the states of the status LEDs can be found in the *SIMOTION D4x5-2* Commissioning and Hardware Installation Manual, *Diagnostics* chapter.

Interface characteristics

Table 2- 3 X127 P1 and X130 P1

Feature	Version
Connector type	RJ45 socket
Cable type	Industrial Ethernet cable • 4- and 8-wire cables can be used for 10/100 Mbit/s • 8-wire cables must be used for 1000 Mbit/s
Max. cable length	100 m
Autocrossing	Yes
Dust protection blanking plugs for sealing unused Ethernet ports	Five blanking plugs contained in the D4x5-2 scope of delivery Blanking plugs (50 pcs) order number: 6SL3 066-4CA00-0AA0
Miscellaneous	X127 and X130 P1 are full-duplex 10/100/1000 Mbit/s Ethernet ports

Note

The MAC addresses are imprinted on an adhesive label that is located behind the protective cover and can be seen from the front.

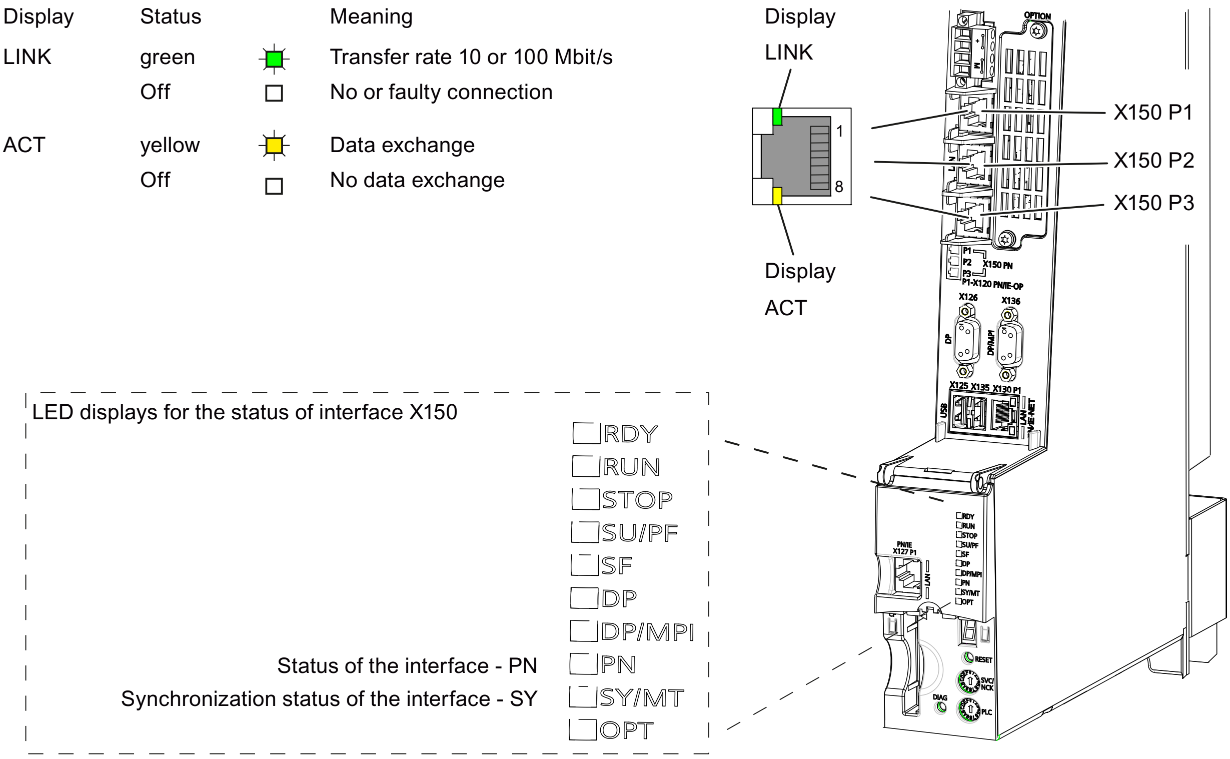
2.2.2 PROFINET IO interface

PROFINET is an open component-based industrial communication system using Ethernet for distributed automation systems.

SIMOTION D4x5-2 DP/PN has a PROFINET interface with three ports (X150 P1-P3) onboard. The PROFINET interface supports operation of a SIMOTION D4x5-2 DP/PN as an IO controller and/or as an I device.

Interface position

The following figure contains information about the PROFINET interface of the control unit. The position of the interface, the labeling of the ports and the associated displays are described.



Display	Status		Meaning
LINK	green	🟩	Transfer rate 10 or 100 Mbit/s
	Off	□	No or faulty connection
ACT	yellow	🟨	Data exchange
	Off	□	No data exchange

Figure 2-2 The position of the PROFINET X150 P1 to P3 interfaces and their displays (SIMOTION D445-2 DP/PN)

Note

The third port of the PROFINET IO interface X150 P3 is also designated as X120 PN/IE OP. This designation is not relevant for SIMOTION D.

Additional references

Detailed information about the states of the status LEDs can be found in the *SIMOTION D4x5-2* Commissioning and Hardware Installation Manual, *Diagnostics* chapter.

Interface characteristics

Table 2- 4 Ports X150 P1 to P3

Feature	Version
Connector type	RJ45plus socket
Cable type	PROFINET
Maximum cable length	100 m
Minimum transmission cycle	0.25 ms
Autocrossing	Yes i.e. crossed and uncrossed cables can be used
Dust protection blanking plugs for sealing unused PROFINET ports	Five blanking plugs contained in the D4x5-2 scope of delivery Blanking plugs (50 pcs) order number: 6SL3066-4CA00-0AA0

Connectable devices

The following devices can be connected to the PROFINET IO interface:

- PG/PC programming devices (communication with SIMOTION SCOUT / STEP 7)
- SIMATIC HMI devices
- SIMATIC controllers with PROFINET interface
- Distributed I/O
- Drive units with PROFINET IO interface (standard devices)

The SIMOTION D4x5-2 DP/PN then assumes the role of a PROFINET IO controller and can offer the following functions:

- PROFINET IO controller, I device (also controller and device simultaneously)
- Supports real-time classes of PROFINET IO:
  - RT (real-time)
  - IRT (isochronous real-time).

The following functions are also supported by Industrial Ethernet:

- Communication between SIMOTION and SIMATIC NET OPC

  The "SIMATIC NET SOFTNET-S7 (S7-OPC server)" software must be installed on the PG/PC for this function.
- Communication with other devices using TCP/IP or UDP communication
- IT communication (via SIMOTION IT DIAG, SIMOTION IT OPC XML-DA, SIMOTION IT Virtual Machine)

For more information regarding the software packages, see Catalog PM 21, refer to the list of references (separate document) for the order number.

Note

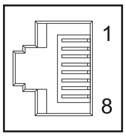
A list of the modules released with SIMOTION is available at (http://support.automation.siemens.com/WW/view/en/11886029).

The list is updated regularly and contains information on the use of these modules.

Take note of the documentation on the individual modules or devices!

Interface assignment

Table 2- 5 Assignment of the ports X150 P1 to P3

Representation	Pin	Name	Description
	1	TXP	Transmit data +
	2	TXN	Transmit data -
	3	RXP	Receive data +
	4	-	Reserved, do not use
	5	-	Reserved, do not use
	6	RXN	Receive data -
	7	-	Reserved, do not use
	8	-	Reserved, do not use

Second PROFINET interface

The Ethernet Communication Board (CBE30-2) provides as option a second PROFINET interface for the D4x5-2 DP/PN Control Units.

The CBE30-2 cannot be used in SIMOTION D4x5 2 DP Control Units.

2.2.3 Digital I/Os

Interface characteristics

The digital I/Os on the X122, X132 and X142 connectors are provided for the connection of sensors and actuators.

Table 2- 6 Wiring of X122, X132 and X142

Characteristics		Version
Connector type	Mini Combicon	
Connectable cable types and conductor cross-sections	Rigid Flexible Flexible with end sleeve AWG	0.2 mm^2 to 1.5 mm^2 0.2 mm^2 to 1.5 mm^2 0.25 mm^2 to 1.5 mm^2 24 to 16
Stripped length	10 mm	
Tool	Screwdriver 0.4 x 2.0 mm	
Max. cable length	30 m	
Max. current carrying capacity (ground)	8 A	

Position of the connectors

The following figure shows the position of the interface connectors on the D4x5-2 and the assignment of the various digital I/Os.

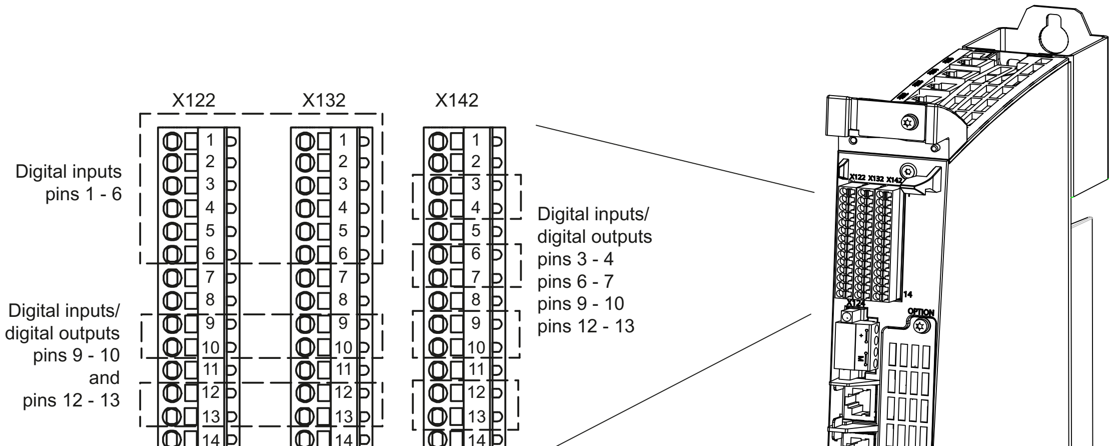

X122
X132
X142
Digital inputs pins 1 - 6
Digital inputs/ digital outputs pins 9 - 10 and pins 12 - 13
Digital inputs/ digital outputs pins 3 - 4 pins 6 - 7 pins 9 - 10 pins 12 - 13
1
2
3
4
5
6
7
8
9
10
11
12
13
14
X122 X132 X142
14
OPTION

Figure 2-3 Position of the X122, X132 and X142 digital interfaces (example of SIMOTION D445-2 DP/PN)

2.3 Interfaces of the Communication Board Ethernet 30 (CBE30-2)

Properties of the CBE30-2

A second PROFINET interface can be implemented for the SIMOTION D4x5-2 DP/PN with the CBE30-2 Ethernet communication board.

The CBE30-2 cannot be used with the SIMOTION D4x5-2 DP.

The CBE30-2 offers the following functions:

- PROFINET IO controller, I device (also controller and device simultaneously)
- 100 Mbit/s full duplex / autocrossing
- Supports real-time classes of PROFINET IO:
  - RT (real-time)
  - IRT (isochronous real-time).

The CBE30-2 has an X1400 interface with integrated 4-port switch based on PROFINET ASIC ERTEC 400.

View

The connections and LED displays are provided on the front of the CBE30-2.

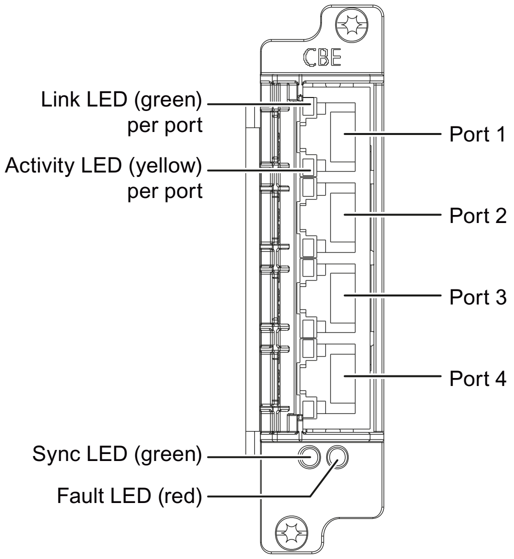


Figure 2-4 CBE30-2, X1400 interface front view

Software structure 3

3.1 Software structure

The complete software has a modular structure. Each function module (e.g. Hoist, Gantry, Trolley etc.) has its own Software Unit. Using two application programs generated in the graphic programming language "Motion Control Chart", the required function blocks are called from the library "Cranes FB Library" in every unit to run the appropriate function module. Further, every function module has a setpoint channel generated in the graphic programming language "Drive Control Charts" (DCC) for velocity, acceleration and deceleration. In addition, the DCC monitors various functions (e.g. current distribution monitoring, overspeed).

Furthermore, a separate unit is responsible for cross-module monitoring and homing. If needed, the special cornering functions for gantry, synchronization of the time on the SIMOTION/SINAMICS by SIMATIC PLC and AutoSetting for the commissioning of the load-dependent field weakening special functions can be activated.

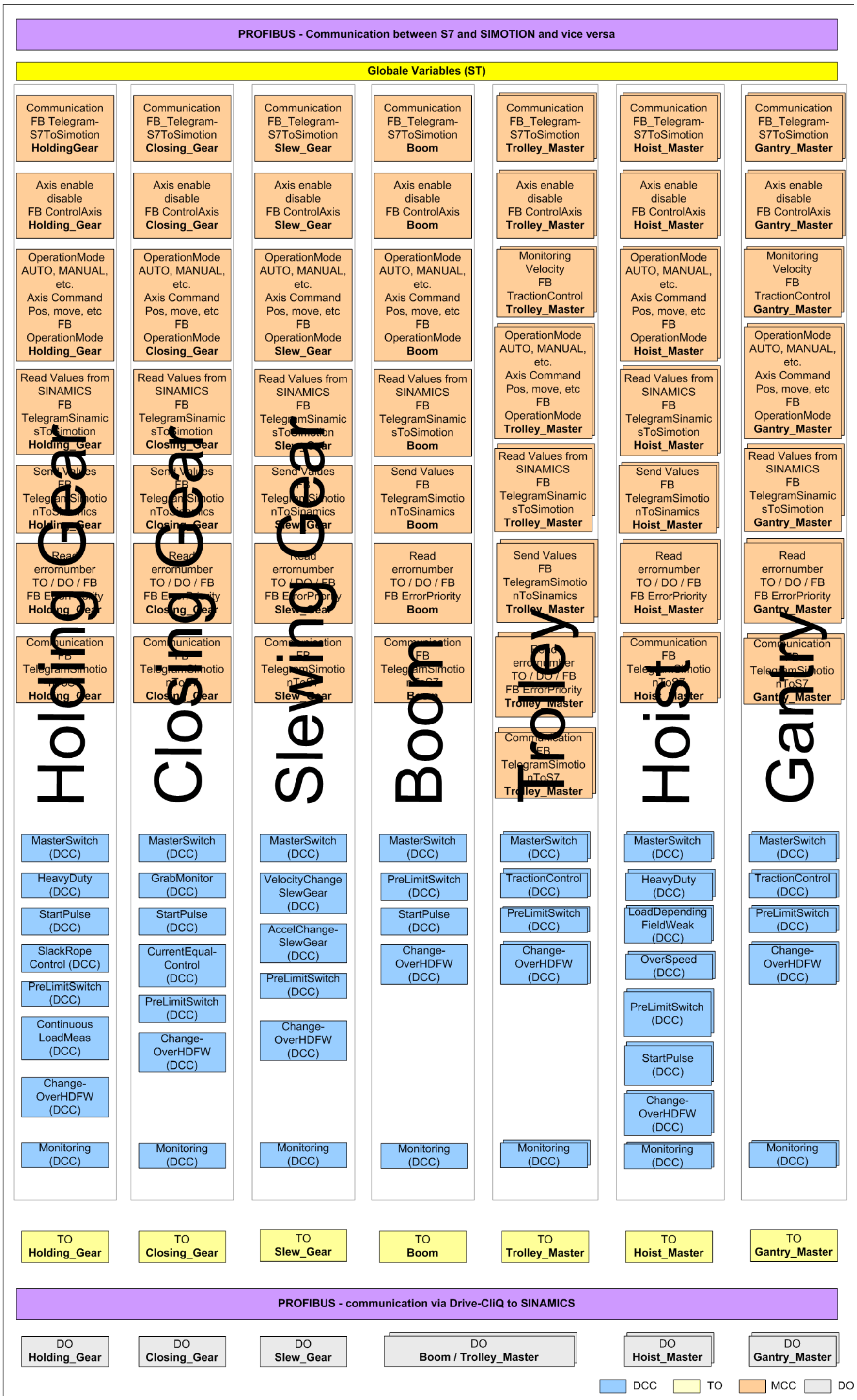

PROFIBUS - Communication between S7 and SIMOTION and vice versa
Globale Variables (ST)
Holding Gear
Closing Gear
Slewing Gear
Boom
Trolley
Hoist
Gantry
Communication FB Telegram-S7ToSimotion HoldingGear
Communication FB_Telegram-S7ToSimotion Closing_Gear
Communication FB_Telegram-S7ToSimotion Slew_Gear
Communication FB_Telegram-S7ToSimotion Boom
Communication FB_Telegram-S7ToSimotion Trolley_Master
Communication FB_Telegram-S7ToSimotion Hoist_Master
Communication FB_Telegram-S7ToSimotion Gantry_Master
Axis enable disable FB ControlAxis Holding_Gear
Axis enable disable FB ControlAxis Closing_Gear
Axis enable disable FB ControlAxis Slew_Gear
Axis enable disable FB ControlAxis Boom
Axis enable disable FB ControlAxis Trolley_Master
Axis enable disable FB ControlAxis Hoist_Master
Axis enable disable FB ControlAxis Gantry_Master
OperationMode AUTO, MANUAL, etc. Axis Command Pos, move, etc FB OperationMode Holding_Gear
OperationMode AUTO, MANUAL, etc. Axis Command Pos, move, etc FB OperationMode Closing_Gear
OperationMode AUTO, MANUAL, etc. Axis Command Pos, move, etc FB OperationMode Slew_Gear
OperationMode AUTO, MANUAL, etc. Axis Command Pos, move, etc FB OperationMode Boom
Monitoring Velocity FB TractionControl Trolley_Master
OperationMode AUTO, MANUAL, etc. Axis Command Pos, move, etc FB OperationMode Hoist_Master
Monitoring Velocity FB TractionControl Gantry_Master
OperationMode AUTO, MANUAL, etc. Axis Command Pos, move, etc FB OperationMode Trolley_Master
OperationMode AUTO, MANUAL, etc. Axis Command Pos, move, etc FB OperationMode Gantry_Master
Read Values from SINAMICS FB TelegramSinamicsToSimotion Holding_Gear
Read Values from SINAMICS FB TelegramSinamicsToSimotion Closing_Gear
Read Values from SINAMICS FB TelegramSinamicsToSimotion Slew_Gear
Read Values from SINAMICS FB TelegramSinamicsToSimotion Boom
Read Values from SINAMICS FB TelegramSinamicsToSimotion Hoist_Master
Read Values from SINAMICS FB TelegramSinamicsToSimotion Trolley_Master
Read Values from SINAMICS FB TelegramSinamicsToSimotion Gantry_Master
Send Values FB TelegramSimotionToSinamics Holding_Gear
Send Values FB TelegramSimotionToSinamics Closing_Gear
Send Values FB TelegramSimotionToSinamics Slew_Gear
Send Values FB TelegramSimotionToSinamics Boom
Send Values FB TelegramSimotionToSinamics Hoist_Master
Send Values FB TelegramSimotionToSinamics Trolley_Master
Read errornumber TO / DO / FB FB ErrorPriority Holding_Gear
Read errornumber TO / DO / FB FB ErrorPriority Closing_Gear
Read errornumber TO / DO / FB FB ErrorPriority Slew_Gear
Read errornumber TO / DO / FB FB ErrorPriority Boom
Read errornumber TO / DO / FB FB ErrorPriority Hoist_Master
Read errornumber TO / DO / FB FB ErrorPriority Gantry_Master
Read errornumber TO / DO / FB FB ErrorPriority Trolley_Master
Communication FB TelegramSimotionToS7 Holding_Gear
Communication FB TelegramSimotionToS7 Closing_Gear
Communication FB TelegramSimotionToS7 Slew_Gear
Communication FB TelegramSimotionToS7 Boom
Communication FB TelegramSimotionToS7 Hoist_Master
Communication FB TelegramSimotionToS7 Gantry_Master
Communication FB TelegramSimotionToS7 Trolley_Master
MasterSwitch (DCC)
HeavyDuty (DCC)
StartPulse (DCC)
SlackRope Control (DCC)
PreLimitSwitch (DCC)
Continuous LoadMeas (DCC)
Change-OverHDFW (DCC)
Monitoring (DCC)
MasterSwitch (DCC)
GrabMonitor (DCC)
StartPulse (DCC)
CurrentEqual-Control (DCC)
PreLimitSwitch (DCC)
Change-OverHDFW (DCC)
Monitoring (DCC)
MasterSwitch (DCC)
VelocityChange SlewGear (DCC)
AccelChange-SlewGear (DCC)
PreLimitSwitch (DCC)
Change-OverHDFW (DCC)
Monitoring (DCC)
MasterSwitch (DCC)
PreLimitSwitch (DCC)
StartPulse (DCC)
Change-OverHDFW (DCC)
Monitoring (DCC)
MasterSwitch (DCC)
TractionControl (DCC)
PreLimitSwitch (DCC)
Change-OverHDFW (DCC)
Monitoring (DCC)
MasterSwitch (DCC)
HeavyDuty (DCC)
LoadDepending FieldWeak (DCC)
OverSpeed (DCC)
PreLimitSwitch (DCC)
StartPulse (DCC)
Change-OverHDFW (DCC)
Monitoring (DCC)
MasterSwitch (DCC)
TractionControl (DCC)
PreLimitSwitch (DCC)
Change-OverHDFW (DCC)
Monitoring (DCC)
TO Holding_Gear
TO Closing_Gear
TO Slew_Gear
TO Boom
TO Trolley_Master
TO Hoist_Master
TO Gantry_Master
PROFIBUS - communication via Drive-CliQ to SINAMICS
DO Holding_Gear
DO Closing_Gear
DO Slew_Gear
DO Boom / Trolley_Master
DO Hoist_Master
DO Gantry_Master
DCC
TO
MCC
DO

Figure 3-1 Software structure of a function module

3.2 Structure of a function module

The structure of a function module is split up in the unit into two MCC programs. One program operates in the normal cycle, while the other requires a high-speed cycle, e.g. communication with SINAMICS. A separate unit is responsible for cross-axis monitoring and homing. In these programs, the function blocks are called from the library "Crane FB library" as required by the sequencer.

All function modules essentially have an identical structure. Exceptions exist for travel gears on rails (e.g. trolley) and for master-slave applications for the slave. In these particular cases, additional functional blocks are inserted into the MCC program (e.g. for Gantp_1: Gantry_1_TractionControl). The structure of the unit for Gantry is described in the example. The function blocks are described in detail in Chapter Crane function blocks (Page 69).

In the unit "Gantp_1" the program "Gantrymcc_1_s" is executed according to the following sequence:

Normal execution in Gantrymcc_1_s

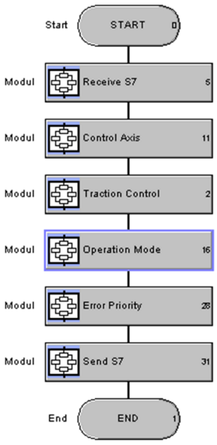


Figure 3-2 Unit "Gantp_1", program sequence "Gantrymcc_1_s"

1. The "FB_TelegramS7ToSimotion" function block is called.
   The function block reads in the data received from the SIMATIC S7. The setpoints are prepared in the DCC.
2. The function block "FB_ControlAxis" is called.
   Using this block, the control and status words between the technology object (TO) and the drive object (DO) are set and read.
3. The function block "FB_TractionControl" is called.
   The actual velocity of the motor encoder and external encoder is fetched using this DCC block. The setpoints are prepared in the DCC.
4. The function block "FB_OperationMode" is called.
   The operating mode management and the travel commands are executed using this block. The setpoints prepared in DCC are used for the motion commands.

5. The function block "FB_ErrorPriority" is called.
   Faults and alarms associated with technology objects, drive objects, and function blocks are fetched and prioritized using this block.
6. The function block "FB_TelegramSimotionToS7" is called.
   The send data from the SIMOTION is transferred to SIMATIC S7 using this block.

Fast execution in Gantrymcc_1_f

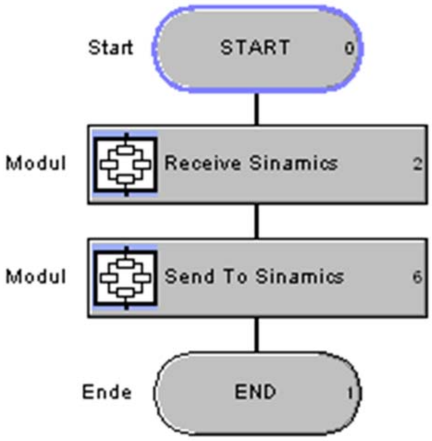


Figure 3-3 Unit "Gantp_1", program sequence "Gantrymcc_1_f"

1. The function block "FB_TelegramSinamicsToSimotion" is called.
   The process data from the drive object is evaluated using this block.
2. The function block "FB_TelegramSimotionToSinamics" is called.
   The process data from the technology object is evaluated using this block.

The gantry is monitored in the cross-axis "Mon_Ref" unit, and, when required, homed on the fly and an offset for a synchronous operation compensated. In addition, the cornering movement and the commissioning support by "FB_AutoSetting" are called in this unit.

Separate Unit Mon_Ref:

- The function block "FB_Cornering" is called in the IPOSynchronousTask. Cornering movement for the gantry is made possible using this block. For a detailed description, refer to Chapter FB_Cornering (Page 280)

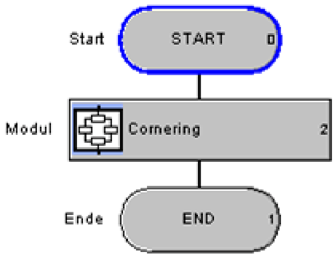


Figure 3-4 Unit "Mon_Ref", program sequence "Corneringmcc_1"

- The function block "FB_Monitoring" is called in an IPOSynchronousTask (e.g. IPO2). The master-slave connection or synchronous relationship is monitored using this block.

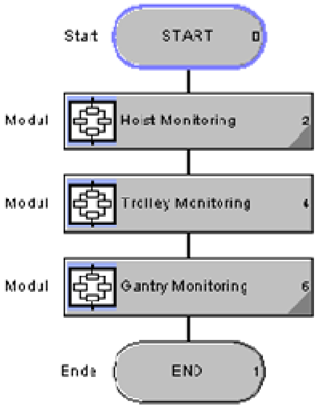


Figure 3-5 Unit "Mon_Ref", program sequence "Monitoringmcc_1"

- The function block "FB_ReferenceMode" is called in the IPOSynchronousTask. This block is used to execute flying homing to a parameterized position. The function block also offers the option to compensate the offset between the master and slave for synchronous operation.

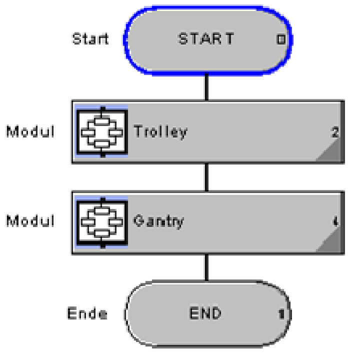


Figure 3-6 Unit "Mon_Ref", program sequence "Referencingmcc_1"

- The "FB_SynchronizeRTC" function block for setting the time of day is called as background task.

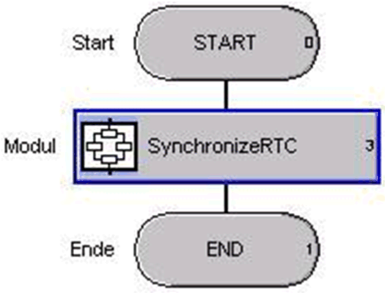


Figure 3-7 Unit "Mon_Ref", program sequence "TimeSynchronizeRTCmcc_1"

- The "FB_AutoSetting" function block for commissioning the load-dependent field weakening and continuous load measurement is called in the IPO synchronous task.

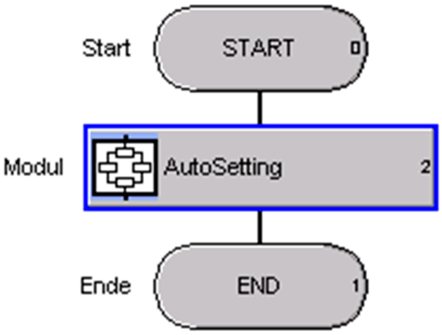


Figure 3-8 Unit "Mon_Ref", program sequence "Autosettingmcc_1"

3.3 Control Structure

The control (closed-loop) is structured so that the setpoints for velocity, acceleration, and deceleration are read into the program "Gantrymcc_1_s" from the SIMATIC S7 using the function block "FB_TelegramS7ToSimotion". These setpoints are interconnected at the output of the function block with the DCC function blocks from the crane DCC library. The setpoint channel of the three setpoints are described in more detailed in the following:

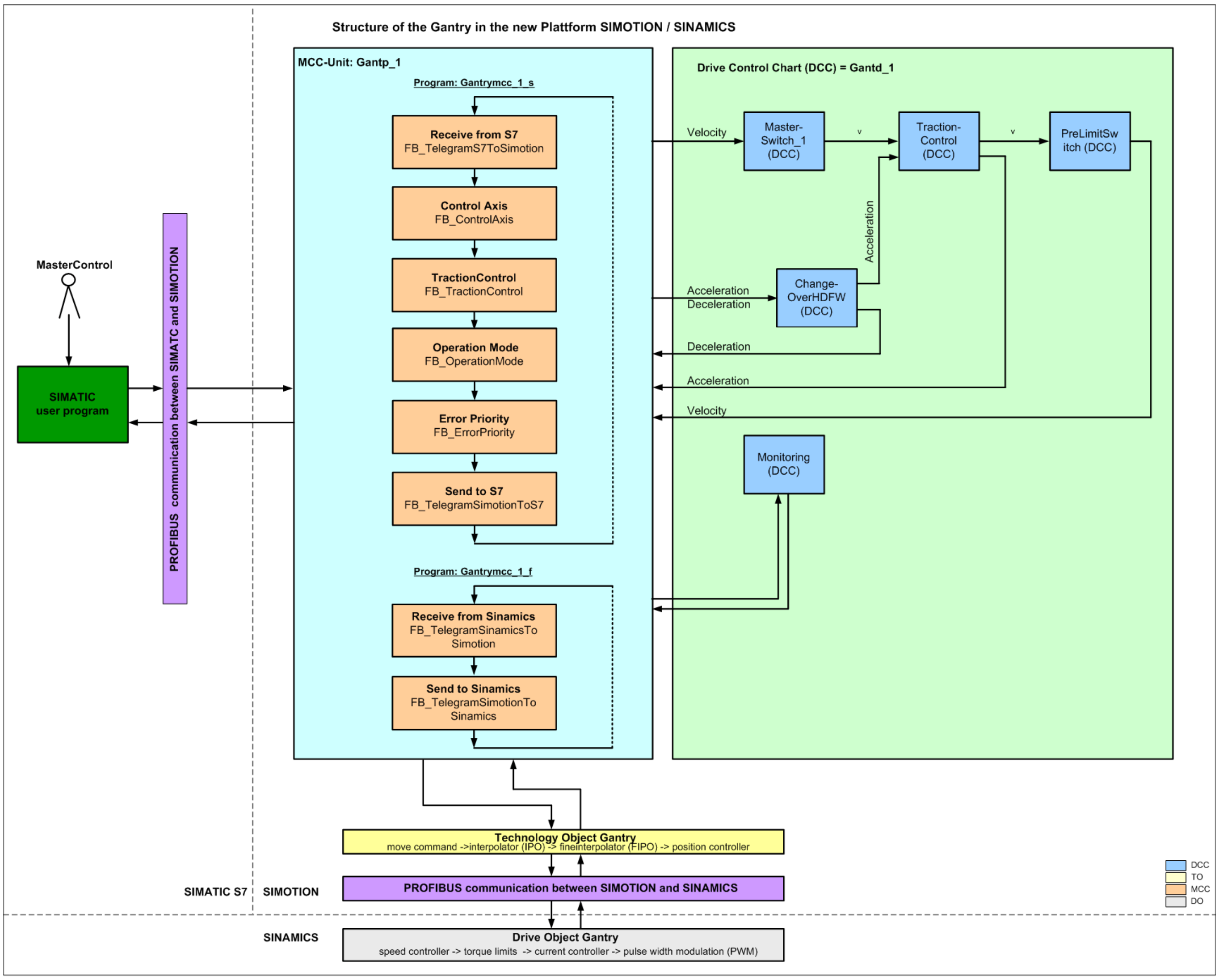


Figure 3-9 Setpoint channel of the three setpoints

Velocity path

The setpoint velocity is first interconnected to the "MasterSwitch_1" block in the DCC. If the master switch functionality is to be used, then the master switch curve must be configured and enabled. Otherwise, the velocity setpoint is directly connected from the input to the block output without any change.

The velocity setpoint is then connected to the block "TractionControl". This block detects any wheel slip of the travel gear running on rails. If wheel slip is identified, then the velocity can be reduced step-by-step if the appropriate enable signals are set in the block.

The velocity setpoint is connected to the block "PreLimitSwitch". Depending on the control, the setpoint velocity is reduced. Finally, the setpoint acceleration is interconnected to the function block "FB_OperationMode".

Acceleration path

The acceleration setpoint is firstly interconnected to the block "ChangeOverHDFW" in the DCC. In this block, the acceleration setpoint in heavy duty operation can be changed over from the value specified from the SIMATIC S7 to a permanently configured value, if the required enable signal is set.

The acceleration setpoint is then connected to the block "TractionControl". This block detects any wheel slip of the travel gear running on rails. If wheel slip is identified, then the acceleration can be reduced step-by-step if the appropriate enable signals are set in the block.

Finally, the setpoint velocity is interconnected to the function block "FB_OperationMode".

Deceleration path

The deceleration setpoint is firstly interconnected to the block "ChangeOverHDFW" in the DCC. In this block, the deceleration setpoint in heavy duty operation can be changed over from the value specified from the SIMATIC S7 to a permanently configured value if the required enable signal is set.

Finally, the setpoint deceleration is interconnected to the function block "FB_OperationMode".

Technology object (TO)

The travel commands – with velocity, acceleration and deceleration – are transferred to the technology object in the function block "FB_OperationMode". Depending on the active operating mode, the position, velocity, acceleration and deceleration profile for the travel motion are calculated in the technology object. The setpoint channel of the technology object is shown in the diagram below (see also Ref. [13]).

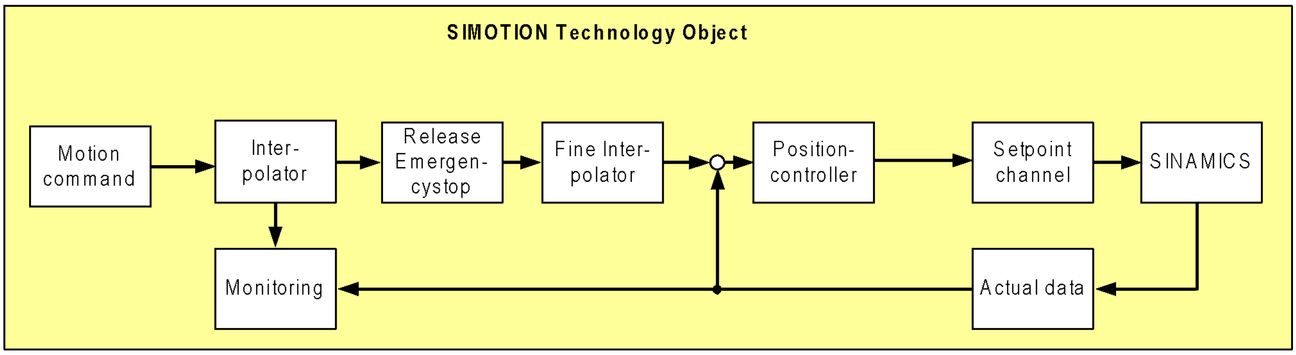


Figure 3-10 Setpoint channel of the technology object

Function modules 4

In terms of software, a function module contains an MCC unit, a DCC chart, a technology object (TO), and a drive object (DO); also refer to the section titled Structure of a function module (Page 31). A technology object models an axis at the SIMOTION level, also a drive object at the SINAMICS level. Both objects are coupled through the appropriate configuration. According to the sequence control, the function blocks are called from the library "Crane_FB_library" in the MCC unit. In addition to the MCC unit, setpoint preparation also requires the DCC blocks from the "Crane_DCC_Library".

4.1 Hoist

The function module Hoist comprises two axes – Hoist_1 (master) and Hoist_2 (slave). In the section titled Ship-to-shore tandem crane (STS tandem) (Page 470), the Hoist function module comprises four axes, where Hoist_3 is analogous to Hoist_1 and Hoist_4 is analogous to Hoist_2. Every axis has an MCC unit and a DCC chart. The units of the MCC program and the DCC blocks of the relevant function module are subdivided into the modules Hoistp_1 (master) / Hoistd_1 and Hoistp_2 (slave) / Hoistd_2. The MCC programs required are grouped in the Hoistp_1 and Hoistp_2 units. The DCC charts are grouped in the Hoistd_1 and Hoistd_2 units.

Using a data set switchover for the particular technology object (TO-Hoist_1 and TO-Hoist_2) - when required - it is possible to differentiate between the position actual value from the motor encoder (data set 1) and a position actual value from an external encoder (data set 2).

A data set switchover for the associated drive object (DO-Hoist_1 or DO-Hoist_2) allows different motor or encoder data sets to be selected (data set 0/1).

CAUTION
"Speed control without motor encoder (data set 1)" control mode for hoist is not permissible for safety reasons.

Hoistp_1 (FB blocks) comprises the following:

- FB_TelegramS7ToSimotion
- FB_ControlAxis
- FB_OperationMode
- FB_TelegramSinamicsToSimotion
- FB_TelegramSimotionToSinamics
- FB_ErrorPriority
- FB_TelegramSimotionToS7

Hoistp_2 (FB blocks) comprises the following:

- FB_TelegramS7ToSimotion
- FB_ControlAxis
- FB_OperationMode
- FB_TelegramSinamicsToSimotion
- FB_TelegramSimotionToSinamics
- FB_ErrorPriority
- FB_TelegramSimotionToS7

Hoistd_1 (DCC blocks) comprises the following:

- MasterSwitch_1
- HeavyDuty
- LoadDependFieldWeak_1
- OverSpeed
- PreLimitSwitch
- StartPulse_1
- ChangeoverHDFW
- Monitoring

Hoistd_2 (DCC blocks) comprises the following:

- MasterSwitch_1
- HeavyDuty
- LoadDependFieldWeak_1
- OverSpeed
- PreLimitSwitch
- StartPulse_1
- ChangeoverHDFW
- CurrentDistribution
- Monitoring

Mon_Ref comprises the following:

- FB_Monitoring
- FB_ReferenceMode
- FB_Autosetting

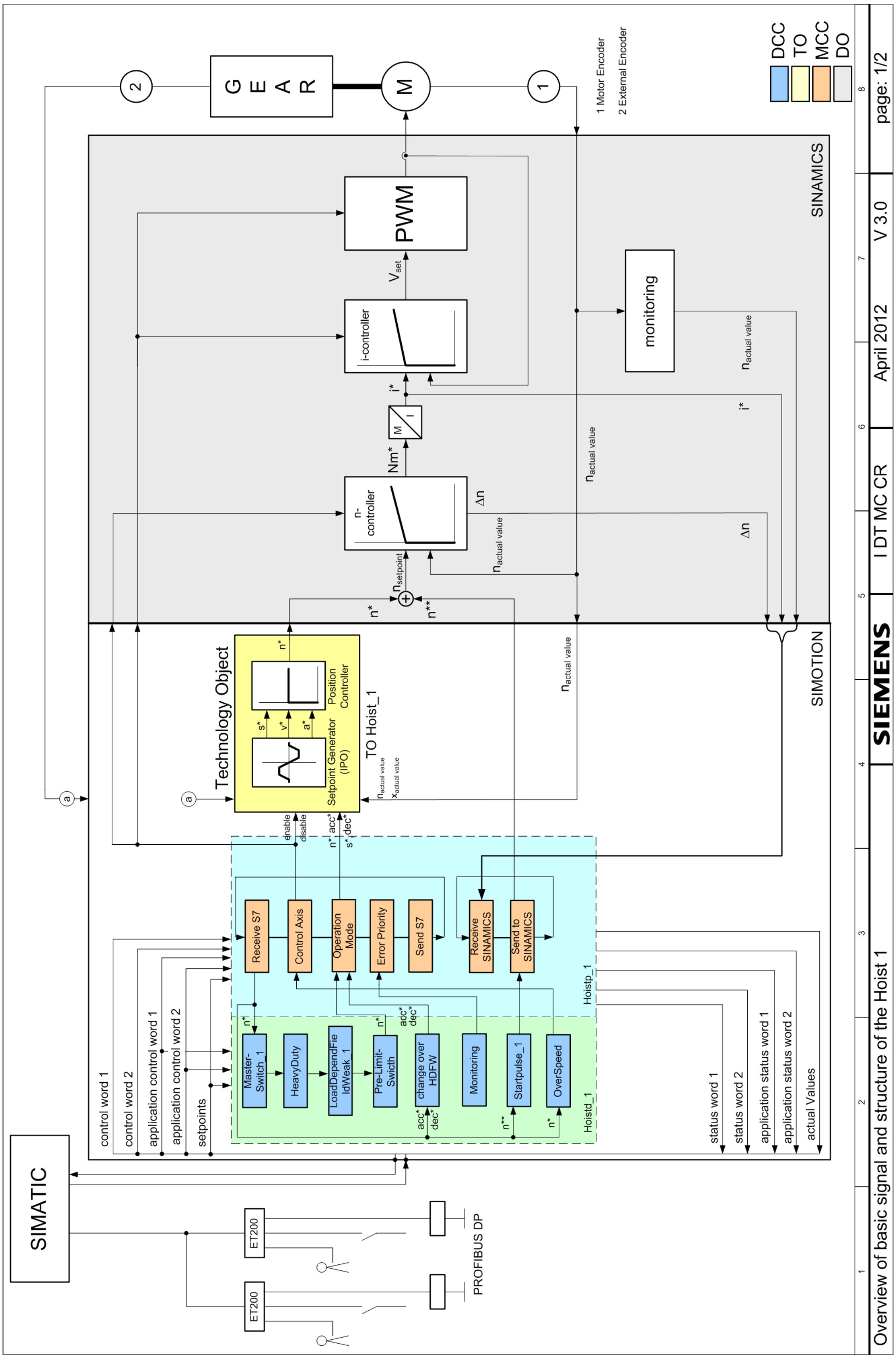


Figure 4-1 Hoist 1, basic signals 1

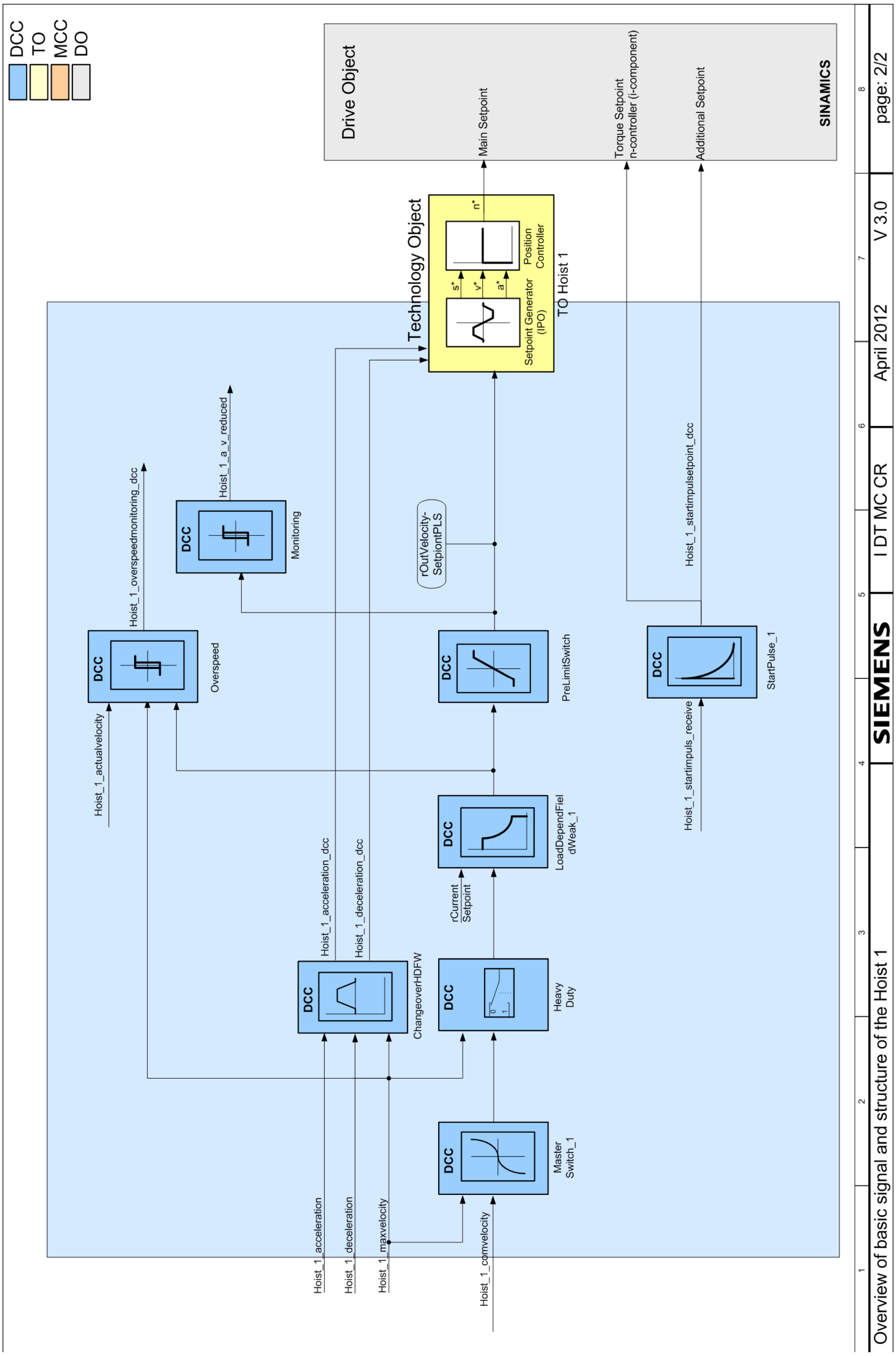


Figure 4-2 Hoist 1, basic signals 2

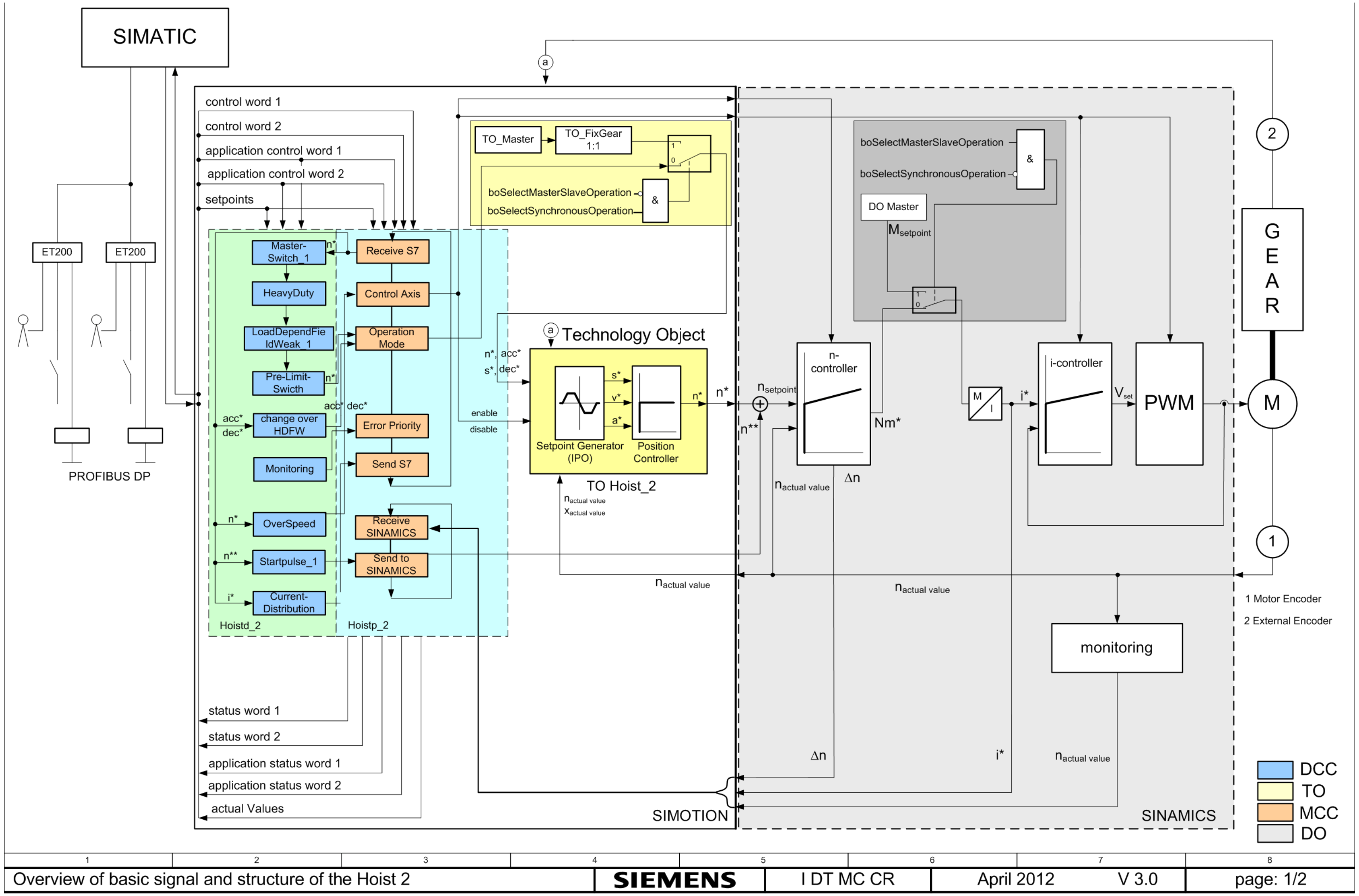


Figure 4-3 Hoist 2, basic signals 1

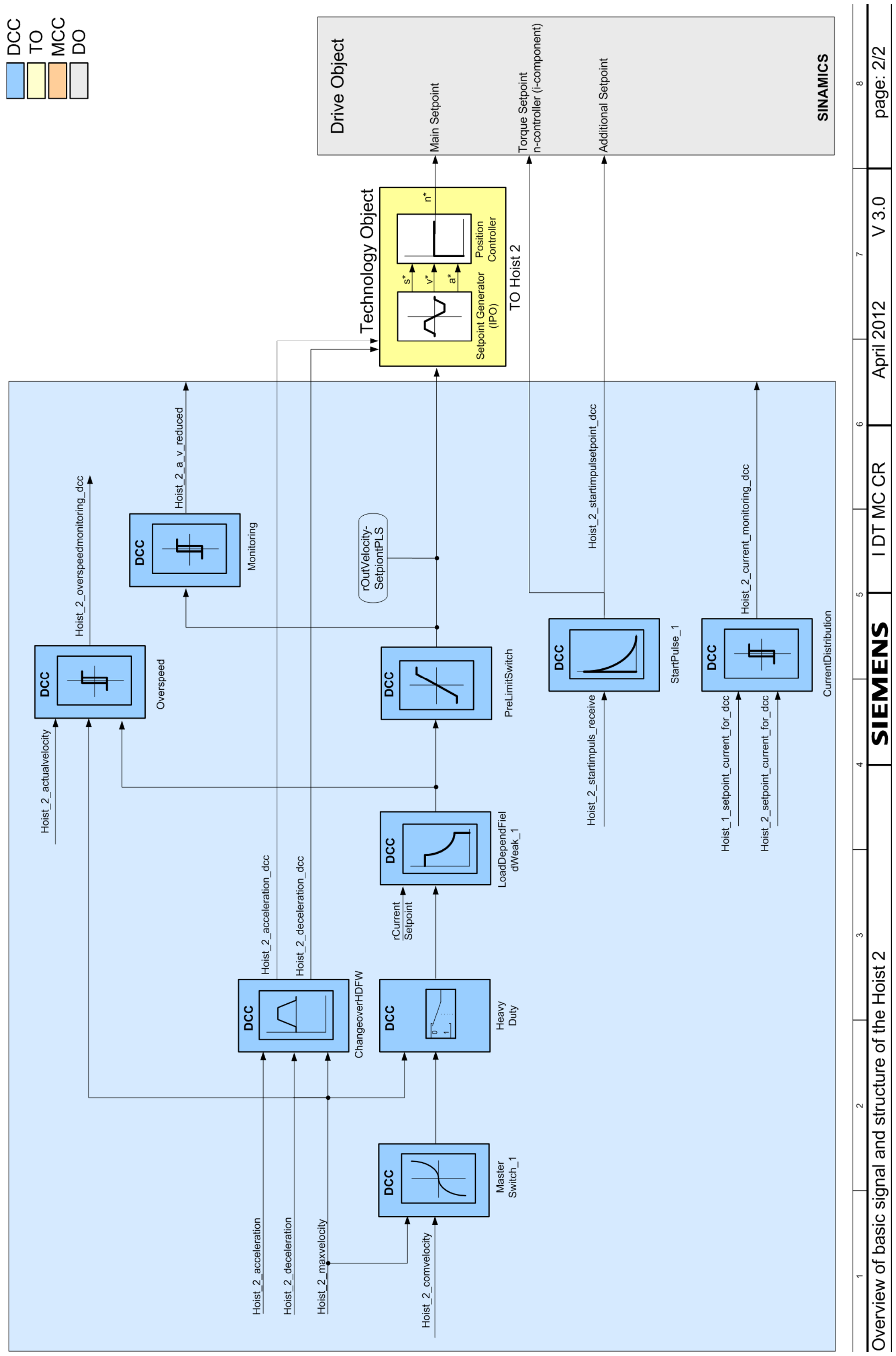


Figure 4-4 Hoist 2, basic signals 2

4.2 Trolley(Cross Travel)

The function module Trolley comprises two axes – Trolley_1 (master) and Trolley_2 (slave). Every axis has an MCC unit and a DCC chart. The units of the MCC program and the DCC blocks of the relevant function module are subdivided into the modules Trollp_1 (Master) / Trolld_1 and Trollp_2 (Slave) / Trolld_2. The MCC programs required are grouped in the Trollp_1 and Trollp_2 units. The DCC charts are grouped in the Trolld_1 and Trolld_2 units.

Using a data set switchover for the particular technology object (TO-Trolley_1 and TO-Trolley_2) - when required - it is possible to differentiate between the position actual value from the motor encoder (data set 1) and a position actual value from an external encoder (data set 2).

Using a data set switchover for the particular drive object (DO-Trolley_1 or DO-Trolley_2) it is possible to differentiate between closed-loop speed control with a motor encoder (data set 0) and closed-loop speed control without a motor encoder (data set 1).

Trollp_1 (FB blocks) comprise the following:

- FB_TelegramS7ToSimotion
- FB_ControlAxis
- FB_TractionControl
- FB_OperationMode
- FB_TelegramSinamicsToSimotion
- FB_TelegramSimotionToSinamics
- FB_ErrorPriority
- FB_TelegramSimotionToS7

Trollp_2 (FB blocks) comprise the following:

- FB_TelegramS7ToSimotion
- FB_ControlAxis
- FB_TractionControl
- FB_OperationMode
- FB_TelegramSinamicsToSimotion
- FB_TelegramSimotionToSinamics
- FB_ErrorPriority
- FB_TelegramSimotionToS7

Trollyd_1 (DCC blocks) comprise the following:

- MasterSwitch_1
- TractionControl
- PreLimitSwitch
- ChangeoverHDFW
- Monitoring

Trollyd_2 (DCC blocks) comprise the following:

- MasterSwitch_1
- TractionControl
- PreLimitSwitch
- ChangeoverHDFW
- CurrentDistribution
- Monitoring

Mon_Ref comprises:

- FB_Monitoring
- FB_ReferenceMode

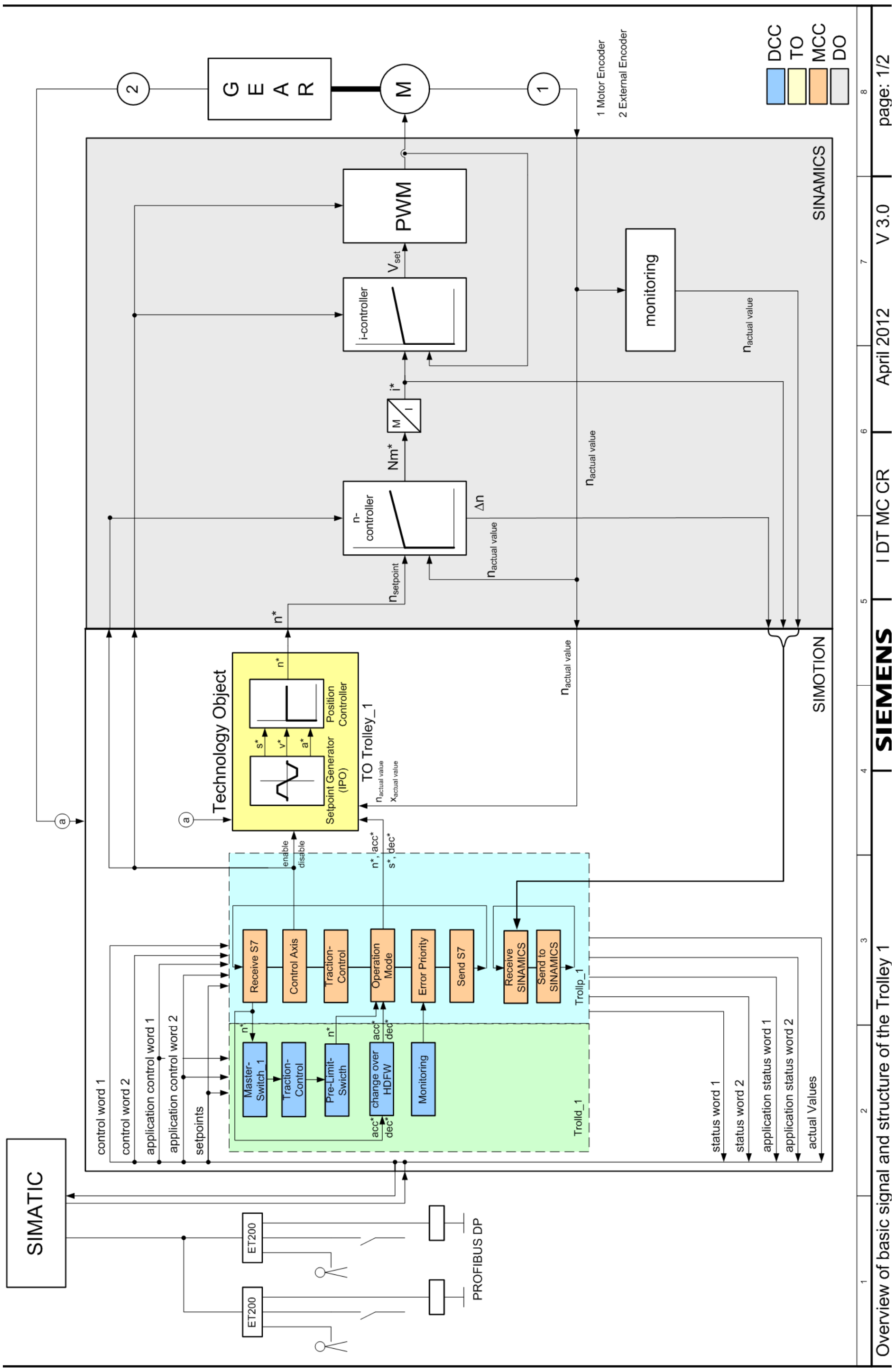


Figure 4-5 Trolley 1, basic signals_1

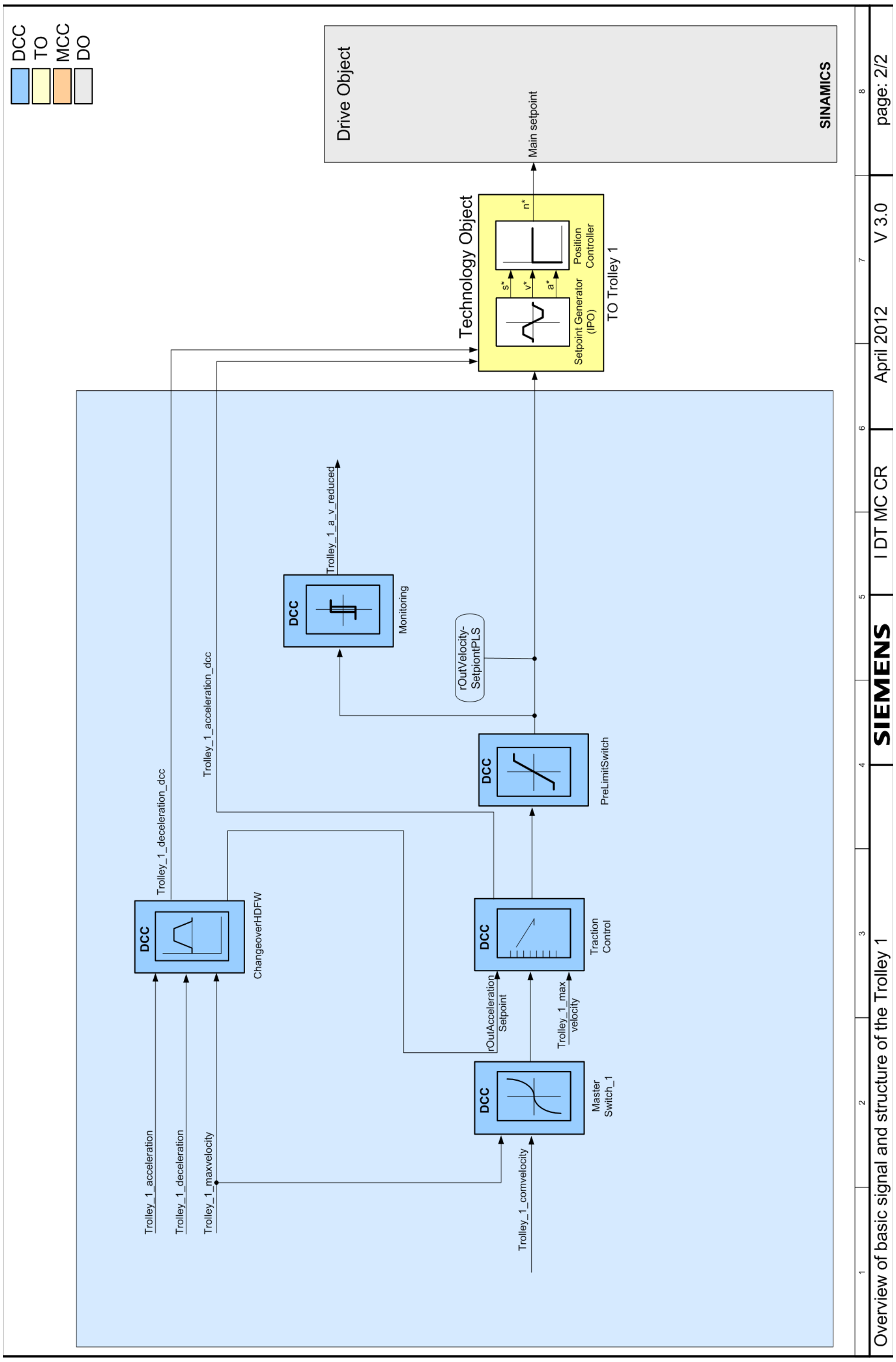


Figure 4-6 Trolley 1, basic signals_2

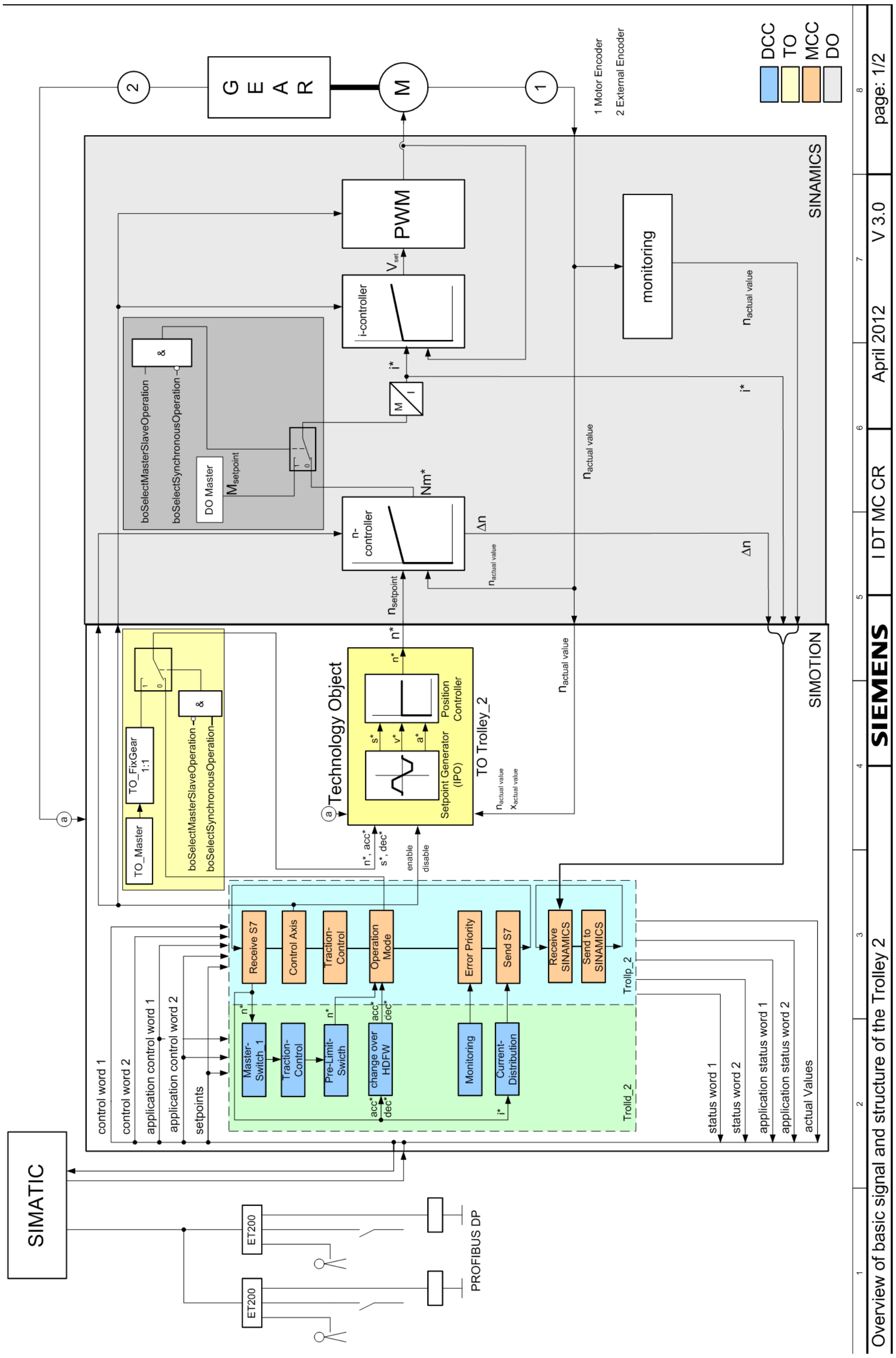


Figure 4-7 Trolley 2, basic signals_1

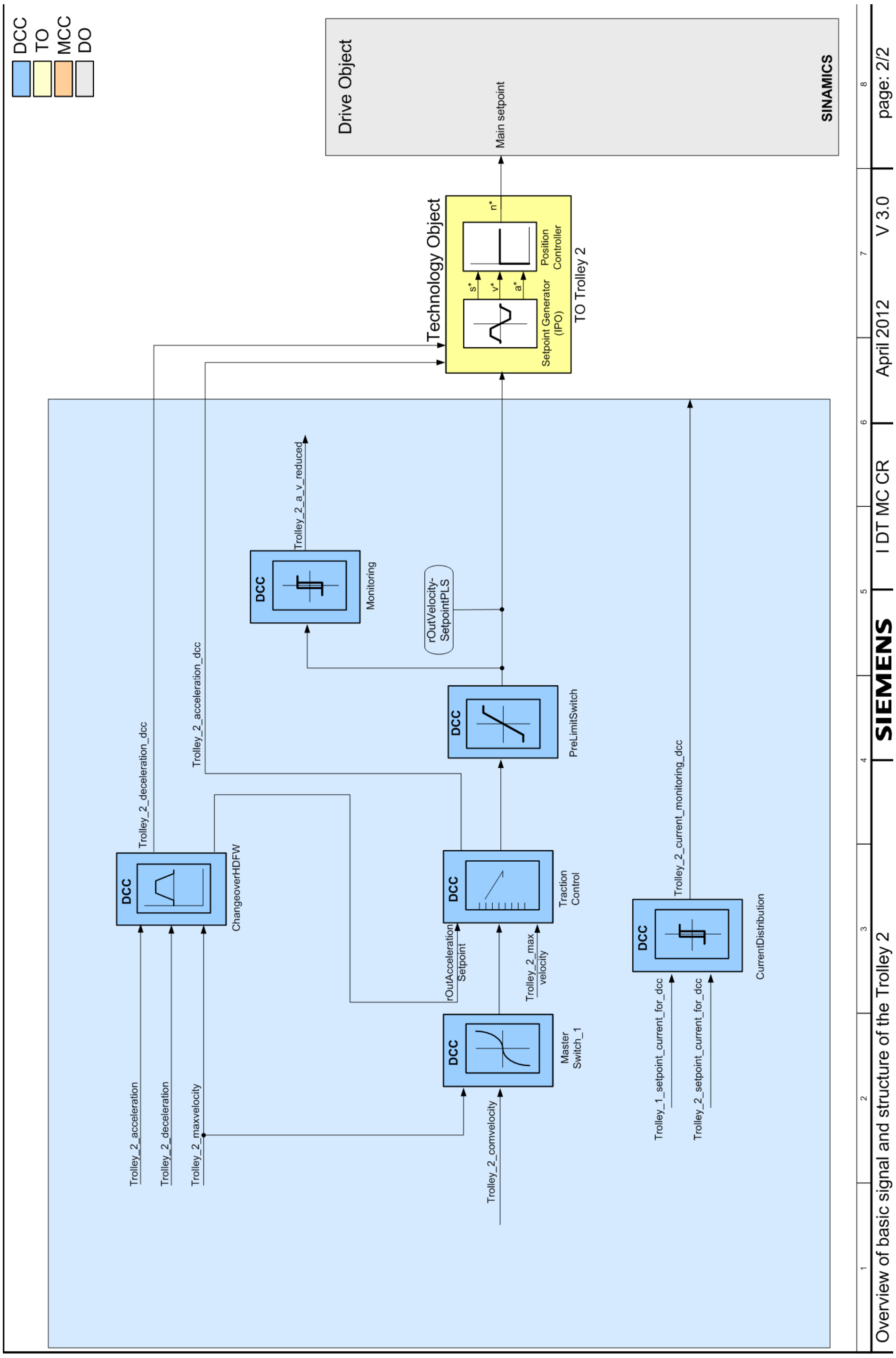


Figure 4-8 Trolley 2, basic signals_2

4.3 Gantry (Long Travel)

The function module Gantry comprises two axes – Gantry_1 (master) and Gantry_2 (slave). Every axis has an MCC unit and a DCC chart. The units of the MCC program and the DCC blocks of the relevant function module are subdivided into the modules Gantp_1 (master) / Gantd_1 and Gantp_2 (slave) / Gantd_2. The MCC programs required are grouped in the Gantp_1 and Gantp_2 units. The DCC charts are grouped in the Gantd_1 and Gantd_2 units.

Using a data set switchover for the particular technology object (TO-Gantry_1 and TO-Gantry_2) it is possible to differentiate between the position actual value from a motor encoder (data set 1) and a position actual value from an external encoder (data set 2).

Using a data set switchover for the particular drive object (DO-Gantry_1 & DO-Gantry_2) it is possible to differentiate between closed-loop speed control with a motor encoder (data set 0) and closed-loop speed control without a motor encoder (data set 1).

Gantp_1 (function blocks) comprises the following:

- FB_TelegramS7ToSimotion
- FB_ControlAxis
- FB_TractionControl
- FB_OperationMode
- FB_ErrorPriority
- FB_TelegramSimotionToS7
- FB_TelegramSinamicsToSimotion
- FB_TelegramSimotionToSinamics

Gantp_2 (function blocks) comprises the following:

- FB_TelegramS7ToSimotion
- FB_ControlAxis
- FB_TractionControl
- FB_OperationMode
- FB_ErrorPriority
- FB_TelegramSimotionToS7
- FB_TelegramSinamicsToSimotion
- FB_TelegramSimotionToSinamics

Gantd_1 (DCC blocks) comprises the following:

- MasterSwitch_1
- TractionControl
- PreLimitSwitch
- ChangeoverHDFW
- Monitoring

Gantd_2 (DCC blocks) comprises the following:

- MasterSwitch_1
- TractionControl
- PreLimitSwitch
- ChangeoverHDFW
- Monitoring
- CurrentDistribution

Mon_Ref (FB blocks) comprises the following:

- FB_Monitoring
- FB_ReferenceMode
- FB_Cornering

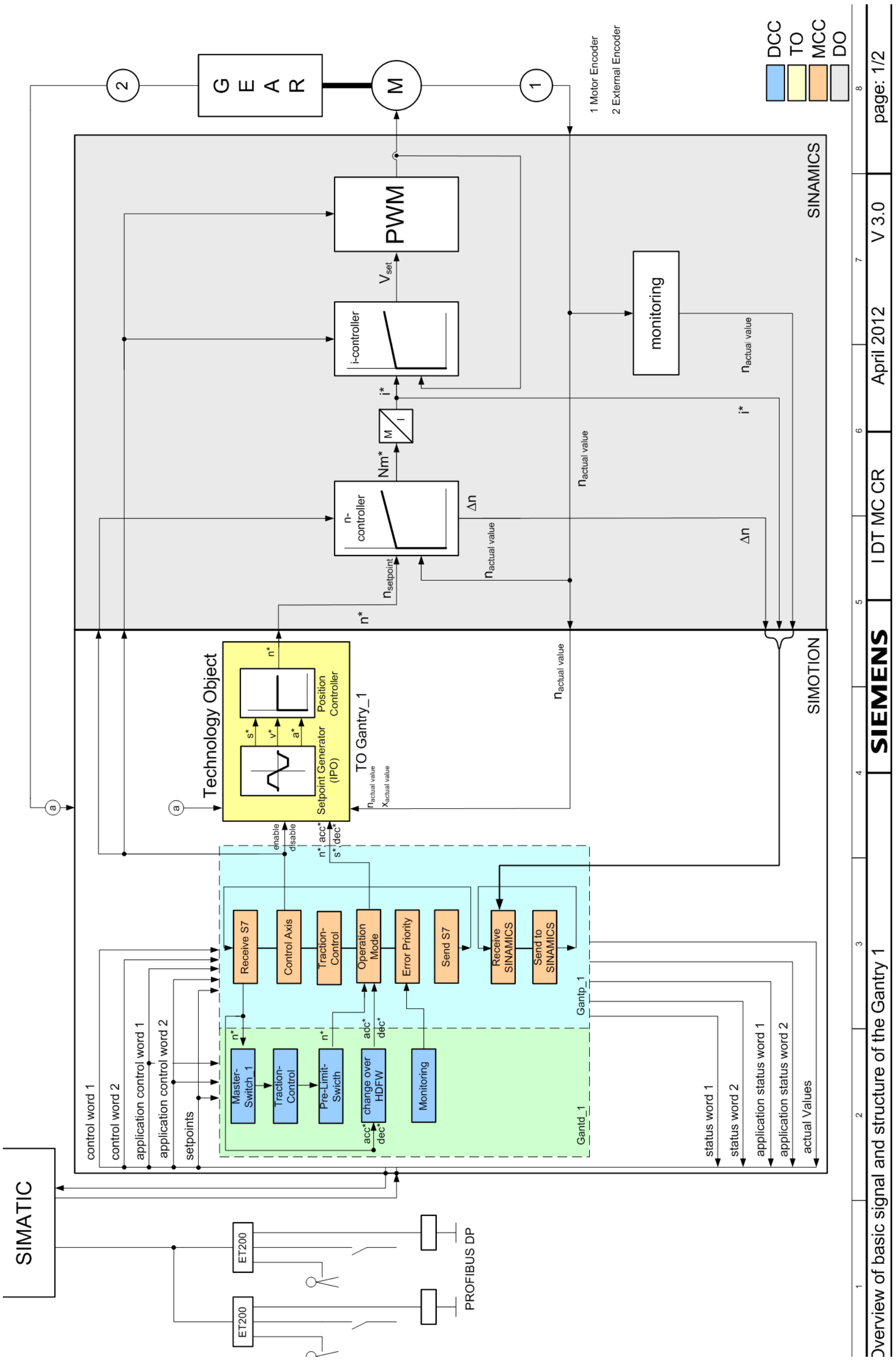


Figure 4-9 Gantry 1, basic signals_1

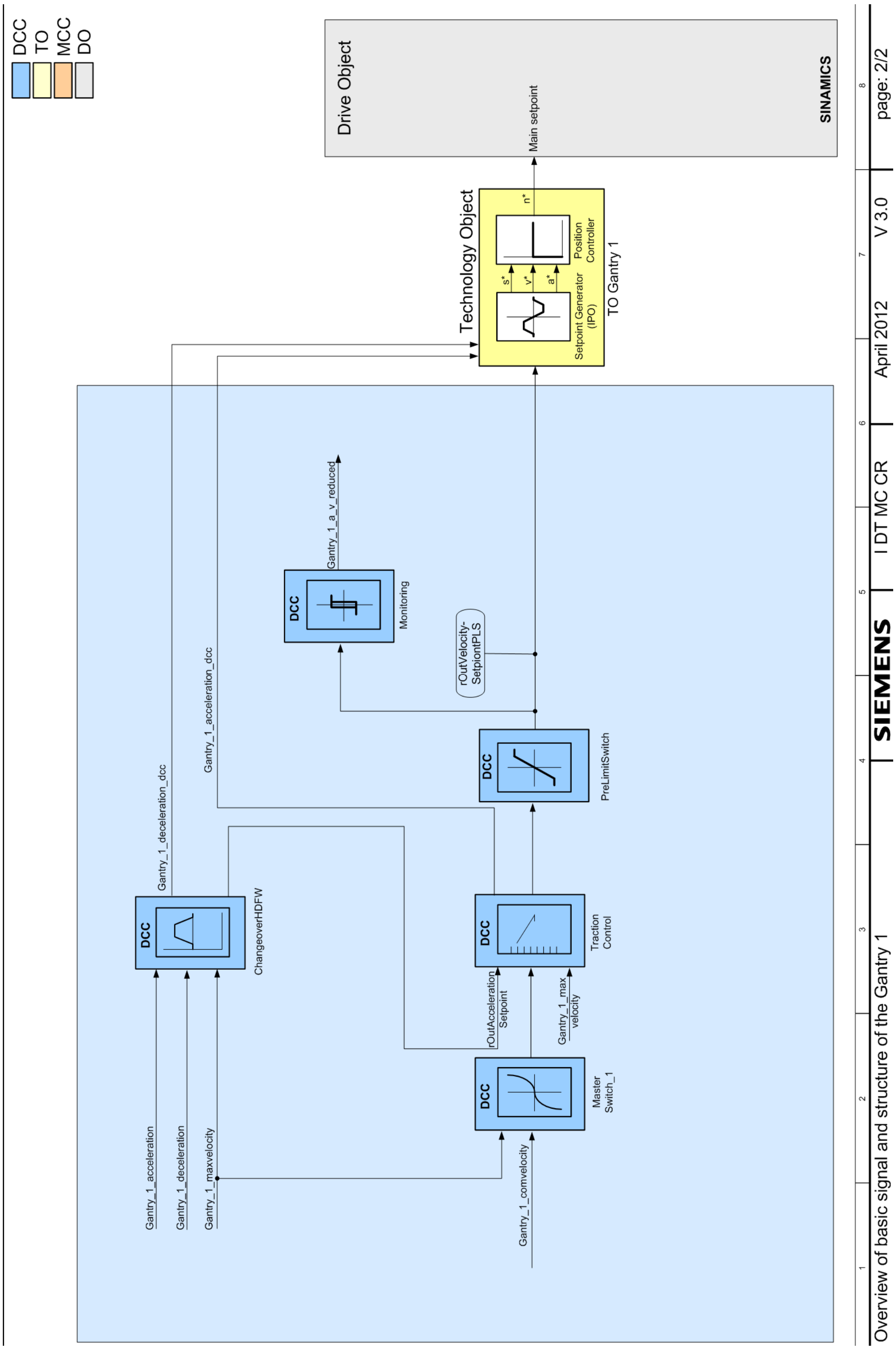


Figure 4-10 Gantry 1, basic signals_2

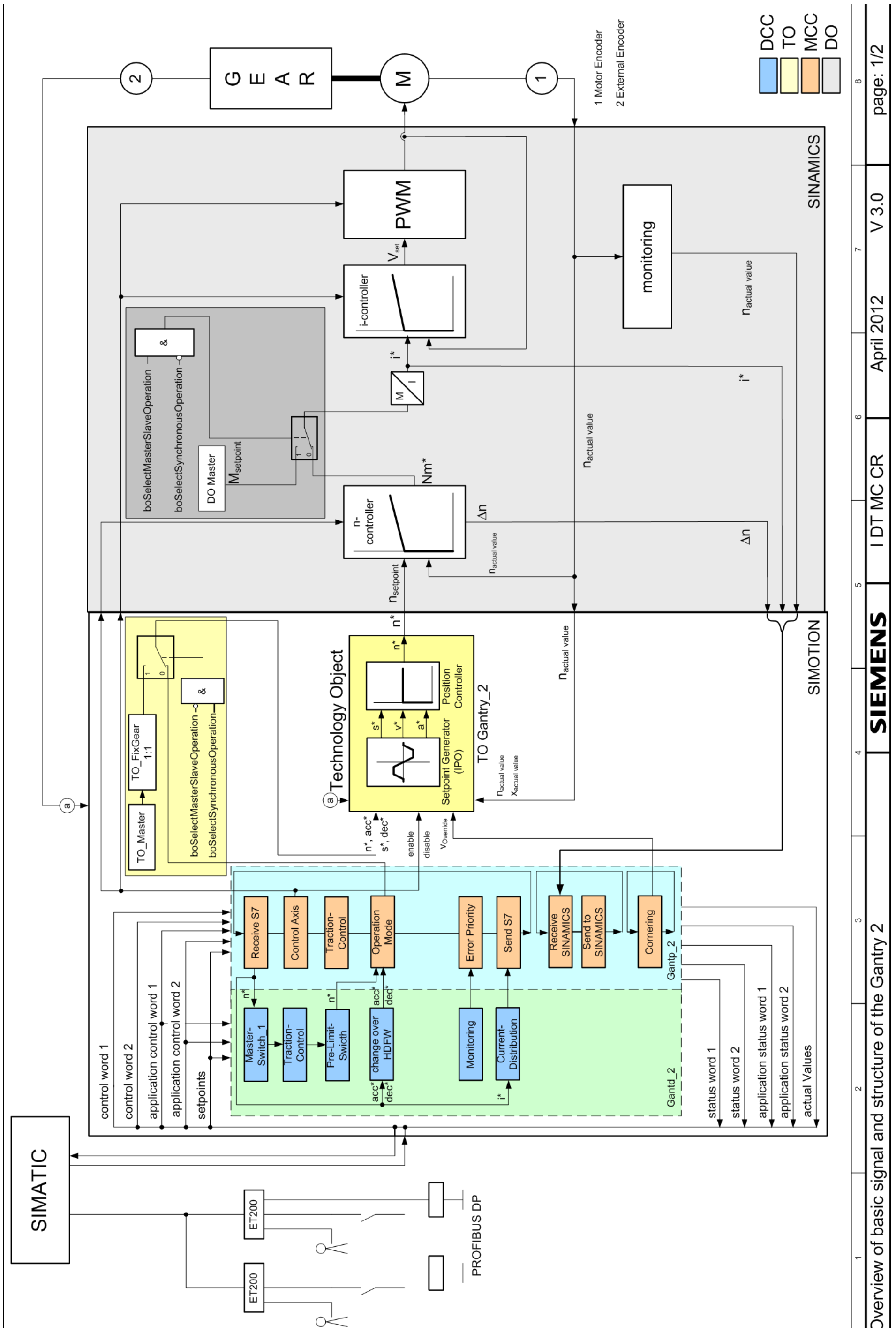


Figure 4-11 Gantry 2, basic signals_1

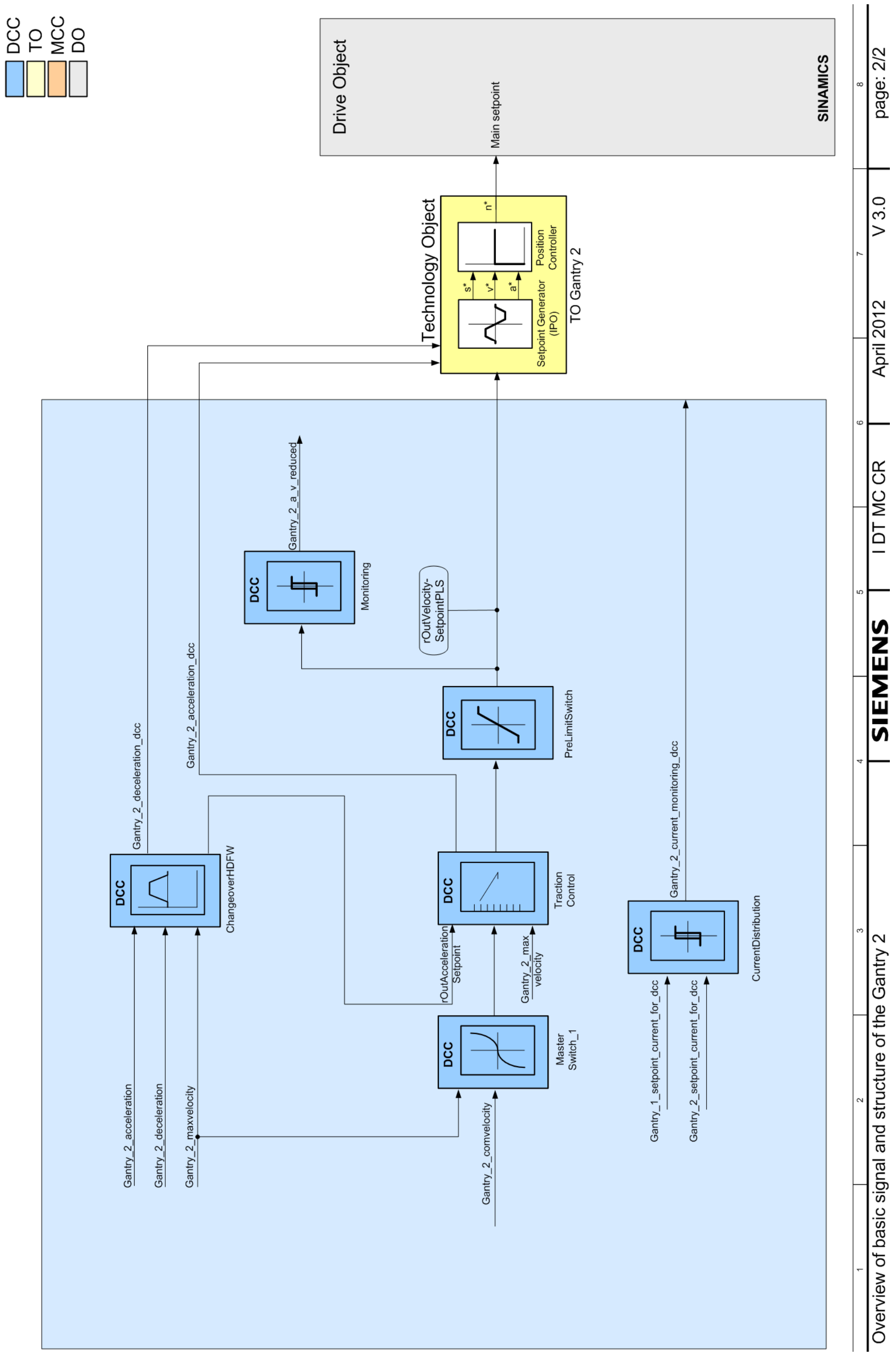


Figure 4-12 Gantry 2, basic signals_2

4.4 Boom

The Boom function module comprises a technology object (TO), an MCC unit, and a DCC chart, but no drive object (DO). By default, the Boom function module is not activated until an external TO switchover to the trolley is made. TO-Boom and TO-Trolley share a drive object (DO-Trolley).

A data set switchover for the associated drive object (DO-Trolley_1 or DO-Trolley_2) allows different motor or encoder data sets to be selected (data set 2/3).

CAUTION
"Speed control without motor encoder (data set 3)" control mode for boom is not permissible for safety reasons.

Boomp_1 (FB blocks) comprises the following:

- FB_TelegramS7ToSimotion
- FB_ControlAxis
- FB_OperationMode
- FB_ErrorPriority
- FB_TelegramSimotionToS7
- FB_TelegramSinamicsToSimotion
- FB_TelegramSimotionToSinamics

Boomd_1 (DCC blocks) comprises the following:

- MasterSwitch_1
- PreLimitSwitch
- StartPulse_1
- ChangeoverHDFW
- Monitoring

Boom_2 is used if the Boom motor is also being run with the Trolley_2 converter. The Boomp_2 and Boomd_2 programs are similar to Boomp_1 and Boomd_1. Boom_1 and Boom_2 so do not form a master-slave or synchronous axis.

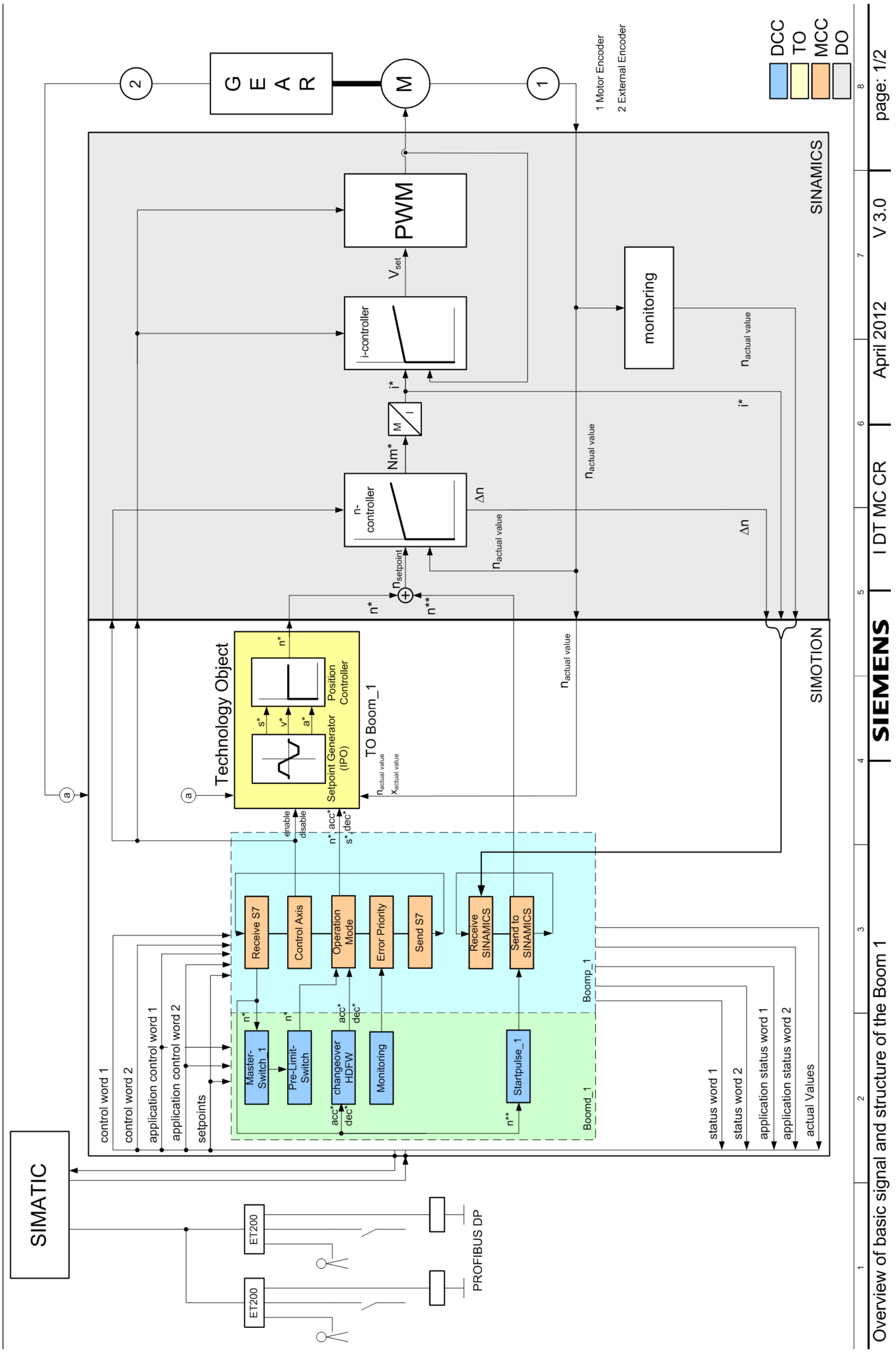


Figure 4-13 Boom 1, basic signals_1

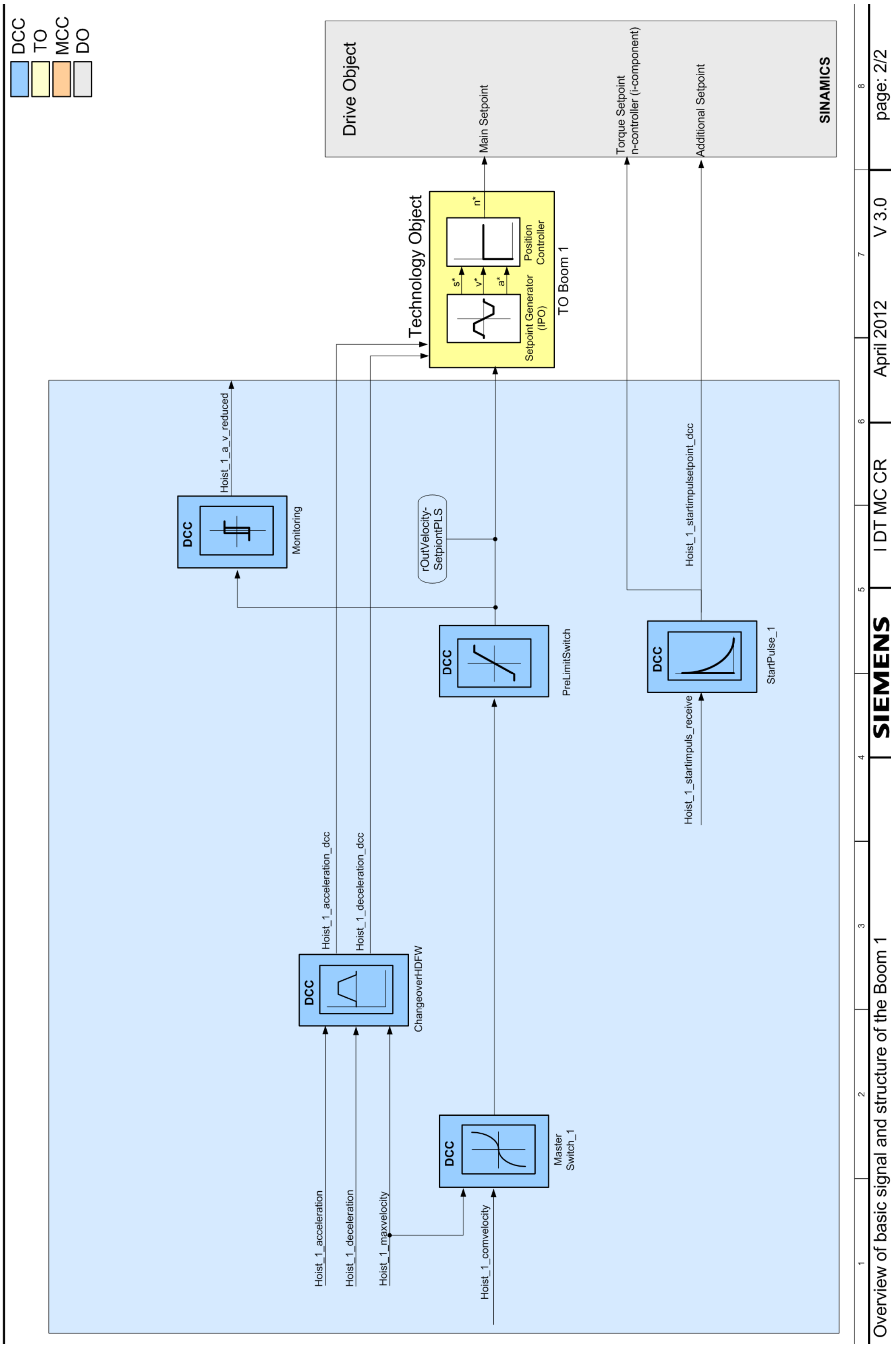


Figure 4-14 Boom 1, basic signals_2

4.5 Holding Gear

The function module Holding Gear comprises an MCC unit and a DCC chart. The required MCC programs are stored in unit Holdingp_1 and the DCC chart in Holdingd_1.

Using a data set switchover for the particular technology object (TO-HoldingGear_1) - when required - it is possible to differentiate between the position actual value from a motor encoder (data set 1) and a position actual value from an external encoder (data set 2).

A data set switchover for the drive object (DO-HoldingGear_1) allows different motor or encoder data sets to be selected (data set 0/1).

CAUTION
"Speed control without motor encoder (data set 1)" control mode for holding gear is not permissible for safety reasons.

Holdingp_1 (FB blocks) comprises the following:

- FB_TelegramS7ToSimotion
- FB_ControlAxis
- FB_OperationMode
- FB_ErrorPriority
- FB_TelegramSimotionToS7
- FB_TelegramSinamicsToSimotion
- FB_TelegramSimotionToSinamics

Holdingd_1 (DCC blocks) comprises the following:

- MasterSwitch_1
- HeavyDuty
- LoadDependFieldWeak_1
- PreLimitSwitch
- StartPulse_1
- SlackRopeControl_1
- ContinuousLoadMeas_1
- ChangeoverHDFW
- Monitoring

Mon_Ref comprises the following:

- FB_AutoSetting

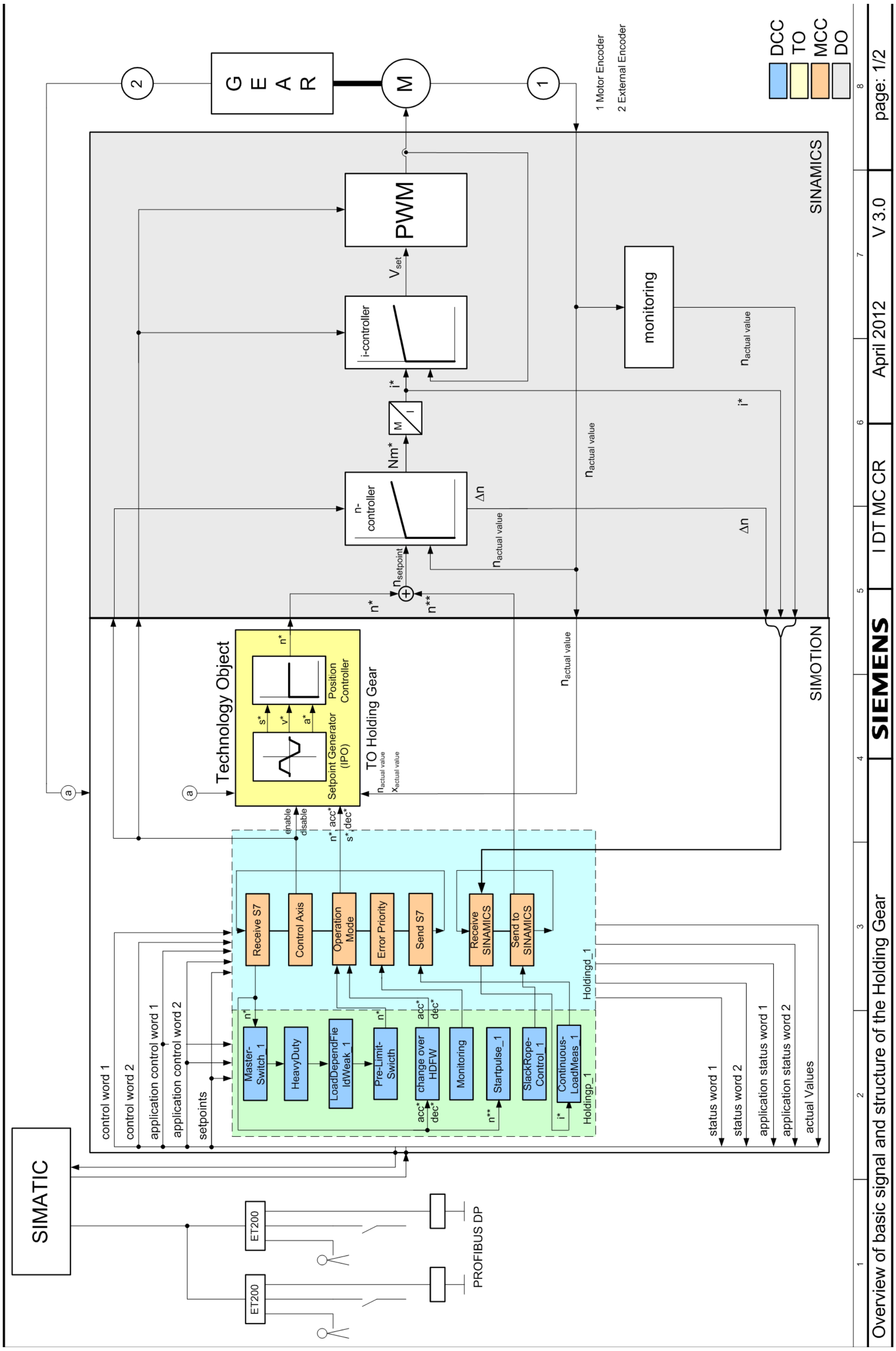


Figure 4-15 Holding Gear, basic signals 1

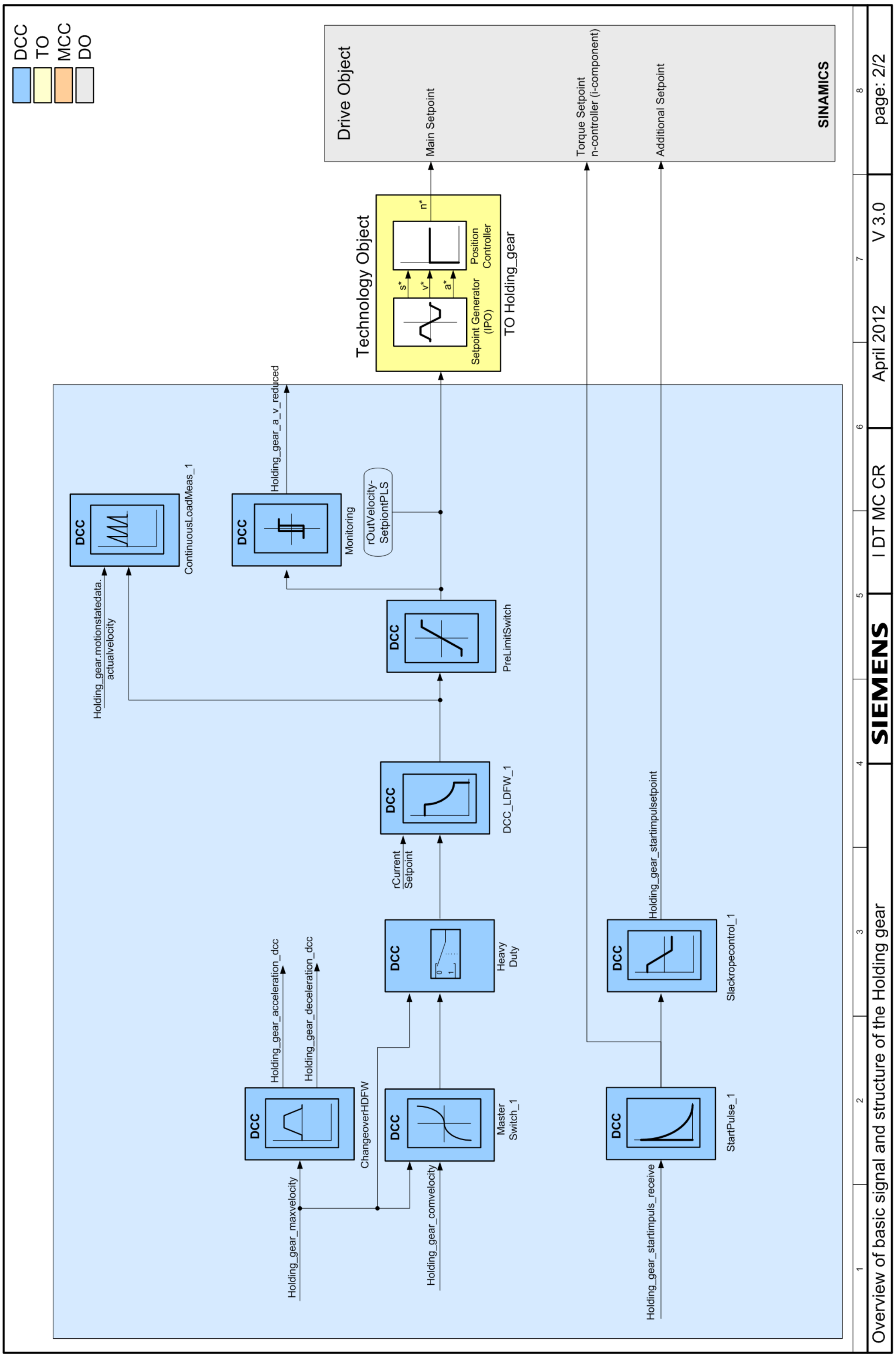


Figure 4-16 Holding Gear, basic signals 2

4.6 Closing Gear

The function module Closing Gear comprises an MCC unit and a DCC chart. The required MCC programs are stored in unit Closingp_1 and the DCC chart in Closingd_1.

Using a data set switchover for the particular technology object (TO-ClosingGear_1) - when required - it is possible to differentiate between the position actual value from a motor encoder (data set 1) and a position actual value from an external encoder (data set 2).

A data set switchover for the drive object (DO-ClosingGear_1) allows different motor or encoder data sets to be selected (data set 0/1).

CAUTION
"Speed control without motor encoder (data set 1)" control mode for closing gear is not permissible for safety reasons.

Closingp_1 (FB blocks) comprises the following:

- FB_TelegramS7ToSimotion
- FB_ControlAxis
- FB_OperationMode
- FB_ErrorPriority
- FB_TelegramSimotionToS7
- FB_TelegramSinamicsToSimotion
- FB_TelegramSimotionToSinamics

Closingd_1 (DCC blocks) comprises the following:

- MasterSwitch_1
- HeavyDuty
- PreLimitSwitch
- ChangeoverHDFW
- StartPulse_1
- CurrentEqualControl_1
- Monitoring
- GrabMonitor

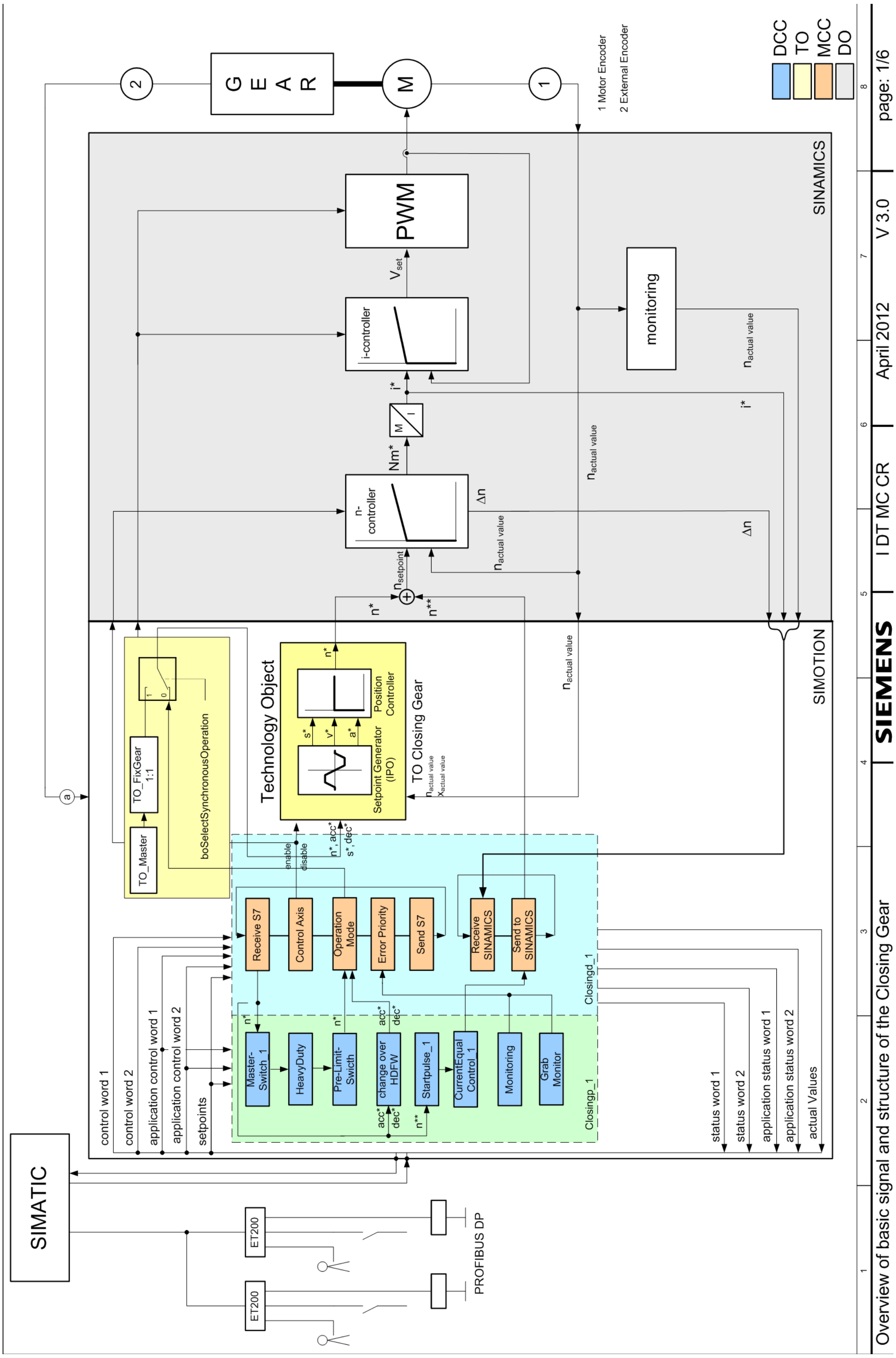


Figure 4-17 Closing Gear, basic signals_1

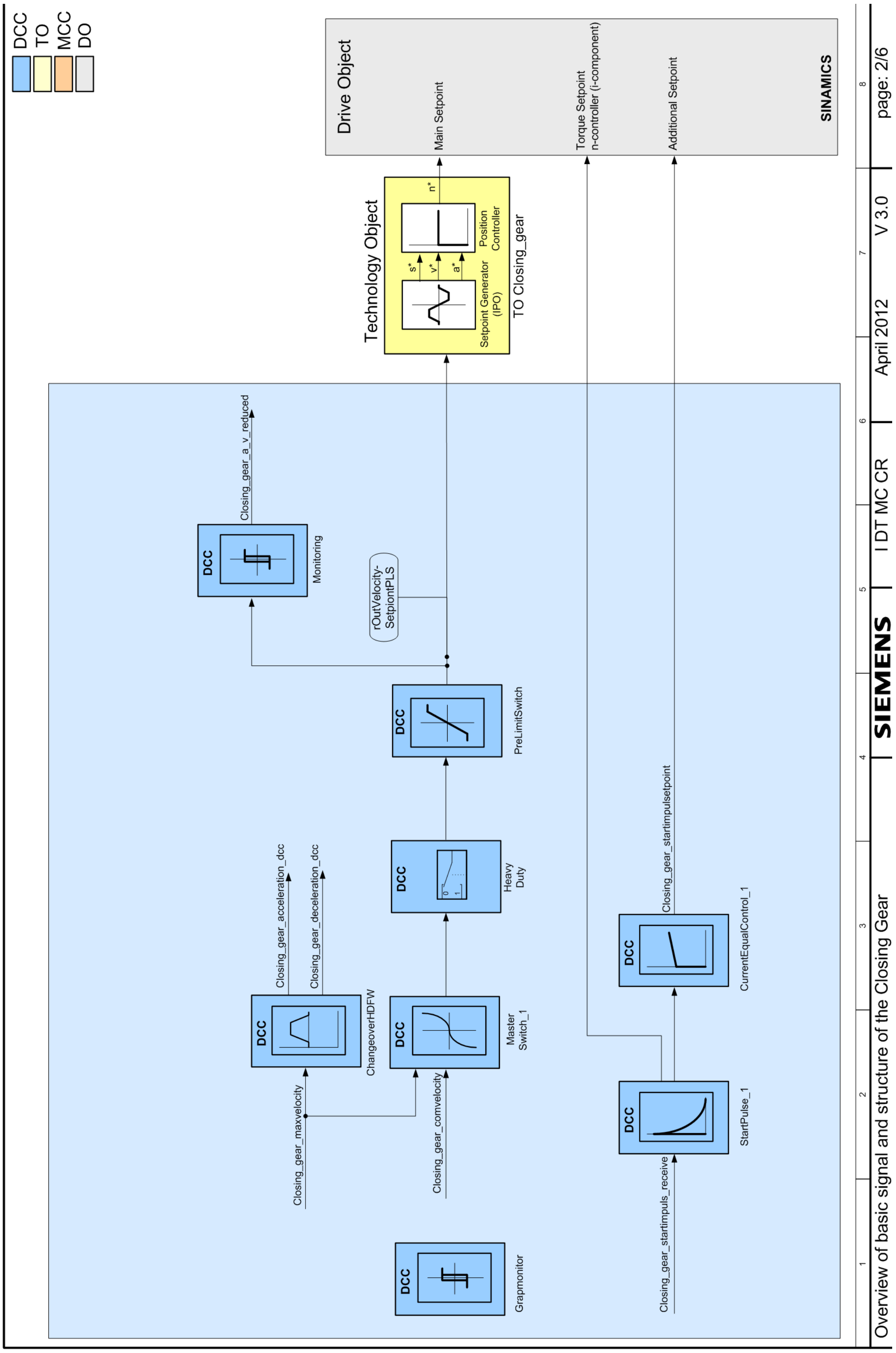


Figure 4-18 Closing Gear, basic signals_2

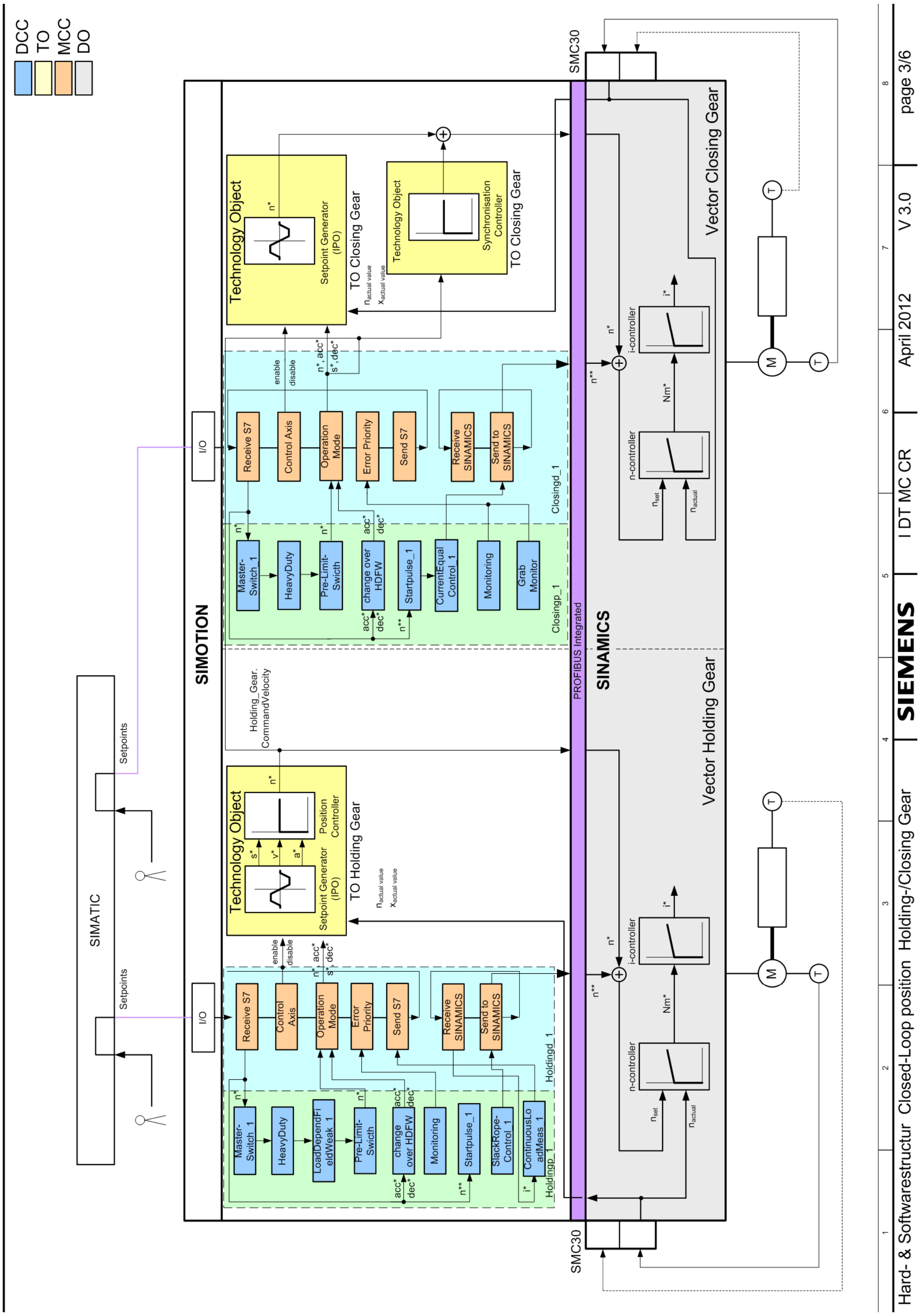


Figure 4-19 Closing Gear, basic signals_3

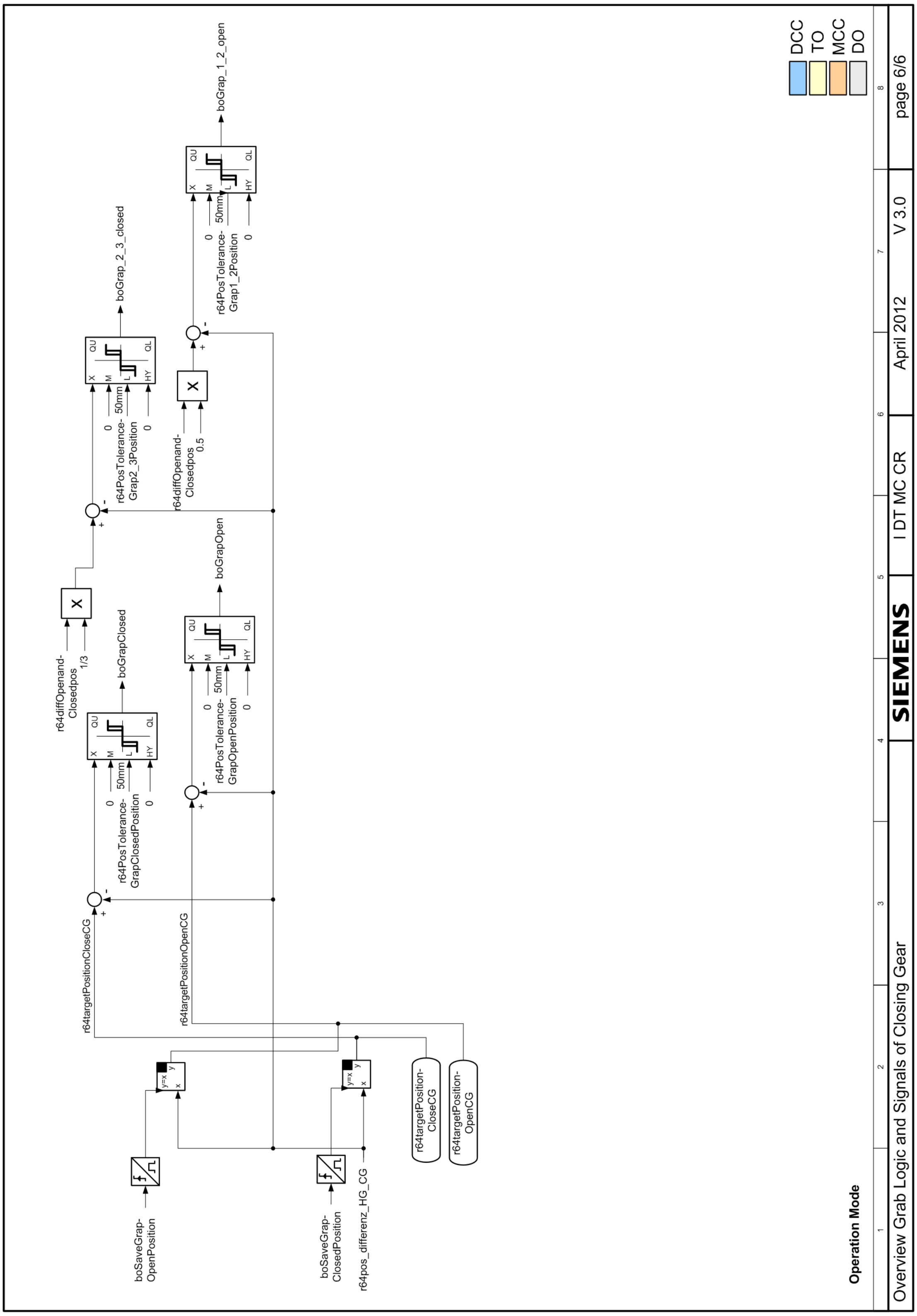


Figure 4-20 Closing Gear, basic signals_6

4.7 Slewing Gear

The function module Slewing Gear comprises an MCC unit and a DCC chart. The required MCC programs are stored in unit Slewingp_1 and the DCC chart in Slewingd_1.

Using a data set switchover for the technology object (TO-SlewingGear_1) it is possible to differentiate between the position actual value from a motor encoder (data set 1) and a position actual value from an external encoder (data set 2).

Using a data set switchover for the drive object (DO-SlewingGear_1) it is possible to differentiate between closed-loop speed control with a motor encoder (data set 0) and closed-loop speed control without a motor encoder (data set 1).

Slewingp_1 (FB blocks) comprise the following:

- FB_TelegramS7ToSimotion
- FB_ControlAxis
- FB_OperationMode
- FB_ErrorPriority
- FB_TelegramSimotionToS7
- FB_TelegramSinamicsToSimotion
- FB_TelegramSimotionToSinamics

Slewingd_1 (DCC blocks) comprise the following:

- MasterSwitch_1
- VelocityChangeSlewGear
- PreLimitSwitch
- AccelChangeSlewGear
- ChangeoverHDFW
- Monitoring

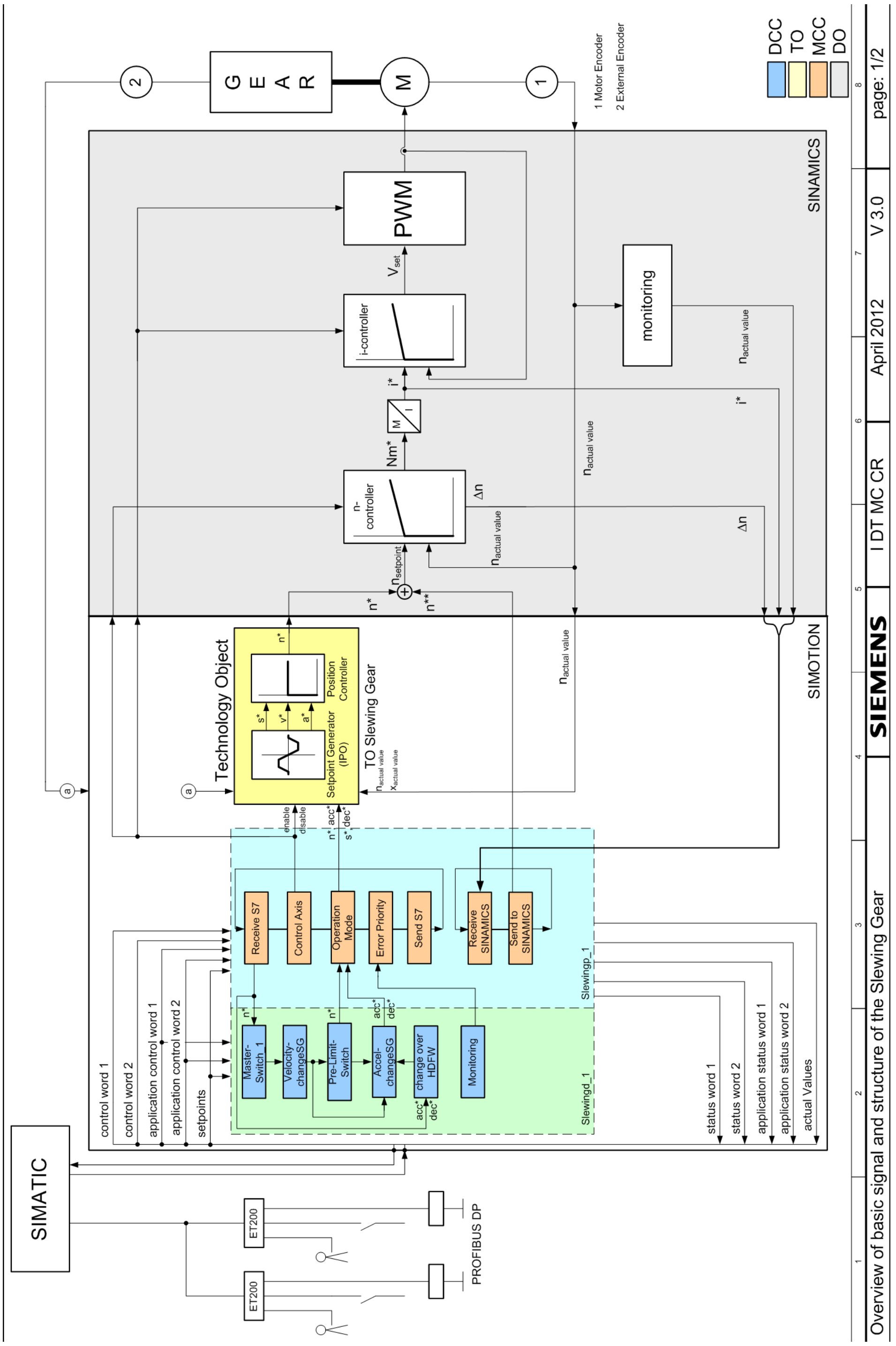


Figure 4-21 Slewing Gear, basic signals 1

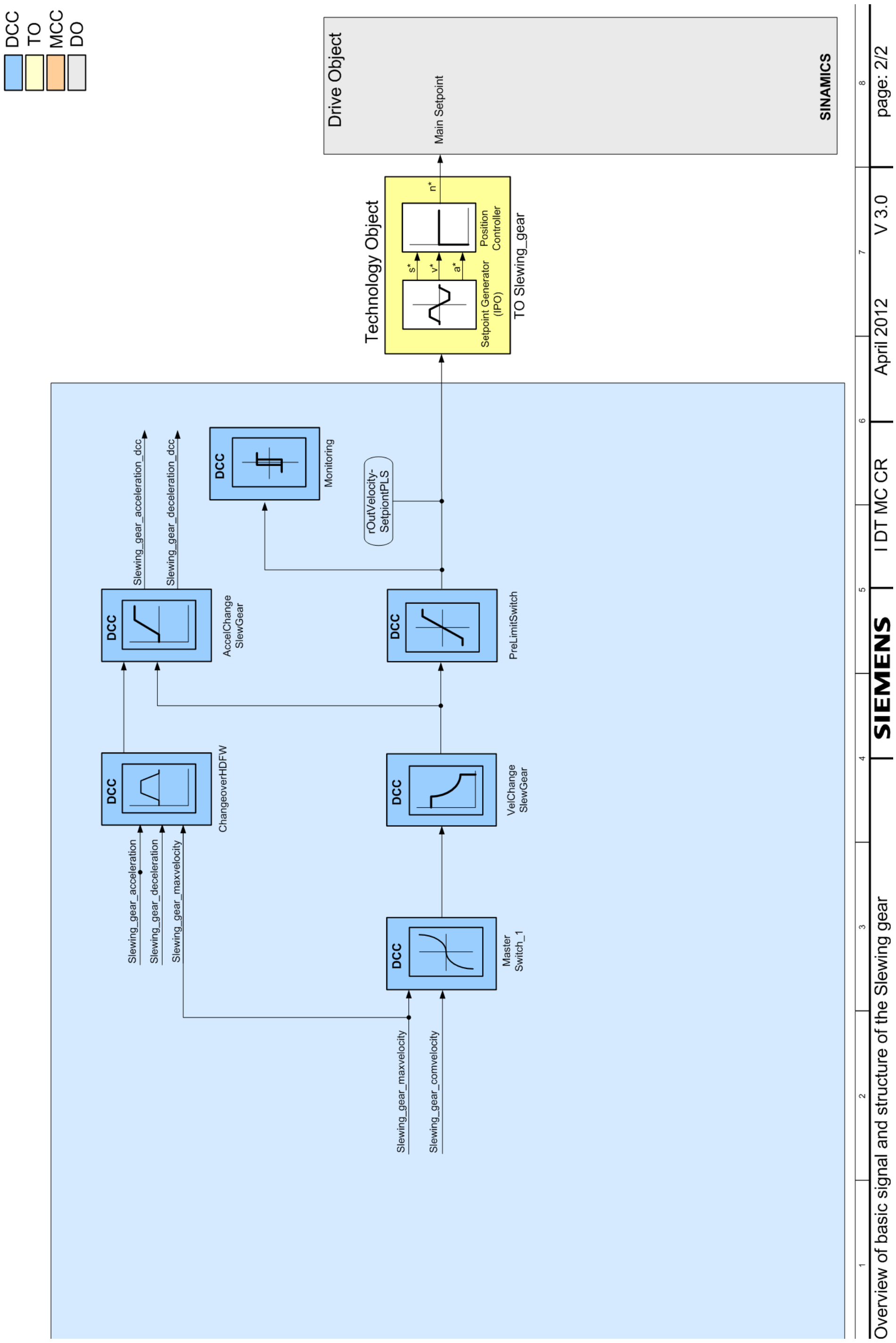


Figure 4-22 Slewing Gear, basic signals 2

Crane function blocks 5

5.1 Nomenclature of the crane function blocks

A function block in the Crane DCC and FB library is shown below as an example:

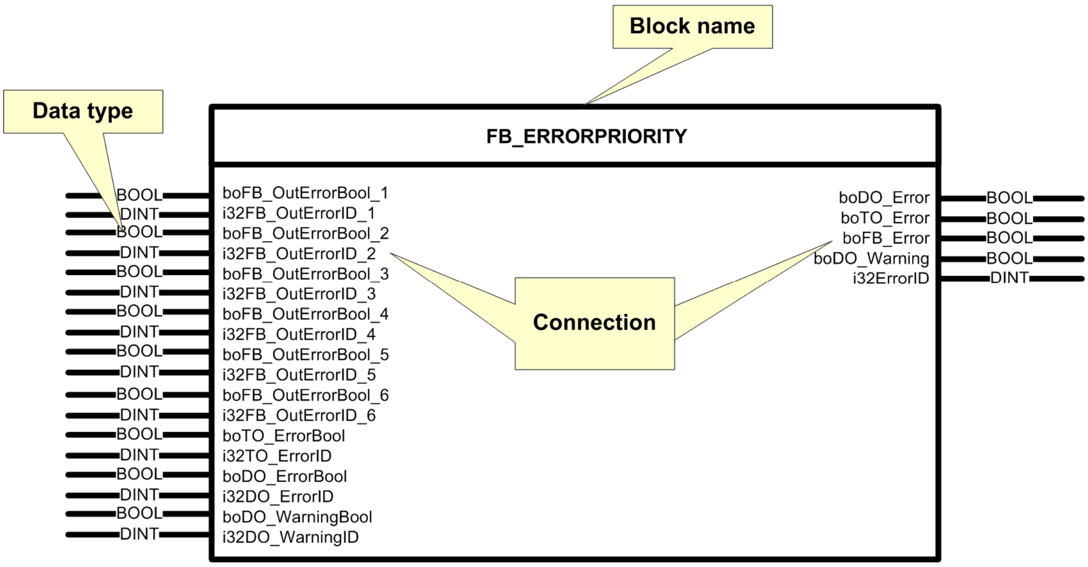


Figure 5-1 Symbol of a function block

Types of function blocks

Prefix	Type
DCC	Crane DCC function blocks
FB	Crane ST function block

Examples:

DCC_LoadDependingFieldWeak:
DCC block for the load-dependent field weakening

FB_OperationMode:
ST block for managing operating modes and travel motion

Connections and data types

The connection name must often be abbreviated due to the restricted number of characters. The abbreviated designators of the data types and prefixes are listed in the following table.

Prefix DCC	Prefix FB	Elementary data types (designator)		Range of values
bo	bo	BOOL	Bit (1)	TRUE, FALSE
b	b8	BYTE	Byte (8)	16#00...16#FF
b	b16 or w	WORD	Word (16)	-2**15...2**15 -1
b	b32 or dw	DWORD	Double word (32)	-2**31...2**31 -1
i	i16	INT	Integer number (16)	-2**15...2**15 -1
i	i32	DINT	Double integer number (32)	-2**31...2**31 -1
-	i16	UINT	Integer number (16) without sign	0…65535
r	r32	REAL	Floating-point number (32)	Refer to IEC 559
r	r64	LREAL	Long floating-point number (64)	Refer to IEC 559
r	r32	SDTIME Time	Real time data (as floating-point number)	0 ms ... 3.4 E38 ms
-	-	Anyobject	Connection of TO	-
-	-	ENUM	System data types for enumeration,	e.g. ACTIVE, INACTIVE

Connection type

Abbreviation	Connection type
IN	input
OUT	Output

Examples

Table 5- 1 DCC blocks

boEnableTC	BOOL	Input	Enables the traction control
rInStartpulseS7	REAL	Input	Start pulse value from the "S7"
boOut_EnableFieldWeak	BOOL	Output	Field weakening active
rOutStartpulse	REAL	Output	Start pulse output

Table 5- 2 Crane FB blocks

inSTW1bit0	BOOL	Input	Control word 1 bit 0
r64TargetPosition	LREAL	Input	Target position specification for the AUTOMATIC, MANUAL, EASY_POSITIONING operating modes
outSpeedControllerActive	BOOL	Output	Speed controller active
i32errorIDFunctionBlock	DINT	Output	Fault number from the function block, OperationMode

Note

"in" or "out" is not included in every connection name. This is only true for I/O variables of the same name.

5.2 Crane DCC library

5.2.1 General information

Note

The source code of every DCC block is closed. A license for SIMOTION Crane Basic Technology is required to use the Crane DCC library. If, when booting, the appropriate licenses are not found, then the application is brought into the stop state.

If SIMOTION SCOUT is reinstalled, the crane package (setupBasicTechnology) must also be reinstalled.

Descriptions of how to install and update the Crane_DCC_Library are provided in the section titled Setup and version update for crane libraries (Page 289).

If not otherwise noted, then the following statements apply to all DCC blocks:

Firmware version

- SIMOTION Drive Based (D4x5-2): Firmware as of V4.3 SP1
- SINAMICS: Firmware as of V4.5

Configuring

The DCC blocks can be configured in cyclic tasks.

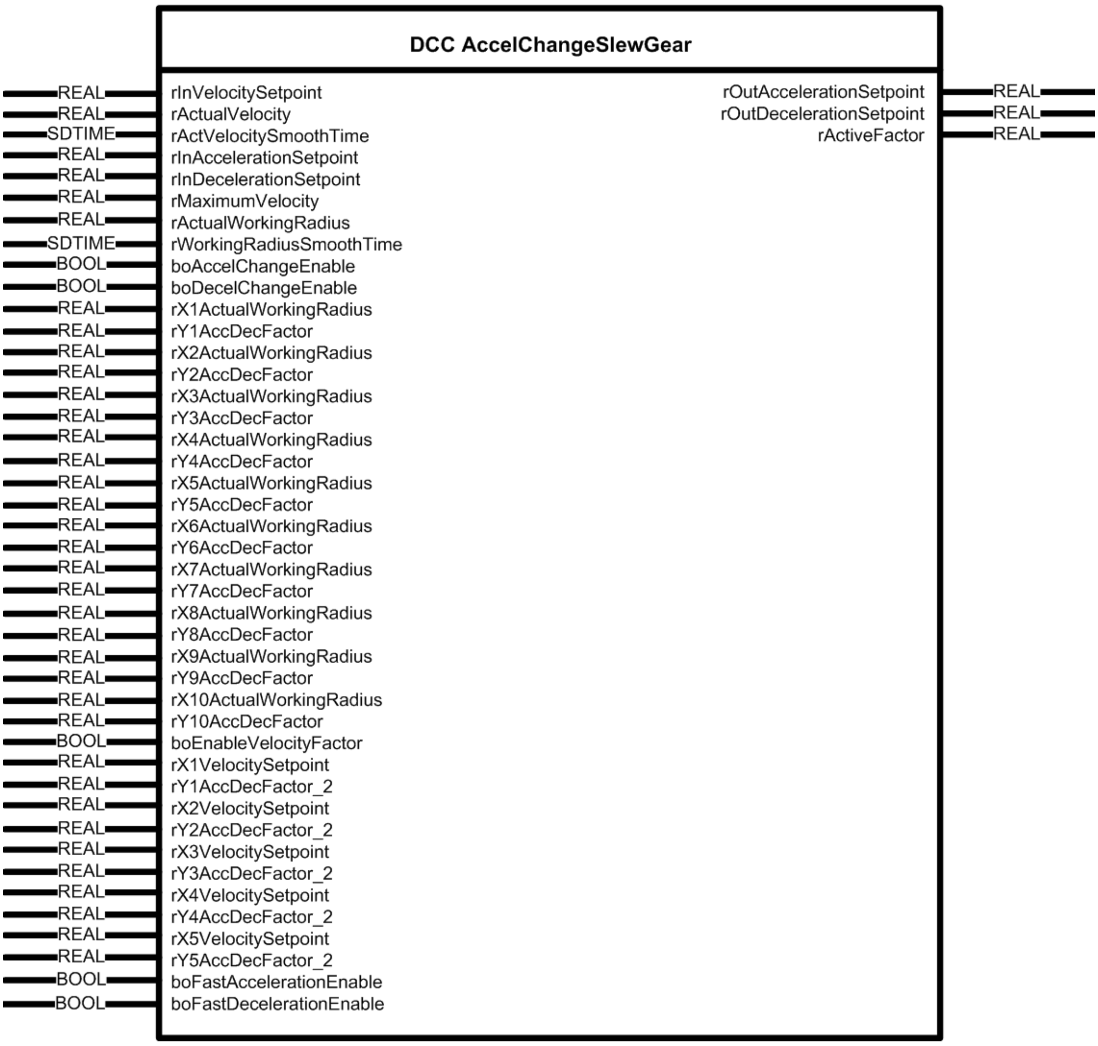
Explanation

All inputs/outputs of DCC blocks have a pre-assigned value. The value can be pre-assigned in three ways:

1. If the input/output is not assigned, then the default value applies.
2. The input/output is linked to a variable from the MCC interface section.
3. The input/output is linked to a variable from the DCC block.

5.2.2 DCC_AccelChangeSlewGear

Symbol



Note

With this DCC block, the term "velocity" refers to "angular velocity".

Brief description

The DCC block DCC_AccelChangeSlewGear allows the ramp-up time and/or the ramp-down time for a slewing gear to be optimally adjusted as a function of the working radius and/or the angular velocity (not too long but also not so short that the current limit is reached).

Mode of operation

The slewing gear on cranes is purely an acceleration drive. The larger the working radius of the luffing gear, the larger the accelerating torques for the slewing gear.

To prevent the current limit from being reached on ramp-up, the ramp-up time for the ramp-function generator of the slewing gear would normally need to be set for the largest working radius. However, this would waste time with smaller working radii. For this reason, either the ramp-up times or ramp-down times of the ramp-function generator are adjusted appropriately as a function of the working radius and setpoint angular velocity.

Applying an additional option allows the high-speed up-ramp and/or down-ramp to be used. In this case, the velocity evaluation is deactivated and only the acceleration and/or acceleration component dependent on the working radius is applied.

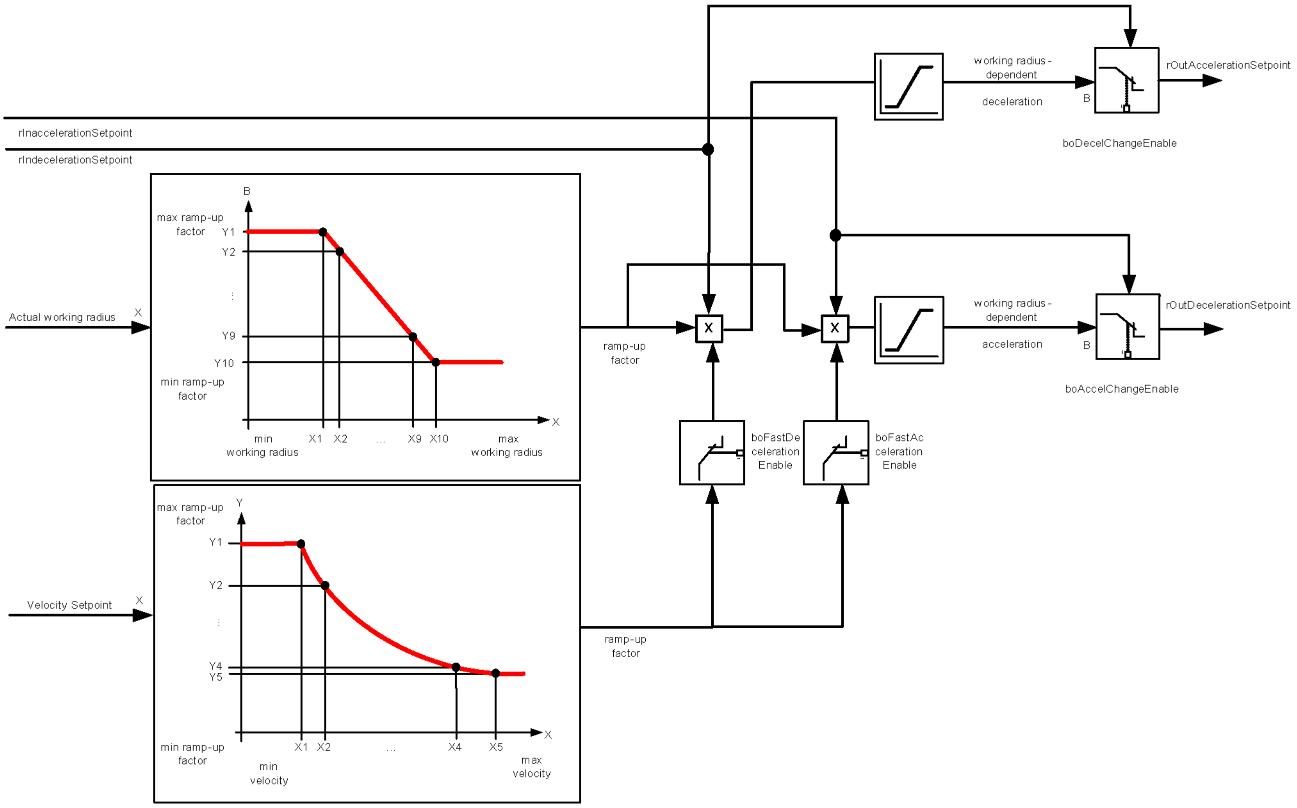


Figure 5-2 Operating principle of the block

Note

The effective acceleration factor is limited to a maximum of -200% and +200%.

NOTICE

Enter the maximum velocity in [mm/s] at input "rMaximumVelocity" as a fixed value, or interconnect the maximum velocity with this input in the user program (e.g. refer to Technology Object: D4x5.SlewingGear_1.TypeOfAxis.MaxVelocity.maximum or FB_OperationMode (Page 194), output variable r64maximumVelocity). The block does not work correctly otherwise.

⚠WARNING

Ensure that you enter the correct values for the polygon parameters! The input parameters "rX1ActualWorkingRadius" to "rX10ActualWorkingRadius" and "rX1VelocitySetpoint" to "rX5VelocitySetpoint" must be assigned in ascending sequence.

Note

The input parameters "rY1AccDecFactor" to "rY10AccDecFactor" and "rY1AccDecFactor_2" to "rY5AccDecFactor_2" must be assigned in descending order.

⚠ **WARNING**

Proper operation of the block cannot be guaranteed if you do not set the parameters correctly.

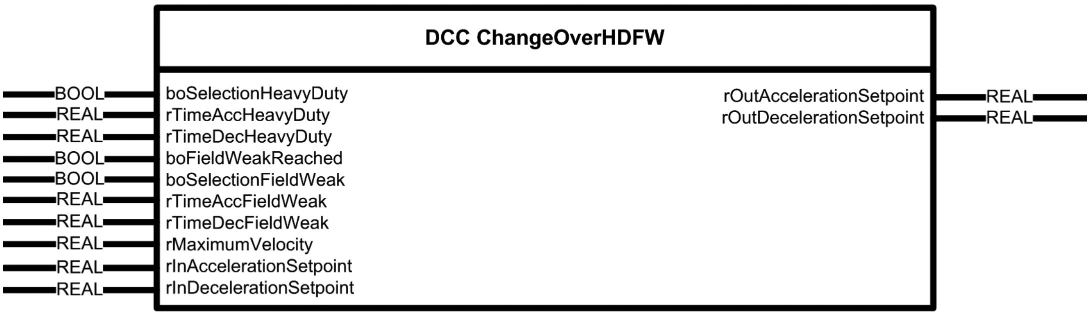
Connections

Name	Connection type	Default setting	Data type	Meaning
rInVelocitySetpoint	IN	DCC	REAL	Setpoint velocity [mm/s]
rActualVelocity	IN	MCC	REAL	Actual velocity [mm/s]
rActVelocitySmoothTime	IN	10	SDTIME	Smoothing time actual velocity [ms]
rInAccelerationSetpoint	IN	DCC	REAL	Setpoint acceleration [mm/s^2]
rInDecelerationSetpoint	IN	DCC	REAL	Setpoint deceleration [mm/s^2]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity (base velocity + field-weakening speed) [mm/s]
rActualWorkingRadius	IN	MCC	REAL	Actual working radius [mm]
rWorkingRadiusSmoothTime	IN	10	SDTIME	Smoothing time actual working radius [ms]
boAccelChangeEnable	IN	False	BOOL	Enable ramp-up time dependent on working radius: FALSE: Effective acceleration = setpoint acceleration TRUE: Changed acceleration
boDecelChangeEnable	IN	False	BOOL	Enable ramp-down time dependent on working radius: FALSE: Effective deceleration = setpoint deceleration TRUE: Changed deceleration
rX1ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X1 [mm]
rY1AccDecFactor	IN	100.0	REAL	Polygon acceleration factor Y1 [%]
rX2ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X2 [mm]
rY2AccDecFactor	IN	90.0	REAL	Polygon acceleration factor Y2 [%]
rX3ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X3 [mm]
rY3AccDecFactor	IN	85.0	REAL	Polygon acceleration factor Y3 [%]
rX4ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X4 [mm]
rY4AccDecFactor	IN	80.0	REAL	Polygon acceleration factor Y4 [%]
rX5ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X5 [mm]
rY5AccDecFactor	IN	75.0	REAL	Polygon acceleration factor Y5 [%]

Name	Connection type	Default setting	Data type	Meaning
rX6ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X6 [mm]
rY6AccDecFactor	IN	70.0	REAL	Polygon acceleration factor Y6 [%]
rX7ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X7 [mm]
rY7AccDecFactor	IN	65.0	REAL	Polygon acceleration factor Y7 [%]
rX8ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X8 [mm]
rY8AccDecFactor	IN	60.0	REAL	Polygon acceleration factor Y8 [%]
rX9ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X9 [mm]
rY9AccDecFactor	IN	55.0	REAL	Polygon acceleration factor Y9 [%]
rX10ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X10 [mm]
rY10AccDecFactor	IN	50.0	REAL	Polygon acceleration factor Y10 [%]
boEnableVelocityFactor	IN	False	BOOL	either constant (100%) or dependent on velocity: FALSE: Constant = 100% TRUE: Velocity-dependent
rX1VelocitySetpoint	IN	0.0	REAL	Polygon velocity setpoint X1 [%]
rY1AccDecFactor_2	IN	100.0	REAL	Polygon acceleration factor Y1 [%]
rX2VelocitySetpoint	IN	25.0	REAL	Polygon velocity setpoint X2 [%]
rY2AccDecFactor_2	IN	90.0	REAL	Polygon acceleration factor Y2 [%]
rX3VelocitySetpoint	IN	50.0	REAL	Polygon velocity setpoint X3 [%]
rY3AccDecFactor_2	IN	75.0	REAL	Polygon acceleration factor Y3 [%]
rX4VelocitySetpoint	IN	75.0	REAL	Polygon velocity setpoint X4 [%]
rY4AccDecFactor_2	IN	60.0	REAL	Polygon acceleration factor Y4 [%]
rX5VelocitySetpoint	IN	100.0	REAL	Polygon velocity setpoint X5 [%]
rY5AccDecFactor_2	IN	50.0	REAL	Polygon acceleration factor Y5 [%]
boFastAccelerationEnable	IN	False	BOOL	Enable for high-speed acceleration only: FALSE: Effective acceleration = setpoint deceleration x effective acceleration factor x effective velocity factor TRUE: Effective acceleration = setpoint acceleration x effective acceleration factor
boFastDecelerationEnable	IN	False	BOOL	Enable for high-speed deceleration only: FALSE: Effective deceleration = setpoint deceleration x effective acceleration factor x effective velocity factor TRUE: Effective deceleration = setpoint deceleration x effective deceleration factor
rOutAccelerationSetpoint	OUT	MCC DCC	REAL	Effective acceleration [mm/s^2]
rOutDecelerationSetpoint	OUT	MCC DCC	REAL	Effective deceleration [mm/s^2]
rActiveFactor	OUT	0.0	REAL	Effective acceleration factor [%]

5.2.3 DCC_ChangeOverHDFW

Symbol


DCC ChangeOverHDFW
BOOL boSelectionHeavyDuty
REAL rTimeAccHeavyDuty
REAL rTimeDecHeavyDuty
BOOL boFieldWeakReached
BOOL boSelectionFieldWeak
REAL rTimeAccFieldWeak
REAL rTimeDecFieldWeak
REAL rMaximumVelocity
REAL rInAccelerationSetpoint
REAL rInDecelerationSetpoint
rOutAccelerationSetpoint REAL
rOutDecelerationSetpoint REAL

Brief description

Using the DCC block DCC_ChangeOverHDFW, the acceleration and deceleration times are modified in heavy-duty operation (HeavyDuty) or in field weakening (FieldWeakening).

Mode of operation

This function block is required, for example, for Hoist with field weakening in order to prevent the current limits from being reached during acceleration in the field-weakening range.

At low loads (steady-state load torque), in order to increase the handling capacity, the speed is increased to a value that is permissible for a particular load by reducing the field current. This means that in the field-weakening range, the acceleration torque available is also correspondingly reduced. For the steady-state load torque and the acceleration torque of the load, only the lower torque corresponding to the lower load is required. However, the acceleration torque of the rotating masses remains the same due to the fact that the moments of inertia have not changed (load dependent).

For hoists, the acceleration and deceleration times are normally set so that the drive, when accelerating and hoisting with rated load and at the full motor field, doesn't quite reach the current limit so that acceleration and deceleration distances are as short as possible.

If the acceleration and deceleration times are set so that the drive (also in field weakening) when accelerating does not reach the current limit, then with constant acceleration in the rated speed and field-weakening range, extremely long approach and deceleration distances are obtained. In order to optimize these, the ramp-up time can be extended when transitioning into the field-weakening range.

For heavy duty, to reduce the acceleration torques, when heavy-duty operation is selected, then a long ramp-up time is immediately switched-in.

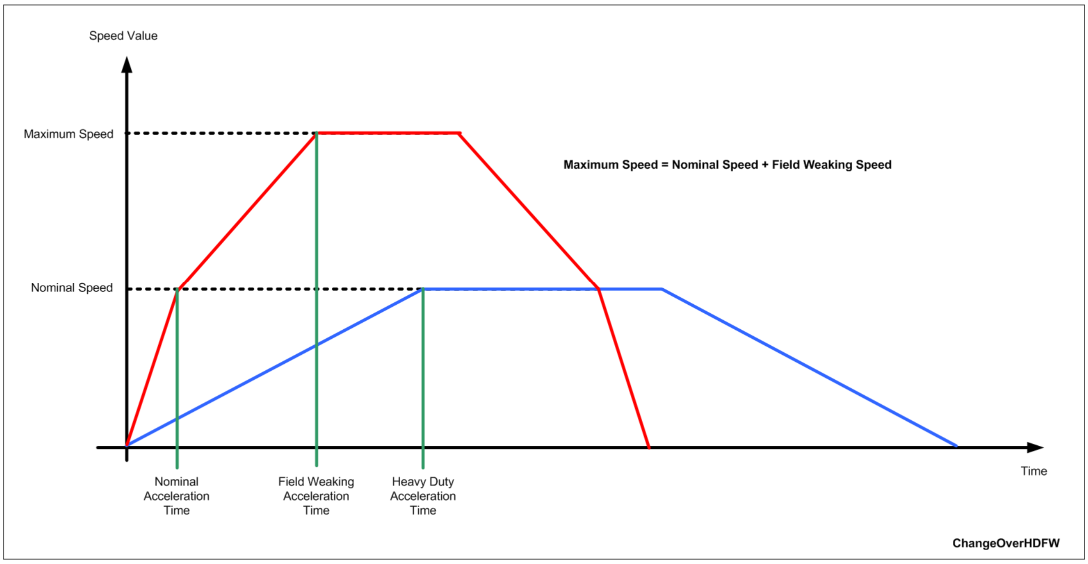


Figure 5-3 Changing the ramp-up/ramp-down times for field weakening and heavy-duty operation

CAUTION
Consult the crane manufacturer for basic information about data settings. Carry out the procedure described below, however, only if corresponding information is not available.

Note

Change of the velocity setpoint, acceleration time, delay time, initial rounding-off and final rounding-off

a) The velocity, acceleration and deceleration can be changed at any time in the SPEED_CONTROLLED operating mode.

b) The velocity and acceleration can be changed at any time in the AUTOMATIC, EASY_POSITIONING and MANUAL operating modes.
A longer delay time is not applied in the delay phase. A shorter delay time, however, will be applied. If the delay time is reduced during the braking phase, the target position is no longer approached, but rather the drive is braked with the shorter delay time.

c) In the SENSORLESS EMERGENCY operating mode, the velocity can only be changed up to the parameterizable limit "r64VelocitySensorlessEmergency".

d) In all operating modes, a change of the initial rounding-off and the final rounding-off is applied only with a deflected master switch. The exception is the AUTOMATIC operating mode. In this case, the change is applied immediately.

NOTICE

Enter the maximum velocity in [mm/s] at input "rMaximumVelocity" as fixed value, or interconnect the maximum velocity with this input in the user program (e.g.: refer to Technology Object: D4x5.Gantry_1.TypeOfAxis.MaxVelocity.maximum, or function block FB_OperationMode output variable r64maximumVelocity). The block does not work correctly otherwise.

Note

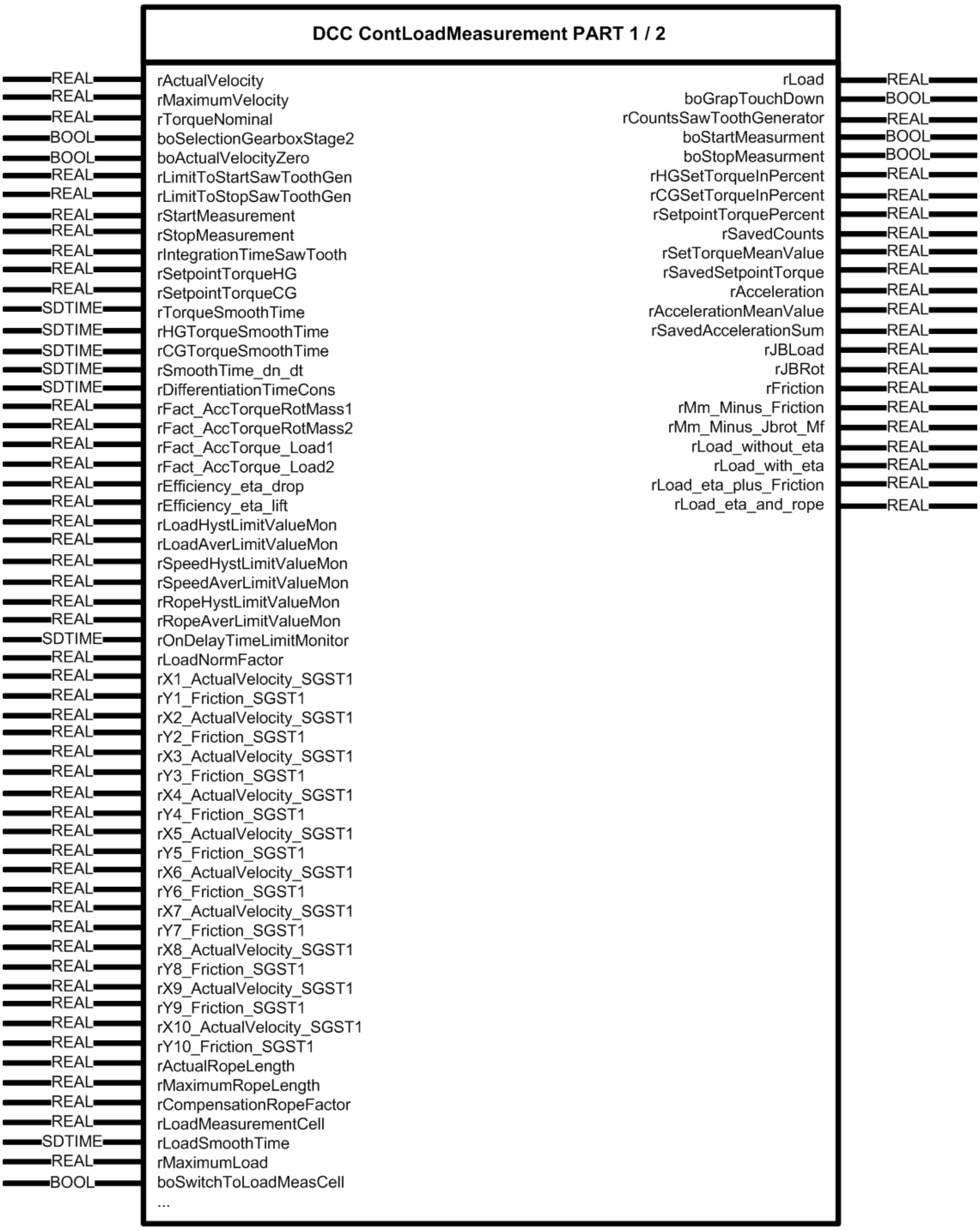
Values for ramp-up/ramp-down time ≤ 0 do not make any sense. The value 0 would mean an infinite acceleration. The higher the value, the smaller the acceleration or deceleration, and so the more current available for the static torque.

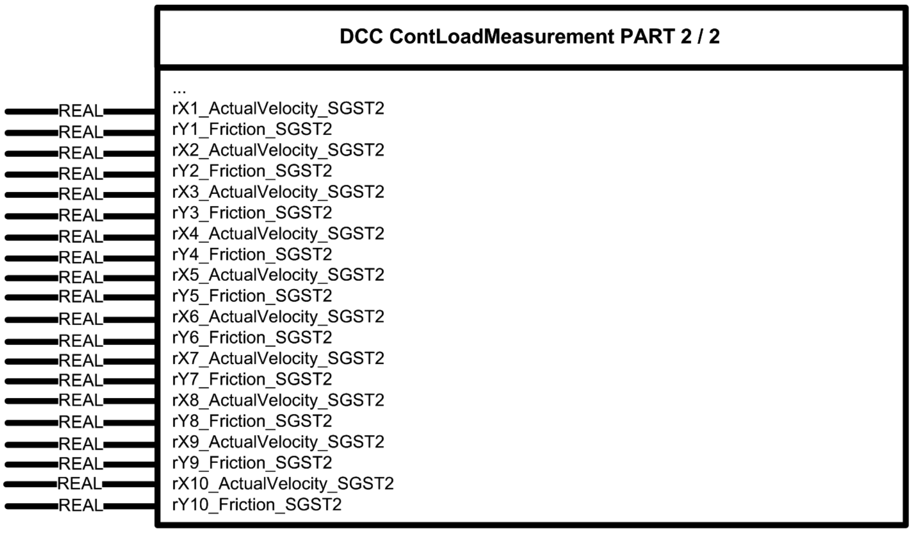
Connections

Name	Connection type	Default setting	Data type	Meaning
boSelectionHeavyDuty	IN	MCC	BOOL	Selects heavy-duty operation
rTimeAccHeavyDuty	IN	10000.0	REAL	Ramp-up time for heavy-duty operation [ms]
rTimeDecHeavyDuty	IN	10000.0	REAL	Ramp-down time for heavy-duty operation [ms]
boFieldWeakReached	IN	DCC	BOOL	Field weakening reached
boSelectionFieldWeak	IN	False	BOOL	Selects constant field weakening
rTimeAccFieldWeak	IN	10000.0	REAL	Ramp-up time in field-weakening operation [ms]
rTimeDecFieldWeak	IN	10000.0	REAL	Ramp-down time in field-weakening operation [ms]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity [mm/s]
rInAccelerationSetpoint	IN	MCC	REAL	Acceleration setpoint [mm/s^2] from the S7 controller
rInDecelerationSetpoint	IN	MCC	REAL	Deceleration setpoint [mm/s^2] from the S7 controller
rOutAccelerationSetpoint	OUT	MCC DCC	REAL	Effective acceleration for the drive [mm/s^2]
rOutDecelerationSetpoint	OUT	MCC DCC	REAL	Effective deceleration for the drive [mm/s^2]

5.2.4 DCC_ContLoadMeasurement

Symbol


DCC ContLoadMeasurement PART 1 / 2
REAL rActualVelocity
REAL rMaximumVelocity
REAL rTorqueNominal
BOOL boSelectionGearboxStage2
BOOL boActualVelocityZero
REAL rLimitToStartSawToothGen
REAL rLimitToStopSawToothGen
REAL rStartMeasurement
REAL rStopMeasurement
REAL rIntegrationTimeSawTooth
REAL rSetpointTorqueHG
REAL rSetpointTorqueCG
SDTIME rTorqueSmoothTime
SDTIME rHGTorqueSmoothTime
SDTIME rCGTorqueSmoothTime
SDTIME rSmoothTime_dn_dt
SDTIME rDifferentiationTimeCons
REAL rFact_AccTorqueRotMass1
REAL rFact_AccTorqueRotMass2
REAL rFact_AccTorque_Load1
REAL rFact_AccTorque_Load2
REAL rEfficiency_eta_drop
REAL rEfficiency_eta_lift
REAL rLoadHystLimitValueMon
REAL rLoadAverLimitValueMon
REAL rSpeedHystLimitValueMon
REAL rSpeedAverLimitValueMon
REAL rRopeHystLimitValueMon
REAL rRopeAverLimitValueMon
SDTIME rOnDelayTimeLimitMonitor
REAL rLoadNormFactor
REAL rX1_ActualVelocity_SGST1
REAL rY1_Friction_SGST1
REAL rX2_ActualVelocity_SGST1
REAL rY2_Friction_SGST1
REAL rX3_ActualVelocity_SGST1
REAL rY3_Friction_SGST1
REAL rX4_ActualVelocity_SGST1
REAL rY4_Friction_SGST1
REAL rX5_ActualVelocity_SGST1
REAL rY5_Friction_SGST1
REAL rX6_ActualVelocity_SGST1
REAL rY6_Friction_SGST1
REAL rX7_ActualVelocity_SGST1
REAL rY7_Friction_SGST1
REAL rX8_ActualVelocity_SGST1
REAL rY8_Friction_SGST1
REAL rX9_ActualVelocity_SGST1
REAL rY9_Friction_SGST1
REAL rX10_ActualVelocity_SGST1
REAL rY10_Friction_SGST1
REAL rActualRopeLength
REAL rMaximumRopeLength
REAL rCompensationRopeFactor
REAL rLoadMeasurementCell
SDTIME rLoadSmoothTime
REAL rMaximumLoad
BOOL boSwitchToLoadMeasCell
...
rLoad REAL
boGrapTouchDown BOOL
rCountsSawToothGenerator REAL
boStartMeasurment BOOL
boStopMeasurment BOOL
rHGSetTorqueInPercent REAL
rCGSetTorqueInPercent REAL
rSetpointTorquePercent REAL
rSavedCounts REAL
rSetTorqueMeanValue REAL
rSavedSetpointTorque REAL
rAcceleration REAL
rAccelerationMeanValue REAL
rSavedAccelerationSum REAL
rJBLoad REAL
rJBRot REAL
rFriction REAL
rMm_Minus_Friction REAL
rMm_Minus_Jbrot_Mf REAL
rLoad_without_eta REAL
rLoad_with_eta REAL
rLoad_eta_plus_Friction REAL
rLoad_eta_and_rope REAL


DCC ContLoadMeasurement PART 2 / 2
...
REAL rX1_ActualVelocity_SGST2
REAL rY1_Friction_SGST2
REAL rX2_ActualVelocity_SGST2
REAL rY2_Friction_SGST2
REAL rX3_ActualVelocity_SGST2
REAL rY3_Friction_SGST2
REAL rX4_ActualVelocity_SGST2
REAL rY4_Friction_SGST2
REAL rX5_ActualVelocity_SGST2
REAL rY5_Friction_SGST2
REAL rX6_ActualVelocity_SGST2
REAL rY6_Friction_SGST2
REAL rX7_ActualVelocity_SGST2
REAL rY7_Friction_SGST2
REAL rX8_ActualVelocity_SGST2
REAL rY8_Friction_SGST2
REAL rX9_ActualVelocity_SGST2
REAL rY9_Friction_SGST2
REAL rX10_ActualVelocity_SGST2
REAL rY10_Friction_SGST2

Brief description

The DCC block DCC_ContLoadMeasurement performs a continuous load measurement on the holding gear.

Mode of operation

This function is useful for every grab crane when no external load measuring cell is available.

The load measurement is performed continuously. The crane driver can always see the weight of the hoisted load. In the case of dredgers, for example, the crane driver cannot see the submerged grab bucket. With the continuous load measurement function, the driver can lift a fully loaded bucket out of the water every time. A "Grab touchdown" signal is also displayed.

The continuous load measurement function is implemented with a sawtooth generator which can be started and stopped via inputs "rLimittoStartSawToothGen" and "rLimittoStopSawToothGen" respectively. When the generator reaches the stop limit, it starts from the beginning again. You can view the result at output "rCountsSawToothGenerator". You can set the counting increments with "rIntegrationTimeSawTooth".

Mode of operation of the input "rIntegrationTimeSawTooth":
The block is linked into runtime group T2. Let us assume that T2 equals 9 ms. If you now specify 9 ms via the input "rIntegrationTimeSawTooth", then it will add 1 at each cycle. If you set "rIntegrationTimeSawTooth" to 90 ms, then it will add 0.1 at each cycle. This setting acts on inputs "rStartMeasurement" and "rStopMeasurement" as the count value of the sawtooth generator (rCountsSawToothGenerator) must be entered here as the start and stop count values.
In this way, you can store the desired sampling operations in the measuring window.

Example:
If you enter 2.0 in "rStartMeasurement" and 12.0 in "rStopMeasurement", either 10 cycles or 100 cycles will be available depending on the count value in T2 (rCountsSawToothGenerator).

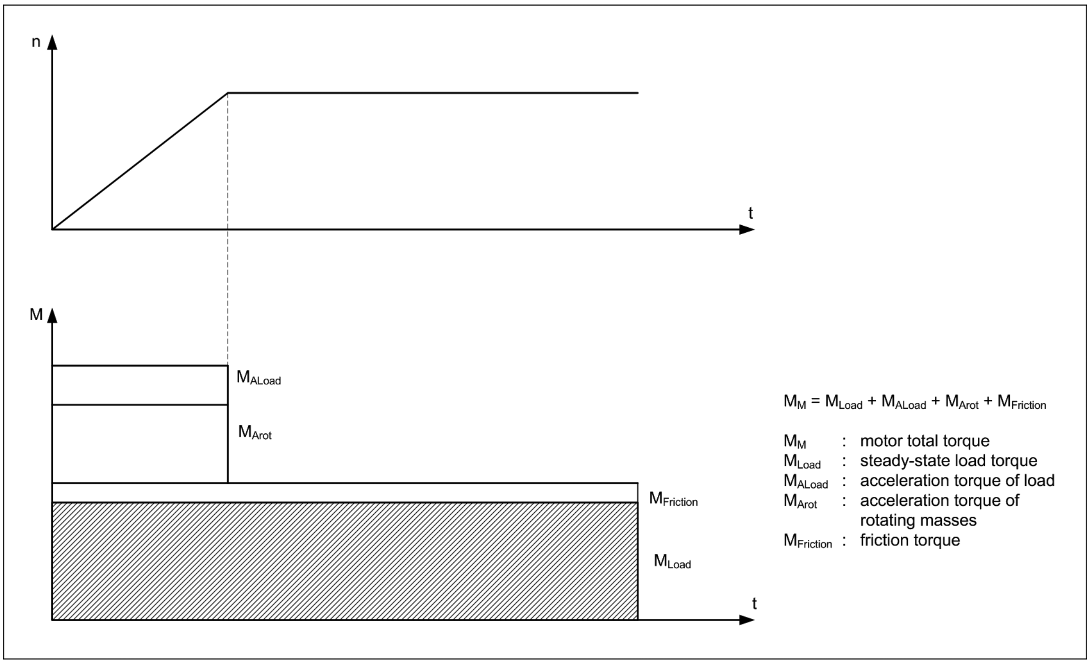


Figure 5-4 Speed torque characteristic

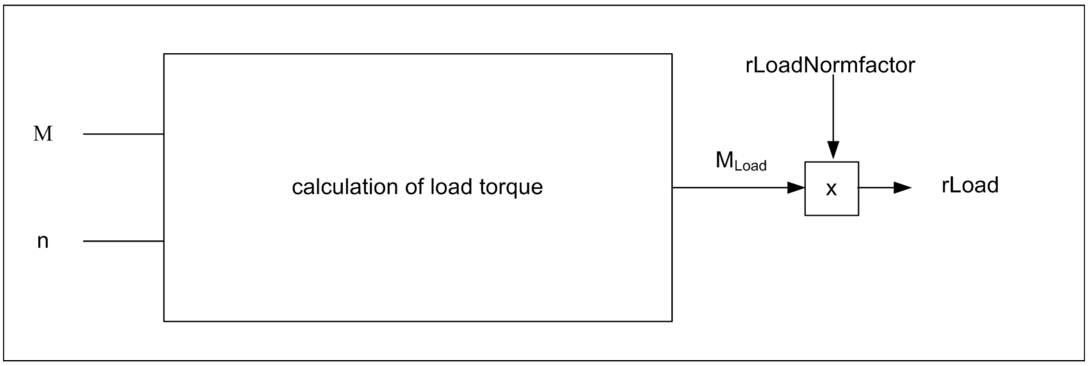


Figure 5-5 Principle of calculating the actual load value

NOTICE
The reference torque in [Nm] must be entered at input "rTorqueNominal" as a fixed value, or the maximum current must be interconnected with this input in the user program (refer to the SINAMICS parameter p2003). Enter the maximum velocity in [mm/s] at input "rMaximumVelocity" as fixed value, or interconnect the maximum velocity with this input in the user program. (refer to e.g. Technology Object: D4x5.HoldingGear_1.TypeOfAxis.MaxVelocity.maximum or function block FB_OperationMode output variable r64maximumVelocity). The block will not work correctly otherwise.

⚠ WARNING

Ensure that you enter the correct values for the polygon parameters! The input parameters "rX1_ActualVelocity_SGST1" to "rX10_ActualVelocity_SGST1" and "rX1_ActualVelocity_SGST2" to "rX10_ActualVelocity_SGST2" must be assigned in ascending sequence. Proper operation of the block cannot be guaranteed if you do not set the parameters correctly.

Connections

Name	Connection type	Default setting	Data type	Meaning
rActualVelocity	IN	MCC	REAL	Actual velocity [mm/s]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity (base velocity + field-weakening speed) [mm/s]
rTorqueNominal	IN	MCC	REAL	Reference torque [Nm]
boSelectionGearboxStage2	IN	False	BOOL	Selection gear stage 2
boActualVelocityZero	IN	MCC	BOOL	Standstill signal, zero speed signal
rLimitToStartSawToothGen	IN	10.0	REAL	Interval average [%] of current velocity for starting the sawtooth generator relative to maximum velocity
rLimitToStopSawToothGen	IN	30.0	REAL	Stop limit from which the sawtooth generator begins counting from the beginning again.
rStartMeasurement	IN	2.0	REAL	Start limit from which the sawtooth generator starts the load measurement. (boStartMeasurement)
rStopMeasurement	IN	30.0	REAL	Stop limit from which the sawtooth generator stops the load measurement. (boStopMeasurement)
rIntegrationTimeSawTooth	IN	1.0	REAL	Integration time for the sawtooth generator [ms]
rSetpointTorqueHG	IN	MCC	REAL	Torque setpoint holding gear [Nm]
rSetpointTorqueCG	IN	MCC	REAL	Torque setpoint closing gear [Nm]
rTorqueSmoothTime	IN	0	SDTIME	Smoothing time for calculated total torque (holding gear + closing gear) in [ms]
rHGTorqueSmoothTime	IN	0	SDTIME	Smoothing time for holding gear torque setpoint [ms]
rCGTorqueSmoothTime	IN	0	SDTIME	Smoothing time for closing gear torque setpoint [ms]
rSmoothTime_dn_dt	IN	20	SDTIME	Smoothing time [ms] for the velocity setpoint applied to the derivative action element.
rDifferentiationTimeCons	IN	3000	SDTIME	Derivative action time constant [ms] for the velocity setpoint applied to the derivative action element; normally the motor ramp-up time to n_{max}

Name	Connection type	Default setting	Data type	Meaning
rFact_AccTorqueRotMass1	IN	0.0	REAL	Evaluation for acceleration torque of rotating masses [%] in gear stage 1
rFact_AccTorqueRotMass2	IN	0.0	REAL	Evaluation for acceleration torque of rotating masses [%] in gear stage 2
rFact_AccTorque_Load1	IN	0.0	REAL	Evaluation for acceleration torque of load [%] in gear stage 1
rFact_AccTorque_Load2	IN	0.0	REAL	Evaluation for acceleration torque of load [%] in gear stage 2
rEfficiency_eta_drop	IN	100.0	REAL	Efficiency correction for lowering [%]
rEfficiency_eta_lift	IN	100.0	REAL	Efficiency correction for hoisting [%]
rLoadHystLimitValueMon	IN	0.5	REAL	Hysteresis for load monitoring [%] relative to "rMaximumLoad" for message "Grab touchdown"
rLoadAverLimitValueMon	IN	9.0	REAL	Average interval for load monitoring [%] relative to "rMaximumLoad" for message "Grab touchdown"
rSpeedHystLimitValueMon	IN	0.5	REAL	Hysteresis for current velocity monitoring [%] relative to maximum velocity
rSpeedAverLimitValueMon	IN	5.0	REAL	Average interval for current velocity monitoring [%] relative to maximum velocity
rRopeHystLimitValueMon	IN	0.5	REAL	Hysteresis for rope length monitoring [%] relative to "rMaximumRopeLength" for message "Grab touchdown"
rRopeAverLimitValueMon	IN	0.0	REAL	Average interval for rope length monitoring [%] relative to "rMaximumRopeLength" for message "Grab touchdown"
rOnDelayTimeLimitMonitor	IN	100	SDTIME	ON delay for "Grab touchdown" if condition is fulfilled [ms].
rLoadNormFactor	IN	1.0	REAL	Load conversion factor from current value to the physical unit tons
rX1_ActualVelocity_SGST1	IN	10.0	REAL	X1 (current velocity) [%] gear stage 1
rY1_Friction_SGST1	IN	2.0	REAL	Y1 friction [%] gear stage 1
rX2_ActualVelocity_SGST1	IN	20.0	REAL	X2 (current velocity) [%] gear stage 1
rY2_Friction_SGST1	IN	2.0	REAL	Y2 friction [%] gear stage 1
rX3_ActualVelocity_SGST1	IN	30.0	REAL	X3 (current velocity) [%] gear stage 1
rY3_Friction_SGST1	IN	2.0	REAL	Y3 friction [%] gear stage 1
rX4_ActualVelocity_SGST1	IN	40.0	REAL	X4 (current velocity) [%] gear stage 1
rY4_Friction_SGST1	IN	2.0	REAL	Y4 friction [%] gear stage 1
rX5_ActualVelocity_SGST1	IN	50.0	REAL	X5 (current velocity) [%] gear stage 1

Name	Connection type	Default setting	Data type	Meaning
rY5_Friction_SGST1	IN	2.0	REAL	Y5 friction [%] gear stage 1
rX6_ActualVelocity_SGST1	IN	60.0	REAL	X6 (current velocity) [%] gear stage 1
rY6_Friction_SGST1	IN	2.0	REAL	Y6 friction [%] gear stage 1
rX7_ActualVelocity_SGST1	IN	70.0	REAL	X7 (current velocity) [%] gear stage 1
rY7_Friction_SGST1	IN	2.5	REAL	Y7 friction [%] gear stage 1
rX8_ActualVelocity_SGST1	IN	80.0	REAL	X8 (current velocity) [%] gear stage 1
rY8_Friction_SGST1	IN	2.8	REAL	Y8 friction [%] gear stage 1
rX9_ActualVelocity_SGST1	IN	90.0	REAL	X9 (current velocity) [%] gear stage 1
rY9_Friction_SGST1	IN	3.2	REAL	Y9 friction [%] gear stage 1
rX10_ActualVelocit_SGST1	IN	100.0	REAL	X10 (current velocity) [%] gear stage 1
rY10_Friction_SGST1	IN	3.6	REAL	Y10 friction [%] gear stage 1
rActualRopeLength	IN	0.0	REAL	Actual rope length [mm]
rMaximumRopeLength	IN	1.0	REAL	Maximum possible rope length [mm]
rCompensationRopeFactor	IN	1.0	REAL	Compensation factor for rope weight
rLoadMeasurementCell	IN	0.0	REAL	Measured load value from load measuring cell
rLoadSmoothTime	IN	0	SDTIME	Smoothing time for actual load [ms]
rMaximumLoad	IN	1.0	REAL	Maximum possible load, e.g. [t]
boSwitchToLoadMeasCell	IN	False	BOOL	Selection of actual load value from load measuring cell
rX1_ActualVelocity_SGST2	IN	0.0	REAL	X1 (current velocity) [%] gear stage 2
rY1_Friction_SGST2	IN	0.0	REAL	Y1 friction [%] gear stage 2
rX2_ActualVelocity_SGST2	IN	0.0	REAL	X2 (current velocity) [%] gear stage 2
rY2_Friction_SGST2	IN	0.0	REAL	Y2 friction [%] gear stage 2
rX3_ActualVelocity_SGST2	IN	0.0	REAL	X3 (current velocity) [%] gear stage 2
rY3_Friction_SGST2	IN	0.0	REAL	Y3 friction [%] gear stage 2
rX4_ActualVelocity_SGST2	IN	0.0	REAL	X4 (current velocity) [%] gear stage 2

Name	Connection type	Default setting	Data type	Meaning
rY4_Friction_SGST2	IN	0.0	REAL	Y4 friction [%] gear stage 2
rX5_ActualVelocity_SGST2	IN	0.0	REAL	X5 (current velocity) [%] gear stage 2
rY5_Friction_SGST2	IN	0.0	REAL	Y5 friction [%] gear stage 2
rX6_ActualVelocity_SGST2	IN	0.0	REAL	X6 (current velocity) [%] gear stage 2
rY6_Friction_SGST2	IN	0.0	REAL	Y6 friction [%] gear stage 2
rX7_ActualVelocity_SGST2	IN	0.0	REAL	X7 (current velocity) [%] gear stage 2
rY7_Friction_SGST2	IN	0.0	REAL	Y7 friction [%] gear stage 2
rX8_ActualVelocity_SGST2	IN	0.0	REAL	X8 (current velocity) [%] gear stage 2
rY8_Friction_SGST2	IN	0.0	REAL	Y8 friction [%] gear stage 2
rX9_ActualVelocity_SGST2	IN	0.0	REAL	X9 (current velocity) [%] gear stage 2
rY9_Friction_SGST2	IN	0.0	REAL	Y9 friction [%] gear stage 2
rX10_ActualVelocit_SGST2	IN	0.0	REAL	X10 (current velocity) [%] gear stage 2
rY10_Friction_SGST2	IN	0.0	REAL	Y10 friction [%] gear stage 2
rLoad	OUT	MCC	REAL	Evaluated actual load value [e.g. in tons], depending on rLoadNormfactor
boGrapTouchDown	OUT	MCC	BOOL	Grab touchdown
rCountsSawToothGenerator	OUT	0.0	REAL	Current counter reading of sawtooth generator
boStartMeasurement	OUT	False	BOOL	Measuring window start signal
boStopMeasurement	OUT	False	BOOL	Measuring window stop signal
rHGSetTorqueInPercent	OUT	0.0	REAL	Torque setpoint holding gear [%]
rCGSetTorqueInPercent	OUT	0.0	REAL	Torque setpoint closing gear [%]
rSetpointTorquePercent	OUT	0.0	REAL	Calculated total torque setpoint [%] (holding gear + closing gear) / 2
rSavedCounts	OUT	0.0	REAL	Number of sampling operations in the measuring window
rSetTorqueMeanValue	OUT	0.0	REAL	Mean value of torque setpoint [%]
rSavedSetpointTorque	OUT	0.0	REAL	Torque setpoint in measuring window [%]
rAcceleration	OUT	0.0	REAL	Current acceleration [%]
rAccelerationMeanValue	OUT	0.0	REAL	Mean value of acceleration [%]
rSavedAccelerationSum	OUT	0.0	REAL	Acceleration total [%]
rJBLoad	OUT	0.0	REAL	Acceleration torque of load [%] relative to rTorqueNominal

Name	Connection type	Default setting	Data type	Meaning
rJBRot	OUT	0.0	REAL	Acceleration torque of rotating masses [%] relative to rTorqueNominal
rFriction	OUT	0.0	REAL	Friction [%] dependent on current velocity
rMm_Minus_Friction	OUT	0.0	REAL	M_{tot} – $M_{friction}$ (total torque – frictional torque) [%] relative to rTorqueNominal
rMm_minus_Jbrot_Mf	OUT	0.0	REAL	Total torque – frictional torque – acceleration torque of rotating masses [%] relative to rTorqueNominal
rLoad_without_eta	OUT	0.0	REAL	Actual load without efficiency correction [%]
rLoad_with_eta	OUT	0.0	REAL	Actual load with efficiency correction [%]
rLoad_eta_plus_Friction	OUT	0.0	REAL	Actual load with efficiency correction plus friction torque [%]
rLoad_eta_and_rope	OUT	0.0	REAL	Actual load with efficiency correction and rope weight correction [%]

Note

The following applies to all friction values:
The input value rXx (corresponding to the respective current velocity in percent relative to "rMaximumVelocity") belongs to the corresponding friction value (in percent) rYx. Linear interpolation is performed between the points. The friction value is included in the load value calculation.

Note

Typical setting values:

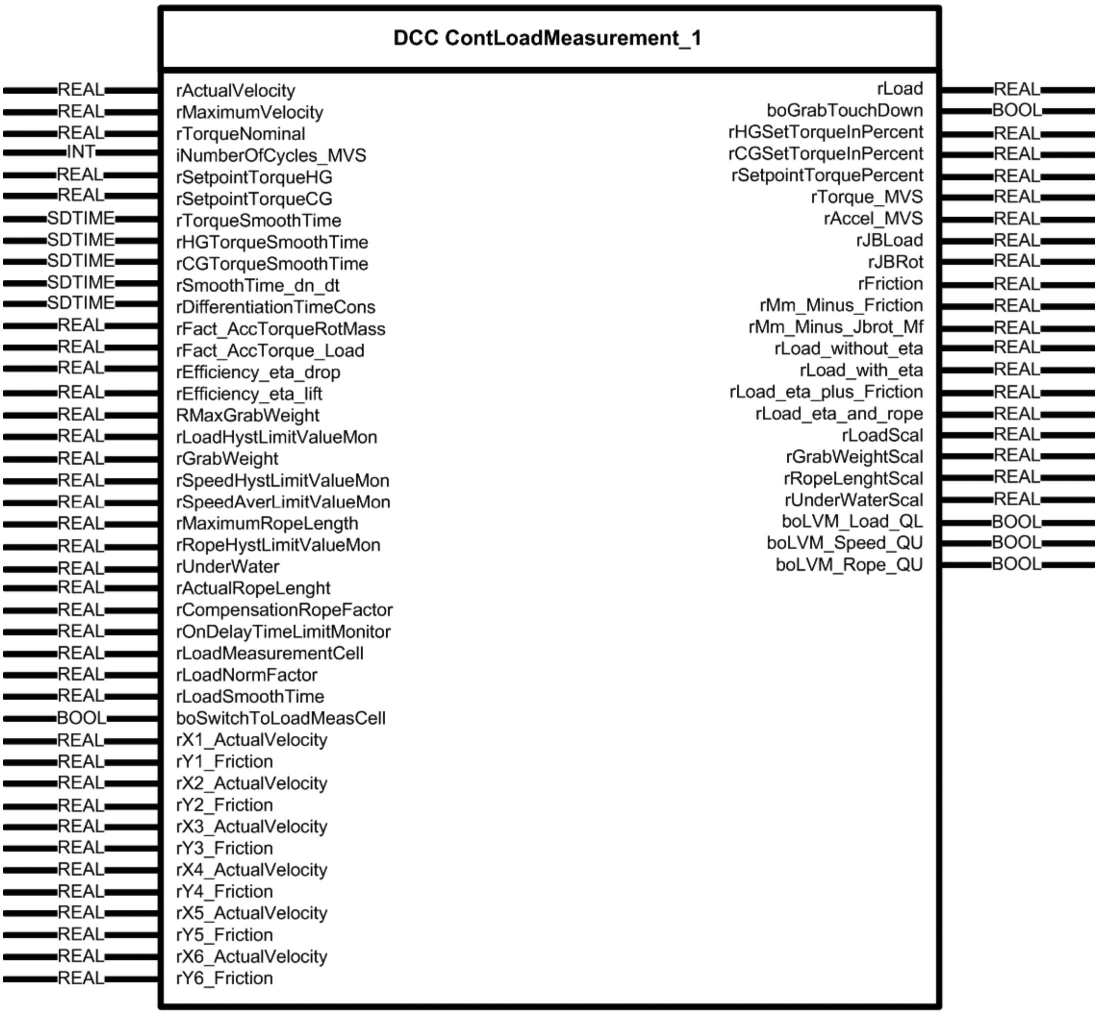
- rDifferentiationTimeCons: Ramp-up time holding gear, e.g. 3000 ms
- rFact_AccTorqueRotMass1: Approx. 20% - 30%
- rFact_AccTorque_Load1: Approx. 5% - 10%
- rEfficiency_eta_drop: Approx. 110%
- rEfficiency_eta_lift: Approx. 90% - 100%
- rLimitToStartSawToothGen: Approx. 10% (to overcome the breakaway torque)
- Friction: Approx. 2% - 4%

Note

For a detailed description of this block, refer to Chapter Continuous load measurement (CLM) (Page 434)

5.2.5 DCC_ContLoadMeasurement_1

Symbol



Brief description

The DCC block DCC_ContLoadMeasurement_1 has a 6-point polygon.

The DCC block DCC_ContLoadMeasurement_1 performs a continuous load measurement on the holding gear.

Mode of operation

The mode of operation of this block corresponds to that of the DCC_ContLoadMeasurement block (refer to Chapter DCC_ContLoadMeasurement (Page 80)). The existing 10-point polygon for the travel characteristic is however simplified to six points and the second gearbox has been eliminated.

The DCC_ContLoadMeasurement_1 block contains a sliding mean value generator (MVS; "Sliding-type mean value generator") to determine the torque and acceleration. The alternative DCC_ContLoadMeasurement block uses instead of this, a sawtooth generator.

The limit values for load and depth can be provided for the SIMATIC controller or for the visualization.

NOTICE

The reference torque in [Nm] must be entered at input "rTorqueNominal" as a fixed value, or the maximum current must be interconnected with this input in the user program (refer to the SINAMICS parameter p2003).

The maximum velocity must be entered in [mm/s] at input "rMaximumVelocity" as fixed value, or the maximum velocity interconnected with this input in the user program. (refer to e.g. Technology Object: D4x5.HoldingGear_1.TypeOfAxis.MaxVelocity.maximum or function block FB_OperationMode output variable r64maximumVelocity).
The block will not work correctly otherwise.

⚠ WARNING

Ensure that you enter the correct values for the polygon parameters! The input parameters "rX1_ActualVelocity" to "rX6_ActualVelocity" must be assigned in ascending sequence. Proper operation of the block cannot be guaranteed if you do not set the parameters correctly.

Connections

Name	Connection type	Default setting	Data type	Meaning
rActualVelocity	IN	MCC	REAL	Actual velocity [mm/s]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity (base velocity + field-weakening speed) [mm/s]
rTorqueNominal	IN	MCC	REAL	Reference torque [Nm]
iNumberOfCycles_MVS	IN	10	INT	Number of computation cycles, over which a sliding mean value generation (MVS) is realized for acceleration and torque
rSetpointTorqueHG	IN	MCC	REAL	Torque setpoint holding gear [Nm]
rSetpointTorqueCG	IN	MCC	REAL	Torque setpoint closing gear [Nm]
rTorqueSmoothTime	IN	0.0	SDTIME	Smoothing time for calculated total torque (holding gear + closing gear) in [ms]
rHGTorqueSmoothTime	IN	0.0	SDTIME	Smoothing time for holding gear torque setpoint [ms]
rCGTorqueSmoothTime	IN	0.0	SDTIME	Smoothing time for closing gear torque setpoint [ms]
rSmoothTime_dn_dt	IN	20.0	SDTIME	Smoothing time for the velocity setpoint applied to the derivative action element [ms]
rDifferentiationTimeCons	IN	3000.0	SDTIME	Derivative action time constant for the velocity setpoint applied to the derivative action element; normally the motor ramp-up time to n_{max}[ms]
rFact_AccTorqueRotMass	IN	0.0	REAL	Evaluation for accelerating torque of the rotating masses [%]
rFact_AccTorque_Load	IN	0.0	REAL	Evaluation for the load acceleration torque [%]
rEfficiency_eta_drop	IN	100.0	REAL	Efficiency correction for lowering [%]
rEfficiency_eta_lift	IN	100.0	REAL	Efficiency correction for hoisting [%]
rMaxGrabWeight	IN	35.0	REAL	Maximum possible load, e.g. [t]
rLoadHystLimitValueMon	IN	0.5	REAL	Hysteresis for load monitoring [%] relative to "rMaxGrabWeight" for the message "Grab touchdown"
rGrabWeight	IN	0.0	REAL	Average interval for load monitoring relative to "rMaxGrabWeight" for message "Grab touchdown" [%]
rSpeedHystLimitValueMon	IN	0.5	REAL	Hysteresis for current velocity monitoring relative to the maximum velocity [%]
rSpeedAverLimitValueMon	IN	5.0	REAL	Average interval for current velocity monitoring relative to the maximum velocity [%]
rMaximumRopeLength	IN	80000.0	REAL	Maximum possible rope length [mm]
rRopeHystLimitValueMon	IN	0.5	REAL	Hysteresis for rope length monitoring relative to "rMaximumRopeLength" for message "Grab touchdown" [%]

Name	Connection type	Default setting	Data type	Meaning
rUnderWater	IN	15000.0	REAL	Average interval for rope length monitoring relative to "rMaximumRopeLength" for message "Grab touchdown" [%]
rActualRopeLength	IN	MCC	REAL	Actual rope length [mm]
rCompensationRopeFactor	IN	0.0	REAL	Compensation factor for rope weight
rOnDelayTimeLimitMonitor	IN	100.0	SDTIME	ON delay for "Grab touchdown" if condition is fulfilled [ms]
rLoadMeasurementCell	IN	0.0	REAL	Measured load value from load measuring cell
rLoadNormFactor	IN	1.0	REAL	Load conversion factor from current value to the physical unit tons
rLoadSmoothTime	IN	0.0	SDTIME	Smoothing time for actual load [ms]
boSwitchToLoadMeasCell	IN	False	BOOL	Selection of actual load value from load measuring cell
rX1_ActualVelocity	IN	10.0	REAL	X1 (actual velocity) [%]
rY1_Friction	IN	2.0	REAL	Y1 friction [%]
rX2_ActualVelocity	IN	20.0	REAL	X2 (actual velocity) [%]
rY2_Friction	IN	2.0	REAL	Y2 friction [%]
rX3_ActualVelocity	IN	30.0	REAL	X3 (actual velocity) [%]
rY3_Friction	IN	2.0	REAL	Y3 friction [%]
rX4_ActualVelocity	IN	40.0	REAL	X4 (actual velocity) [%]
rY4_Friction	IN	2.0	REAL	Y4 friction [%]
rX5_ActualVelocity	IN	50.0	REAL	X5 (actual velocity) [%]
rY5_Friction	IN	2.0	REAL	Y5 friction [%]
rX6_ActualVelocity	IN	60.0	REAL	X6 (actual velocity) [%]
rY6_Friction	IN	2.0	REAL	Y6 friction [%]
rLoad	OUT	MCC	REAL	Evaluated actual load value [e.g. in tons], depending on rLoadNormfactor
boGrabTouchDown	OUT	MCC	BOOL	Grab touchdown
rHGSetTorqueInPercent	OUT	0.0	REAL	Torque setpoint holding gear [%]
rCGSetTorqueInPercent	OUT	0.0	REAL	Torque setpoint closing gear [%]
rSetpointTorquePercent	OUT	0.0	REAL	Calculated total torque setpoint [%] (holding gear + closing gear) / 2
rTorque_MVS	OUT	0.0	REAL	Mean value of torque setpoint [%]
rAccel_MVS	OUT	0.0	REAL	Mean value of acceleration [%]
rJBLoad	OUT	0.0	REAL	Acceleration torque of load [%] relative to rTorqueNominal
rJBRot	OUT	0.0	REAL	Acceleration torque of rotating masses [%] relative to rTorqueNominal
rFriction	OUT	0.0	REAL	Friction [%] dependent on current velocity
rMm_Minus_Friction	OUT	0.0	REAL	M_{tot} – $M_{friction}$ (total torque – frictional torque) [%] relative to rTorqueNominal

Name	Connection type	Default setting	Data type	Meaning
rMm_minus_Jbrot_Mf	OUT	0.0	REAL	Total torque – frictional torque – acceleration torque of rotating masses [%] relative to rTorqueNominal
rLoad_without_eta	OUT	0.0	REAL	Actual load without efficiency correction [%]
rLoad_with_eta	OUT	0.0	REAL	Actual load with efficiency correction [%]
rLoad_eta_plus_Friction	OUT	0.0	REAL	Actual load with efficiency correction plus friction torque [%]
rLoad_eta_and_rope	OUT	0.0	REAL	Actual load with efficiency correction and rope weight correction [%]
rLoadScal	OUT	0.0	REAL	Calculated load referred to the maximum grab weight [%]
rGrabWeightScal	OUT	0.0	REAL	Limit value, load for message "Grab touchdown" referred to the maximum grab weight [%]
rRopeLengthScal	OUT	0.0	REAL	Actual rope length referred to the maximum rope length [%]
rUnderWaterScal	OUT	0.0	REAL	Limit value, rope length for message "Grab touchdown" referred to the maximum rope length [%]
boLVM_Load_QL	OUT	False	BOOL	Limit monitor: Load actual value < load limit value (rGrabWeight); important for the message "Grab touchdown"
boLVM_Speed_QU	OUT	False	BOOL	Limit monitor: Velocity actual value > velocity limit value (rSpeedAverLimitValueMon); important for the message "Grab touchdown"
boLVM_Rope_QU	OUT	False	BOOL	Limit monitor: Rope length > rope length limit value (rUnderWater); important for the message "Grab touchdown"

Note

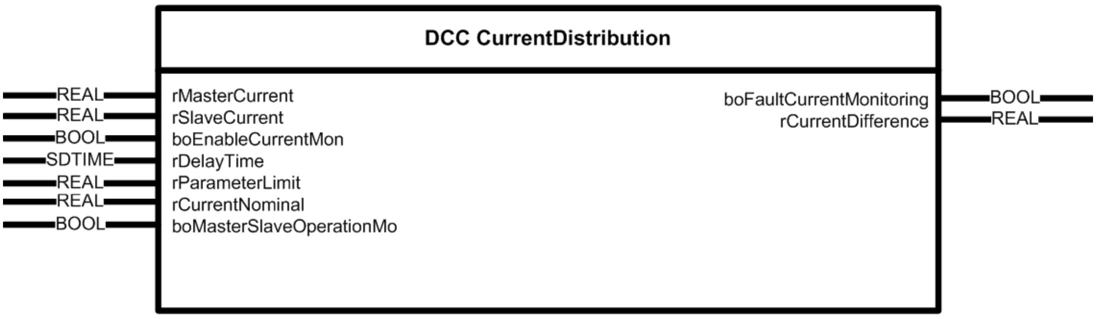
The following applies to all friction values:
The input values "rX1_ActualVelocity" to "rX6_ActualVelocity" (corresponding to the respective actual velocity as a percentage relative to "rMaximumVelocity") belong to the corresponding friction value (as a percentage) "rY1_Friction" to "rY6_Friction". Linear interpolation is performed between the points. The friction value is included in the load value calculation.

Note

For a detailed description of this block, refer to Chapter Continuous load measurement (CLM) (Page 434)

5.2.6 DCC_CurrentDistribution

Symbol



Brief description

The current of the master and slave is monitored using the DCC block DCC_CurrentDistribution. An error message is generated if any deviation is exceeded.

Mode of operation

This function can be used for master-slave operation or synchronous operation. The block checks that the total current of both drives is being distributed evenly. However, this is not always the case for crane operation.

In synchronous operation, the current setpoint is monitored for the two drives. In master-slave operation, it is the current actual value that is monitored.

Bit 3 "boFaultCurrentMonitoring" is set in application status word 1 if the difference between the setpoint currents of the two drives exceeds the values configured for deviation and time.

An example of synchronous operation is illustrated below.

CAUTION
Enter the maximum current in [A] at input "rCurrentNormalizeFactor" as fixed value, or interconnect the maximum current with this input in the user program (refer to the SINAMICS S List Manual, parameter p2002). The block does not work correctly otherwise.

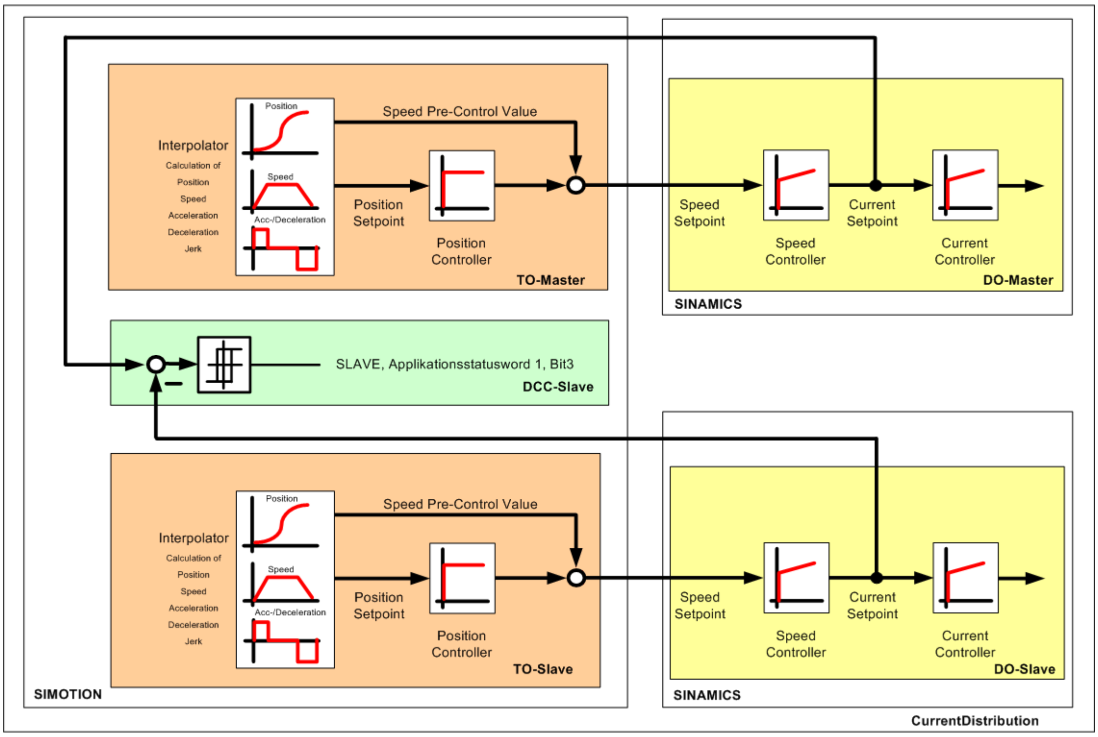


Figure 5-6 Principle of current distribution monitoring based on the example of synchronous operation

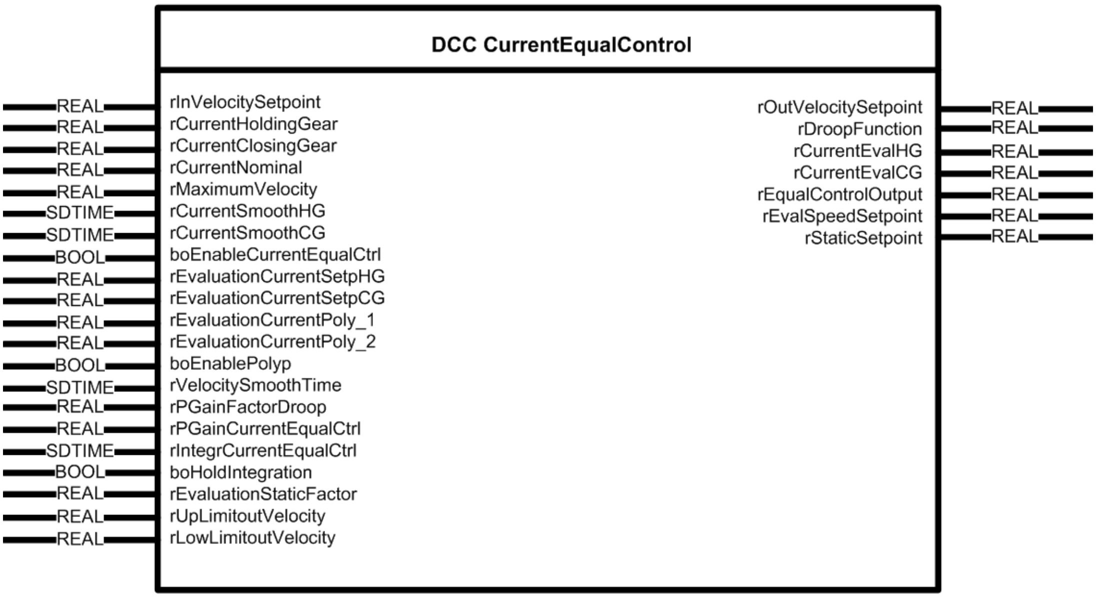
Connections

Name	Connection type	Default setting	Data type	Meaning
rMasterCurrent	IN	MCC	REAL	Master current setpoint [A]
rSlaveCurrent	IN	MCC	REAL	Slave current setpoint [A]
boEnableCurrentMon	IN	True	BOOL	Enables the current comparison monitoring
rDelayTime	IN	1000	SDTIME	Deceleration time [ms]; if a deviation occurs between the master and slave current, the output "boFaultCurrentMonitoring" is only set if the deceleration time has expired.
rParameterLimit	IN	20.0	REAL	Deviation threshold [%; relative to rCurrentNominal]. The master and the slave current are subtracted and then compared with the deviation threshold.
rCurrentNominal	IN	MCC	REAL	Reference current [A]
boMasterSlaveOperationMo	IN	MCC	BOOL	Master-slave operating mode or synchronous operation is active. Current distribution is only monitored in master-slave operation or in synchronous operation.

Name	Connection type	Default setting	Data type	Meaning
boFaultCurrentMonitoring	OUT	MCC	BOOL	The master and the slave current deviate from one another too significantly.
rCurrentDifference	OUT	0.0	REAL	Actual result of the subtraction between master and slave [A]

5.2.7 DCC_CurrentEqualControl

Symbol



Short description

The DCC block DCC_CurrentEqualControl is required by the closing gear to distribute the necessary hoisting power optimally between holding gear and closing gear.

Mode of operation

A basic distinction is made between two current compensatory controller modes, i.e. "orange-peel bucket mode" and "normal mode". The current compensatory controller (load compensation controller) also features a speed droop feedback. The value of the droop can be selected with "rPGainFactorDroop" (Kp of the P controller) and "rEvaluationStaticFactor" and monitored with "rDroopFunction" or "rStaticSetpoint".

This function is needed on every grab crane.

When the load is raised or lowered with the grab bucket closed, the tension levels in the holding and closing ropes should be approximately the same. This means that the hoisting power is optimally distributed between the two hoisting motors. This requirement is fulfilled by a compensatory control between the two drives.

The current setpoints generated by the speed controllers of the hoisting gear (holding gear and closing gear) are compared and the difference between them is applied to the closing gear speed setpoint via a controller; see the figure below.

The highest rope torque occurs when the grab closes almost completely. The positioning data is falsified by the unavoidable rope elongation, i.e. the closing process is automatically terminated when the grab is not quite fully closed.

This falsification is also corrected by enabling the current compensatory control in response to the message "Grab closed".

In the case of multi-functional grab cranes (used for container, spreader beam, and hook operation), it may be necessary to activate the current compensatory controller depending on the application and mechanical construction of the crane.

Orange-peel bucket mode

In this operating mode, the grabs generally require a higher tension level in the closing rope than in the holding rope. A supplementary circuit in the current compensatory control allows the current to be divided asymmetrically in the required ratio.

This capability is required, for example, when compacted scrap, tree trunks and other bulky loads are handled. When loads of this type are handled by pronged grabs (orange-peel buckets), the grabs are only partially closed and the material is held only by the prong tips.

To prevent the load from slipping out of the grab, the highest possible closing force must be achieved. For this purpose, the current compensatory controller distributes the load asymmetrically.

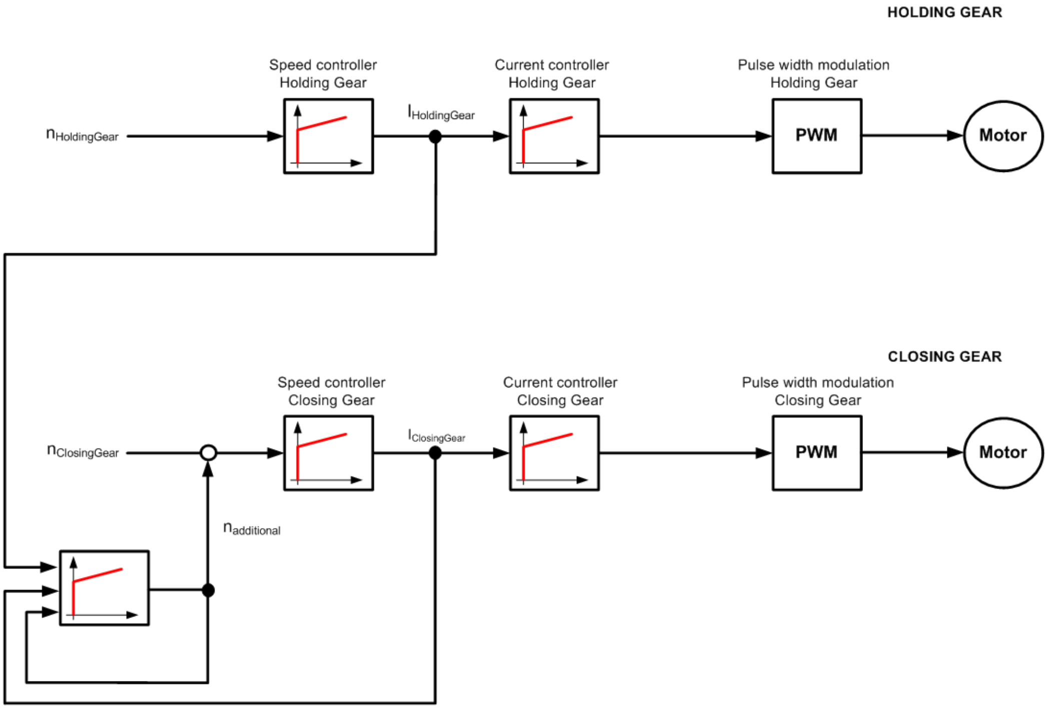

HOLDING GEAR
Speed controller Holding Gear
Current controller Holding Gear
Pulse width modulation Holding Gear
$n_{HoldingGear}$
$I_{HoldingGear}$
PWM
Motor
CLOSING GEAR
Speed controller Closing Gear
Current controller Closing Gear
Pulse width modulation Closing Gear
$n_{ClosingGear}$
$I_{ClosingGear}$
$n_{additional}$
PWM
Motor

Figure 5-7 Circuit diagram of the current compensatory controller

Note

The current compensatory controller is a PI controller. At the beginning, the kp gain factor (rPGainCurrentEqualCtrl) should be a low value and the integral time (rIntegrCurrentEqualCtrl) a high value, see default setting. The fundamental principles of closed-loop control and the properties of a PI controller must be taken into account.

NOTICE

Both "Start Pulse" and "Current compensatory control" functions are used for Closing Gear. As both blocks DCC_StartPulse and DCC_CurrentEqualControl act on the shared supplementary speed setpoint in SINAMICS, the output of block "DCC_StartPulse" must be connected to input "rInVelocitySetpoint".

For technical reasons, only one of the two "StartPulse" or "CurrentEqualControl" functions may be active at any one time.

A setpoint applied at input "rInVelocitySetpoint" is switched through directly to output "rOutVelocitySetpoint" as long as input "boEnableCurrentEqualCtrl" is not set. In this case, the current compensatory controller is disabled and does not specify a setpoint.

A setpoint applied at input "rInVelocitySetpoint" is not switched through directly to output "rOutVelocitySetpoint" as long as input "boEnableCurrentEqualCtrl" is set. In this case, the current compensatory controller is enabled and does specify a setpoint.

NOTICE
Enter the maximum current in [A] at input "rCurrentNominal" as fixed value, or interconnect the maximum current with this input in the user program (refer to the SINAMICS parameter p2002). Enter the maximum velocity in [mm/s] at input "rMaximumVelocity" as fixed value, or interconnect the maximum velocity with this input in the user program. (refer to e.g. Technology Object: D4x5.ClosingGear_1.TypeOfAxis.MaxVelocity.maximum or function block FB_OperationMode output variable r64maximumVelocity). The block will not work correctly otherwise.

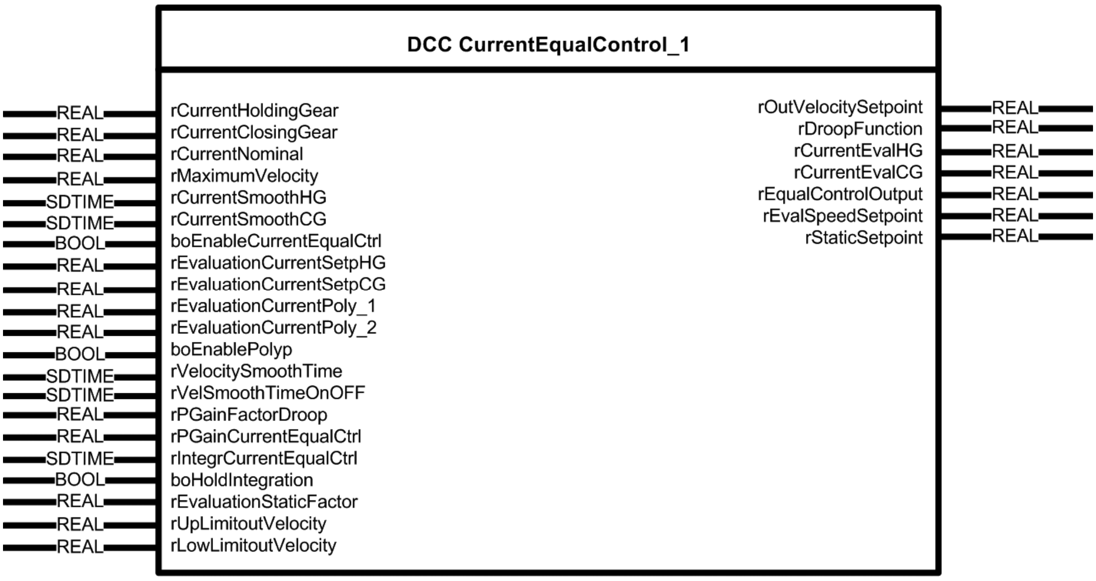
Connections

Name	Connection type	Default setting	Data type	Meaning
rInVelocitySetpoint	IN	DCC	REAL	Setpoint velocity [mm/s]
rCurrentHoldingGear	IN	MCC	REAL	Holding Gear setpoint current [A]
rCurrentClosingGear	IN	MCC	REAL	Closing gear setpoint current [A]
rCurrentNominal	IN	MCC	REAL	Reference current [A]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity (base velocity + field-weakening speed) [mm/s]
rCurrentSmoothHG	IN	10	SDTIME	Smoothing time for the holding gear current [ms]
rCurrentSmoothCG	IN	10	SDTIME	Smoothing time for the closing gear current [ms]
boEnableCurrentEqualCtrl	IN	MCC	BOOL	Enable current compensatory controller FALSE: rOutVelocitySetpoint = rInVelocitySetpoint TRUE: rOutVelocitySetpoint = velocity of current compensatory controller
rEvaluationCurrentSetpHG	IN	100.0	REAL	Current setpoint evaluation [%] of holding gear for current compensatory controller
rEvaluationCurrentSetpCG	IN	100.0	REAL	Current setpoint evaluation [%] of closing gear for current compensatory controller
rEvaluationCurrentPoly_1	IN	50.0	REAL	Current setpoint evaluation [%] of holding gear for current compensatory controller in orange-peel bucket mode for torque direction 1
rEvaluationCurrentPoly_2	IN	50.0	REAL	Current setpoint evaluation [%] of closing gear for current compensatory controller in orange-peel bucket mode for torque direction 2
boEnablePolyp	IN	MCC	BOOL	Enable orange-peel bucket mode
rVelocitySmoothTime	IN	20	SDTIME	Smoothing time for the setpoint velocity [ms]

Name	Connection type	Default setting	Data type	Meaning
rPGainFactorDroop	IN	0.0	REAL	P gain for P droop current compensatory controller [%]
rPGainCurrentEqualCtrl	IN	1.0	REAL	P gain current compensatory controller [%]
rIntegrCurrentEqualCtrl	IN	500	SDTIME	Integral time current compensatory controller [ms]
boHoldIntegration	IN	True	BOOL	Hold the integrator time of the current compensatory controller
rEvaluationStaticFactor	IN	0.0	REAL	Evaluation factor for P droop feedback on current compensatory controller [%]
rUpLimitoutVelocity	IN	30.0	REAL	Upper speed limiting value [%] for the P controller output. The value must lie between 0 and 100%.
rLowLimitoutVelocity	IN	-30.0	REAL	Lower speed limiting value [%] for the P controller output. The value must lie between 0 and -100%.
rOutVelocitySetpoint	OUT	MCC	REAL	Setpoint velocity of the current compensatory controller [mm/s]
rDroopFunction	OUT	0.0	REAL	Effective P droop [%]
rCurrentEvalHG	OUT	0.0	REAL	Holding gear effective current setpoint [%]
rCurrentEvalCG	OUT	0.0	REAL	Closing gear effective current setpoint [%]
rEqualControlOutput	OUT	0.0	REAL	Effective output value from current compensatory controller [%]
rEvalSpeedSetpoint	OUT	0.0	REAL	Evaluated setpoint velocity [mm/s]
rStaticSetpoint	OUT	0.0	REAL	Evaluated droop feedback [%]

5.2.8 DCC_CurrentEqualControl_1

Symbol


DCC CurrentEqualControl_1
REAL rCurrentHoldingGear
REAL rCurrentClosingGear
REAL rCurrentNominal
REAL rMaximumVelocity
SDTIME rCurrentSmoothHG
SDTIME rCurrentSmoothCG
BOOL boEnableCurrentEqualCtrl
REAL rEvaluationCurrentSetpHG
REAL rEvaluationCurrentSetpCG
REAL rEvaluationCurrentPoly_1
REAL rEvaluationCurrentPoly_2
BOOL boEnablePolyp
SDTIME rVelocitySmoothTime
SDTIME rVelSmoothTimeOnOFF
REAL rPGainFactorDroop
REAL rPGainCurrentEqualCtrl
SDTIME rIntegrCurrentEqualCtrl
BOOL boHoldIntegration
REAL rEvaluationStaticFactor
REAL rUpLimitoutVelocity
REAL rLowLimitoutVelocity
rOutVelocitySetpoint REAL
rDroopFunction REAL
rCurrentEvalHG REAL
rCurrentEvalCG REAL
rEqualControlOutput REAL
rEvalSpeedSetpoint REAL
rStaticSetpoint REAL

Brief description

This DCC block is functionally identical with the DCC_CurrentEqualControl block, only that input "rInVelocitySetpoint" has been removed. This input was originally added to switch through from the setpoint channel of the start pulse. However, in the future, the setpoint channel of the start pulse will be externally connected in parallel; refer to the standard application software. In addition, the "rVelocitySmoothTimeOnOff" input has been added.

Connections

Name	Connection type	Default setting	Data type	Meaning
rInVelocitySetpoint	IN	DCC	REAL	Setpoint velocity [mm/s]
rCurrentHoldingGear	IN	MCC	REAL	Holding Gear setpoint current [A]
rCurrentClosingGear	IN	MCC	REAL	Closing gear setpoint current [A]
rCurrentNominal	IN	MCC	REAL	Reference current [A]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity (base velocity + field-weakening speed) [mm/s]
rCurrentSmoothHG	IN	10	SDTIME	Smoothing time for the holding gear current [ms]
rCurrentSmoothCG	IN	10	SDTIME	Smoothing time for the closing gear current [ms]

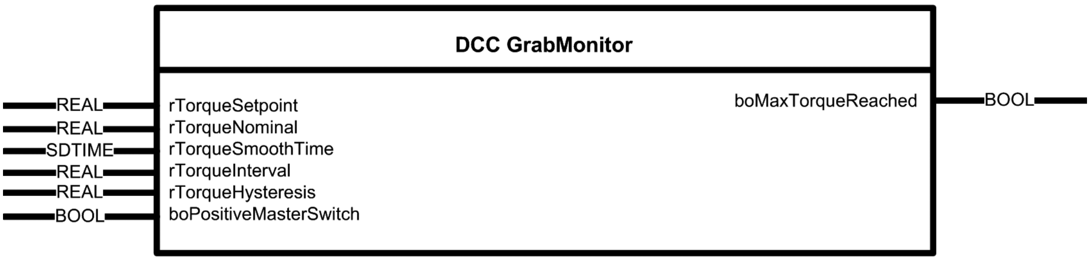
Name	Connection type	Default setting	Data type	Meaning
boEnableCurrentEqualCtrl	IN	MCC	BOOL	Enable current compensatory controller FALSE: rOutVelocitySetpoint = rInVelocitySetpoint TRUE: rOutVelocitySetpoint = velocity of current compensatory controller
rEvaluationCurrentSetpHG	IN	100.0	REAL	Current setpoint evaluation [%] of holding gear for current compensatory controller
rEvaluationCurrentSetpCG	IN	100.0	REAL	Current setpoint evaluation [%] of closing gear for current compensatory controller
rEvaluationCurrentPoly_1	IN	50.0	REAL	Current setpoint evaluation [%] of holding gear for current compensatory controller in orange-peel bucket mode for torque direction 1
rEvaluationCurrentPoly_2	IN	50.0	REAL	Current setpoint evaluation [%] of closing gear for current compensatory controller in orange-peel bucket mode for torque direction 2
boEnablePolyp	IN	MCC	BOOL	Enable orange-peel bucket mode
rVelocitySmoothTime	IN	20	SDTIME	Smoothing time for the setpoint velocity [ms]
rVelSmoothTimeOnOff	IN	500	SDTIME	Smoothing time during on/off [ms]
rPGainFactorDroop	IN	0.0	REAL	P gain for P droop current compensatory controller [%]
rPGainCurrentEqualCtrl	IN	1.0	REAL	P gain current compensatory controller [%]
rIntegrCurrentEqualCtrl	IN	500	SDTIME	Integral time current compensatory controller [ms]
boHoldIntegration	IN	True	BOOL	Hold the integrator time of the current compensatory controller
rEvaluationStaticFactor	IN	0.0	REAL	Evaluation factor for P droop feedback on current compensatory controller [%]
rUpLimitoutVelocity	IN	30.0	REAL	Upper speed limiting value [%] for the P controller output. The value must lie between 0 and 100%.
rLowLimitoutVelocity	IN	-30.0	REAL	Lower speed limiting value [%] for the P controller output. The value must lie between 0 and -100%.
rOutVelocitySetpoint	OUT	MCC	REAL	Setpoint velocity of the current compensatory controller [mm/s]
rDroopFunction	OUT	0.0	REAL	Effective P droop [%]
rCurrentEvalHG	OUT	0.0	REAL	Holding gear effective current setpoint [%]
rCurrentEvalCG	OUT	0.0	REAL	Closing gear effective current setpoint [%]
rEqualControlOutput	OUT	0.0	REAL	Effective output value from current compensatory controller [%]
rEvalSpeedSetpoint	OUT	0.0	REAL	Evaluated setpoint velocity [mm/s]
rStaticSetpoint	OUT	0.0	REAL	Evaluated droop feedback [%]

Note

The "rVelSmoothTimeOnOff" input is a PT1 element that applies only during switch-on and off. When switching off, the inverse function of the PT 1 element is used so there is no jump for either switch-on or switch-off. The time value entered in the "rVelSmoothTimeOnOff" corresponds to 1 Tau (1 τ).

5.2.9 DCC_GrabMonitor

Symbol



Short description

The DCC block DCC_GrabMonitor can be used to detect bulky load material in the closing gear.

Mode of operation

When the grab of the closing gear attempts to close around bulky material (wood, for example), it may not be able to close completely. The message "boMaxTorqueReached" is generated if an adjustable torque value is exceeded as the grab closes.

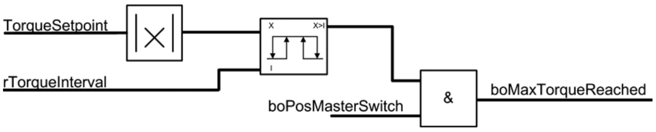


Figure 5-8 Overview of GrabMonitor

Note

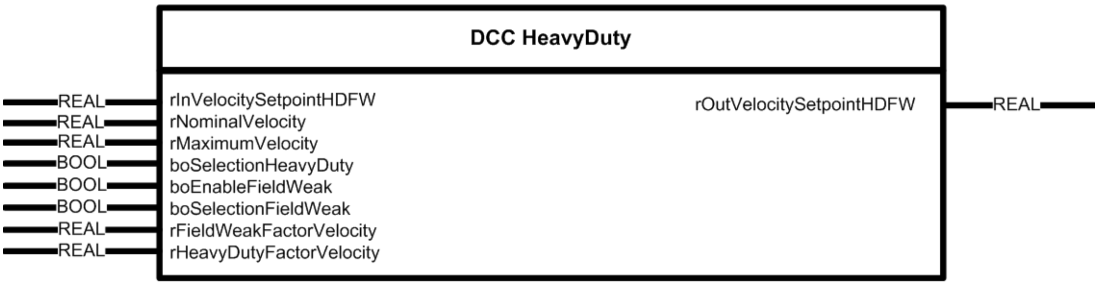
The reference torque in unit [Nm] at input "rTorqueNominal" must be entered as a fixed value, or the nominal torque must be interconnected at this input in the user program (refer to the SINAMICS parameter p2003).
The block does not work correctly otherwise.

Connections

Name	Connection type	Default setting	Data type	Meaning
rTorqueSetpoint	IN	MCC	REAL	Setpoint torque [Nm]
rTorqueNominal	IN	MCC	REAL	Reference torque [Nm]
rTorqueSmoothTime	IN	0	SDTIME	Smoothing time for setpoint torque [ms]
rTorqueInterval	IN	0.0	REAL	Interval limit [%]
rTorqueHysteresis	IN	0.0	REAL	Hysteresis [%]
boPositiveMasterSwitch	IN	MCC	BOOL	Close grab
boMaxTorqueReached	OUT	MCC	BOOL	Rough timber

5.2.10 DCC_HeavyDuty

Symbol



Short description

Using the DCC block DCC_HeavyDuty the drive also allows heavy duty operation (HeavyDuty) by changing the velocity, or operation with constant field weakening (FieldWeak).

Mode of operation

This function is required, if with Hoist, loads greater than the rated load are to be raised. In heavy duty operation, loads exceeding the rated load can be raised. However, this is generally only possible with reduced velocity due to the crane mechanical system (static). This is the reason that for this operating mode, the master switch setpoint is multiplied by a selectable factor <1. This means that the deflection range is fully utilized with reduced setpoint input.

Note

Enter the rated velocity in [mm/s] at input "rNominalVelocity" as fixed value, or interconnect the rated velocity with this input in the user program (e.g.:

Enter the maximum velocity in [mm/s] at input "rMaximumVelocity" as fixed value, or interconnect the maximum velocity with this input in the user program (e.g. refer to Technology Object: D4x5.Hoist_1.TypeOfAxis.MaxVelocity.maximum or function block FB_OperationMode output variable r64maximumVelocity).
The block will not work correctly otherwise.

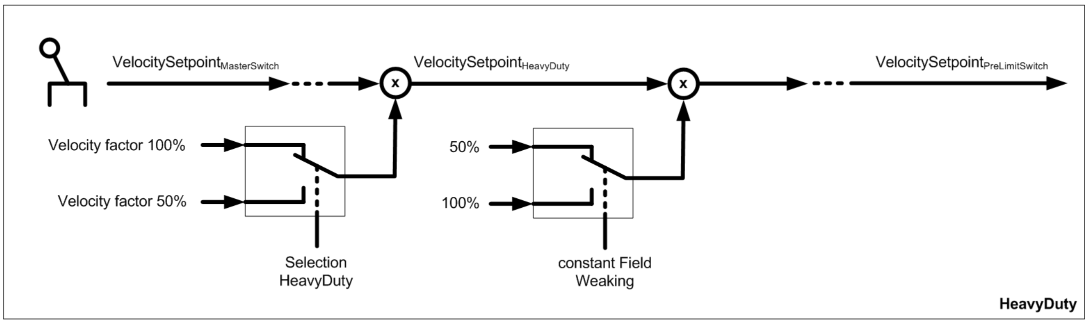

VelocitySetpoint$_{MasterSwitch}$
VelocitySetpoint$_{HeavyDuty}$
VelocitySetpoint$_{PreLimitSwitch}$
x
x
Velocity factor 100%
Velocity factor 50%
50%
100%
Selection HeavyDuty
constant Field Weaking
HeavyDuty

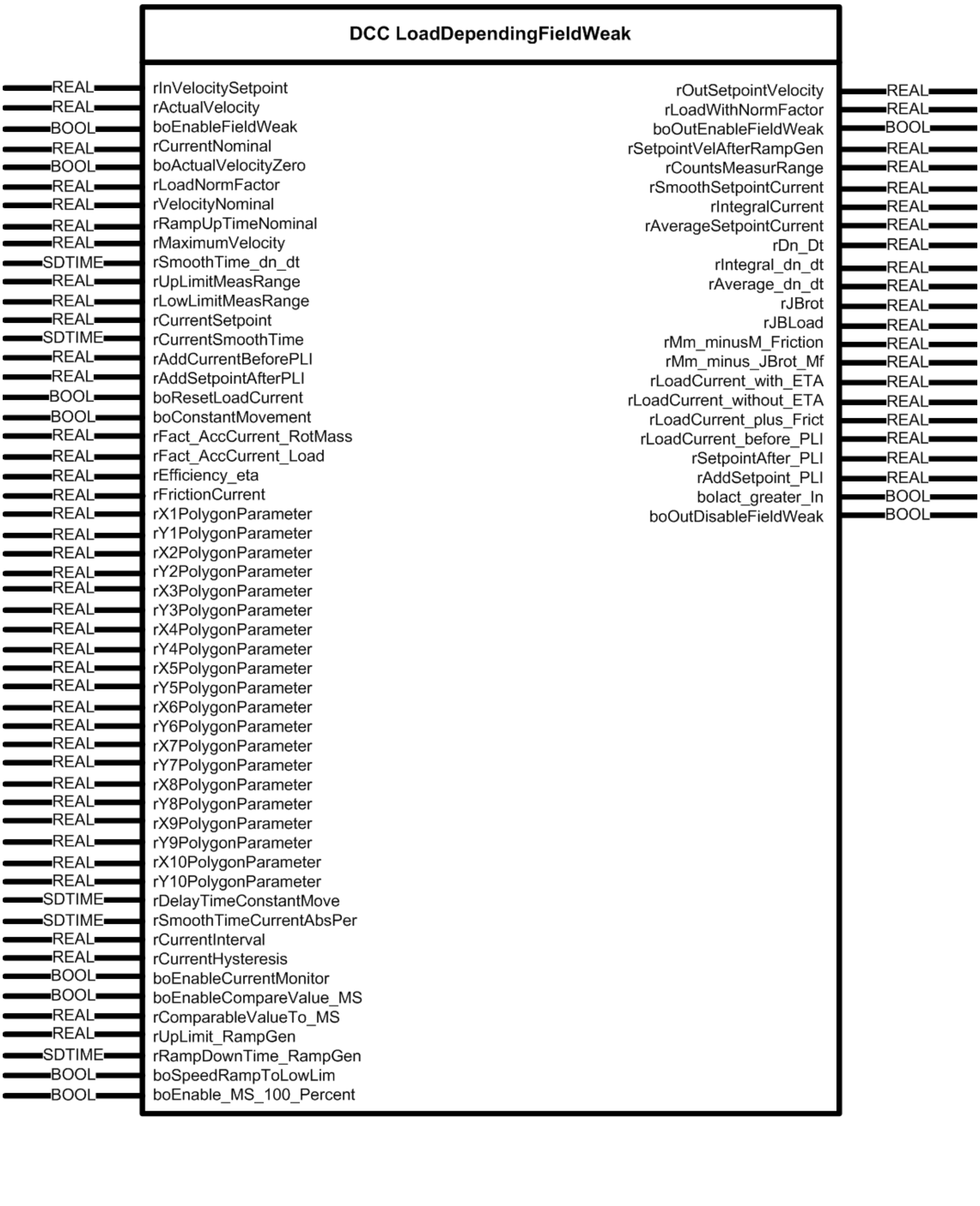
Figure 5-9 Principle of the changeover for heavy loads

Connections

Name	Connection type	Default setting	Data type	Meaning
rInVelocitySetpointHDFW	IN	DCC	REAL	Velocity input value [mm/s]
rNominalVelocity	IN	MCC	REAL	Rated velocity [mm/s]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity [mm/s]
boSelectionHeavyDuty	IN	MCC	BOOL	Selects heavy duty operation. When heavy duty operation is selected, the heavy duty factor is evaluated with the velocity input value.
boEnableFieldWeak	IN	False	BOOL	Enables constant field weakening. If field weakening and the selection of field weakening are enabled, then the velocity input value is evaluated with the field-weakening operating factor.
boSelectionFieldWeak	IN	False	BOOL	Selects constant field weakening. This must be set so that when enable is set, then constant field-weakening operation is also active.
rFieldWeakFactorVelocity	IN	100.0	REAL	Evaluation factor for constant field-weakening operation. This should lie in the range between 40 % and 100 [%].
rHeavyDutyFactorVelocity	IN	50.0	REAL	Evaluation factor for heavy duty operation. This should lie in the range between 0 % and 50 [%].
rOutVelocitySetpointHDFW	OUT	DCC	REAL	Velocity output value [mm/s]

5.2.11 DCC_LoadDependingFieldWeak

Symbol



Brief description

A supplementary speed setpoint is calculated, depending on the load, using the DCC block DCC_LoadDependingFieldWeak. This speed increase for partial loads above the rated speed is required for cranes to increase the handling capacity. This is realized using the motor curve that should be saved.

Mode of operation

When selecting field weakening, e.g. using the master switch, depending on the load, in addition to the maximum value for the rated speed range specified from the master switch, a supplementary speed setpoint for field weakening, which is permissible for the actual load, is generated.

The load actual value is calculated from the torque-generating current and the acceleration torque, which is determined using the speed actual value. The values for current and acceleration are integrated over several sampling times to obtain a more precise figure. An arithmetic average value is formed from this.

During the acceleration phase, the torque-generating current is measured, which in the rated speed range, is directly proportional to the total torque M_M provided by the motor (refer to the formula and its description below). The acceleration torque of the rotating masses (moment of inertia J), the acceleration torque of the load and the friction torque are subtracted from this total torque.

The steady-state load current is calculated using the following torque equation:

$$M_M = M_{Load} + M_{ALoad} + M_{Arot} + M_{Friction}$$

M_M: Motor torque

M_{Load}: Load torque

M_{ALoad}: Load acceleration torque

M_{Arot}: Acceleration torque of the rotating masses

$M_{Friction}$: Frictional torque

After the steady-state load actual value has been calculated (steady-state torque M_L of the load), depending on the calculated torque, the permissible supplementary speed setpoint for constant power $P = P_N$ for field weakening is generated (using the drive PLI, implemented as programmable polygon).

The calculated load-dependent speed limiting value is then incorporated into the setpoint input for the velocity. Once acceleration is over, the total speed setpoint is limited to the calculated load-dependent speed limiting value.

Note

For a detailed description of this block, refer to Chapter Field weakening (LDFW) (Page 414). This chapter also describes how a second gear stage is integrated.

Error messages

The block only supplies one error message that indicates if the current actual value lies above the rated current value. The block output boIact_greater_In then signals a False.

WARNING

This setting should only be carried out by qualified personnel. Particular attention must be paid to the reduced breakdown torque of the drive within the field weakening range. Allowances must be made for appropriate safety reserve in terms of the data provided by the machine manufacturer.

WARNING

Enter the maximum current in [A] at input "rCurrentNominal" as fixed value, or interconnect the maximum current with this input in the user program (refer to the SINAMICS parameter p2002).

Enter the rated velocity in [mm/s] at input "rNominalVelocity" as fixed value, or interconnect the rated velocity with this input in the user program (e.g.:

Enter the maximum velocity in [mm/s] at input "rMaximumVelocity" as fixed value, or interconnect the maximum velocity with this input in the user program. (e.g. refer to Technology Object: D4x5.Hoist_1.TypeOfAxis.MaxVelocity.maximum or function block FB_OperationMode output variable r64maximumVelocity).
The block will not work correctly otherwise.

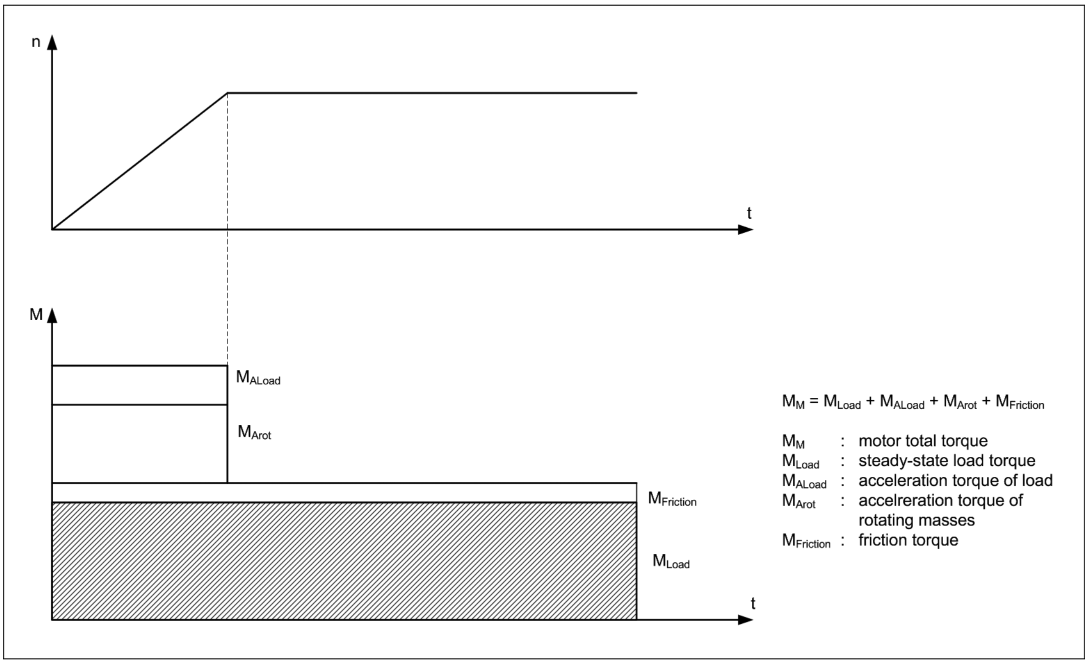

n
t
M
M_{ALoad}
M_{Arot}
$M_{Friction}$
M_{Load}
t
$M_M = M_{Load} + M_{ALoad} + M_{Arot} + M_{Friction}$
M_M : motor total torque
M_{Load} : steady-state load torque
M_{ALoad} : acceleration torque of load
M_{Arot} : accelreration torque of rotating masses
$M_{Friction}$: friction torque

Figure 5-10 Speed – torque profile

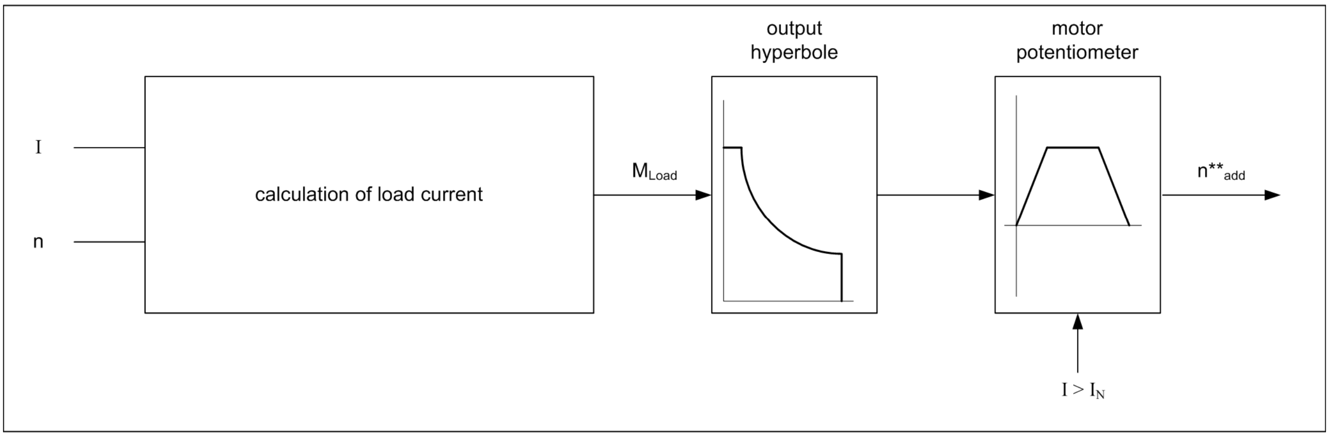


Figure 5-11 Principle when calculating the supplementary speed setpoint

⚠ WARNING

Ensure that you enter the correct values for the polygon parameters! The input parameters "rX1PolygonParameter" to "rX10PolygonParameter" must be assigned in ascending sequence. Proper operation of the block cannot be guaranteed if you do not set the parameters correctly.

Connections

Name	Connection type	Default setting	Data type	Meaning
rInVelocitySetpoint	IN	DCC	REAL	Velocity setpoint [mm/s]
rActualVelocity	IN	MCC	REAL	Velocity actual value [mm/s]
boEnableFieldWeak	IN	MCC	BOOL	Activates field weakening
rCurrentNominal	IN	MCC	REAL	Reference current [A]
boActualVelocityZero	IN	MCC	BOOL	Standstill signal, zero speed signal
rLoadNormFactor	IN	1.0	REAL	Load conversion factor from current value to the physical unit tons
rVelocityNominal	IN	MCC	REAL	Rated velocity (base velocity) [mm/s]
rRampUpTimeNominal	IN	0.0	REAL	Ramp-up time (to the maximum velocity) [ms]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity (base velocity + field-weakening speed) [mm/s]
rSmoothTime_dn_dt	IN	24	SDTIME	Smoothing time [ms] for the velocity setpoint applied to the derivative action element.

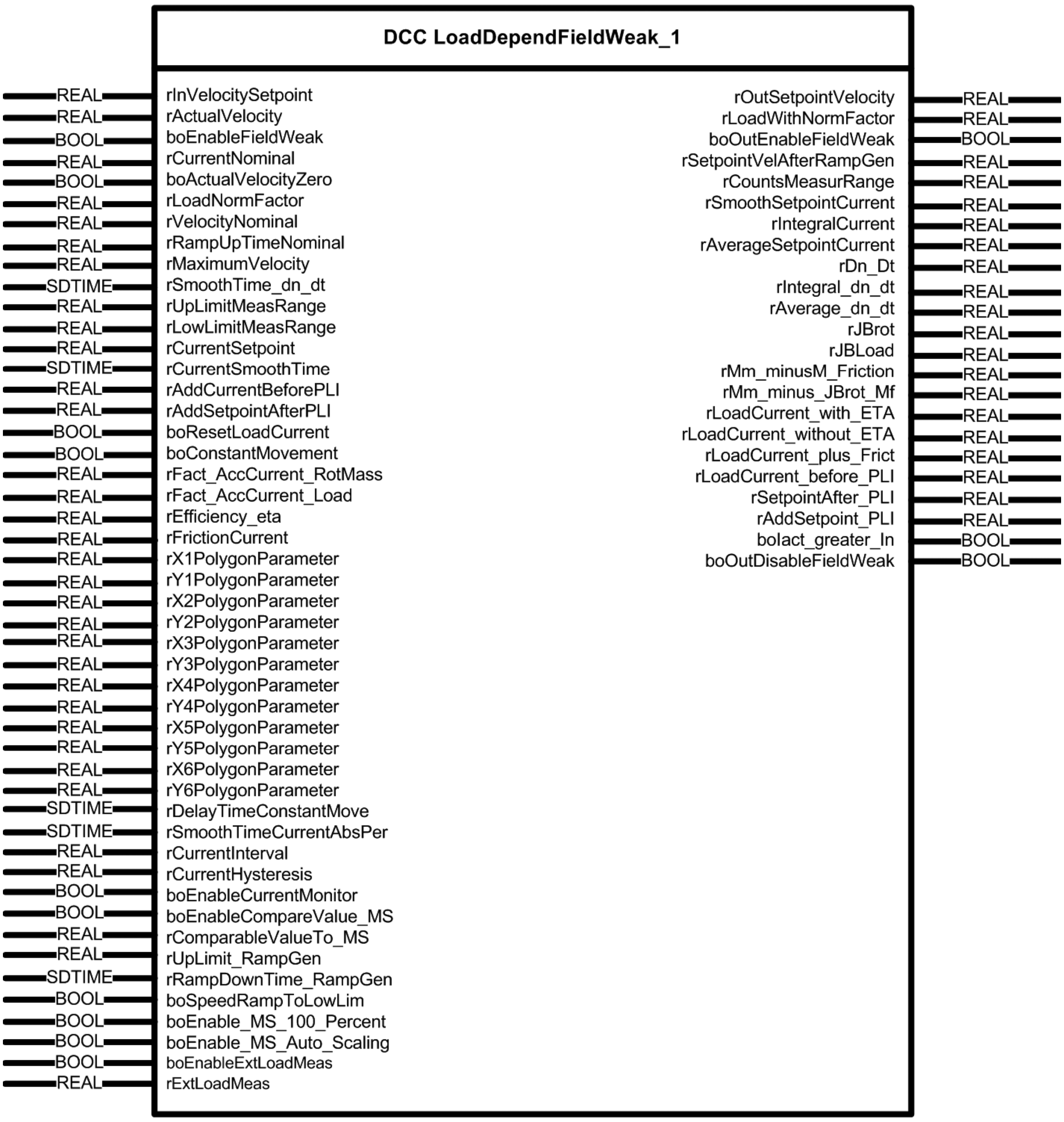
Name	Connection type	Default setting	Data type	Meaning
rUpLimitMeasRange	IN	90.0	REAL	Upper measuring window limit (speed up to which the measuring series is performed) [%]
rLowLimitMeasRange	IN	40.0	REAL	Lower measuring window limit (speed from which the measuring series is carried out) [%]
rCurrentSetpoint	IN	MCC	REAL	Current setpoint [A]
rCurrentSmoothTime	IN	120	SDTIME	Smoothing time for the current [ms]
rAddCurrentBeforePLI	IN	0.0	REAL	Offset [%] of the rated load current before PLI
rAddSetpointAfterPLI	IN	0.0	REAL	Offset [%] for the velocity setpoint after PLI
boResetLoadCurrent	IN	MCC	BOOL	Resets the load current measurement
boConstantMovement	IN	MCC	BOOL	Constant velocity travel
rFact_AccCurrent_RotMass	IN	0.0	REAL	Evaluation for acceleration current of the rot. masses [%]
rFact_AccCurrent_Load	IN	0.0	REAL	Evaluation for the load acceleration current [%]
rEfficiency_eta	IN	100.0	REAL	$1/\eta^2$; (efficiency correction between hoisting and lowering) [%]
rFrictionCurrent	IN	0.0	REAL	Current due to friction [%]
rX1PolygonParameter	IN	0.0	REAL	Polygon, load value X1
rY1PolygonParameter	IN	0.0	REAL	Polygon velocity Y1 [%] with reference to rMaximumVelocity
rX2PolygonParameter	IN	0.0	REAL	Polygon, load value X2
rY2PolygonParameter	IN	0.0	REAL	Polygon velocity Y2 [%] with reference to rMaximumVelocity
rX3PolygonParameter	IN	0.0	REAL	Polygon, load value X3
rY3PolygonParameter	IN	0.0	REAL	Polygon velocity Y3 [%] with reference to rMaximumVelocity
rX4PolygonParameter	IN	0.0	REAL	Polygon, load value X4
rY4PolygonParameter	IN	0.0	REAL	Polygon velocity Y4 [%] with reference to rMaximumVelocity
rX5PolygonParameter	IN	0.0	REAL	Polygon, load value X5
rY5PolygonParameter	IN	0.0	REAL	Polygon velocity Y5 [%] with reference to rMaximumVelocity
rX6PolygonParameter	IN	0.0	REAL	Polygon, load value X6
rY6PolygonParameter	IN	0.0	REAL	Polygon velocity Y6 [%] with reference to rMaximumVelocity
rX7PolygonParameter	IN	0.0	REAL	Polygon, load value X7
rY7PolygonParameter	IN	0.0	REAL	Polygon velocity Y7 [%] with reference to rMaximumVelocity
rX8PolygonParameter	IN	0.0	REAL	Polygon, load value X8
rY8PolygonParameter	IN	0.0	REAL	Polygon velocity Y8 [%] with reference to rMaximumVelocity

Name	Connection type	Default setting	Data type	Meaning
rX9PolygonParameter	IN	0.0	REAL	Polygon, load value X9
rY9PolygonParameter	IN	0.0	REAL	Polygon velocity Y9 [%] with reference to rMaximumVelocity
rX10PolygonParameter	IN	0.0	REAL	Polygon, load value X10
rY10PolygonParameter	IN	0.0	REAL	Polygon velocity Y10 [%] with reference to rMaximumVelocity
rDelayTimeConstantMove	IN	2000	SDTIME	Switch-in delay for current monitoring for constant velocity motion [ms]
rSmoothTimeCurrentAbsPer	IN	20	SDTIME	Smoothing time for the absolute percentage current [ms]
rCurrentInterval	IN	105.0	REAL	Limit value for message I > I_N [%] relative to rCurrentNominal
rCurrentHysteresis	IN	5.0	REAL	Hysteresis for message I > I_N [%] relative to rCurrentNominal
boEnableCurrentMonitor	IN	True	BOOL	Decision, current monitoring (if the current limit is violated, then the drive is ramped down)
boEnableCompareValue_MS	IN	True	BOOL	Decision whether comparison value to deactivate field weakening.
rComparableValueTo_MS	IN	95.0	REAL	Percentage as of which the master switch setpoint (or deflection of the master switch) activates field weakening [%], relative to rMaxVelocity. If the master switch setpoint is lower, field weakening is deactivated.
rUpLimit_RampGen	IN	100.0	REAL	Upper limit value for the determined supplementary speed setpoint [%]
rRampDownTime_RampGen	IN	12000	SDTIME	Ramp-down time to reduce the setpoint [ms]
boSpeedRampToLowLim	IN	True	BOOL	For true, the setpoint is reduced to the rated velocity
boEnable_MS_100_Percent	IN	False	BOOL	Deflection of the master switch across the entire range. No limit for changeover response when reaching the master switch setpoint (rComparableValueTo_MS)
rOutSetpointVelocity	OUT	DCC	REAL	Effective overall velocity setpoint [mm/s]
rLoadWithNormFactor	OUT	MCC	REAL	Evaluated actual position value [e.g. in tons, depending on rLoadNormfactor]
boOutEnableFieldWeak	OUT	MCC DCC	BOOL	Field weakening active (all measurements are valid)
rSetpointVelAfterRampGen	OUT	DCC	REAL	Effective speed setpoint in [%] relative to rMaximumVelocity
rCountsMeasurRange	OUT	0.0	REAL	Number of sampling operations in the measuring window

Name	Connection type	Default setting	Data type	Meaning
rSmoothSetpointCurrent	OUT	0.0	REAL	Smoothed actual current [%] relative to rCurrentNominal
rIntegralCurrent	OUT	0.0	REAL	Additive current setpoint [%] relative to rCurrentNominal
rAverageSetpointCurrent	OUT	0.0	REAL	Arithmetic mean value of current setpoint [%] relative to rCurrentNominal
rDn_Dt	OUT	0.0	REAL	Actual dn / dt [%]
rIntegral_dn_dt	OUT	0.0	REAL	Summed dn / dt [%]
rAverage_dn_dt	OUT	0.0	REAL	Arithmetic average value dn / dt [%]
rJBrot	OUT	0.0	REAL	Acceleration current of the rot. masses [%] relative to rCurrentNominal
rJBLoad	OUT	0.0	REAL	Acceleration current [%] relative to rCurrentNominal
rMm_minusM_Friction	OUT	0.0	REAL	$I_{tot} – I_{friction}$ (total current – frictional current) [%] relative to rCurrentNominal
rMm_minus_JBrot_Mf	OUT	0.0	REAL	$I_{tot} – I_{friction} – I_{Brot}$ (total current - frictional component - rot. masses) [%] relative to rCurrentNominal
rLoadCurrent_with_eta	OUT	0.0	REAL	$I_{load} • 1/η^2$ load current with efficiency correction [%] relative to rCurrentNominal
rLoadCurrent_without_eta	OUT	0.0	REAL	Load current [%] relative to rCurrentNominal
rLoadCurrent_plus_Frict	OUT	0.0	REAL	$I_{load} + I_{friction}$ (load current + frictional current) [%] relative to rCurrentNominal
rLoadCurrent_before_PLI	OUT	0.0	REAL	Load current before PLI [%] relative to rCurrentNominal
rSetpointAfter_PLI	OUT	0.0	REAL	Auxiliary speed setpoint in [%] relative to rMaximumVelocity
rAddSetpoint_PLI	OUT	0.0	REAL	Offset in [%] relative to rMaximumVelocity
boIact_greater_In	OUT	True	BOOL	Message $I > I_N$ signal 0 (for reason of safety) signal 1 if $I < I_N$
boOutDisableFieldWeak	OUT	False	BOOL	Displays reset, field weakening

5.2.12 DCC_LoadDependFieldWeak_1

Symbol



Brief description

A supplementary speed setpoint depending on the load is calculated using the DCC block DCC_LoadDependFieldWeak_1. This speed increase for partial loads above the rated speed is required for cranes to increase the handling capacity. This is realized using the motor curve that should be saved.

The block has a 6-point polygon.

Mode of operation

In principle, this block works in the same way as the DCC block DCC_LoadDependingFieldWeak (refer to Chapter DCC_LoadDependingFieldWeak (Page 106)). The 10-point polygon was reduced to 6 points so that the motor curve can be mapped with 6 points.

In addition, it is now possible to connect the normalized value of a load measuring cell using rExtLoadmeas and enable it using the boEnableExtLoadmeas input. By enabling the external load measuring cell, the load calculated in the block is not forwarded immediately, but compared with the value of the external load measuring cell. Only the maximum value is forwarded and a setpoint depending on the motor curve is enabled only with this value.

The last change affects the behavior of the master switch when field weakening is enabled. In this case, there were the following two variants in DCC_LoadDependingFieldWeak:

Variant 1:
boenable_MS_100_Percent = False

The master switch has two different reference speeds. At the start of a hoisting movement, the reference speed is the rated speed. This means that the deflection (as percentage) is multiplied with the rated speed provided the field weakening has not been enabled. After the enable of the field weakening, the reference speed changes immediately from the rated speed to the maximum speed. The deflection (as percentage) is now multiplied with the maximum speed.

This switchover has the advantage that the velocity at the start can be set finer because the full deflection of the master switch cannot exceed the rated speed.

Example: rCompareValueTo_MS = 95%, boEnableCompareValue_MS = True

Up to 95% of the master switch deflection means up to 95% of the rated speed. This gives a large range of the master switch in order to set the velocity finely in the rated speed range. As soon as the deflection exceeds 95% and the field weakening is enabled, the maximum speed is the reference speed. The deflection of the master switch between 95 - 100% thus causes a speed change of 0.95 * rated speed - maximum speed. If the value exceeds or drops below the 95%, this causes an immediate setpoint change because the reference speed changes immediately. This means the velocity transition from rated speed to field weakening is not linear.

If the block, for example, releases 97% of the maximum velocity, a deflection of the final three percent of the master switch does not change the velocity. This means the deflection range 97-100% remains unused.

Variant 2:
boenable_MS_100_Percent = TRUE

The master switch always has the maximum speed as reference speed. The deflection of the master switch between 0% and 100% causes a velocity change between 0 - maximum speed without a switchover occurring. The velocity transition from rated speed to field weakening remains linear.

If the block, for example, releases 97% of the maximum velocity, a deflection of the final three percent of the master switch does not change the velocity. This means the deflection range 97-100% remains unused.

DCC_LoadDependFieldWeak_1 has been given an additional option.

Variant 3: boenable_Auto_Scaling = TRUE, boenable_MS_100_Percent = FALSE

Like variant 1, the master switch has two reference speeds. Provided the field weakening is not enabled, the rated speed is the reference speed. As soon as field weakening is enabled, the reference speed changes immediately from the rated speed to the calculated maximum speed. If the block, for example, releases 97% of the maximum velocity, the reference speed is then 0.97 * maximum velocity.

This means a deflection between 0 - 100% causes a velocity change between 0 - 97%. This means the deflection range of the master switch is fully utilized.

If boenable_Auto_Scaling and boenable_MS_100_Percent are both TRUE, variant 2 remains active.

Note

In variant 3, the reference speed after the internal enabling of the field weakening is always the calculated maximum speed. This means the velocity change for the deflection always depends on the load. If the load measurement 90% of the maximum velocity permits, the deflection range is then scaled to 90%. If the load measurement releases only 55%, the deflection range is scaled to 55%. The same deflection but different load also results in different setpoint velocities.

Note

For a detailed description of the commissioning for this block, refer to Chapter Field weakening (LDFW) (Page 414). This chapter also describes how a second gear stage is integrated.

Error messages

The block only supplies one error message that indicates if the current actual value lies above the rated current value. The block output boIact_greater_In then signals a False.

⚠ WARNING

This setting should only be carried out by qualified personnel. Particular attention must be paid to the reduced breakdown torque of the drive within the field weakening range. Allowances must be made for appropriate safety reserve in terms of the data provided by the machine manufacturer.

⚠WARNING

Enter the maximum current in [A] at input "rCurrentNominal" as fixed value, or interconnect the maximum current with this input in the user program (refer to the SINAMICS parameter p2002).

Enter the rated velocity in [mm/s] at input "rVelocityNominal" as fixed value or interconnect the rated velocity with this input in the user program (e.g.:

Enter the maximum velocity in [mm/s] at input "rMaximumVelocity" as fixed value, or interconnect the maximum velocity with this input in the user program. (e.g. refer to Technology Object: D4x5.Hoist_1.TypeOfAxis.MaxVelocity.maximum or function block FB_OperationMode output variable r64maximumVelocity).
The block will not work correctly otherwise.

⚠WARNING

Ensure that you enter the correct values for the polygon parameters! The input parameters "rX1PolygonParameter" to "rX6PolygonParameter" must be assigned in ascending sequence. Proper operation of the block cannot be guaranteed if you do not set the parameters correctly.

Connections

Name	Connection type	Default setting	Data type	Meaning
rInVelocitySetpoint	IN	DCC	REAL	Velocity setpoint [mm/s]
rActualVelocity	IN	MCC	REAL	Velocity actual value [mm/s]
boEnableFieldWeak	IN	MCC	BOOL	Activates field weakening
rCurrentNominal	IN	MCC	REAL	Reference current [A]
boActualVelocityZero	IN	MCC	BOOL	Standstill signal, zero speed signal
rLoadNormFactor	IN	1.0	REAL	Load conversion factor from current value to the physical unit tons
rVelocityNominal	IN	MCC	REAL	Rated velocity (base velocity) [mm/s]
rRampUpTimeNominal	IN	0.0	REAL	Rated ramp-up time [ms]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity (base velocity + field-weakening speed) [mm/s]
rSmoothTime_dn_dt	IN	24	SDTIME	Smoothing time [ms] for the velocity setpoint applied to the derivative action element.
rUpLimitMeasRange	IN	90.0	REAL	Upper measuring window limit (speed up to which the measuring series is performed) [%]
rLowLimitMeasRange	IN	40.0	REAL	Lower measuring window limit (speed from which the measuring series is carried out) [%]
rCurrentSetpoint	IN	MCC	REAL	Current setpoint [A]

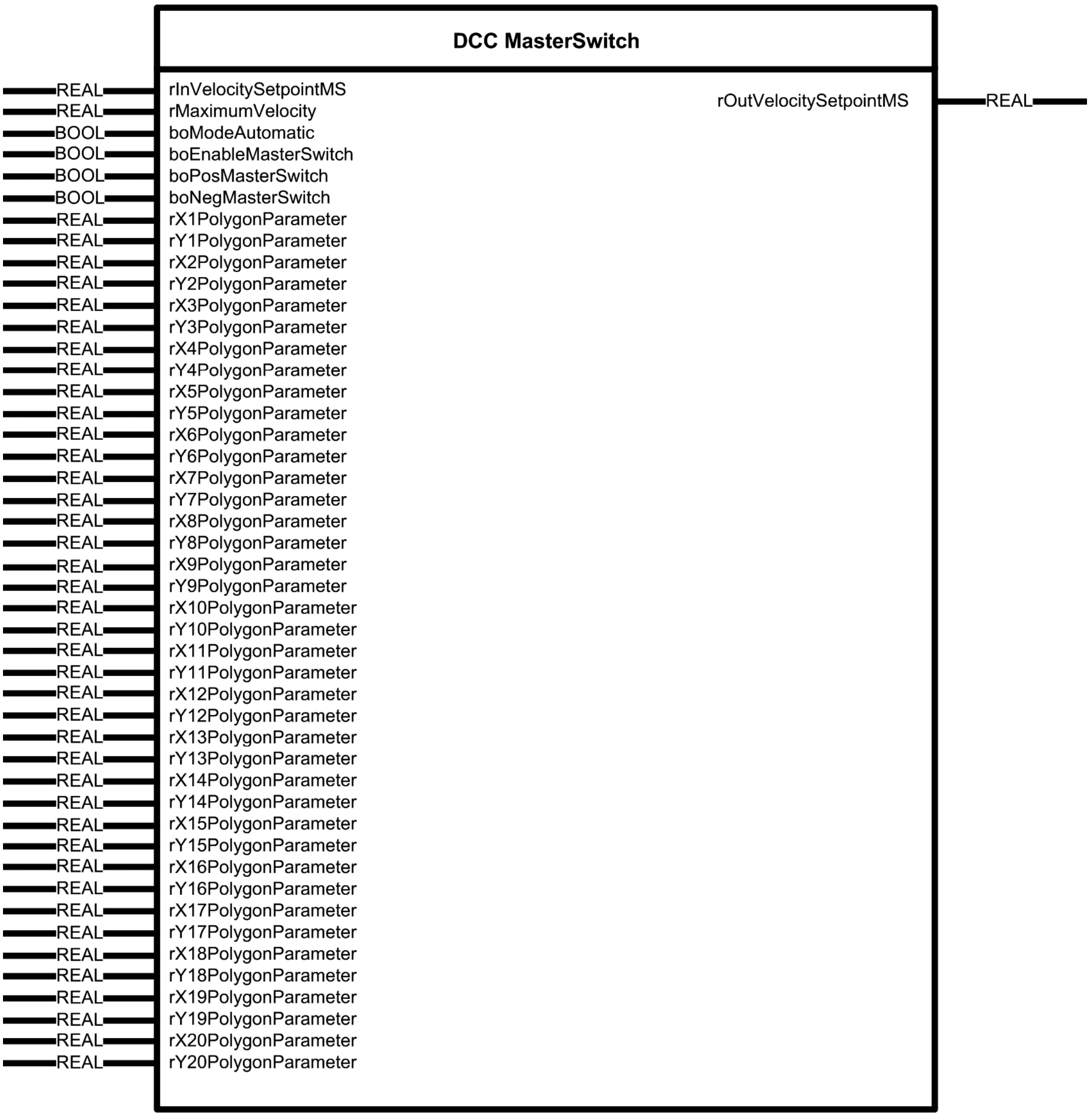
Name	Connection type	Default setting	Data type	Meaning
rCurrentSmoothTime	IN	120	SDTIME	Smoothing time for the current [ms]
rAddCurrentBeforePLI	IN	0.0	REAL	Offset [%] of the rated load current before PLI
rAddSetpointAfterPLI	IN	0.0	REAL	Offset [%] for the velocity setpoint after PLI
boResetLoadCurrent	IN	MCC	BOOL	Resets the load current measurement
boConstantMovement	IN	MCC	BOOL	Constant velocity travel
rFact_AccCurrent_RotMass	IN	0.0	REAL	Evaluation for acceleration current of the rot. masses [%]
rFact_AccCurrent_Load	IN	0.0	REAL	Evaluation for the load acceleration current [%]
rEfficiency_eta	IN	100.0	REAL	1/η^2; (efficiency correction between hoisting and lowering) [%]
rFrictionCurrent	IN	0.0	REAL	Current due to friction [%]
rX1PolygonParameter	IN	0.0	REAL	Polygon, load value X1
rY1PolygonParameter	IN	0.0	REAL	Polygon velocity Y1 [%] with reference to rMaximumVelocity
rX2PolygonParameter	IN	0.0	REAL	Polygon, load value X2
rY2PolygonParameter	IN	0.0	REAL	Polygon velocity Y2 [%] with reference to rMaximumVelocity
rX3PolygonParameter	IN	0.0	REAL	Polygon, load value X3
rY3PolygonParameter	IN	0.0	REAL	Polygon velocity Y3 [%] with reference to rMaximumVelocity
rX4PolygonParameter	IN	0.0	REAL	Polygon, load value X4
rY4PolygonParameter	IN	0.0	REAL	Polygon velocity Y4 [%] with reference to rMaximumVelocity
rX5PolygonParameter	IN	0.0	REAL	Polygon, load value X5
rY5PolygonParameter	IN	0.0	REAL	Polygon velocity Y5 [%] with reference to rMaximumVelocity
rX6PolygonParameter	IN	0.0	REAL	Polygon, load value X6
rY6PolygonParameter	IN	0.0	REAL	Polygon velocity Y6 [%] with reference to rMaximumVelocity
rX7PolygonParameter	IN	0.0	REAL	Polygon, load value X7
rY7PolygonParameter	IN	0.0	REAL	Polygon velocity Y7 [%] with reference to rMaximumVelocity
rX8PolygonParameter	IN	0.0	REAL	Polygon, load value X8
rY8PolygonParameter	IN	0.0	REAL	Polygon velocity Y8 [%] with reference to rMaximumVelocity
rX9PolygonParameter	IN	0.0	REAL	Polygon, load value X9
rY9PolygonParameter	IN	0.0	REAL	Polygon velocity Y9 [%] with reference to rMaximumVelocity
rX10PolygonParameter	IN	0.0	REAL	Polygon, load value X10
rY10PolygonParameter	IN	0.0	REAL	Polygon velocity Y10 [%] with reference to rMaximumVelocity

Name	Connection type	Default setting	Data type	Meaning
rDelayTimeConstantMove	IN	2000	SDTIME	Switch-in delay for current monitoring for constant velocity motion [ms]
rSmoothTimeCurrentAbsPer	IN	20	SDTIME	Smoothing time for the absolute percentage current [ms]
rCurrentInterval	IN	105.0	REAL	Limit value for message I > I_N [%] relative to rCurrentNominal
rCurrentHysteresis	IN	5.0	REAL	Hysteresis for message I > I_N [%] relative to rCurrentNominal
boEnableCurrentMonitor	IN	True	BOOL	Decision, current monitoring (if the current limit is violated, then the drive is ramped down)
boEnableCompareValue_MS	IN	True	BOOL	Decision whether comparison value to deactivate field weakening.
rComparableValueTo_MS	IN	95.0	REAL	Percentage as of which the master switch setpoint (or deflection of the master switch) activates field weakening [%], relative to rMaxVelocity. If the master switch setpoint is lower, field weakening is deactivated.
rUpLimit_RampGen	IN	100.0	REAL	Upper limit value for the determined supplementary speed setpoint [%]
rRampDownTime_RampGen	IN	12000	SDTIME	Ramp-down time to reduce the setpoint [ms]
boSpeedRampToLowLim	IN	True	BOOL	For true, the setpoint is reduced to the rated velocity
boEnable_MS_100_Percent	IN	False	BOOL	Deflection of the master switch across the entire range. No limit for changeover response when reaching the master switch setpoint (rComparableValueTo_MS)
boEnable_MS_Auto_Scaling	IN	False	BOOL	Deflection range of the master switch is scaled to the calculated maximum speed.
boEnableExtLoadMeas	IN	False	BOOL	External load measuring cell is enabled.
rExtLoadMeas	IN	0.0	REAL	Input for normalized value from the load measuring cell
rOutSetpointVelocity	OUT	DCC	REAL	Effective overall velocity setpoint [mm/s]
rLoadWithNormFactor	OUT	MCC	REAL	Evaluated actual position value [e.g. in tons, depending on rLoadNormfactor]
boOutEnableFieldWeak	OUT	MCC DCC	BOOL	Field weakening active (all measurements are valid)
rSetpointVelAfterRampGen	OUT	DCC	REAL	Effective speed setpoint in [%] relative to rMaximumVelocity
rCountsMeasurRange	OUT	0.0	REAL	Number of sampling operations in the measuring window

Name	Connection type	Default setting	Data type	Meaning
rSmoothSetpointCurrent	OUT	0.0	REAL	Smoothed actual current [%] relative to rCurrentNominal
rIntegralCurrent	OUT	0.0	REAL	Additive current setpoint [%] relative to rCurrentNominal
rAverageSetpointCurrent	OUT	0.0	REAL	Arithmetic mean value of current setpoint [%] relative to rCurrentNominal
rDn_Dt	OUT	0.0	REAL	Actual dn / dt [%]
rIntegral_dn_dt	OUT	0.0	REAL	Summed dn / dt [%]
rAverage_dn_dt	OUT	0.0	REAL	Arithmetic average value dn / dt [%]
rJBrot	OUT	0.0	REAL	Acceleration current of the rot. masses [%] relative to rCurrentNominal
rJBLoad	OUT	0.0	REAL	Acceleration current [%] relative to rCurrentNominal
rMm_minusM_Friction	OUT	0.0	REAL	I_{tot} – $I_{friction}$ (total current – frictional current) [%] relative to rCurrentNominal
rMm_minus_JBrot_Mf	OUT	0.0	REAL	I_{tot} – $I_{friction}$ – I_{Brot} (total current - frictional component - rot. masses) [%] relative to rCurrentNominal
rLoadCurrent_with_eta	OUT	0.0	REAL	I_{load} • $1/\eta^2$ load current with efficiency correction [%] relative to rCurrentNominal
rLoadCurrent_without_eta	OUT	0.0	REAL	Load current [%] relative to rCurrentNominal
rLoadCurrent_plus_Frict	OUT	0.0	REAL	I_{load} + $I_{friction}$ (load current + frictional current) [%] relative to rCurrentNominal
rLoadCurrent_before_PLI	OUT	0.0	REAL	Load current before PLI [%] relative to rCurrentNominal
rSetpointAfter_PLI	OUT	0.0	REAL	Auxiliary speed setpoint in [%] relative to rMaximumVelocity
rAddSetpoint_PLI	OUT	0.0	REAL	Offset in [%] relative to rMaximumVelocity
boIact_greater_In	OUT	True	BOOL	Message $I > I_N$ signal 0 (for reason of safety) signal 1 if $I < I_N$
boOutDisableFieldWeak	OUT	False	BOOL	Displays reset, field weakening

5.2.13 DCC_MasterSwitch

Symbol


DCC MasterSwitch
REAL rInVelocitySetpointMS
REAL rMaximumVelocity
BOOL boModeAutomatic
BOOL boEnableMasterSwitch
BOOL boPosMasterSwitch
BOOL boNegMasterSwitch
REAL rX1PolygonParameter
REAL rY1PolygonParameter
REAL rX2PolygonParameter
REAL rY2PolygonParameter
REAL rX3PolygonParameter
REAL rY3PolygonParameter
REAL rX4PolygonParameter
REAL rY4PolygonParameter
REAL rX5PolygonParameter
REAL rY5PolygonParameter
REAL rX6PolygonParameter
REAL rY6PolygonParameter
REAL rX7PolygonParameter
REAL rY7PolygonParameter
REAL rX8PolygonParameter
REAL rY8PolygonParameter
REAL rX9PolygonParameter
REAL rY9PolygonParameter
REAL rX10PolygonParameter
REAL rY10PolygonParameter
REAL rX11PolygonParameter
REAL rY11PolygonParameter
REAL rX12PolygonParameter
REAL rY12PolygonParameter
REAL rX13PolygonParameter
REAL rY13PolygonParameter
REAL rX14PolygonParameter
REAL rY14PolygonParameter
REAL rX15PolygonParameter
REAL rY15PolygonParameter
REAL rX16PolygonParameter
REAL rY16PolygonParameter
REAL rX17PolygonParameter
REAL rY17PolygonParameter
REAL rX18PolygonParameter
REAL rY18PolygonParameter
REAL rX19PolygonParameter
REAL rY19PolygonParameter
REAL rX20PolygonParameter
REAL rY20PolygonParameter
rOutVelocitySetpointMS REAL

Brief description

Using this DCC block DCC_MasterSwitch, the drive can be moved with a fine sensitivity using the master switch for manual positioning.

Mode of operation

Often, the master switch supplies a setpoint that is directly proportional to its angle of deflection. In order that for low deflection angles lower speed setpoints are obtained than those that correspond linearly to the deflection angle, the master switch setpoint is passed through a non-linear function. This allows the drive to be precisely positioned in the manual mode. For low deflection angles, this results in lower speed changes than for the same change of the deflection angle in the medium and higher deflection range. The non-linear function is realized in quadrants I and III using 20 adjustable straight line sections.

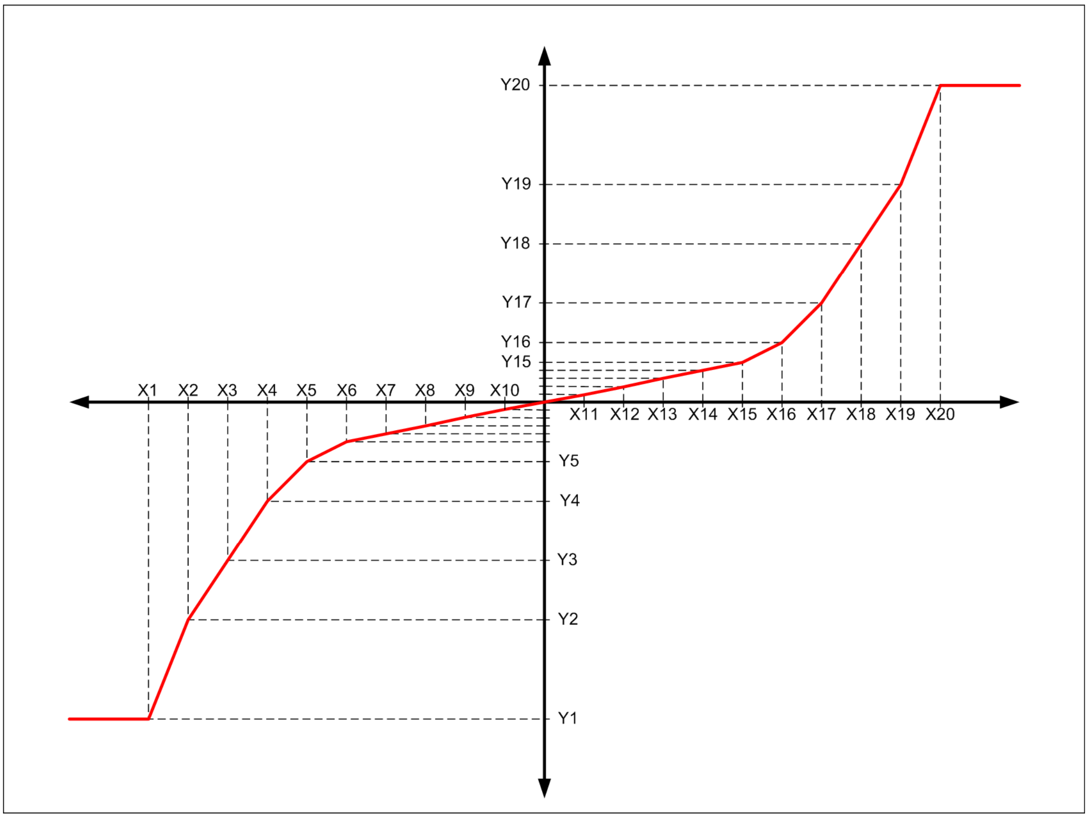


Figure 5-12 Non-linear simulation of the master switch

NOTICE
The maximum speed [mm/s] is either entered at the "rMaximumVelocity" parameter, or "rMaximumVelocity" is linked with the corresponding program variable. Output "rOutVelocitySetpointMS" is limited to "± rMaximumVelocity".

⚠WARNING

Ensure that you enter the correct values for the polygon parameters! The input parameters "rX1PolygonParameter" to "rX20PolygonParameter" and "rY1PolygonParameter" to "rY20PolygonParameter" must be assigned in ascending sequence. Proper operation of the block cannot be guaranteed if you do not set the parameters correctly.

The master switch curve is invalid in AUTOMATIC mode. In this case, the value of "rInVelocitySetpointMS" bypasses the stored curve.

The value converted from word to integer must be set at the X polygon parameters in all other operating modes. The percentage of maximum velocity is entered at the Y polygon parameters.

Connections

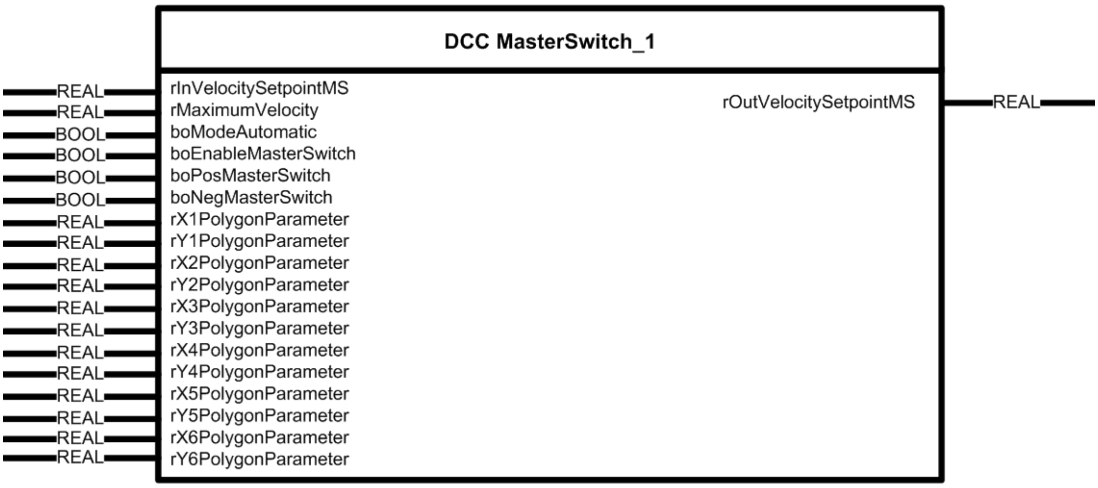
Name	Connection type	Default setting	Data type	Meaning
rInVelocitySetpointMS	IN	MCC	REAL	Master switch input value, implementation deflection angle, input word to integer [-]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity [mm/s]
boModeAutomatic	IN	MCC	BOOL	AUTOMATIC mode is active and the master switch function is enabled. If AUTOMATIC is active, the master switch does not change the setpoint without being deflected. The setpoint is influenced by the master switch curve or an operating mode change as soon as the master switch is deflected.
boEnableMasterSwitch	IN	False	BOOL	Enables the master switch function. After this function is enabled, the master switch curve influences the master switch setpoint in all operating modes except in AUTOMATIC mode. There is a special feature in this mode; refer to the meaning of input "boModeAutomatic".
boPosMasterSwitch	IN	MCC	BOOL	Master switch deflected in the positive direction.
boNegMasterSwitch	IN	MCC	BOOL	Master switch deflected in the negative direction.
rX1PolygonParameter	IN	0.0	REAL	Polygon, value X1
rY1PolygonParameter	IN	0.0	REAL	Polygon velocity Y1 [%] with reference to rMaximumVelocity
rX2PolygonParameter	IN	0.0	REAL	Polygon, value X2
rY2PolygonParameter	IN	0.0	REAL	Polygon velocity Y2 [%] with reference to rMaximumVelocity
rX3PolygonParameter	IN	0.0	REAL	Polygon, value X3

Name	Connection type	Default setting	Data type	Meaning
rY3PolygonParameter	IN	0.0	REAL	Polygon velocity Y3 [%] with reference to rMaximumVelocity
rX4PolygonParameter	IN	0.0	REAL	Polygon, value X4
rY4PolygonParameter	IN	0.0	REAL	Polygon velocity Y4 [%] with reference to rMaximumVelocity
rX5PolygonParameter	IN	0.0	REAL	Polygon, value X5
rY5PolygonParameter	IN	0.0	REAL	Polygon velocity Y5 [%] with reference to rMaximumVelocity
rX6PolygonParameter	IN	0.0	REAL	Polygon, value X6
rY6PolygonParameter	IN	0.0	REAL	Polygon velocity Y6 [%] with reference to rMaximumVelocity
rX7PolygonParameter	IN	0.0	REAL	Polygon, value X7
rY7PolygonParameter	IN	0.0	REAL	Polygon velocity Y7 [%] with reference to rMaximumVelocity
rX8PolygonParameter	IN	0.0	REAL	Polygon, value X8
rY8PolygonParameter	IN	0.0	REAL	Polygon velocity Y8 [%] with reference to rMaximumVelocity
rX9PolygonParameter	IN	0.0	REAL	Polygon, value X9
rY9PolygonParameter	IN	0.0	REAL	Polygon velocity Y9 [%] with reference to rMaximumVelocity
rX10PolygonParameter	IN	0.0	REAL	Polygon, value X10
rY10PolygonParameter	IN	0.0	REAL	Polygon velocity Y10 [%] with reference to rMaximumVelocity
rX11PolygonParameter	IN	0.0	REAL	Polygon, value X11
rY11PolygonParameter	IN	0.0	REAL	Polygon velocity Y11 [%] with reference to rMaximumVelocity
rX12PolygonParameter	IN	0.0	REAL	Polygon, value X12
rY12PolygonParameter	IN	0.0	REAL	Polygon velocity Y12 [%] with reference to rMaximumVelocity
rX13PolygonParameter	IN	0.0	REAL	Polygon, value X13
rY13PolygonParameter	IN	0.0	REAL	Polygon velocity Y13 [%] with reference to rMaximumVelocity
rX14PolygonParameter	IN	0.0	REAL	Polygon, value X14
rY14PolygonParameter	IN	0.0	REAL	Polygon velocity Y14 [%] with reference to rMaximumVelocity
rX15PolygonParameter	IN	0.0	REAL	Polygon, value X15
rY15PolygonParameter	IN	0.0	REAL	Polygon velocity Y15 [%] with reference to rMaximumVelocity
rX16PolygonParameter	IN	0.0	REAL	Polygon, value X16
rY16PolygonParameter	IN	0.0	REAL	Polygon velocity Y16 [%] with reference to rMaximumVelocity
rX17PolygonParameter	IN	0.0	REAL	Polygon, value X17
rY17PolygonParameter	IN	0.0	REAL	Polygon velocity Y17 [%] with reference to rMaximumVelocity
rX18PolygonParameter	IN	0.0	REAL	Polygon, value X18

Name	Connection type	Default setting	Data type	Meaning
rY18PolygonParameter	IN	0.0	REAL	Polygon velocity Y18 [%] with reference to rMaximumVelocity
rX19PolygonParameter	IN	0.0	REAL	Polygon, value X19
rY19PolygonParameter	IN	0.0	REAL	Polygon velocity Y19 [%] with reference to rMaximumVelocity
rX20PolygonParameter	IN	0.0	REAL	Polygon, value X20
rY20PolygonParameter	IN	0.0	REAL	Polygon velocity Y20 [%] with reference to rMaximumVelocity
rOutVelocitySetpointMS	OUT	DCC	REAL	Master switch output value [mm/s]

5.2.14 DCC_MasterSwitch_1

Symbol



Brief description

The DCC block DCC_ContLoadMeasurement_1 has a 6-point polygon. Using this DCC block DCC_MasterSwitch_1, the drive can be moved with a fine sensitivity using the master switch for manual positioning.

Mode of operation

The mode of operation of the DCC_MasterSwitch_1 block essentially corresponds to the mode of operation of the DCC_MasterSwitch block (refer to Chapter DCC_MasterSwitch (Page 120)). The 20 point polygon DCC_MasterSwitch block for speed characteristic is however simplified for DCC_MasterSwitch_1 to a 6-point polygon, whereby the zero point is given. In so doing, the negative component of the characteristic is generated by mirroring quadrant I.

Further, "boPosMasterSwitch" and "boNegMasterSwitch" direction bits are read in. In this case, only one bit may be active. If both bits are simultaneously active, then the block outputs a velocity setpoint of ZERO.

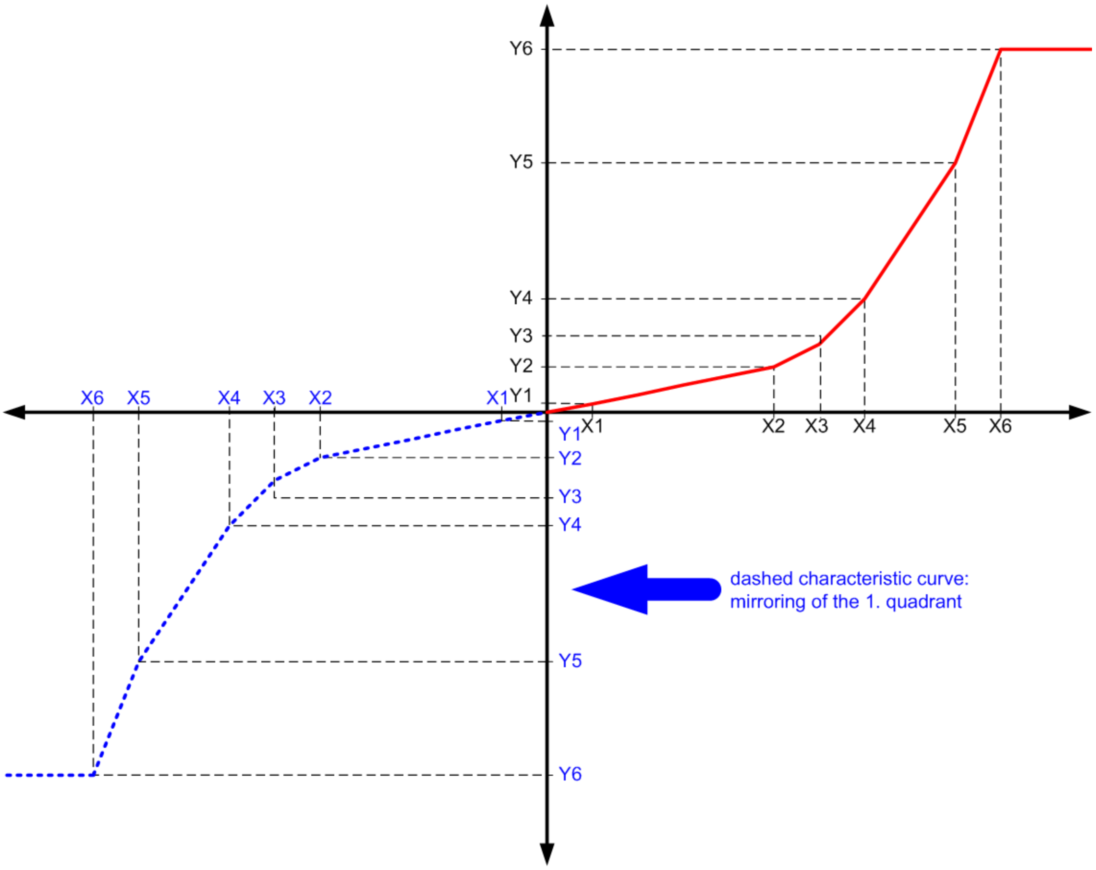


Figure 5-13 Non-linear simulation of the master switch

The master switch curve is invalid in AUTOMATIC mode. In this case, the value of "rInVelocitySetpointMS" bypasses the stored curve.

The value converted from word to integer must be set at the X polygon parameters in all other operating modes. The percentage of maximum velocity is entered at the Y polygon parameters.

NOTICE

The maximum speed [mm/s] is either entered at the "rMaximumVelocity" parameter, or "rMaximumVelocity" is linked with the corresponding program variable. Output "rOutVelocitySetpointMS" is limited to "± rMaximumVelocity".

⚠WARNING

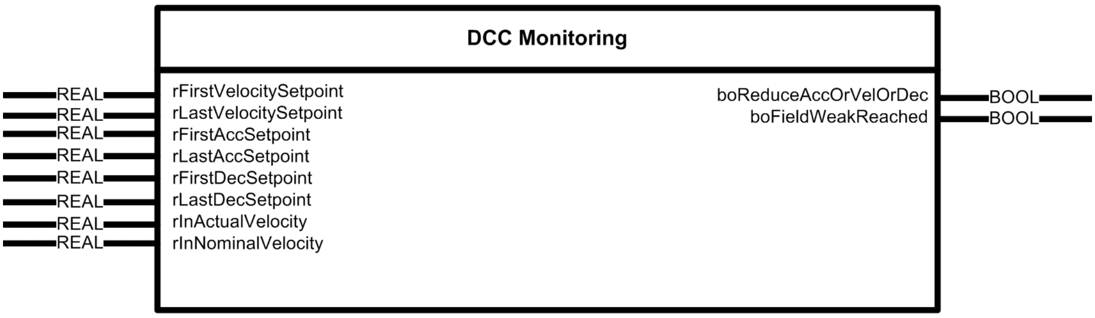
Ensure that you enter the correct values for the polygon parameters! The input parameters "rX1PolygonParameter" to "rX6PolygonParameter" and "rY1PolygonParameter" to "rY6PolygonParameter" must be assigned in ascending sequence. Proper operation of the block cannot be guaranteed if you do not set the parameters correctly.

Connections

Name	Connection type	Default setting	Data type	Meaning
rInVelocitySetpointMS	IN	MCC	REAL	Master switch input value, implementation deflection angle, input word to integer [-]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity [mm/s]
boModeAutomatic	IN	MCC	BOOL	AUTOMATIC operating mode active and the master switch function is enabled. If the AUTOMATIC operating mode is active, and the master switch is not deflected, then the setpoint is not influenced by the master switch curve. As soon as the master switch is deflected, the setpoint is influenced by the master switch curve.
boEnableMasterSwitch	IN	False	BOOL	Enables the master switch function. After this function is enabled, the master switch curve influences the master switch setpoint in all operating modes except in AUTOMATIC mode. There is a special feature in this mode; refer to the meaning of input "boModeAutomatic".
boPosMasterSwitch	IN	MCC	BOOL	Master switch deflected in the positive direction.
boNegMasterSwitch	IN	MCC	BOOL	Master switch deflected in the negative direction.
rX1PolygonParameter	IN	0.0	REAL	Polygon, value X1
rY1PolygonParameter	IN	0.0	REAL	Polygon velocity Y1 [%] with reference to rMaximumVelocity
rX2PolygonParameter	IN	0.0	REAL	Polygon, value X2
rY2PolygonParameter	IN	0.0	REAL	Polygon velocity Y2 [%] with reference to rMaximumVelocity
rX3PolygonParameter	IN	0.0	REAL	Polygon, value X3
rY3PolygonParameter	IN	0.0	REAL	Polygon velocity Y3 [%] with reference to rMaximumVelocity
rX4PolygonParameter	IN	0.0	REAL	Polygon, value X4
rY4PolygonParameter	IN	0.0	REAL	Polygon velocity Y4 [%] with reference to rMaximumVelocity
rX5PolygonParameter	IN	0.0	REAL	Polygon, value X5
rY5PolygonParameter	IN	0.0	REAL	Polygon velocity Y5 [%] with reference to rMaximumVelocity
rX6PolygonParameter	IN	0.0	REAL	Polygon, value X6
rY6PolygonParameter	IN	0.0	REAL	Polygon velocity Y6 [%] with reference to rMaximumVelocity
rOutVelocitySetpointMS	OUT	DCC	REAL	Master switch output value [mm/s]

5.2.15 DCC_Monitoring

Symbol



Short description

The DCC block DCC_Monitoring is used to monitor as to whether the velocity, acceleration or deceleration have been reduced. Further, it is monitored as to whether the drive is in field weakening.

Method of operation

This function is used to identify whether the velocity, acceleration or deceleration has been reduced. The values sent from the control are read-in and compared with the values that go to the technology object. If a deviation occurs, "Message a/v reduction" is transferred to the control in application status word_2_S7, bit 3.

Further, it is monitored as to whether the drive is in field weakening. The actual velocity is compared with the rated speed. If the actual velocity is greater than the rated speed, then the drive is in field weakening.

Note

Enter the rated velocity in [mm/s] at input "rNominalVelocity" as fixed value, or interconnect the rated velocity with this input in the user program.
The block will otherwise not work correctly.

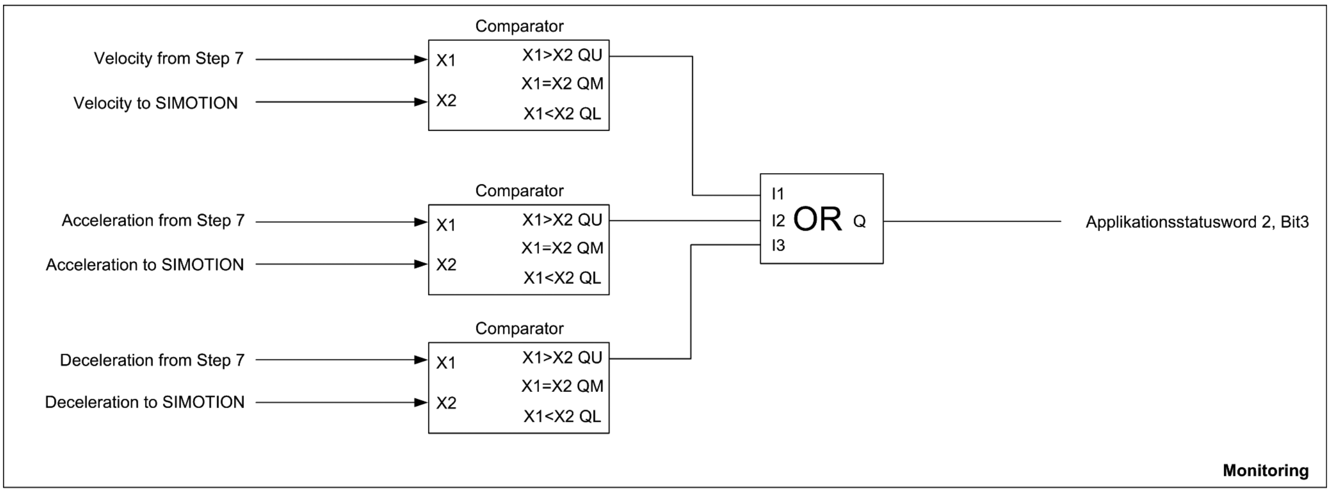


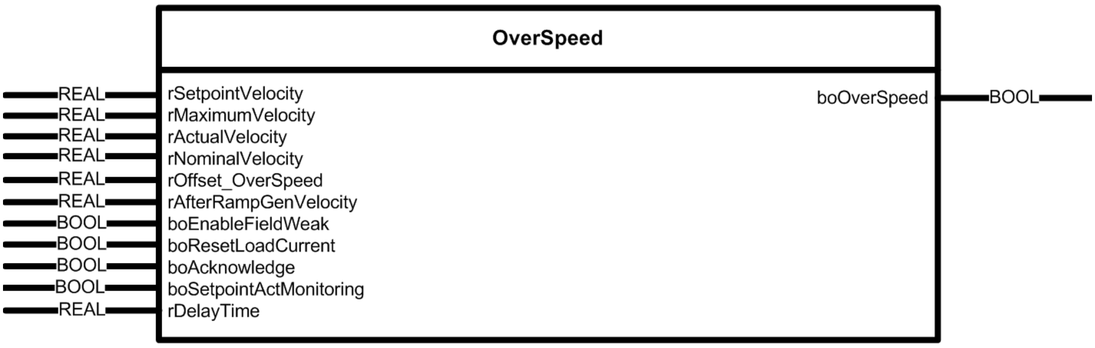
Figure 5-14 Principle of the monitoring function

Connections

Name	Connection type	Default setting	Data type	Meaning
rFirstVelocitySetpoint	IN	MCC	REAL	Velocity setpoint from the control "S7" [mm/s]
rLastVelocitySetpoint	IN	DCC	REAL	Velocity setpoint to the technology object "TO" [mm/s]
rFirstAccSetpoint	IN	MCC	REAL	Acceleration setpoint from the control "S7" [mm/s^2]
rLastAccSetpoint	IN	DCC	REAL	Acceleration setpoint to the technology object "TO" [mm/s^2]
rFirstDecSetpoint	IN	MCC	REAL	Deceleration setpoint from the control "S7" [mm/s^2]
rLastDecSetpoint	IN	DCC	REAL	Deceleration setpoint to the technology object "TO" [mm/s^2]
rInActualVelocity	IN	MCC	REAL	Actual velocity from the drive [mm/s]
rInNominalVelocity	IN	MCC	REAL	Base velocity from the drive [mm/s]
boReduceAccOrVelOrDec	OUT	MCC	BOOL	Velocity, acceleration or deceleration reduction active
boFieldWeakReached	OUT	DCC	BOOL	Drive operates in the field-weakening range.

5.2.16 DCC_OverSpeed

Symbol



Brief description

For hoisting applications, the overspeed can be monitored (in the rated speed range and also in the field weakening range) using the DCC block DCC_OverSpeed, or a setpoint-actual value deviation is detected.

Note

The overspeed monitoring can also be handled in SINAMICS; see Overspeed signal from SINAMICS (Page 338)

Mode of operation

Using this function, either the actual velocity can be monitored for an overspeed condition or a setpoint-actual value deviation detected. These two functions cannot be simultaneously activated.

A limit is defined for the setpoint-actual value deviation. If the actual velocity exceeds this limit for the parameterized delay time "rDelayTime", then output "boOverSpeed" is set. The limit comprises the sum of the setpoint velocity and offset. The setpoint-actual value monitoring is activated using the "boSetpointActMonitoring" input.

The overspeed detection is implemented in the software for various velocities (field weakening range and rated velocity). If the specified offset is exceeded for one of the two velocities longer than the parameterized delay time, then this is signaled using the "boOverSpeed" output.

The setpoint velocity of the load-dependent field weakening is only used as comparison value if the drive is operating in the field-weakening range and field-weakening operation was enabled for the drive. The comparison value of the load-dependent field weakening is used until the determined load is reset.

The overspeed monitoring signal is sent from SIMOTION D to SINAMICS via PROFIBUS Integrated in control word 1 bit 15. In SINAMICS, the signal is connected to the onboard I/O by means of BICO technology, see the table below. The SIMATIC S7 is connected to the SINAMICS CU onboard I/Os via a wire connection, reads this signal in and evaluates the signal status.

Table 5- 3 STS/GSU interconnection of the overspeed signal on SINAMICS Integrated CU

Drives	Terminal	CU output
Hoist 1 / Holding Gear 1	-X122.10	DO 9
Hoist 2 / Holding Gear 2	-X122.13	DO 11
Boom	-X132.9	DO 12

Table 5- 4 Tandem STS/GSU interconnection of the overspeed signal on SINAMICS Integrated CU

Drives	Terminal	CU output
Hoist 1 / Holding Gear 1	-X122.10	DO 9
Hoist 2 / Holding Gear 2	-X122.13	DO 11
Hoist 3	-X132.10	DO 13
Hoist 4	-X132.13	DO 15

Table 5- 5 Tandem STS/GSU interconnection of the overspeed signal on SINAMICS CX32-2 CU

Drives	Terminal	CU output
Boom	-X122.13	DO 11

Note

If there is a fault (boOverSpeed = True), then this is only acknowledged if the condition is fulfilled, so that the fault can no longer occur. Only then is the fault acknowledged.

For DCC_OverSpeed, setting a fault dominates over a reset.

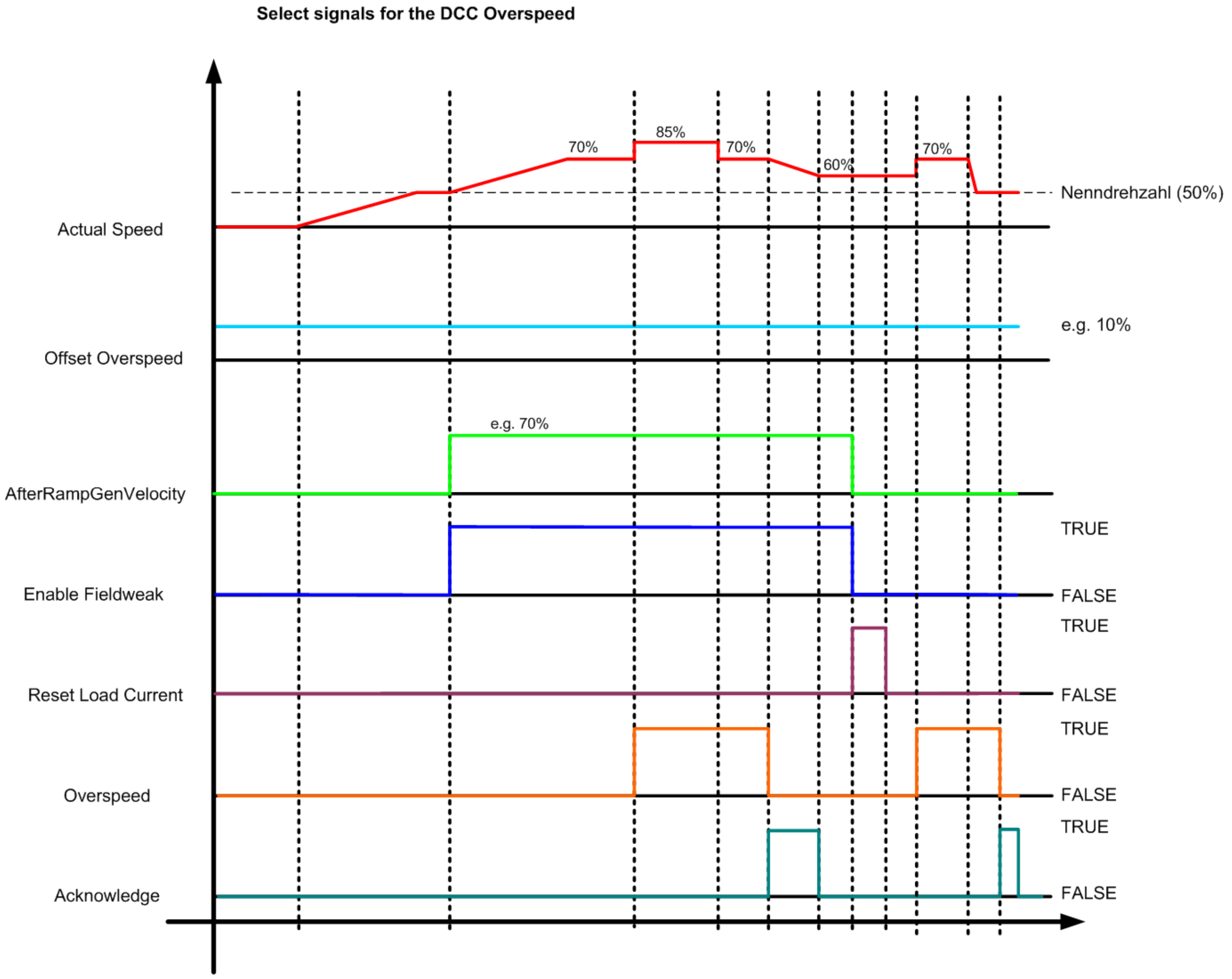


Figure 5-15 Signals to DCC_OverSpeed, variable "boSetpointActMonitoring" = FALSE

CAUTION

Enter the rated velocity in [mm/s] at input "rNominalVelocity" as fixed value, or interconnect the rated velocity with this input in the user program (e.g.:

Enter the maximum velocity in [mm/s] at input "rMaximumVelocity" as fixed value, or interconnect the maximum velocity with this input in the user program•(e.g.: refer to Technology Object: D4x5.Hoist_1.TypeOfAxis.MaxVelocity.maximum or function block FB_OperationMode output variable r64maximumVelocity).
The block will not work correctly otherwise.

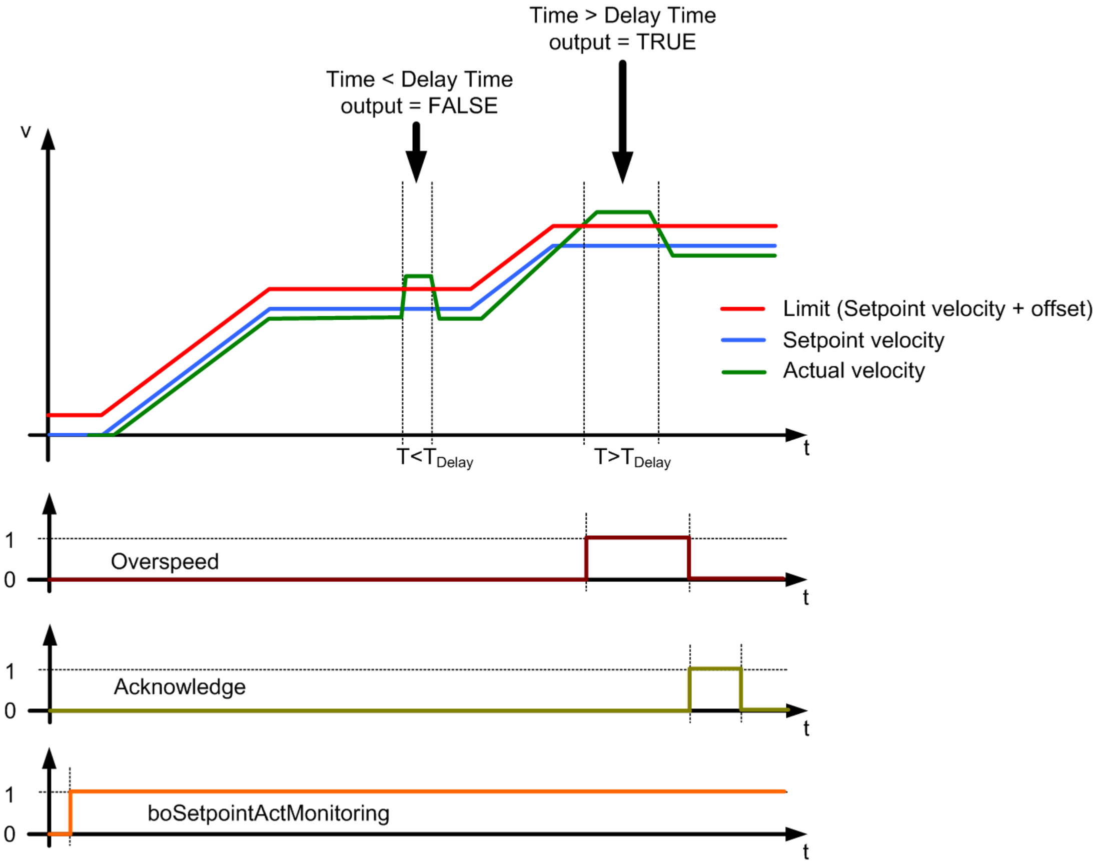


Figure 5-16 Signals to DCC_OverSpeed, variable "boSetpointActualMonitoring" = TRUE

Note

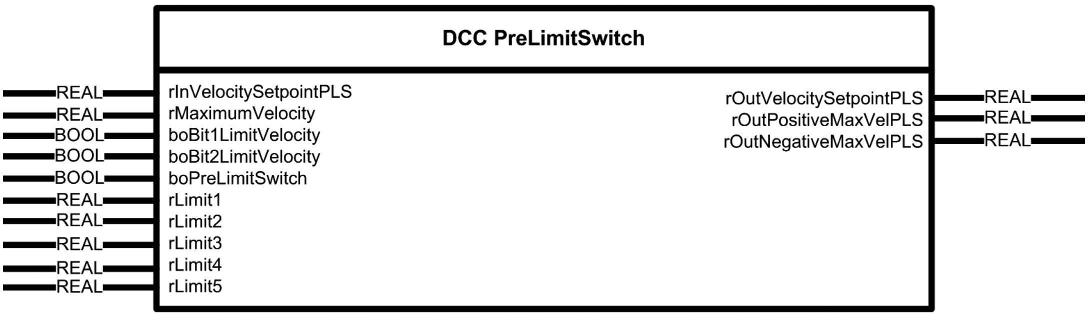
"rOffset_Overspeed" is relative to "rMaximumVelocity" and not to "rSetpointVelocity".

Connections

Name	Connection type	Default setting	Data type	Meaning
rSetpointVelocity	IN	MCC	REAL	Setpoint velocity [mm/s]
rActualVelocity	IN	MCC	REAL	Actual velocity of the drive [mm/s]
rNominalVelocity	IN	MCC	REAL	Rated velocity [mm/s]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity [mm/s]
rOffset_OverSpeed	IN	5.0	REAL	Permissible deviation between the actual velocity and setpoint velocity before an overspeed condition is identified [as %, based on "rMaximumVelocity"]
rAfterRampGenVelocity	IN	DCC	REAL	Velocity setpoint for the load-dependent field weakening
rDelayTime	IN	MCC	SDTIME	Delay time [ms]
boEnableFieldWeak	IN	DCC	BOOL	Field weakening active
boResetLoadCurrent	IN	MCC	BOOL	Load is reset from the load-dependent field weakening.
boAcknowledge	IN	MCC	BOOL	Output "boOverSpeed" is reset
boSetpointActMonitoring	IN	MCC	BOOL	Activating the setpoint-actual value deviation
boOverSpeed	OUT	MCC	BOOL	Overspeed active

5.2.17 DCC_PreLimitSwitch

Symbol



Short description

Using the DCC block DCC_PreLimitSwitch, when a pre-defined pre-limit switch is reached, the drive velocity can be influenced.

Mode of operation

This function prevents that the drive moves with full velocity to the limit switch or to the safety buffer. A total of five different velocity limits can be configured.

Interconnect the maximum velocity with velocity limit 1. This limits velocity to the maximum value.

This means that effectively there are four velocity limits available. Velocity limits 2 to 4 are controlled through the combination of inputs "Bit1LimitVelocity" and "Bit2LimitVelocity".

Velocity limiting 5 (prelimit switch limit) is controlled using the input "boPrelimitSwitch".

Selecting the velocity limit	Bit1Limit-Velocity	Bit2Limit-Velocity	PreLimitSwitch
rLimit 1	FALSE	FALSE	FALSE
rLimit 2	TRUE	FALSE	FALSE
rLimit 3	FALSE	TRUE	FALSE
rLimit 4	TRUE	TRUE	FALSE
rLimit 5 (prelimit switch limit)	no influence	no influence	TRUE

Setting the velocity limits:

The velocity limits should be set as follows:

1. Velocity limit 1 > velocity limit 2
2. Velocity limit 2 > velocity limit 3
3. Velocity limit 3 > velocity limit 4
4. Velocity limit 4 > velocity limit 5
   (prelimit switch limit)

Setting tip:

Velocity limit 1	=	Maximum velocity
Velocity limit 2	=	50 % of the maximum velocity
Velocity limit 3	=	25 % of the maximum velocity
Velocity limit 4	=	12 % of the maximum velocity
Velocity limit 5	=	5 % of the maximum velocity

NOTICE

Enter the maximum velocity in [mm/s] at input "rMaximumVelocity" as fixed value, or interconnect the maximum velocity with this input in the user program (e.g.: refer to Technology Object: D4x5.Gantry_1.TypeOfAxis.MaxVelocity.maximum or function block FB_OperationMode output variable r64maximumVelocity).
The block will not work correctly otherwise.

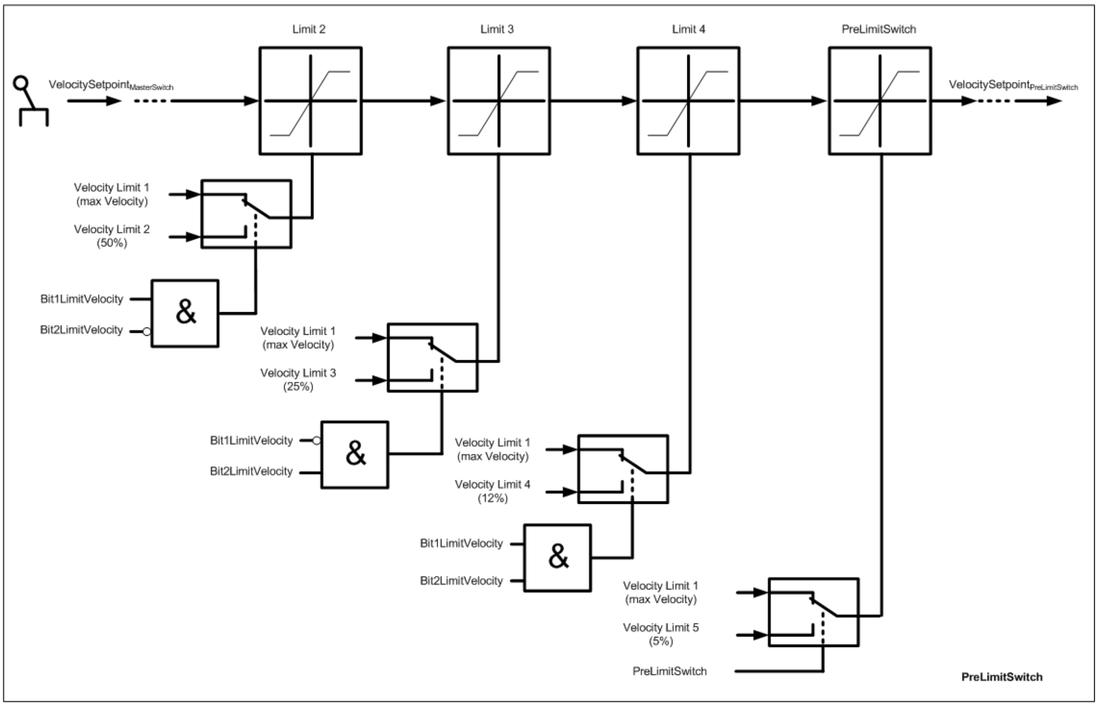


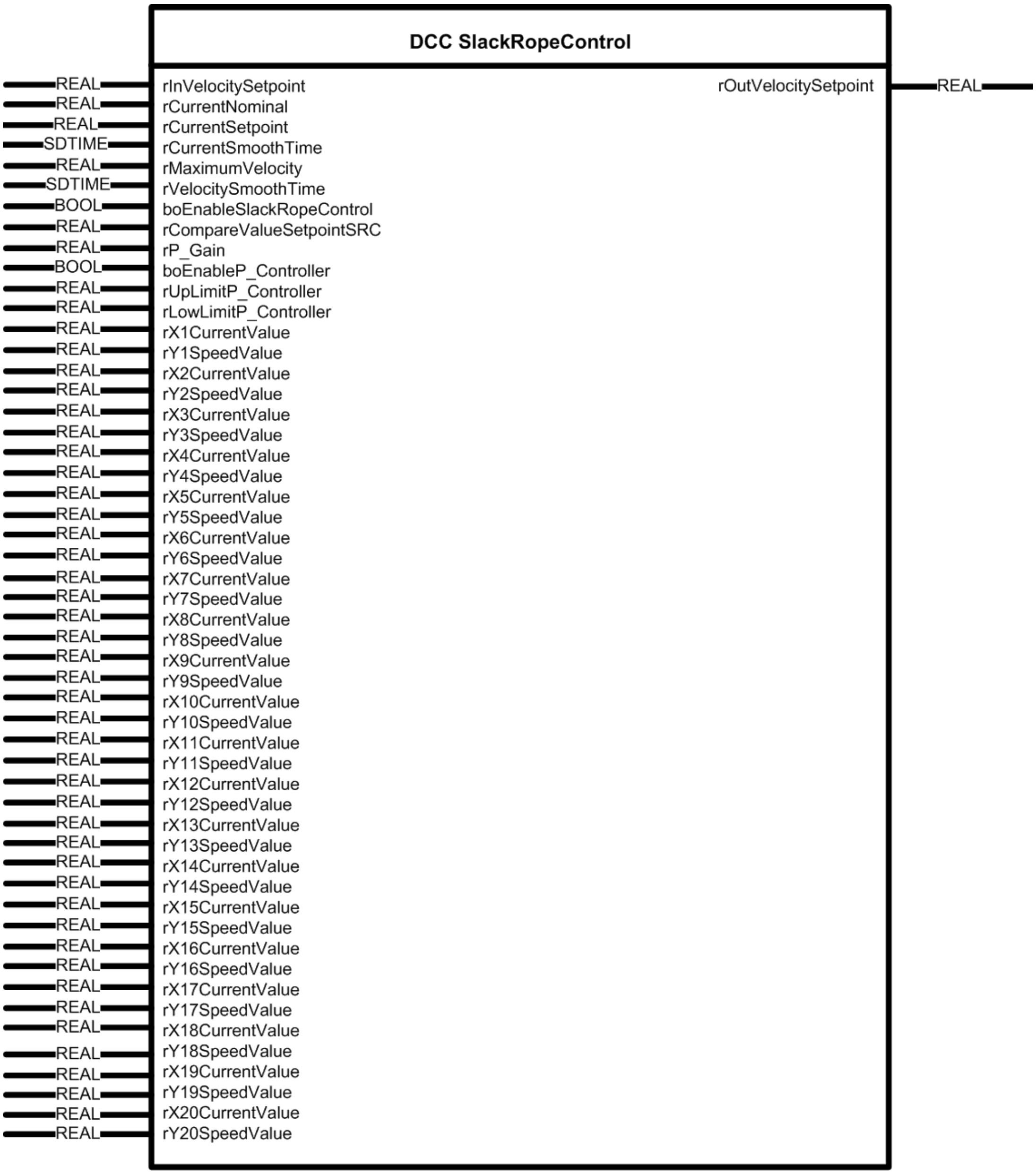
Figure 5-17 Method of operation of the DCC_PreLimitSwitch

Connections

Name	Connection type	Default setting	Data type	Meaning
rInVelocitySetpointPLS	IN	DCC	REAL	Velocity input value [mm/s]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity [mm/s]
boBit1LimitVelocity	IN	MCC	BOOL	Select velocity limit, bit 1
boBit2LimitVelocity	IN	MCC	BOOL	Selects, velocity limit bit 2
boPreLimitSwitch	IN	MCC	BOOL	Selects prelimit switch limiting
rLimit1	IN	100.0	REAL	Velocity limit 1. This limit should be set as maximum velocity [%].
rLimit2	IN	50.0	REAL	Velocity limit 2 [%]
rLimit3	IN	25.0	REAL	Velocity limit 3 [%]
rLimit4	IN	12.0	REAL	Velocity limit 4 [%]
rLimit5	IN	5.0	REAL	Velocity limit 5 [%] (prelimit)
rOutVelocitySetpointPLS	OUT	MCC DCC	REAL	Velocity output value [mm/s]
rOutPositiveMaxVelPLS	OUT	0.0	REAL	Active positive velocity [mm/s]
rOutNegativeMaxVelPLS	OUT	0.0	REAL	Active negative velocity [mm/s]

5.2.18 DCC_SlackRopeControl

Symbol



Brief description

The DCC block DCC_SlackRopeControl is needed by the holding gear to tighten a slack rope if required.

Mode of operation

If the holding gear brake is simply closed when the grab closes, the grab cannot dig into the material to be lifted and the bucket cannot fill sufficiently.

On the other hand, if the brake on the holding gear (holding winch) is simply released when the grab closes, there is a risk that the taut rope will cause the drum on the holding gear to accelerate in the lower direction and cause and undesirable slack rope.

The slack rope control prevents both these problems. The slack rope controller can prevent slackness from developing in the rope when the grab is closed. This function also ensures that the grab can bury itself into the material – therefore ensuring the maximum filling level.

The speed setpoint for the holding gear is input as the closing gear begins to close by means of a slack-rope block which specifies the speed characteristics as illustrated in the diagram below.

When the grab closes, the brake is opened and the holding rope is held with slight tension by the slack rope control. Any rope slack is coiled in at slow speed. The holding gear drum then remains under a residual tension, which is capable of keeping the rope tight enough to prevent rope slack, but not so tight that the grab cannot bury itself into the material to be moved. The grab submerges with the rope held under tension. This residual torque is adjustable and must at least compensate for the weight of the longest holding rope that is used.

If for any reason the grab starts to close in the air, the slack rope control also prevents the grab from sagging quickly at an uncontrolled speed. In this instance, the grab attempts to accelerate the holding gear in a vertical direction by its own weight. However, this speed is likewise limited to a maximum sag speed (characteristic in diagram below) by the slack rope control.

The DCC block SlackRopeControl is not a classic closed-loop control with setpoint/actual value comparison. Instead, it is possible to choose between closed-loop control based on a comparison value (for P controller, see note after connection table) or a stored curve (polygon).

Two slack rope control options:

1. P controller active (boEnableP_Controller = True):
   A current setpoint [%] can be entered for the P controller via the "rCompareValueSetpointSRC" input. This is compared to input "rCurrentSetpoint" and amplified according to the controller gain input "rP_Gain". This gain can be smoothed via "rVelocitySmoothTime".

2. Characteristic active (boEnableP_Controller = False):
   20 interpolation points are available for the characteristic. The input "rCurrentSetpoint" is controlled by the characteristic, as a result of which a speed setpoint that is dependent on the actual current value (starting at instant "Close grab") is specified. This speed setpoint can be smoothed via "rVelocitySmoothTime".

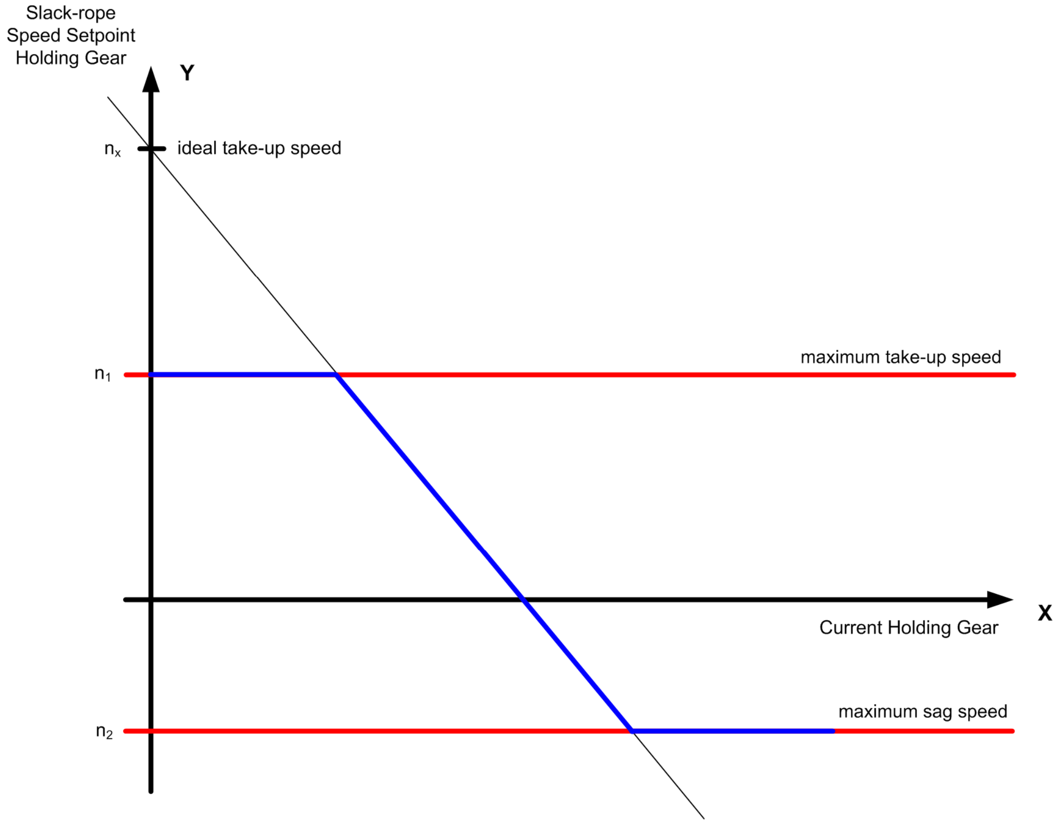


Figure 5-18 Characteristic of the slack rope controller; the blue "curve" represents the ideal curve of the holding gear speed.

NOTICE

The setpoint from the slack rope controller is transferred as a supplementary speed setpoint from SIMOTION to SINAMICS via the same process word used for the start pulse.

The two blocks DCC_StartPulse and DCC_SlackRopeControl are used for the holding gear. As both DCC blocks act on the common supplementary speed setpoint in SINAMICS, the output of "DCC_StartPulse" must be connected to the input "rInVelocitySetpoint". For technical reasons, these two DCC blocks must not be used or activated simultaneously.

A setpoint applied at input "rInVelocitySetpoint" is switched through directly to output "rOutVelocitySetpoint" as long as input "boEnableSlackRopeControl" is not set. In this case, the slack rope controller is disabled and does not specify a setpoint.

A setpoint applied at input "rInVelocitySetpoint" is not switched through directly to output "rOutVelocitySetpoint" as long as input "boEnableSlackRopeControl" is set. In this case, the slack rope controller is enabled and does specify a setpoint.

The ramp-function generator enable (control word 1, bit 4) must be reset before the slack rope controller is activated.

⚠WARNING

Ensure that you enter the correct values for the polygon parameters! The input parameters "rX1CurrentValue" to "rX20CurrentValue" must be assigned in ascending sequence. Proper operation of the block cannot be guaranteed if you do not set the parameters correctly.

NOTICE

Enter the maximum current in [A] at input "rCurrentNominal" as fixed value, or interconnect the maximum current with this input in the user program (refer to the SINAMICS parameter p2002).

Enter the maximum velocity in [mm/s] at input "rMaximumVelocity" as fixed value, or interconnect the maximum velocity with this input in the user program (e.g. refer to Technology Object: D4x5.HoldingGear_1.TypeOfAxis.MaxVelocity.maximum or function block FB_OperationMode output variable r64maximumVelocity).
The block will not work correctly otherwise.

Connections

Name	Connection type	Default setting	Data type	Meaning
rInVelocitySetpoint	IN	DCC	REAL	Setpoint velocity [mm/s]
rCurrentNominal	IN	MCC	REAL	Reference current [A]
rCurrentSetpoint	IN	MCC	REAL	Current setpoint [A]
rCurrentSmoothTime	IN	10	SDTIME	Smoothing time for the current [ms]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity (base velocity + field-weakening speed) [mm/s]
rVelocitySmoothTime	IN	20	SDTIME	Smoothing time for the setpoint velocity [ms]
boEnableSlackRopeControl	IN	MCC	BOOL	Enable slack rope controller FALSE: rOutVelocitySetpoint = rInVelocitySetpoint TRUE: rOutVelocitySetpoint = setpoint velocity of the slack rope controller
rCompareValueSetpointSRC	IN	2.0	REAL	Comparison value for the P controller (current setpoint) [%] relative to rCurrentNominal
rP_Gain	IN	0.0	REAL	Gain factor for the P controller
boEnableP_Controller	IN	False	BOOL	P controller enable instead of polygon enable TRUE: P controller active FALSE: Characteristic active
rUpLimitP_Controller	IN	30.0	REAL	Upper speed limiting value [%] for the P controller output. The value must lie between 0 and 100%.
rLowLimitP_Controller	IN	-30.0	REAL	Lower speed limiting value [%] for the P controller output. The value must lie between 0 and -100%.
rX1CurrentValue	IN	0.0	REAL	Polygon current value X1 [%] of rCurrentNominal
rY1SpeedValue	IN	0.0	REAL	Polygon velocity Y1 [%] of rMaximumVelocity
rX2CurrentValue	IN	0.0	REAL	Polygon current value X2 [%] of rCurrentNominal
rY2SpeedValue	IN	0.0	REAL	Polygon velocity Y2 [%] of rMaximumVelocity
rX3CurrentValue	IN	0.0	REAL	Polygon current value X3 [%] of rCurrentNominal
rY3SpeedValue	IN	0.0	REAL	Polygon velocity Y3 [%] of rMaximumVelocity
rX4CurrentValue	IN	0.0	REAL	Polygon current value X4 [%] of rCurrentNominal
rY4SpeedValue	IN	0.0	REAL	Polygon velocity Y4 [%] of rMaximumVelocity
rX5CurrentValue	IN	0.0	REAL	Polygon current value X5 [%] of rCurrentNominal

Name	Connection type	Default setting	Data type	Meaning
rY5SpeedValue	IN	0.0	REAL	Polygon velocity Y5 [%] of rMaximumVelocity
rX6CurrentValue	IN	0.0	REAL	Polygon current value X6 [%] of rCurrentNominal
rY6SpeedValue	IN	0.0	REAL	Polygon velocity Y6 [%] of rMaximumVelocity
rX7CurrentValue	IN	0.0	REAL	Polygon current value X7 [%] of rCurrentNominal
rY7SpeedValue	IN	0.0	REAL	Polygon velocity Y7 [%] of rMaximumVelocity
rX8CurrentValue	IN	0.0	REAL	Polygon current value X8 [%] of rCurrentNominal
rY8SpeedValue	IN	0.0	REAL	Polygon velocity Y8 [%] of rMaximumVelocity
rX9CurrentValue	IN	0.0	REAL	Polygon current value X9 [%] of rCurrentNominal
rY9SpeedValue	IN	0.0	REAL	Polygon velocity Y9 [%] of rMaximumVelocity
rX10CurrentValue	IN	0.0	REAL	Polygon current value X10 [%] of rCurrentNominal
rY10SpeedValue	IN	0.0	REAL	Polygon velocity Y10 [%] of rMaximumVelocity
rX11CurrentValue	IN	0.0	REAL	Polygon current value X11 [%] of rCurrentNominal
rY11SpeedValue	IN	0.0	REAL	Polygon velocity Y11 [%] of rMaximumVelocity
rX12CurrentValue	IN	0.0	REAL	Polygon current value X12 [%] of rCurrentNominal
rY12SpeedValue	IN	0.0	REAL	Polygon velocity Y12 [%] of rMaximumVelocity
rX13CurrentValue	IN	0.0	REAL	Polygon current value X13 [%] of rCurrentNominal
rY13SpeedValue	IN	0.0	REAL	Polygon velocity Y13 [%] of rMaximumVelocity
rX14CurrentValue	IN	0.0	REAL	Polygon current value X14 [%] of rCurrentNominal
rY14SpeedValue	IN	0.0	REAL	Polygon velocity Y14 [%] of rMaximumVelocity
rX15CurrentValue	IN	0.0	REAL	Polygon current value X15 [%] of rCurrentNominal
rY15SpeedValue	IN	0.0	REAL	Polygon velocity Y15 [%] of rMaximumVelocity
rX16CurrentValue	IN	0.0	REAL	Polygon current value X16 [%] of rCurrentNominal
rY16SpeedValue	IN	0.0	REAL	Polygon velocity Y16 [%] of rMaximumVelocity

Name	Connection type	Default setting	Data type	Meaning
rX17CurrentValue	IN	0.0	REAL	Polygon current value X17 [%] of rCurrentNominal
rY17SpeedValue	IN	0.0	REAL	Polygon velocity Y17 [%] of rMaximumVelocity
rX18CurrentValue	IN	0.0	REAL	Polygon current value X18 [%] of rCurrentNominal
rY18SpeedValue	IN	0.0	REAL	Polygon velocity Y18 [%] of rMaximumVelocity
rX19CurrentValue	IN	0.0	REAL	Polygon current value X19 [%] of rCurrentNominal
rY19SpeedValue	IN	0.0	REAL	Polygon velocity Y19 [%] of rMaximumVelocity
rX20CurrentValue	IN	0.0	REAL	Polygon current value X20 [%] of rCurrentNominal
rY20SpeedValue	IN	0.0	REAL	Polygon velocity Y20 [%] of rMaximumVelocity
rOutVelocitySetpoint	OUT	MCC	REAL	Setpoint velocity of the slack rope controller [mm/s]

Note

The P controller can be used via input "boEnableP_Controller" instead of the value specified via the polygon. Input "boEnableSlackRopeControl" and input "boEnableP_Controller" must both be set to True before this function can be enabled.

The polygon function and the P controller can never be active at the same time.

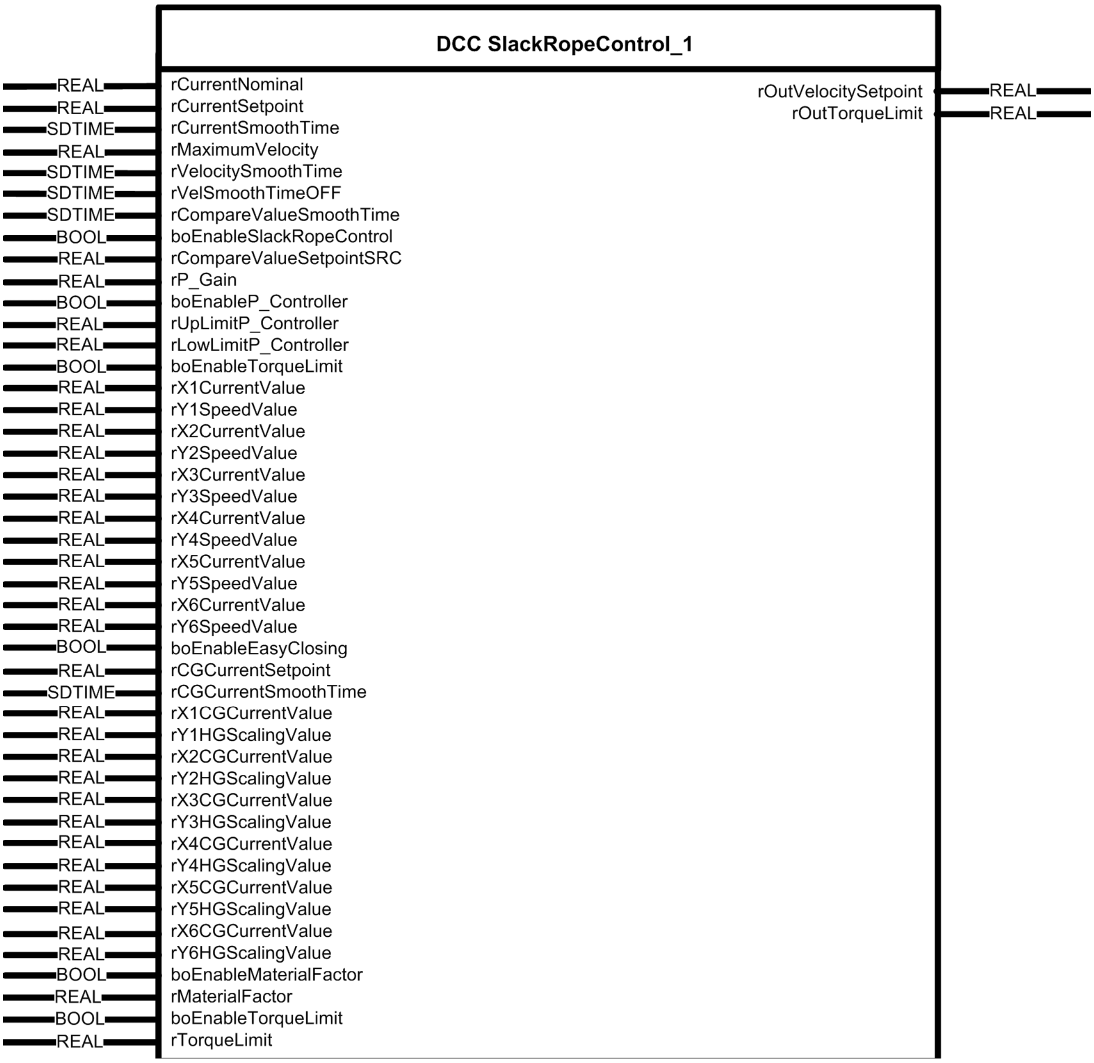
Note

All input values (rX1CurrentValue to rX20CurrentValue) are relative to the input value of "rCurrentNominal".

All input values (rY1SpeedValue to rY20SpeedValue) are relative to the input value of "rMaximumVelocity".

5.2.19 DCC_SlackRopeControl_1

Symbol



Brief description

The block DCC_SlackRopeControl_1 has a 6-point polygon. This evaluates the holding current and allows a slack rope to be avoided. In addition, the block offers three options of the slack rope control, the EasyClosing function and adaptation to several types of material.

Mode of operation

The mode of operation of the DCC_SlackRopeControl_1 block essentially corresponds to the mode of operation of the DCC_SlackRopeControl block (refer to Chapter DCC_SlackRopeControl (Page 137)). The 20-point polygon DCC_SlackRopeControl block for speed characteristic is however simplified for DCC_SlackRopeControl_1 to a 6-point polygon.

In addition, the "rInVelocitySetpoint" input is removed. This input was originally added to switch through from the setpoint channel of the start pulse. However, in the future, the setpoint channel will be externally connected in parallel; refer to the standard application software. In addition, the "rVelocitySmoothTimeOff" and "rCompareValueSmoothTime" inputs have been added.

The DCC_SlackRopeControl_1 block does not concern a classic closed-loop control with setpoint-actual value comparison, but includes the three options that can be selected:

- Closed-loop control using a setpoint value (P controller)
- Closed-loop control using a saved characteristic (polygon)
- Specifying a torque limit (from S7)

Three slack rope control options:

1. The P control (boEnableP_Controller = True) operates as follows:

   A current setpoint [%] can be entered for the P controller via the "rCompareValueSetpointSRC" input. This is compared to the input "rCurrentSetpoint" and amplified according to the controller gain input "rP_Gain". This gain can be smoothed via "rVelocitySmoothTime".

2. The characteristic (boEnableP_Controller = False) operates as follows:

   Six interpolation points are available for the characteristic. The input "rCurrentSetpoint" is controlled by the characteristic, as a result of which a speed setpoint that is dependent on the actual current value (starting at instant "Close grab") is specified. This speed setpoint can be smoothed via "rVelocitySmoothTime".

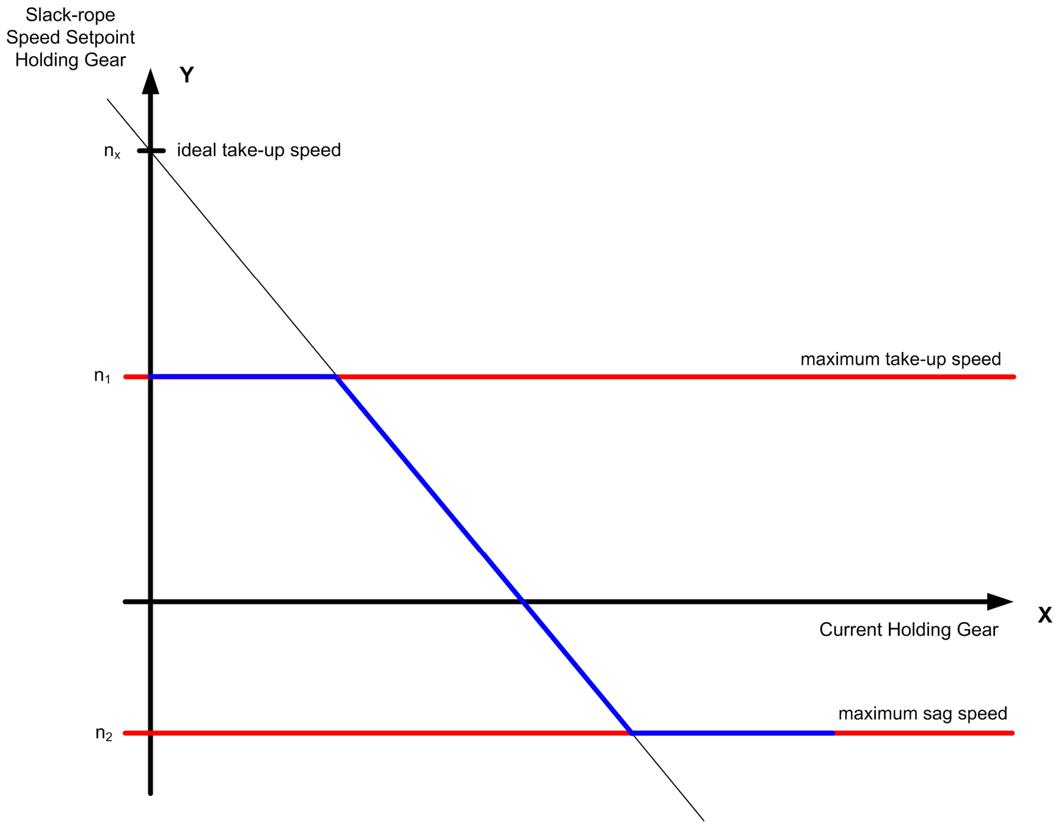


Figure 5-19 Characteristic of the slack rope controller; the blue "curve" represents the ideal curve of the holding gear speed.

In order to avoid a setpoint step (jump) when activating/deactivating the slack rope control from the two methods described above, a PT1 filter "rSmoothTimeOutput" is implemented after the slack rope control or before the signal output.

3. Using the input variable "boEnableTorqueLimit", in addition, the drive torque limits can be connected from the higher-level control. These are entered via "rTorqueLimit". Besides, as part of the user-engineered application software, the message "Motor blocked" (p2144) must be deactivated using a bit in the drive. The "rOutTorqueLimit" output is interconnected as a percentage to the torque limits of the drive.

Easy Closing function

When the grab is overfilled, the grab can be prevented from sinking into the material during closing by either permanently activating the Easy Closing function or via the S7.

An additional 6-point polygon is made available to allow hoisting as a function of the closing current. This function is enabled using "boEnableEasyClosing".

Adapting to material or grab types

An additional evaluation at the slack rope controller output is implemented for various material types and grab types. A multiplier with the "rMaterialFactor" is enabled when "boEnableMaterialFactor" is enabled. The S7 enters the material factor into the SIMOTION - e.g. via PZD13.

NOTICE

The ramp-function generator enable (control word 1, bit 4) must be reset before the slack rope controller is activated.

⚠WARNING

Ensure that you enter the correct values for the polygon parameters! Proper operation of the block cannot be guaranteed if you do not set the parameters correctly.

NOTICE

Enter the maximum current in unit [A] at input "rCurrentNominal" as fixed value, or interconnect the maximum current with this input in the user program (refer to the SINAMICS parameter p2002).

Enter the maximum velocity in unit [mm/s] at input "rMaximumVelocity" as fixed value, or interconnect the maximum velocity with this input in the user program (e.g. refer to Technology Object: D4x5.HoldingGear_1.TypeOfAxis.MaxVelocity.maximum or function block FB_OperationMode output variable r64maximumVelocity).

The block does not work correctly otherwise.

Connections

Name	Connection type	Default setting	Data type	Meaning
rCurrentNominal	IN	MCC	REAL	Reference current [A]
rCurrentSetpoint	IN	MCC	REAL	Current setpoint for holding gear [A]
rCurrentSmoothTime	IN	10	SDTIME	Smoothing time for current, holding gear [ms]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity (base velocity + field-weakening speed) [mm/s]
rVelocitySmoothTime	IN	10	SDTIME	Smoothing time for the setpoint velocity [ms]
rVelSmoothTimeOff	IN	500	SDTIME	Smoothing time during switch-off [ms]
rCompareValueSmoothTime	IN	200	SDTIME	Smoothing time of the setpoint or comparison value [ms]
boEnableSlackRopeControl	IN	MCC	BOOL	Enable slack rope controller
rCompareValueSetpointSRC	IN	2.0	REAL	Comparison value for the P controller (current setpoint) [%] relative to rCurrentNominal
rP_Gain	IN	0.5	REAL	Gain factor for the P controller

Name	Connection type	Default setting	Data type	Meaning
boEnableP_Controller	IN	False	BOOL	P controller enable instead of polygon enable TRUE: P controller active FALSE: Characteristic active
rUpLimitP_Controller	IN	30.0	REAL	Upper speed limiting value [%] for the P controller output. The value must lie between 0 and 100%.
rLowLimitP_Controller	IN	-30.0	REAL	Lower speed limiting value [%] for the P controller output. The value must lie between 0 and -100%.
rX1CurrentValue	IN	-20	REAL	Polygon current value X1 [%] of rCurrentNominal
rY1SpeedValue	IN	10	REAL	Polygon velocity Y1 [%] of rMaximumVelocity
rX2CurrentValue	IN	10	REAL	Polygon current value X2 [%] of rCurrentNominal
rY2SpeedValue	IN	5	REAL	Polygon velocity Y2 [%] of rMaximumVelocity
rX3CurrentValue	IN	-5	REAL	Polygon current value X3 [%] of rCurrentNominal
rY3SpeedValue	IN	2	REAL	Polygon velocity Y3 [%] of rMaximumVelocity
rX4CurrentValue	IN	5	REAL	Polygon current value X4 [%] of rCurrentNominal
rY4SpeedValue	IN	-2	REAL	Polygon velocity Y4 [%] of rMaximumVelocity
rX5CurrentValue	IN	10	REAL	Polygon current value X5 [%] of rCurrentNominal
rY5SpeedValue	IN	-5	REAL	Polygon velocity Y5 [%] of rMaximumVelocity
rX6CurrentValue	IN	20	REAL	Polygon current value X6 [%] of rCurrentNominal
rY6SpeedValue	IN	-10	REAL	Polygon velocity Y6 [%] of rMaximumVelocity
boEnableEasyClosing	IN	FALSE	REAL	Enable signal for the EasyClosing function
rCGCurrentSetpoint	IN	MCC	REAL	Current setpoint, closing gear [A]
rCGCurrentSmoothTime	IN	0.0	SDTIME	Smoothing time for current, closing gear [ms]
rX1CGCurrentValue	IN	60	REAL	Polygon, current closing gear X1 [%]
rY1HGScalingValue	IN	100	REAL	Polygon for holding gear supplementary speed setpoint Y1 [%]
rX2CGCurrentValue	IN	70	REAL	Polygon, current closing gear X2 [%]
rY2HGScalingValue	IN	100	REAL	Polygon for holding gear supplementary speed setpoint Y2 [%]
rX3CGCurrentValue	IN	80	REAL	Polygon, current closing gear X3 [%]
rY3HGScalingValue	IN	105	REAL	Polygon for holding gear supplementary speed setpoint Y3 [%]

Name	Connection type	Default setting	Data type	Meaning
rX4CGCurrentValue	IN	90	REAL	Polygon, current closing gear X4 [%]
rY4HGScalingValue	IN	110	REAL	Polygon for holding gear supplementary speed setpoint Y4 [%]
rX5CGCurrentValue	IN	100	REAL	Polygon, current closing gear X5 [%]
rY5HGScalingValue	IN	120	REAL	Polygon for holding gear supplementary speed setpoint Y5 [%]
rX6CGCurrentValue	IN	120	REAL	Polygon, current closing gear X6 [%]
rY6HGScalingValue	IN	150	REAL	Polygon for holding gear supplementary speed setpoint Y6 [%]
boEnableMaterialFactor	IN	FALSE	BOOL	Enable signal for material factor function
rMaterialFactor	IN	100.0	REAL	Material assessment factor [%]
boEnableTorqueLimit	IN	FALSE	BOOL	Enable of the torque limits in the drive
rTorqueLimit	IN	MCC	REAL	Torque for the torque limits in the drive
rOutTorqueLimit	OUT	1.0	REAL	Torque for the torque limits in the drive [%]
rOutVelocitySetpoint	OUT	MCC	REAL	Setpoint velocity of the slack rope controller [mm/s]

Note

The "rVelSmoothTimeOnOff" input is a PT1 element that applies only during switch-on and off. When switching off, the inverse function of the PT 1 element is used so there is no jump for either switch-on or switch-off. The time value entered in the "rVelSmoothTimeOnOff" corresponds to 1 Tau (1 τ).

Note

The P controller can be used via input "boEnableP_Controller" instead of the value specified via the polygon. Input "boEnableSlackRopeControl" and input "boEnableP_Controller" must both be set to True before this function can be enabled.

The polygon is then not effective. Either the polygon function, the P controller or the torque limit input is active at any one time - never together.

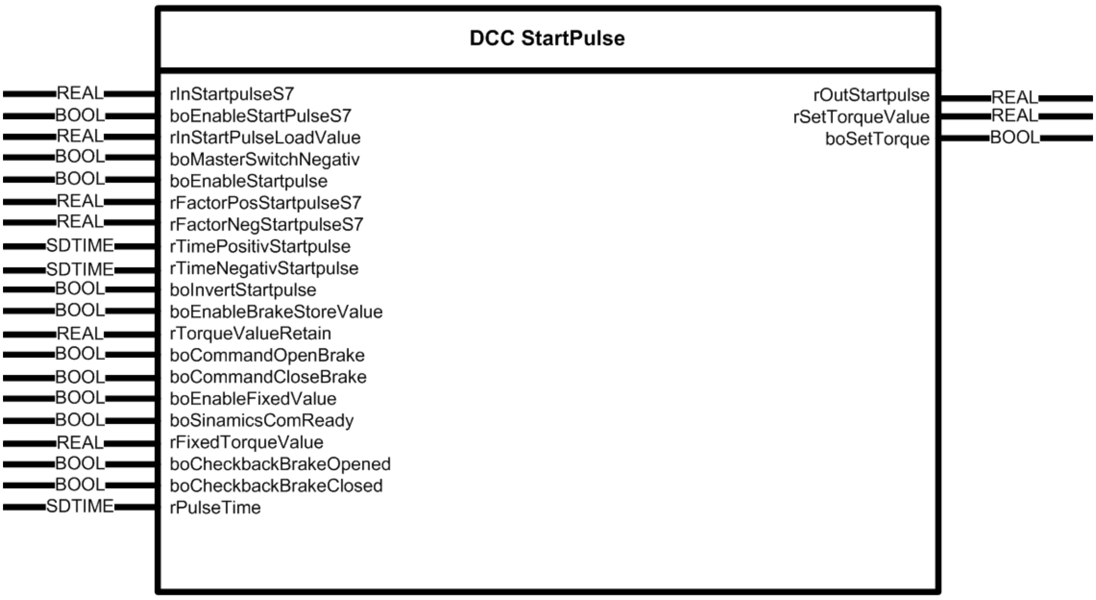
Note

All input values (rX1HGCurrentValue to rX6HGCurrentValue or rX1CGCurrentValue to rX6CGCurrentValue) are referred to the input value of "rCurrentNominal".

All input values (rY1HGSpeedValue to rY6HGSpeedValue) are referred to the input value of "rMaximumVelocity".

5.2.20 DCC_StartPulse

Symbol



Brief description

Using the block DCC_StartPulse, "load sag" can either be prevented or reduced when starting hoists with suspended load.

Mode of operation

For hoists, when starting (i.e. opening the hoisting gear brake) with freely suspended load, often the load undesirably sags. The reason for this is that the torque is not available when starting; the torque is relatively slowly established once the travel command has been output as a result of the processing time of the systems, the time constants of the control (closed-loop) and the time constants of the control loop. However, when starting with a suspended load, the torque must be quickly established. There are two methods for the start pulse.

1. Method

In order to prevent the load sagging, a start pulse that decays after a defined time is applied as supplementary speed in front of the speed controller. This is done in the hoisting gear when starting and when the hoist or lower command is enabled. The characteristic of this start pulse is that of an inverse e function that is "discharged" when enabled. This start pulse can be constant (via "rInStartPulseLoadValue") or load-dependent for the S7 (via "rInStartPulseS7 ") as supplementary speed in front of the speed controller that decays after a defined time is applied here.

The start pulse quickly establishes the torque and so prevents load sag. The direction of the start pulse is always in the hoisting direction.

The start pulse is optimized experimentally using pre-optimized speed control for the subsequent trial with load. When starting from zero speed with an attached, freely-suspended load, the control lever is set for a minimum speed setpoint and then the starting behavior of the drive is monitored at the hook or at the rope drum.

When setting a start pulse with a constant load, you must make a compromise between the start with rated load and the start with empty hook; the reason for this is that the set, constant value is only optimally set for a single load. Then, for other loads, either the load will slightly sag or the empty hook will move in the hoisting direction.

2. Method

- To save the I component of the speed controller when the "Close brake" command is issued.
- To set the I component when the "Release brake" command is issued.

  The close or open brake is indicated using the AppSTW2 bit 0 (save and set the I component (start pulse)).

  – If the brake is closed and the I component should be saved, the bit must change from TRUE to FALSE.

  – If the brake is opened and the I component should be set, the bit must change from FALSE to TRUE;

  see also Figure 5-21 Open and close the brake for DCC_StartPulse_1 (Page 155).

Note

Also refer to the detailed description in Chapter Start pulse (Page 451).

CAUTION

Set the enable of the "boEnableStartpulse" start pulse for only 50 ms.
The "rTimePositivStartpulse" or "rTimeNegativStartpulse" input corresponds to one Tau (1 τ). The length of the start pulse is entered via this input and cannot be specified from the higher-level control.

⚠WARNING

Incorrect start pulse settings may lead to injury and material damage.

⚠WARNING

For method 1 and method 2, the pulse must not be too large or be present too long. The pulse must be set correctly. Otherwise, the IGBTs can be damaged.

Method 2 is preferred.

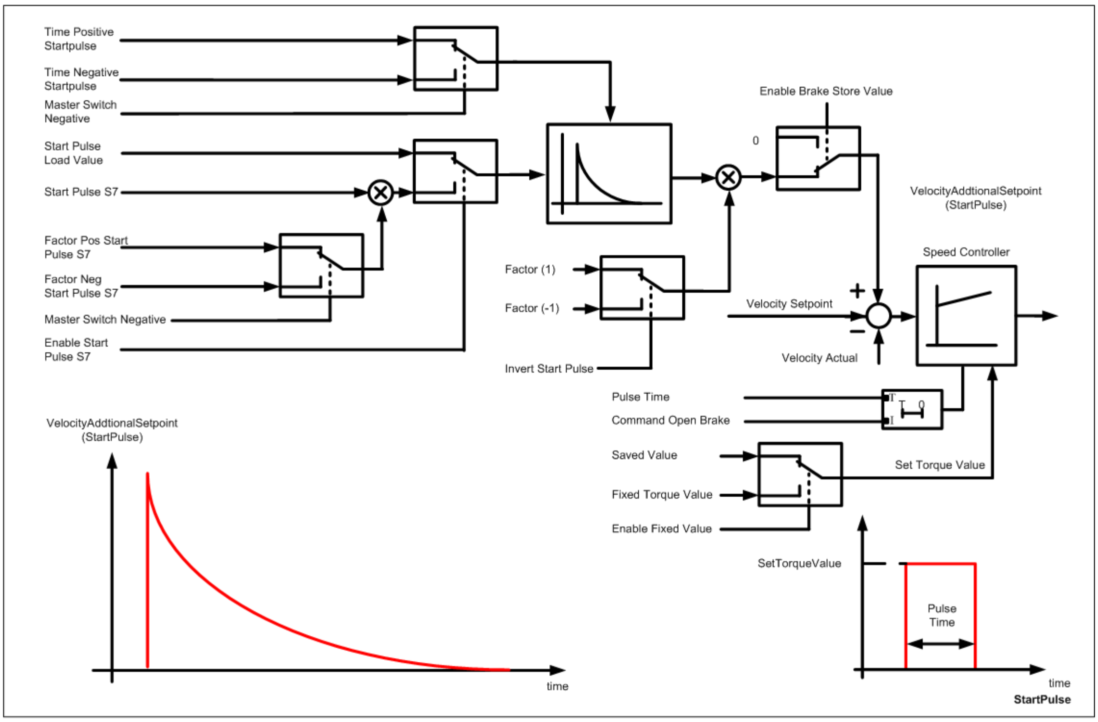


Figure 5-20 Mode of operation DCC_StartPulse

Connections

Name	Connection type	Default setting	Data type	Meaning
rInStartpulseS7	IN	MCC	REAL	Start pulse value from the "S7" controller [mm/s]
boEnableStartpulseS7	IN	False	BOOL	The "S7" controller enables the start pulse value
rInStartpulseLoadValue	IN	0.0	REAL	Configured start pulse value [mm/s]
boMasterSwitchNegativ	IN	MCC	BOOL	Deflection signal of the master switch in the negative direction
boEnableStartpulse	IN	MCC	BOOL	Enable of the start pulse
rFactorPosStartpulseS7	IN	100.0	REAL	Evaluation factor [%] of the start pulse in the positive direction when the "S7" controller issues the start pulse
rFactorNegStartpulseS7	IN	100.0	REAL	Evaluation factor [%] of the start pulse in the negative direction when the "S7" controller issues the start pulse
rTimePositivStartpulse	IN	500	SDTIME	Decay time of the start pulse in the positive direction [ms]
rTimeNegativStartpulse	IN	500	SDTIME	Decay time of the start pulse in the negative direction [ms]
boInvertStartpulse	IN	False	BOOL	Inverts the start pulse
boEnableBrakeStoreValue	IN	False	BOOL	Activates the torque setpoint from the speed controller
rTorqueValueRetain	IN	MCC	REAL	I component of the speed controller

Name	Connection type	Default setting	Data type	Meaning
boCommandOpenBrake	IN	False	BOOL	"Release brake" signal
boCommandCloseBrake	IN	False	BOOL	"Close brake" signal
boEnableFixedValue	IN	False	BOOL	Enables the fixed torque setpoint instead of the speed controller I component
boSinamicsComReady	IN	MCC	BOOL	SINAMICS is ready to communicate
rFixedTorqueValue	IN	0.0	REAL	Fixed torque setpoint for speed controller I component
boCheckbackBrakeOpened	IN	False	BOOL	Checkback signal "Brake released"
boCheckBackbrakeClosed	IN	False	BOOL	Checkback signal "Brake closed"
rPulseTime	IN	0	SDTIME	Longest application period for the boSetTorque output signal
rOutStartpulse	OUT	MCC	REAL	Start pulse output [mm/s]
rSetTorqueValue	OUT	MCC	REAL	Torque setting value for speed controller I component
boSetTorque	OUT	DCC	BOOL	"Set the rSetTorqueValue as an I component" message

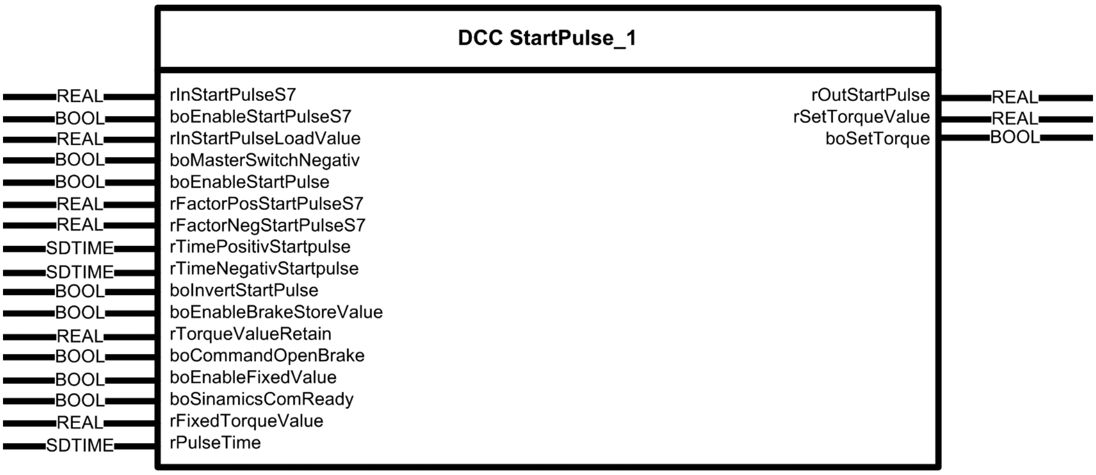
Note

The "rFactorPosStartPulseS7" and "rFactorNegStartPulseS7" input values are limited to max. -200% and +200%.

The "boSinamicsComReady" input must be linked to the "DO-name_boCyclicinterface" variable. If "boSinamicsComReady" = false, the I component of the speed controller cannot be saved.

5.2.21 DCC_StartPulse_1

Symbol



Brief description

This DCC block is functionally identical with the DCC_StartPulse block except that the block inputs have been optimized with regard to the brakes.

Connections

Name	Connection type	Default setting	Data type	Meaning
rInStartpulseS7	IN	MCC	REAL	Start pulse value from the "S7" controller [mm/s]
boEnableStartpulseS7	IN	False	BOOL	The "S7" controller enables the start pulse value
rInStartpulseLoadValue	IN	0.0	REAL	Configured start pulse value [mm/s]
boMasterSwitchNegativ	IN	MCC	BOOL	Deflection signal of the master switch in the negative direction
boEnableStartpulse	IN	MCC	BOOL	Enable of the start pulse
rFactorPosStartpulseS7	IN	100.0	REAL	Evaluation factor [%] of the start pulse in the positive direction when the "S7" controller issues the start pulse
rFactorNegStartpulseS7	IN	100.0	REAL	Evaluation factor [%] of the start pulse in the negative direction when the "S7" controller issues the start pulse
rTimePositivStartpulse	IN	500	SDTIME	Decay time of the start pulse in the positive direction [ms]
rTimeNegativStartpulse	IN	500	SDTIME	Decay time of the start pulse in the negative direction [ms]
boInvertStartpulse	IN	False	BOOL	Inverts the start pulse

Name	Connection type	Default setting	Data type	Meaning
boEnableBrakeStoreValue	IN	False	BOOL	Activates the torque setpoint from the speed controller
rTorqueValueRetain	IN	MCC	REAL	I component of the speed controller
boCommandOpenBrake	IN	False	BOOL	"Release brake" signal
boEnableFixedValue	IN	False	BOOL	Enables the fixed torque setpoint instead of the speed controller I component
boSinamicsComReady	IN	MCC	BOOL	SINAMICS is ready to communicate
rFixedTorqueValue	IN	0.0	REAL	Fixed torque setpoint for speed controller I component
rPulseTime	IN	0	SDTIME	Longest application period for the boSetTorque output signal
rOutStartpulse	OUT	MCC	REAL	Start pulse output [mm/s]
rSetTorqueValue	OUT	MCC	REAL	Torque setting value for speed controller I component
boSetTorque	OUT	DCC	BOOL	"Set the rSetTorqueValue as an I component" message

The following diagram shows a normal sequence for opening and closing the brake with the following method:

- Save the I component of the speed controller for the reset (1→0) of "boCommandOpenBreak" via AppSTW2 bit 0 because the brake is then applied.
- Set the I component for the "open brake" command via AppSTW2 bit 0 (0 → 1).

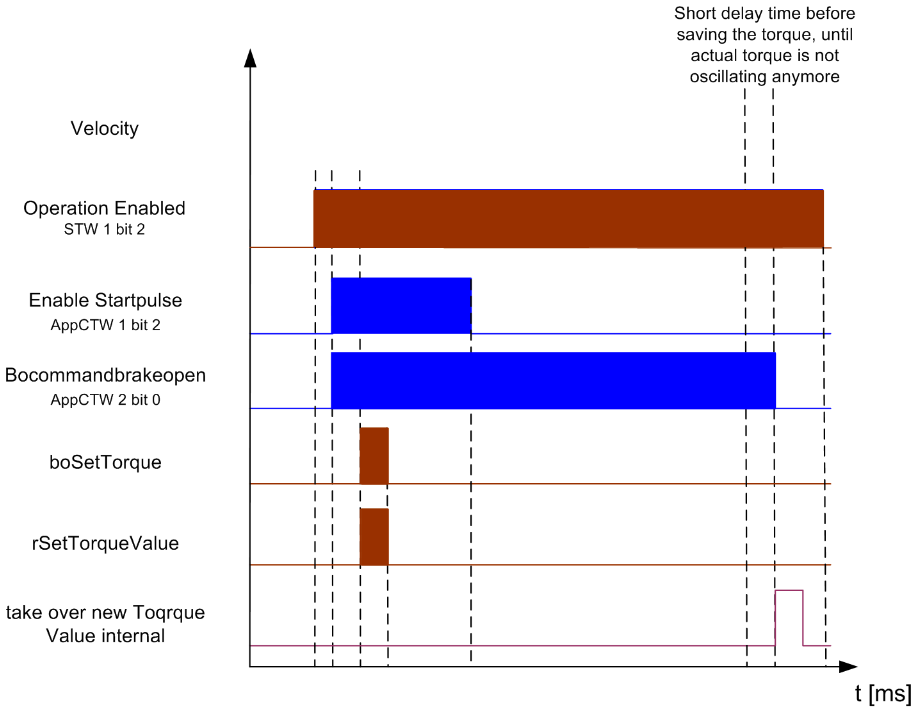
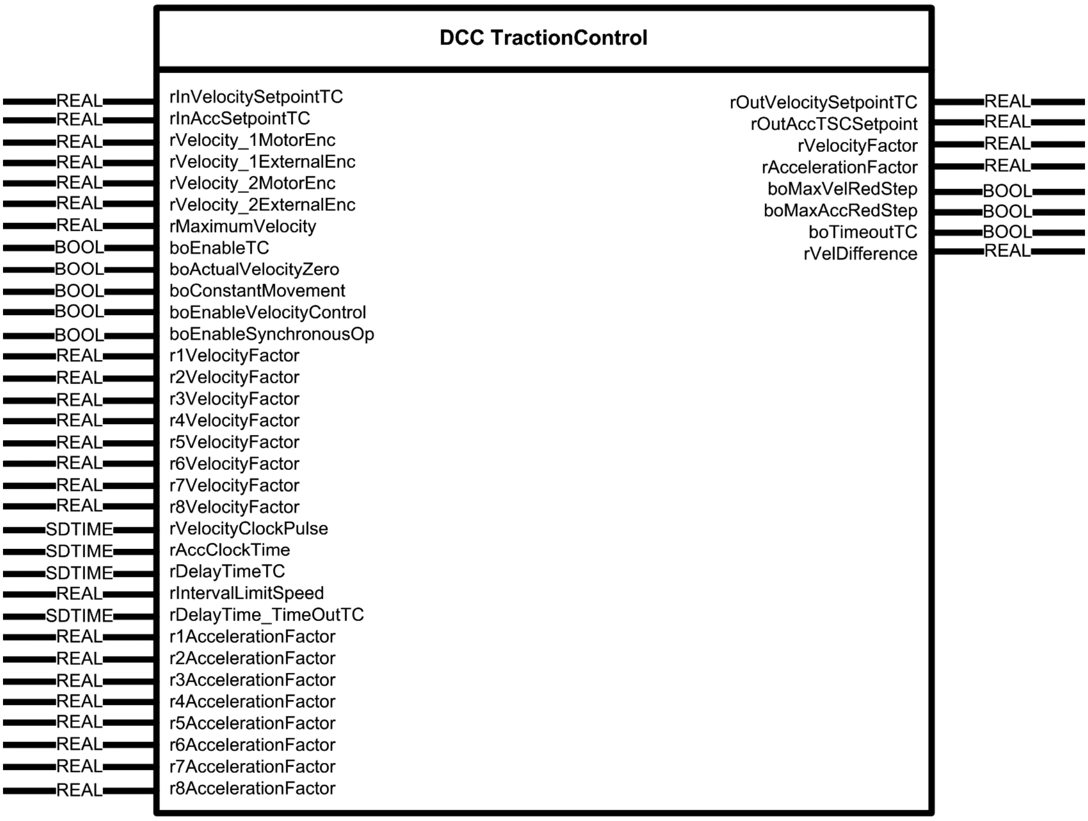


Figure 5-21 Open and close the brake for DCC_StartPulse_1

5.2.22 DCC_TractionControl

Symbol



Short description

The velocity between the motor encoder and the external encoder is monitored using the DCC block DCC_TractionControl. If an excessively high velocity deviation occurs, the velocity of the acceleration is adapted.

Mode of operation

This function is used to compare the velocities from the motor encoder and external encoder. If the velocity difference exceeds the configured velocity threshold, then while accelerating, the acceleration is reduced step-by-step and in the constant velocity phase, the velocity is reduced step-by-step. If the last reduction stage is reached, then this is signaled at the block output.

In the synchronous operation control type, the reduction in acceleration or velocity acts directly on the master only, not directly on the slave.

Note

The acceleration and velocity stages must be configured as follows. Velocity reduction stage 1 should be 100 %; the following should be continually smaller up to the last velocity reduction stage 8; this must be the smallest.

1. r1VelocityFactor = 100%
2. r2VelocityFactor < r1VelocityFactor
3. r3VelocityFactor < r2VelocityFactor
4. r4VelocityFactor < r3VelocityFactor
5. r5VelocityFactor < r4VelocityFactor
6. r6VelocityFactor < r5VelocityFactor
7. r7VelocityFactor < r6VelocityFactor
8. r8VelocityFactor < r7VelocityFactor

Velocity reduction stage 1 should be 100%. The following reduction stages should be continually smaller up to the last acceleration reduction stage 8; this must be the smallest:

1. r1AccelerationFactor = 100%
2. r2AccelerationFactor < r1AccelerationFactor
3. r3AccelerationFactor < r2AccelerationFactor
4. r4AccelerationFactor < r3AccelerationFactor
5. r5AccelerationFactor < r4AccelerationFactor
6. r6AccelerationFactor < r5AccelerationFactor
7. r7AccelerationFactor < r6AccelerationFactor
8. r8AccelerationFactor < r7AccelerationFactor

This deceleration is not reduced.

Note

The pulse width [ms] for the changeover from the acceleration to the deceleration reduction stages must be proportional to twice the defined time.

NOTICE

Enter the maximum velocity in [mm/s] at input "rMaximumVelocity" as fixed value, or interconnect the maximum velocity with this input in the user program (e.g.: refer to Technology Object: D4x5.Gantry_1.TypeOfAxis.MaxVelocity.maximum or function block FB_OperationMode output variable r64maximumVelocity).
The block will not work correctly otherwise.

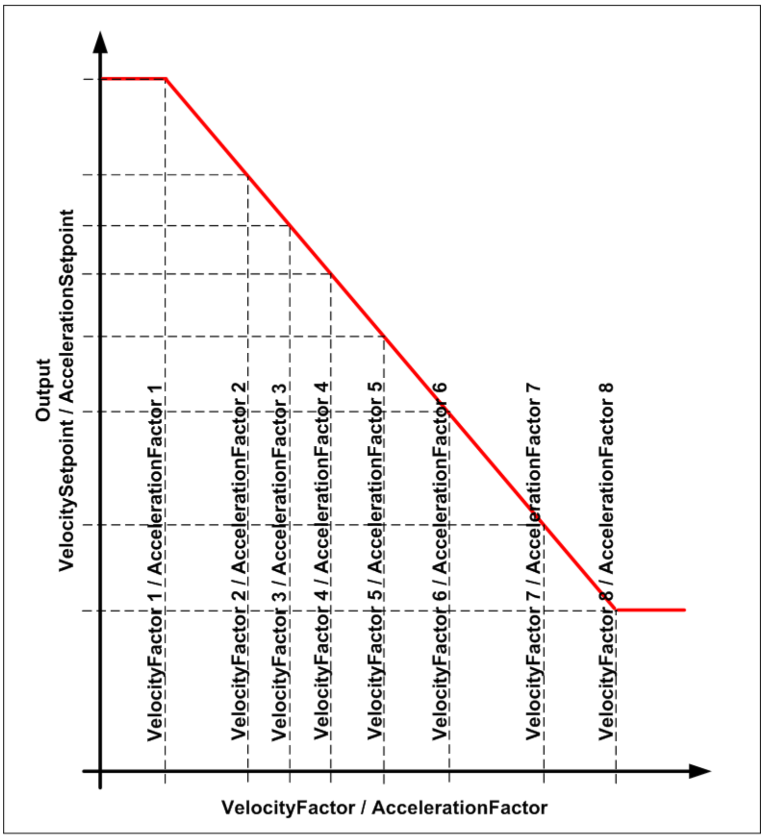


Figure 5-22 Method of operation DCC_TractionControl

Connections

Name	Connection type	Default setting	Data type	Meaning
rInVelocitySetpointTC	IN	DCC	REAL	Velocity input value [mm/s]
rInAccSetpointTC	IN	DCC	REAL	Acceleration input value [mm/s^2]
rVelocity_1MotorEnc	IN	MCC	REAL	Velocity of motor encoder 1 [mm/s]
rVelocity_1ExternalEnc	IN	MCC	REAL	Velocity of external encoder 1 [mm/s]
rVelocity_2MotorEnc	IN	0.0	REAL	Velocity of motor encoder 2 [mm/s]
rVelocity_2ExternalEnc	IN	0.0	REAL	Velocity of external encoder 2 [mm/s]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity [mm/s]
boEnableTC	IN	MCC	BOOL	Enables the traction control

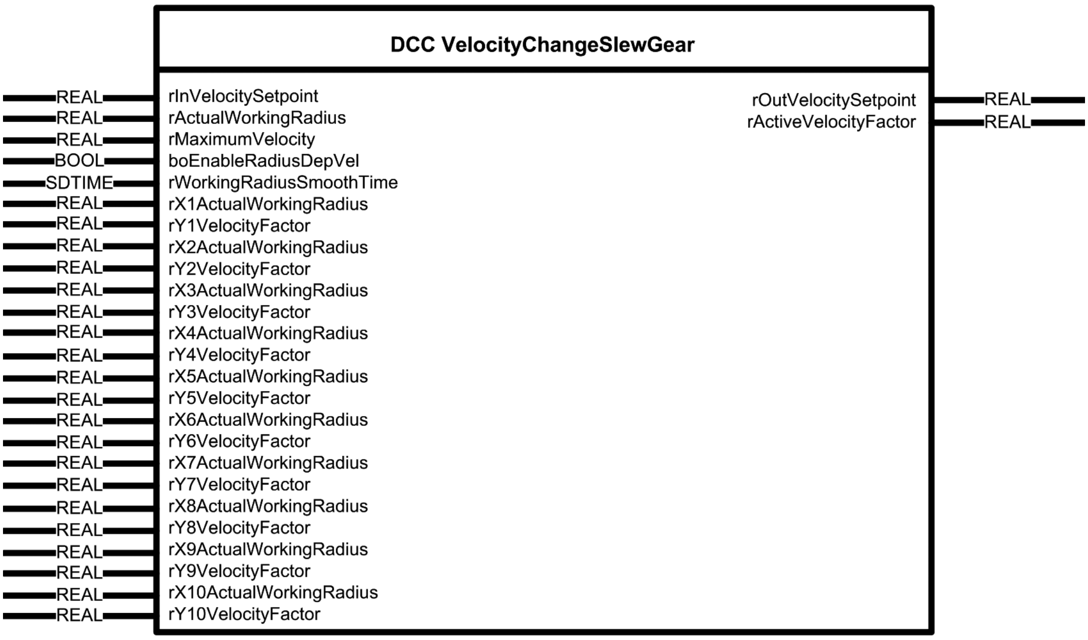
boActualVelocityZero	IN	MCC	BOOL	Drive standstill;
				FALSE = drive moving
				TRUE = Drive at a standstill

Name	Connection type	Default setting	Data type	Meaning
boConstantMovement	IN	MCC	BOOL	Constant movement, that is, the drive is moving at constant velocity; FALSE = drive is at a standstill, accelerating or deceleration TRUE = TRUE = drive moving at constant velocity
boEnableVelocityControl	IN	True	BOOL	Enables the traction monitoring in the constant velocity phase
boEnableSynchronousOp	IN	MCC	BOOL	Synchronous operation control type is active
r1VelocityFactor	IN	100.0	REAL	Velocity reduction stage 1 [%] of rInVelocitySetpointTC
r2VelocityFactor	IN	90.0	REAL	Velocity reduction stage 2 [%] of rInVelocitySetpointTC
r3VelocityFactor	IN	80.0	REAL	Velocity reduction stage 3 [%] of rInVelocitySetpointTC
r4VelocityFactor	IN	65.0	REAL	Velocity reduction stage 4 [%] of rInVelocitySetpointTC
r5VelocityFactor	IN	50.0	REAL	Velocity reduction stage 5 [%] of rInVelocitySetpointTC
r6VelocityFactor	IN	30.0	REAL	Velocity reduction stage 6 [%] of rInVelocitySetpointTC
r7VelocityFactor	IN	20.0	REAL	Velocity reduction stage 7 [%] of rInVelocitySetpointTC
r8VelocityFactor	IN	10.0	REAL	Velocity reduction stage 8 [%] of rInVelocitySetpointTC
rVelocityClockpulse	IN	1000	SDTIME	Pulse length [ms] to change over the velocity reduction stages
rAccClockTime	IN	1000	SDTIME	Pulse length [ms] to change over the acceleration reduction stages
rDelayTimeTC	IN	500	SDTIME	Delay time [ms] for a velocity difference greater than the permissible velocity deviation
rIntervalLimitSpeed	IN	10.0	REAL	Valid velocity difference [%]; useful setting: 10% to 30%
rDelayTime_TimeOutTC	IN	500	SDTIME	Delay time before output "boTimeoutTC" is set. [ms]
r1AccelerationFactor	IN	100.0	REAL	Acceleration reduction stage 1 [%] of rInAccSetpointTC
r2AccelerationFactor	IN	90.0	REAL	Acceleration reduction stage 2 [%] of rInAccSetpointTC
r3AccelerationFactor	IN	80.0	REAL	Acceleration reduction stage 3 [%] of rInAccSetpointTC
r4AccelerationFactor	IN	65.0	REAL	Acceleration reduction stage 4 [%] of rInAccSetpointTC
r5AccelerationFactor	IN	50.0	REAL	Acceleration reduction stage 5 [%] of rInAccSetpointTC
r6AccelerationFactor	IN	30.0	REAL	Acceleration reduction stage 6 [%] of rInAccSetpointTC

Name	Connection type	Default setting	Data type	Meaning
r7AccelerationFactor	IN	20.0	REAL	Acceleration reduction stage 7 [%] of rInAccSetpointTC
r8AccelerationFactor	IN	10.0	REAL	Acceleration reduction stage 8 [%] of rInAccSetpointTC
rOutVelocitySetpointTC	OUT	DCC	REAL	Velocity output value [mm/s]
rOutAccTSCSetpoint	OUT	MCC DCC	REAL	Velocity output value [mm/s^2]
rVelocityFactor	OUT	0.0	REAL	Active velocity reduction factor [%]
rAccelerationFactor	OUT	0.0	REAL	Active acceleration reduction factor [%]
boMaxVelRedStep	OUT	False	BOOL	Maximum velocity reduction stage reached
boMaxAccRedStep	OUT	False	BOOL	Maximum acceleration reduction stage reached
boTimeoutTC	OUT	False	BOOL	Velocity difference greater than the permissible deviation over the configured time (rDelayTime_TimeoutTC)
rVelDifference	OUT	0.0	REAL	Actual velocity difference [mm/s]

5.2.23 DCC_VelocityChangeSlewGear

Symbol


DCC VelocityChangeSlewGear
REAL rInVelocitySetpoint
REAL rActualWorkingRadius
REAL rMaximumVelocity
BOOL boEnableRadiusDepVel
SDTIME rWorkingRadiusSmoothTime
REAL rX1ActualWorkingRadius
REAL rY1VelocityFactor
REAL rX2ActualWorkingRadius
REAL rY2VelocityFactor
REAL rX3ActualWorkingRadius
REAL rY3VelocityFactor
REAL rX4ActualWorkingRadius
REAL rY4VelocityFactor
REAL rX5ActualWorkingRadius
REAL rY5VelocityFactor
REAL rX6ActualWorkingRadius
REAL rY6VelocityFactor
REAL rX7ActualWorkingRadius
REAL rY7VelocityFactor
REAL rX8ActualWorkingRadius
REAL rY8VelocityFactor
REAL rX9ActualWorkingRadius
REAL rY9VelocityFactor
REAL rX10ActualWorkingRadius
REAL rY10VelocityFactor
rOutVelocitySetpoint REAL
rActiveVelocityFactor REAL

Note

For this DCC block, the term "velocity" refers to "angular velocity".

Brief description

The DCC block DCC_VelocityChangeSlewGear requests for slewing gear a peripheral speed for the boom head which is independent of the working radius.

Mode of operation

This function is required to maintain the boom head at a constant peripheral speed on slewable luffing jib cranes. Depending on the working radius (radius r) of the luffing gear, the speed n (angular velocity ω) of the slewing gear is adjusted according to relation ω = v/r.

It is implemented using a hyperbolic function and a polygon.

For this purpose, the setpoint specified via the master switch (peripheral speed) is converted into a speed setpoint (angular velocity) that is dependent on the working radius.

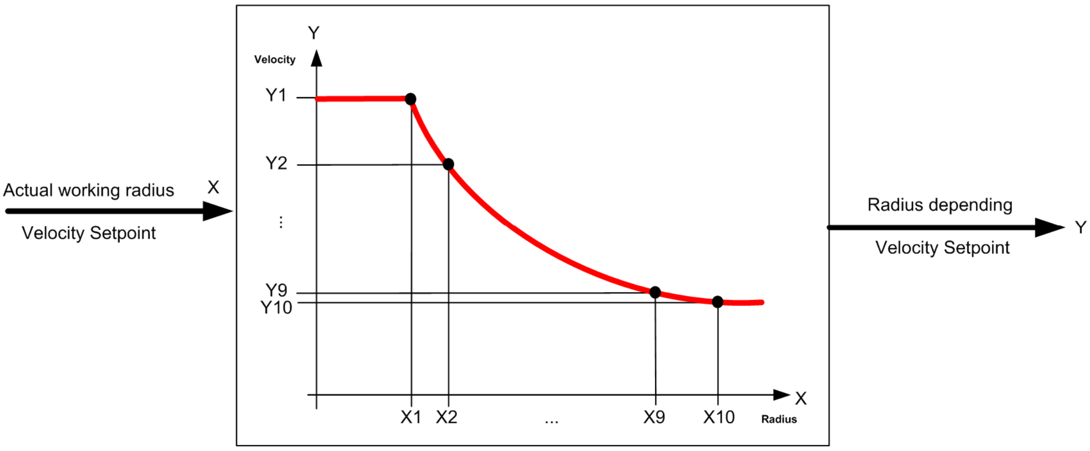


Figure 5-23 Characteristic of rotation speed dependent on working radius

Note

The effective velocity factor is limited to a range of maximum -200% to +200%.

NOTICE

Enter the maximum velocity in [mm/s] at input "rMaximumVelocity" as fixed value, or interconnect the maximum velocity with this input in the user program refer to Technology Object: D4x5.SlewingGear_1.TypeOfAxis.MaxVelocity.maximum or function block FB_OperationMode output variable r64maximumVelocity). The block does not work correctly otherwise.

⚠WARNING

Ensure that you enter the correct values for the polygon parameters! The input parameters "rX1ActualAngle" to "rX10ActualAngle" must be assigned in ascending sequence. Proper operation of the block cannot be guaranteed if you do not set the parameters correctly.

The input parameters "rY1VelocityFactor" to "rY10VelocityFactor" must be assigned in descending sequence.

⚠WARNING

Proper operation of the block cannot be guaranteed if you do not set the parameters correctly.

Connections

Name	Connection type	Default setting	Data type	Meaning
rInVelocitySetpoint	IN	DCC	REAL	Setpoint velocity [mm/s]
rActualWorkingRadius	IN	MCC	REAL	Actual working radius [mm]
rMaximumVelocity	IN	MCC	REAL	Maximum velocity (base velocity + field-weakening speed) [mm/s]
boEnableRadiusDepVel	IN	False	BOOL	Enable velocity dependent on working radius: FALSE: rOutVelocitySetpoint = rInVelocitySetpoint TRUE: rOutVelocitySetpoint = rInVelocitySetpoint x effective velocity factor
rWorkingRadiusSmoothTime	IN	0	SDTIME	Smoothing time for actual working radius [ms]
rX1ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X1 [mm]
rY1VelocityFactor	IN	100.0	REAL	Polygon velocity factor Y1 [%]
rX2ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X2 [mm]
rY2VelocityFactor	IN	95.0	REAL	Polygon velocity factor Y2 [%]
rX3ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X3 [mm]
rY3VelocityFactor	IN	90.0	REAL	Polygon velocity factor Y3 [%]
rX4ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X4 [mm]
rY4VelocityFactor	IN	85.0	REAL	Polygon velocity factor X4 [%]
rX5ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X5 [mm]
rY5VelocityFactor	IN	80.0	REAL	Polygon velocity factor Y5 [%]
rX6ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X6 [mm]
rY6VelocityFactor	IN	75.0	REAL	Polygon velocity factor Y6 [%]
rX7ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X7 [mm]

Name	Connection type	Default setting	Data type	Meaning
rY7VelocityFactor	IN	70.0	REAL	Polygon velocity factor Y7 [%]
rX8ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X8 [mm]
rY8VelocityFactor	IN	65.0	REAL	Polygon velocity factor Y8 [%]
rX9ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X9 [mm]
rY9VelocityFactor	IN	60.0	REAL	Polygon velocity factor Y9 [%]
rX10ActualWorkingRadius	IN	0.0	REAL	Polygon actual working radius X10 [mm]
rY10VelocityFactor	IN	50.0	REAL	Polygon velocity factor Y10 [%]
rOutVelocitySetpoint	OUT	DCC	REAL	Effective setpoint velocity [mm/s]
rActiveVelocityFactor	OUT	0.0	REAL	Effective velocity factor [%]

5.3 Crane FB library

5.3.1 General information

The source code of each block is open for customer-specific adaptation.

A description of how to update the Crane_FB_Library is provided in the chapter titled Setup and version update for crane libraries (Page 289).

If not otherwise explained, the following statements apply to all ST function blocks.

Firmware version

- SIMOTION Drive Based (D4x5-2): Firmware as of V4.3 SP1
- SINAMICS: Firmware as of V4.5

Call

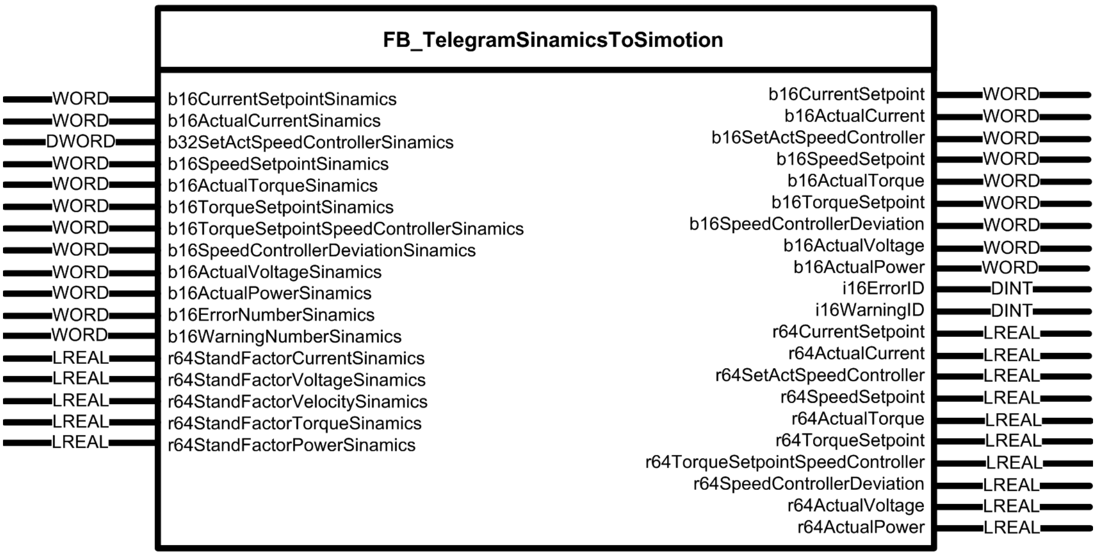
The function blocks must be called in a cyclic task as data is cyclically exchanged between SIMOTION and SINAMICS.

5.3.2 FB_TelegramSinamicsToSimotion

Task

The receive data words from SINAMICS are converted for further processing in SIMOTION using the function block FB_TelegramSinamicsToSimotion.

Template for the call (FBD representation type)


FB_TelegramSinamicsToSimotion
WORD b16CurrentSetpointSinamics
WORD b16ActualCurrentSinamics
DWORD b32SetActSpeedControllerSinamics
WORD b16SpeedSetpointSinamics
WORD b16ActualTorqueSinamics
WORD b16TorqueSetpointSinamics
WORD b16TorqueSetpointSpeedControllerSinamics
WORD b16SpeedControllerDeviationSinamics
WORD b16ActualVoltageSinamics
WORD b16ActualPowerSinamics
WORD b16ErrorNumberSinamics
WORD b16WarningNumberSinamics
LREAL r64StandFactorCurrentSinamics
LREAL r64StandFactorVoltageSinamics
LREAL r64StandFactorVelocitySinamics
LREAL r64StandFactorTorqueSinamics
LREAL r64StandFactorPowerSinamics
b16CurrentSetpoint WORD
b16ActualCurrent WORD
b16SetActSpeedController WORD
b16SpeedSetpoint WORD
b16ActualTorque WORD
b16TorqueSetpoint WORD
b16SpeedControllerDeviation WORD
b16ActualVoltage WORD
b16ActualPower WORD
i16ErrorID DINT
i16WarningID DINT
r64CurrentSetpoint LREAL
r64ActualCurrent LREAL
r64SetActSpeedController LREAL
r64SpeedSetpoint LREAL
r64ActualTorque LREAL
r64TorqueSetpoint LREAL
r64TorqueSetpointSpeedController LREAL
r64SpeedControllerDeviation LREAL
r64ActualVoltage LREAL
r64ActualPower LREAL

Connections

Name	Connection type	Default setting	Data type	Meaning
b16CurrentSetpointSinamics	IN	16#0	WORD	Current setpoint, torque-generating
b16ActualCurrentSinamics	IN	16#0	WORD	Current actual value, total
b32SetActSpeedControllerSinamics	IN	16#0	DWORD	Actual speed
b16SpeedSetpointSinamics	IN	16#0	WORD	Speed setpoint
b16ActualTorqueSinamics	IN	16#0	WORD	Actual torque value
b16TorqueSetpointSinamics	IN	16#0	WORD	Torque setpoint
b16TorqueSetpointSpeedControllerSinamics	IN	16#0	WORD	I component of the speed controller
b16SpeedControllerDeviationSinamics	IN	16#0	WORD	Speed deviation
b16ActualVoltageSinamics	IN	16#0	WORD	Motor voltage actual value
b16ActualPowerSinamics	IN	16#0	WORD	Actual power value
b16ErrorNumberSinamics	IN	16#0	WORD	Actual fault code
b16WarningNumberSinamics	IN	16#0	WORD	Actual alarm code

Name	Connection type	Default setting	Data type	Meaning
r64StandFactorCurrentSinamics	IN	1.0	LREAL	Standard factor to convert current
r64StandFactorVoltageSinamics	IN	1.0	LREAL	Standard factor to convert voltage
r64StandFactorVelocitySinamics	IN	1.0	LREAL	Standard factor to convert velocity
r64StandFactorTorqueSinamics	IN	1.0	LREAL	Standard factor to convert torque
r64StandFactorPowerSinamics	IN	1.0	LREAL	Standard factor for power conversion
b16CurrentSetpoint	OUT	16#0	WORD	Current setpoint, torque-generating
b16ActualCurrent	OUT	16#0	WORD	Current actual value, total
b16SetActSpeedController	OUT	16#0	WORD	Actual speed
b16SpeedSetpoint	OUT	16#0	WORD	Speed setpoint
b16ActualTorque	OUT	16#0	WORD	Actual torque value
b16TorqueSetpoint	OUT	16#0	WORD	Torque setpoint
b16SpeedControllerDeviation	OUT	16#0	WORD	Speed deviation
b16ActualVoltage	OUT	16#0	WORD	Motor voltage actual value
b16ActualPower	OUT	16#0	WORD	Actual power value
i16ErrorID	OUT	0	DINT	Actual fault code
i16WarningID	OUT	0	DINT	Actual alarm code
r64CurrentSetpoint	OUT	0.0	LREAL	Current setpoint, torque-generating
r64ActualCurrent	OUT	0.0	LREAL	Current actual value, total
r64SetActSpeedController	OUT	0.0	LREAL	Actual speed
r64SpeedSetpoint	OUT	0.0	LREAL	Speed setpoint
r64ActualTorque	OUT	0.0	LREAL	Actual torque value
r64TorqueSetpoint	OUT	0.0	LREAL	Torque setpoint
r64TorqueSetpointSpeedController	OUT	0.0	LREAL	I component of the speed controller
r64SpeedControllerDeviation	OUT	0.0	LREAL	Speed deviation
r64ActualVoltage	OUT	0.0	LREAL	Motor voltage actual value
r64ActualPower	OUT	0.0	LREAL	Actual power value

Functionality

The data words to be connected at the inputs (e.g. b16ActualCurrentSinamics) are prepared for further use and are available at the output as WORD (e.g. b16ActualCurrent) and as LREAL value (e.g. r64ActualCurrent).

The LREAL variable r64ActualCurrent is formed from the current actual value b16ActualCurrentSinamics using the following formula:

$$\text{r64ActualCurrent} = \frac{\text{b16ActualCurrentSinamics} \bullet \text{r64StandFactorCurrentSinamics}}{16384}$$

The current actual value is available as both an LREAL value and WORD for further processing by various programs.

CAUTION
If current, voltage, power, torque or speed should be used in SIMOTION, the reference data must be entered in the associated standard factors (refer to the SINAMICS S120, S150 List Manual, p2000, p2001, p2002, p2003, r2004). Value 1 can otherwise be retained as the standard factor.

Error messages

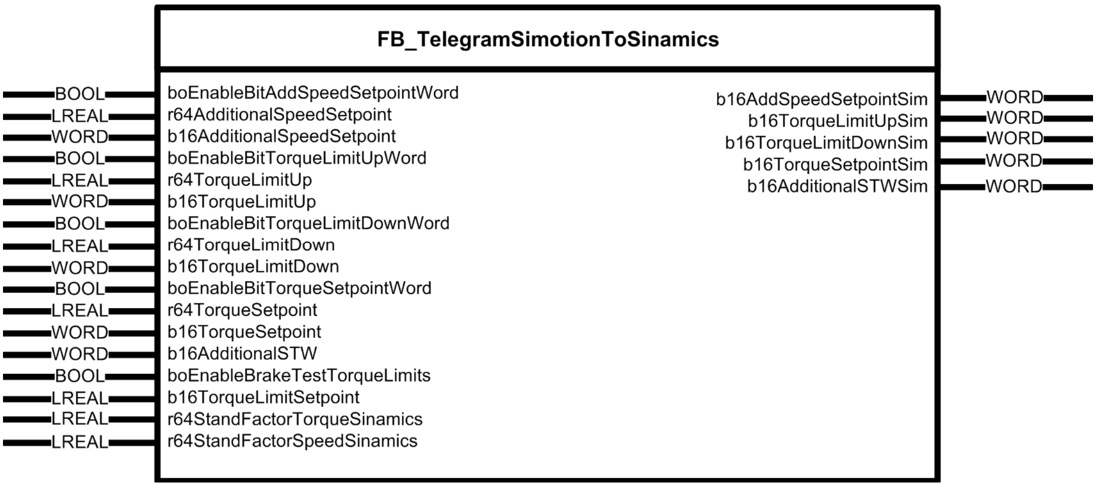
None

5.3.3 FB_TelegramSimotionToSinamics

Task

The set data words from SIMOTION to SINAMICS are processed using the function block FB_TelegramSimotionToSinamics.

Template for the call (FBD representation type)


FB_TelegramSimotionToSinamics
BOOL boEnableBitAddSpeedSetpointWord
LREAL r64AdditionalSpeedSetpoint
WORD b16AdditionalSpeedSetpoint
BOOL boEnableBitTorqueLimitUpWord
LREAL r64TorqueLimitUp
WORD b16TorqueLimitUp
BOOL boEnableBitTorqueLimitDownWord
LREAL r64TorqueLimitDown
WORD b16TorqueLimitDown
BOOL boEnableBitTorqueSetpointWord
LREAL r64TorqueSetpoint
WORD b16TorqueSetpoint
WORD b16AdditionalSTW
BOOL boEnableBrakeTestTorqueLimits
LREAL b16TorqueLimitSetpoint
LREAL r64StandFactorTorqueSinamics
LREAL r64StandFactorSpeedSinamics
b16AddSpeedSetpointSim WORD
b16TorqueLimitUpSim WORD
b16TorqueLimitDownSim WORD
b16TorqueSetpointSim WORD
b16AdditionalSTWSim WORD

Connections

Name	Connection type	Default setting	Data type	Meaning
boEnableBitAddSpeedSetpointWord	IN	False	BOOL	Switchover from the LREAL input, supplementary speed setpoint to word input
r64AdditionalSpeedSetpoint	IN	0.0	LREAL	Supplementary speed setpoint (start pulse)
b16AdditionalSpeedSetpoint	IN	16#0	WORD	Supplementary speed setpoint (start pulse)
boEnableBitTorqueLimitUpWord	IN	False	BOOL	Switchover from the LREAL input, torque limiting (upper) to word input
r64TorqueLimitUp	IN	0.0	LREAL	Torque limiting (upper)
b16TorqueLimitUp	IN	16#0	WORD	Torque limiting (upper)
boEnableBitTorqueLimitDownWord	IN	False	BOOL	Switchover from the LREAL input, torque limiting (lower) to word input
r64TorqueLimitDown	IN	0.0	LREAL	Torque limiting (lower)
b16TorqueLimitDown	IN	16#0	WORD	Torque limiting (lower)
boEnableBitTorqueSetpointWord	IN	False	BOOL	Enable additional torque
r64TorqueSetpoint	IN	0.0	LREAL	Additional torque value
b16TorqueSetpoint	IN	16#0	WORD	Additional torque value
b16AdditionalSTW	IN	16#0	WORD	Additional word for control functions in Sinamics Bit 1 : Set I component
boEnableBrakeTestTorqueLimits	IN	False	BOOL	Enable torque limits for the brake test
b16TorqueLimitSetpoint	IN	16#0	WORD	Torque limit for the brake test, e.g. PZD8
r64StandFactorTorqueSinamics	IN	1.0	LREAL	Standard factor to convert torque limit
r64StandFactorSpeedSinamics	IN	1.0	LREAL	Standard factor to convert supplementary speed setpoint (start pulse)
b16AddSpeedSetpointSim	OUT	16#0	WORD	Supplementary speed setpoint (start pulse)
b16TorqueLimitUpSim	OUT	16#0	WORD	Torque limiting (upper)
b16TorqueLimitDownSim	OUT	16#0	WORD	Torque limiting (lower)
b16TorqueSetpointSim	OUT	16#0	WORD	Torque setpoint
b16AdditionalSTWSIM	OUT	16#0	WORD	Additional word for control functions in Sinamics Bit 1 : Set I component

Functionality

A switchover is made from the standard LREAL input (e.g. r64AdditionalSpeedSetpoint) to the WORD input (b16AdditionalSpeedSetpoint) using a true signal of the boolean enable bits (e.g. boEnableBitAddSpeedSetpointWord) - precisely the same as for the torque limits (upper and lower).

The standard factors for the torque (r64StandFactorTorqueSetpoint) and the supplementary speed setpoint (r64StandFactorAddSpeedSetpoint) are required for the conversion and must therefore have as a minimum the value 1.0.

The calculation of the corresponding data word for SINAMICS is explained here as an example for the upper torque limiting (b16TorqueSetpointSim). The formula for the calculation is as follows:

$$\text{b16TorqueSetpointSim} = \frac{\text{r64TorqueSetpoint} \bullet 16384}{\text{r64StandFactorTorqueSetpoint}}$$

The calculated value is available at the function block output as data word for further processing.

CAUTION
If the torque or speed are to be transferred from SIMOTION to SINAMICS, the reference data must be entered in the default factors for torque (refer to the SINAMICS S List Manual, p2003) and speed (refer to TO configuration parameter TypeOfAxis.MaxVelocity.Maximum).

Error messages

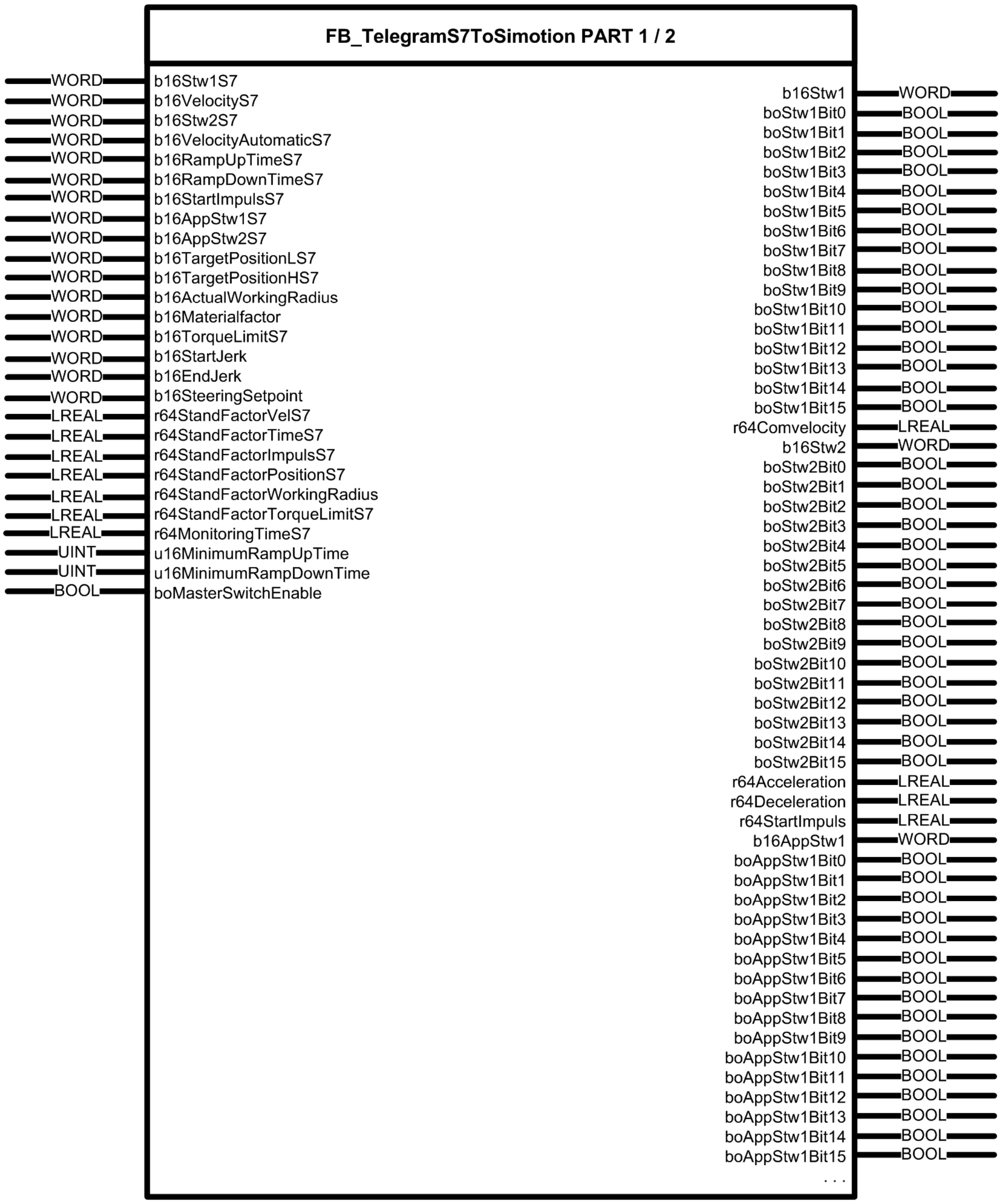
None

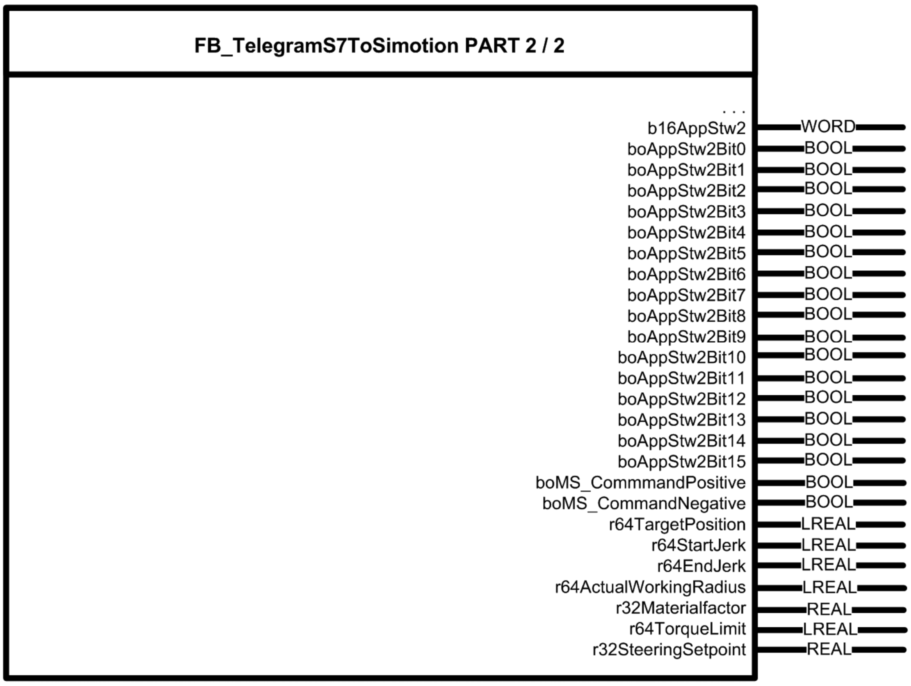
5.3.4 FB_TelegramS7ToSimotion

Task

The receive data words for S7 for further processing in SIMOTION are converted using the function block FB_TelegramS7ToSimotion.

Template for the call (FBD representation type)





Connections

Name	Connection type	Default setting	Data type	Meaning
b16Stw1S7	IN	16#0	WORD	Control word 1 from the S7 module
b16VelocityS7	IN	16#0	WORD	Setpoint for velocity [mm/s]
b16Stw2S7	IN	16#0	WORD	Control word 2 from the S7 module
b16VelocityAutomaticS7	IN	16#0	WORD	Setpoint for velocity in the AUTOMATIC operating mode [mm/s]
b16RampUpTimeS7	IN	16#0	WORD	Ramp-up time [ms]
b16RampDownTimeS7	IN	16#0	WORD	Ramp-down time [ms]
b16StartImpulsS7	IN	16#0	WORD	Setpoint for the start pulse [mm/s]
b16AppStw1S7	IN	16#0	WORD	Application control word 1 from the S7 module
b16AppStw2S7	IN	16#0	WORD	Application control word 2 from the S7 module
b16TargetPositionLS7	IN	16#0	WORD	Target position L word [mm]
b16TargetPositionHS7	IN	16#0	WORD	Target position H word [mm]
b16ActualWorkingRadius	IN	16#0	WORD	Current actual working radius
b16Materialfactor	IN	16#0	WORD	Material factor
b16TorqueLimitS7	IN	16#0	WORD	Torque limit [0-100%]
b16SteeringSetpoint	IN	16#0	WORD	Steering setpoint [-100 to 100%]
b16StartJerk	IN	16#0	WORD	Initial rounding-off of the acceleration
b16EndJerk	IN	16#0	WORD	Final rounding-off of the delay

Name	Connection type	Default setting	Data type	Meaning
r64StandFactorVelS7	IN	1.0	LREAL	Standard factor to convert velocity
r64StandFactorTimeS7	IN	1.0	LREAL	Standard factor to convert acceleration and deceleration
r64StandFactorImpulsS7	IN	1.0	LREAL	Standard factor to convert start pulse
r64StandFactorPositionS7	IN	1.0	LREAL	Standard factor to convert position
r64StandFactorTorqueLimitS7	IN	1.0	LREAL	Standard factor to convert torque limit
r64StandFactorWorkingRadius	IN	1.0	LREAL	Standard factor for actual working radius
r64MonitoringTimeS7	IN	100.0	LREAL	Monitoring time [ms] of the S7 sign-of-life signal (STW1, bit 10). All outputs are set to 0 on timeout.
u16MinimumRampUpTime	IN	1	UINT	Minimum time is saved if the ramp-up time should be zero [ms]
u16MinimumRampDownTime	IN	1	UINT	Minimum time is saved if the ramp-down time should be zero [ms]
boMasterSwitchEnable	IN	False	BOOL	Using the bit, the DCC_MasterSwitch block is used.
b16Stw1	OUT	16#0	WORD	Control word 1
boStw1Bit0	OUT	False	BOOL	On/ off 1
boStw1Bit1	OUT	False	BOOL	Off 2
boStw1Bit2	OUT	False	BOOL	Off 3
boStw1Bit3	OUT	False	BOOL	Pulse enable
boStw1Bit4	OUT	False	BOOL	Ramp-function generator enable
boStw1Bit5	OUT	False	BOOL	Reserved
boStw1Bit6	OUT	False	BOOL	Setpoint enable
boStw1Bit7	OUT	False	BOOL	Acknowledge fault
boStw1Bit8	OUT	False	BOOL	Speed controller enable
boStw1Bit9	OUT	False	BOOL	Reserved
boStw1Bit10	OUT	False	BOOL	Master control by PLC
boStw1Bit11	OUT	False	BOOL	Reserved
boStw1Bit12	OUT	False	BOOL	Used internally
boStw1Bit13 to boStw1Bit15	OUT	False	BOOL	Reserved
r64Comvelocity	OUT	0.0	LREAL	Setpoint velocity [mm/s]
b16Stw2	OUT	16#0	WORD	Control word 2
boStw2Bit0	OUT	False	BOOL	PositiveSuperimpose
boStw2Bit1	OUT	False	BOOL	NegativeSuperimpose
boStw2Bit2	OUT	False	BOOL	TandemHoming
boStw2Bit3	OUT	False	BOOL	TandemMode
boStw2Bit4	OUT	False	BOOL	SlaveTandemMode
boStw2Bit5	OUT	False	BOOL	DriveMasterSuperimpose
boStw2Bit6	OUT	False	BOOL	SelectMasterAxis2
boStw2Bit7	OUT	False	BOOL	SelectMasterAxis3
boStw2Bit8	OUT	False	BOOL	SelectCheckbackSlave
boStw2Bit9	OUT	False	BOOL	SelectCheckbackSlaveTandem

Name	Connection type	Default setting	Data type	Meaning
boStw2Bit10	OUT	False	BOOL	SelectCheckbackSlaveSlaveTandem
boStw2Bit11	OUT	False	BOOL	boOffsetHoming
boStw2Bit12	OUT	False	BOOL	boOffsetMode
boStw2Bit13 to boStw2Bit15	OUT	False	BOOL	Reserved
r64Acceleration	OUT	0.0	LREAL	Acceleration [mm/s2]
r64Deceleration	OUT	0.0	LREAL	Deceleration [mm/s2]
r64StartImpuls	OUT	0.0	LREAL	Setpoint for the start pulse [mm/s]
b16AppStw1	OUT	16#0	WORD	Application control word 1
boAppStw1Bit0	OUT	False	BOOL	The bit is not used.
boAppStw1Bit1	OUT	False	BOOL	The bit is not used.
boAppStw1Bit2	OUT	False	BOOL	Enable start pulse
boAppStw1Bit3	OUT	False	BOOL	Select heavy-duty operation
boAppStw1Bit4	OUT	False	BOOL	Enable field-weakening
boAppStw1Bit5	OUT	False	BOOL	Reset load memory
boAppStw1Bit6	OUT	False	BOOL	Selecting master-slave operation
boAppStw1Bit7	OUT	False	BOOL	Selecting synchronous operation
boAppStw1Bit8	OUT	False	BOOL	Save offset
boAppStw1Bit9	OUT	False	BOOL	Start AUTOMATIC request
boAppStw1Bit10	OUT	False	BOOL	AUTOMATIC operating mode
boAppStw1Bit11	OUT	False	BOOL	MANUAL operating mode
boAppStw1Bit12	OUT	False	BOOL	The bit is not used.
boAppStw1Bit13	OUT	False	BOOL	SPEED_CONTROLLED operating mode
boAppStw1Bit14	OUT	False	BOOL	SENSORLESS EMERGENCY operating mode
boAppStw1Bit15	OUT	False	BOOL	SWAYCONTROL operating mode
b16AppStw2	OUT	16#0	WORD	Application control word 2
boAppStw2Bit0	OUT	False	BOOL	The bit is not used.
boAppStw2Bit1	OUT	False	BOOL	Select velocity limit, bit 1
boAppStw2Bit2	OUT	False	BOOL	Select velocity limit, bit 2
boAppStw2Bit3	OUT	False	BOOL	Selects prelimit switch
boAppStw2Bit4	OUT	False	BOOL	Enable slack rope controller
boAppStw2Bit5	OUT	False	BOOL	Command, save grab open
boAppStw2Bit6	OUT	False	BOOL	Command, save grab closed
boAppStw2Bit7	OUT	False	BOOL	Select orange-peel bucket
boAppStw2Bit8	OUT	False	BOOL	Enable current equalization controller
boAppStw2Bit9	OUT	False	BOOL	Select grab change
boAppStw2Bit10	OUT	False	BOOL	Homing
boAppStw2Bit11	OUT	False	BOOL	Select torque limiting
boAppStw2Bit12	OUT	False	BOOL	Encoder switchover
boAppStw2Bit13	OUT	False	BOOL	Selects slave operation, otherwise master operation is active
boAppStw2Bit14	OUT	False	BOOL	Select brake test

Name	Connection type	Default setting	Data type	Meaning
boAppStw2Bit15	OUT	False	BOOL	Select technology object active
boMS_CommandPositive	OUT	False	BOOL	Command, positive direction
boMS_CommandNegative	OUT	False	BOOL	Command, negative direction
r64TargetPosition	OUT	0.0	LREAL	Target position [mm]
r64StartJerk	OUT	0.0	LREAL	Initial rounding-off of the acceleration
r64EndJerk	OUT	0.0	LREAL	Final rounding-off of the delay
r64ActualWorkingRadius	OUT	0.0	LREAL	Current actual working radius
r32Materialfactor	OUT	0.0	REAL	Material factor
r64TorqueLimit	OUT	0.0	LREAL	Torque limit
r32SteeringSetpoint	OUT	0.0	REAL	Steering setpoint

Functionality

The control words (e.g. b16Stw2S7) to be connected to the inputs are conditioned for further use and are available at the output as word (e.g. b16Stw2) and as boolean value (e.g. boStw2Bit0 to boStw2Bit15).

The LREAL variable "r64ComVelocity" is generated from the setpoint velocity word b16VelocityS7 under the following formula; this is available for further processing for various programs:

$$\text{r64ComVelocity} = \frac{\text{b16VelocityS7} \bullet \text{r64StandFactorTimeS7}}{16384}$$

The same applies to the start pulse. For the ramp-up and ramp-down time from the S7 (e.g. b16RampUpTimeS7), the LREAL variables are formed (e.g. r64Acceleration). If a value of zero is in one of the two variables (e.g. b16RampUpTimeS7), then the zero is replaced by the variable saved as minimum time (ms) (e.g. u16MinimumRampUpTime).

The target position "b16TargetPositionLS7" and "b16TargetPositionHS7" is formed using the data words "r64TargetPosition".

NOTICE

The function block assigns zero to all outputs when the timer r64MonitoringtimeS7 has run down and True is not applied to bit 10 in control word 1.

CAUTION

The reference data for the speed and start pulse values transferred from the S7 to SIMOTION must be entered in the standard factor (refer to TO configuration parameter TypeOfAxis.MaxVelocity.Maximum).

If the DINT range of values is to be extended, the position standard factor must be adapted accordingly (r64StandFactorPositionS7).

If the UINT range of values is to be extended, the time standard factor must be adapted accordingly (r64StandFactorTimeS7).

Error messages

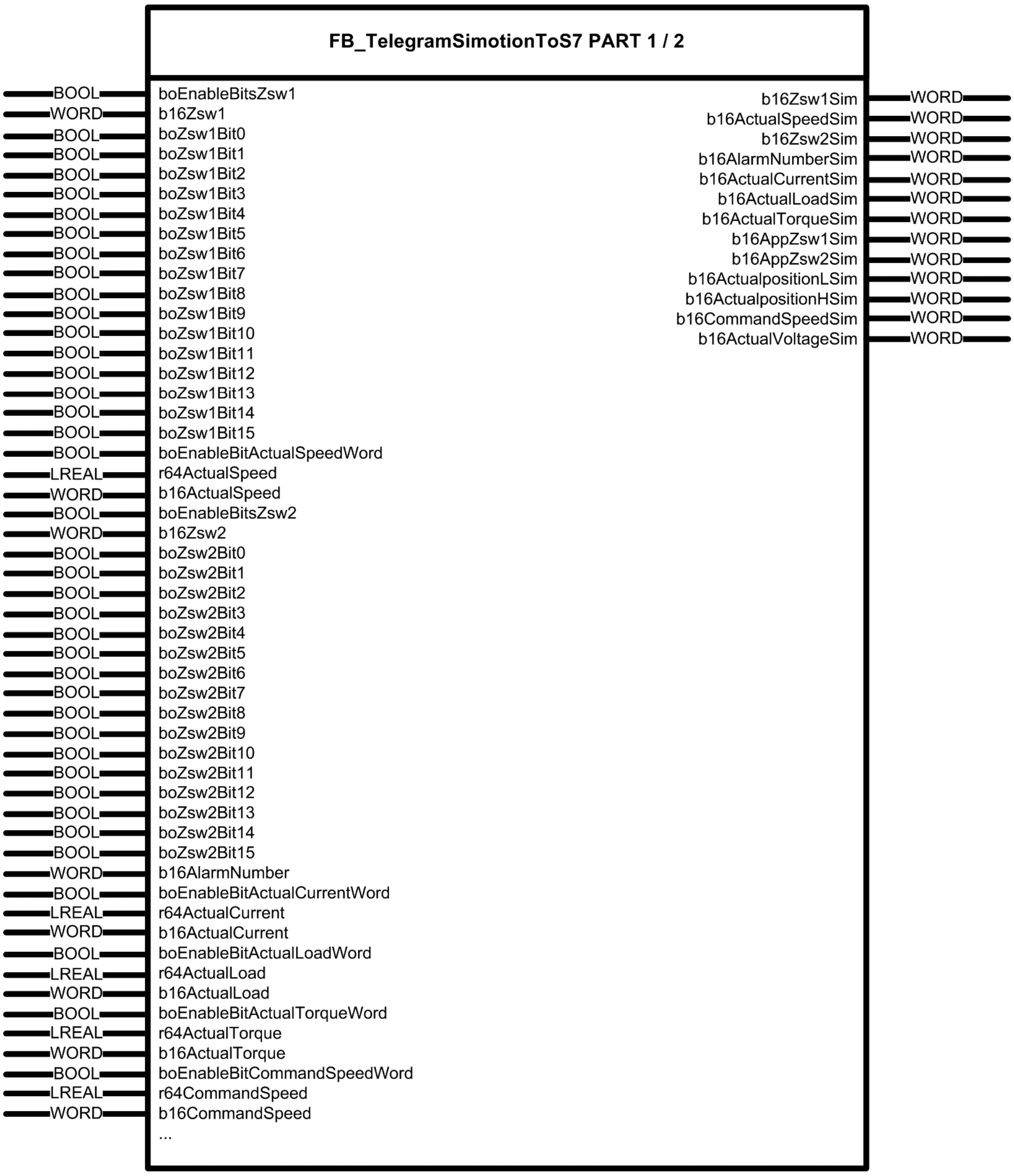
None

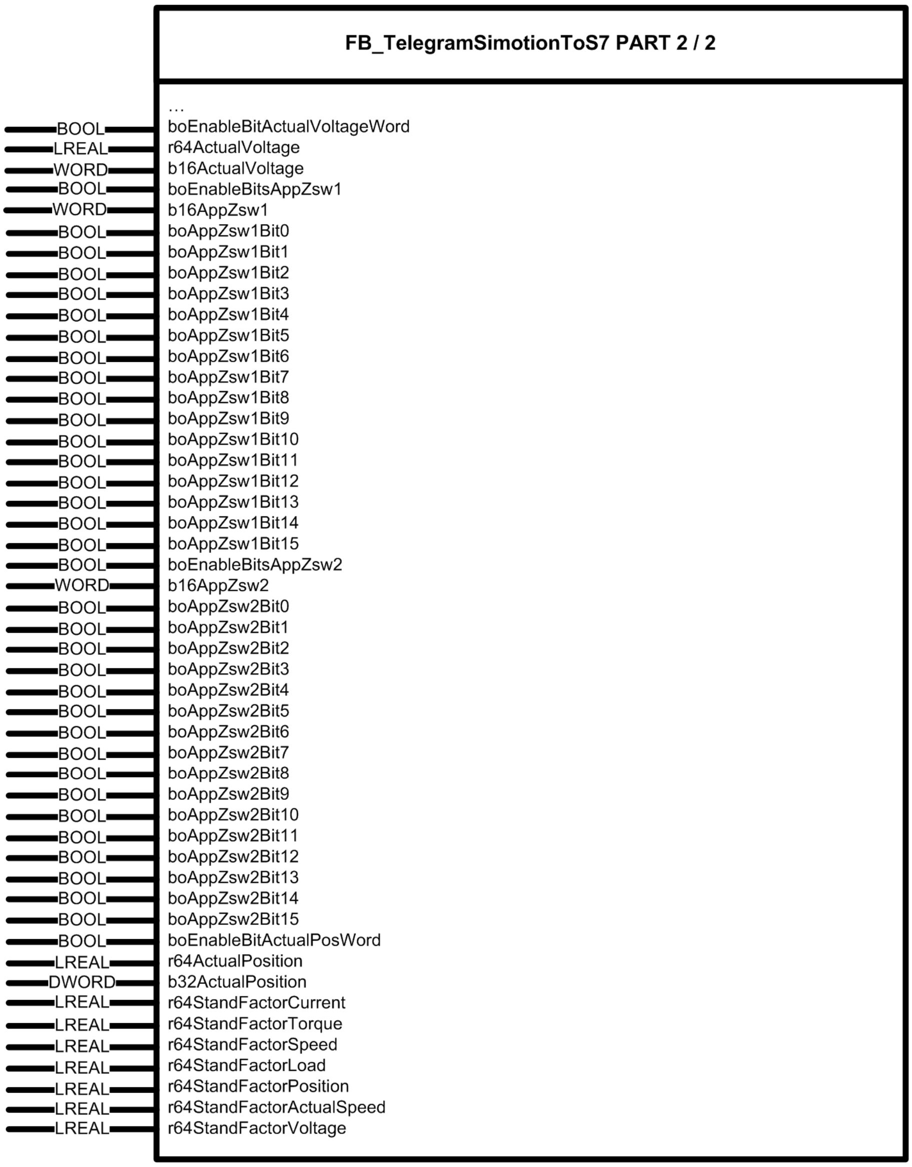
5.3.5 FB_TelegramSimotionToS7

Task

The send data words of SIMOTION to S7 are processed using function block FB_TelegramSimotionToS7.

Template for the call (FBD representation type)





Connections

Name	Connection type	Default setting	Data type	Meaning
boEnableBitsZSW1	IN	False	BOOL	Switchover from word to boolean input ZSW1
b16Zsw1	IN	16#0	WORD	Status word 1
boZsw1Bit0	IN	False	BOOL	Ready to switch-on
boZsw1Bit1	IN	False	BOOL	Ready
boZsw1Bit2	IN	False	BOOL	Enable operation
boZsw1Bit3	IN	False	BOOL	Fault present
boZsw1Bit4	IN	False	BOOL	Coasting down active (OFF2)

Name	Connection type	Default setting	Data type	Meaning
boZsw1Bit5	IN	False	BOOL	Fast stop active (OFF3)
boZsw1Bit6	IN	False	BOOL	Power-on inhibit
boZsw1Bit7	IN	False	BOOL	Alarm present
boZsw1Bit8	IN	False	BOOL	Speed setpoint-actual value deviation in the tolerance range
boZsw1Bit9	IN	False	BOOL	Control requested to the PLC
boZsw1Bit10	IN	False	BOOL	f or n comparison value reached or exceeded
boZsw1Bit11	IN	False	BOOL	I, M or P limit reached or exceeded
boZsw1Bit12	IN	False	BOOL	Holding brake open
boZsw1Bit13	IN	False	BOOL	Interpolator active (ramp-function generator active)
boZsw1Bit14	IN	False	BOOL	n_act ≥ 0
boZsw1Bit15	IN	False	BOOL	Alarm, drive converter thermal overload
boEnableBitActualSpeedWord	IN	False	BOOL	Switchover from the LREAL input, actual velocity to word input actual velocity
r64ActualSpeed	IN	0.0	LREAL	Actual velocity [mm/s]
b16ActualSpeed	IN	16#0	WORD	Actual velocity [mm/s]
boEnableBitsZSW2	IN	False	BOOL	Switchover from word to boolean input ZSW2
b16Zsw2	IN	16#0	WORD	Status word 2
boZsw2Bit0	IN	False	BOOL	The bit is not used.
boZsw2Bit1	IN	False	BOOL	Safe Torque Off active
boZsw2Bit2	IN	False	BOOL	TandemHoming active
boZsw2Bit3	IN	False	BOOL	TandemMode active
boZsw2Bit4	IN	False	BOOL	SlaveTandemMode active
boZsw2Bit5	IN	False	BOOL	DriveMasterSuperimpose
boZsw2Bit6	IN	False	BOOL	SelectMasterAxis2 active
boZsw2Bit7	IN	False	BOOL	SelectMasterAxis3 active
boZsw2Bit8	IN	False	BOOL	CheckbackSlave active
boZsw2Bit9	IN	False	BOOL	CheckbackSlaveTandem active
boZsw2Bit10	IN	False	BOOL	CheckbackSlaveSlaveTandem active
boZsw2Bit11	IN	False	BOOL	boOffsetHomingActive
boZsw2Bit12	IN	False	BOOL	boOffsetModeActive
boZsw2Bit13 to boZsw2Bit15	IN	False	BOOL	The bits are not used.
b16AlarmNumber	IN	16#0	WORD	Fault and alarm numbers
boEnableBitActualCurrentWord	IN	False	BOOL	Switchover from the LREAL input, current actual value to word input current actual value
r64ActualCurrent	IN	0.0	LREAL	Current actual value [A]
b16ActualCurrent	IN	16#0	WORD	Current actual value [A]
boEnableBitActualLoadWord	IN	False	BOOL	Switchover from the LREAL input, load actual value to word input load actual value
r64ActualLoad	IN	0.0	LREAL	Load actual value
b16ActualLoad	IN	16#0	WORD	Load actual value

Name	Connection type	Default setting	Data type	Meaning
boEnableBitActualTorqueWord	IN	False	BOOL	Switchover from the LREAL input, torque actual value to word input torque actual value
r64ActualTorque	IN	0.0	LREAL	Torque actual value [Nm]
boEnableBitSetpointTorqueWord	IN	False	BOOL	Switchover from the LREAL input, torque setpoint to word input torque setpoint
r64SetpointTorque	IN	0.0	LREAL	Torque setpoint [Nm]
b16SetpointTorque	IN	16#0	WORD	Torque setpoint [Nm]
b16ActualTorque	IN	16#0	WORD	Torque actual value [Nm]
boEnableBitCommandSpeedWord	IN	False	BOOL	Switchover from the LREAL input, speed setpoint to word input speed setpoint
r64CommandSpeed	IN	0.0	LREAL	Speed setpoint before the speed controller [mm/s]
b16CommandSpeed	IN	16#0	WORD	Speed setpoint before the speed controller [mm/s]
boEnableBitActualVoltageWord	IN	False	BOOL	Switchover from the LREAL input, voltage actual value to word input voltage actual value
r64ActualVoltage	IN	0.0	LREAL	Voltage actual value [V]
b16ActualVoltage	IN	16#0	WORD	Voltage actual value [V]
boEnableBitsAppZSW1	IN	False	BOOL	Switchover from word to boolean input AppZSW1
b16AppZsw1	IN	16#0	WORD	Application status word 1
boAppZsw1Bit0	IN	False	BOOL	Axis moves in the positive direction (H)
boAppZsw1Bit1	IN	False	BOOL	Axis moves in the negative direction (H)
boAppZsw1Bit2	IN	False	BOOL	Message, drive stationary (H)
boAppZsw1Bit3	IN	False	BOOL	Message, current distribution monitoring responded (H)
boAppZsw1Bit4	IN	False	BOOL	Message, field weakening enabled (H)
boAppZsw1Bit5	IN	False	BOOL	Message, AUTOMATIC operating mode, target position reached (H)
boAppZsw1Bit6	IN	False	BOOL	Message, master-slave operation active (H)
boAppZsw1Bit7	IN	False	BOOL	Message, synchronous operation active (H)
boAppZsw1Bit8	IN	False	BOOL	Message, offset active (H)
boAppZsw1Bit9	IN	False	BOOL	Message, AUTOMATIC request active (H)
boAppZsw1Bit10	IN	False	BOOL	AUTOMATIC operating mode (H)
boAppZsw1Bit11	IN	False	BOOL	MANUAL operating mode (H)
boAppZsw1Bit12	IN	False	BOOL	The bit is not used.
boAppZsw1Bit13	IN	False	BOOL	SPEED_CONTROLLED (H) operating mode
boAppZsw1Bit14	IN	False	BOOL	Operating mode SENSORLESS EMERGENCY (H)
boAppZsw1Bit15	IN	False	BOOL	SwayControl operating mode
boEnableBitsAppZSW2	IN	False	BOOL	Switchover from word to boolean input AppZSW2
b16AppZsw2	IN	16#0	WORD	Application status word 2

Name	Connection type	Default setting	Data type	Meaning
boAppZsw2Bit0	IN	False	BOOL	Closed-loop torque control active (H) or grab ½ open (H)
boAppZsw2Bit1	IN	False	BOOL	Speed control active (H)
boAppZsw2Bit2	IN	False	BOOL	Closed-loop position control active (H) or grab 2/3 closed (H)
boAppZsw2Bit3	IN	False	BOOL	Message a/v reduction (H)
boAppZsw2Bit4	IN	False	BOOL	Message, grab open (H)
boAppZsw2Bit5	IN	False	BOOL	Message, grab closed (H)
boAppZsw2Bit6	IN	False	BOOL	SIMOTION fault (H)
boAppZsw2Bit7	IN	False	BOOL	SINAMICS fault (H)
boAppZsw2Bit8	IN	False	BOOL	Function block fault (H)
boAppZsw2Bit9	IN	False	BOOL	Message, torque limiting active
boAppZsw2Bit10	IN	False	BOOL	Message, homed (H)
boAppZsw2Bit11	IN	False	BOOL	Position difference outside tolerance
boAppZsw2Bit12	IN	False	BOOL	Encoder switchover active
boAppZsw2Bit13	IN	False	BOOL	Slave operation active, otherwise master operation is active
boAppZsw2Bit14	IN	False	BOOL	Brake test active
boAppZsw2Bit15	IN	False	BOOL	Technology object is active (H) or message "Grab touchdown" (H)
boEnableBitActualPosWord	IN	False	BOOL	Switchover from the LREAL input, position actual value to word input position actual value
r64ActualPosition	IN	0.0	LREAL	Position actual value [mm]
b32ActualPosition	IN	16#0	DWORD	Position actual value [mm]
r64SetpointAcceleration	IN	1.0	LREAL	Acceleration setpoint [mm/s²]
r64StandFactorCurrent	IN	1.0	LREAL	Standard factor to convert current
r64StandFactorTorque	IN	1.0	LREAL	Standard factor to convert torque
r64StandFactorSpeed	IN	1.0	LREAL	Standard factor for convert speed setpoint
r64StandFactorLoad	IN	1.0	LREAL	Standard factor to convert load
r64StandFactorPosition	IN	1.0	LREAL	Standard factor to convert position actual value
r64StandFactorActualSpeed	IN	1.0	LREAL	Standard factor to convert actual velocity
r64StandFactorVoltage	IN	1.0	LREAL	Standard factor to convert voltage actual value
r64StandFactorAcceleration	IN	1.0	LREAL	Standard factor to convert the acceleration setpoint
b16Zsw1Sim	OUT	16#0	WORD	Status word 1 SIMOTION
b16ActualSpeedSim	OUT	16#0	WORD	Speed setpoint SIMOTION [mm/s]
b16Zsw2Sim	OUT	16#0	WORD	Status word 2 SIMOTION
b16AlarmNumberSim	OUT	16#0	WORD	Fault and alarm numbers
b16ActualCurrentSim	OUT	16#0	WORD	Smoothed accumulated actual current SIMOTION [A]
b16ActualLoadSim	OUT	16#0	WORD	Actual load SIMOTION

Name	Connection type	Default setting	Data type	Meaning
b16ActualTorqueSim	OUT	16#0	WORD	Smoothed actual torque SIMOTION [N/m]
b16SetpointTorqueSim	OUT	16#0	WORD	SIMOTION torque setpoint [Nm]
b16AppZsw1Sim	OUT	16#0	WORD	Application status word 1 SIMOTION
b16AppZsw2Sim	OUT	16#0	WORD	Application status word 2 SIMOTION
b16ActualpositionLSim	OUT	16#0	WORD	Actual position SIMOTION Low Word [mm/s]
b16ActualpositionHSim	OUT	16#0	WORD	Actual position SIMOTION High Word [mm/s]
b16CommandSpeedSim	OUT	16#0	WORD	Speed setpoint before the speed controller SIMOTION [mm/s]
b16ActualVoltageSim	OUT	16#0	WORD	Voltage actual value SIMOTION [V]
b16SetpointAcceleration	OUT	16#0	WORD	Acceleration setpoint [mm/s²]

Functionality

Using the various Bool enable bits of the status words (e.g. boEnableBitsZSW1) it can be selected whether the corresponding status word is created as bit or as data word. With a True at the input of a Bool enable bit, inputs (e.g. boZsw1Bit0 to boZsw1Bit15 of ZSW1) are fetched, otherwise the corresponding data word (here, b16zsw1) is evaluated.

Using the boolean inputs (e.g. boEnableBitActualSpeedWord), with a true, the data transfer into the block can be activated as data word and not as LREAL. The corresponding is true for the speed setpoint as well as the actual values of current, load, position and torque.

The standard factors for current, torque, velocity, load, position, actual velocity are required for the conversion and must therefore have, as a minimum, a value of 1.0.

The calculation of the corresponding data words for the S7 are explained here using as an example the speed actual value (b16ActualSpeedSim):

$$\text{b16ActualSpeedSim} = \frac{\text{r64ActualSpeed} \cdot 16384}{\text{r64StandfactorActualSpeed}}$$

CAUTION
The standard factors for current, torque, velocity, load, position and actual velocity are required for the conversion and must therefore have, as a minimum, a value of 1.0. If an absolute value is to be transferred from SIMOTION to the S7, the reference value must be entered in the corresponding standard factor.

Error messages

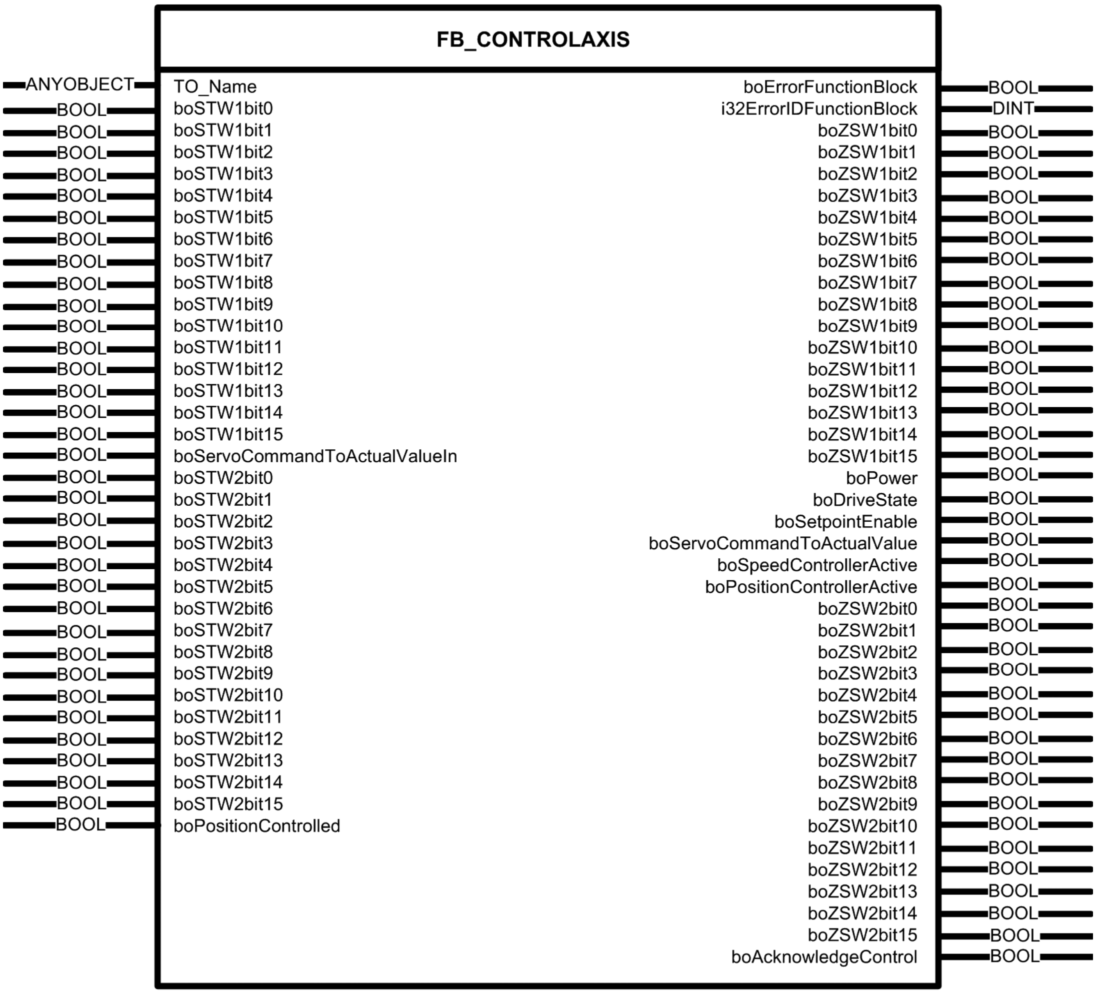
None

5.3.6 FB_ControlAxis

Task

Control words 1 and 2 are controlled using function block FB_ControlAxis. The function block supplies status words 1 and 2 at the output.

Template for the call (FBD representation type)



Connections

Name	Connection type	Default setting	Data type	Meaning
TO_Name	IN	-	ANYOBJECT	Technology object name
boSTW1bit0	IN	False	BOOL	On / Off1; see table "Control_word_1_S7" in Chapter SIMATIC S7 → SIMOTION (Page 307) to boSTW1bit15
boSTW1bit1	IN	False	BOOL	Off 2
boSTW1bit2	IN	False	BOOL	Off 3
boSTW1bit3	IN	False	BOOL	Pulse enable
boSTW1bit4	IN	False	BOOL	Ramp-function generator enable
boSTW1bit5	IN	False	BOOL	Reserved
boSTW1bit6	IN	False	BOOL	Enable setpoint
boSTW1bit7	IN	False	BOOL	Acknowledge fault
boSTW1bit8	IN	False	BOOL	Enable speed controller
boSTW1bit9	IN	False	BOOL	Reserved
boSTW1bit10	IN	False	BOOL	Master control by programmable controller
boSTW1bit11	IN	False	BOOL	Reserved
boSTW1bit12	IN	False	BOOL	Select closed-loop torque control
boSTW1bit13 to boSTW1bit14	IN	False	BOOL	Reserved
boSTW1bit15	IN	False	BOOL	Intelligent overspeed signal
boServoCommandToActualValueIn	IN	False	BOOL	The bit switches into the follow-up mode. As a result of the follow-up mode, the position controller is switched out and the setpoint position tracks the actual position.
boSTW2bit0	IN	False	BOOL	Selects drive data set DDS bit 0
boSTW2bit1	IN	False	BOOL	Selects drive data set DDS bit 1
boSTW2bit2	IN	False	BOOL	Selects drive data set DDS bit 2
boSTW2bit3 to boSTW2bit6	IN	False	BOOL	Reserved for internal use
boSTW2bit7	IN	False	BOOL	Acknowledge SINAMICS fault
boSTW2bit8 to boSTW2bit11	IN	False	BOOL	Reserved
boSTW2bit12 to boSTW2bit15	IN	False	BOOL	Reserved for internal use
boPositionControlled	IN	False	Bool	Using the bit, the axis is enabled, closed-loop position controlled, otherwise only closed-loop speed controlled. This input must be connected to the "boPositionControlled" output of FB_OperationMode.
boErrorFunctionBlock	OUT	False	BOOL	Display, fault present
i32ErrorIDFunctionBlock	OUT	0	DINT	Display, error number
boZSW1bit0	OUT	False	BOOL	Ready to power up see table "Status_word_1_SINAMICS" in Chapter SINAMICS → SIMOTION (Page 328) to boZSW1bit15

Name	Connection type	Default setting	Data type	Meaning
boZSW1bit1	OUT	False	BOOL	Ready
boZSW1bit2	OUT	False	BOOL	Enable operation
boZSW1bit3	OUT	False	BOOL	Fault present
boZSW1bit4	OUT	False	BOOL	Coasting down active (OFF2)
boZSW1bit5	OUT	False	BOOL	Fast stop active (OFF3)
boZSW1bit6	OUT	False	BOOL	Power-on inhibit
boZSW1bit7	OUT	False	BOOL	Alarm present
boZSW1bit8	OUT	False	BOOL	Speed setpoint-actual value deviation in the tolerance range
boZSW1bit9	OUT	False	BOOL	Control requested from the automation system
boZSW1bit10	OUT	False	BOOL	f or n comparison value reached or exceeded
boZSW1bit11	OUT	False	BOOL	I, M or P limit reached or exceeded
boZSW1bit12	OUT	False	BOOL	Holding brake closed
boZSW1bit13	OUT	False	BOOL	Alarm overtemperature motor
boZSW1bit14	OUT	False	BOOL	n_act >= 0
boZSW1bit15	OUT	False	BOOL	Alarm, drive converter thermal overload
boPower	OUT	False	BOOL	If this bit has a high signal, this means that the power enable is set.
boDriveState	OUT	False	BOOL	If this bit has a high signal, this means that the drive enable is set.
boSetpointEnable	OUT	False	BOOL	If this bit has a high signal, this means that the setpoint enable is set.
boServoCommandToActualValue	OUT	False	BOOL	If this bit has a high signal, this means that the follow-up mode is active. This means that a position controller is switched out.
boSpeedControllerActive	OUT	False	BOOL	If this bit has a high signal, this means that the speed controller is active.
boPositionControllerActive	OUT	False	BOOL	If this bit has a high signal, this means that the position controller is active.
boZSW2bit0	OUT	False	BOOL	Drive data set DDS effective Bit 0; see table "Status_word_2_SINAMICS" in Chapter"SINAMICS → SIMOTION" (Page 328) to boZSW2bit15
boZSW2bit1	OUT	False	BOOL	Drive data set DDS active, bit 1
boZSW2bit2	OUT	False	BOOL	Drive data set DDS active, bit 2
boZSW2bit3 to boZSW2bit4	OUT	False	BOOL	Reserved
boZSW2bit5	OUT	False	BOOL	Safe Torque Off (was: Safe Standstill) Control Unit active
boZSW2bit6	OUT	False	BOOL	Safe Torque Off (was: Safe Standstill) Motor Module active
boZSW2bit7	OUT	False	BOOL	Reserved

Name	Connection type	Default setting	Data type	Meaning
boZSW2bit8	OUT	False	BOOL	V/f control active
boZSW2bit9	OUT	False	BOOL	Closed-loop torque control active
boZSW2bit10	OUT	False	BOOL	Closed-loop speed control active
boZSW2bit11	OUT	False	BOOL	Reserved
boZSW2bit12	OUT	False	BOOL	Bits 12 to 15 are used in status word 2 to generate the slave sign-of-life character and therefore cannot be used.
boZSW2bit13	OUT	False	BOOL	
boZSW2bit14	OUT	False	BOOL	
boZSW2bit15	OUT	False	BOOL	
boAcknowledgeControl	OUT	False	BOOL	Acknowledge an error for the FB_OperationMode block

Description of the individual bits (refer to the previous table)

Table 5- 6 Connection description of the FB_ControlAxis

Name	STW2 bit 0	STW2 bit 1	STW2 bit 2	Meaning
Drive data set 0	FALSE	FALSE	FALSE	Drive data set 0 is active
Drive data set 1	TRUE	FALSE	FALSE	Drive data set 1 is active
Drive data set 2	FALSE	TRUE	FALSE	Drive data set 2 is active
Drive data set 3	TRUE	TRUE	FALSE	Drive data set 3 is active
Drive data set 4	FALSE	FALSE	TRUE	Drive data set 4 is active
Drive data set 5	TRUE	FALSE	TRUE	Drive data set 5 is active

Note

The input "boPositionControlled" must be connected with the "boPositionControlled" output at function block FB_OperationMode. Only then is it ensured that the drive is also enabled, closed-loop position controlled.

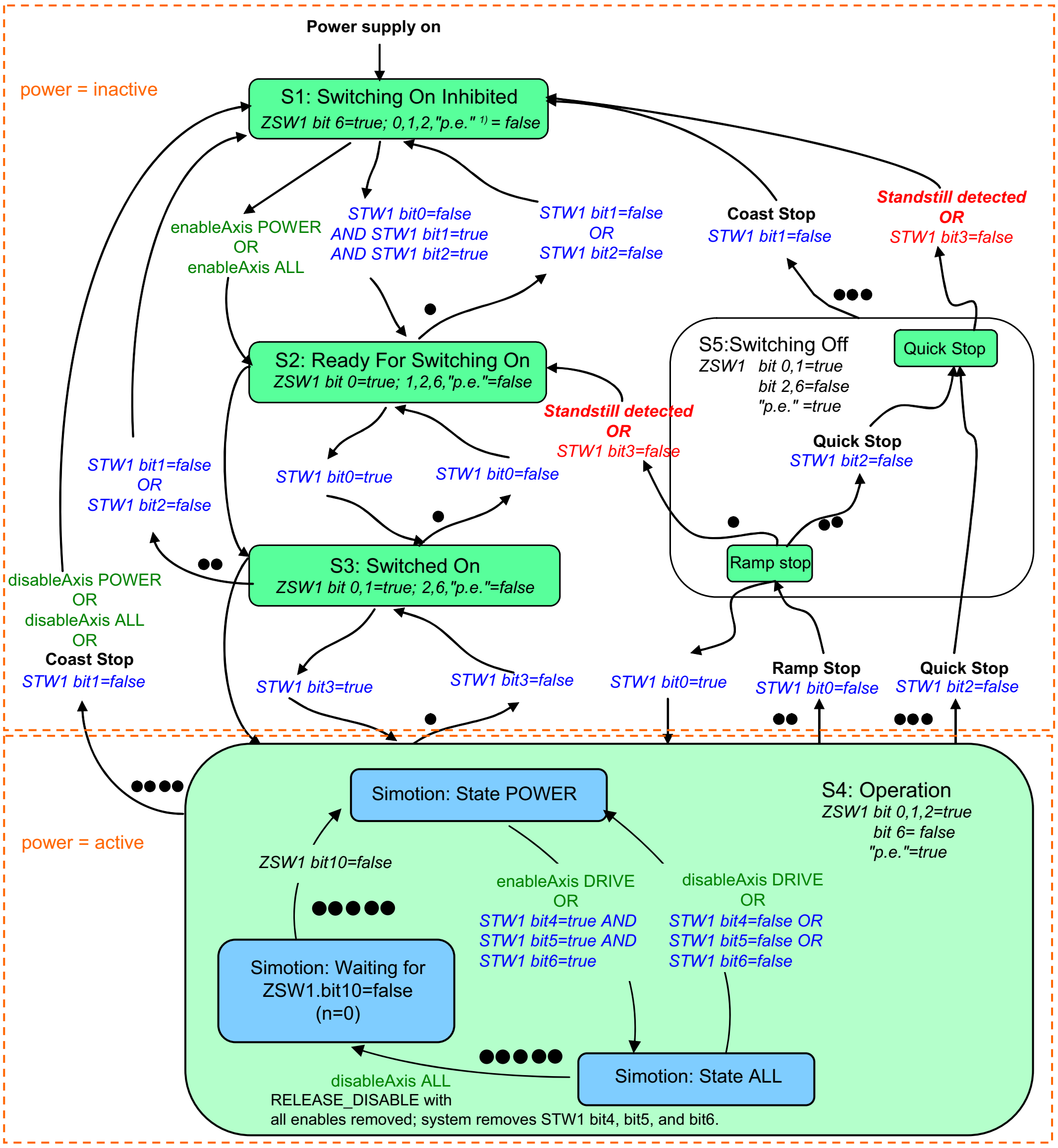
Functionality

The function block has the task of controlling control words 1 and 2. Status words 1 and 2 are output at the block output. When controlling bits 0 to 6, PROFIdrive V3.1 must be carefully taken into consideration (see below, General State Diagram).

The following control bits cannot be used because these are already internally influenced by the technology object:

Table 5- 7 Bits that the user cannot use

Bit for technology object	PROFIdrive V3.1
Control word 1, bit 10	Master control by programmable controller
Control word 2, bit 12	Generation of master sign-of-life character bit 12
Control word 2, bit 13	Generation of master sign-of-life character bit 13
Control word 2, bit 14	Generation of master sign-of-life character bit 14
Control word 2, bit 15	Generation of master sign-of-life character bit 15



STW1 bit x,y = These control word bits must be set by the controller.

ZSW1 bit x,y = These status word bits indicate the current status.

[1] Abbreviation: "p.e." = "pulses enabled" optional, not supported by SIMOTION

Standstill detected: an internal result of a Stop operation

●●●…: The number of points at the illustrated transitions determines the priority. The more points, the higher the priority.

- With SIMOTION V3.2 and higher, two variations are possible for the parameter assignment of the _enableAxis() command:

  POWER/DRIVE/ALL: The StateMachine is controlled by the Axis technology object.

  BY_STW_BIT: The StateMachine is controlled by the user program.

- The drivestate system variable is active in all states S1-S4 if STW1 bits 4 to 6 are set to true.

Figure 5-24 PROFIdrive V3.1 General State Diagram

Error messages

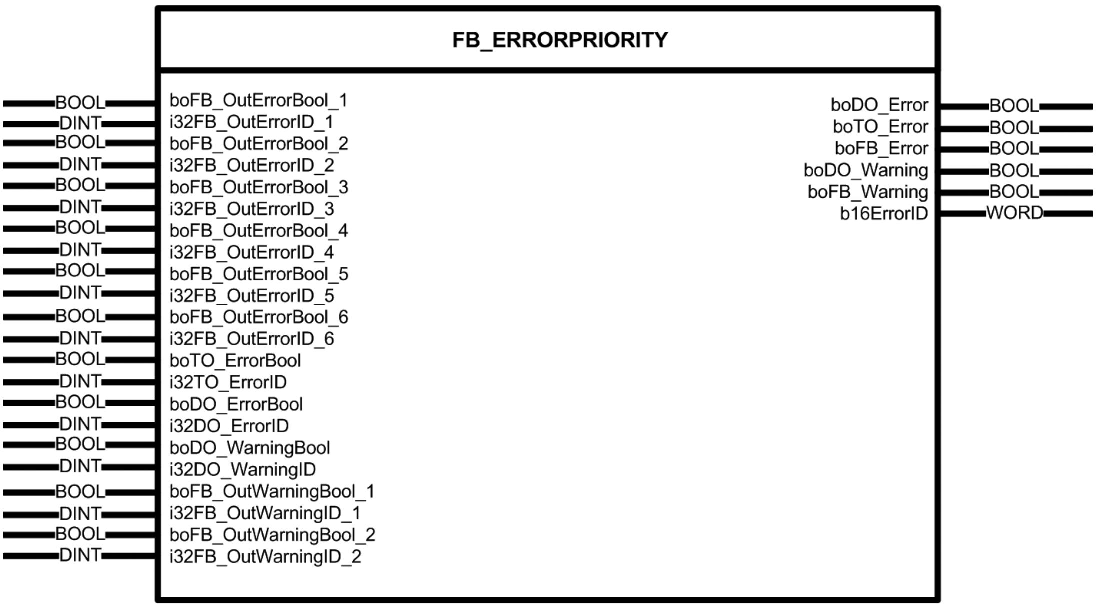
If an error occurs, output "boErrorFunctionBlock" is set and an error number is issued at output "i32ErrorIDFunctionBlock"; see the Chapter Alarm, error and system messages (Page 333).

5.3.7 FB_ErrorPriority

Task

Errors from function blocks SIMOTION technology objects, SINAMICS drive objects and alarms from SINAMICS drive objects and function blocks are compiled and assigned priorities using the function block FB_ErrorPriority.

Template for the call (FBD representation type)



Connections

Name	Connection type	Default setting	Data type	Meaning
boFB_OutErrorBool_1	IN	False	BOOL	Error bit from the function block
i32FB_OutErrorID_1	IN	0	DINT	Error number from the function block
boFB_OutErrorBool_2	IN	False	BOOL	Error bit from the function block
i32FB_OutErrorID_2	IN	0	DINT	Error number from the function block
boFB_OutErrorBool_3	IN	False	BOOL	Error bit from the function block
i32FB_OutErrorID_3	IN	0	DINT	Error number from the function block

Name	Connection type	Default setting	Data type	Meaning
boFB_OutErrorBool_4	IN	False	BOOL	Error bit from the function block
i32FB_OutErrorID_4	IN	0	DINT	Error number from the function block
boFB_OutErrorBool_5	IN	False	BOOL	Error bit from the function block
i32FB_OutErrorID_5	IN	0	DINT	Error number from the function block
boFB_OutErrorBool_6	IN	False	BOOL	Error bit from the function block
i32FB_OutErrorID_6	IN	0	DINT	Error number from the function block
boTO_ErrorBool	IN	False	BOOL	Error bit from the SIMOTION technology object
i32TO_ErrorID	IN	0	DINT	Error number from the SIMOTION technology object
boDO_ErrorBool	IN	False	BOOL	Error bit from the SINAMICS drive object
i32DO_ErrorID	IN	0	DINT	Error number from the SINAMICS drive object
boDO_WarningBool	IN	False	BOOL	Alarm bit from the SINAMICS drive object
i32DO_WarningID	IN	0	DINT	Alarm number from the SINAMICS drive object
boFB_OutWarningBool_1	IN	False	BOOL	Alarm bit from the function block
i32FB_OutWarningID_1	IN	0	DINT	Alarm number from the function block
boFB_OutWarningBool_2	IN	False	BOOL	Alarm bit from the function block
i32FB_OutWarningID_2	IN	0	DINT	Alarm number from the function block
boFB_OutWarningBool_3	IN	False	BOOL	Alarm bit from the function block
i32FB_OutWarningID_3	IN	0	DINT	Alarm number from the function block
boFB_OutWarningBool_4	IN	False	BOOL	Alarm bit from the function block
i32FB_OutWarningID_4	IN	0	DINT	Alarm number from the function block
boFB_OutWarningBool_5	IN	False	BOOL	Alarm bit from the function block
i32FB_OutWarningID_5	IN	0	DINT	Alarm number from the function block
boFB_OutWarningBool_6	IN	False	BOOL	Alarm bit from the function block
i32FB_OutWarningID_6	IN	0	DINT	Alarm number from the function block
boFB_OutWarningBool_1	IN	False	BOOL	Alarm bit from the function block
i32FB_OutWarningID_1	IN	0	DINT	Alarm number from the function block
boFB_OutWarningBool_2	IN	False	BOOL	Alarm bit from the function block
i32FB_OutWarningID_2	IN	0	DINT	Alarm number from the function block
boDO_Error	OUT	False	BOOL	SINAMICS drive object error is active.
boTO_Error	OUT	False	BOOL	SIMOTION technology object error is active.
boFB_Error	OUT	False	BOOL	Function block error is active.
boDO_Warning	OUT	False	BOOL	SINAMICS drive object alarm is active.
boFB_Warning	OUT	False	BOOL	Function block alarm is active.
b16ErrorID	OUT	16#0	WORD	Outputs the error number / alarm number

Functionality

The following priority assignment is defined:

1st priority:	Fault, SINAMICS
2nd priority:	SIMOTION fault
3rd priority:	Fault, function block
4th priority:	SINAMICS alarm
5th priority:	Alarm, function block

Active fault or alarm numbers at the function block input are reported to the S7 controller in accordance with the priority defined above. The function block does not output any internal fault numbers.

The SINAMICS drive object returns fault or alarm numbers. The SIMOTION technology object and the function blocks only return fault numbers. Up to six function block faults can be evaluated at the block input.

These error messages are described in chapter Alarm, error and system messages (Page 333).

Error messages

none

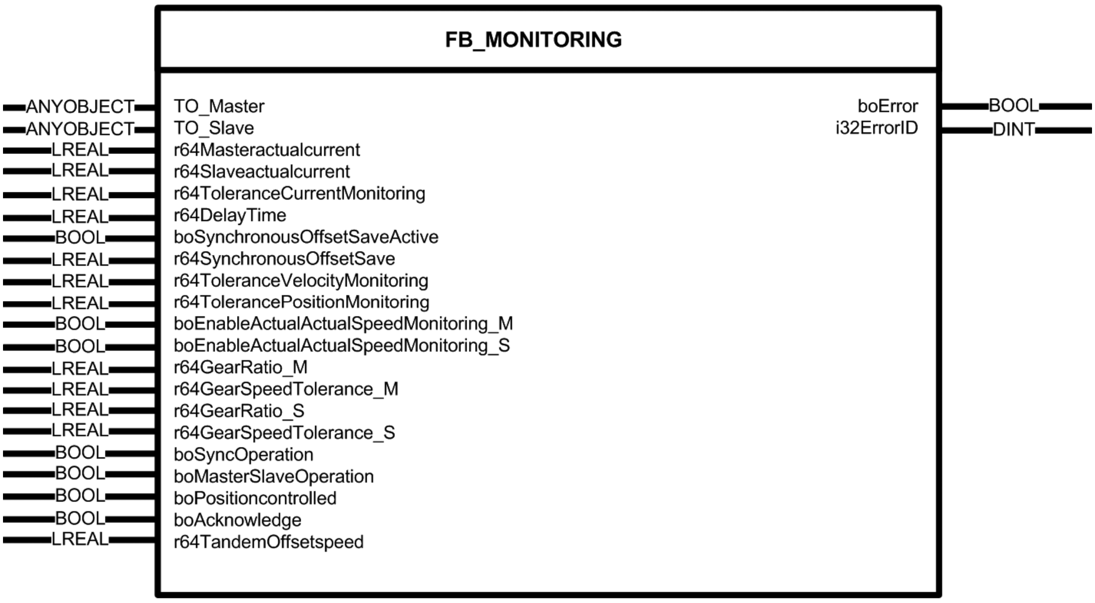
5.3.8 FB_Monitoring

Task

The master-slave connection or synchronous relationship is monitored using the function block FB_Monitoring. The synchronous relationship also includes tandem mode. For a master-slave connection, the actual velocity of the master and that of the slave are monitored. For synchronous operation, the system monitors for an excessively high velocity difference or an excessively high position deviation. A fault signal is generated if the difference of the velocity or the position is higher than the set tolerance.

Furthermore, input "boEnableActualActualSpeedMonitoring" can also be used to monitor the actual-actual speed of a drive equipped with actual speed and position encoders.

Template for the call (FBD representation type)


FB_MONITORING
ANYOBJECT TO_Master
ANYOBJECT TO_Slave
LREAL r64Masteractualcurrent
LREAL r64Slaveactualcurrent
LREAL r64ToleranceCurrentMonitoring
LREAL r64DelayTime
BOOL boSynchronousOffsetSaveActive
LREAL r64SynchronousOffsetSave
LREAL r64ToleranceVelocityMonitoring
LREAL r64TolerancePositionMonitoring
BOOL boEnableActualActualSpeedMonitoring_M
BOOL boEnableActualActualSpeedMonitoring_S
LREAL r64GearRatio_M
LREAL r64GearSpeedTolerance_M
LREAL r64GearRatio_S
LREAL r64GearSpeedTolerance_S
BOOL boSyncOperation
BOOL boMasterSlaveOperation
BOOL boPositioncontrolled
BOOL boAcknowledge
LREAL r64TandemOffsetspeed
boError BOOL
i32ErrorID DINT

Connections

Name	Connection type	Default setting	Data type	Meaning
TO_Master	IN	-	AnyObject	Enter the technology object name of the master (master object)
TO_Slave	IN	-	AnyObject	Enter the technology object name of the slave (slave object)
r64Masteractualcurrent	IN	0.0	LREAL	Actual current of the master (master object) in [A]
r64Slaveactualcurrent	IN	0.0	LREAL	Actual current of the slave (slave object) in [A]
r64ToleranceCurrentMonitoring	IN	0.0	LREAL	In synchronous or master-slave operation, the maximum current difference [A] is set.
r64DelayTime	IN	500.0	LREAL	Delay time [ms]; the velocity or position deviation must lie outside the tolerance range for the configured time before an error is initiated.
boSynchronousOffsetSaveActive	IN	False	BOOL	Feedback signal from block FB_OperationMode of the slave indicating that an offset is active in synchronous operation.
r64SynchronousOffsetSave	IN	0.0	LREAL	Offset value [mm] from block FB_OperationMode of the slave

Name	Connection type	Default setting	Data type	Meaning
r64ToleranceVelocityMonitoring	IN	5.0	LREAL	In synchronous (closed-loop speed controlled) or master-slave operation, the maximum velocity difference [mm/s] is set between the master (master object) and slave (slave object).
r64TolerancePositionMonitoring	IN	5.0	LREAL	In synchronous operation (closed-loop position controlled), the maximum position deviation [mm] between the master (master object) and slave (slave object) is set.
boEnableActualActualSpeedMonitoring_M	IN	False	BOOL	Activates the actual-actual speed monitoring function between the (master) motor encoder and the external encoder.
boEnableActualActualSpeedMonitoring_S	IN	False	BOOL	Activates the actual-actual speed monitoring function between the (slave) motor encoder and the external encoder.
r64GearRatio_M	IN	1.0	LREAL	Gear ratio (motor encoder/external encoder) for master
r64GearSpeedTolerance_M	IN	1.0	LREAL	Permissible tolerance between the two encoders [as a percentage of maximum velocity] (master drive)
r64GearRatio_S	IN	1.0	LREAL	Gear ratio (motor encoder/external encoder) for slave
r64GearSpeedTolerance_S	IN	1.0	LREAL	Permissible tolerance between the two encoders [as a percentage of maximum velocity] (slave drive)
boSyncOperation	IN	False	BOOL	Feedback signal from block FB_OperationMode of the slave indicating that synchronous operation is active.
boMasterSlaveOperation	IN	False	BOOL	Feedback signal from block FB_OperationMode of the slave indicating that the master-slave mode is active.
boPositionControlled	IN	False	BOOL	Feedback signal from block FB_OperationMode of the master indicating that the closed-loop position controlled mode is active.
boAcknowledge	IN	False	BOOL	Acknowledge error
r64TandemOffsetspeed	IN	0.0	LREAL	Supplementary velocity of tandem or offset operation
boError	OUT	False	BOOL	Error bit
i32ErrorID	OUT	0	DINT	Outputs the error number

Functionality

The function block must be cyclically called in the interpolator 1 or interpolator 2 task. The function block has the task:

- In master-slave operation, to monitor the velocity between the master and the slave.
- In synchronous operation, in the AUTOMATIC or MANUAL operating mode, to monitor the position between the master (leading axis) and the slave (following axis) and to only monitor the velocity in all other operating modes.
- An offset can also be taken into account in the AUTOMATIC or MANUAL operating mode.
- The actual current between the master and the slave is monitored in synchronous and master-slave operation.
- If a drive is equipped with a pulse generator for the actual speed and also features a pulse generator for the actual position, which is mounted on the drum rather than on the motor, for example, a monitoring mechanism can be implemented by evaluating the speed of both encoders. This functionality can be used, for example, for gear fracture and coupling monitoring on two drives with rigid coupling. The gear factors "r64GearRatio_M" and "r64GearRatio_S" are also effective for the second encoder. The gear difference between the first and second encoder must be entered in this. If there is no difference, a "1" must be entered.
- For master and slave monitoring, all that is required is one subroutine call of the monitoring block. Gear monitoring of master and slave is also performed in this same subroutine call.
- A total of three subroutine calls are needed for tandem mode.
  - The first subroutine call is needed to compare the first master (tandem master) and its slave (e.g. Hoist_1 and Hoist_2).
  - The second subroutine call is needed to compare the second master (tandem slave) and its slave (e.g. Hoist_3 and Hoist_4).
  - The third subroutine call is needed to compare the tandem master and tandem slave (Hoist_1 and Hoist_3).

There are also other points to consider in terms of interconnection for tandem mode:

- The two variables "TO_tolerancevelocitymonitoring" and "TO_tolerancepositionmonitoring", which are only available for the slave drives in the interface of the MCC unit, must be inserted here for the master drive which is operated as the tandem slave in the interface of the MCC unit. These tolerances are required for comparing the tandem master with the tandem slave.

  Continuing with our example from above (Hoist_1 = tandem master and Hoist_3 = tandem slave), the variables have to be inserted for Hoist_3 in the interface of the MCC unit.
- In tandem mode, the two variables (Hoist_3_tolerancevelocitymonitoring and Hoist_3_tolerancepositionmonitoring) have to be connected with the input variables "r64tolerancevelocitymonitoring" and "r64tolerancepositionmonitoring" in the subroutine call (monitoring).

- For the purpose of variable assignment after the subroutine call, two variables must be created for the tandem master drive in the interface of the MCC unit. One of these variables must be a Boolean variable, which transmits the error signal, and the other must be a DINT variable, which transmits the error code. During variable assignment, errors and error codes from the subroutine call for tandem mode are connected with these two new variables.
- These two new variables then have to be connected to a free "bofb_outerrorbool" and "bofb_outerrorid" position in the "Errorpriority" module of the tandem master in the subroutine call.

A brief sequence when an error is detected is listed in the following graphic. In this example, in master-slave operation, the velocity between the master and the slave is monitored. In point 1, it is identified that the velocities between the master and slave lie outside the permissible tolerance but the velocities lie again within the tolerance before the delay time expires. This is the reason that an error is not output. In point 2, the velocities lie outside the permissible tolerance - even after the delay time has expired. This is the reason that an error is output.

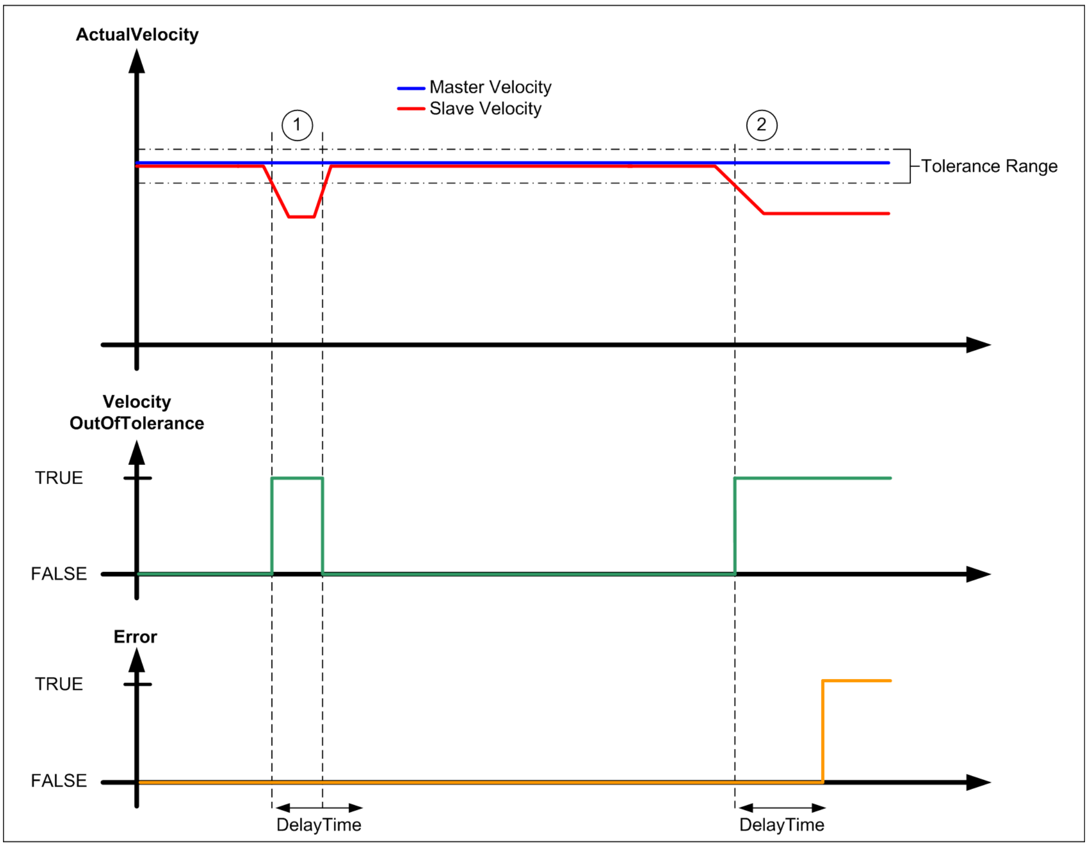


Figure 5-25 Error detection sequence

Error messages

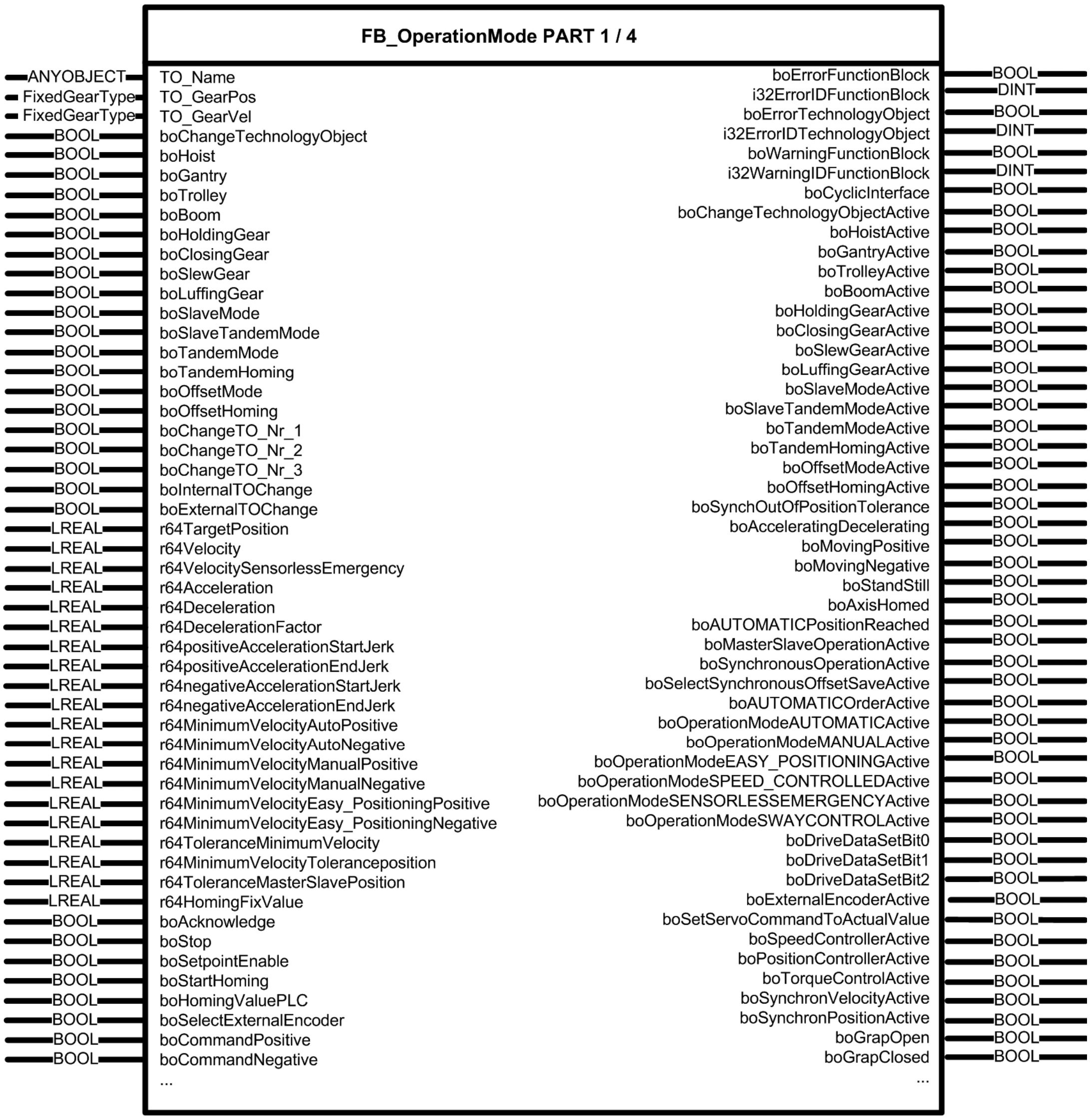
If an error occurs, output "boError" is set and an error number is output at output "i32ErrorID"; see Application error messages and alarm messages (Page 339).

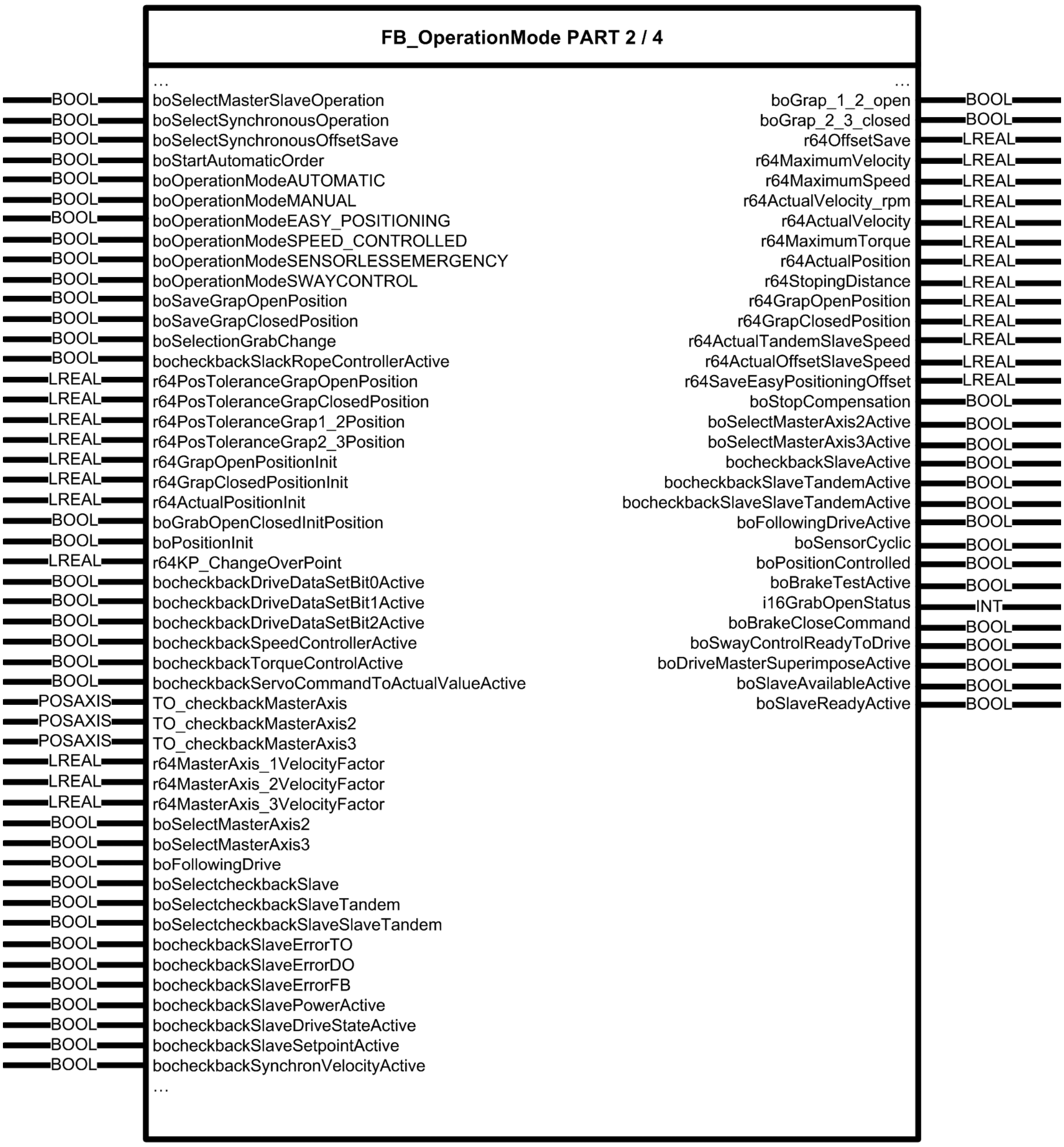
5.3.9 FB_OperationMode

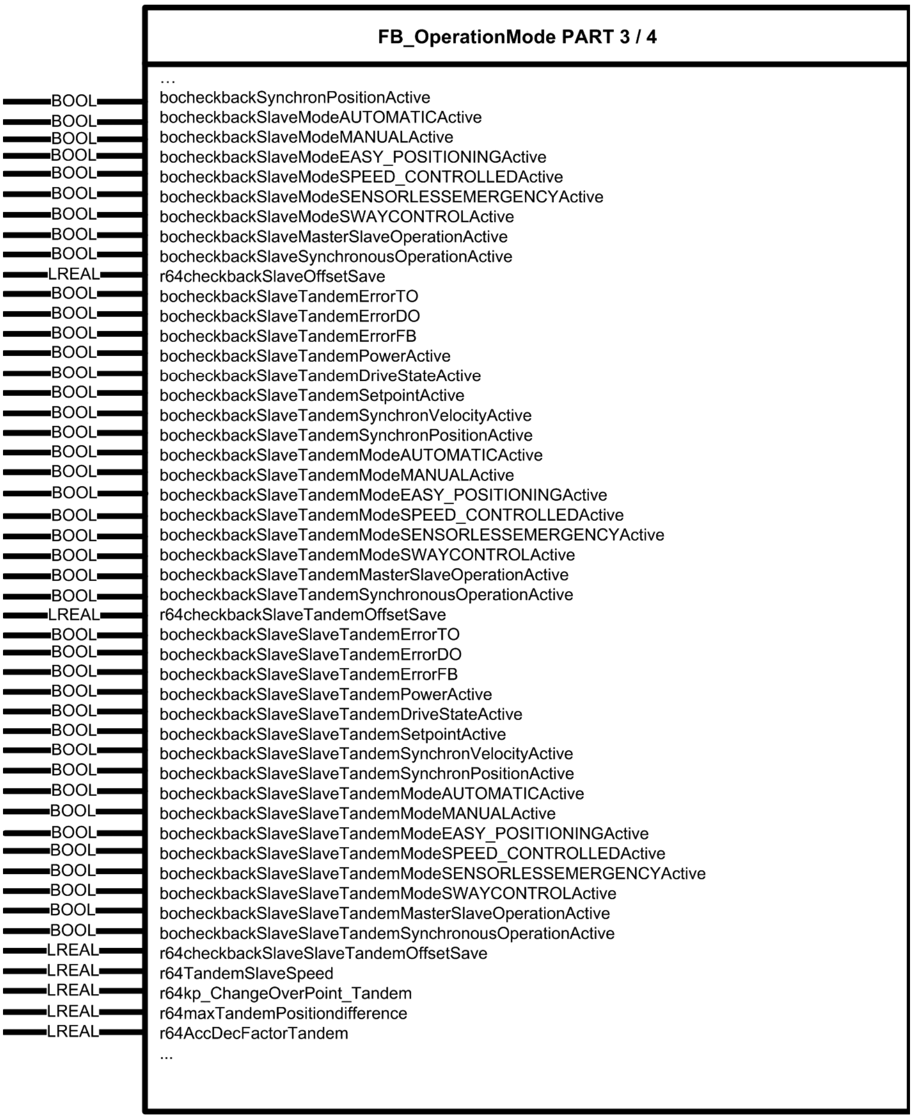
Task

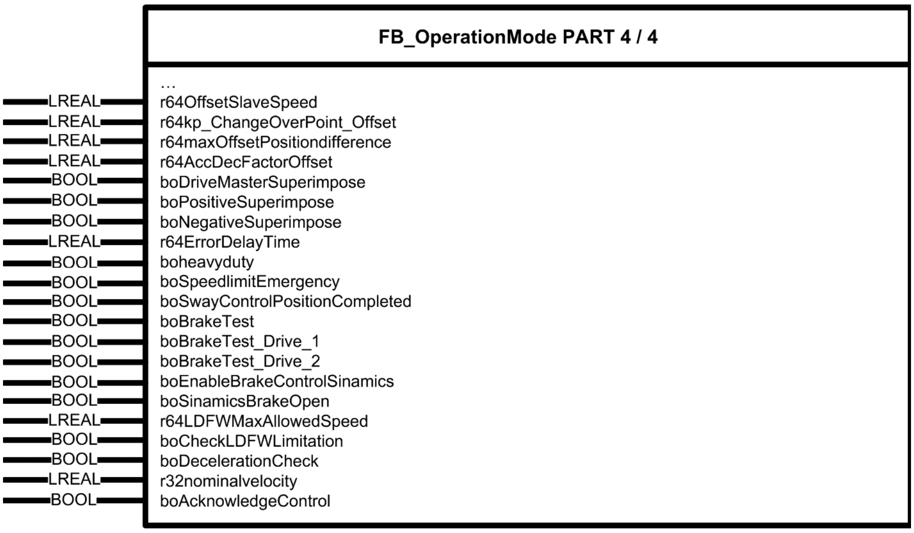
The operating modes and travel motion are managed using the function block FB_OperationMode.

Template for the call


FB_OperationMode PART 1 / 4
ANYOBJECT TO_Name
FixedGearType TO_GearPos
FixedGearType TO_GearVel
BOOL boChangeTechnologyObject
BOOL boHoist
BOOL boGantry
BOOL boTrolley
BOOL boBoom
BOOL boHoldingGear
BOOL boClosingGear
BOOL boSlewGear
BOOL boLuffingGear
BOOL boSlaveMode
BOOL boSlaveTandemMode
BOOL boTandemMode
BOOL boTandemHoming
BOOL boOffsetMode
BOOL boOffsetHoming
BOOL boChangeTO_Nr_1
BOOL boChangeTO_Nr_2
BOOL boChangeTO_Nr_3
BOOL boInternalTOChange
BOOL boExternalTOChange
LREAL r64TargetPosition
LREAL r64Velocity
LREAL r64VelocitySensorlessEmergency
LREAL r64Acceleration
LREAL r64Deceleration
LREAL r64DecelerationFactor
LREAL r64positiveAccelerationStartJerk
LREAL r64positiveAccelerationEndJerk
LREAL r64negativeAccelerationStartJerk
LREAL r64negativeAccelerationEndJerk
LREAL r64MinimumVelocityAutoPositive
LREAL r64MinimumVelocityAutoNegative
LREAL r64MinimumVelocityManualPositive
LREAL r64MinimumVelocityManualNegative
LREAL r64MinimumVelocityEasy_PositioningPositive
LREAL r64MinimumVelocityEasy_PositioningNegative
LREAL r64ToleranceMinimumVelocity
LREAL r64MinimumVelocityToleranceposition
LREAL r64ToleranceMasterSlavePosition
LREAL r64HomingFixValue
BOOL boAcknowledge
BOOL boStop
BOOL boSetpointEnable
BOOL boStartHoming
BOOL boHomingValuePLC
BOOL boSelectExternalEncoder
BOOL boCommandPositive
BOOL boCommandNegative
...
boErrorFunctionBlock BOOL
i32ErrorIDFunctionBlock DINT
boErrorTechnologyObject BOOL
i32ErrorIDTechnologyObject DINT
boWarningFunctionBlock BOOL
i32WarningIDFunctionBlock DINT
boCyclicInterface BOOL
boChangeTechnologyObjectActive BOOL
boHoistActive BOOL
boGantryActive BOOL
boTrolleyActive BOOL
boBoomActive BOOL
boHoldingGearActive BOOL
boClosingGearActive BOOL
boSlewGearActive BOOL
boLuffingGearActive BOOL
boSlaveModeActive BOOL
boSlaveTandemModeActive BOOL
boTandemModeActive BOOL
boTandemHomingActive BOOL
boOffsetModeActive BOOL
boOffsetHomingActive BOOL
boSynchOutOfPositionTolerance BOOL
boAcceleratingDecelerating BOOL
boMovingPositive BOOL
boMovingNegative BOOL
boStandStill BOOL
boAxisHomed BOOL
boAUTOMATICPositionReached BOOL
boMasterSlaveOperationActive BOOL
boSynchronousOperationActive BOOL
boSelectSynchronousOffsetSaveActive BOOL
boAUTOMATICOrderActive BOOL
boOperationModeAUTOMATICActive BOOL
boOperationModeMANUALActive BOOL
boOperationModeEASY_POSITIONINGActive BOOL
boOperationModeSPEED_CONTROLLEDActive BOOL
boOperationModeSENSORLESSEMERGENCYActive BOOL
boOperationModeSWAYCONTROLActive BOOL
boDriveDataSetBit0 BOOL
boDriveDataSetBit1 BOOL
boDriveDataSetBit2 BOOL
boExternalEncoderActive BOOL
boSetServoCommandToActualValue BOOL
boSpeedControllerActive BOOL
boPositionControllerActive BOOL
boTorqueControlActive BOOL
boSynchronVelocityActive BOOL
boSynchronPositionActive BOOL
boGrapOpen BOOL
boGrapClosed BOOL
...


FB_OperationMode PART 2 / 4
…
BOOL boSelectMasterSlaveOperation
BOOL boSelectSynchronousOperation
BOOL boSelectSynchronousOffsetSave
BOOL boStartAutomaticOrder
BOOL boOperationModeAUTOMATIC
BOOL boOperationModeMANUAL
BOOL boOperationModeEASY_POSITIONING
BOOL boOperationModeSPEED_CONTROLLED
BOOL boOperationModeSENSORLESSEMERGENCY
BOOL boOperationModeSWAYCONTROL
BOOL boSaveGrapOpenPosition
BOOL boSaveGrapClosedPosition
BOOL boSelectionGrabChange
BOOL bocheckbackSlackRopeControllerActive
LREAL r64PosToleranceGrapOpenPosition
LREAL r64PosToleranceGrapClosedPosition
LREAL r64PosToleranceGrap1_2Position
LREAL r64PosToleranceGrap2_3Position
LREAL r64GrapOpenPositionInit
LREAL r64GrapClosedPositionInit
LREAL r64ActualPositionInit
BOOL boGrabOpenClosedInitPosition
BOOL boPositionInit
LREAL r64KP_ChangeOverPoint
BOOL bocheckbackDriveDataSetBit0Active
BOOL bocheckbackDriveDataSetBit1Active
BOOL bocheckbackDriveDataSetBit2Active
BOOL bocheckbackSpeedControllerActive
BOOL bocheckbackTorqueControlActive
BOOL bocheckbackServoCommandToActualValueActive
POSAXIS TO_checkbackMasterAxis
POSAXIS TO_checkbackMasterAxis2
POSAXIS TO_checkbackMasterAxis3
LREAL r64MasterAxis_1VelocityFactor
LREAL r64MasterAxis_2VelocityFactor
LREAL r64MasterAxis_3VelocityFactor
BOOL boSelectMasterAxis2
BOOL boSelectMasterAxis3
BOOL boFollowingDrive
BOOL boSelectcheckbackSlave
BOOL boSelectcheckbackSlaveTandem
BOOL boSelectcheckbackSlaveSlaveTandem
BOOL bocheckbackSlaveErrorTO
BOOL bocheckbackSlaveErrorDO
BOOL bocheckbackSlaveErrorFB
BOOL bocheckbackSlavePowerActive
BOOL bocheckbackSlaveDriveStateActive
BOOL bocheckbackSlaveSetpointActive
BOOL bocheckbackSynchronVelocityActive
…
…
boGrap_1_2_open BOOL
boGrap_2_3_closed BOOL
r64OffsetSave LREAL
r64MaximumVelocity LREAL
r64MaximumSpeed LREAL
r64ActualVelocity_rpm LREAL
r64ActualVelocity LREAL
r64MaximumTorque LREAL
r64ActualPosition LREAL
r64StopingDistance LREAL
r64GrapOpenPosition LREAL
r64GrapClosedPosition LREAL
r64ActualTandemSlaveSpeed LREAL
r64ActualOffsetSlaveSpeed LREAL
r64SaveEasyPositioningOffset LREAL
boStopCompensation BOOL
boSelectMasterAxis2Active BOOL
boSelectMasterAxis3Active BOOL
bocheckbackSlaveActive BOOL
bocheckbackSlaveTandemActive BOOL
bocheckbackSlaveSlaveTandemActive BOOL
boFollowingDriveActive BOOL
boSensorCyclic BOOL
boPositionControlled BOOL
boBrakeTestActive BOOL
i16GrabOpenStatus INT
boBrakeCloseCommand BOOL
boSwayControlReadyToDrive BOOL
boDriveMasterSuperimposeActive BOOL
boSlaveAvailableActive BOOL
boSlaveReadyActive BOOL


FB_OperationMode PART 3 / 4
…
BOOL bocheckbackSynchronPositionActive
BOOL bocheckbackSlaveModeAUTOMATICActive
BOOL bocheckbackSlaveModeMANUALActive
BOOL bocheckbackSlaveModeEASY_POSITIONINGActive
BOOL bocheckbackSlaveModeSPEED_CONTROLLEDActive
BOOL bocheckbackSlaveModeSENSORLESSEMERGENCYActive
BOOL bocheckbackSlaveModeSWAYCONTROLActive
BOOL bocheckbackSlaveMasterSlaveOperationActive
BOOL bocheckbackSlaveSynchronousOperationActive
LREAL r64checkbackSlaveOffsetSave
BOOL bocheckbackSlaveTandemErrorTO
BOOL bocheckbackSlaveTandemErrorDO
BOOL bocheckbackSlaveTandemErrorFB
BOOL bocheckbackSlaveTandemPowerActive
BOOL bocheckbackSlaveTandemDriveStateActive
BOOL bocheckbackSlaveTandemSetpointActive
BOOL bocheckbackSlaveTandemSynchronVelocityActive
BOOL bocheckbackSlaveTandemSynchronPositionActive
BOOL bocheckbackSlaveTandemModeAUTOMATICActive
BOOL bocheckbackSlaveTandemModeMANUALActive
BOOL bocheckbackSlaveTandemModeEASY_POSITIONINGActive
BOOL bocheckbackSlaveTandemModeSPEED_CONTROLLEDActive
BOOL bocheckbackSlaveTandemModeSENSORLESSEMERGENCYActive
BOOL bocheckbackSlaveTandemModeSWAYCONTROLActive
BOOL bocheckbackSlaveTandemMasterSlaveOperationActive
BOOL bocheckbackSlaveTandemSynchronousOperationActive
LREAL r64checkbackSlaveTandemOffsetSave
BOOL bocheckbackSlaveSlaveTandemErrorTO
BOOL bocheckbackSlaveSlaveTandemErrorDO
BOOL bocheckbackSlaveSlaveTandemErrorFB
BOOL bocheckbackSlaveSlaveTandemPowerActive
BOOL bocheckbackSlaveSlaveTandemDriveStateActive
BOOL bocheckbackSlaveSlaveTandemSetpointActive
BOOL bocheckbackSlaveSlaveTandemSynchronVelocityActive
BOOL bocheckbackSlaveSlaveTandemSynchronPositionActive
BOOL bocheckbackSlaveSlaveTandemModeAUTOMATICActive
BOOL bocheckbackSlaveSlaveTandemModeMANUALActive
BOOL bocheckbackSlaveSlaveTandemModeEASY_POSITIONINGActive
BOOL bocheckbackSlaveSlaveTandemModeSPEED_CONTROLLEDActive
BOOL bocheckbackSlaveSlaveTandemModeSENSORLESSEMERGENCYActive
BOOL bocheckbackSlaveSlaveTandemModeSWAYCONTROLActive
BOOL bocheckbackSlaveSlaveTandemMasterSlaveOperationActive
BOOL bocheckbackSlaveSlaveTandemSynchronousOperationActive
LREAL r64checkbackSlaveSlaveTandemOffsetSave
LREAL r64TandemSlaveSpeed
LREAL r64kp_ChangeOverPoint_Tandem
LREAL r64maxTandemPositiondifference
LREAL r64AccDecFactorTandem
...


FB_OperationMode PART 4 / 4
…
LREAL r64OffsetSlaveSpeed
LREAL r64kp_ChangeOverPoint_Offset
LREAL r64maxOffsetPositiondifference
LREAL r64AccDecFactorOffset
BOOL boDriveMasterSuperimpose
BOOL boPositiveSuperimpose
BOOL boNegativeSuperimpose
LREAL r64ErrorDelayTime
BOOL boheavyduty
BOOL boSpeedlimitEmergency
BOOL boSwayControlPositionCompleted
BOOL boBrakeTest
BOOL boBrakeTest_Drive_1
BOOL boBrakeTest_Drive_2
BOOL boEnableBrakeControlSinamics
BOOL boSinamicsBrakeOpen
LREAL r64LDFWMaxAllowedSpeed
BOOL boCheckLDFWLimitation
BOOL boDecelerationCheck
LREAL r32nominalvelocity
BOOL boAcknowledgeControl

Connections

Name	Connection type	Default setting	Data type	Meaning
TO_Name	IN	-	AnyObject	Enter the name of the technology object
TO_GearPos	IN	-	FixedGear Type	Enter the name of the fixed gear object position
TO_GearVel	IN	-	FixedGear Type	Enter the name of the fixed gear object velocity
boChangeTechnologyObject	IN	False	BOOL	Selects Change technology object; True means that the relevant technology object is active, false means that it is deactivated.
boHoist	IN	False	BOOL	Selects the Hoist function module
boGantry	IN	False	BOOL	Selects Gantry function module
boTrolley	IN	False	BOOL	Selects Trolley function module
boBoom	IN	False	BOOL	Selects Boom function module
boHoldingGear	IN	False	BOOL	Selects Holding Gear function module
boClosingGear	IN	False	BOOL	Selects Closing Gear function module
boSlewGear	IN	False	BOOL	Selects Slewing Gear function module
boLuffingGear	IN	False	BOOL	Selects Luffing Gear function module
boSlaveMode	IN	False	BOOL	Selects slave operation, otherwise master operation is active.
boSlaveTandemMode	IN	False	BOOL	Selects slave tandem mode, otherwise master tandem mode is active.
boTandemMode	IN	False	BOOL	Selects tandem mode
boTandemHoming	IN	False	BOOL	Starts tandem homing
boOffsetMode	IN	False	BOOL	0 = no offset mode, 1 = offset mode

Name	Connection type	Default setting	Data type	Meaning
boOffsetHoming	IN	False	BOOL	The tandem slave starts the homing process.
boChangeTO_Nr_1	IN	False	BOOL	The technology object works with the drive data set 0/1.
boChangeTO_Nr_2	IN	False	BOOL	The technology object works with the drive data set 2/3.
boChangeTO_Nr_3	IN	False	BOOL	The technology object works with the drive data set 4/5.
boInternalTOChange	IN	False	BOOL	Switchover of the drive data set with an internal TO switchover.
boExternalTOChange	IN	False	BOOL	Switchover of the drive data set with an external TO switchover.
r64TargetPosition	IN	0.0	LREAL	Target position input for the AUTOMATIC and MANUAL operating modes [mm]
r64Velocity	IN	0.0	LREAL	Velocity setpoint [mm/s]
r64VelocitySensorlessEmergency	IN	15.0	LREAL	Velocity limiting SENSORLESS EMERGENCY [mm/s]
r64Acceleration	IN	0.0	LREAL	Acceleration value [mm/s^2]
r64Deceleration	IN	0.0	LREAL	Deceleration value [mm/s^2]
r64DecelerationFactor	IN	1.0	LREAL	The deceleration factor can be used to extend the deceleration time defined at input r64Deceleration. This extension is always active in AUTOMATIC operating mode. In MANUAL operating mode it is only active in the target approach phase. The deceleration time set by the S7 controller is used for braking after the master switch was released. The deceleration factor can be set in the range from 1.0 to 2.0. Lower or higher values are rejected and are limited to 1.0 or 2.0. An input value of 0.5 is limited to 1.0 and an input value of 4.0 is limited to 2.0.
r64positiveAccelerationStartJerk	IN	0.0	LREAL	Initial jerk for the acceleration start [ms] The jerk is specified by the higher-level controller. [ms]
r64positiveAccelerationStartJerk	IN	0.0	LREAL	Final jerk for the acceleration end [ms] The jerk is specified by the higher-level controller. [ms]
r64negativeAccelerationStartJerk	IN	0.0	LREAL	Initial jerk for the start of deceleration [ms]
r64negativeAccelerationEndJerk	IN	0.0	LREAL	End jerk for the end of deceleration [ms]
r64MinimumVelocityAutoPositive	IN	0.0	LREAL	Minimum velocity in the AUTOMATIC operating mode; if a value not equal to zero is entered in this variable, then when the target position is reached, the system changes over to a crawl velocity [mm/s].

Name	Connection type	Default setting	Data type	Meaning
r64MinimumVelocityAutoNegative	IN	0.0	LREAL	Minimum velocity in the AUTOMATIC operating mode; if a value not equal to zero is entered in this variable, then when the target position is reached, the system changes over to a crawl velocity [mm/s].
r64MinimumVelocityManualPositive	IN	0.0	LREAL	Minimum velocity in the MANUAL operating mode; if a value not equal to zero is entered in this variable, then when the target position is reached, the system changes over to a crawl velocity [mm/s].
r64MinimumVelocityManualNegative	IN	0.0	LREAL	Minimum velocity in the MANUAL operating mode; if a value not equal to zero is entered in this variable, then when the target position is reached, the system changes over to a crawl velocity [mm/s].
r64MinimumVelocityEasy_PositioningPositive	IN	0.0	LREAL	Minimum velocity in the EASY_POSITIONING operating mode; if a value not equal to zero is entered in this variable, the system changes over to a crawl velocity [mm/s] when the target position is reached.
r64MinimumVelocityEasy_PositioningNegative	IN	0.0	LREAL	Minimum velocity in the EASY_POSITIONING operating mode; if a value not equal to zero is entered in this variable, the system changes over to a crawl velocity [mm/s] when the target position is reached.
r64ToleranceMinimumVelocity	IN	0.0	LREAL	Override point at which percentage (0 - 500 %) above the minimum velocity the switchover to minimum velocity must be triggered. This allows for constant velocity transitions.
r64MinimumVelocityToleranceposition	IN	200.0	LREAL	Starting at this residual position, a switch is made to the minimum velocity.
r64ToleranceMasterSlavePosition	IN	100	LREAL	Position difference that can be tolerated [mm] between the master (master drive) and slave (slave drive) (in closed-loop position controlled operation).
r64HomingFixValue	IN	0.0	LREAL	Fixed homing position [mm]
boAcknowledge	IN	False	BOOL	Acknowledge error
boStop	IN	False	BOOL	Stops the axis
boSetpointEnable	IN	False	BOOL	Setpoint enable
boStartHoming	IN	False	BOOL	Starts the homing operation
boHomingValuePLC	IN	False	BOOL	S7 sets the homing position. If the input is statically set to TRUE, the homing position is derived from variable r64TargetPosition. Otherwise, the homing position is taken from input r64HomingFixValue.
boSelectExternalEncoder	IN	False	BOOL	Selects the encoder switchover from motor encoder to external encoder

Name	Connection type	Default setting	Data type	Meaning
boCommandPositive	IN	False	BOOL	Selects positive travel direction.
boCommandNegative	IN	False	BOOL	Selects negative travel direction.
boSelectMasterSlaveOperation	IN	False	BOOL	Selecting master-slave operation
boSelectSynchronousOperation	IN	False	BOOL	Selecting synchronous operation
boSelectSynchronousOffsetSave	IN	False	BOOL	Selects save offset
boStartAutomaticOrder	IN	False	BOOL	Starts the travel request in the AUTOMATIC operating mode
boOperationModeAUTOMATIC	IN	False	BOOL	Selects AUTOMATIC operating mode
boOperationModeMANUAL	IN	False	BOOL	Selection MANUAL operating mode
boOperationModeEasy_Positioning	IN	False	BOOL	Selects EASY_POSTIONING mode selection
boOperationModeSPEED_CONTROLLED	IN	False	BOOL	Selects SPEED_CONTROLLED mode
boOperationModeSENSORLESSEMERGENCY	IN	False	BOOL	Selects SENSORLESS EMERGENCY operating mode
boOperationModeSWAYCONTROL	IN	False	BOOL	Selects SWAYCONTROL operating mode
boSaveGrabOpenPosition	IN	False	BOOL	Save position Grab Open
boSaveGrabClosedPosition	IN	False	BOOL	Save position Grab Closed
boSelectionGrabChange	IN	False	BOOL	Change Grab command
bocheckbackSlackRopeControllerActive	IN	False	BOOL	Checkback signal from master indicating whether slack rope controller is active
r64PosToleranceGrabOpenPosition	IN	100.0	LREAL	Tolerance for message Grab Open [mm]
r64PosToleranceGrabClosedPosition	IN	100.0	LREAL	Tolerance for message Grab Closed [mm]
r64PosToleranceGrab1_2Position	IN	100.0	LREAL	Tolerance for message Grab 1/2 Open [mm]
r64PosToleranceGrab2_3Position	IN	100.0	LREAL	Tolerance for message Grab 2/3 Closed [mm]
r64GrapOpenPositionInit	IN	0.0	LREAL	Sets the position for the open state of the grab [mm] for initialization once the controller has been powered up.
r64GrapClosedPositionInit	IN	0.0	LREAL	Sets the position for the closed state of the grab [mm] for initialization once the controller has been powered up.
r64ActualPositionInit	IN	0.0	LREAL	Sets the actual position of the grab [mm] for initialization once the controller has been powered up.
boGrabOpenClosedInitPosition	IN	False	BOOL	When selected, the position of the grab once the controller has been powered up is initialized.
boPositionInit	IN	False	BOOL	When selected, the actual position of the grab is initialized.
r64KP_ChangeOverPoint	IN	0.5	LREAL	Gain factor for changeover point delta s

Name	Connection type	Default setting	Data type	Meaning
bocheckbackDriveDataSetBit0Active	IN	False	BOOL	Feedback signal drive data set bit 0 active from this technology object.
bocheckbackDriveDataSetBit1Active	IN	False	BOOL	Feedback signal drive data set bit 1 active from this technology object.
bocheckbackDriveDataSetBit2Active	IN	False	BOOL	Checkback signal drive data set bit 2 active from this technology object.
bocheckbackSpeedControllerActive	IN	False	BOOL	Feedback signal speed controller active from this technology object.
bocheckbackTorqueControlActive	IN	False	BOOL	Feedback signal torque control active from this technology object.
bocheckbackServoCommandToActualValueActive	IN	False	BOOL	Feedback signal position controller follow-up mode active from this technology object.
TO_checkbackMasterAxis	IN	-	PosAxis	TO master1 name for synchronous operation
TO_checkbackMasterAxis2	IN	-	PosAxis	TO master2 name for synchronous operation
TO_checkbackMasterAxis3	IN	-	PosAxis	TO master3 name for synchronous operation
r64MasterAxis_1VelocityFactor	IN	1.0	LREAL	Velocity factor between master1 and its slave
r64MasterAxis_2VelocityFactor	IN	1.0	LREAL	Velocity factor between master2 and its slave
r64MasterAxis_3VelocityFactor	IN	1.0	LREAL	Velocity factor between master3 and its slave
boSelectMasterAxis2	IN	False	BOOL	Selection of the master
boSelectMasterAxis3	IN	False	BOOL	
boFollowingDrive	IN	False	BOOL	When selected, the master-slave connection is established; otherwise synchronous operation is used, if selected.
boSelectcheckbackSlave	IN	False	BOOL	When selected, all of the slave checkback signals are displayed.
boSelectcheckbackSlaveTandem	IN	False	BOOL	When selected, all of the slave tandem checkback signals are displayed.
boSelectcheckbackSlaveSlaveTandem	IN	False	BOOL	When selected, all of the slave slave tandem checkback signals are displayed.
bocheckbackSlaveErrorTO	IN	False	BOOL	Feedback signal error present at the technology object slave
bocheckbackSlaveErrorDO	IN	False	BOOL	Checkback signal error present at the drive object slave
bocheckbackSlaveErrorFB	IN	False	BOOL	Checkback signal error present in a function block (slave)
bocheckbackSlavePowerActive	IN	False	BOOL	Feedback signal power enable active at the technology object slave
bocheckbackSlaveDriveStateActive	IN	False	BOOL	Feedback signal drive enable active at the technology object slave

Name	Connection type	Default setting	Data type	Meaning
bocheckbackSlaveSetpointActive	IN	False	BOOL	Checkback signal setpoint enable active at the technology object slave
bocheckbackSynchronVelocityActive	IN	False	BOOL	Checkback signal velocity gearing active at the technology object slave
bocheckbackSynchronPositionActive	IN	False	BOOL	Feedback signal position synchronous operation active at the technology object slave
bocheckbackSlaveModeAUTOMATICActive	IN	False	BOOL	Feedback signal AUTOMATIC operating mode active at the technology object slave
bocheckbackSlaveModeMANUALActive	IN	False	BOOL	Feedback signal MANUAL operating mode active at the technology object slave
bocheckbackSlaveModeEASY_POSITIONINGActive	IN	False	BOOL	EASY_POSITIONING operating mode feedback signal active at the slave technology object
bocheckbackSlaveModeSPEED_CONTROLLEDActive	IN	False	BOOL	Feedback signal SPEED_CONTROLLED operating mode active at the technology object slave
bocheckbackSlaveModeSENSORLESSEMERGENCYActive	IN	False	BOOL	Feedback signal SENSORLESS EMERGENCY operating mode active at the technology object slave
bocheckbackSlaveModeSWAYCONTROLActive	IN	False	BOOL	Checkback signal SWAYCONTROL operating mode active at the technology object slave
bocheckbackSlaveMasterSlaveOperationActive	IN	False	BOOL	Feedback signal master-slave operation active at the technology object slave
bocheckbackSlaveSynchronousOperationActive	IN	False	BOOL	Checkback signal synchronous operation active at the technology object slave
r64checkbackSlaveOffsetSave	IN	0.0	LREAL	Feedback value saved offset [mm] at the technology object slave
bocheckbackSlaveTandemErrorTO	IN	False	BOOL	Feedback signal error present at the technology object slave tandem
bocheckbackSlaveTandemErrorDO	IN	False	BOOL	Checkback signal error present at the drive object slave tandem
bocheckbackSlaveTandemErrorFB	IN	False	BOOL	Checkback signal error present in a function block (slave tandem)
bocheckbackSlaveTandemPowerActive	IN	False	BOOL	Checkback signal power enable active at the technology object slave tandem
bocheckbackSlaveTandemDriveStateActive	IN	False	BOOL	Checkback signal drive enable active at the technology object slave tandem
bocheckbackSlaveTandemSetpointActive	IN	False	BOOL	Checkback signal setpoint enable active at the technology object slave tandem
bocheckbackSlaveTandemSynchronVelocityActive	IN	False	BOOL	Checkback signal velocity gearing active at the technology object slave tandem
bocheckbackSlaveTandemSynchronPositionActive	IN	False	BOOL	Checkback signal position synchronous operation active at the technology object slave tandem

Name	Connection type	Default setting	Data type	Meaning
bocheckbackSlaveTandemModeAUTOMATICActive	IN	False	BOOL	Checkback signal AUTOMATIC operating mode active at the technology object slave tandem
bocheckbackSlaveTandemModeMANUALActive	IN	False	BOOL	Checkback signal MANUAL operating mode active at the technology object slave tandem
bocheckbackSlaveTandemModeEASY_POSITIONINGActive	IN	False	BOOL	EASY_POSITIONING operating mode feedback signal active at the slave tandem technology object
bocheckbackSlaveTandemModeSPEED_CONTROLLEDActive	IN	False	BOOL	Checkback signal SPEED_CONTROLLED operating mode active at the technology object slave tandem
bocheckbackSlaveTandemModeSENSORLESSEMERGENCYActive	IN	False	BOOL	Checkback signal SENSORLESS EMERGENCY operating mode active at the technology object slave tandem
bocheckbackSlaveTandemModeSWAYCONTROLActive	IN	False	BOOL	Checkback signal SWAYCONTROL operating mode active at the technology object slave tandem
bocheckbackSlaveTandemMasterSlaveOperationActive	IN	False	BOOL	Checkback signal master-slave operation active at the technology object slave tandem
bocheckbackSlaveTandemSynchronousOperationActive	IN	False	BOOL	Checkback signal synchronous operation active at the technology object slave tandem
r64checkbackSlaveTandemOffsetSave	IN	0.0	LREAL	Checkback value saved offset [mm] at the technology object slave tandem
bocheckbackSlaveSlaveTandemErrorTO	IN	False	BOOL	Checkback signal error present at the technology object slave slave tandem
bocheckbackSlaveSlaveTandemErrorDO	IN	False	BOOL	Checkback signal error present at the drive object slave slave tandem
bocheckbackSlaveSlaveTandemErrorFB	IN	False	BOOL	Checkback signal error present in a function block (slave slave tandem)
bocheckbackSlaveSlaveTandemPowerActive	IN	False	BOOL	Checkback signal power enable active at the technology object slave slave tandem
bocheckbackSlaveSlaveTandemDriveStateActive	IN	False	BOOL	Checkback signal drive enable active at the technology object slave slave tandem
bocheckbackSlaveSlaveTandemSetpointActive	IN	False	BOOL	Checkback signal setpoint enable active at the technology object slave slave tandem
bocheckbackSlaveSlaveTandemSynchronVelocityActive	IN	False	BOOL	Checkback signal velocity gearing active at the technology object slave slave tandem
bocheckbackSlaveSlaveTandemSynchronPositionActive	IN	False	BOOL	Checkback signal position synchronous operation active at the technology object slave slave tandem
bocheckbackSlaveSlaveModeAUTOMATICActive	IN	False	BOOL	Checkback signal AUTOMATIC operating mode active at the technology object slave slave tandem

Name	Connection type	Default setting	Data type	Meaning
bocheckbackSlaveSlaveTandemModeMANUALActive	IN	False	BOOL	Checkback signal MANUAL operating mode active at the technology object slave slave tandem
bocheckbackSlaveSlaveTandemModeEASY_POSITIONINGActive	IN	False	BOOL	EASY_POSITIONING operating mode feedback signal active at the slave slave tandem technology object
bocheckbackSlaveSlaveTandemModeSPEED_CONTROLLEDActive	IN	False	BOOL	Checkback signal SPEED_CONTROLLED operating mode active at the technology object slave slave tandem
bocheckbackSlaveSlaveTandemModeSENSORLESSEMERGENCYActive	IN	False	BOOL	Checkback signal SENSORLESS EMERGENCY operating mode active at the technology object slave slave tandem
bocheckbackSlaveSlaveTandemModeSWAYCONTROLActive	IN	False	BOOL	Checkback signal SWAYCONTROL operating mode active at the technology object slave slave tandem
bocheckbackSlaveSlaveTandemMasterSlaveOperationActive	IN	False	BOOL	Checkback signal master-slave operation active at the technology object slave slave tandem
bocheckbackSlaveSlaveTandemSynchronousOperationActive	IN	False	BOOL	Checkback signal synchronous operation active at the technology object slave slave tandem
r64checkbackSlaveSlaveTandemOffsetSave	IN	0.0	LREAL	Checkback value saved offset [mm] at the technology object slave slave tandem
r64TandemSlaveSpeed	IN	0.0	LREAL	Velocity [%] for tandem-slave from the MaximumVelocity up to max. 50 %
r64kp_ChangeOverPoint_Tandem	IN	0.5	LREAL	Gain factor for changeover point delta s
r64maxTandemPositiondifference	IN	0.0	LREAL	Maximum position difference [mm] between tandem master and tandem slave
r64AccDecFactorTandem	IN	0.01	LREAL	Acceleration and deceleration factor [%] for the superimposed motion of the tandem from the actual acceleration
r64OffsetSlaveSpeed	IN	0.0	LREAL	Velocity [%] of the slave with offset, if it performs a superimposed motion - from the MaximumVelocity up to max. 50%.
r64kp_ChangeOverPoint_Offset	IN	0.5	LREAL	Gain factor for the changeover point of the clearance controller
r64maxOffsetPositiondifference	IN	0.0	LREAL	Maximum position deviation [mm] between offset master and offset slave
r64AccDecFactorOffset	IN	0.01	LREAL	Acceleration and deceleration factor [%] for the superimposed motion of the actual acceleration
boDriveMasterSuperimpose	IN	False	BOOL	Must be set, if the master is also to be moved with superimposition
boPositiveSuperimpose	IN	False	BOOL	Positive, superimposed travel
boNegativeSuperimpose	IN	False	BOOL	Negative, superimposed travel
r64ErrorDelayTime	IN	2000.0	LREAL	An error message becomes active after the set time [ms].
boheavyduty	IN	False	BOOL	Heavy-duty operation is active.

Name	Connection type	Default setting	Data type	Meaning
boSpeedlimitEmergency	IN	False	BOOL	Limits the setpoint of the position controller
boSwayControlPositionCompleted	IN	False	BOOL	A high signal indicates that positioning in SwayControl mode is complete. This input has to be linked to the Sway Control block's corresponding "Positioning completed" feedback signal on Hoist and on Trolley.
boBrakeTest	IN	False	BOOL	Select brake test
boBrakeTest_Drive_1	IN	False	BOOL	Brake test for the technology object can be performed with drive data set 6.
boBrakeTest_Drive_2	IN	False	BOOL	Brake test for the technology object can be performed with drive data set 7.
boEnableBrakeControlSinamics	IN	False	BOOL	Note, select brake checkback signal
boSinamicsBrakeOpen	IN	False	BOOL	Brake checkback signal from SINAMICS
r64LDFWMaxAllowedSpeed	IN	50.0	LREAL	Maximum permissible velocity of the LDFW block
boCheckLDFWLimitation	IN	True	BOOL	Enable of the tandem velocity for the LDFW block
boDecelerationCheck	IN	False	BOOL	Collision protection is activated.
r32nominalvelocity	IN	500.0	REAL	Rated speed in the unit (mm/s)
boAcknowledgeControl	IN	False	BOOL	Error acknowledgment by the FB_ControlAxis block.
boErrorFunctionBlock	OUT	False	BOOL	Error bit from the function block Mode
i32errorIDFunctionBlock	OUT	0	DINT	Error number from the function block Mode
boErrorTechnologyObject	OUT	False	BOOL	Error bit from the technology object
i32errorIDTechnologyObject	OUT	0	DINT	Error number from the technology object
boWarningfunctionblock	OUT	False	BOOL	Alarm bit from FB_OperationMode block
i32WarningIDfunctionblock	OUT	0	DINT	Alarm number from FB_OperationMode block
boCyclicInterface	OUT	False	BOOL	The output variable indicates at the axis that the drive is operating in cyclic mode.
boChangeTechnologyObjectActive	OUT	False	BOOL	Technology object active
boHoistActive	OUT	False	BOOL	Hoist module active
boGantryActive	OUT	False	BOOL	Gantry module active
boTrolleyActive	OUT	False	BOOL	Trolley module active
boBoomActive	OUT	False	BOOL	Boom module active
boHoldingGearActive	OUT	False	BOOL	Holding Gear module
boClosingGearActive	OUT	False	BOOL	Closing Gear module
boSlewGearActive	OUT	False	BOOL	Slewing Gear module
boLuffingGearActive	OUT	False	BOOL	Luffing Gear module
boSlaveModeActive	OUT	False	BOOL	Slave operation active, otherwise master operation is active
boSlaveTandemModeActive	OUT	False	BOOL	A low signal indicates that the master tandem is active, a high signal indicates that the slave tandem is active.

Name	Connection type	Default setting	Data type	Meaning
boTandemModeActive	OUT	False	BOOL	Tandem mode is activated.
boTandemHomingActive	OUT	False	BOOL	The tandem slave is currently performing a homing operation.
boOffsetModeActive	OUT	False	BOOL	0 = offset mode not active, 1 = offset mode active
boOffsetHomingActive	OUT	False	BOOL	The offset slave is currently performing a homing operation.
boSynchOutOfPositionTolerance	OUT	False	BOOL	Synchronous operation, the position between the master (master drive) and slave (slave drive) lies outside the permissible position tolerance.
boAcceleratingDecelerating	OUT	False	BOOL	Acceleration or deceleration is active.
boMovingPositive	OUT	False	BOOL	Technology object moves in the positive direction.
boMovingNegative	OUT	False	BOOL	Technology object moves in the negative direction.
boStandStill	OUT	False	BOOL	Technology object at a standstill, the axis is stationary.
boAxisHomed	OUT	False	BOOL	Technology object is homed.
boAUTOMATICPositionReached	OUT	False	BOOL	Target of the AUTOMATIC travel request reached.
boMasterSlaveOperationActive	OUT	False	BOOL	Master-slave operation active
boSynchronousOperationActive	OUT	False	BOOL	Synchronous operation active
boSelectSynchronousOffsetSaveActive	OUT	False	BOOL	Saved offset active
boAUTOMATICOrderActive	OUT	False	BOOL	AUTOMATIC travel request being processed
boOperationModeAUTOMATICActive	OUT	False	BOOL	AUTOMATIC operating mode active
boOperationModeMANUALActive	OUT	False	BOOL	MANUAL operating mode active
boOperationModeEASY_POSTIONINGActive	OUT	False	BOOL	EASY_POSITIONING operating mode active
boOperationModeSPEED_CONTROLLEDActive	OUT	False	BOOL	SPEED_CONTROLLED operating mode active
boOperationModeSENSORLESSEMERGENCYActive	OUT	False	BOOL	SENSORLESS EMERGENCY operating mode active
boOperationModeSWAYCONTROLActive	OUT	False	BOOL	SWAYCONTROL operating mode active
boDriveDataSetBit0	OUT	False	BOOL	Drive data set bit 0
boDriveDataSetBit1	OUT	False	BOOL	Drive data set bit 1
boDriveDataSetBit2	OUT	False	BOOL	Drive data set bit 2
boExternalEncoderActive	OUT	False	BOOL	External encoder active
boSetservoCommandToActualValue	OUT	False	BOOL	Position controller follow-up mode active
boSpeedControllerActive	OUT	False	BOOL	Closed-loop speed control active

Name	Connection type	Default setting	Data type	Meaning
boPositionControllerActive	OUT	False	BOOL	Closed-loop position control active
boTorqueControlActive	OUT	False	BOOL	Torque control active
boSynchronVelocityActive	OUT	False	BOOL	Velocity synchronous operation active
boSynchronPositionActive	OUT	False	BOOL	Position synchronous operation active
boGrabOpen	OUT	False	BOOL	Message, grab open
boGrabClosed	OUT	False	BOOL	Message, grab closed
boGrab_1_2_open	OUT	False	BOOL	Message, grab 1/2 open
boGrab_2_3_closed	OUT	False	BOOL	Message, grab 2/3 closed
r64offsetSave	OUT	0.0	LREAL	Saved offset value
r64maximumVelocity	OUT	0.0	LREAL	Maximum velocity in the units mm/s
r64maximumSpeed	OUT	0.0	LREAL	Maximum velocity in the units revolutions per minute [rpm]
r64ActualVelocity_rpm	OUT	0.0	LREAL	Actual velocity in the units revolutions per minute [rpm]
r64ActualVelocity	OUT	0.0	LREAL	Actual velocity in the units mm/s
r64maximumTorque	OUT	0.0	LREAL	Maximum torque in the units Newton meter [N/m]
r64ActualPosition	OUT	0.0	LREAL	Actual position in the units millimeters [mm]
r64stopingDistance	OUT	0.0	LREAL	Braking distance is displayed
r64GrabOpenPosition	OUT	0.0	LREAL	Grab open position
r64GrabClosedPosition	OUT	0.0	LREAL	Grab closed position
r64ActualTandemSlaveSpeed	OUT	0.0	LREAL	Actual additional velocity [mm/s] of the slave tandem for the superimposed motion
r64ActualOffsetSlaveSpeed	OUT	0.0	LREAL	Actual additional velocity of the superimposed motion of the slave
r64SaveEasyPositioningOffset	OUT	0.0	LREAL	Saved offset position in tandem operation in the EASY_POSITIONING operating mode
boStopCompensation	OUT	False	BOOL	Stop superimposed traversing motion
boSelectMasterAxis2Active	OUT	False	BOOL	The setpoint of master2 is active.
boSelectMasterAxis3Active	OUT	False	BOOL	The setpoint of master3 is active.
bocheckbackSlaveActive	OUT	False	BOOL	Monitoring of checkback signals from slave 1 is active.
bocheckbackSlaveTandemActive	OUT	False	BOOL	Monitoring of checkback signals from slave 2 is active.
bocheckbackSlaveSlaveTandemActive	OUT	False	BOOL	Monitoring of checkback signals from slave 3 is active.
boFollowingDriveActive	OUT	False	BOOL	Monitoring of checkback signals from the following axis is active.
boSensorCyclic	OUT	False	BOOL	Cyclic communication with the encoder
bopositioncontrolled	OUT	False	BOOL	Closed-loop position controller active
boBrakeTestActive	OUT	False	BOOL	Brake test selection
i16GrabOpenStatus	OUT	0	INT	Status "Grab opened" as a %

Name	Connection type	Default setting	Data type	Meaning
boBrakeCloseCommand	OUT	False	BOOL	"Close brake" command from SIMOTION, if it does not come from the S7.
boSwayControlReadyToDrive	OUT	False	BOOL	Brake open for sway control
boDriveMasterSuperimposeActive	OUT	False	BOOL	Superimposed movement of the master is selected.
boSlaveAvailableActive	OUT	False	BOOL	Technology object slave present active
boSlaveReadyActive	OUT	False	BOOL	Slave is okay.

CAUTION

The following generally applies to every function module:

For the external switchover, boChangeTechnologyObject = TRUE must be given so that the setpoints can be passed to the technology object. If boChangeTechnologyObject is set to False, the relevant technology object is deactivated and cannot accept any values for the drive.

This does not apply to the internal data set switchover. In this case:

- boChangeTechnologyObject = False: 1. Drive is active.
- boChangeTechnologyObject = True: 2. Drive is active.

5.3.9.1 Functionality

SIMOTION technology object data sets

Two technology object data sets can be set up in SIMOTION when required for hoist, gantry, trolley, etc. The data set switchover is required to be able to switch over between a motor encoder and external encoder for the position actual value sensing at SIMOTION.

TO data set 1:	Hoist, Gantry, Trolley, Boom, Holding Gear, Closing Gear, Slewing Gear The motor encoder evaluates the position actual value.
TO data set 2: (when required):	Hoist, Gantry, Trolley, Holding Gear, Closing Gear, Slewing Gear The external encoder evaluates the position actual value.

Note

TO data set 2 is used only when using an external encoder, if this encoder is directly connected to SIMOTION / SINAMICS. The external encoder is used for positioning.

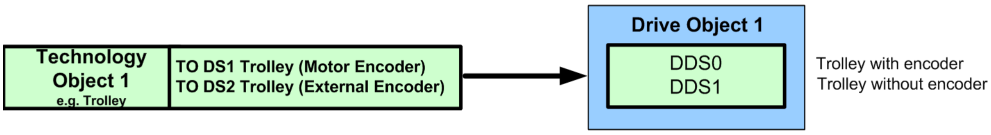


Figure 5-26 External encoder in TO

SINAMICS drive object data sets

In SINAMICS, two drive object data sets must be set up for the Hoist, Gantry, etc. function modules. The second data set (DO data set 1) is required only for drives **without** active load for speed control in the SENSORLESS EMERGENCY operating mode.

"Active load" means that the load for opened brake accelerates automatically (Hoist, Boom, Holding Gear, etc.).

DO data set 0:	Hoist, Gantry, Holding Gear, Closing Gear, Slewing Gear Closed-loop speed control with motor encoder
DO data set 1:	Gantry, Trolley, Slewing Gear Closed-loop speed control without motor encoder

Analog to this, further drive object data sets must be created in SINAMICS for the associated switchover options. Where other switchover options are required, e.g. Gantry to Boom or Gantry to Hoist, the relevant data sets must also be set up based on the example illustrated here.

DO data set 0:	Trolley Closed-loop speed control with motor encoder
DO data set 1:	Trolley Closed-loop speed control without motor encoder
DO data set 2:	Boom Closed-loop speed control with motor encoder
DO data set 3:	Boom Closed-loop speed control with motor encoder (because active load)
DO data set 4:	Gantry Closed-loop speed control with motor encoder
DO data set 5:	Gantry Closed-loop speed control without motor encoder

Note

DO data sets 4 and 5 can be selected only for the external TO switchover (refer to Chapter Technology object switchover (Page 214)). They cannot be selected for the internal TO switchover. Data sets 4 and 5 must be created for both TO switchover variants when a brake test is required. A brake test is possible only with data set 6 and/or data set 7.

Note

DO data set 6 or 7 is used for the brake test.
The brake test is implemented in the standard "STS Crane" application, refer to STS crane (ship-to-shore) (Page 498)).

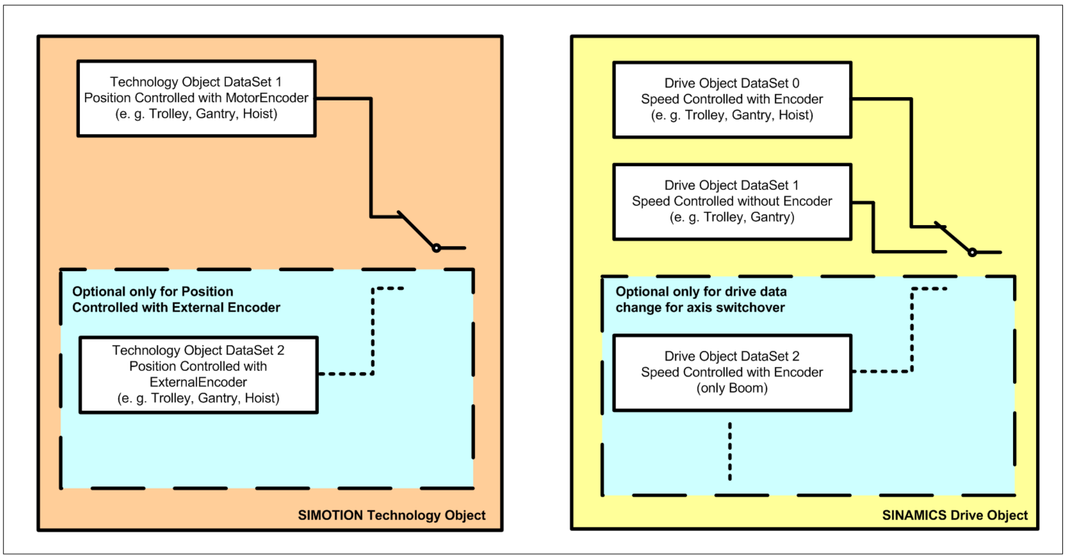

Technology Object DataSet 1
Position Controlled with MotorEncoder
(e. g. Trolley, Gantry, Hoist)
Optional only for Position
Controlled with External Encoder
Technology Object DataSet 2
Position Controlled with
ExternalEncoder
(e. g. Trolley, Gantry, Hoist)
SIMOTION Technology Object
Drive Object DataSet 0
Speed Controlled with Encoder
(e. g. Trolley, Gantry, Hoist)
Drive Object DataSet 1
Speed Controlled without Encoder
(e. g. Trolley, Gantry)
Optional only for drive data
change for axis switchover
Drive Object DataSet 2
Speed Controlled with Encoder
(only Boom)
SINAMICS Drive Object

Figure 5-27 FB_OperationMode, SIMOTION and SINAMICS data sets

5.3.9.2 Selecting a function module

The function module must only be defined once in the commissioning phase and then it may no longer be changed. At this point, it must be decided whether the module is a Hoist, Gantry, Trolley, Boom, Holding Gear, Closing Gear or Slewing Gear.

Prerequisites for selection without function module changeover:

- The drive has no power enable signal (power = INACTIVE)
- The drive is stationary (boStandstill = TRUE)
- The plausibility check has been completed successfully

Table 5- 8 Selecting/deselecting function modules

			①: Selection	②: Deselection
Function module	Connection type	Name	Feedback signal	Feedback signal
Hoist	IN	boHoist	TRUE	FALSE
	OUT	boHoistActive	TRUE	FALSE
Gantry	IN	boGantry	TRUE	FALSE
	OUT	boGantryActive	TRUE	FALSE
Trolley	IN	boTrolley	TRUE	FALSE

			①: Selection	②: Deselection
Function module	Connection type	Name	Feedback signal	Feedback signal
	OUT	boTrolleyActive	TRUE	FALSE
Boom	IN	boBoom	TRUE	FALSE
	OUT	boBoomActive	TRUE	FALSE
Holding Gear	IN	boHoldingGear	TRUE	FALSE
	OUT	boHoldingGearActive	TRUE	FALSE
Closing Gear	IN	boClosingGear	TRUE	FALSE
	OUT	boClosingGearActive	TRUE	FALSE
Slewing Gear	IN	boSlewGear	TRUE	FALSE
	OUT	boSlewGearActive	TRUE	FALSE

The Hoist and Gantry function modules will now be explained in more detail using an example.

Hoist active

The control and feedback signals are shown in the following sequence diagram. You will see the selection as case ① and de-selection as case ②.

Case 1:
Output "boHoistActive" is set, if, when selecting input "boHoist" the plausibility check is OK.

Case 2:
Output "boHoistActive" is reset if, when de-selecting the input "boHoist" the plausibility check is OK.

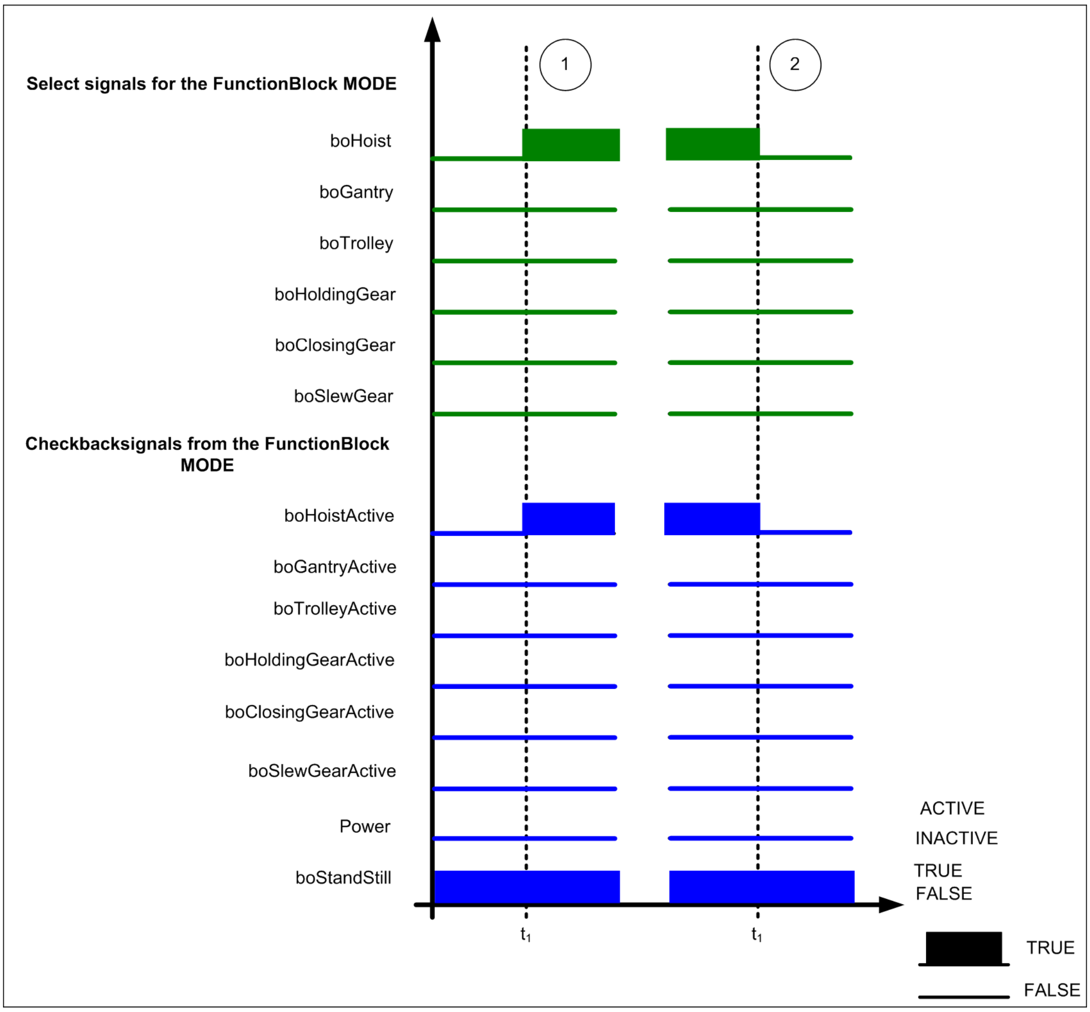


Figure 5-28 FB_OperationMode, Hoist active

Gantry active

The control and feedback signals are shown in the following sequence diagram. You will see the selection as case ① and de-selection as case ②.

Case 1:
Output "boGantryActive" is set if, when selecting the input "boGantry" the plausibility check is OK.

Case 2:
Output "boGantryActive" is reset if, when de-selecting the input "boGantry" the plausibility check is OK.

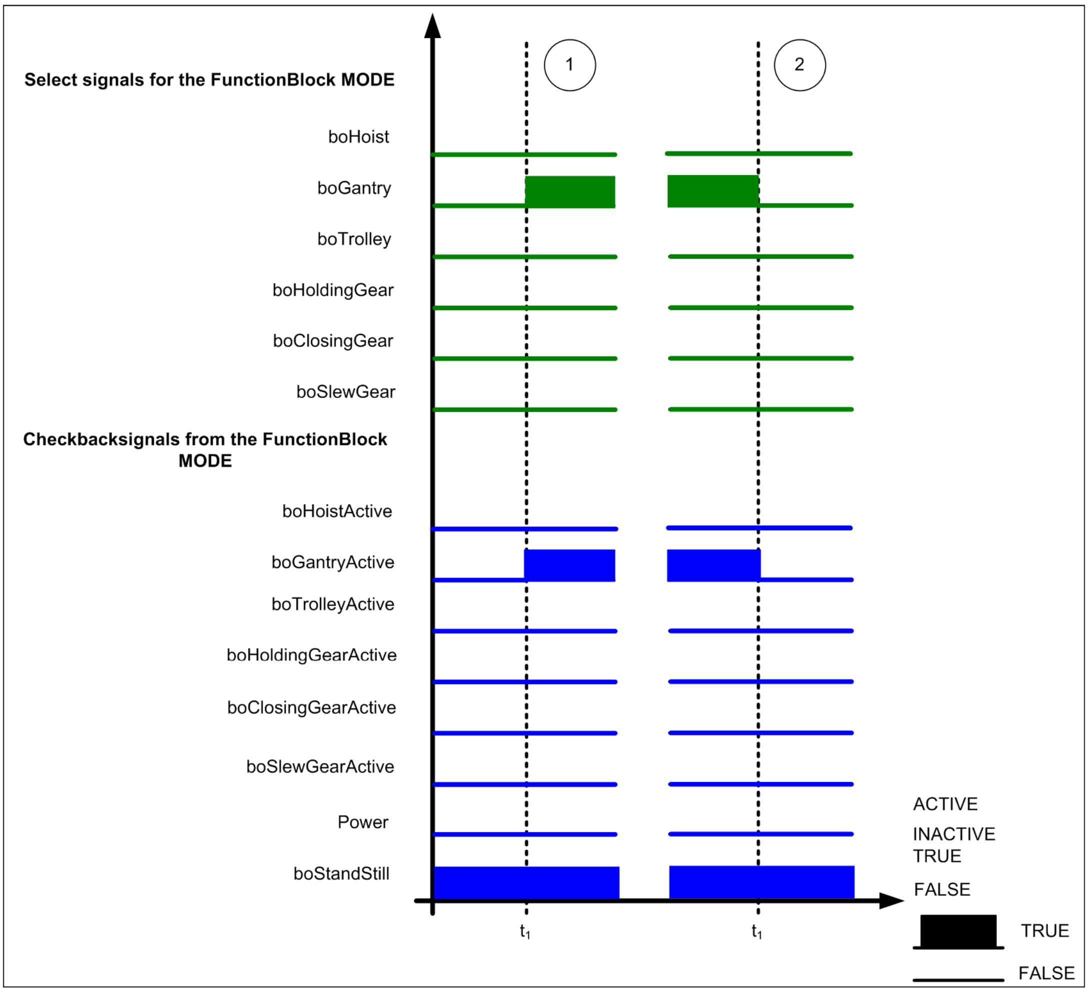


Figure 5-29 FB_OperationMode, Travel gear active

5.3.9.3 Technology object switchover

The technology object switchover (TO switchover) enables different motors to be driven alternately via one Motor Module, e.g. Trolley and Boom via the Trolley drive or Gantry and Boom via the Gantry drive.

Prerequisites for all function modules:

- Power: INACTIVE
- boStandStill: TRUE
- boErrorFunctionBlock: FALSE

Note

General settings for the configuration data of the axes:

- Restart.behaviourInvalidSysvarAccess = 2
- Restart.restartActivationSetting = 0
- RestartCondition.restartAxisCondition = 0
- TypeOfAxis.DecodingConfig.transferSuperimposedPosition = 2
- TypeOfAxis.NumberofDataSets.changeMode = 7
- TypeOfAxis.DriveControlConfig.pulsesEnabled.bitNumber = 11
- TypeOfAxis.DriveControlConfig.pulsesEnabled.pzdNumber = 14
- TypeOfAxis.DriveControlConfig.releaseDisableMode = 7

There are two options for TO switchover:

- Internal TO data set switchover (1 TO, 1 DO)
- External TO switchover (up to 3 TOs, 1 DO)

Note

The "boChangeTechnologyObjectActive" feedback signal in AppZSW2 bit 15 = TRUE is made for an internal switchover only when the switchover itself is active.

- Example: Switchover between TO data set 1 (Trolley) and TO data set 3 (Boom)

  If "boChangeTechnologyObject" = FALSE, TO data set 1 becomes active and no feedback occurs ("boChangeTechnologyObjectActive" = FALSE). If "boChangeTechnologyObject" = TRUE, TO data set 3 becomes active and a feedback occurs ("boChangeTechnologyObjectActive" = TRUE).

For external switchover, the feedback "boChangeTechnologyObjectActive" in AppZSW2 bit 15 = TRUE occurs once the TO becomes activated.

- Example:

  TO Trolley "boChangeTechnologyObject" = FALSE → "boChangeTechnologyObjectActive" = FALSE
  "boChangeTechnologyObject" = TRUE → "boChangeTechnologyObjectActive" = TRUE

Internal TO switchover

With internal TO data set switchover the technology object does not change. An additional TO data set is simply set up containing the configuration data for the second drive. However, this data set is created as the third data set because the second data set is reserved for external encoders.

A second DO data set is also set up for the second motor in the drive object (DO). The internal TO data set switchover functions only between two drives:

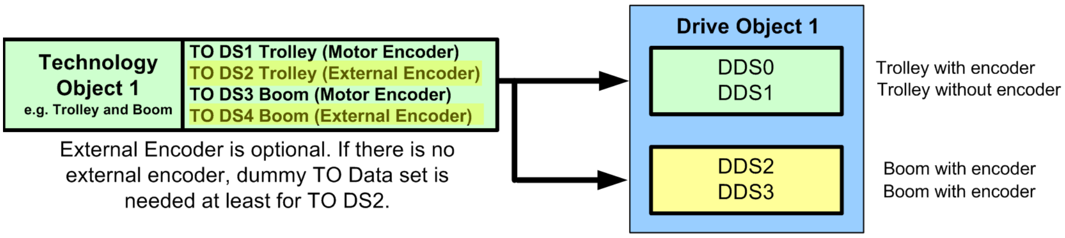

Technology Object 1
e.g. Trolley and Boom
TO DS1 Trolley (Motor Encoder)
TO DS2 Trolley (External Encoder)
TO DS3 Boom (Motor Encoder)
TO DS4 Boom (External Encoder)
External Encoder is optional. If there is no external encoder, dummy TO Data set is needed at least for TO DS2.
Drive Object 1
DDS0
DDS1
DDS2
DDS3
Trolley with encoder
Trolley without encoder
Boom with encoder
Boom with encoder

Figure 5-30 Internal TO switchover

Please note the following:

- Which motor is being driven via which converter must be determined. Example for Trolley and Boom: The Boom motor can be driven via the Trolley converter or the Trolley motor via the Boom converter.
- The function of each drive must be indicated in the program.

  Example: In the Boom drive, boBoomActive must be set to True. In the Trolley drive, boTrolleyActive must be set to True.
- The "boInternalTOChange" input for the OperationMode block must be set to TRUE and the "boExternalTOChange" input must not be selected.

Note

The external encoder is optional. If no external encoder is present: Create a dummy TO data set as TO data set 2, otherwise an internal switchover cannot occur.

TO data set 4 is required only when an external encoder is actually present.

Example:

Boom is being driven via the Trolley converter. Under "Axes" in the SCOUT project tree there is just one (1) axis or just one (1) "Trolley" TO. This axis is used to move Trolley and Boom.

In the technology object, at least three data sets must be created in the "Trolley" axis:

- TO data set 1: Trolley drive with motor encoder
- TO data set 2: Trolley drive with external encoder (if none is present, set up the data set all the same and do not use it any further)
- TO data set 3: Boom drive with motor encoder

Three drive data sets must also be created in the drive object for the Trolley drive.

- DO data set 0: Trolley drive with motor encoder
- DO data set 1: Trolley drive without encoder
- DO data set 2: Boom drive with encoder

Note

With internal TO switchover, the options for adapting the configuration data to the second motor are limited. Only the configuration data under TypeOfAxis.NumberOfDataSets and TypeOfAxis.NumberOfEncoders can be adapted.

Other important variables, such as the maximum velocity or the leadscrew pitch, are fixed (permanently stored) and, as such, are independent of individual data sets. (TypeOfAxis.NumberOfEncoders: An encoder can be permanently allocated to a data set.)

Procedure:

Note

1. Operating modes must be deselected and may only be reactivated once the switchover has been carried out.
2. There must be no errors present.
3. Drives must be at a standstill and switched-off.
4. Interconnections such as synchronous mode, master-slave operation or external encoder selection have to be disconnected.

The control and feedback signals are shown in the sequence diagram below. Selection is described in case ① and deselection in case ②.

- **Case 1: Changeover from Trolley to Boom**

1. For trolley: boInternalTOChange = TRUE
2. Deselect operating mode in Trolley: AppSTW1 bit 10 - 15 = FALSE (depending on the selected operating mode)
   Check whether the operating mode is deselected: AppZSW1 bit 10 – 15 = FALSE
3. Deselect pulse enable: STW1 bit 3 = FALSE
4. Select "boChangeTechnologyObject" in Trolley: AppSTW2 Bit 15 = TRUE
   Check whether "boChangeTechnologyObjectActive" is selected: AppZSW2 bit 15 = TRUE
5. Select operating mode in Trolley: AppSTW1 bit 10 - 15 = TRUE (depending on the selected operating mode)
   Check whether the operating mode is selected: AppZSW1 bit 10 - 15
6. Select pulse enable: STW1 bit 3 = TRUE

The internal TO switchover from Trolley to Boom is now complete.

- **Case 2: Changeover back from Boom to Trolley**

1. Deselect operating mode in Trolley: AppSTW1 bit 10 - 15 = FALSE (depending on the selected operating mode)
   Check whether the operating mode is deselected: AppZSW1 bit 10 - 15 = FALSE
2. Deselect pulse enable: STW1 bit 3 = FALSE
3. Deselect "boChangeTechnologyObject" in Trolley: AppSTW2 Bit 15 = FALLS
   Check whether "boChangeTechnologyObjectActive" is deselected: AppZSW2 bit 15 = FALSE
4. Select operating mode in Trolley: AppSTW1 bit 10 - 15 = TRUE (depending on the selected operating mode)
   Check whether the operating mode is selected: AppZSW1 bit 10 - 15
5. Select pulse enable: STW1 bit 3 = TRUE

The internal TO switchover from Boom to Trolley is now complete.

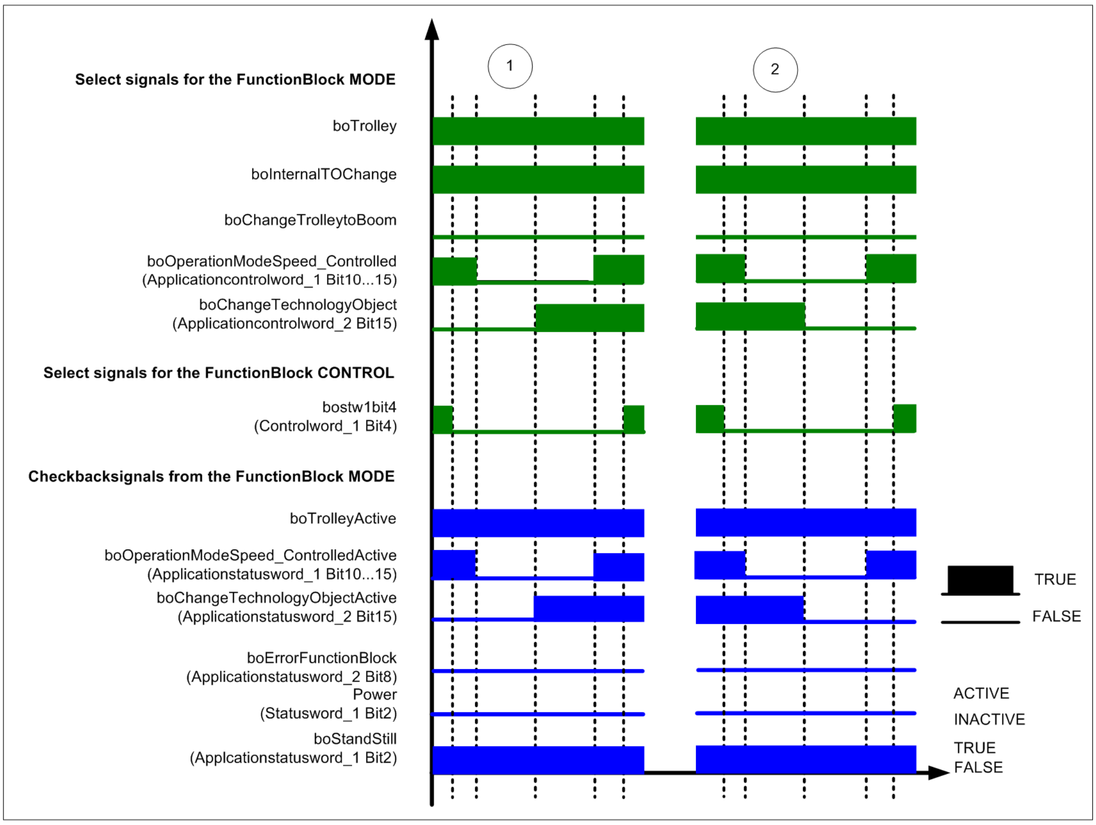


Figure 5-31 Internal TO switchover (case ①: selection; case ②: deselection)

External TO switchover (default solution)

With external TO switchover, a changeover is performed between two technology objects. There is no need for an additional data set in the technology object. The drive object still contains just one (1) drive with several data sets for both technology objects.

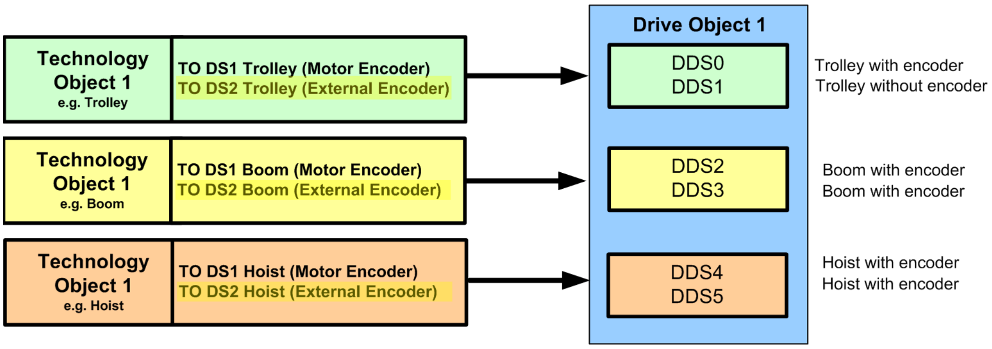


Figure 5-32 External TO switchover

The following points must be taken into account for external TO switchover:

- Which motor is being driven via which converter must be determined. The boom motor via the trolley or gantry converter or the gantry motor via the hoist converter.
- The FB_OperationMode must specify for each drive the associated function and its TO.

  Example:

  In the drive, i.e. DO data set, the following order must be maintained:

  – DO 0/1 = Hoist

  – DO 2/3 = Trolley

  – DO 4/5 = Boom

  This means the following inputs must be set in the associated function modules in FB_OperationMode:

  – Hoist boChangeTO_NR_1 = TRUE

  – Trolley boChangeTO_NR_2 = TRUE

  – Boom boChangeTO_NR_3 = TRUE

  In addition, the boExternalTOChange = TRUE and boInternalTOChange = FALSE inputs must be set for all technology objects involved in the external switchover.

  Example:

  For the Gantry motor, boGantry must be set to TRUE; for the Trolley motor, boTrolley must be set to TRUE.

Note

The setting for which technology object is active following power-up can be made in the configuration data for the technology object, e.g. D435.Trolley_1.TypeOfAxis.SetpointDriverInfo.interfaceAllocation

This setting can be made for each encoder under D435.Trolley_1.TypeOfAxis.NumberOfEncoders.Encoder_XY.interfaceAllocation. There are three options here:

- EXCLUSIVE (939):
  If no switchover is performed between multiple technology objects, EXCLUSIVE must be set.
- NON_EXCLUSIVE_AND_STARTUP_DEACTIVATED (940):
  This setting is selected for the appropriate technology objects that should be switched (usually Boom, for example).

Bit 15 must be set in application control word 2 so that the drive evaluates status word 1, bit 9. Otherwise, it does not evaluate status word 1, bit 9.

The application status word 2, bit 15 is set when application control word 2, bit ? and status word 1, bit 9 are set.

Example:

The Boom is being driven via the Trolley converter. Under "Axes" in the SCOUT project tree there are two axes, "Trolley" and "Boom". This means that each axis represents a motor. Nevertheless, there is only one drive object (DO), i.e. only one converter for both motors, which are operated alternately.

Only one data set is needed in the "Trolley" technology object (without external encoder).

- Data set 1: Trolley drive with motor encoder
- Data set 2: Trolley drive with external encoder (only if required)

Only one data set is needed in the "Boom" technology object (without external encoder).

- Data set 1: Boom drive with motor encoder
- Data set 2: Boom drive with external encoder (only if required)

3 or 4 drive data sets are needed in the "Trolley" drive object.

- DO data set 0: Trolley drive with motor encoder
- DO data set 1: Trolley drive without encoder
- DO data set 2: Boom drive with motor encoder
- DO data set 3: Drive without encoder (exception: Drives with active load. They are always operated with encoder.)

Note

With external TO switchover, every motor has its own TO axis. The configuration data can be individually adapted in the technology objects.

Procedure:

Note

1. There must be no errors present.
2. Drives have to be switched off.
3. Interconnections such as synchronous mode, master-slave operation or external encoder selection have to be disconnected.

Note

The control of the external Trolley to Boom switchover and Boom to Trolley switchover with the signal diagram for the S7 project are described in the Commissioning (Page 351). Further, the changeover sequence is available in an S7 project on the CD that is supplied when the system is shipped.

The control and feedback signals are shown in the sequence diagram below. Selection is described in case ① and deselection in case ②.

- Case 1: Changeover from Trolley to Boom

1. Deselect pulse enable: STW1 bit 3 = FALSE
2. Deselect "boChangeTechnologyObject" in Trolley: AppSTW2 bit 15 = FALSE
   Check trolley drive: ZSW1 bit 9 = FALSE
3. Select new TO Boom "boChangeTechnologyObject": AppStw2 bit 15 = TRUE
   Check Boom drive: ZSW1 bit 9 = TRUE

The external TO switchover from Trolley to Boom is now complete.

- Case 2: Changeover back from Boom to Trolley

1. Deselect pulse enable: STW1 bit 3 = FALSE
2. Deselect "boChangeTechnologyObject" in Boom: AppSTW2 bit 15 = FALSE
   Check Boom drive: ZSW1 bit 9 = FALSE
3. Select new TO Trolley "boChangeTechnologyObject": AppStw2 bit 15 = TRUE
   Check trolley drive: ZSW1 bit 9 = TRUE

The external TO switchover from Boom to Trolley is now complete.

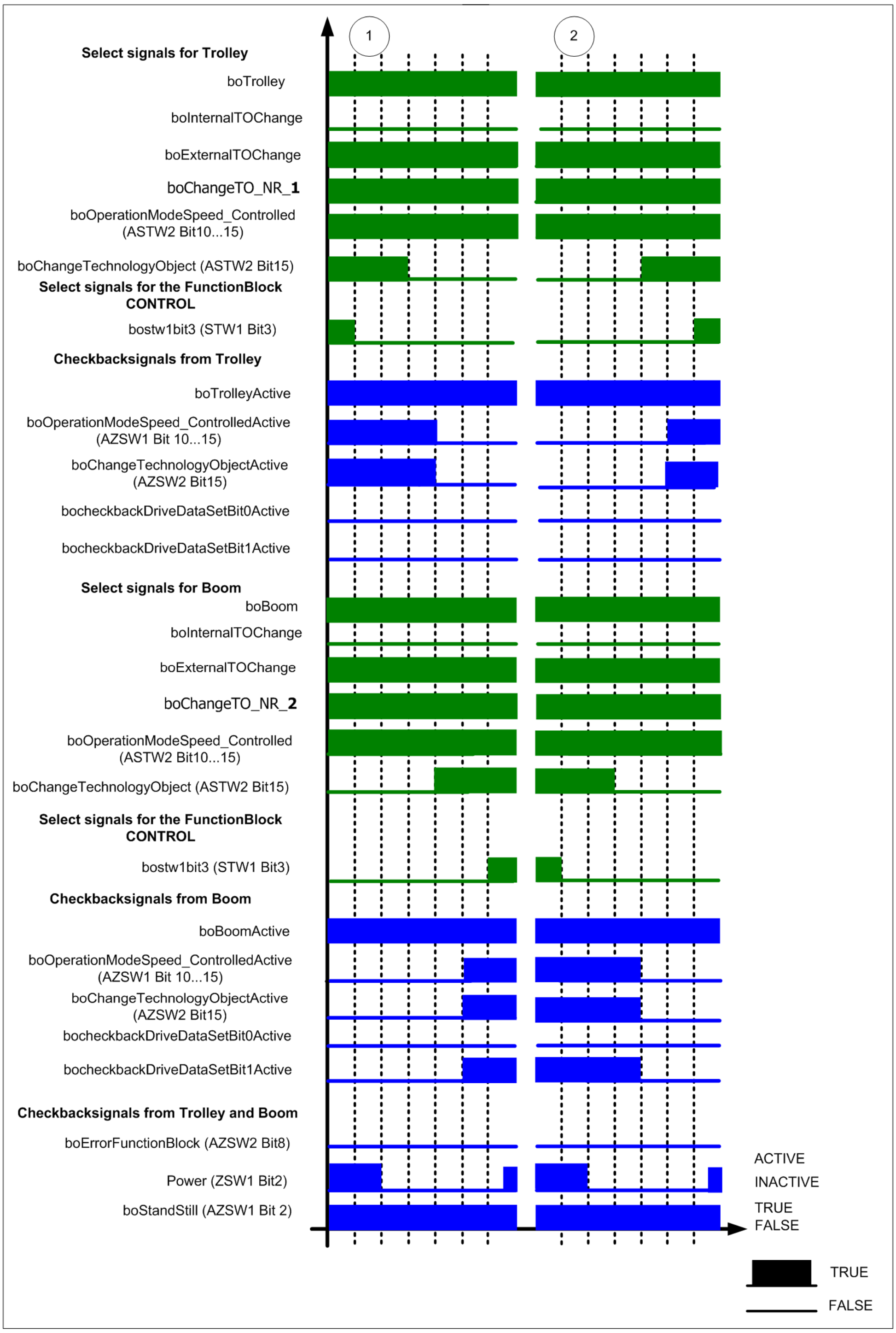


Figure 5-33 External TO switchover (case ①: selection; case ②: deselection)

For a switchover between Boom and another drive (Gantry or Trolley), external TO switchover with Trolley is usually performed. For this reason, both function modules are contained in the project in full.

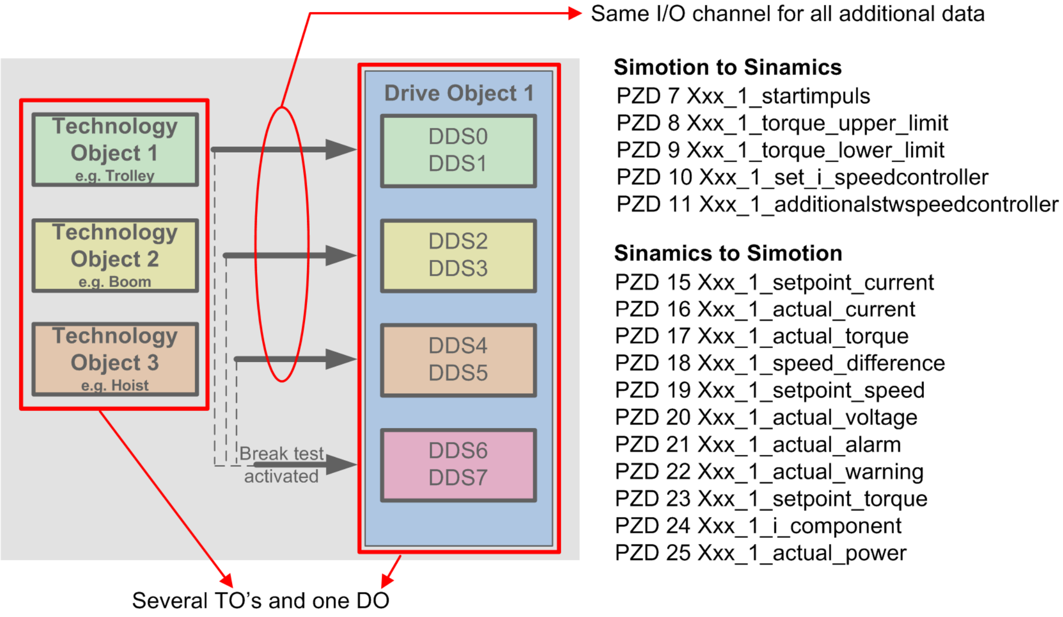
If a different switchover is required, for example, between Gantry and Hoist, or if the switchover should be extended to three, for example for Trolley and Boom to Trolley and Boom and Hoist, the following must be taken into account:

All switchable technology objects ultimately operate on a drive object (DO) in the SINAMICS. The data is transferred between a technology object and a drive object using a standard telegram. Depending on which the drive object is active, the connection is established automatically using the telegram. The user does not need to make any additions. Additional information, however, is exchanged between technology object and drive object not using a standard telegram but rather using the I/O interface or the address list of the SIMOTION. This must be be adapted in the case a reconfiguration of the external switchover.

All technology objects that can be switched between each other access the I/O variables of the master TO axis. If, therefore, the master TO axis is a Trolley, the other TOs involved in the switchover access those I/O variables whose name other with "Trolley".

Example:

Switchover between Trolley, Boom and Hoist: → TO Trolley, TO Boom and TO Hoist work on DO trolley (converter). The following figure summarizes the relationship. In our example, Xxx would be Trolley.

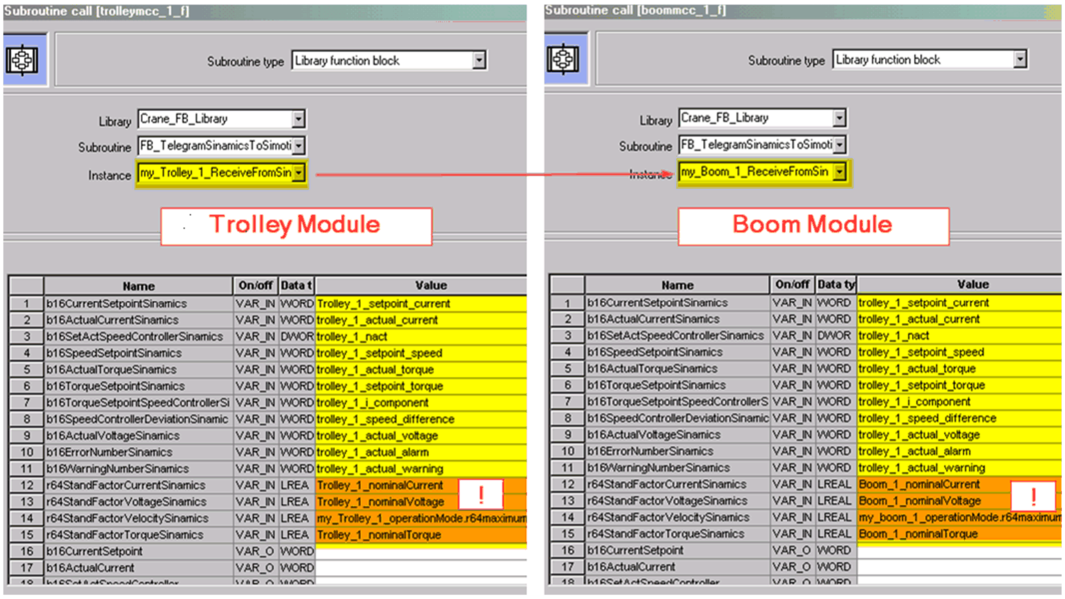


Simotion to Sinamics
PZD 7 Xxx_1_startimpuls
PZD 8 Xxx_1_torque_upper_limit
PZD 9 Xxx_1_torque_lower_limit
PZD 10 Xxx_1_set_i_speedcontroller
PZD 11 Xxx_1_additionalstwspeedcontroller

Sinamics to Simotion
PZD 15 Xxx_1_setpoint_current
PZD 16 Xxx_1_actual_current
PZD 17 Xxx_1_actual_torque
PZD 18 Xxx_1_speed_difference
PZD 19 Xxx_1_setpoint_speed
PZD 20 Xxx_1_actual_voltage
PZD 21 Xxx_1_actual_alarm
PZD 22 Xxx_1_actual_warning
PZD 23 Xxx_1_setpoint_torque
PZD 24 Xxx_1_i_component
PZD 25 Xxx_1_actual_power

Figure 5-34 Several technology objects on a drive object

If a switchover is to be made to TO Boom, TO Boom must access the same variables in the I/O as the TO Trolley in order to also receive the additional information. If the switchover is extended to the Hoist, TO Hoist must also access the same I/O variables as TO Trolley and TO Boom. The following figure shows that the variable assignments for the Boom (right-hand side) access the variables for the Trolley (left-hand side) . Observe the possible differences between Trolley and Boom for the nominal values! The variable assignment for Hoist then corresponds exactly to that for Boom.



Subroutine call [trolleymcc_1_f]

Subroutine type: Library function block

Library: Crane_FB_Library

Subroutine: FB_TelegramSinamicsToSimoti

Instance: my_Trolley_1_ReceiveFromSin

	Name	On/off	Data t	Value
1	b16CurrentSetpointSinamics	VAR_IN	WORD	Trolley_1_setpoint_current
2	b16ActualCurrentSinamics	VAR_IN	WORD	trolley_1_actual_current
3	b16SetActSpeedControllerSinamics	VAR_IN	DWOR	trolley_1_nact
4	b16SpeedSetpointSinamics	VAR_IN	WORD	trolley_1_setpoint_speed
5	b16ActualTorqueSinamics	VAR_IN	WORD	trolley_1_actual_torque
6	b16TorqueSetpointSinamics	VAR_IN	WORD	trolley_1_setpoint_torque
7	b16TorqueSetpointSpeedControllerSi	VAR_IN	WORD	trolley_1_i_component
8	b16SpeedControllerDeviationSinamic	VAR_IN	WORD	trolley_1_speed_difference
9	b16ActualVoltageSinamics	VAR_IN	WORD	trolley_1_actual_voltage
10	b16ErrorNumberSinamics	VAR_IN	WORD	trolley_1_actual_alarm
11	b16WarningNumberSinamics	VAR_IN	WORD	trolley_1_actual_warning
12	r64StandFactorCurrentSinamics	VAR_IN	LREA	Trolley_1_nominalCurrent
13	r64StandFactorVoltageSinamics	VAR_IN	LREA	Trolley_1_nominalVoltage
14	r64StandFactorVelocitySinamics	VAR_IN	LREA	my_Trolley_1_operationMode.r64maximum
15	r64StandFactorTorqueSinamics	VAR_IN	LREA	Trolley_1_nominalTorque
16	b16CurrentSetpoint	VAR_O	WORD	
17	b16ActualCurrent	VAR_O	WORD	

Subroutine call [boommcc_1_f]

Subroutine type: Library function block

Library: Crane_FB_Library

Subroutine: FB_TelegramSinamicsToSimoti

Instance: my_Boom_1_ReceiveFromSin

	Name	On/off	Data ty	Value
1	b16CurrentSetpointSinamics	VAR_IN	WORD	trolley_1_setpoint_current
2	b16ActualCurrentSinamics	VAR_IN	WORD	trolley_1_actual_current
3	b16SetActSpeedControllerSinamics	VAR_IN	DWOR	trolley_1_nact
4	b16SpeedSetpointSinamics	VAR_IN	WORD	trolley_1_setpoint_speed
5	b16ActualTorqueSinamics	VAR_IN	WORD	trolley_1_actual_torque
6	b16TorqueSetpointSinamics	VAR_IN	WORD	trolley_1_setpoint_torque
7	b16TorqueSetpointSpeedControllerS	VAR_IN	WORD	trolley_1_i_component
8	b16SpeedControllerDeviationSinamic	VAR_IN	WORD	trolley_1_speed_difference
9	b16ActualVoltageSinamics	VAR_IN	WORD	trolley_1_actual_voltage
10	b16ErrorNumberSinamics	VAR_IN	WORD	trolley_1_actual_alarm
11	b16WarningNumberSinamics	VAR_IN	WORD	trolley_1_actual_warning
12	r64StandFactorCurrentSinamics	VAR_IN	LREAL	Boom_1_nominalCurrent
13	r64StandFactorVoltageSinamics	VAR_IN	LREAL	Boom_1_nominalVoltage
14	r64StandFactorVelocitySinamics	VAR_IN	LREAL	my_boom_1_operationMode.r64maximum
15	r64StandFactorTorqueSinamics	VAR_IN	LREAL	Boom_1_nominalTorque
16	b16CurrentSetpoint	VAR_O	WORD	
17	b16ActualCurrent	VAR_O	WORD	

Figure 5-35 Technology object (TO) - drive object (DO) interconnections

5.3.9.4 Description of the operating modes

The following information and rules should be carefully observed:

Information and rules when using a master switch:

- If the master switch is deflected in the positive direction, then in the AUTOMATIC, MANUAL and EASY_POSITIONING operating modes
  - a target position larger than the actual position must be entered (target position > actual position), and
  - a positive velocity "r64Velocity" must be entered through the controller (S7).
- If the master switch is deflected in the negative direction, then in the AUTOMATIC, MANUAL and EASY_POSITIONING operating modes
  - a target position smaller than the actual position must be entered (target position < actual position), and
  - a negative velocity "r64Velocity" must be entered through the controller (S7).

Information on the minimum velocity:

- The minimum velocity in the positive direction for the AUTOMATIC, MANUAL and EASY_POSITIONING operating modes must also be specified as positive values (r64MinimumVelocityAUTOPositive or r64MinimumVelocityManualPositive or r64MinimumVelocityEASY_POSITIONINGPositive).
- The minimum velocity in the negative direction for the AUTOMATIC, MANUAL and EASY_POSITIONING operating modes must also be specified as negative values r64MinimumVelocityAUTONegative or r64MinimumVelocityManualNegative or r64MinimumVelocityEASY_POSITIONINGPositive).

Change of the velocity setpoint, acceleration time, delay time, initial rounding-off and final rounding-off:

- Velocity, acceleration and deceleration can be changed at any time in the SPEED_CONTROLLED operating mode.
- In the AUTOMATIC, EASY_POSITIONING and MANUAL operating modes, velocity, acceleration and deceleration can be changed at any time:

  In the delay phase, a longer delay time is not accepted although a shorter delay time is accepted. If the delay time is reduced, the target position is **no** longer approached, but rather the drive is braked with the shorter delay time.
- In the SENSORLESS EMERGENCY operating mode, the velocity can only be changed up to the parameterizable limit "r64VelocitySensorlessEmergency". Acceleration and deceleration can be changed at any time.
- In all operating modes, a change of the initial rounding-off and the final rounding-off is applied only with a deflected master switch. The exception is the AUTOMATIC operating mode. In this case, the change is applied immediately.

Operating mode: AUTOMATIC control type: Closed-loop position controlled with/without synchronous operation

The AUTOMATIC operating mode is only accepted if the prerequisites for "selecting operating modes" are fulfilled.

In the AUTOMATIC operating mode, the drive moves position-controlled to the specified position (with the specified values for velocity, ramp-up time and ramp-down time). The request is started with the specified values of the S7 using the control bit "start automatic operation". If the master switch is deflected during automatic operation, the request is interrupted and the master switch specifies the velocity setpoint.

If the master switch is deflected in the positive direction, then a positive velocity (r64Velocity) and a target position (r64TargetPosition) that is greater than the actual position, must be entered.

If the master switch is deflected in the negative direction, then a negative velocity (r64Velocity) and a target position (r64TargetPosition) that is less than the actual position, must be entered.

The target position is precisely approached if a minimum velocity is not configured. A configured minimum velocity not equal to zero for a positive or negative direction (e.g. r64MinimumVelocityAutoPositive) and if the master switch is not deflected, then when the specified target position is reached, then the drive continues to move with the minimum velocity (crawl) until the S7 specifically stops axis motion).

AUTOMATIC request

In the AUTOMATIC operating mode, the controller (S7) specifies the target position, velocity, acceleration and deceleration. Then, by starting the automatic request, the target position is executed with the specified, dynamic values.

The various sequences of an automatic request are shown in the following sequence diagram. The following should be observed regarding this sequence diagram:

- The operating mode was already correctly selected, i.e. the plausibility check is OK.
- Velocity, acceleration and deceleration are not changed in this sequence diagram.
- The master switch is not deflected.

Note

AUTOMATIC requests can only be executed if the module is in MasterMode (boSlaveModeActive = FALSE).

Section 1:
In the first section of the sequence diagram, a target position of 20,000 mm is specified. Then, the AUTOMATIC request is started with a positive edge of the input signal "boStartAutomaticOrder". The feedback signal "boAutomaticOrderActive" is set as soon as the request is executed. When the specified target position is reached, the feedback signal "boAutomaticOrderActive" is reset and the feedback signal "boAutomaticPositionReached" is set.

Section 2:
In the second section of the sequence diagram, a target position of 40,000 mm is specified. As soon as the request is executed, the checkback signal "boAutomaticPositionReached" is set. The same sequence as in section 1 is executed.

Section 3:
In the third section of the sequence diagram, a target position of 60,000 mm is specified. The same sequence as in section 2 is executed. During an automatic request that is running, the target position is changed from 60,000 mm to 0 mm. The new target position is accepted immediately, i.e. the old target position is rejected and the new target position is approached. When the specified target position is reached, the feedback signal "boAutomaticOrderActive" is reset and the feedback signal "boAutomaticPositionReached" is set.

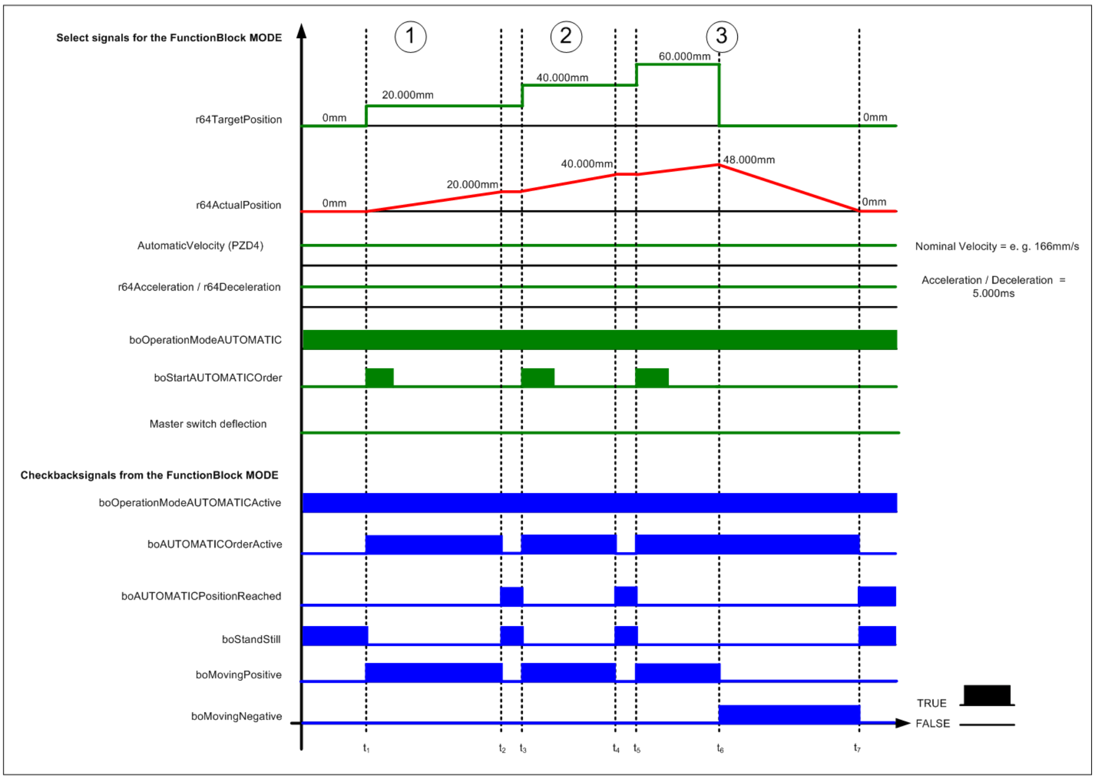


Figure 5-36 FB_OperationMode, AUTOMATIC request

Minimum velocity for the AUTOMATIC operating mode

The minimum velocity should be separately configured for the positive and negative directions. If a minimum velocity is configured, the specified target position is not precisely approached, but the axis moves past the target position with the configured minimum velocity. When approaching the target position, the drive transitions from constant velocity into the deceleration phase. The override point of the AUTOMATIC request is a percentage value above the configured minimum velocity and an absolute value before the target position. The minimum velocity can be set using the r64toleranceMinimumVelocity input between 0-500% based on the minimum velocity; the target position is set using the r64MinimumVelocityTolerancePosition input. This means that a continuous velocity transition is achieved. The drive must be stopped from the higher-level controller (S7).

Use: Minimum velocity

As soon as the target positions are not precisely known - e.g. due to different heights of the containers or trucks, the minimum velocity should be used.

Sequence diagram

- Various sequences of an AUTOMATIC request - with and without configured minimum velocity - are shown in the following sequence diagram. The same secondary conditions and constraints apply as in the diagram above.

Section 1:
In the first section of the sequence diagram, a positive minimum velocity is not configured. A target position of 10,000 mm is specified. Then, the AUTOMATIC request is started with a positive edge of the input signal "boStartAutomaticOrder". The feedback signal "boAutomaticOrderActive" is set as soon as the request is executed. When the specified target position is reached, the feedback signal "boAutomaticOrderActive" is reset and the feedback signal "boAutomaticPositionReached" is set.

Section 2:
In the second section of the sequence diagram, a positive minimum velocity is configured. A target position of 20,000 mm is specified. The same sequence as in section 1 is started. The AUTOMATIC request is cut-out in the deceleration phase as soon as the actual velocity reaches the configured minimum velocity. The "boAutomaticOrderActive" feedback signal is then reset. The drive continues with the configured minimum velocity until the power-down command comes from the controller.

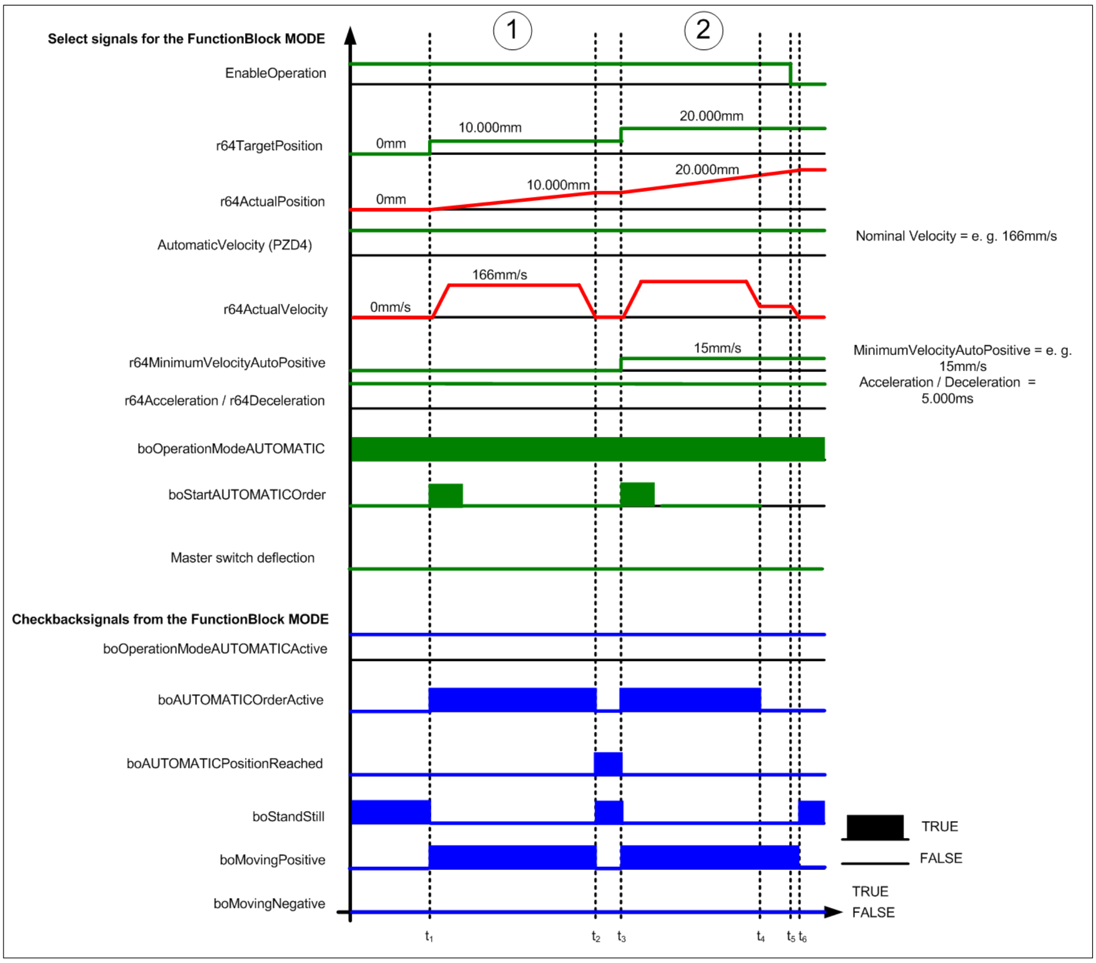


Figure 5-37 FB_OperationMode, AUTOMATIC operating mode, minimum velocity

Deflecting the master switch in the AUTOMATIC operating mode

An AUTOMATIC request is started or executed if the master switch is not deflected. The crane driver can however, intervene in an AUTOMATIC request that is running at any time by deflecting the master switch. When the master switch is deflected, then the active AUTOMATIC request is interrupted and the crane driver can control the crane. The intervention of the crane driver is only intended for emergencies and does **not** change the operating mode. After intervening, the crane driver can restart the AUTOMATIC request. It should be noted that the velocity for AUTOMATIC is sent via PZD4 and the velocity of the deflection is sent via PZD2 - as is usually the case in the MANUAL and SPEED_CONTROLLED operating modes.

Note

As soon as the master switch is deflected in the AUTOMATIC operating mode, the control (S7) must issue the following:

positive velocity setpoint in PZD2 and target position > actual position, or
negative velocity setpoint in PZD2 and target position < actual position

Sequence diagram

Various sequences of an AUTOMATIC request without deflected master switch and with deflected master are shown in the following sequence diagram. It should be noted that acceleration and deceleration are not changed in this sequence diagram.

Section 1:
In the first section of the sequence diagram, the master switch is not configured. A target position of 10,000 mm is specified. The AUTOMATIC request then runs as shown in the diagram "FB_OperationMode, AUTOMATIC request" (refer above).

Section 2:
In the second section of the sequence diagram, the master switch is deflected during an AUTOMATIC request. A target position of 20,000 mm is specified. The AUTOMATIC request is then started with a positive edge of the input signal "boStartAutomaticOrder" and the feedback signal "boAutomaticOrderActive" is set. While the automatic request is being processed, the master switch is deflected in the positive direction. The AUTOMATIC request is interrupted and the control (S7) specifies a new maximum, positive target position. Using his master switch, the crane driver has control over his crane.

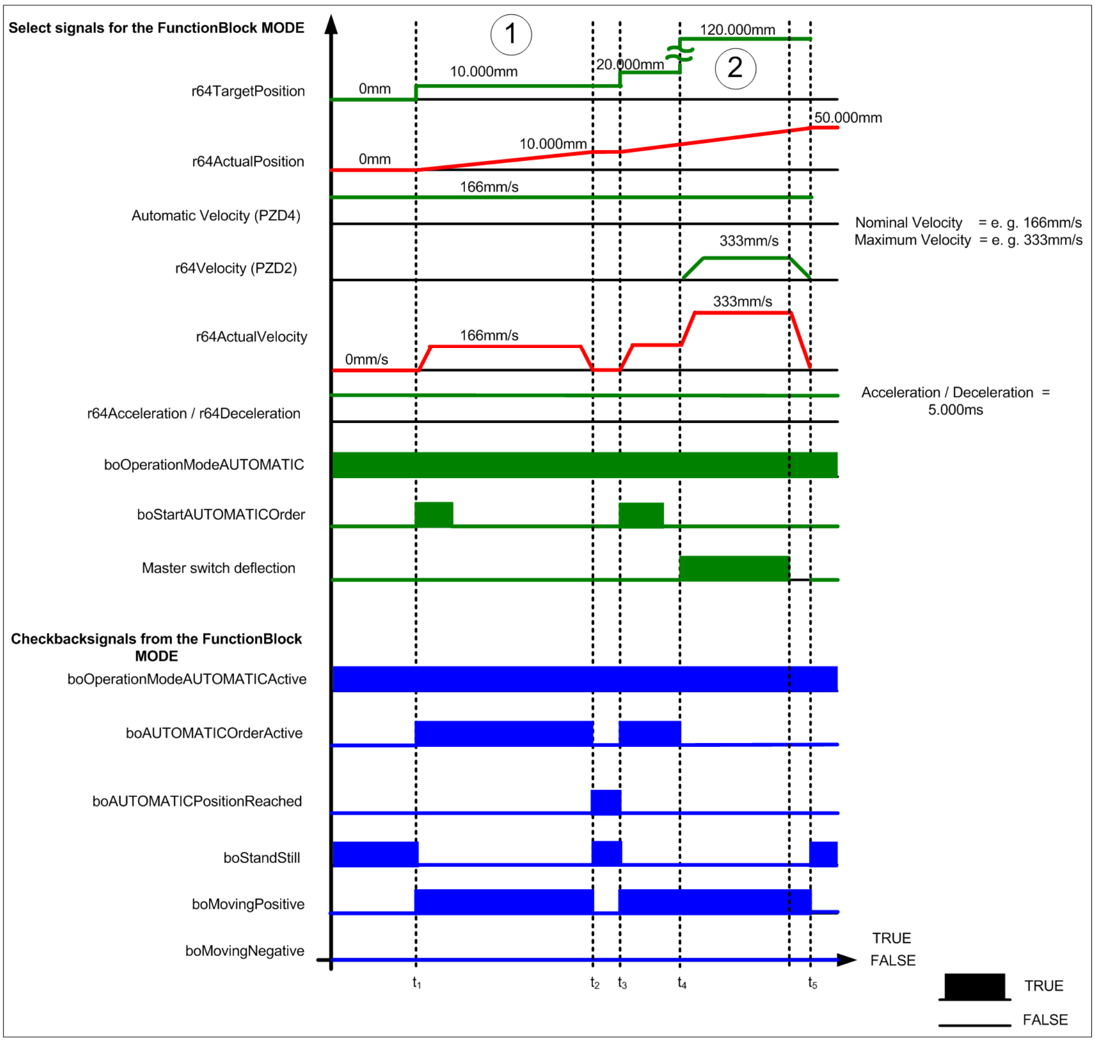


Figure 5-38 FB_OperationMode, AUTOMATIC operating mode, interrupt by the master switch

Change in the AUTOMATIC operating mode

A target position change in the AUTOMATIC operating mode is only accepted if the target position has still not been reached. Velocity and acceleration can be changed at any time. Deceleration can only be changed and accepted in the following cases:

1. The drive is stationary.
2. The drive is in the acceleration phase.
3. The drive is in the constant velocity phase.
4. The drive is in the constant velocity phase and the remaining distance (r64distanceToGo) is greater than the braking distance (r64StopingDistance).

As soon as the remaining distance (r64distanceToGo) is less than the braking distance (r64Stoping Distance) and the drive is in the deceleration phase, it is no longer possible to change the deceleration time. The drive is braked with the last valid deceleration time. This safety function has been incorporated so that if the deceleration ramp is unfavorably/incorrectly set, then the drive does not move past the target.

Sequence diagram

Various sequences of an AUTOMATIC request where the deceleration time is changed are shown in the following sequence diagram. In the sequence diagram, a differentiation is made between the specified deceleration (r64Deceleration) from the control (S7) and the accepted deceleration (r64Deceleration_internal). It should be noted that acceleration and velocity are not changed in this sequence diagram.

Section 1:
In the first section of the sequence diagram, a target position of 20,000 mm is specified and a deceleration time of 9,000 ms. Then, a normal sequence of the AUTOMATIC request starts. During the constant-velocity travel, the delay time is changed from 9,000 ms to 7,000 ms. This change is also accepted as the drive is in the constant-velocity phase. When the specified target position is reached, the feedback signal "boAutomaticOrderActive" is reset and the feedback signal "boAutomaticPositionReached" is set.

Section 2:
In the second section of the sequence diagram, a target position of 40,000 mm and a deceleration time of 7,000 ms are specified. Then, an AUTOMATIC process starts. The distance to go (r64distanceToGo) is less than the braking distance (r64Stopping Distance) and the drive is decelerating. In this phase, the delay time is changed from 7,000 ms to 5,000 ms. This change is not accepted as the remaining distance is less than the braking distance and the drive is decelerating, i.e. the drive decelerates with the last valid deceleration time. When the specified target position is reached, the feedback signal "boAutomaticOrderActive" is reset and the "boAutomaticPositionReached" is set.

Section 3:
In the third section of the sequence diagram, a target position of 60,000 mm is specified and a deceleration time of 9,000 ms. The AUTOMATIC request is then started. While accelerating, the deceleration time is changed from 9,000 ms to 10,000 ms. This change is accepted because the drive is accelerating. When the specified target position is reached, the feedback signal "boAutomaticOrderActive" is reset and the feedback signal "boAutomaticPositionReached" is set.

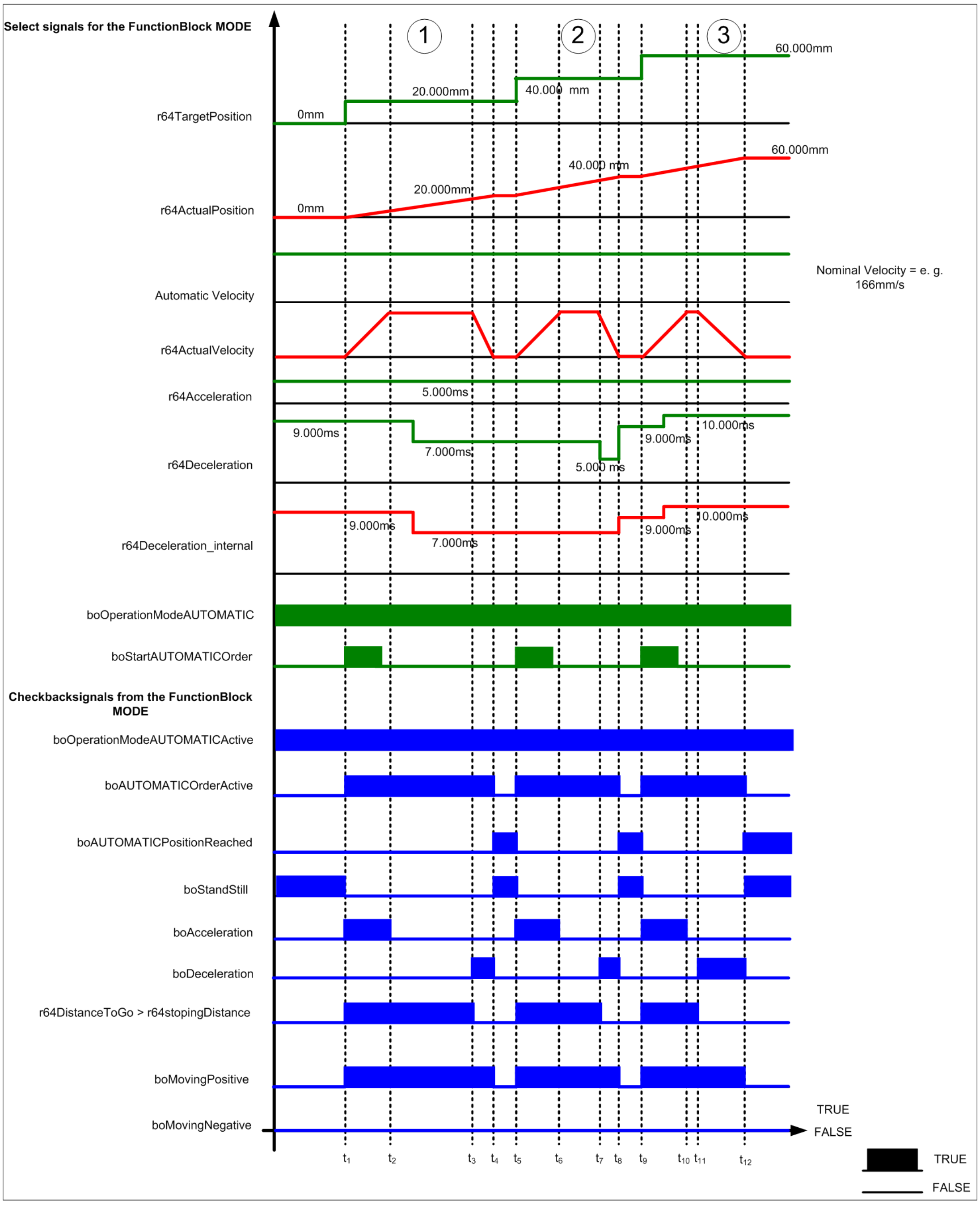


Figure 5-39 FB_OperationMode, deceleration ramp in the AUTOMATIC operating mode

Operating mode: MANUAL
control mode: Closed-loop position controlled with/without synchronous operation

The MANUAL operating mode is only accepted if the prerequisites for "select operating modes" are fulfilled.

In the MANUAL operating mode, the axis moves closed-loop position controlled. In so doing, the S7 specifies the target position, velocity, ramp-up time and ramp-down time. The travel direction is specified by the master switch via the velocity. If the master switch is deflected in the positive direction, then a positive velocity (r64Velocity) and a target position (r64TargetPosition) that is greater than the actual position, must be entered.

If the master switch is deflected in the negative direction, then a negative velocity (r64Velocity) and a target position (r64TargetPosition) that is less than the actual position, must be entered.

The target position is precisely approached if a minimum velocity is not configured. Once the preset ramp-down time has elapsed, a velocity limit is applied shortly before the target position (smart slow-down). A configured minimum velocity not equal to zero for a positive or negative direction (e.g. "r64MinimumVelocityManualPositive") and deflected master switch means that when the specified target position is reached, then the axis continues to move with the minimum velocity (crawl).

Sequence diagram

In the following sequence diagram, various sequences associated with the MANUAL operating mode are shown without configured minimum velocity and with configured minimum velocity. It should be noted that acceleration and deceleration are not changed in this sequence diagram.

Section 1:
In the first section of the sequence diagram, a positive minimum velocity is not configured. The master switch is deflected in the positive direction. In so doing, the maximum positive target position (r64TargetPosition) of 120,000 mm is specified. The velocity (r64Velocity) is defined by the degree of deflection of the master switch. The velocity is slowly reduced down to standstill (zero speed) using the master switch.

Section 2:
In the second section of the sequence diagram, a positive minimum velocity is configured. The master switch is deflected in the positive direction. In so doing, the maximum positive target position (r64TargetPosition) of 120,000 mm is specified. When the maximum positive target position is reached and therefore the actual velocity reaches the configured minimum velocity, then in the deceleration phase a changeover is made to the minimum velocity. The drive continues to move with the configured minimum velocity until the master switch is no longer deflected or the power-down command comes from the control.

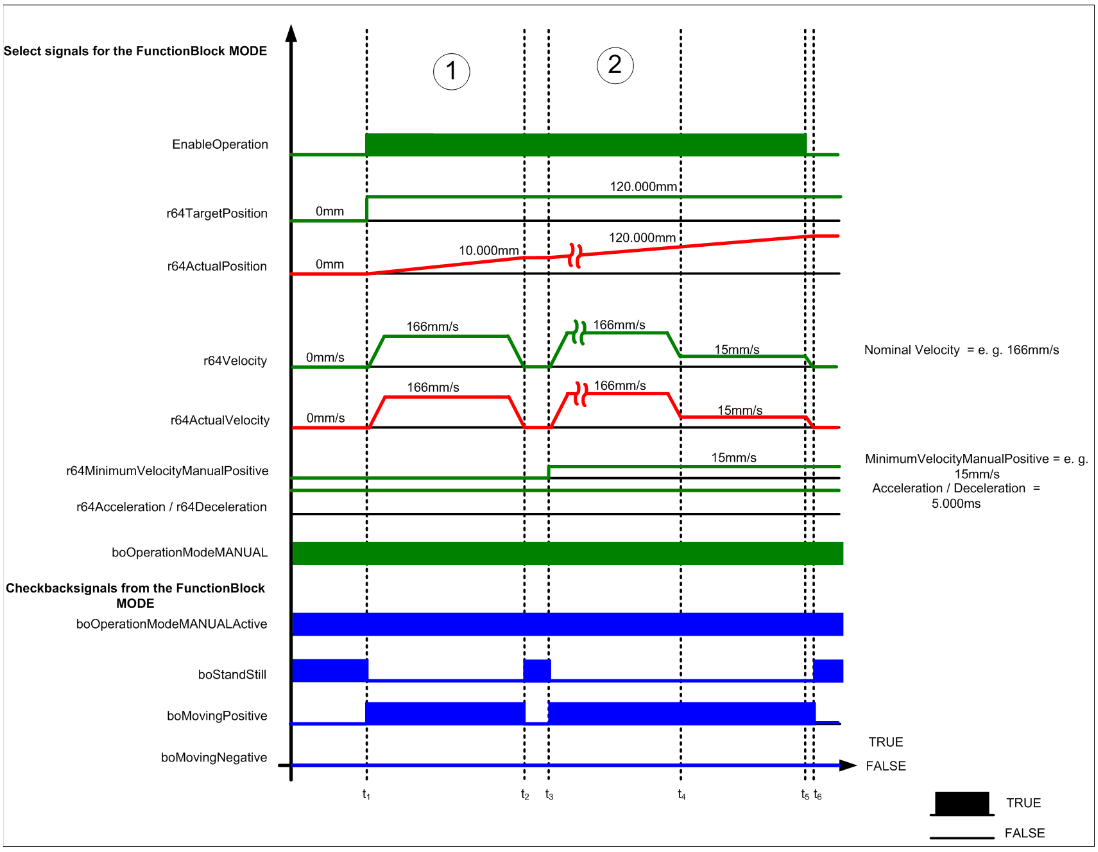


Figure 5-40 FB_OperationMode, MANUAL operating mode, minimum velocity

Operating mode: EASY_POSITIONING control type: Closed-loop position controlled with/without synchronous operation

The EASY_POSITIONING operating mode is accepted only when the prerequisites for "selecting operating modes" are fulfilled.

In the EASY_POSITIONING operating mode, controlled movement is made to the target position. In so doing, the S7 specifies the target position, velocity, ramp-up time and ramp-down time. The travel direction is specified by the master switch via the velocity. If the master switch is deflected in the positive direction, a positive velocity (r64Velocity) and a target position (r64TargetPosition) that is greater than the actual position must be entered.

If the master switch is deflected in the negative direction, a negative velocity (r64Velocity) and a target position (r64TargetPosition) that is less than the actual position must be entered.

The target position is precisely approached if a minimum velocity is not configured.

A velocity limit is applied shortly before the target position after the predefined ramp-down time. A configured minimum velocity not equal to zero for a positive or negative direction (e.g. r64MinimumVelocityEASY_POSITIONINGPositive) and deflected master switch means that when the specified target position is reached the axis continues to move with the minimum velocity (crawl).

For EASY_POSITIONING, the r64KP_ChangeOverPoint gain must be set for positioning. This sets the deceleration ramp and therefore the approach.

The gain has the default value of 0.5. This results in a slow approach (depending on the acceleration and the maximum velocity). Increasing the gain can achieve a steeper deceleration.

Note

During the first commissioning, a positioning must ensure that the gain is set correctly and no overshoot occurs. The positioning must be tested again after every change of the gain, deceleration or jerk.

Setting the positioning

The positioning is controlled using the "r64KP_ChangeOverPoint" variable. The positioning generally controls the maximum permissible velocity setpoint until the end position is approached (grab open or closed). The velocity setpoint can be limited at any time by deflecting the master switch. Shortly before the end position is reached, the "r64KP_ChangeOverPoint" variable accepts the velocity setpoint and approaches the end position.

"r64KP_ChangeOverPoint" is required for the P element. The root function keeps the setpoint lower than with a P element in the event of significant positional differences. When the end position is reached and the positional difference is, therefore, minor, the route function is replaced by the P element, since its characteristic gets very steep towards the end. Here, the P element becomes active and ensures a linear path.

The maximum gain near to the target point is set by the K_P factor.

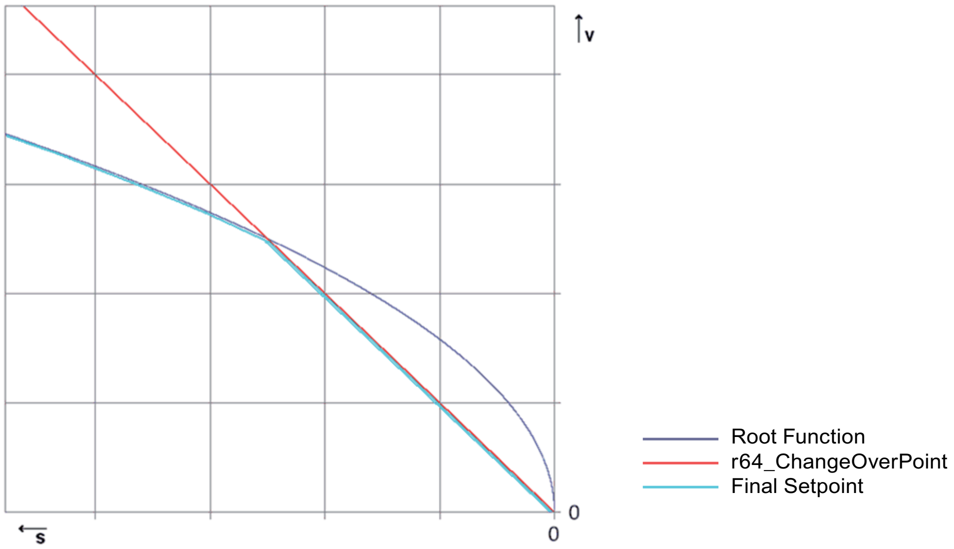


Figure 5-41 Theoretical characteristic of KP variables and results

Note

Experience has shown that the value 0.5 (default value for "r64KP_ChangeOverPoint") works very well in practice.

The gain factor defines the length of the constant velocity phase and the start of the deceleration phase. The specified ramp-up time is not changed, but only the ramp-down time. As a consequence, the total delay time can be influenced.

NOTICE
The slope of the deceleration ramp determined by the K_P variable must be set so that the predefined ramp-down time (set by the higher-level controller, e.g. S7) is not undershot. This is to stop the drive overshooting the target.

If "r64KP_ChangeOverPoint" is too small, the drive will go straight into the deceleration phase during ramp-up without having reached the constant velocity phase because the K_P factor is too weak and the drive must start to brake earlier.

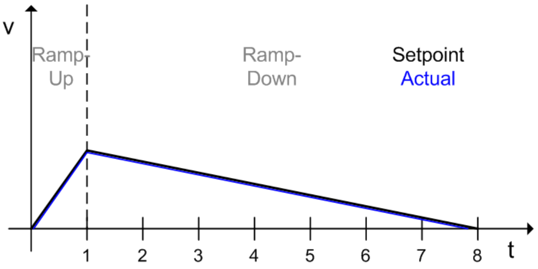


Figure 5-42 K_P factor too weak → long delay time

A delay time close to requirements with reserves can be achieved by increasing "r64KP_ChangeOverPoint". The ramp-down time is somewhat longer than the ramp-up time. The delay time could be reduced further. The purpose of a reserve is to allow the predefined ramp-down time (set by the higher-level controller) to be increased without having to adapt the K_P factor.

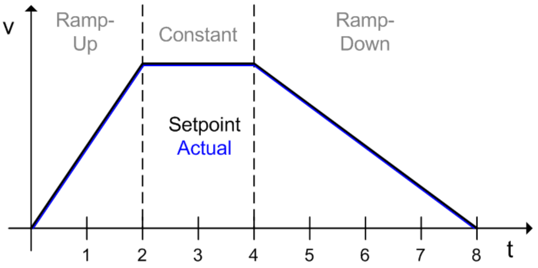


Figure 5-43 Increased K_P factor with reserves → good delay time

If the KP factor is optimized in terms of time, the deceleration ramp will match the predefined deceleration ramp. In this context, it is important that the predefined deceleration is not undershot. This is to stop the drive overshooting the target. The predefined ramp-down time cannot be increased further without adapting the KP factor accordingly.

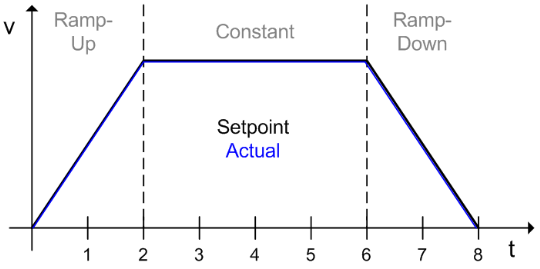


Figure 5-44 Time-optimized KP setting

Setting a KP factor which is too high shortens the ramp-down time and makes the ramp steeper than would be the case with the predefined ramp-down time. The drive will overshoot the target in this case.

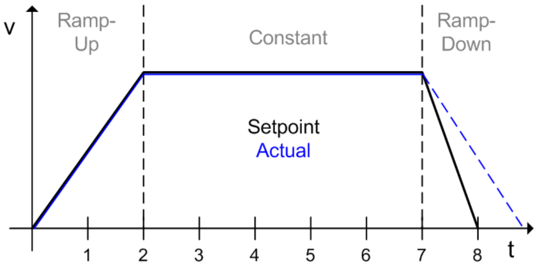


Figure 5-45 KP factor too high → target overshot

Note

Ensure when positioning that the target is not overshot, not even briefly. The positioning should be tested again after every change of the maximum velocity, gain, deceleration or jerk. The target position for testing should be set so that an overshoot will not cause any damage; otherwise the KP factor is too large.

Sequence diagram

In the following sequence diagram, various sequences associated with the EASY_POSITIONING operating mode are shown without configured minimum velocity and with configured minimum velocity. It should be noted that acceleration and deceleration are not changed in this sequence diagram.

Section 1:

No positive minimum velocity is configured in the first section of the sequence diagram. The master switch is deflected in the positive direction. In so doing, the maximum positive target position (r64TargetPosition) of 120000 mm is specified. The velocity (r64Velocity) is defined by the degree of deflection of the master switch. The velocity is slowly reduced down to standstill (zero speed) using the master switch.

Section 2:

A positive minimum velocity is configured in the second section of the sequence diagram. The master switch is deflected in the positive direction. In so doing, the maximum positive target position (r64TargetPosition) of 120000 mm is specified. When the maximum positive target position is reached and therefore the actual velocity reaches the configured minimum velocity, then in the deceleration phase a changeover is made to the minimum velocity. The drive continues to move with the configured minimum velocity until the master switch is no longer deflected or the power-down command comes from the controller.

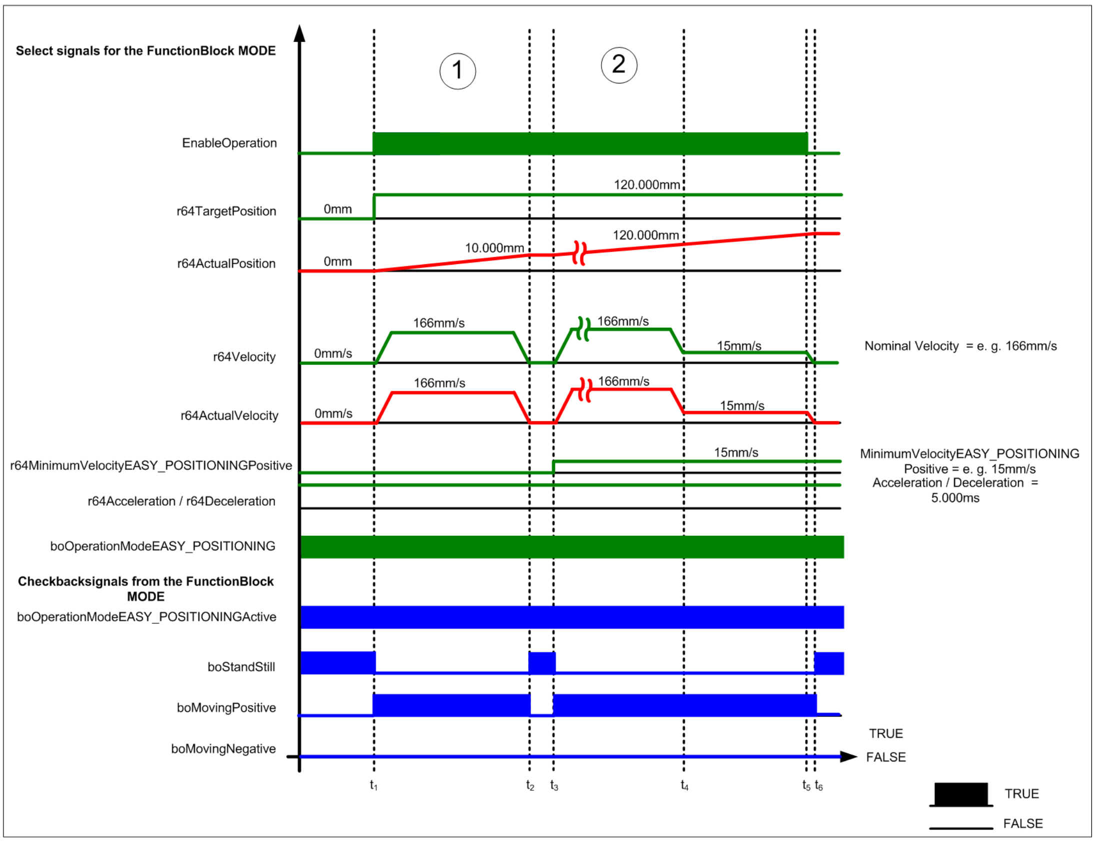

Select signals for the FunctionBlock MODE
1
2
EnableOperation
r64TargetPosition
0mm
120.000mm
r64ActualPosition
0mm
10.000mm
120.000mm
r64Velocity
0mm/s
166mm/s
166mm/s
15mm/s
Nominal Velocity = e. g. 166mm/s
r64ActualVelocity
0mm/s
166mm/s
166mm/s
15mm/s
r64MinimumVelocityEASY_POSITIONINGPositive
15mm/s
MinimumVelocityEASY_POSITIONING Positive = e. g. 15mm/s
Acceleration / Deceleration = 5.000ms
r64Acceleration / r64Deceleration
boOperationModeEASY_POSITIONING
Checkbacksignals from the FunctionBlock MODE
boOperationModeEASY_POSITIONINGActive
boStandStill
boMovingPositive
boMovingNegative
TRUE
FALSE
t1
t2
t3
t4
t5
t6
TRUE
FALSE

Figure 5-46 FB_OperationMode, EASY_POSITIONING operating mode, minimum velocity

Operating mode: SPEED_CONTROLLED
Control mode: Closed-loop speed controlled with/without synchronous operation

Note

The SPEED_CONTROLLED operating mode can be used instead of the previous JOGGING operating mode; however, the limitation to the required velocity must be realized in the PLC.

The SPEED_CONTROLLED operating mode is accepted only if the prerequisites for "selecting operating modes" are fulfilled.

In the SPEED_CONTROLLED operating mode, the axis moves with the velocity, ramp-up and ramp-down times specified from the S7. The travel direction is specified using the setpoint (PZD2). If the master switch is deflected in the positive direction, then a positive velocity must be specified (r64Velocity). If the master switch is deflected in the negative direction, then a negative velocity must be specified (r64Velocity).

Sequence diagram

Various sequences of SPEED_CONTROLLED operation are shown in the following sequence diagram. It should be noted that acceleration and deceleration are not changed in this sequence diagram.

Section 1:
In the first section of the sequence diagram, the master switch is deflected in the positive direction. The velocity (r64Velocity) is defined by the degree of deflection of the master switch. The master switch is slowly deflected in the positive direction up to approx. 50% and after a short constant-velocity phase is again slowly returned to standstill.

Section 2:
In the second section of the sequence diagram, the master switch is deflected in the negative direction. The velocity (r64Velocity) is defined by the degree of deflection of the master switch. The master switch is slowly deflected in the negative direction up to approx. -50 % and after a short constant-velocity phase is again slowly returned to standstill.

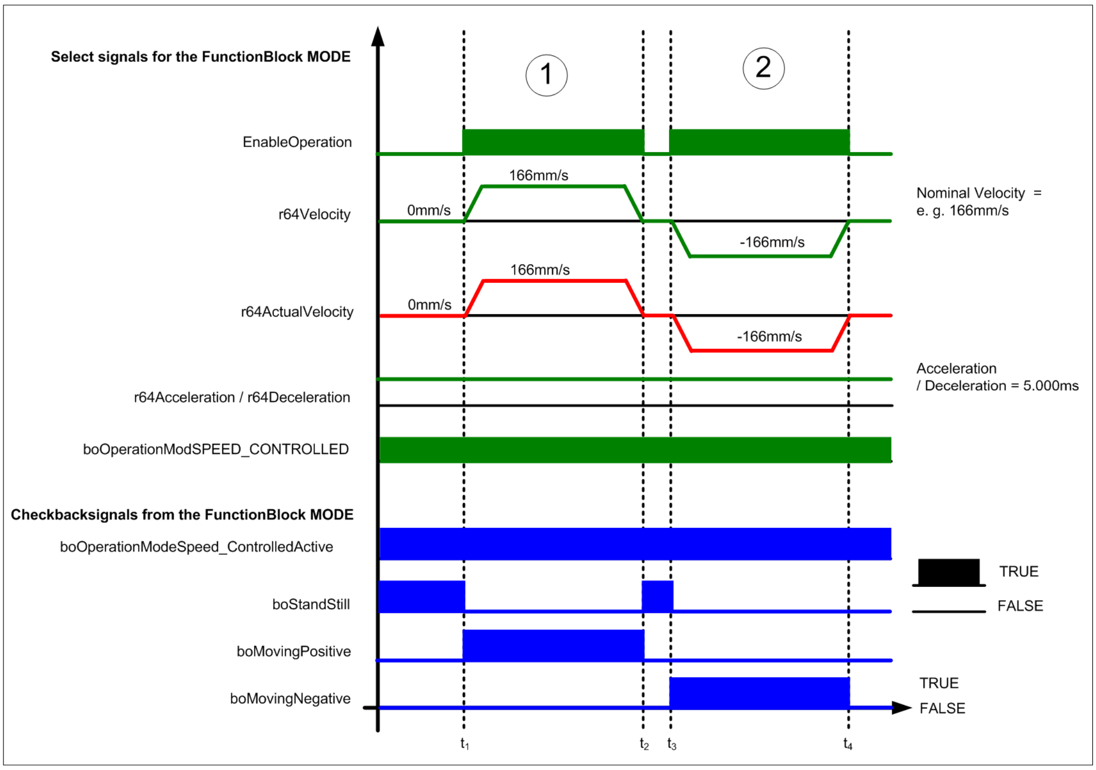


Figure 5-47 FB_OperationMode, SPEED_CONTROLLED operating mode

Operating mode: SENSORLESS EMERGENCY
Control mode: Closed-loop speed controlled without encoder or V/f control

The SENSORLESS EMERGENCY operating mode is only accepted if the prerequisites for "select operating modes" are fulfilled.

In the SENSORLESS EMERGENCY operating mode, the axis moves with the velocity and ramp-up and ramp-down times specified from the S7. S7 limits the velocity setpoint to a value that can be configured (r64VelocitySensorlessEmergency). The travel direction and velocity is defined by the deflection of the master switch.

Note

The V/f control is permissible only in the SENSORLESS EMERGENCY operating mode.

⚠WARNING

When using the V/f control type, the technology object (TO) must not be deleted and created as the V/f drive. The technology object (TO) must not be deleted. The p1300 parameter only needs to be parameterized in the drive object (DO) in the correct drive data set (DDS) for speed control with or without encoder on V/f control.

⚠WARNING

Active loads must not be operated without an encoder. "Active load" means that the load for opened brake accelerates automatically (Hoist, Boom, Holding Gear, etc.).

Sequence diagram

Various sequences of SENSORLESS EMERGENCY are shown in the following sequence diagram. It should be noted that acceleration and deceleration are not changed in this sequence diagram.

Section 1:
In the first section of the sequence diagram, the master switch is deflected in the positive direction. The velocity (r64Velocity) is defined by the degree of deflection of the master switch and by the velocity configured for SENSORLESS EMERGENCY. The master switch is slowly deflected in the positive direction up to approx. 50% but the velocity is limited to the velocity configured for SENSORLESS EMERGENCY and after a short constant-velocity phase is again slowly returned to standstill.

Section 2:
In the second section of the sequence diagram, the master switch is deflected in the negative direction. The velocity (r64Velocity) is defined by the degree of deflection of the master switch and by the velocity configured for SENSORLESS EMERGENCY (r64VelocitySENSORLESSEMERGENCY). The master switch is slowly deflected in the negative direction up to approx. -50%, but the velocity is limited to the velocity configured for SENSORLESS EMERGENCY and after a short constant-velocity phase is again slowly returned to standstill.

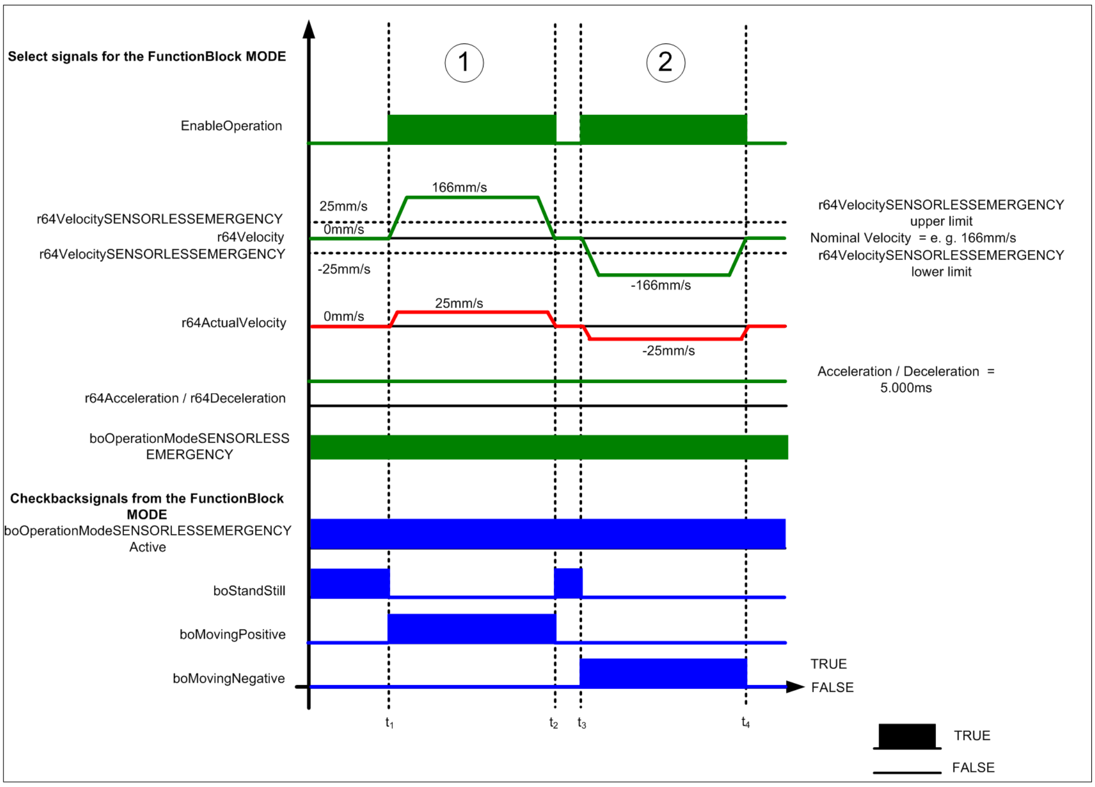


Figure 5-48 FB_OperationMode, SENSORLESS EMERGENCY operating mode

Operating mode: SWAYCONTROL
Control mode: Speed-controlled with encoder

The SWAYCONTROL operating mode is accepted only if the prerequisites for "selecting operating modes" are fulfilled.

In the SWAYCONTROL mode, all of the motion control tasks associated with the drive are performed by the Advanced Technology (sway control) and no longer by Basic Technology (refer to the figure below). The velocity defined by the S7 is forwarded to the relevant Sway Control block. The ramp-up and ramp-down times are fixed (permanently stored) using the CeComm calculation tool. For detailed information, refer to the SIMOCRANE Sway Control Operating Instructions (Ref. [12]).

In SWAYCONTROL operating mode, all DCC blocks are calculated, i.e. including the Basic Technology blocks. However, only the blocks described below are used in SWAYCONTROL operating mode, no others.

For hoisting gear applications, block DCC_LoadDependingFieldWeak is taken into account for the permissible maximum velocity. Block DCC_StartPulse can always be used regardless of the operating mode. For grabs, even more blocks (DCC_SlackRopeControl and DCC_CurrentEqualControl) are used independent of the operating mode.

Heavy-duty operation cannot be selected in SWAYCONTROL operating mode.

In SWAYCONTROL operating mode the technology object only checks the maximum velocity and acceleration. The acceleration and velocity limits must not be exceeded in the technology object if limits are in place. The technology object no longer carries out positioning, standstill, or following error monitoring.

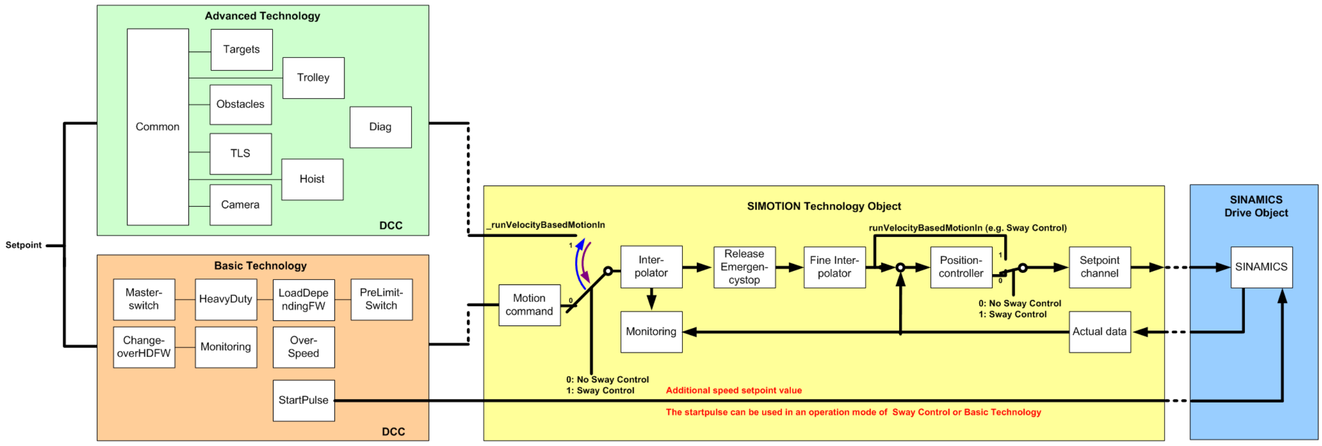


Figure 5-49 FB_OperationMode, SWAYCONTROL operating mode

5.3.9.5 Selecting operating modes

Prerequisites:

1. Only one operating mode is selected.
2. The drive is not being controlled (the drive is at standstill).
3. Activation of the technology object.
4. It is only necessary to home the axis when AUTOMATIC, MANUAL, EASY_POSITIONING or SWAYCONTROL is selected.

Note

If the selection of the operating mode is not plausible, however, no error message is output.

AUTOMATIC, MANUAL, EASY_POSITIONING or SWAYCONTROL operating mode can only be selected if the drive has been homed in compliance with the conditions specified above; in other words, if there is a clear reference between the measuring system and the travel path.

Table 5- 9 Checkback signal drive data set selection

Drive data set	bocheckbackDriveDataSet-Bit0active	bocheckbackDriveDataSet-Bit1active	bocheckbackDriveDataSet-Bit2active	Encoder present
0	False	False	False	Yes
1	True	False	False	No
2	False	True	False	Yes
3	True	True	False	No
4	False	False	True	Yes
5	True	False	True	No
6	False	True	True	Yes
7	True	True	True	Yes

The control of the operating modes is shown in various case examples in the following pages. The POWER signal indicates the power enable signal at the technology object.

AUTOMATIC operating mode

Case 1:
The drive is not homed, i.e. there is no reference between the measuring system and travel path. The AUTOMATIC operating mode is selected. The AUTOMATIC operating mode is not accepted because the drive has not been homed.

Case 2:
The SPEED_CONTROLLED operating mode is active and the axis moves. The AUTOMATIC operating mode is selected. The AUTOMATIC operating mode is not accepted because the drive is moving and is still in closed-loop control.

Case 3:
The OperationMode function block has an error. The AUTOMATIC operating mode is selected. The AUTOMATIC operating mode is accepted.

Case 4:
The AUTOMATIC operating mode is selected. No other operating mode is selected, the drive is stationary (zero speed), the drive is not being controlled, the OperationMode function block does not have an error, and drive data set 0 or 2 is active. AUTOMATIC ACTIVE is signaled back. This means that the operating mode was correctly accepted.

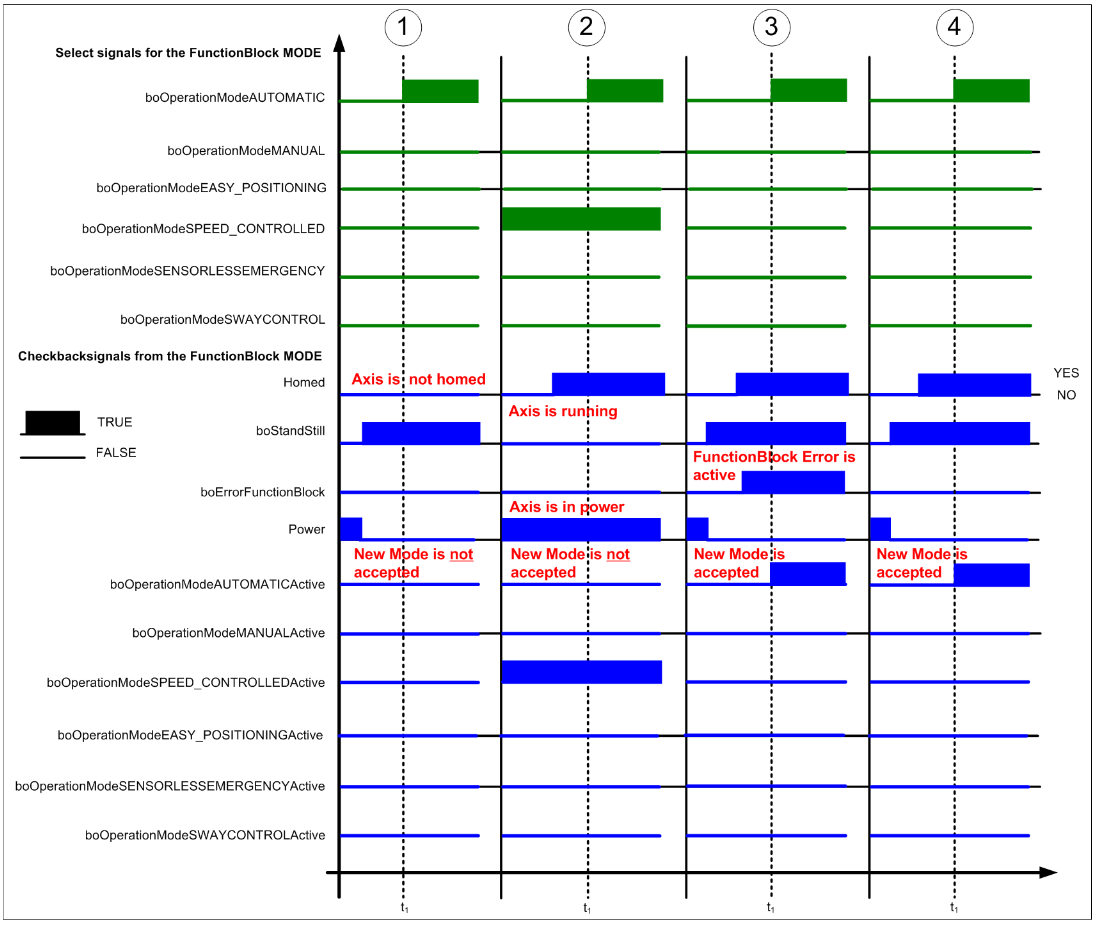

1
2
3
4
Select signals for the FunctionBlock MODE
boOperationModeAUTOMATIC
boOperationModeMANUAL
boOperationModeEASY_POSITIONING
boOperationModeSPEED_CONTROLLED
boOperationModeSENSORLESSEMERGENCY
boOperationModeSWAYCONTROL
Checkbacksignals from the FunctionBlock MODE
Homed
TRUE
FALSE
boStandStill
boErrorFunctionBlock
Power
boOperationModeAUTOMATICActive
boOperationModeMANUALActive
boOperationModeSPEED_CONTROLLEDActive
boOperationModeEASY_POSITIONINGActive
boOperationModeSENSORLESSEMERGENCYActive
boOperationModeSWAYCONTROLActive
Axis is not homed
Axis is running
FunctionBlock Error is active
Axis is in power
New Mode is not accepted
New Mode is not accepted
New Mode is accepted
New Mode is accepted
YES
NO
t1
t1
t1
t1

Figure 5-50 FB_OperationMode, AUTOMATIC operating mode

MANUAL operating mode

Case 1:

The drive is not homed, i.e. there is no reference between the measuring system and the travel path. The MANUAL operating mode is selected. The Manual operating mode is not accepted because the drive has still not been homed.

Case 2:

The SPEED_CONTROLLED operating mode is active and the axis moves. The MANUAL operating mode is selected. The MANUAL operating mode is not accepted because the drive is moving and is still in closed-loop control.

Case 3:

The OperationMode function block has an error. The MANUAL operating mode is selected. The MANUAL operating mode is accepted.

Case 4:
The MANUAL operating mode is selected. No other operating mode is selected, the drive is stationary (zero speed), the drive is not subject to closed-loop control, the OperationMode function block does not have an error, and drive data set 0 or 2 is active. MANUAL Active is signaled back. This means that the operating mode was correctly accepted.

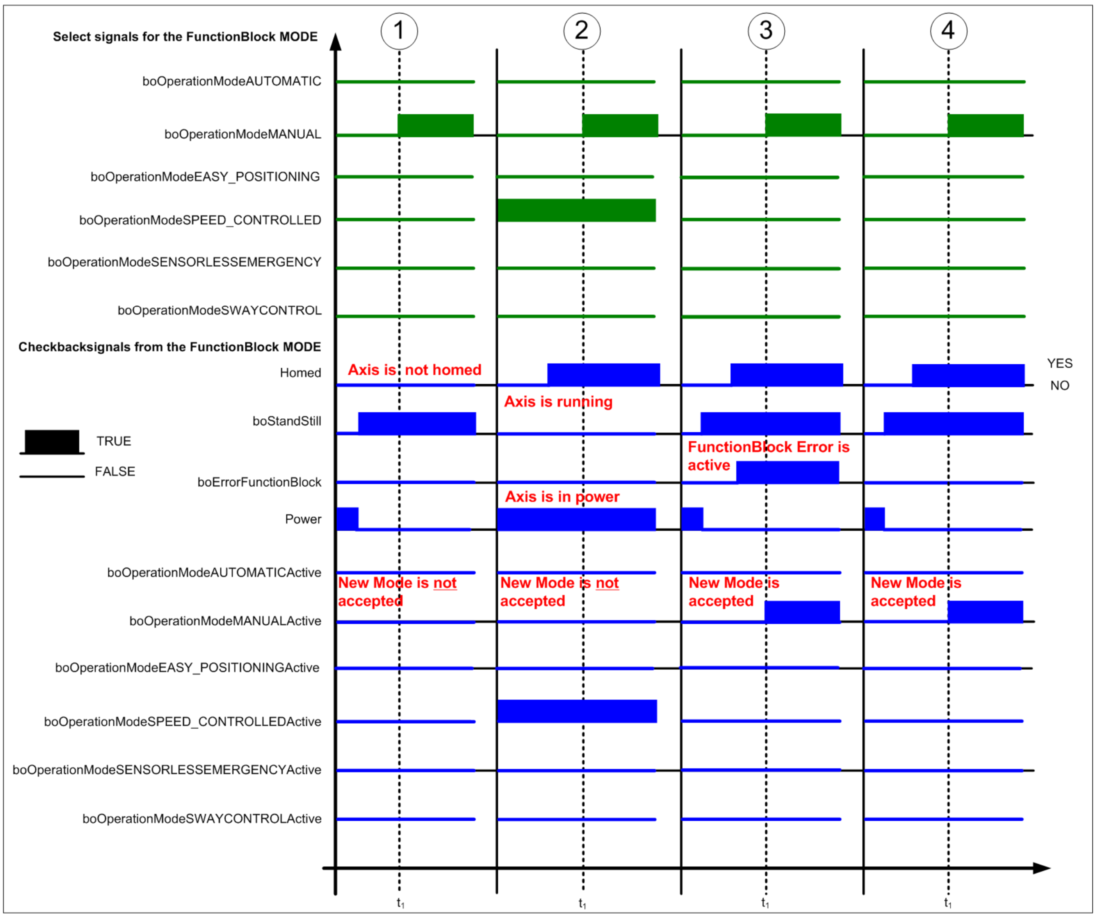

Select signals for the FunctionBlock MODE
boOperationModeAUTOMATIC
boOperationModeMANUAL
boOperationModeEASY_POSITIONING
boOperationModeSPEED_CONTROLLED
boOperationModeSENSORLESSEMERGENCY
boOperationModeSWAYCONTROL
Checkbacksignals from the FunctionBlock MODE
Homed
boStandStill
boErrorFunctionBlock
Power
boOperationModeAUTOMATICActive
boOperationModeMANUALActive
boOperationModeEASY_POSITIONINGActive
boOperationModeSPEED_CONTROLLEDActive
boOperationModeSENSORLESSEMERGENCYActive
boOperationModeSWAYCONTROLActive
TRUE
FALSE
YES
NO
1
2
3
4
Axis is not homed
Axis is running
FunctionBlock Error is active
Axis is in power
New Mode is not accepted
New Mode is not accepted
New Mode is accepted
New Mode is accepted
t1
t1
t1
t1

Figure 5-51 FB_OperationMode, MANUAL operating mode

EASY_POSITIONING operating mode

Case 1:
The drive is not homed, i.e. there is no reference between the measuring system and the travel path. The EASY_POSITIONING operating mode is selected. The EASY_POSITIONING operating mode is not accepted because the drive has not been homed.

Case 2:
The SPEED_CONTROLLED operating mode is active and the axis moves. The EASY_POSITIONING operating mode is selected. The EASY_POSITIONING operating mode is not accepted because the drive is moving and is still in closed-loop control.

Case 3:
The OperationMode function block has an error. The EASY_POSITIONING operating mode is selected. The EASY_POSITIONING operating mode is accepted.

Case 4:
The EASY_POSITIONING operating mode is selected. No other operating mode is selected, the drive is stationary (zero speed), the drive is not subject to closed-loop control, the OperationMode function block does not have an error, and drive data set 0 or 2 is active. EASY_POSITIONING active is returned. This means that the operating mode was correctly accepted.

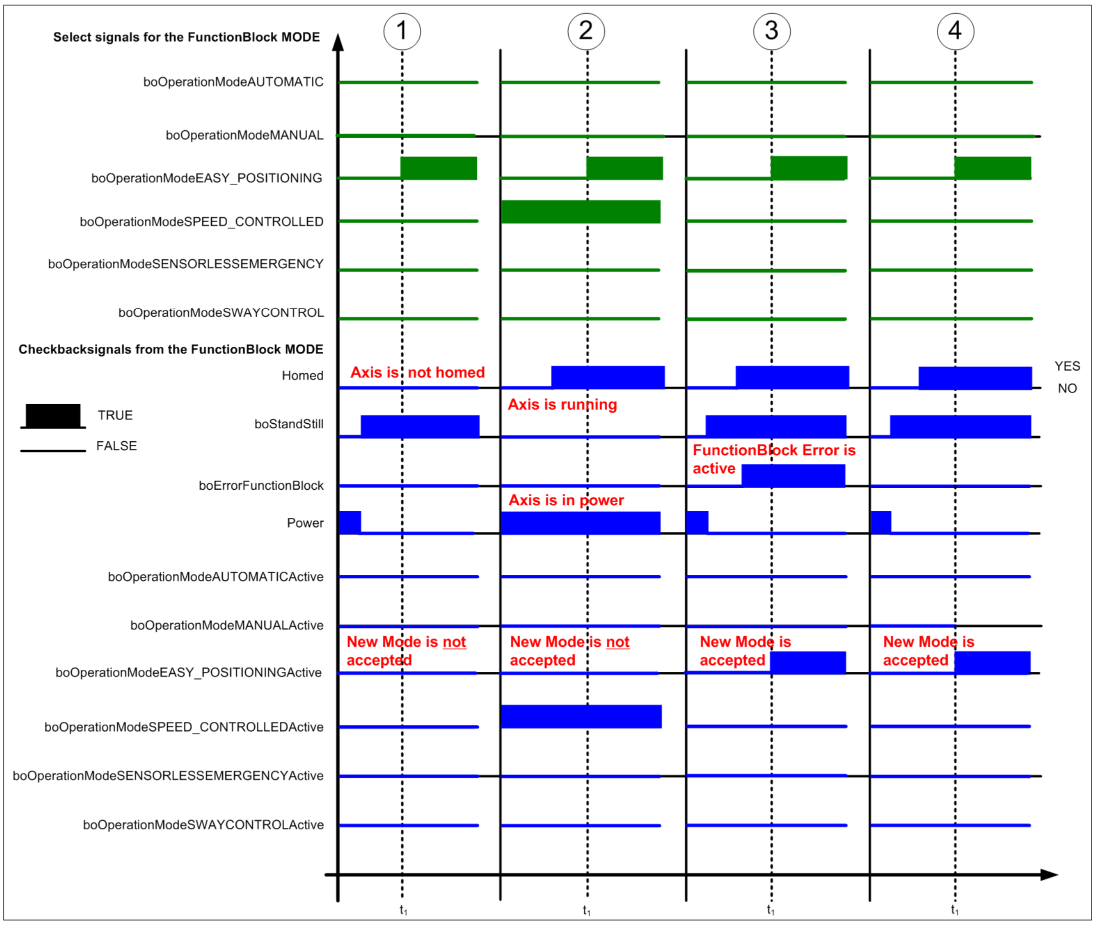

Select signals for the FunctionBlock MODE
1
2
3
4
boOperationModeAUTOMATIC
boOperationModeMANUAL
boOperationModeEASY_POSITIONING
boOperationModeSPEED_CONTROLLED
boOperationModeSENSORLESSEMERGENCY
boOperationModeSWAYCONTROL
Checkbacksignals from the FunctionBlock MODE
Homed
Axis is not homed
YES
NO
TRUE
FALSE
boStandStill
Axis is running
FunctionBlock Error is active
boErrorFunctionBlock
Axis is in power
Power
boOperationModeAUTOMATICActive
boOperationModeMANUALActive
New Mode is not accepted
New Mode is not accepted
New Mode is accepted
New Mode is accepted
boOperationModeEASY_POSITIONINGActive
boOperationModeSPEED_CONTROLLEDActive
boOperationModeSENSORLESSEMERGENCYActive
boOperationModeSWAYCONTROLActive
t_1
t_1
t_1
t_1

Figure 5-52 FB_OperationMode, EASY_POSITIONING operating mode

SPEED_CONTROLLED operating mode

Case 1:
SENSORLESS EMERGENCY mode is active and the axis is moving. The SPEED_CONTROLLED operating mode is selected. The SPEED_CONTROLLED operating mode is not accepted because the drive is moving and is still in closed-loop control.

Case 2:
The OperationMode function block has an error. The SPEED_CONTROLLED operating mode is selected. The SPEED_CONTROLLED operating mode is accepted.

Case 3:
The SPEED_CONTROLLED operating mode is selected. No other operating mode is selected, the drive is stationary (zero speed), the drive is not subject to closed-loop control, the OperationMode function block does not have an error, and drive data set 0 or 2 is active. Then, the SPEED_CONTROLLED ACTIVE feedback signal is output. This means that the operating mode was correctly accepted.

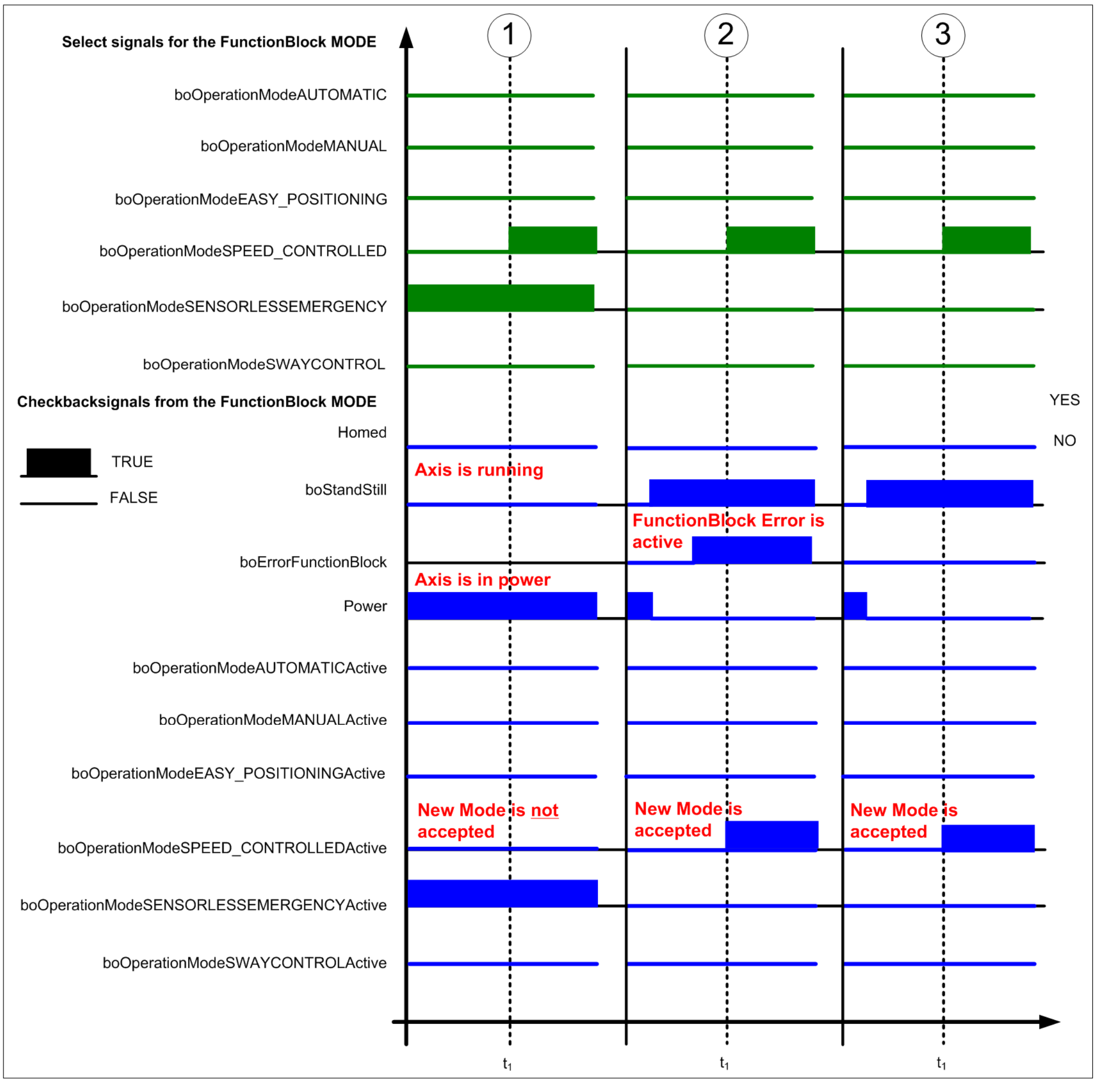


Figure 5-53 FB_OperationMode, SPEED_CONTROLLED operating mode

Operating mode SENSORLESS EMERGENCY

Case 1:
The SPEED_CONTROLLED operating mode is active and the axis moves. The SENSORLESS EMERGENCY operating mode is selected. The SENSORLESS EMERGENCY operating mode is not accepted because the drive is moving and is still in closed-loop control.

Case 2:
The OperationMode function block has an error. The SENSORLESS EMERGENCY operating mode is selected. The SENSORLESS EMERGENCY operating mode is accepted.

Case 3:
The SENSORLESS EMERGENCY operating mode is selected. No other operating mode is selected, the drive is stationary (zero speed), the drive is not in closed-loop control, the OperationMode function block does not have an error. A drive data set is switched over internally (DDS 1 or DDS 3) and then the SENSORLESS EMERGENCY ACTIVE feedback signal is output. This means that the operating mode was correctly accepted.

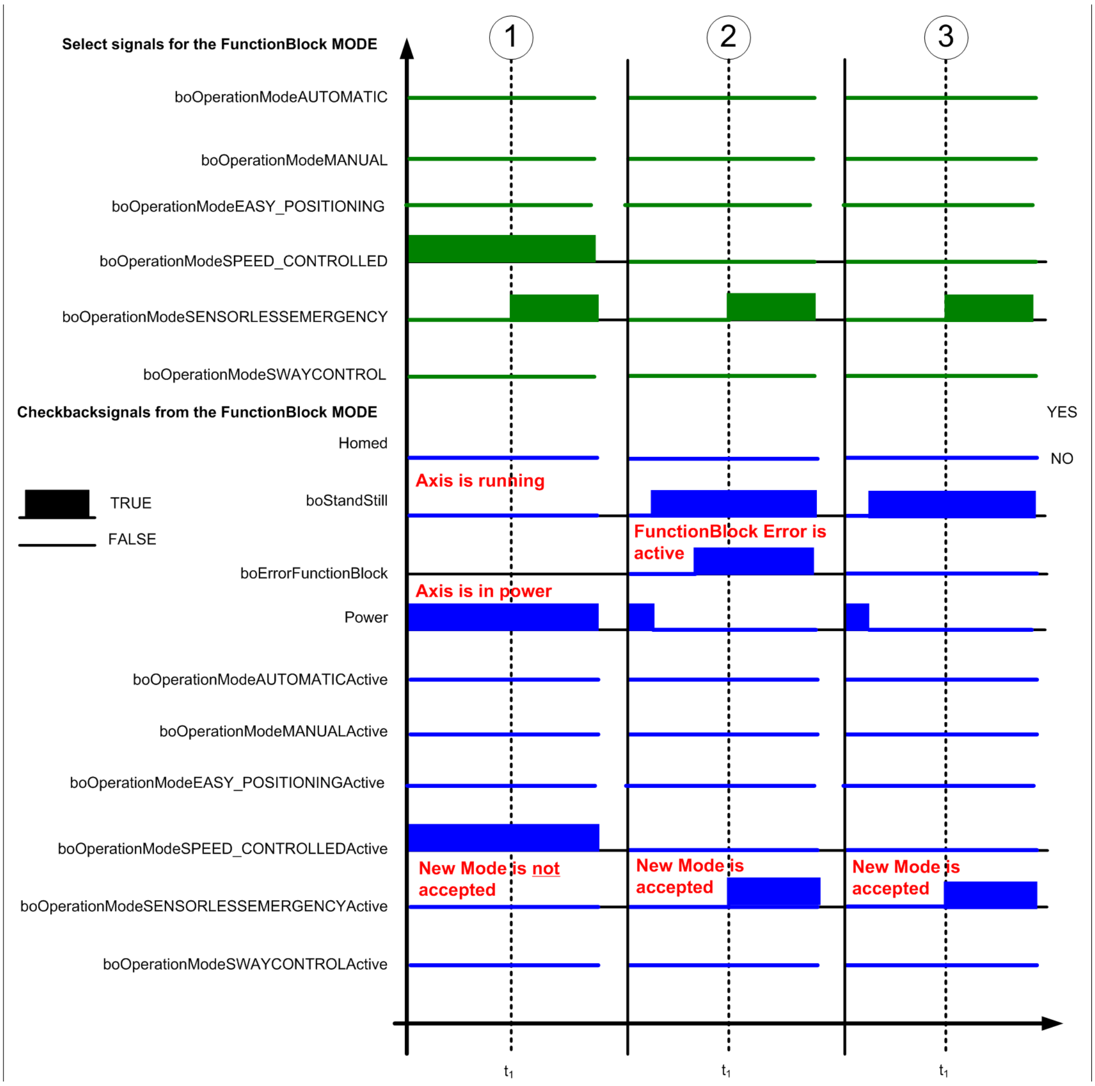

Select signals for the FunctionBlock MODE
1
2
3
boOperationModeAUTOMATIC
boOperationModeMANUAL
boOperationModeEASY_POSITIONING
boOperationModeSPEED_CONTROLLED
boOperationModeSENSORLESSEMERGENCY
boOperationModeSWAYCONTROL
Checkbacksignals from the FunctionBlock MODE
YES
Homed
NO
Axis is running
TRUE
boStandStill
FALSE
FunctionBlock Error is active
boErrorFunctionBlock
Axis is in power
Power
boOperationModeAUTOMATICActive
boOperationModeMANUALActive
boOperationModeEASY_POSITIONINGActive
boOperationModeSPEED_CONTROLLEDActive
New Mode is not accepted
New Mode is accepted
New Mode is accepted
boOperationModeSENSORLESSEMERGENCYActive
boOperationModeSWAYCONTROLActive
t1
t1
t1

Figure 5-54 FB_OperationMode, SENSORLESS EMERGENCY operating mode

SWAYCONTROL operating mode

Case 1:

The drive is not homed, i.e. there is no reference between the measuring system and travel path. SWAYCONTROL mode is selected. SWAYCONTROL mode is not accepted because the drive has not been homed.

Case 2:

The SPEED_CONTROLLED operating mode is active and the axis moves. SWAYCONTROL mode is selected. SWAYCONTROL mode is not accepted because the drive is moving and is still under control.

Case 3:
The OperationMode function block has an error. SWAYCONTROL mode is selected. The SWAYCONTROL operating mode is accepted.

Case 4:
Heavy-duty operation is active and SWAYCONTROL mode is selected. SWAYCONTROL mode is not accepted because the drive is in heavy-duty operation.

Case 5:
SWAYCONTROL mode is selected. No other operating mode is selected, the drive is stationary (zero speed) and referenced, the drive is not in closed-loop control and drive data set 0 or 2 is active. SWAYCONTROL ACTIVE is fed back. This means that the operating mode was correctly accepted.

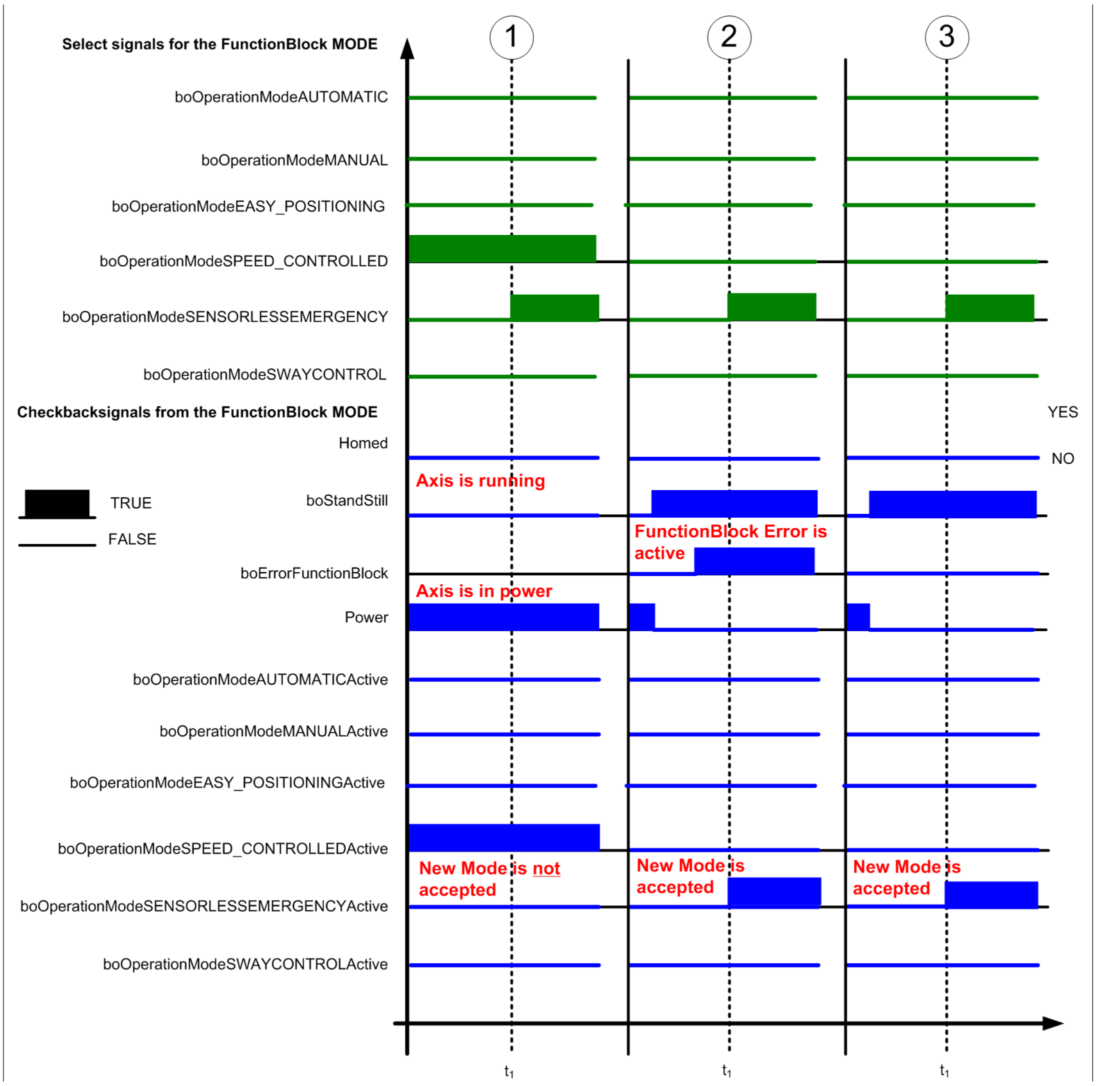

Select signals for the FunctionBlock MODE
1
2
3
boOperationModeAUTOMATIC
boOperationModeMANUAL
boOperationModeEASY_POSITIONING
boOperationModeSPEED_CONTROLLED
boOperationModeSENSORLESSEMERGENCY
boOperationModeSWAYCONTROL
Checkbacksignals from the FunctionBlock MODE
YES
NO
Homed
Axis is running
TRUE
boStandStill
FALSE
FunctionBlock Error is active
boErrorFunctionBlock
Axis is in power
Power
boOperationModeAUTOMATICActive
boOperationModeMANUALActive
boOperationModeEASY_POSITIONINGActive
boOperationModeSPEED_CONTROLLEDActive
New Mode is not accepted
New Mode is accepted
New Mode is accepted
boOperationModeSENSORLESSEMERGENCYActive
boOperationModeSWAYCONTROLActive
t1
t1
t1

Figure 5-55 FB_OperationMode, SWAYCONTROL operating mode

5.3.9.6 Homing

When the power is switched in at the system, generally, the motor encoder and the travel path are not referenced with one another - unless an absolute measuring system for SIMOTION is being used. This means that the motor encoder transfers an actual position to the controller that does not correspond to the actual position of the travel path. Homing is mandatory in the AUTOMATIC, MANUAL, EASY_POSITIONING and SWAYCONTROL operating modes for this reason. Without homing, these operating modes do not become active.

Homing is only possible if the drive has no power enable (power is inactive). There are two homing possibilities:

Possibility 1:
The homing position is specified by the "S7" controller by means of target position "r64TargetPosition". The input "boHomingValuePLC" must be set. The homing position is only accepted from the target position "r64TargetPosition" after this input has been set. The input "boStartHoming" is then started with a positive edge. This means that the actual position "r64ActualPosition" is set to the value of the target position. The output "boAxisHomed" is set after the operation has been completed.

Possibility 2:
The homing position is permanently saved in the variable "r64HomingFixValue". It is not permissible that the input "boHomingValuePLC" is set. The input "boStartHoming" is then started with a positive edge. This means that the actual position "r64ActualPosition" is set to a permanently saved value. The output "boAxisHomed" is set after the operation has been completed.

Note

The input "boStartHoming" is only briefly set using a positive edge.

The input "boHomingValuePLC" is used to select whether the homing value should be taken from the target position "r64targetPosition" of the "S7" controller or from the permanently saved value "r64HomingFixValue".

The output "boAxisHomed" indicates whether a drive has been referenced or not.

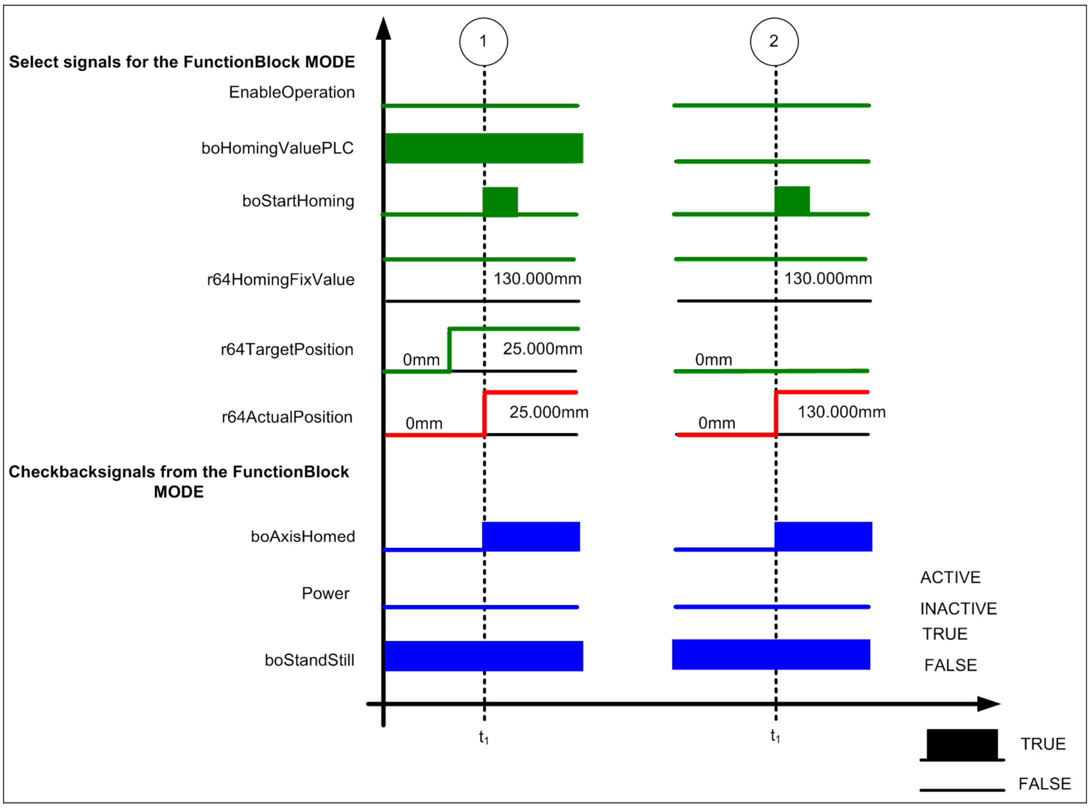


Figure 5-56 FB_OperationMode, homing

5.3.9.7 Master-slave operation control type

The master-slave operation control type is used if two motors are connected to a common shaft (refer to the figure below). The master operates either closed-loop position controlled or closed-loop speed controlled depending on the operating mode. The slave only operates closed-loop torque controlled. The master sends the torque as torque setpoint to the slave.

The drive that operates either closed-loop position controlled or closed-loop speed controlled is the master. The MasterMode must be selected in this module (boSlaveMode = FALSE). For the master, input "boSelectMasterSlaveOperation" must be set and the output "boMasterSlaveOperationActive" must be active. This means that the master does not immediately start to operate, but waits for output signals from the slave. The following output signals from the slave are checked:

1. The active operating mode of the slave is checked; the master and slave must have identical operating modes.
2. The power enable from the slave is checked.
3. Master-slave operation must be active.

The drive that operates closed-loop torque controlled is the slave. The SlaveMode must be selected in this module, i.e. input "boSlaveMode = TRUE" and the output "boSlaveModeActive" must be active. This means that it is also possible to select master-slave operation "boSelectMasterSlaveOperation".

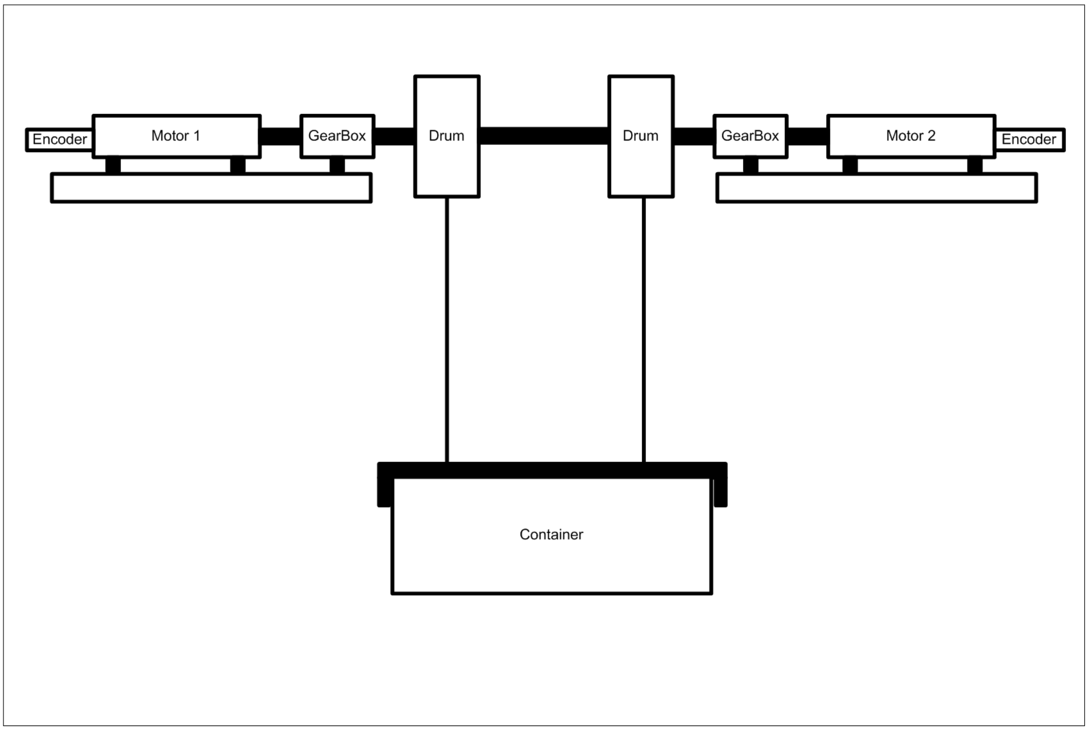


Figure 5-57 Master-slave operation control type

Selection and deselection in the master-slave operation control type for master and slave are illustrated in the sequence diagrams below:

Selecting the master-slave operation control type

The selection is made, if

- there is no power enable (Power = FALSE, STW1 bit 3).
- the drive is at a standstill (zero speed) (boStandStill = TRUE, AppZSW1 bit 2).
- the master and slave operating modes are the same.

Control word / status word	Bit	Designation	Master (Hoist_1)	SLAVE (Hoist_2)
Appl_STW1	6	Selecting master-slave operation	TRUE	TRUE
	7	Selecting synchronous operation	FALSE	FALSE
Appl_STW2	13	SlaveMode	FALSE	TRUE
Appl_ZSW1	6	Message, master-slave operation active	TRUE	TRUE
	7	Message, synchronous operation active	FALSE	FALSE
Appl_ZSW2	13	SlaveModeActive	FALSE	TRUE

Master-slave operation selection is now active, i.e. the two drives now operate in this mode. The master is normally controlled as in the operating modes that have already been described.

Note

For the slave, no setpoint may be specified and the STW1 must be set as for the master. The "boSelectMasterSlaveOperation" input must be set for the master and slave.

Note

For master-slave operation, the control by the control word 1 must remain active until both drives have come to a standstill.
The setpoint enable (STW1, bit 6) must not be reset if the drives are in motion.

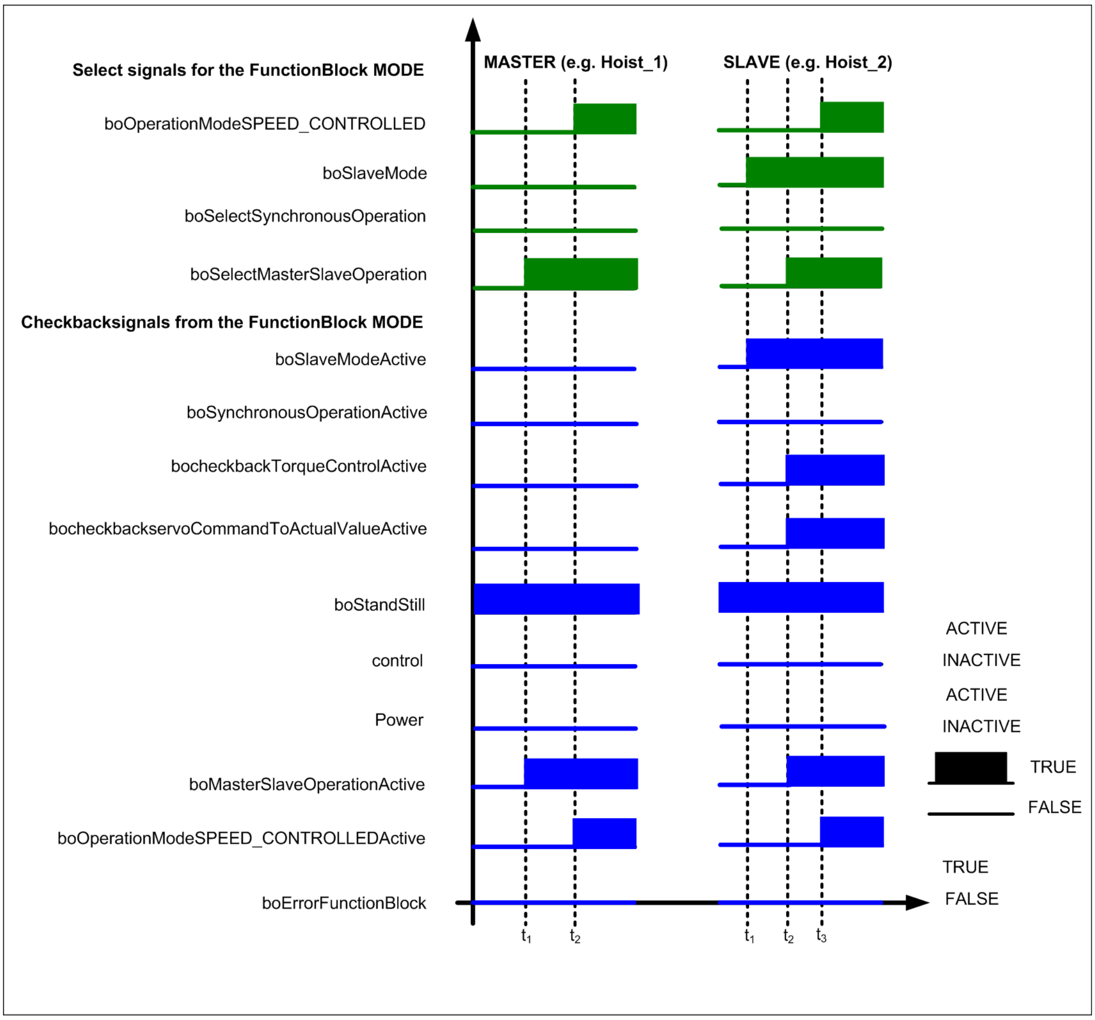

Select signals for the FunctionBlock MODE
boOperationModeSPEED_CONTROLLED
boSlaveMode
boSelectSynchronousOperation
boSelectMasterSlaveOperation
Checkbacksignals from the FunctionBlock MODE
boSlaveModeActive
boSynchronousOperationActive
bocheckbackTorqueControlActive
bocheckbackservoCommandToActualValueActive
boStandStill
control
Power
boMasterSlaveOperationActive
boOperationModeSPEED_CONTROLLEDActive
boErrorFunctionBlock
MASTER (e.g. Hoist_1)
SLAVE (e.g. Hoist_2)
t1
t2
t1
t2
t3
ACTIVE
INACTIVE
ACTIVE
INACTIVE
TRUE
FALSE
TRUE
FALSE

Figure 5-58 Selecting the master-slave operation control type

Deselecting the master-slave operation control type

Master-slave operation is deactivated if there is no power enable (Power = FALSE) and the drive is at a standstill (zero speed). Input "boSelectMasterSlaveOperation" is reset for both the master and slave. Then, the output "boMasterSlaveOperationActive" must be inactive. In addition, for the slave the "boSlaveMode" input must be reset. This means that master-slave operation is de-selected.

Note

If the master needs to be moved by itself after the master-slave operation control type has been deselected, the "boSelectMasterSlaveOperation" input must be reset. The master can only move alone if the output "boSelectMasterSlaveOperationActive" has been reset (the output signals from the slave are no longer checked).

NOTICE

In the master-slave operation control mode, the setpoint and the position may be defined **at the master only**. The setpoint that is entered must match the position. If this requirement is not met, the drive command will be withdrawn.

The torque setpoint of the slave (p1503) is permanently connected to the torque setpoint of the master (r0079). However, this does not have any effect on the selection of master-slave operation.

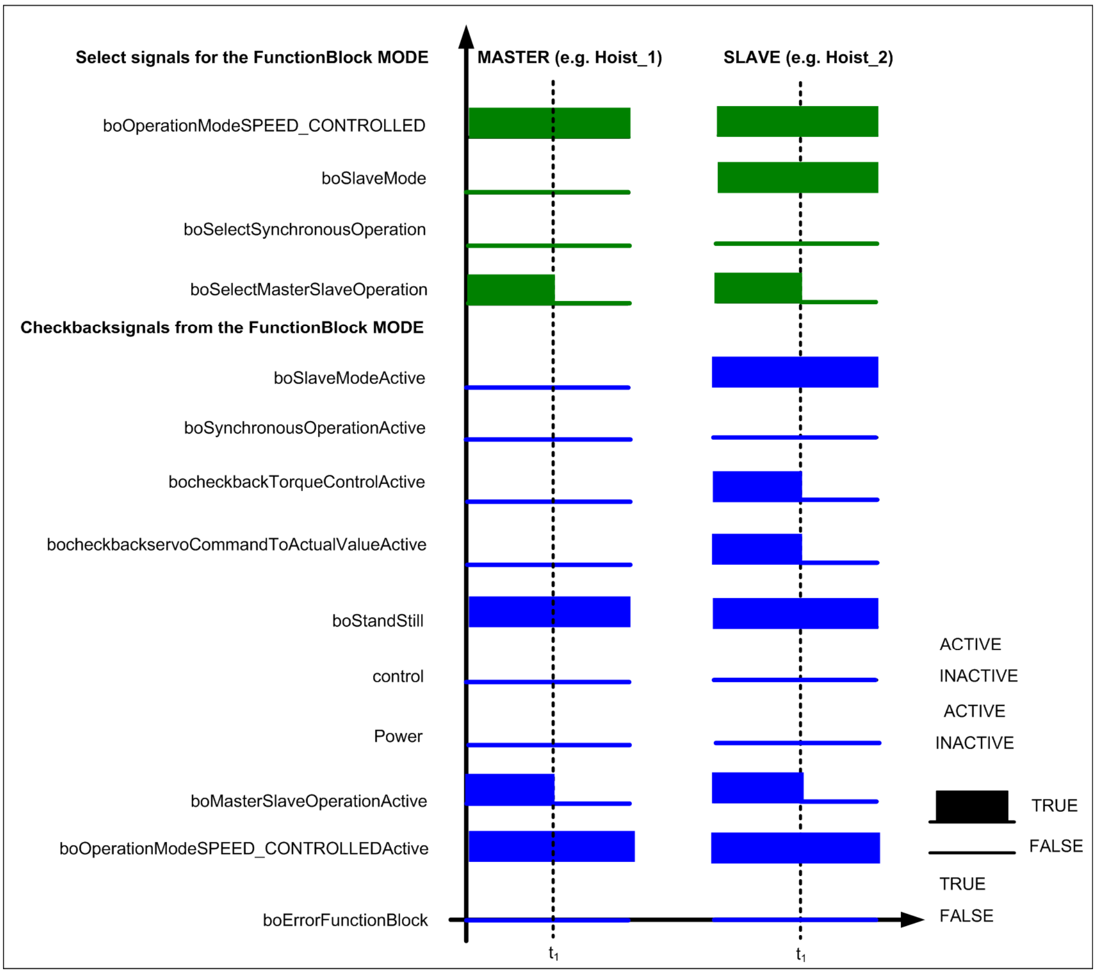


Figure 5-59 Deselecting the master-slave operation control type

5.3.9.8 Synchronous operation control type

The synchronous operation control type (SynchronousOperation) is used if two motors are connected to a common load (refer to the figure below). Depending on the operating mode, the master and slave operate as a function of the operating modes "closed-loop position controlled" or "closed-loop speed controlled". The slave receives from the master through a gear (gear ratio 1 : 1) depending on the operating mode, either a speed or position setpoint.

Two fixed gears must be created for each synchronous mode coupling (Hoist, Gantry, Trolley) in the SIMOTION SCOUT project "Technology". One of the fixed gears must be set up for speed-controlled operation ("Ignore position" must be selected). The second fixed gear must be set up for position-controlled operation.

Both fixed gears are configured **only** for the slave at the OperationMode block (TO_GearVel and TO_GearPos) in the user program.

The fixed gears are interconnected as follows: The fixed gears are interconnected with the master by way of setpoint coupling. The motion output of the corresponding fixed gear is used to connect the slave as the motion input (see Ref. [9], Chapter 1.1).

Synchronous operation operates as a function of the operating mode either as closed-loop speed controlled synchronous operation or as closed-loop position controlled synchronous operation.

In closed-loop speed controlled synchronous operation, the master transfers the velocity to the slave via a gear (gear ratio of 1 : 1).

In closed-loop position controlled synchronous operation, the master transfers the position value to the slave via a gear (gear ratio of 1 : 1).

The master module must be in the master mode, i.e. it is not permissible that input "boSlaveMode" is set and it is not permissible that the output "boSlaveModeActive" is active.

For the master, the input "boSelectSynchronousOperation" must be set and the output "boSynchronousOperationActive" must be active. This means that the master does not immediately start to operate, but waits for output signals from the slave. The following output signals from the slave are checked:

1. The active operating mode of the slave is checked; the master and slave must have identical operating modes.
2. The power enable from the slave is checked.
3. Synchronous operation must be active.

The slave module must be in the slave mode, i.e. the input "boSlaveMode" must be set and the output "boSlaveModeActive" must be active. This means that it is also possible to select synchronous operation "boSelectSynchronousOperation".

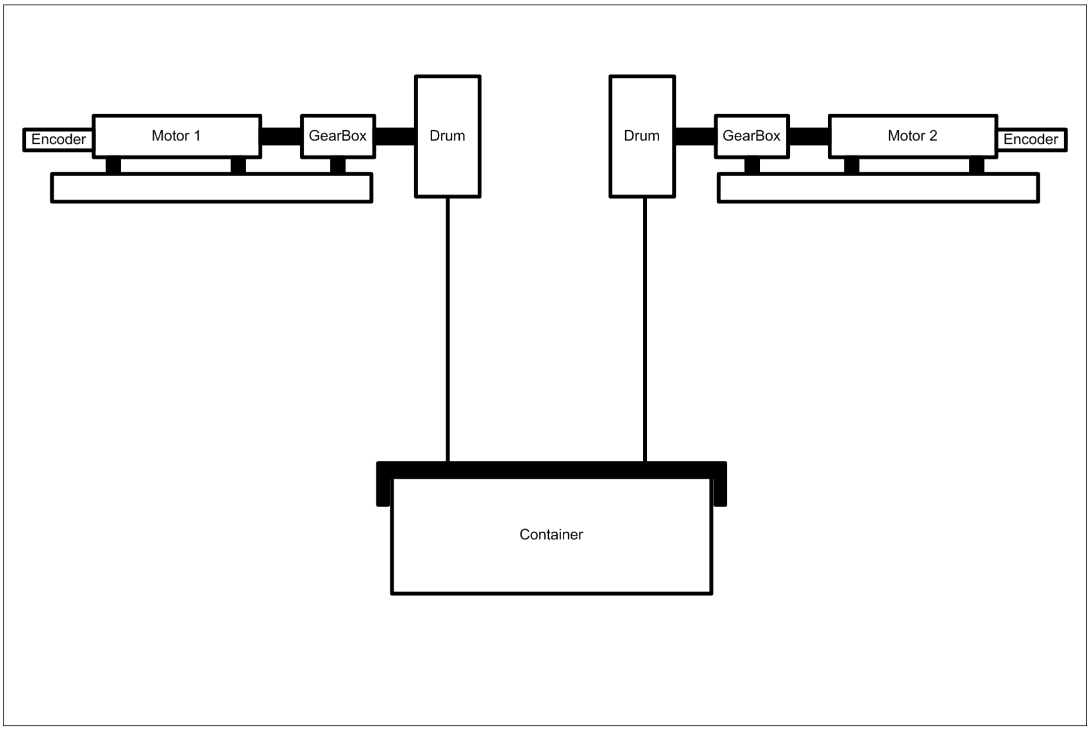


Figure 5-60 Synchronous operation control type

A parameter in the slave – e.g. hoist 2 – is configured for the permissible position difference for synchronous operation.

Parameter threshold 1 (r64toleranceMasterSlavePosition):

Position difference between master and slave within parameter threshold 1. Synchronous operation can then be started.

Position difference between the master and slave outside parameter threshold 1. Synchronous operation must then not be started. AppZSW2 bit 11 signals back whether the position difference is outside the tolerance of "r64ToleranceMasterSlavePosition".

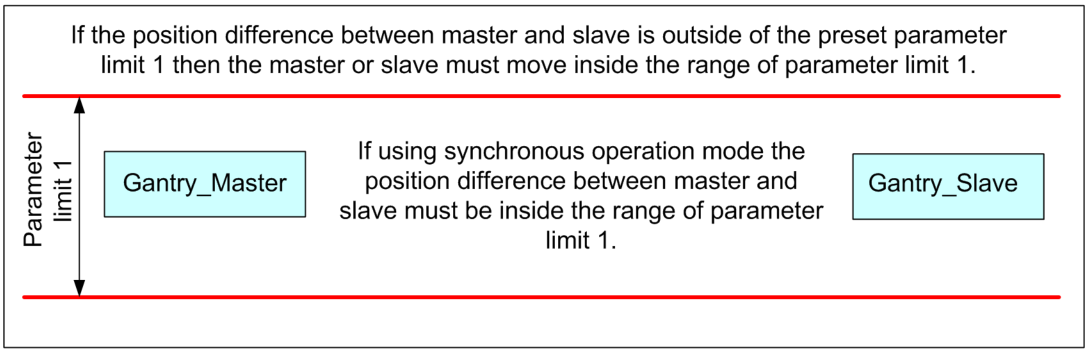

If the position difference between master and slave is outside of the preset parameter limit 1 then the master or slave must move inside the range of parameter limit 1.
Parameter limit 1
Gantry_Master
If using synchronous operation mode the position difference between master and slave must be inside the range of parameter limit 1.
Gantry_Slave

Figure 5-61 Synchronous operation control type – parameters

Selection and deselection for the master-slave operation for master and slave are shown in the sequence diagrams:

Selecting the synchronous operation control type

Synchronous operation control type is selected, if

- there is no power enable (Power = FALSE, STW1 bit 3).
- the drive is at a standstill (zero speed) (boStandStill = TRUE, AppZSW1 bit 2).
- the master and slave operating modes are the same.

Control word / status word	Bit	Designation	Master (Hoist_1)	SLAVE (Hoist_2)
Appl_STW1	6	Selecting master-slave operation	FALSE	FALSE
	7	Selecting synchronous operation	TRUE	TRUE
Appl_STW2	13	SlaveMode	FALSE	TRUE
Appl_ZSW1	6	Message, master-slave operation active	FALSE	FALSE
	7	Message, synchronous operation active	TRUE	TRUE
Appl_ZSW2	13	SlaveModeActive	FALSE	TRUE

The selection of the synchronous operation control type is now active, i.e. both drives now operate with this control type. The master is normally controlled as in the operating modes that have already been described.

Note

Synchronous operation is closed-loop position controlled in the AUTOMATIC and MANUAL operating modes.

Synchronous operation is controlled in a closed speed loop in the SPEED_CONTROLLED and EASY_POSITIONING operating modes.

Synchronous operation should not be activated in the SENSORLESS EMERGENCY operating mode.

Input "boSelectSynchronousOperation" must be set for both the master and slave.

For the master, it must be ensured in FB_OperationMode that the checkback signals of the slave (bocheckbackSlave...) are correctly connected.

For the slave, the technology object Fixed Gear "TO_GearPos" / "TO_GearVel" must be configured in FB_OperationMode.

Note

In the EASY_POSITIONING operating mode, the synchronous operation is speed-controlled and so the positions of the two drives not equalized using TO_GearVel. If a position synchronous operation is required in EASY_POSITIONING operating mode, the FB_ReferenceMode block must also be made operational. This block is used to equalize the positions taking into account of the desired offset; refer to FB_ReferenceMode (Page 273), "Functionality" section.

Note

For synchronous coupling, control by the control word 1 must remain active until both drives have come to a standstill.
The setpoint enable (STW1, bit 6) must not be reset when the drives are in motion.
If the control by the control word 1 is reset during the motion of the drives, an offset between the master and the slave would occur.

NOTICE

In the synchronous operation control mode, the setpoint and the position may be defined **at the master only** - and the STW1 must be set for both the slave and the master.

The setpoint that is entered must match the position. If this requirement is not met, the drive command will be withdrawn.

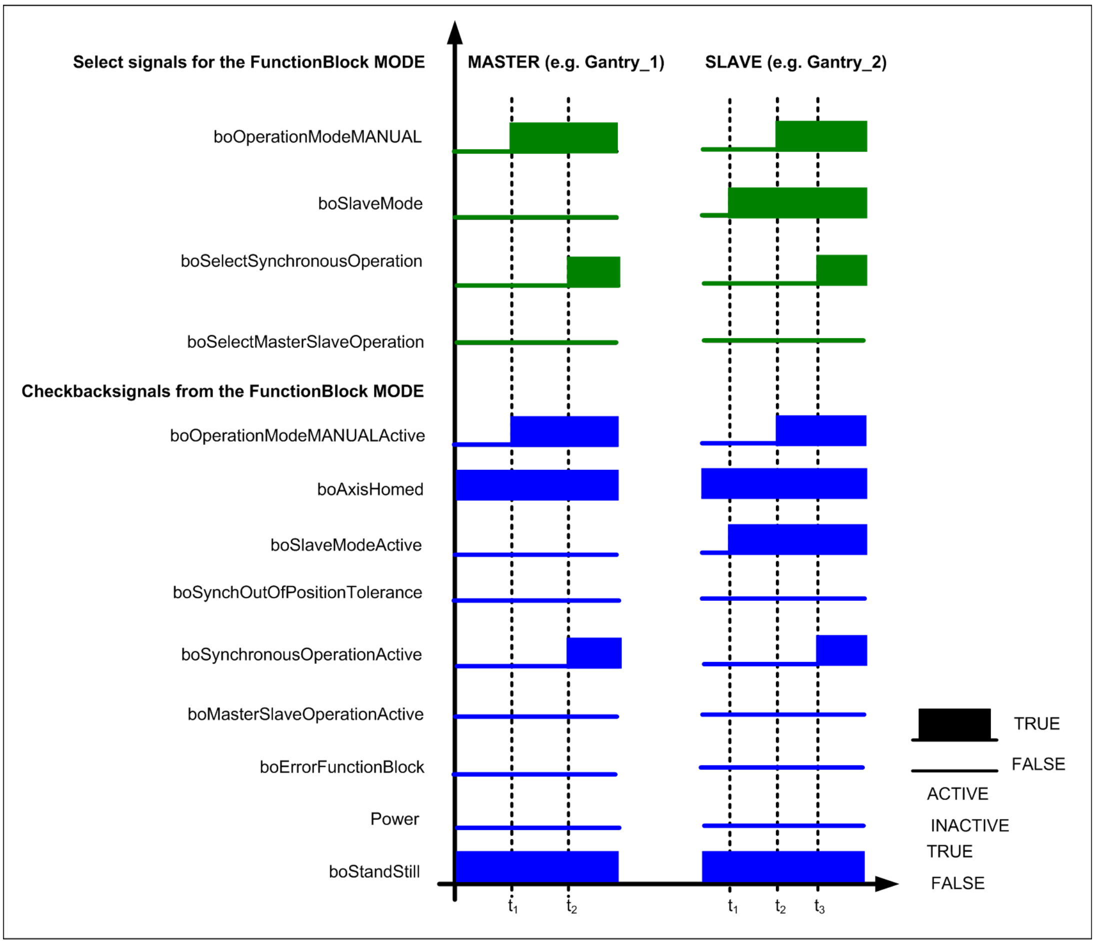

Select signals for the FunctionBlock MODE
MASTER (e.g. Gantry_1)
SLAVE (e.g. Gantry_2)
boOperationModeMANUAL
boSlaveMode
boSelectSynchronousOperation
boSelectMasterSlaveOperation
Checkbacksignals from the FunctionBlock MODE
boOperationModeMANUALActive
boAxisHomed
boSlaveModeActive
boSynchOutOfPositionTolerance
boSynchronousOperationActive
boMasterSlaveOperationActive
boErrorFunctionBlock
Power
boStandStill
TRUE
FALSE
ACTIVE
INACTIVE
TRUE
FALSE
t_1
t_2
t_1
t_2
t_3

Figure 5-62 Selecting synchronous operation (position controlled)

Deselecting the synchronous operation control type

The synchronous operation control type is deselected if there is no power enable "Power = FALSE" and the drives are at a standstill "boStandStill = TRUE". Input "boSelectSynchronousOperation" is reset for both the master and slave. The output "boSynchronousOperationActive" must then be inactive. And finally, for the slave the "boSlaveMode" must be reset. This means that the synchronous operation control type is deselected.

Note

If the master should move alone after synchronous operation control has been deselected, then the input "boSelectSynchronousOperation" must be reset. The master can only move alone after the output "boSlaveAvailableActive" has been reset, i.e. the output signals from the slave are no longer checked.

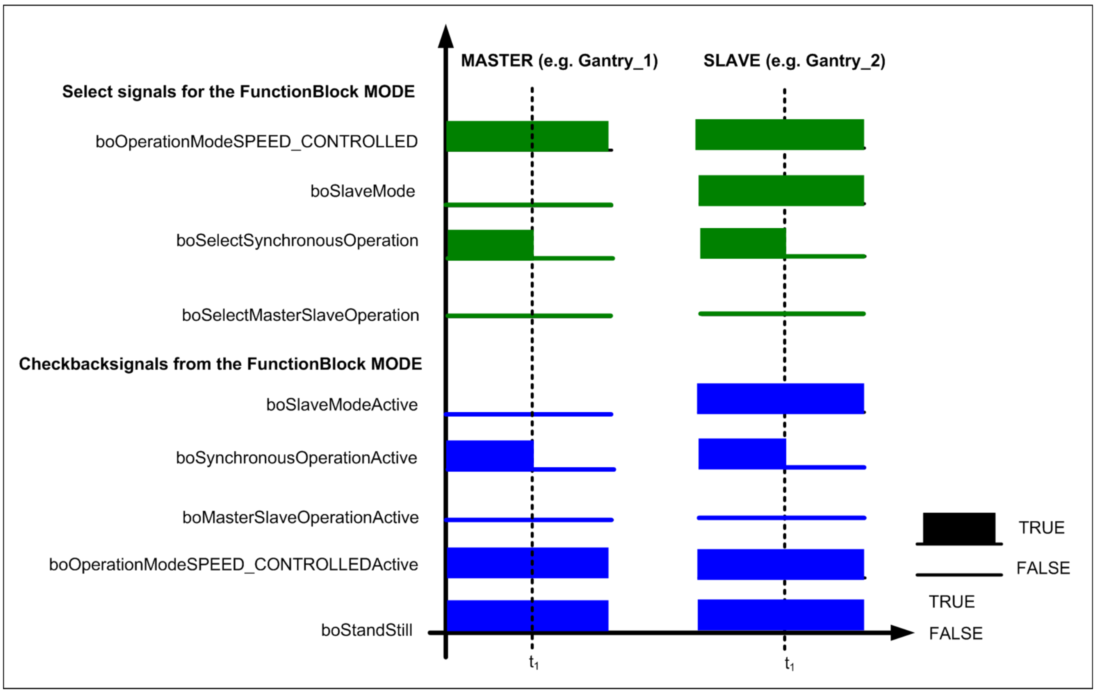

MASTER (e.g. Gantry_1)
SLAVE (e.g. Gantry_2)
Select signals for the FunctionBlock MODE
boOperationModeSPEED_CONTROLLED
boSlaveMode
boSelectSynchronousOperation
boSelectMasterSlaveOperation
Checkbacksignals from the FunctionBlock MODE
boSlaveModeActive
boSynchronousOperationActive
boMasterSlaveOperationActive
boOperationModeSPEED_CONTROLLEDActive
boStandStill
TRUE
FALSE
TRUE
FALSE
t1
t1

Figure 5-63 De-selecting the synchronous operation control type (closed-loop speed controlled)

Selecting the synchronous operation control type with offset

In practice, there are applications where an offset must be established between the master and slave. As a result of this offset, a container can then be lifted out of a ship even if the ship is not in a horizontal position.

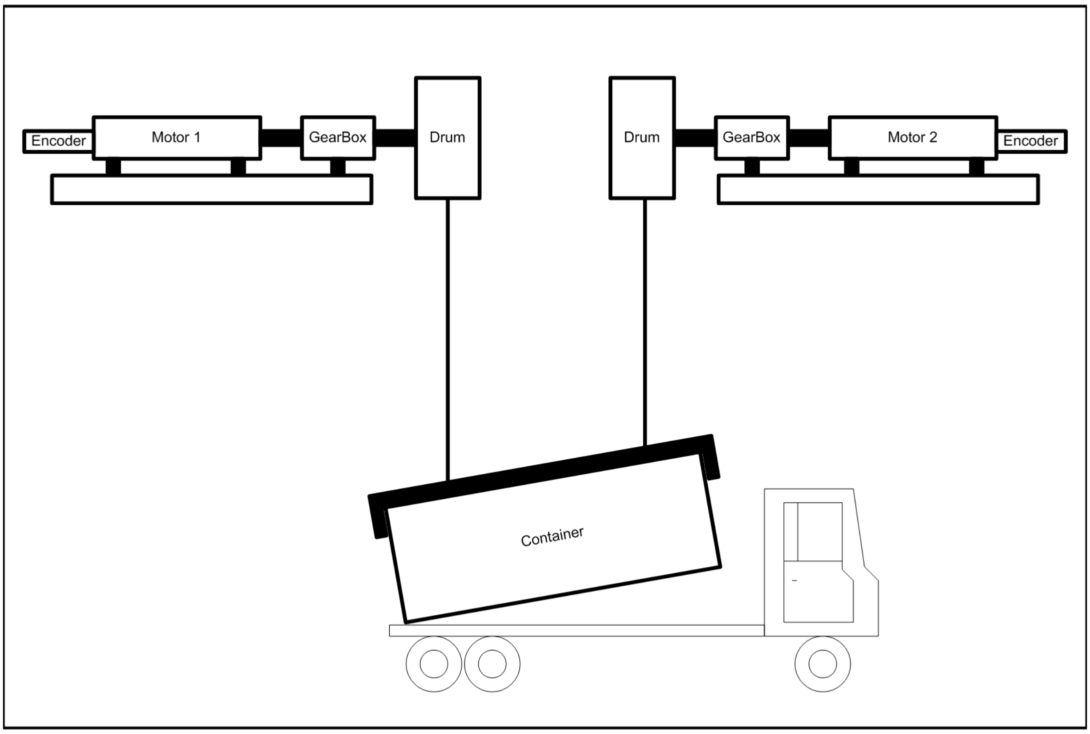


Figure 5-64 Synchronous operation control type with offset

Move the required offset between the master and slave without synchronous operation. Once the required offset has been reached, select synchronous operation as described previously.

If synchronous operation at the master and slave is active, the input "boSelectSynchronousOffsetSave" (AppSTW2 bit 8) at the slave must be set with a positive edge. This means that the offset between the master and slave is determined and saved. This offset is kept for the complete travel.

The target position is corrected if the slave is closer to the target position than the master. Otherwise, the slave would travel beyond the target. Two cases for raising and lowering are shown below:

Case 1: Raising: Target position = +10,000 mm

1. Master is located at the position = +5,000 mm
2. Slave is located at the position = +6,000 mm

→ Corrected target position = +9,000 mm

Case 2: Lowering: Target position = 10,000 mm

1. Master is located at the position = -5,000 mm
2. Slave is located at the position = -6,000 mm

→ Corrected target position = -9,000 mm

Deselecting the synchronous operation control type with offset

The offset value is only deleted if active synchronous operation is deselected (boSynchronousOperationActive" = FALSE).

The conditions for deselecting synchronous operation are described above.

Note

The offset can only be saved if synchronous operation is closed-loop position controlled - i.e. only in the AUTOMATIC and MANUAL operating modes.

5.3.9.9 Switching over from a motor encoder to an external encoder

In many applications, an axis is not only controlled using a motor encoder, but also an external encoder. In this case, the motor encoder is used for the closed-loop speed control and the external encoder for the closed-loop position control. More precise positioning is achieved using the external encoder. An external encoder can, for example, be mounted onto a non-driven wheel or on a cable drum. This increases the redundancy of the drive - i.e. if the external encoder does fail, then operation is possible using the motor encoder.

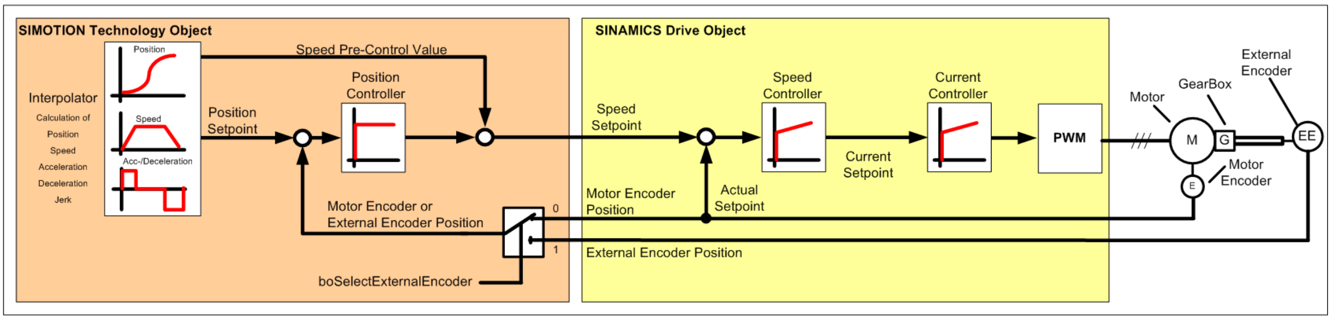


Figure 5-65 Switching over from a motor encoder to an external encoder

Note

Two data sets must be set up at the technology object.

1) **TO data set 1:** Position actual value from the motor encoder for the function modules Gantry, Hoist, Trolley and Boom.

2) **TO data set 2:** Position actual value from an external encoder for the function modules Gantry, Hoist, Trolley and Boom.

Using Gantry as an example

Sequence diagram

Various possibilities of changing over from a motor encoder to an external encoder and back for the Gantry function module are shown in the following sequence diagram. In this case, it should be noted that the power enable is not set in the sequence diagram (power = FALSE, STW1 bit 3) and the drive is stationary (boStandStill = TRUE, AppZSW1 bit 2).

Section 1:
In the first section of the sequence diagram, the input "boSelectExternalEncoder" (AppSTW2 bit 12) is set. This means that the technology object data set "actualdataset" is changed over from 1 to 2. After the switchover, the output "boExternalEncoderActive" (AppZSW2 bit 12) is set.

Section 2:
In the second section of the sequence diagram, the input "boSelectExternalEncoder" (AppSTW2 bit 12) is reset. This means that the technology object data set "actualdataset" is changed over from 2 to 1. After the switchover, the output "boExternalEncoderActive" (AppZSW2 bit 12) is reset.

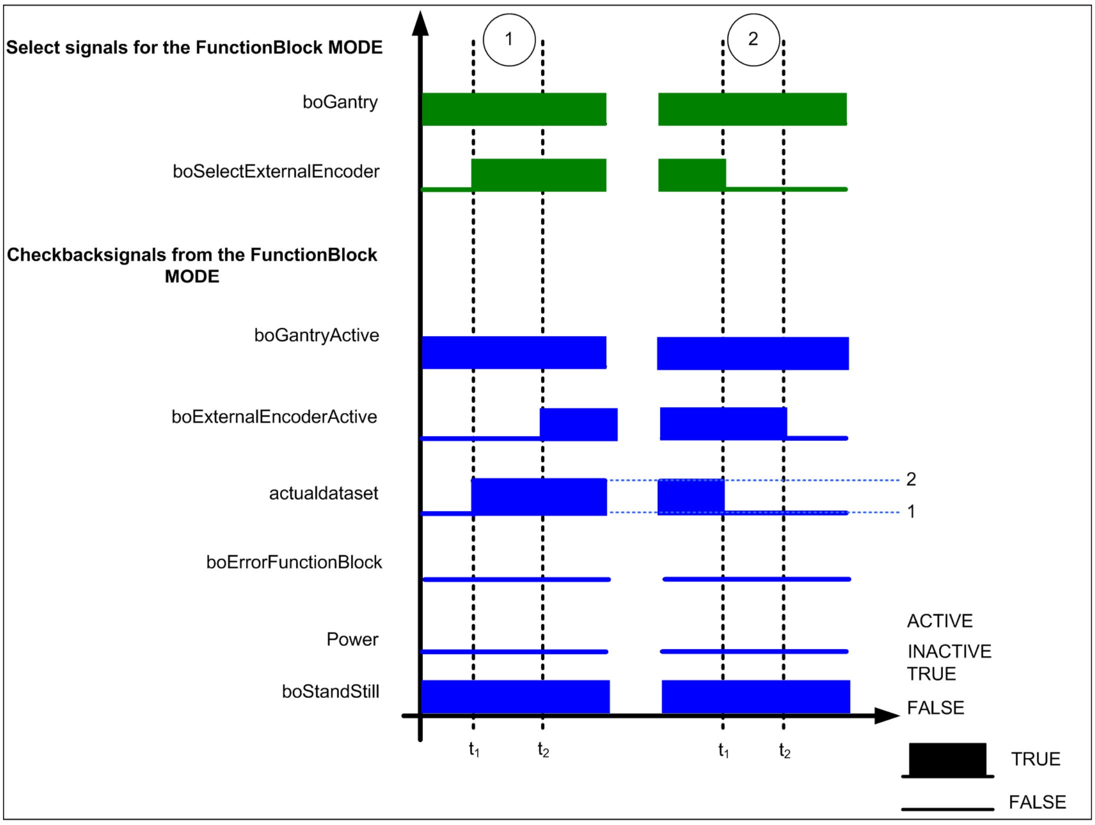


Figure 5-66 Changing over the position actual value from the motor encoder to an external encoder of a gantry.

5.3.9.10 Tandem mode

This chapter describes tandem mode with four drives.

Two spreaders are installed on an STS crane. Each spreader can pick up either one 40-foot or two 20-foot containers. Each spreader is driven by 2 motors and 2 converters, which serve to hoist or lower it. In tandem mode, the two spreaders work in velocity or position synchronous operation, depending on the selected operating mode.

The following rule applies to tandem mode:

- Hoisting-gear motor 1: Master 1 (tandem master)
- Hoisting-gear motor 2: Slave 1
- Hoisting-gear motor 3: Master 2 (tandem slave)
- Hoisting-gear motor 4: Slave 2

Master 1 and slave 1 can be connected to one another via a rigid connection (master-slave operation) or a flexible connection (synchronous operation). The same also applies to master 2 and slave 2. For information on how to select synchronous or master-slave operation for master 1 and slave 1 or master 2 and slave 2, refer to Master-slave operation control type (Page 253) and Synchronous operation control type (Page 257).

In tandem mode, master 1 and master 2 run in synchronous operation. The slaves follow their respective master. This ensures that both spreaders are hoisted and/or lowered at the same time.

The other interconnections and also the traversing of offset in tandem mode are described in more detail in Ship-to-shore tandem crane (STS tandem) (Page 470).

5.3.9.11 Offset mode

The offset mode functionality is comparable with that for the tandem mode. In tandem mode, an offset is applied between the tandem master and the tandem slave. Both of these drives act as masters. In the offset mode, however, an offset is applied between the master and the slave that run with a synchronous-mode coupling. Unlike tandem mode, only the slave can travel through an offset for offset mode.

For further information, refer to Chapter Offset mode between the master and the slave (Page 479).

5.3.9.12 Shortening the distance-to-go

With Gantry, the target position can only be shortened during positioning (so that the new target position is closer than the original one) in manual mode. It may be necessary to do this for collision protection (e.g. several cranes on the same rail). A check is then made to determine whether the maximum deceleration value is sufficient to ensure that the crane can be decelerated within the new, shorter distance-to-go. If it is sufficient, the necessary deceleration is defined automatically. However, if the maximum deceleration value is inadequate for decelerating the crane within the available time, an error is generated and the crane is decelerated immediately. The function does not have to be activated separately, as it becomes active automatically when a shortened distance-to-go is detected with Gantry.

In addition, you must enter the maximum deceleration under Axis → Limiting→ Dynamic behavior → Separation in mm/s^2.

You can also shorten the distance-to-go for all other drives. However, the new shorter target position is rejected and the original position is reinstated if the current deceleration is not sufficient to brake the crane in time. Unlike with Gantry, no checks are made and it is not possible to increase the deceleration value.

5.3.9.13 Brake test

The mechanical brake function (e.g. hoisting gear) should be regularly checked using this function. To do this, the axis moves against the closed brake with a certain torque setpoint in order to check the braking capability of the brake.

Note

The brake test is implemented in the standard "STS Crane" application; refer to STS crane (ship-to-shore) (Page 498).

The brake test sequence is as follows:

1. The SIMATIC S7 sends AppSTW2 bit 14 = True to SIMOTION. This selects the brake test.
2. SIMOTION selects a new drive data set (Drive Data Set, DDS) in SINAMICS. In this drive data set, the same motor data is saved as in DDS0 – or for converters that can be switched over – the same as in DDS2. The monitoring times must be set longer in order that SINAMICS does not output a fault (e.g. p2177 "motor blocked") or the current/frequency must be reduced.
3. The complete brake test is carried out in the SPEED_CONTROLLED operating mode.
4. With the feedback that the correct drive data set is active, SIMOTION returns the signal "brake test active" (AppZSW2 bit14).
5. The S7 then transfers the torque limit as a percentage – depending on SINAMICS parameter p2003 – from the S7 to SIMOTION via PZD8. SIMOTION sends this value to the master as upper and lower torque limit to SINAMICS.
6. The drive is then switched on and a velocity setpoint is entered. The torque limit is slowly increased by the S7.
7. After limiting by SINAMICS, the torque setpoint is sent to SIMOTION and SIMATIC S7 (e.g. PZD13). This means that the S7 can check as to whether the drive is within the torque limit.
8. The actual position and the actual velocity of the drives are also signaled to the S7. This allows the S7 to check whether the drive is moving. When performing the brake test, it is not permissible that the drive moves, as otherwise this would mean that the brake cannot hold the motor.
9. After the brake test, the brake test must be deselected in the PLC (AppSTW2 bit 14 = False). Depending on the selected operating mode, SIMOTION switches back to a valid drive data set and sets the torque limits back to 100%.

The following must be observed depending on the whether an internal or external TO changeover is active.

If no TO changeover is active (neither internally nor externally), the following applies:

The activation of the brake test causes a changeover to DDS6.

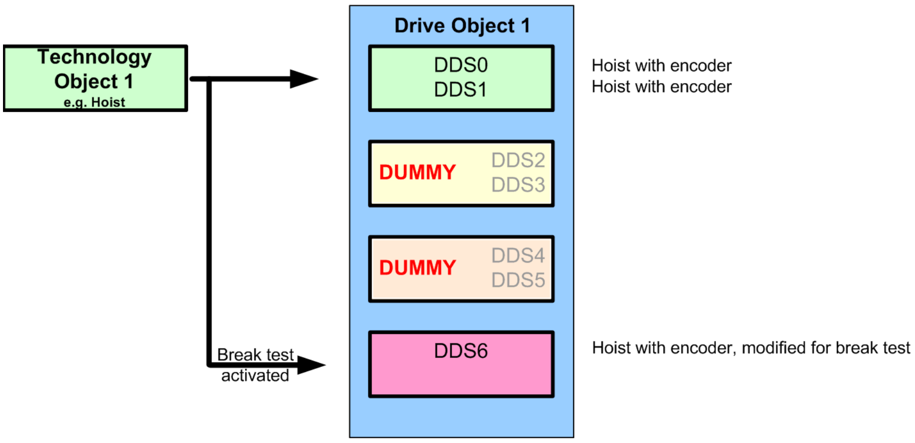


Figure 5-67 Brake test without changeover

If an internal changeover is active, the following applies:

The activation of the brake test causes a changeover to DDS6 or DDS7, depending on boChangeTechnologyObject.

- DDS6 → boChangeTechnologyObject = FALSE and boBreakTest = TRUE
- DDS7 → boChangeTechnologyObject = TRUE and boBreakTest = TRUE

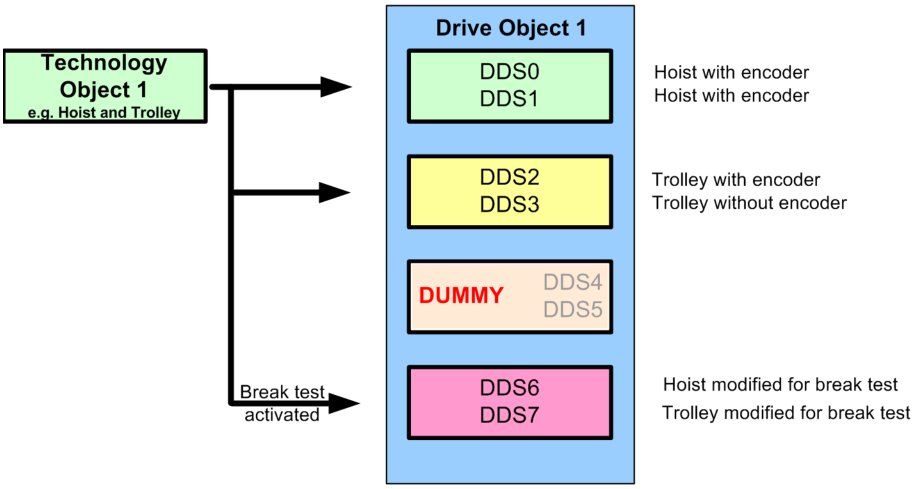


Figure 5-68 Brake test with internal changeover

If an external changeover is active, the following applies:

For the external changeover, three axes can be switched and work on a converter. However, because only two data sets are available for the brake test, you can also only perform a brake test on two technology objects. You can freely choose on which of the two technology objects you want to perform the brake test. To do this, you must not only also activate the brake test, but also activate the boBrakeTest_Drive_1 or boBrakeTest_Drive_2 input (boBrakeTest_Drive_1 for DDS6 and boBrakeTest_Drive2 for DDS7).

- DDS6 → boBrakeTest = TRUE and boBrakeTest_Drive_1 (for TO1, TO2 or TO3)
- DDS7 → boBrakeTest = TRUE and boBrakeTest_Drive_2 (for TO1, TO2 or TO3)

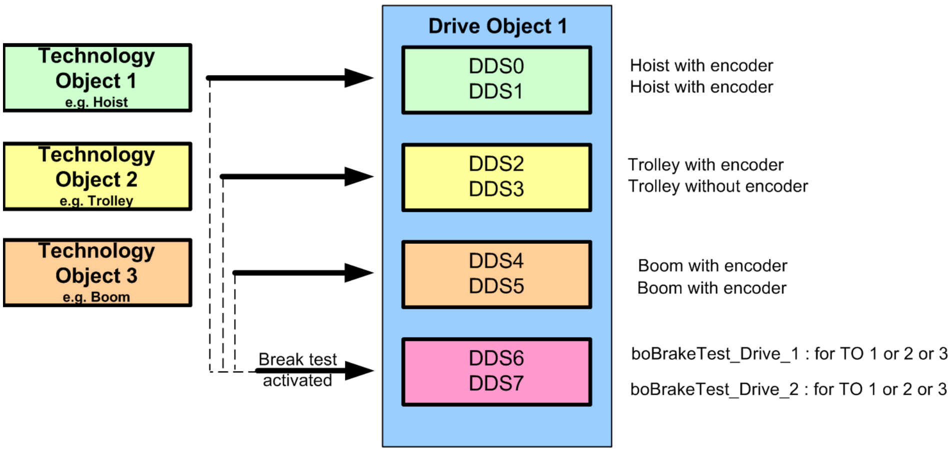


Figure 5-69 Brake test with external changeover

5.3.9.14 Brake control with SINAMICS

The brake control in SINAMICS can be used as of version V2.0 SP2 of the Basic Technology. To do this, in the particular drive when configuring the drive data set, select extendedbrake control. In this case, set the SINAMICS parameter P1215 either to [1] for the booksize format (motor holding brake just the same as the sequence control) or to [3] for the chassis format (motor holding brake just the same as the sequence control, connection via BICO). For the booksize format, the 24 V supply for the brake control is taken directly from terminal strip X11; for chassis units, an external relay must be controlled via the BICO interconnection.

Via the PLC, "open brake" is transferred to the SIMOTION (STW1 bit 9) - and is transferred on to the SINAMICS. This is the reason that after configuring, the communication script must also be called in the particular drive to ensure that the appropriate bits are interconnected. ZSW1 bit 12 is used as feedback signal regarding the brake state.

Note

Set the brake opening time in parameter p1216 to 0 ms, otherwise a fault occurs (the drive switches itself off). The brake opening time should be emulated in the PLC if it must be ruled out that the drive moves against the brakes. No move command must be issued to SIMOTION during the brake opening time.

In the OperationMode block, input "boEnableBrakeControlSinamics" must be set to TRUE if the extended brake control is to be used.

5.3.9.15 Alarms

The OperationMode function block also outputs alarms from version 2.0 SP2 and higher. These should provide users with information as to how they should change the control so that the required command / the required motion can be executed by SIMOTION, e.g. if enable signals are missing or an operating mode has not been selected.

If there is no fault and no alarm from a drive or from SIMOTION, then the function block alarms are sent to the PLC via PZD4, also see Chapter FB_ErrorPriority (Page 187) and Application error messages and alarm messages (Page 339).

5.3.9.16 Error

The function block OperationMode monitors errors in the technology object and the errors that can occur when starting a command. If an error occurs at the technology object, the block outputs the "boErrorTechnologyObject" signal and the corresponding error ID of technology object "i32ErrorIDTechnologyObject". If an error is caused by a command or incorrect operator action, the block outputs an "boErrorFunctionBlock" signal and the corresponding error ID of function block "i32ErrorIDFunctionBlock". As soon as an error occurs, the drive brakes with an active travel motion with the last specified, valid deceleration time. In the case of a fault, as soon as the drive is stationary, the S7 control must reset the power enable and remove the fault; also see Chapters FB_ErrorPriority (Page 187) and Application error messages and alarm messages (Page 339).

Sequence diagram

Two error cases are shown in the following sequence diagram

Section 1:
In the first section of the sequence diagram, an error at the function block is shown. The block outputs the "boErrorFunctionBlock" signal and the corresponding error ID "i32ErrorIDFunctionBlock". If the drive is moving, then the S7 control must stop it in a controlled fashion. Errors can only be acknowledged if the drive is stationary. In this case, the input "boAcknowledge" must be set for a short time using a positive edge. An error that is present is acknowledged, the output "boErrorFunctionBlock" and the error number "i32ErrorIDFunctionBlock" are reset.

Section 2:
In the second section of the sequence diagram, an error at the technology object is shown. The output "boErrorTechnologyObject" must be active and the associated error number "i32ErrorIDTechnologyObject" is output. If the drive is moving, it is stopped in a controlled fashion. The "S7" control should then reset the power enable. Errors can only be acknowledged if the drive is stationary. In this case, the input "boAcknowledge" must be set for a short time using a positive edge. An error that is present is acknowledged, the output "boErrorFunctionBlock" and the error number "i32ErrorIDTechnologyObject" are reset.

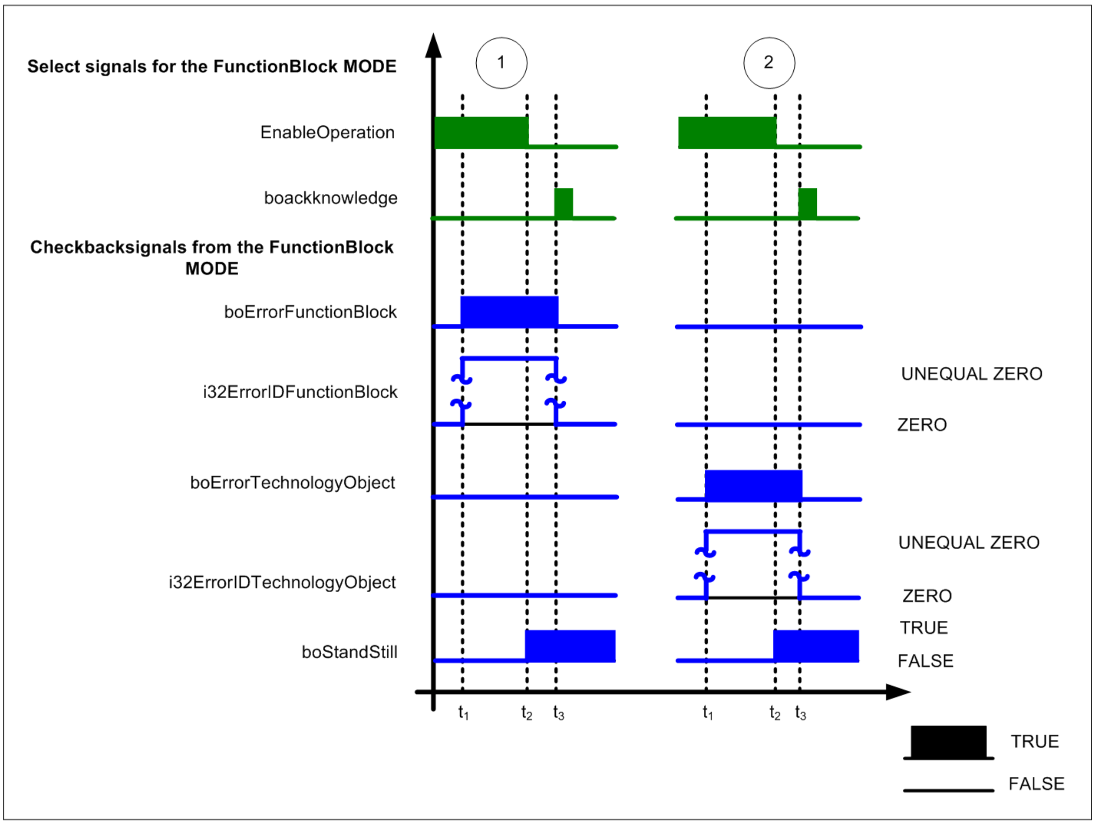


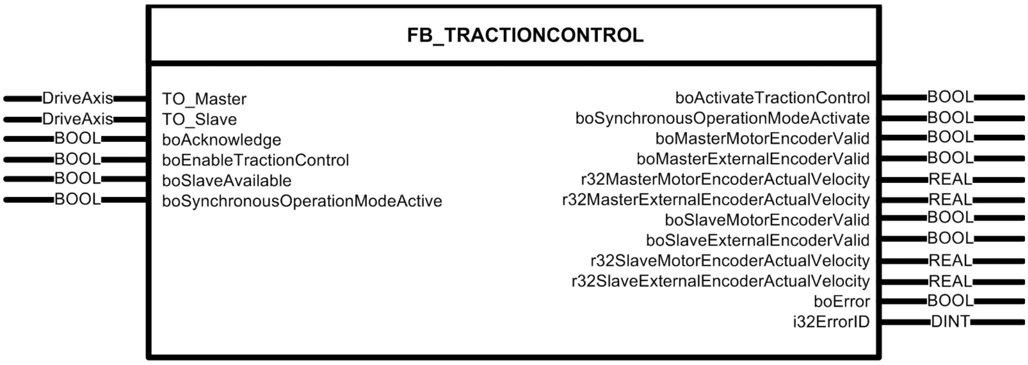
Figure 5-70 Error

5.3.10 FB_TractionControl

Task

The FB_TractionControl function block is used to read the current velocity of the motor encoder or that of the external encoder from the master/slave technology object by means of secure access; this information is then made available to the DCC_TractionControl function block.

Template for the call (FBD representation type)


FB_TRACTIONCONTROL
DriveAxis
DriveAxis
BOOL
BOOL
BOOL
BOOL
TO_Master
TO_Slave
boAcknowledge
boEnableTractionControl
boSlaveAvailable
boSynchronousOperationModeActive
boActivateTractionControl
boSynchronousOperationModeActivate
boMasterMotorEncoderValid
boMasterExternalEncoderValid
r32MasterMotorEncoderActualVelocity
r32MasterExternalEncoderActualVelocity
boSlaveMotorEncoderValid
boSlaveExternalEncoderValid
r32SlaveMotorEncoderActualVelocity
r32SlaveExternalEncoderActualVelocity
boError
i32ErrorID
BOOL
BOOL
BOOL
BOOL
REAL
REAL
BOOL
BOOL
REAL
REAL
BOOL
DINT

Connections

Name	Connection type	Default	Data type	Meaning
TO_Master	IN	-	DriveAxis	Enter the technology object name of the master (master object)
TO_Slave	IN	-	DriveAxis	Enter the technology object name of the slave (slave object)
boAcknowledge	IN	False	BOOL	Acknowledge error
boEnableTractionControl	IN	False	BOOL	Enables the traction control
boSlaveAvailable	IN	False	BOOL	Technology object slave present
boSychronousOperationModeActive	IN	False	BOOL	Synchronous operation between master and slave object active
boActivateTractionControl	OUT	False	BOOL	Activate traction control
boSynchronousOperationModeActivate	OUT	False	BOOL	Activates synchronous operation between the technology object master and slave
boMasterMotorEncoderValid	OUT	False	BOOL	Motor encoder of the master supplies valid values
boMasterExternalEncoderValid	OUT	False	BOOL	External encoder of the master supplies valid values
r32MasterMotorEncoderActualVelocity	OUT	0.0	REAL	Actual velocity motor encoder of the master [mm/s]

Name	Connection type	De-fault	Data type	Meaning
r32MasterExternalEncoderActualVelocity	OUT	0.0	REAL	Actual velocity external encoder of the master [mm/s]
boSlaveMotorEncoderValid	OUT	False	BOOL	Motor encoder of the slave supplies valid values
boSlaveExternalEncoderValid	OUT	False	BOOL	External encoder of the slave supplies valid values
r32SlaveMotorEncoderActualVelocity	OUT	0.0	REAL	Actual velocity motor encoder of the slave [mm/s]
r32SlaveExternalEncoderActualVelocity	OUT	0.0	REAL	Actual velocity external encoder of the slave [mm/s]
boError	OUT	False	BOOL	Error bit
i32ErrorID	OUT	0	DINT	Outputs the error number

Error messages

If an error occurs, output "boError" is set and an error number is returned at output "i32ErrorID" as for the other blocks; refer to Application error messages and alarm messages (Page 339).

5.3.11 FB_ReferenceMode

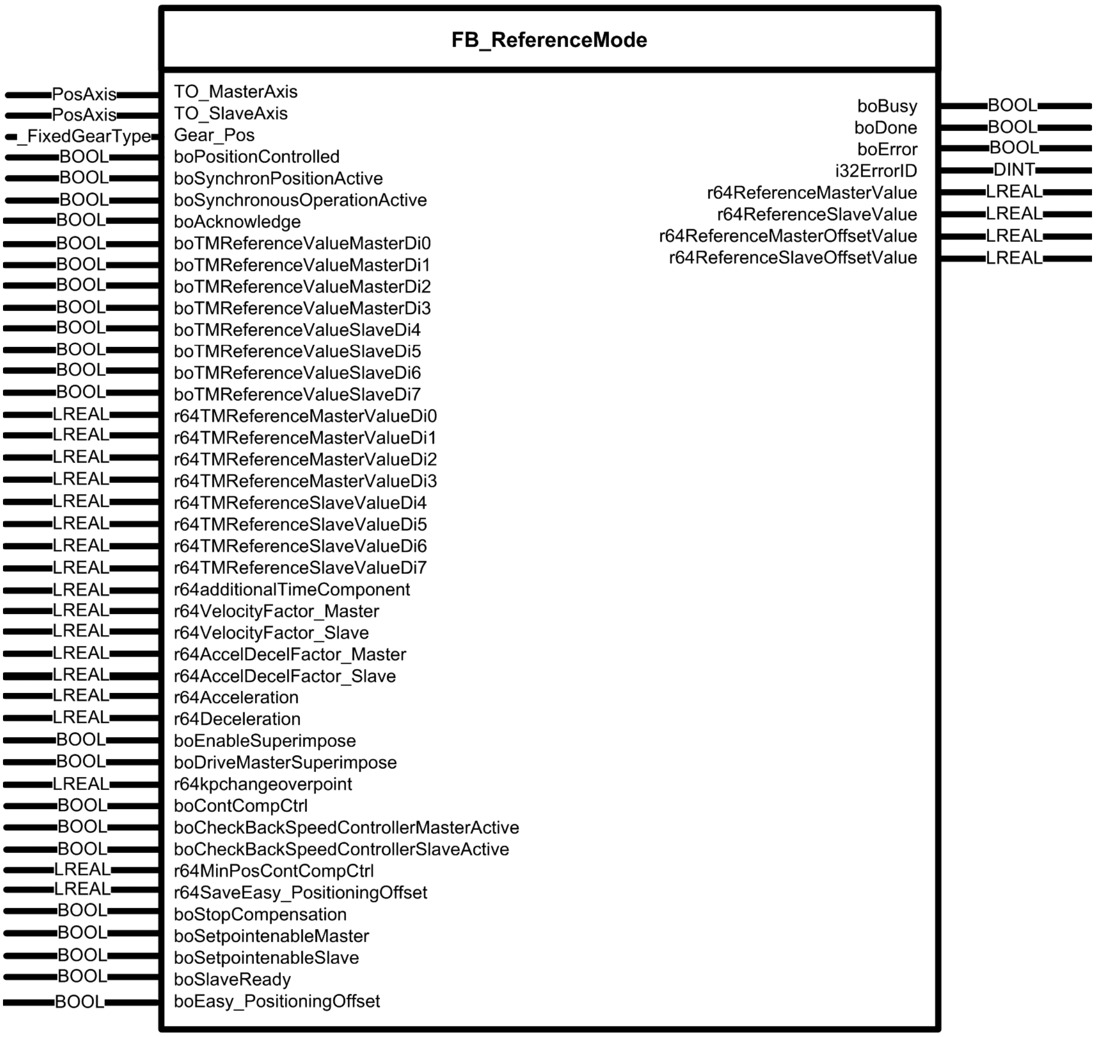
Task

The function block FB_ReferenceMode is used for flying homing and equalizing an offset between two drives that operate in a synchronous coupling (position or speed synchronism).

To home the axis, the digital inputs from this block must be connected to the signal source, e.g. the TM31. If one of the digital inputs of X520 or X530 (DI0 to DI7) on the TM31 is activated, for example by means of a BERO, then the corresponding axis is homed to the value that was stored for the relevant digital input. The positive edge of the BERO signal triggers homing.

On the other hand, for rail-bound drives such as gantry and trolley, in some instances wheels may slip, which will mean the crane no longer runs parallel to the rails. The function block FB_ReferenceMode also offers the option of equalizing this offset between the master (e.g. trolley 1) and the slave (e.g. trolley 2).

Template for the call (FBD representation type)


FB_ReferenceMode
PosAxis
PosAxis
_FixedGearType
BOOL
BOOL
BOOL
BOOL
BOOL
BOOL
BOOL
BOOL
BOOL
BOOL
BOOL
BOOL
LREAL
LREAL
LREAL
LREAL
LREAL
LREAL
LREAL
LREAL
LREAL
LREAL
LREAL
LREAL
LREAL
LREAL
LREAL
BOOL
BOOL
LREAL
BOOL
BOOL
BOOL
LREAL
LREAL
BOOL
BOOL
BOOL
BOOL
BOOL
TO_MasterAxis
TO_SlaveAxis
Gear_Pos
boPositionControlled
boSynchronPositionActive
boSynchronousOperationActive
boAcknowledge
boTMReferenceValueMasterDi0
boTMReferenceValueMasterDi1
boTMReferenceValueMasterDi2
boTMReferenceValueMasterDi3
boTMReferenceValueSlaveDi4
boTMReferenceValueSlaveDi5
boTMReferenceValueSlaveDi6
boTMReferenceValueSlaveDi7
r64TMReferenceMasterValueDi0
r64TMReferenceMasterValueDi1
r64TMReferenceMasterValueDi2
r64TMReferenceMasterValueDi3
r64TMReferenceSlaveValueDi4
r64TMReferenceSlaveValueDi5
r64TMReferenceSlaveValueDi6
r64TMReferenceSlaveValueDi7
r64additionalTimeComponent
r64VelocityFactor_Master
r64VelocityFactor_Slave
r64AccelDecelFactor_Master
r64AccelDecelFactor_Slave
r64Acceleration
r64Deceleration
boEnableSuperimpose
boDriveMasterSuperimpose
r64kpchangeoverpoint
boContCompCtrl
boCheckBackSpeedControllerMasterActive
boCheckBackSpeedControllerSlaveActive
r64MinPosContCompCtrl
r64SaveEasy_PositioningOffset
boStopCompensation
boSetpointenableMaster
boSetpointenableSlave
boSlaveReady
boEasy_PositioningOffset
boBusy
boDone
boError
i32ErrorID
r64ReferenceMasterValue
r64ReferenceSlaveValue
r64ReferenceMasterOffsetValue
r64ReferenceSlaveOffsetValue
BOOL
BOOL
BOOL
DINT
LREAL
LREAL
LREAL
LREAL

Connections

Name	Connection type	De-fault	Data type	Meaning
TO_MasterAxis	IN	-	PosAxis	Enter the technology object name of the master (master object)
TO_SlaveAxis	IN	-	PosAxis	Enter the technology object name of the slave (slave object)
Gear_Pos	IN	-	_FixedGear Type	Enter the name of the fixed gear object position
boPositioncontrolled	IN	False	BOOL	Closed-loop position controlled operation active
boSynchronPositionActive	IN	False	BOOL	Synchronous positioning active at slave

Name	Connection type	De-fault	Data type	Meaning
boSynchronousOperationActive	IN	False	BOOL	Synchronous operation selected for the slave
boAcknowledge	IN	False	BOOL	Acknowledge error
boTMReferenceValueMasterDi0	IN	False	BOOL	Signal input for BERO 1 (for the master)
boTMReferenceValueMasterDi1	IN	False	BOOL	Signal input for BERO 2 (for the master)
boTMReferenceValueMasterDi2	IN	False	BOOL	Signal input for BERO 3 (for the master)
boTMReferenceValueMasterDi3	IN	False	BOOL	Signal input for BERO 4 (for the master)
boTMReferenceValueSlaveDi4	IN	False	BOOL	Signal input for BERO 1 (for the slave)
boTMReferenceValueSlaveDi5	IN	False	BOOL	Signal input for BERO 2 (for the slave)
boTMReferenceValueSlaveDi6	IN	False	BOOL	Signal input for BERO 3 (for the slave)
boTMReferenceValueSlaveDi7	IN	False	BOOL	Signal input for BERO 4 (for the slave)
r64TMReferenceMasterValueDi0	IN	0.0	LREAL	Reference value of digital input 0 (1st position for the master)
r64TMReferenceMasterValueDi1	IN	0.0	LREAL	Reference value of digital input 1 (2nd position for the master)
r64TMReferenceMasterValueDi2	IN	0.0	LREAL	Reference value of digital input 2 (3rd position for the master)
r64TMReferenceMasterValueDi3	IN	0.0	LREAL	Reference value of digital input 3 (4th position for the master)
r64TMReferenceSlaveValueDi4	IN	0.0	LREAL	Reference value of digital input 4 (1st position for the slave)
r64TMReferenceSlaveValueDi5	IN	0.0	LREAL	Reference value of digital input 5 (2nd position for the slave)
r64TMReferenceSlaveValueDi6	IN	0.0	LREAL	Reference value of digital input 6 (3rd position for the slave)
r64TMReferenceSlaveValueDi7	IN	0.0	LREAL	Reference value of digital input 7 (4th position for the slave)
r64additionalTimeComponent	IN	0.0	LREAL	Specification of additional times [ms] such as periphery run times
r64VelocityFactor_Master	IN	1.0	LREAL	Evaluation factor [%] based on the master base velocity; this specifies the additional master compensatory velocity (max. 20%)
r64VelocityFactor_Slave	IN	1.0	LREAL	Evaluation factor [%] based on the master base velocity; this specifies the additional slave compensatory velocity (max. 20%)
r64AccelDecelFactor_Master	IN	1.0	LREAL	Evaluation factor [%] referred to the master acceleration/deceleration as an acceleration value for the slave compensatory velocity (max. 50%)

Name	Connection type	De-fault	Data type	Meaning
r64AccelDecelFactor_Slave	IN	1.0	LREAL	Evaluation factor [%] referred to the master basis acceleration/deceleration as an acceleration value for the master compensatory velocity (max. 50%)
r64Acceleration	IN	1.0	LREAL	Acceleration value of the master [mm/s²]
r64Deceleration	IN	1.0	LREAL	Deceleration value of the master [mm/s²]
boEnableSuperimpose	IN	False	BOOL	Enabling/stopping compensatory control
boDriveMasterSuperimpose	IN	False	BOOL	Generally release compensatory control for the master
r64kpchangeoverpoint	IN	0.5	LREAL	Gain factor path control
boContCompCtrl	IN	False	BOOL	Enable continuous compensatory control
boCheckBackSpeedControllerMasterActive	IN	False	BOOL	Speed controller feedback signal from the master active
boCheckBackSpeedControllerSlaveActive	IN	False	BOOL	Speed controller feedback signal from the slave active
r64MinPosContCompCtrl	IN	5.0	LREAL	Minimum offset [mm] between the master and slave, before compensation is started (only valid for continuous equalization boContCompCtrl =TRUE)
r64SaveEasy_PositioningOffset	IN	0.0	LREAL	Saving the offset in the EASY_POSITIONING operating mode
boStopCompensation	IN	False	BOOL	If a superimposed motion is active in the Block operation mode, this block does not start the superimposed motion.
boSetpointenableMaster	IN	False	BOOL	Setpoint enable at the master active
boSetpointenableSlave	IN	False	BOOL	Setpoint enable at the slave active
boSlaveReady	IN	False	BOOL	Slave is ready.
boEasy_PositioningOffset	IN	FALSE	BOOL	The current offset is considered for the superimposed motion.
boBusy	OUT	False	BOOL	Function block status
boDone	OUT	False	BOOL	Function block status
boError	OUT	False	BOOL	Function block status
i32ErrorID	OUT	0	DINT	Error code
r64ReferenceMasterValue	OUT	0.0	LREAL	Total correction value for the master [mm]
r64ReferenceSlaveValue	OUT	0.0	LREAL	Total correction value for the slave [mm]

Name	Connection type	De-fault	Data type	Meaning
r64ReferenceMasterOffsetValue	OUT	0.0	LREAL	Total correction offset for the master [mm] in position synchronous operation
r64ReferenceSlaveOffsetValue	OUT	0.0	LREAL	Total correction offset for the slave [mm] in position synchronous operation

Functionality

The block has eight digital inputs, four for the master and four for the slave, and the matching (eight) inputs for the position values. If a slave drive has not been created, then the eight inputs are automatically available for the master drive.

The block must be called in the IPO synchronous task.

Input r64additionalTimeComponent can be used to apply additional run times relevant to the position correction. The interpolator cycle clock (IPO) in the SIMOTION system is automatically included in the position correction and need not be added to the value applied via r64additionalTimeComponent.

Compensatory control is activated in synchronous operation by means of a high signal level at "boEnableSuperimpose" and is deactivated with a low signal level. There are two options for compensating for an offset between the drives, which can be selected at the input "boContCompCtrl":

1) "boContCompCtrl" is FALSE: Compensatory control, only if the conditions are fulfilled

Compensation of an offset between the master and slave is only started if a BERO / reference switch signal was signaled for both drives. The sequence is irrelevant. For example, if a wheel slips at a drive (e.g. Gantry_2), then its current position in the controller will be greater than the (real) actual position of the wheel and the crane will run "skewed".

If the first drive (e.g. Gantry_1) travels over the BERO, its current position will be set in the controller. Since the wheel has slipped, the second drive will reach the BERO later than the first and its current position will be reset too. As the first drive (gantry_1) has continued to move too, the two current positions no longer match, i.e. there is an offset. Offset compensation now starts after both drives have a feedback signal from the BERO. The compensation is either performed by the master drive or the slave drive. If "boDriveMasterSuperimpose" = FALSE, then only the slave is used for compensation. If "boDriveMasterSuperimpose" = TRUE, compensation is performed, depending on an internal calculation, by the master or slave.

To compensate for the position difference, for the drive that should compensate the offset, an additional velocity is applied until the two positions are identical. Following compensation, the superimposed motion is stopped and it cannot be started again until the two BEROs (master and slave) have responded again.

The internal calculation for "boDriveMasterSuperimpose" = TRUE checks the maximum and the actual velocity and transfers the supplementary velocity to the slave, if this is not at the velocity limit. However, if the slave is at the limit, the supplementary velocity (with the inverse sign) is transferred to the master.

The master now travels more slowly and allows the slave drive to catch up. The flying change is realized automatically, without requiring any external intervention. As a consequence, the crane always travels with the maximum velocity and the master now only brakes, if the slave cannot catch up to it. However, if "boDriveMasterSuperimpose" = FALSE, the slave cannot compensate the offset if it travels behind the master at the maximum velocity. The slave can only move faster in order to compensate an offset in the acceleration/deceleration phase - or if the crane is not traveling at the maximum velocity. However, if the slave is leading, using the supplementary velocity, it can travel slower at any time so that the master can catch up with the slave.

2) "boContCompCtrl" is TRUE: Compensatory control runs continuously

The offset is compensated as soon as a position difference occurs. Unlike with the first option, both BERO signals are not required here in order to start compensatory control. An offset can also arise if a drive reaches its target velocity with a slight delay due to a load which is not distributed symmetrically (Trolley is fully loaded on one side). Compensatory motion is started as soon as it is identified that the positions of the two drives differ from one another. As described above, generally the slave is used to remove the offset. The supplementary velocity is only transferred to the master if the slave cannot reach the maximum velocity; this supplementary velocity then slows down the master. As a consequence, "boDriveMasterSuperimpose" must also be activated for this version, as otherwise, only the slave compensates the offset. The minimum offset can be set using "r64MinPosContCompCtrl" before a compensatory control is started. This function is only available when "boContCompCtrl" is active, thus ensuring that the controller does not have to intervene for every millimeter of offset.

General

The supplementary velocity for the master can be set at the "r64VelocityFactor_Master" input and for the slave at the "r64VelocityFactor_Slave" input. Compensatory motion can be used in the auto mode, manual mode and speed controlled mode.

In the SPEED_CONTROLLED operating mode compensation is achieved using path control. The "r64kpchangeoverpoint" gain factor should be set only for this mode during the commissioning. The "r64kpchangeoverpoint" calculates the instant that deceleration is started. If the value is set too low, deceleration is initiated too early and the deceleration phase takes a lot of time. However, the target is not overshot. If the value is set too high, deceleration is initiated too late and the deceleration ramp is extremely steep. The motor cannot follow this ramp and overshoots the target position; further, under certain circumstances it can also oscillate around the target point.

"r64kpchangeoverpoint" must therefore be increased until the deceleration of the supplementary velocity is approximately the same magnitude or somewhat lower than the acceleration of the supplementary velocity. This means that the supplementary velocity must have an approximate trapezoidal characteristic.

Chapter Setting the closing velocity (Page 458) provides a more detailed explanation of "r64kpchangeoverpoint".

Note

If "r64kpchangeoverpoint" is selected to be too high, then the drive overshoots the target position; this is because the calculated deceleration is higher than the deceleration of the drive that is actually possible.

Note

During acceleration, the slave can always travel faster than the master. This is because the final velocity has still not been reached. If the slave must accelerate/travel faster in order to compensate the offset, then it will do this up to the maximum velocity. Shortly before the maximum velocity is reached, the velocity of the master is reduced and the slave continues to travel normally (with maximum velocity) if "boDriveMasterSuperimpose" is activated. This means that the supplementary velocity is transferred to the master - so that it travels slower - before the slave reaches its maximum velocity.

Depending on the internal calculation, either both compensatory velocities are changed together, or the slave waits until the master has reached its target velocity and only then is the slave compensatory velocity adapted. This behavior is normal and does not have to be investigated any further as the compensatory velocity isn't noticed in operation.

Note

Alarm 3260 or 3270 can only occur in closed-loop position controlled operation if, after updating the position, the new offset is greater than that permitted for the synchronous coupling. As soon as the slave drive comes to a standstill, the synchronous coupling is automatically withdrawn and cannot be re-established (recoupled) until the offset is less than the maximum permitted value (r64toleranceMasterSlavePosition). This means that "r64toleranceMasterSlavePosition" must be set to the maximum possible offset. This variable is in the FB_OperationMode and only has to be set for the slave drive.

Note

The version with "boContCompCtrl" = FALSE is the preferred version for the compensatory control, as only one compensatory motion takes place if both BEROs (from the master and slave) have feedback signals. As a consequence, before compensation it is ensured that both positions have been updated. For the version with "boContCompCtrl" = TRUE, compensatory motion always takes place, even if only one of the drives has a signal from the BERO. The position from the other drive could also have a deviation, and under certain circumstances, the offset could even be higher. As a consequence, this version should only be selected, if the positions are additionally monitored using external encoders (wheel-driven).

Note

Although flying homing can always be used, the compensatory control should only be used if at both sides (master and slave sides) the positions are also updated from a BERO momentary-contact switch / switch or corrected from a wheel-driven encoder.

Error messages

If an error occurs, output "boError" is set and an error number is returned at output "i32ErrorID" as for the other blocks; refer to Application error messages and alarm messages (Page 339).

5.3.12 FB_Cornering

Task

The function block FB_Cornering is used to perform a cornering movement during speed-controlled operation. There follows a description of the function of a cornering movement, using a ship to shore crane (STS) as an example.

In most cases, the wall of the quay at a shipping pier is straight and the crane rails run parallel with it, which means the crane can be operated in a standard manner. But if a section of the rails is not laid straight and follows a curved path, the crane's drives have to follow this curve too. To make this work, either the wheel on the outside of the curve has to be driven faster, or that on the inside of the curve slower, than the other wheel. Below, the cornering movement is achieved by reducing the velocity of the wheel on the inside of the curve. The factor to reduce the wheel on the inside of the curve is determined by the ratio of the length of the rail on the outside of the curve to that of the rail on the inside of the curve. Alternatively, the ratio of the rail curve radii can also be used. The cornering movement is started by means of a sensor (BERO) at each end of the evenly curved section of track.

Note

Cornering is implemented in the standard "STS Crane" application; refer to STS crane (ship-to-shore) (Page 498).

When commissioning, the "boEnableCornering" input should be used to define as to whether the block should be active or not.

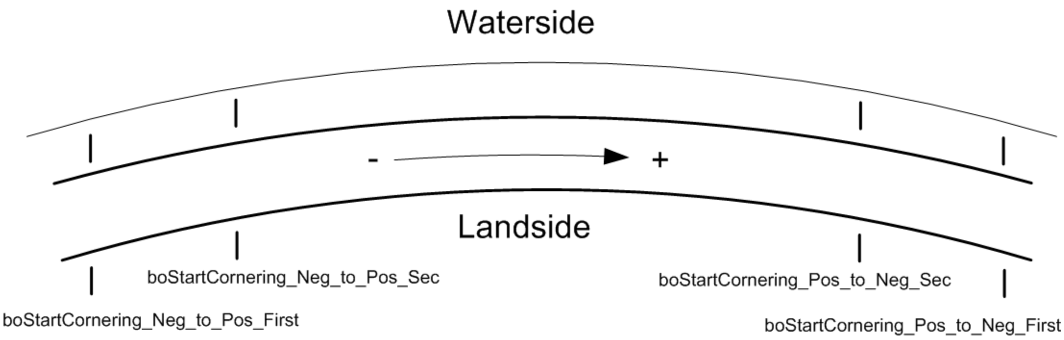

Waterside
-
+
Landside
boStartCornering_Neg_to_Pos_Sec
boStartCornering_Pos_to_Neg_Sec
boStartCornering_Neg_to_Pos_First
boStartCornering_Pos_to_Neg_First

Figure 5-71 Cornering movement

Note

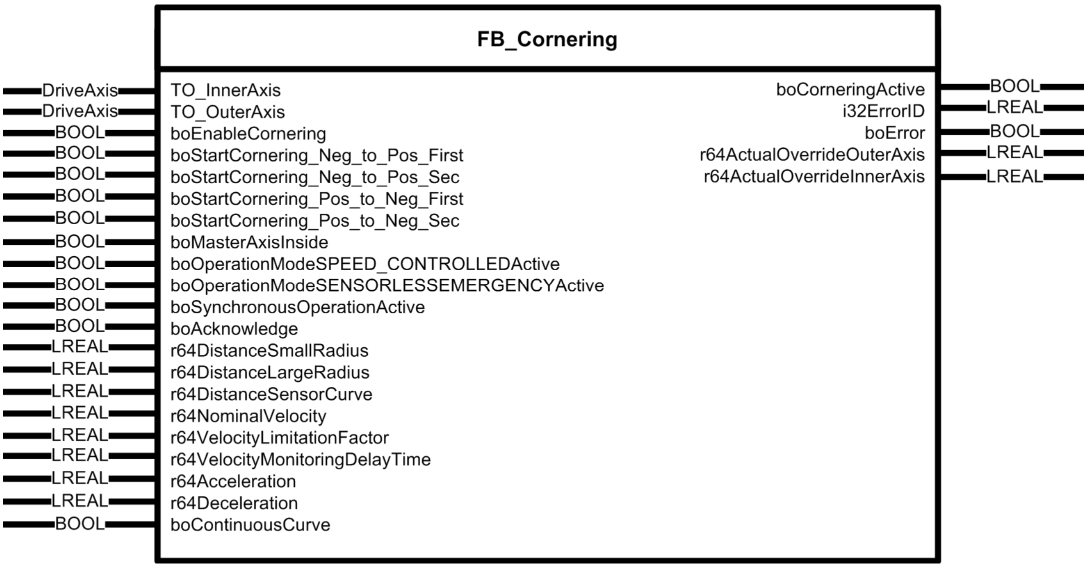
In a standard case, it is assumed that the longer rail of the curved section of track is on the side nearest to the water, and that the curve radius is constant. Furthermore, the MASTER must be on the side nearest to the water (outer edge of the curve) and the SLAVE on the side nearest to the land (inner edge of the curve).

If these conditions are not met, i.e. the SLAVE is on the outer edge of the curve and the MASTER on the inner edge, bit "boMasterAxisInside" must be set in order for the velocity ratio to be calculated correctly.

NOTICE

Block FB_Cornering must be used properly to ensure that, when a cornering movement is required, the inside wheels turn slower than the outside wheels in accordance with the curve radius; if corresponding settings are not made, the gantry may tilt or even derail.

Template for the call (FBD representation type)


FB_Cornering
DriveAxis TO_InnerAxis
DriveAxis TO_OuterAxis
BOOL boEnableCornering
BOOL boStartCornering_Neg_to_Pos_First
BOOL boStartCornering_Neg_to_Pos_Sec
BOOL boStartCornering_Pos_to_Neg_First
BOOL boStartCornering_Pos_to_Neg_Sec
BOOL boMasterAxisInside
BOOL boOperationModeSPEED_CONTROLLEDActive
BOOL boOperationModeSENSORLESSEMERGENCYActive
BOOL boSynchronousOperationActive
BOOL boAcknowledge
LREAL r64DistanceSmallRadius
LREAL r64DistanceLargeRadius
LREAL r64DistanceSensorCurve
LREAL r64NominalVelocity
LREAL r64VelocityLimitationFactor
LREAL r64VelocityMonitoringDelayTime
LREAL r64Acceleration
LREAL r64Deceleration
BOOL boContinuousCurve
boCorneringActive BOOL
i32ErrorID LREAL
boError BOOL
r64ActualOverrideOuterAxis LREAL
r64ActualOverrideInnerAxis LREAL

Connections

Name	Connection type	Default setting	Data type	Meaning
TO_InnerAxis	IN	-	DriveAxis	Technology object name of the master (master object)
TO_OuterAxis	IN	-	DriveAxis	Technology object name of the slave (slave object)
boEnableCornering	IN	False	BOOL	Activates the block

Name	Connection type	Default setting	Data type	Meaning
boStartCornering_Neg_to_Pos_First	IN	False	BOOL	Sensor shortly before curve for moving in positive direction
boStartCornering_Neg_to_Pos_Sec	IN	False	BOOL	Alternative sensor at start of curve for moving in positive direction
boStartCornering_Pos_to_Neg_First	IN	False	BOOL	Sensor shortly before curve for moving in negative direction
boStartCornering_Pos_to_Neg_Sec	IN	False	BOOL	Alternative sensor at start of curve for moving in negative direction
boMasterAxisInside	IN	False	BOOL	Master axis (master object) on the inside of the curve
boOperationModeSPEED_CONTROLLEDActive	IN	False	BOOL	SPEED_CONTROLLED operating mode active
boOperationModeSENSORLESSEMERGENCYActive	IN	False	BOOL	SENSORLESS EMERGENCY operating mode active
boSynchronousOperationActive	IN	False	BOOL	Synchronous operation active
boAcknowledge	IN	False	BOOL	Acknowledge fault
r64DistanceSmallRadius	IN	0.0	LREAL	Length of the inner track
r64DistanceLargeRadius	IN	0.0	LREAL	Length of the outer track
r64DistanceSensorCurve	IN	0.0	LREAL	Distance of the sensors from the curved section of track
r64NominalVelocity	IN	0.0	LREAL	Nominal velocity of the gantry
r64VelocityLimitationFactor	IN	0.0	LREAL	Limitation of the maximum velocity along the curve
r64VelocityMonitoringDelayTime	IN	0.0	LREAL	Delay time of the velocity monitoring function
r64Acceleration	IN	1.0	LREAL	Rated acceleration of function module
r64Deceleration	IN	1.0	LREAL	Rated deceleration of function module
boContinuousCurve	IN	False	BOOL	Total length of curved path
boCorneringActive	OUT	False	BOOL	Cornering movement active
i32ErrorID	OUT	0	DINT	Outputs the error number
boError	OUT	False	BOOL	Error bit
r64ActualOverrideOuterAxis	OUT	100.0	LREAL	Actual FB value calculated for velocity override
r64ActualOverrideInnerAxis	OUT	100.0	LREAL	Actual FB value calculated for velocity override

Functionality

Inputs "boOperationModeSPEED_CONTROLLEDActive", "boOperationModeSENSORLESSEMERGENCYActive", and "boSynchronousOperationActive" must be connected with the outputs of the corresponding FB_OperationMode.

The block must be called in the IPO synchronous task.

If the block is deactivated, the technology objects' override values for velocity and acceleration do not change. The following conditions always apply for cornering movements:

- Sensors (boStartCornering_Neg_to_Pos_First and boStartCornering_Pos_to_Neg_First) have to be installed before the curve and after the curve to trigger deceleration to cornering velocity.
- Once there are no more sensors, after the "r64DistanceSensorCurve" distance has been completed, cornering is started by means of additional deceleration of the inside wheels.
- If appropriate sensors have been installed at the start and the end of the curve, these will ensure that cornering starts once the brakes have been applied. However, the "r64DistanceSensorCurve" parameter must be set correctly if the brakes are to be applied correctly.
- The Motor Modules for the right and left drives must work separately from one another, i.e. the drives must not be cross-operated.
- The difference between the lengths of the two rails must be known.
- The operating mode must be speed-controlled, i.e. SPEED_CONTROLLED or SENSORLESSEMERGENCY - and the drives must run in synchronous operation.
- If the cornering movement is triggered by the sensor, the velocity and the maximum accelerations of the crane will be reduced to a percentage (r64VelocityLimitationFactor) of the nominal velocity in order to prevent derailing.
- The cornering movement is achieved by reducing the velocity of the wheel on the inside of the curve. The rate of reduction depends on the difference between the lengths of the rails.
- Cornering ends when the crane passes a sensor. This can be one of the inside or outside sensors. The additional deceleration of the inside wheels is also lifted at this point.
- Next, the reduction of the travel speed applied to all drives is also lifted.
- If the crane is situated on a continuous curve rail with constant radius, the block can be set to continuous cornering via the "boContinuousCurve" input. In this case, the "boStartCornering_…" inputs become irrelevant.

If the crane enters the curve in an incorrect operating mode (position-controlled and positive edge at the corresponding sensor), an error message is output.

During the cornering movement the block monitors the velocity of the gantry and outputs an error message if the maximum permissible curve velocity is exceeded. The dynamic responses are also monitored during cornering.

Note

When cornering is active, the offset compensatory controller for synchronous operation, which is located in FB_ReferenceMode, has to be deactivated.

Torque compensatory controller

In addition to FB_Cornering, compensatory control can also be activated for the individual drives of the gantry. This control ensures that the same driving torque is active on all motors and serves to optimize the cornering movement.

The torque compensatory controller is activated using the current compensatory controller block DCC_CurrentEqualControl from the Crane_DCC_Library. The block is integrated in the DCC chart of the slave drive and must be connected as described below. In the following example, Gantry 2 is selected as the slave drive.

- "rCurrentHoldingGear" is connected with "Gantry_1_Setpoint_Current_for_DCC"
- "rCurrentClosingGear" is connected with "Gantry_2_Setpoint_Current_for_DCC"
- "rCurrentNominal" is connected with "Gantry_2_NominalCurrent"
- "rMaximumVelocity" is connected with "Gantry_2_MaxVelocity"
- "rOutVelocitySetpoint" is connected with "Gantry_2_AdditionalVelocitySetpoint"
- The output variable of DCC chart "Gantry_2_AdditionalVelocitySetpoint" must be connected with input "r64AdditionalSpeedSetpoint" of FB_TelegramSimotionToSinamics by means of the REAL_to_LREAL() function.

The adjustable parameters for the current compensatory controller being used must be adapted to the conditions of the actual cornering movement in question.

If the torque has to be distributed unevenly, the "orange-peel bucket" operating mode of block DCC_CurrentEqualControl can be used to fine-tune the current compensatory controller.

Note

For more detailed information on settings for the current compensatory controller, refer to the section titled DCC_CurrentEqualControl (Page 95).

Error messages

If an error occurs, output "boError" is set and an error number is issued at output "i32ErrorID"; refer to the section titled Application error messages and alarm messages (Page 339).

5.3.13 FB_SynchronizeRTC

Task

For error diagnostics, many applications require a consistent setting of the time of day in SIMOTION and SINAMICS by the SIMATIC S7 controller. If SIMOTION is de-energized over a longer time period, SIMOTION and SINAMICS lose their set times. The higher-level PLC controller usually has a better emergency power supply. To permit consistent timestamps in the system, the time of day from the higher-level PLC controller should be checked and, if necessary, reset.

The "FB_SynchronizeRTC" function block in SIMOTION processes the higher-level PLC controller request. The function block contains specifications of the higher-level PLC controller (date, time, control word for set time). The date and time are always sent by the higher-level PLC controller. A control word in the user-engineering area is used to set the time of day in SIMOTION. The control word initiates the setting of the time. After the setting command, the current time in the SIMOTION is fetched automatically and passed to the higher-level PLC controller. The "Set time" function is executed in the user-engineering area of the communication between the SIMATIC S7 higher-level PLC controller and SIMOTION.

Functionality

SIMATIC S7 to SIMOTION communication:

- One double word as UDINT (u32ClockTimeS7) address PID 1904
- One word (b16daysS7) address PIW 1908
- One word (b16timestw) address PIW 1910

SIMOTION to SIMATIC S7 communication:

- One double word as UDINT (u32ClockTimeSimotion) address PQD 1904
- One word (b16daysSimotion) address PQW 1908

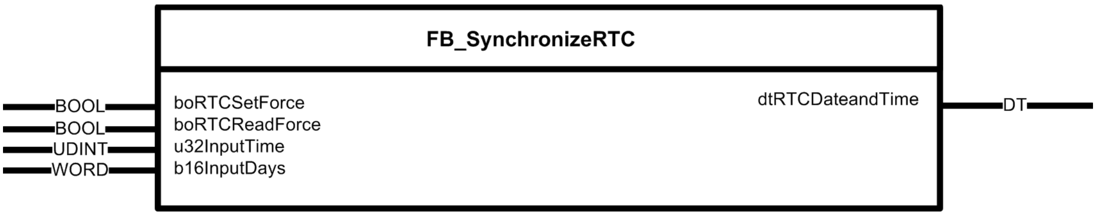
The following additional blocks and functions for the automatic setting are required in the SIMATIC:

- SFC1
- FC1533
- FC1536
- FC1538
- FC1564
- FC1740
- DB4.

They are provided in every standard application and included in the SIMATIC part of the project. The call is made via the OB1.

A positive edge at "boRTCSetForce" transfers the appropriate time and date from the I/O container. The date and time are then fetched automatically. These values are sent to the SIMATIC. The comparison function implemented in FC1740 ensures that the timestamp between SIMATIC and SIMOTION is identical except for the propagation-delay difference. The timestamp has been set to 1 second and must not be changed because the bus runtime and execution time of the OB1 or FC1740 must be considered.

Template for the call (FBD representation type)



Connections

Name	Connection type	Default setting	Data type	Meaning
boRTCSetForce	IN	False	Bool	Setting the time-of-day
boRTCReadForce	IN	False	Bool	Fetching the time-of-day
u32InputTime	IN	0	UDINT	Time specification
b16InputDays	IN	16#0	WORD	Day specification
dtRTCDateandTime	OUT	DT#0001-01-01-0:0:0	DT	Output of the date and time

5.3.14 FB_AutoSettingFW

Task

This FB block simplifies the commissioning of the two DCC_LoadDependFieldweak_1 (LDFW) and DCC_ContLoadMeasurement_1 (CLM) DCC blocks. The use of the FB block is optional. The detailed use is described in Chapters Field weakening (LDFW) (Page 414) and Continuous load measurement (CLM) (Page 434).

When this block is used, the required physical quantities of the system (such as the friction and acceleration compensation) are determined automatically. They must then be transferred to the associated LDFW and/or CLM inputs.

Functions

When this block is used, the rope can be rolled up and the container spreader / grab fully mounted. No measurements need to be performed beforehand.

The commissioning is performed in two steps. In the first step, test runs are performed without load but with container spreader / grab. These are pure test runs that the commissioning engineer must continue to perform until the function block completes the measurement successfully.

In the second step, test runs are performed with the rated load. Also in this case, the commissioning engineer must continue to perform the test runs until the function block completes the measurement successful.

Finally, the values determined at the output must be sent by the commissioning engineer as input values to the LDFW and/or CLM.

If a value for r64minLoad and r64maxLoad is already entered at the input, the parameters for the curve calculation in the LDFW block are also determined at the output ("r64X1ParameterLoad" to "r64X6ParameterLoad" or "r64Y1ParameterVelocity" to "r64Y6ParameterVelocity"). They must also be transferred to the LDFW.

"R64MinLoad" and "r64MaxLoad" can be specified as percentage or physical unit, such as kilograms or tons, depending on how "rLoadNormFactor" has been set for the LDFW.

Template for the call (FBD representation type)

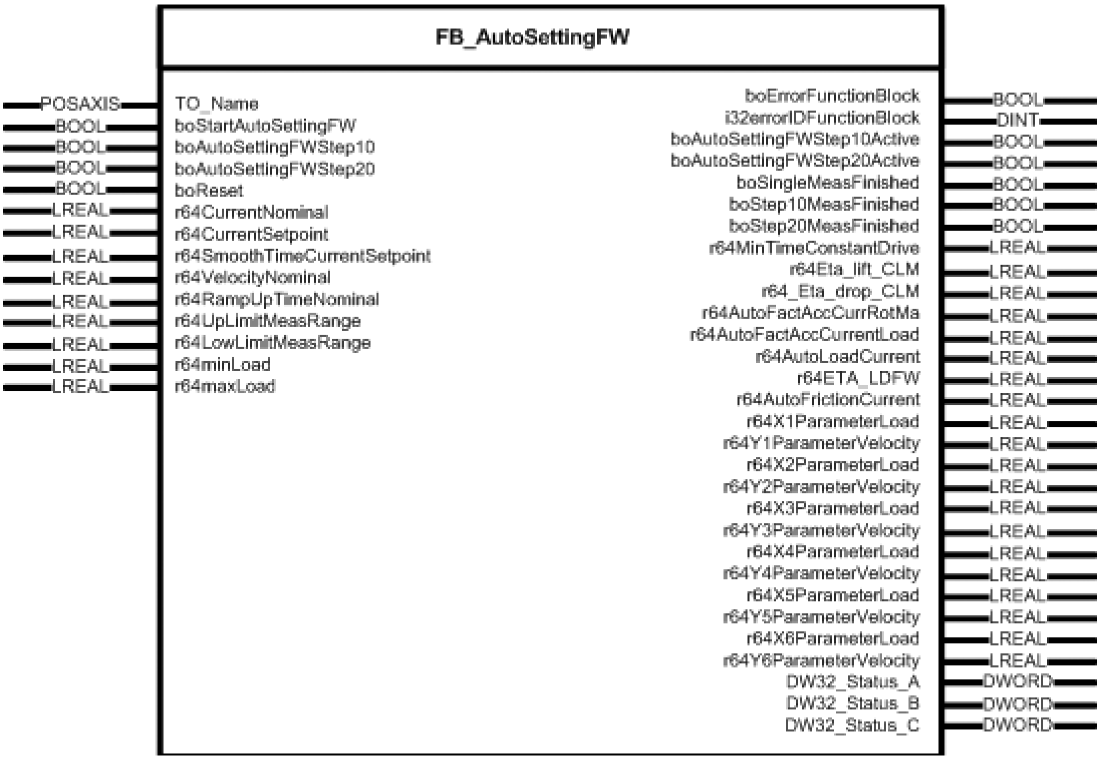

FB_AutoSettingFW
POSAXIS TO_Name
BOOL boStartAutoSettingFW
BOOL boAutoSettingFWStep10
BOOL boAutoSettingFWStep20
BOOL boReset
LREAL r64CurrentNominal
LREAL r64CurrentSetpoint
LREAL r64SmoothTimeCurrentSetpoint
LREAL r64VelocityNominal
LREAL r64RampUpTimeNominal
LREAL r64UpLimitMeasRange
LREAL r64LowLimitMeasRange
LREAL r64minLoad
LREAL r64maxLoad
boErrorFunctionBlock BOOL
i32errorIDFunctionBlock DINT
boAutoSettingFWStep10Active BOOL
boAutoSettingFWStep20Active BOOL
boSingleMeasFinished BOOL
boStep10MeasFinished BOOL
boStep20MeasFinished BOOL
r64MinTimeConstantDrive LREAL
r64Eta_lift_CLM LREAL
r64_Eta_drop_CLM LREAL
r64AutoFactAccCurrRotMa LREAL
r64AutoFactAccCurrentLoad LREAL
r64AutoLoadCurrent LREAL
r64ETA_LDFW LREAL
r64AutoFrictionCurrent LREAL
r64X1ParameterLoad LREAL
r64Y1ParameterVelocity LREAL
r64X2ParameterLoad LREAL
r64Y2ParameterVelocity LREAL
r64X3ParameterLoad LREAL
r64Y3ParameterVelocity LREAL
r64X4ParameterLoad LREAL
r64Y4ParameterVelocity LREAL
r64X5ParameterLoad LREAL
r64Y5ParameterVelocity LREAL
r64X6ParameterLoad LREAL
r64Y6ParameterVelocity LREAL
DW32_Status_A DWORD
DW32_Status_B DWORD
DW32_Status_C DWORD

Figure 5-72 Connections

Name	Connection type	Default setting	Data type	Meaning
TO_Name	IN		POSAXIS	Enter the name of the technology object
boStartAutoSettingFW	IN	FALSE	BOOL	Start of the automatic measurements
boAutoSettingFWStep10	IN	FALSE	BOOL	Start of measuring step 10
boAutoSettingFWStep20	IN	FALSE	BOOL	Start of measuring step 20
boReset	IN	FALSE	BOOL	Reset measuring steps 10 or 20 or both measuring steps
r64CurrentNominal	IN	0.0	LREAL	Reference current [A]
r64CurrentSetpoint	IN	0.0	LREAL	Current setpoint [A]
r64SmoothTimeCurrentSetpoint	IN	50.0	LREAL	Smoothing time for the current [ms]; the time corresponds to 1 Tau
r64VelocityNominal	IN	0.0	LREAL	Rated velocity (base velocity) [mm/s]
r64RampUpTimeNominal	IN	0.0	LREAL	Rated ramp-up time [ms]
r64UpLimitMeasRange	IN	80.0	LREAL	Upper measuring window limit (speed up to which the measuring series is performed) [%]
r64LowLimitMeasRange	IN	40.0	LREAL	Lower measuring window limit (speed above which the measuring series is performed) [%]
r64MinLoad	IN	0.0	LREAL	Minimum load in % or physical unit, such as t (tons), depending on what is present at the LDWF block
r64MaxLoad	IN	0.0	LREAL	Maximum load in % or physical unit, such as t (tons), depending on what is present at the LDWF block
boErrorFunctionBlock	OUT	FALSE	BOOL	Error bit from the function block
i32ErrorIDFunctionBlock	OUT	0	DINT	Error number from the function block
boAutoSettingFWStep10Active	OUT	FALSE	BOOL	Measuring step 10 is active; also refer to Chapter Commissioning support for LDFW with FB_AutoSettingFW (Page 419) or Commissioning support for CLM with FB_AutoSettingFW (Page 439).
boAutoSettingFWStep20Active	OUT	FALSE	BOOL	Measuring step 20 is active.
boSingleMeasFinished	OUT	FALSE	BOOL	Individual measuring step has completed.
boStep10MeasFinished	OUT	FALSE	BOOL	Measuring step 10 has completed.
boStep20MeasFinished	OUT	FALSE	BOOL	Measuring step 20 has completed.
r64MinTimeConstantDrive	OUT	LREAL	0.0	Minimum constant velocity time [ms]
r64Eta_lift_CLM	OUT	LREAL	0.0	Efficiency correction for hoisting; only for DCC_ContLoadMeasurement_1
r64Eta_drop_CLM	OUT	LREAL	0.0	Efficiency correction for lowering; only for DCC_ContLoadMeasurement_1
r64AutoFactAccCurrRotMa	OUT	LREAL	0.0	Evaluation for the acceleration current of the rotating masses

Name	Connection type	Default setting	Data type	Meaning
r64AutoFactAccCurrentLoad	OUT	LREAL	0.0	Evaluation for the load acceleration current
r64AutoLoadCurrent	OUT	LREAL	0.0	Current load [%]
r64Eta_LDFW	OUT	LREAL	0.0	Efficiency correction for hoisting and lowering; only for DCC_LoadDependFieldWeak_1
r64AutoFrictionCurrent	OUT	LREAL	0.0	Friction
r64X1ParameterLoad	OUT	LREAL	0.0	Polygon, load value X1
r64Y1ParameterVelocity	OUT	LREAL	0.0	Polygon, velocity Y1
r64X2ParameterLoad	OUT	LREAL	0.0	Polygon, load value X2
r64Y2ParameterVelocity	OUT	LREAL	0.0	Polygon, velocity Y2
r64X3ParameterLoad	OUT	LREAL	0.0	Polygon, load value X3
r64Y3ParameterVelocity	OUT	LREAL	0.0	Polygon, velocity Y3
r64X4ParameterLoad	OUT	LREAL	0.0	Polygon, load value X4
r64Y4ParameterVelocity	OUT	LREAL	0.0	Polygon, velocity Y4
r64X5ParameterLoad	OUT	LREAL	0.0	Polygon, load value X5
r64Y5ParameterVelocity	OUT	LREAL	0.0	Polygon, velocity Y5
r64X6ParameterLoad	OUT	LREAL	0.0	Polygon, load value X6
r64Y6ParameterVelocity	OUT	LREAL	0.0	Polygon, velocity Y6
DW32_Status_A	OUT	DWORD	0.0	Diagnostic word for internal evaluation
DW32_Status_B	OUT	DWORD	0.0	Diagnostic word for internal evaluation
DW32_Status_C	OUT	DWORD	0.0	Diagnostic word for internal evaluation

Error messages

If an error occurs, output "boErrorFunctionBlock" is set and an error number is issued at output "i32ErrorIDFunctionBlock"; refer to Chapter Application error messages and alarm messages (Page 339).

5.4 Setup and version update for crane libraries

The SIMOCRANE Basic Technology package includes a CD which contains the setup of the Crane_DCC_Library, the online help, the Crane_FB_Library, and the standard projects.

The Crane_DCC_Library and Crane_FB_Library are continuously being developed and updated. To increase performance, the Crane_DCC_Library has been converted to C code (as is standard for the SIMOTION DCC Library) since SIMOTION SCOUT V4.1.4 HF1 and with the setup of the Basic Technology V2.0 SP1, the library no longer appears in the SCOUT project navigator under "Libraries". The functionality of each DCC block is also expanded or optimized. Although some extended blocks are designated with "Name_1", the Crane DCC library remains compatible. To ensure that you can always benefit from the full functional scope of these libraries, a procedure of how to set up the new Crane DCC library and update the existing libraries is described below.

5.4.1 Updating the Crane_FB_Library

To run a version update for the Crane_FB_Library, proceed as follows:

1. Open a project with the old version of the Crane_FB_Library and a new standard project at the same time.
2. Delete the Crane_FB_Library from the old project.
3. Copy the Crane_FB Library from the new project.
4. Paste the Crane_FB_Library from the clipboard into the old project.
5. Select the Crane_FB_Library, right-click to open the context menu, and select "Accept and Compile" to update the library in SCOUT.
6. Once the save and compile operation is complete, the new Crane_FB_Library will be available for use with other applications.

5.4.2 Upgrade of a function in the Crane_FB_Library

Go to the upgrade of a function block (e.g. FB_OperationMode) as follows:

1. Download and unzip the Zip file.
2. Open the application project in SIMOTION SCOUT.
3. In the project navigator, open the LIBRARIES folder and then the Crane_FB_Library.
4. Mark the "Mode" source and delete it.
5. Mark the Crane_FB_Library and after right-click, select "Import external source / ST source".
6. Select the folder that contains the new block. This causes the new 'mode.st' source to be imported into the Crane_FB_Library.
7. Mark the Crane_FB_Library and after right-click, select "Accept and compile".
8. Compile the entire project in the menu with "Project / Save and compile all".

5.4.3 Setting up and updating the Crane_DCC_Library

To set up or update the Crane_DCC_Library, proceed as follows:

1. Select Control Panel → Add or Remove Programs and uninstall the "SIMOCRANE Basic Technology" program as shown in the figure below. Make sure that all SCOUT and Step 7 programs are closed when you do this.

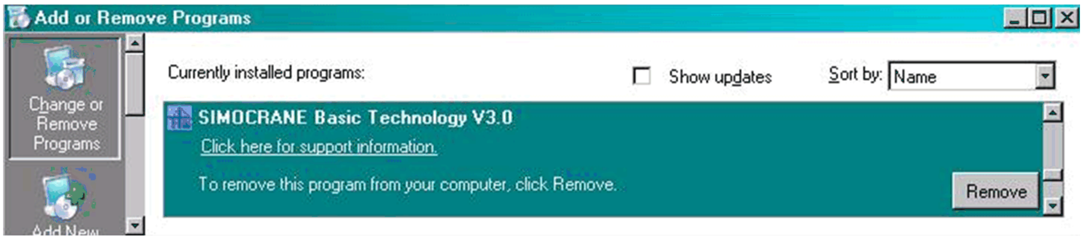

Add or Remove Programs
Change or Remove Programs
Add New
Currently installed programs:
☐ Show updates
Sort by: Name
SIMOCRANE Basic Technology V3.0
Click here for support information.
To remove this program from your computer, click Remove.
Remove

Figure 5-73 Uninstalling SIMOCRANE Basic Technology

2. You can now install the new Crane_DCC_Library by running the setup from the CD.
3. Open your project and delete the "Crane_DCC_Library" from the Libraries folder, if it is still there.
4. Right-click to select "D435" and open the settings for "Select Technology Packages". Select the following technology packages:
   - DCBADM(V4.3)
   - TPdcblib_SIMOTION_4_3
   - TPcrane_dcc_library_SIMOTION_4_1_5 (V30.XX)
   - CAM_EXT (V4.3)
5. Click "Update" and then "OK" (to close the window).
6. Open a DCC chart (e.g. Gantd_1) then select Options → Block Types and delete all imported libraries from the chart by clicking the "<<" button. Then import into the DCC plan all libraries installed in SCOUT by clicking the ">>" button.

Run the save and compile operation to complete the update of the Crane_DCC_Library. All of the functions and features of the Crane_DCC_Library are now available for you to use.

Communication 6

6.1 General overview

The communication between the higher-level SIMATIC S7 controller and the drive-related SIMOTION D435-2 DP/PN controller is established via PROFIBUS DP / PROFINET RT (up to max. 16 PZD for each drive). With PROFIBUS DP, up to 244 bytes can be transferred in each direction and 16 PZD per drive allow up to seven drives to be implemented. With PROFINET RT, up to 1400 bytes can be transferred in each direction and with 16 PZD per drive, all of the drives required for a crane plus additional data for Advanced Technology can be implemented. The SINAMICS Control Unit drive is integrated in the drive-based SIMOTION D. Communication between both (SIMOTION and SINAMICS) works in the same way as with PROFIBUS DP, but with Motion Control expansion up to a maximum of 32 PZD per drive (according to the PROFIdrive profile for drive technology, refer to Ref. [1]).

SIMOTION D435-2 DP/PN has three PROFIBUS DP interfaces (two DP interfaces and one DP Integrated interface). The communication for PROFIBUS operation between SIMATIC and SIMOTION D435-2 DP/PN for Basic Technology is configured on DP 1. Communication between SIMOTION and SINAMICS is permanently configured on integrated DP. In PROFIBUS operation, the DP 2 interface is reserved for communication between SIMATIC and SIMOTION D435-2 DP/PN involving sway control or other options, refer to Ref. [12].

SIMOTION D435-2 DP/PN has two PROFINET interfaces (an integrated PN interface with three ports and an interface with the CBE20-2 options board) and assumes the role of a PROFINET I/O controller. The communication between SIMATIC and SIMOTION D435-2 DP/PN for Basic Technology and Advanced Technology is configured at port 1 for PROFINET RT operation.

Note

The definition of the telegram for Advanced Technology is not part of this documentation.

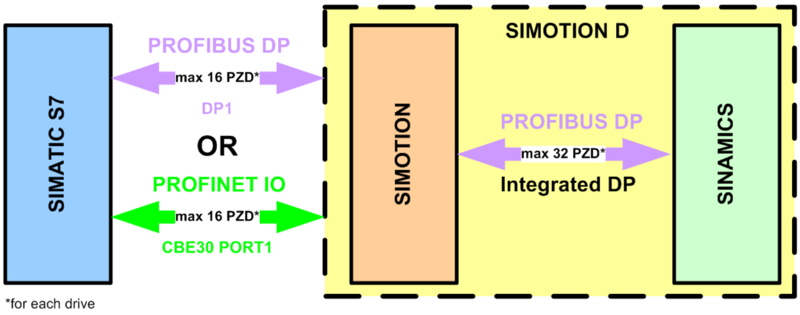


Figure 6-1 Communication overview

Task distribution

The following list provides an overview of task distribution between SIMATIC S7, SIMOTION, and SINAMICS.

SIMATIC S7:

- Overview of the crane
- I/O signal processing
- Infeed control
- Safety-related monitoring
- Setpoint and control commands for main drives
- Setpoint and control commands for auxiliary drives
- Brake control (optional)

SIMOTION:

- Crane technology in DCC
- Setpoint generator, position control, synchronous operation, and communication with the drives in the TO
- Communication with SIMATIC and SINAMICS in MCC
- Sequence control (monitoring, use of Operation Mode, troubleshooting, etc.) in MCC
- Additional crane-specific functions in MCC, such as cornering movement, brake test, flying homing, offset mode, auto-setting …
- The use of a Web-based tool (WebStart) for commissioning and diagnostics is possible as of version Basic Technology V3.0.

SINAMICS:

- Speed control
- Current control
- Monitoring, if required
- Brake control (optional)

Prerequisites

In order to establish disturbance-free communication between these three parties, the reference data in SIMATIC, SIMOTION and SINAMICS modules must be identically configured. The reference data must be normalized to the maximum data.

The following parameters are involved in the modules:

SIMATIC:

- Reference speed
- Reference voltage
- Reference current
- Reference torque
- Reference power

SIMOTION:

- TypeOfAxis.SetPointDriverInfo.DriveData.maxSpeed: Drive speed
- TypeOfAxis.MaxVelocity.maximum: Maximum permissible velocity (refer to Chapter SIMOTION technology object (Page 407))
- MCC Interface "axis"_nominalVelocity, _nominalCurrent, _nominalVoltage, _nominalTorque, _nominalPower

SINAMICS:

- p2000 = reference speed reference frequency (≈ p1082 = maximum speed)
- p2001 = reference voltage
- p2002 = reference current
- p2003 = reference torque
- r2004 = reference power

NOTICE

All reference data must be normalized to their particular maximum value:

n_{Max} = 100% = n_{rated} + n_{Fw}
(maximum speed = rated speed + field weakening speed)

I_{Max} = 100% = I_{rated} + $I_{overload}$
(maximum current = rated current + overload current)

M_{Max} = 100% = M_{rated} + $M_{overload}$
(maximum torque = rated torque + overload torque)

Note

A script is created to fetch all reference values from the SINAMICS and enter in SIMOTION; also refer to Chapter General (Page 361), "Script files" section

The definition for the relationship between the percentage value and hexadecimal value is shown in the following table and in the following diagram.

If only positive values are used in the process data, then the value range is doubled; this means that the range starts at 0 and can be increased up to 65,535 (UINT, 0 to FFFF).

Table 6- 1 Relationship between hexadecimal value and percentage value

1 word: Connector	
Percentage value	Hexadecimal value
0	16#0000
100	16#4000
199,994	16#7FFF
- 200	16#8000
- 100	16#C000
0,006	16#1

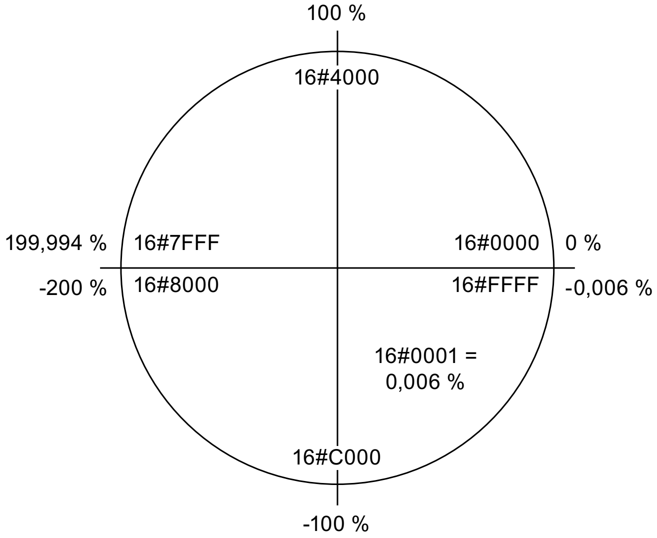


Figure 6-2 Representing the value range of a word

New features with version 3.0

The time synchronization has been added in version 3.0. The following I/O areas are occupied:

SIMATIC S7 to SIMOTION communication:

- One double word as UDINT (u32ClockTimeS7) address PID 1904
- One word (b16daysS7) address PIW 1908
- One word (b16timestw) address PIW 1910

SIMOTION to SIMATIC S7 communication:

- One double word as UDINT (u32ClockTimeSimotion) address PQD 1904
- One word (b16daysSimotion) address PQW 1908

6.2 Communication, SIMATIC – SIMOTION D

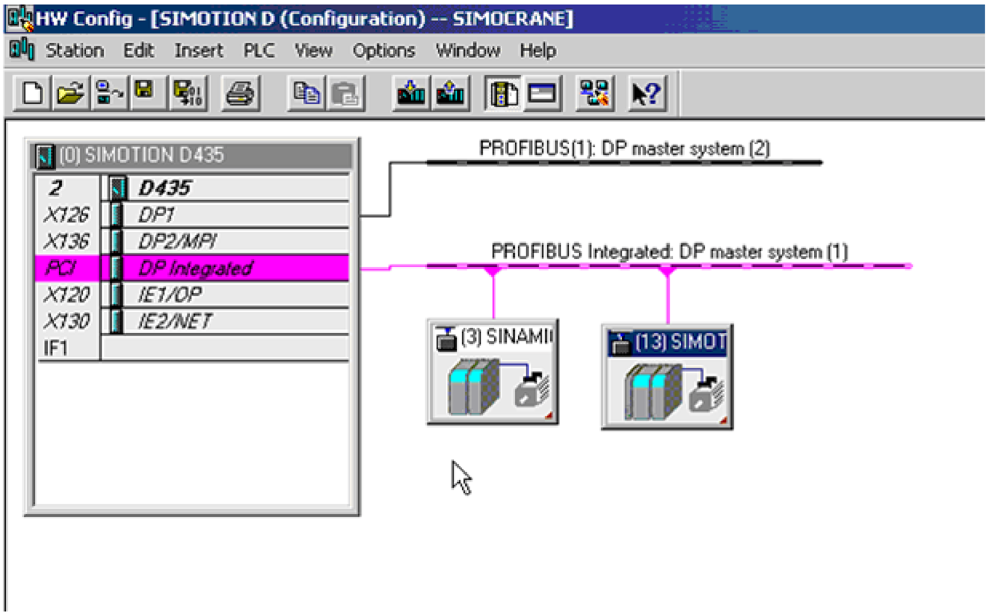
6.2.1 Configuring the connection

PROFIBUS DP

Communication between SIMATIC and SIMOTION D is established via PROFIBUS DP. A connection from the DP interface of the S7 module to the SIMOTION DP1 interface is established at the SIMATIC. A PROFIBUS data transfer rate of **6 Mbit/s** is set as default value. The S7 module is master and SIMOTION is an intelligent slave.

The DP1 segment is the default D-MASTER in the SIMOTION hardware configuration.

The configuration of the connection for PROFINET RT is similar to that for PROFIBUS DP and is described in the following documentation, refer to Ref. 14. Additional information can be taken from Ref. 15 and 16.

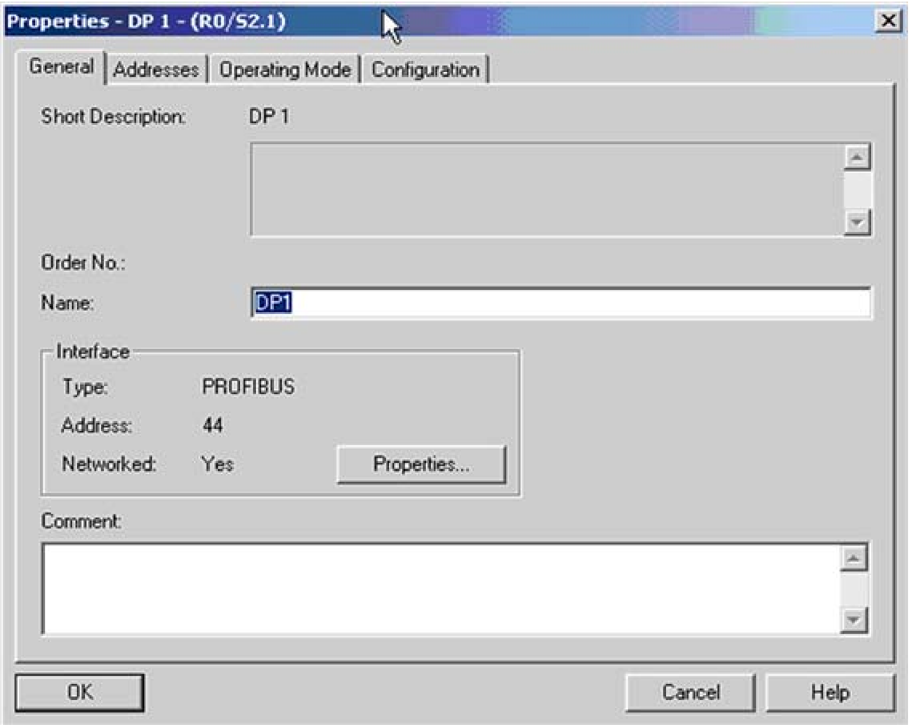

HW Config - [SIMOTION D (Configuration) -- SIMOCRANE]
Station
Edit
Insert
PLC
View
Options
Window
Help
PROFIBUS(1): DP master system (2)
PROFIBUS Integrated: DP master system (1)
(3) SINAMI
(13) SIMOT

(0) SIMOTION D435

2	D435
X126	DP1
X136	DP2/MPI
PCI	DP Integrated
X120	IE1/OP
X130	IE2/NET
IF1	

Figure 6-3 Hardware configuration SIMOTION

The PROFIBUS address of the DP1 line of SIMOTION should be defined as 44.



Properties - DP 1 - (R0/S2.1)

Short Description: DP 1

Order No.:

Name: DP1

Interface

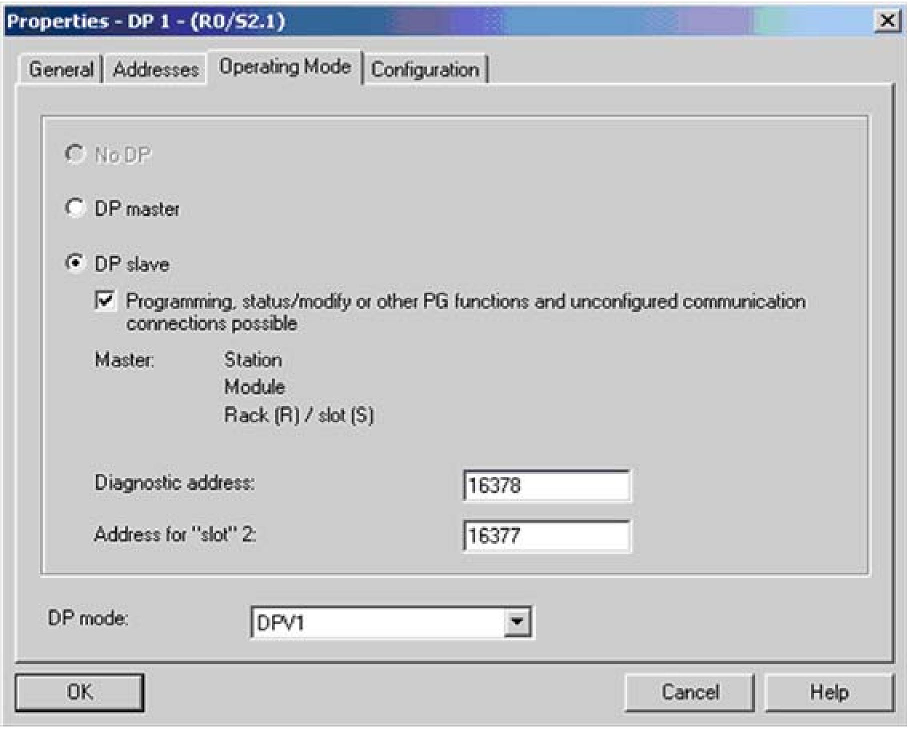
Type: PROFIBUS

Address: 44

Networked: Yes

Figure 6-4 Object properties of DP1

The DP1 line must now be configured as intelligent DP-SLAVE. This is done under **Properties – DP1** on the tab **Operating Mode**. The checkbox for **Programming, status/modify or other PG functions and unconfigures communication connections possible** MUST be set.



○ No DP

○ DP master

● DP slave

☑ Programming, status/modify or other PG functions and unconfigured communication connections possible

Master: Station
Module
Rack (R) / slot (S)

Diagnostic address: 16378

Address for "slot" 2: 16377

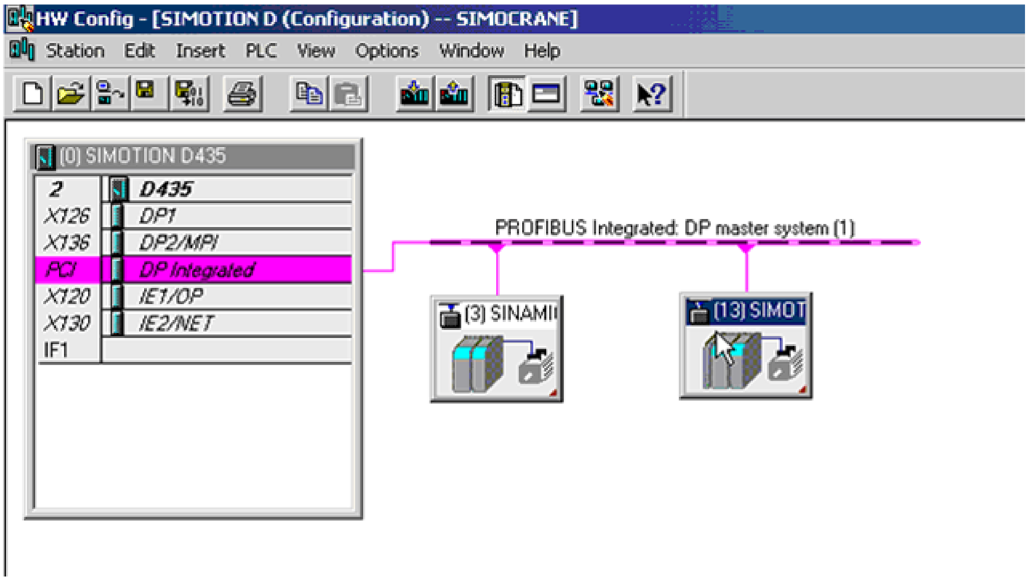
DP mode: DPV1

Figure 6-5 Setting up an intelligent DP slave

After the DP1 line has been configured as an intelligent SLAVE, the hardware configuration looks like that shown below.

Note

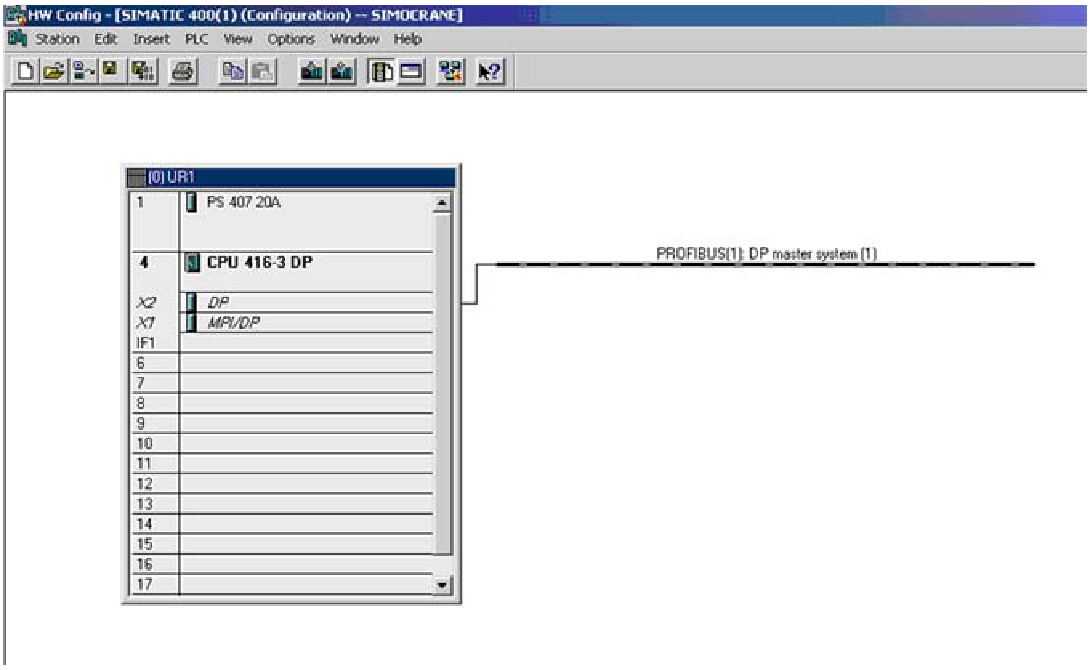
The DP1 line can no longer be seen in the hardware configuration after this has been configured as intelligent slave.


HW Config - [SIMOTION D (Configuration) -- SIMOCRANE]
Station Edit Insert PLC View Options Window Help
(0) SIMOTION D435
PROFIBUS Integrated: DP master system (1)
(3) SINAMI
(13) SIMOT

2	D435
X126	DP1
X136	DP2/MPI
PCI	DP Integrated
X120	IE1/OP
X130	IE2/NET
IF1	

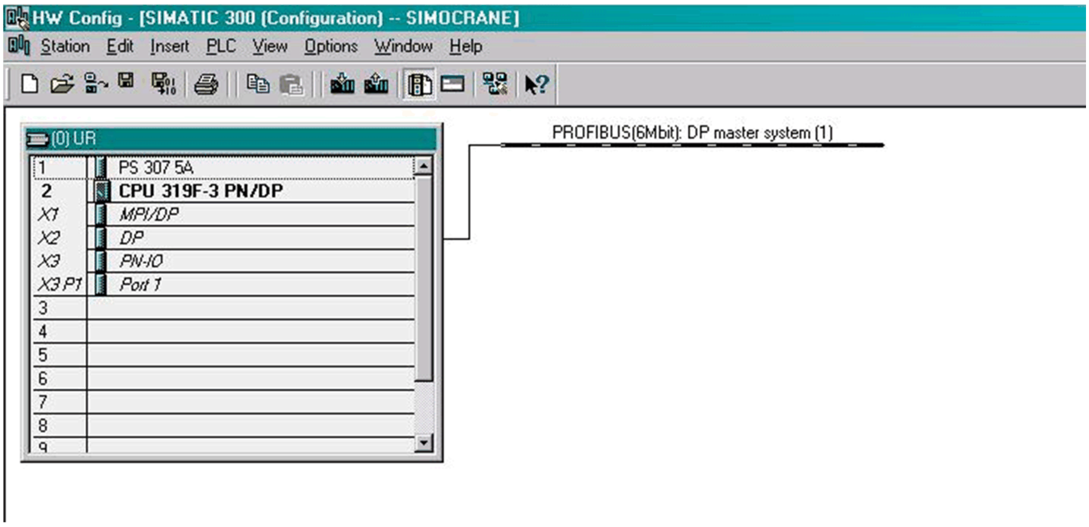
Figure 6-6 Overview of the hardware configuration after setup of an intelligent DP slave

The DP segment is set per default in the hardware configuration of SIMATIC as PROFIBUS with a data transfer rate of **6 Mbit/s**. If the 6 Mbit/s cannot be attained, a different data transfer rate can be set. Then, the SIMOTION is inserted from the catalog under "already configured stations":


HW Config - [SIMATIC 400(1) (Configuration) -- SIMOCRANE]
Station Edit Insert PLC View Options Window Help
(0) UR1
PROFIBUS(1): DP master system (1)

1	PS 407 20A
4	CPU 416-3 DP
X2	DP
X1	MPI/DP
IF1	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	

Figure 6-7 Overview of the SIMATIC S7-400 HW configuration


HW Config - [SIMATIC 300 (Configuration) -- SIMOCRANE]
Station Edit Insert PLC View Options Window Help
(0) UR
PROFIBUS(6Mbit): DP master system (1)

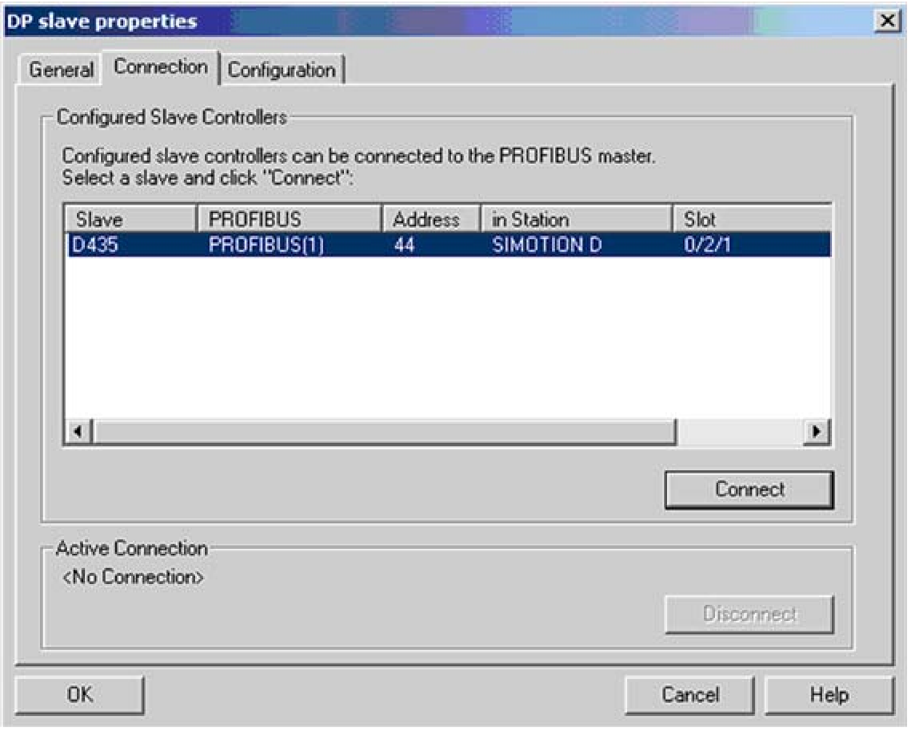
1	PS 307 5A
2	CPU 319F-3 PN/DP
X1	*MPI/DP*
X2	*DP*
X3	*PN-IO*
X3 P1	*Port 1*
3	
4	
5	
6	
7	
8	
9	

Figure 6-8 Overview of the SIMATIC S7-300 HW configuration

When inserting the already configured SIMOTION station, a coupling is listed, refer to the diagram below. Click the **Connect** button so that the coupling is established between SIMATIC and SIMOTION.

Note

Communication between SIMATIC and SIMOTION can only be configured after the coupling has been established.



Configured slave controllers can be connected to the PROFIBUS master.
Select a slave and click "Connect":

Slave	PROFIBUS	Address	in Station	Slot
D435	PROFIBUS(1)	44	SIMOTION D	0/2/1

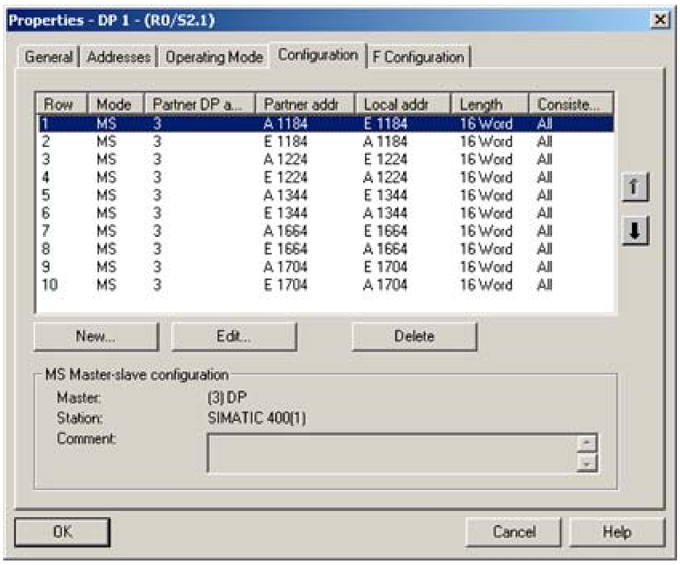
Active Connection
<No Connection>

Figure 6-9 Object properties of DP

After the coupling has been established, data transfer between SIMATIC and SIMOTION must be configured. 16 process data items must be configured for every node (TO). The address areas for communication between SIMATIC and SIMOTION are defined as follows in the document "CraneSoft-7".

Drives	PROFIBUS address	I/O address area
Hoist 1 / Holding Gear 1	44	1184 … 1223
Hoist 2 / Closing Gear 1	44	1224 … 1263
Hoist 3/Holding Gear 2	44	1264 … 1303
Hoist 4 / Closing Gear 2	44	1304 … 1343
Trolley 1	44	1344 … 1383
Trolley 2	44	1384 … 1423
Boom 1/Luffing Gear 1	44	1504 … 1543
Boom 2	44	1544 … 1583
Gantry 1	44	1664 … 1703
Gantry 2	44	1704 … 1743
Slewing Gear 1	44	1824 … 1863
Slewing Gear 2	44	1864 … 1903
Application extensions	44	1904 … 1943

For every node, these addresses must be generated under **DP slave properties** under the tab **Configuration.**



Properties - DP 1 - (R0/S2.1)

Row	Mode	Partner DP a...	Partner addr	Local addr	Length	Consiste...
1	MS	3	A 1184	E 1184	16 Word	All
2	MS	3	E 1184	A 1184	16 Word	All
3	MS	3	A 1224	E 1224	16 Word	All
4	MS	3	E 1224	A 1224	16 Word	All
5	MS	3	A 1344	E 1344	16 Word	All
6	MS	3	E 1344	A 1344	16 Word	All
7	MS	3	A 1664	E 1664	16 Word	All
8	MS	3	E 1664	A 1664	16 Word	All
9	MS	3	A 1704	E 1704	16 Word	All
10	MS	3	E 1704	A 1704	16 Word	All

MS Master-slave configuration

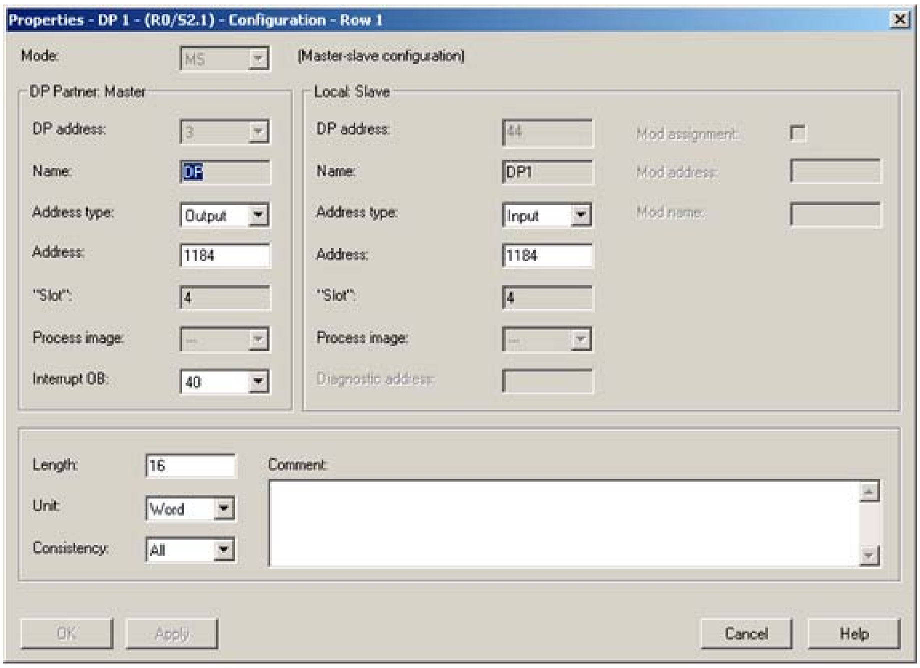
Master: (3) DP
Station: SIMATIC 400(1)
Comment:

Figure 6-10 DP configuration

The send and receive direction must be configured for every node. Hoist 1 with the starting address 1184 is configured once from the master as output and once from the slave as input.

The following settings in the DP slave properties window are very important:
– **Length**: 16
– **Units**: Word
– **Consistency**: All

This guarantees that communication between SIMATIC and SIMOTION is consistent when sending and receiving with 16 words.



Properties - DP 1 - (R0/S2.1) - Configuration - Row 1

Mode: MS (Master-slave configuration)

DP Partner: Master

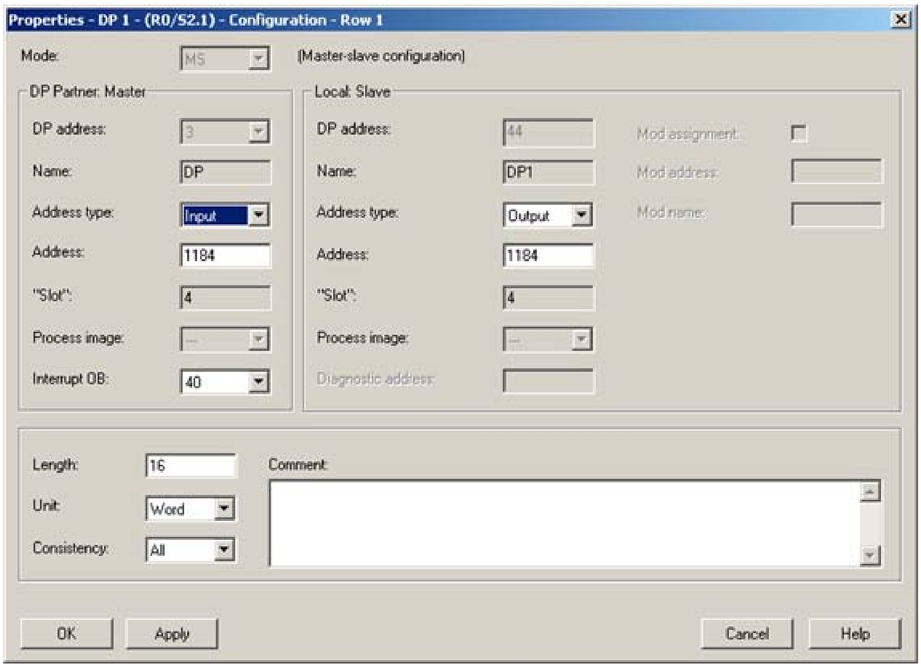
- DP address: 3
- Name: DP
- Address type: Output
- Address: 1184
- "Slot": 4
- Process image: ---
- Interrupt OB: 40

Local: Slave

- DP address: 44
- Name: DP1
- Address type: Input
- Address: 1184
- "Slot": 4
- Process image: ---
- Diagnostic address:
- Mod assignment: □
- Mod address:
- Mod name:

Length: 16
Unit: Word
Consistency: All
Comment:

Figure 6-11 Properties of the output words to be created



Properties - DP 1 - (R0/S2.1) - Configuration - Row 1

Mode: MS (Master-slave configuration)

DP Partner: Master

- DP address: 3
- Name: DP
- Address type: Input
- Address: 1184
- "Slot": 4
- Process image: ---
- Interrupt OB: 40

Local: Slave

- DP address: 44
- Name: DP1
- Address type: Output
- Address: 1184
- "Slot": 4
- Process image: ---
- Diagnostic address:
- Mod assignment: □
- Mod address:
- Mod name:

Length: 16
Unit: Word
Consistency: All
Comment:

Figure 6-12 Properties of the input words to be created

After all of the devices of the basic technology have been configured, the hardware configuration must be compiled in the SIMATIC and SIMOTION.

The complete SIMATIC and SIMOTION hardware is displayed in the "NETPRO" screen. The PG/PC should be connected at the DP1 line. An active connection must be configured at the PG/PC; otherwise routing to SINAMICS_Integrated and the CX32 module is not possible.

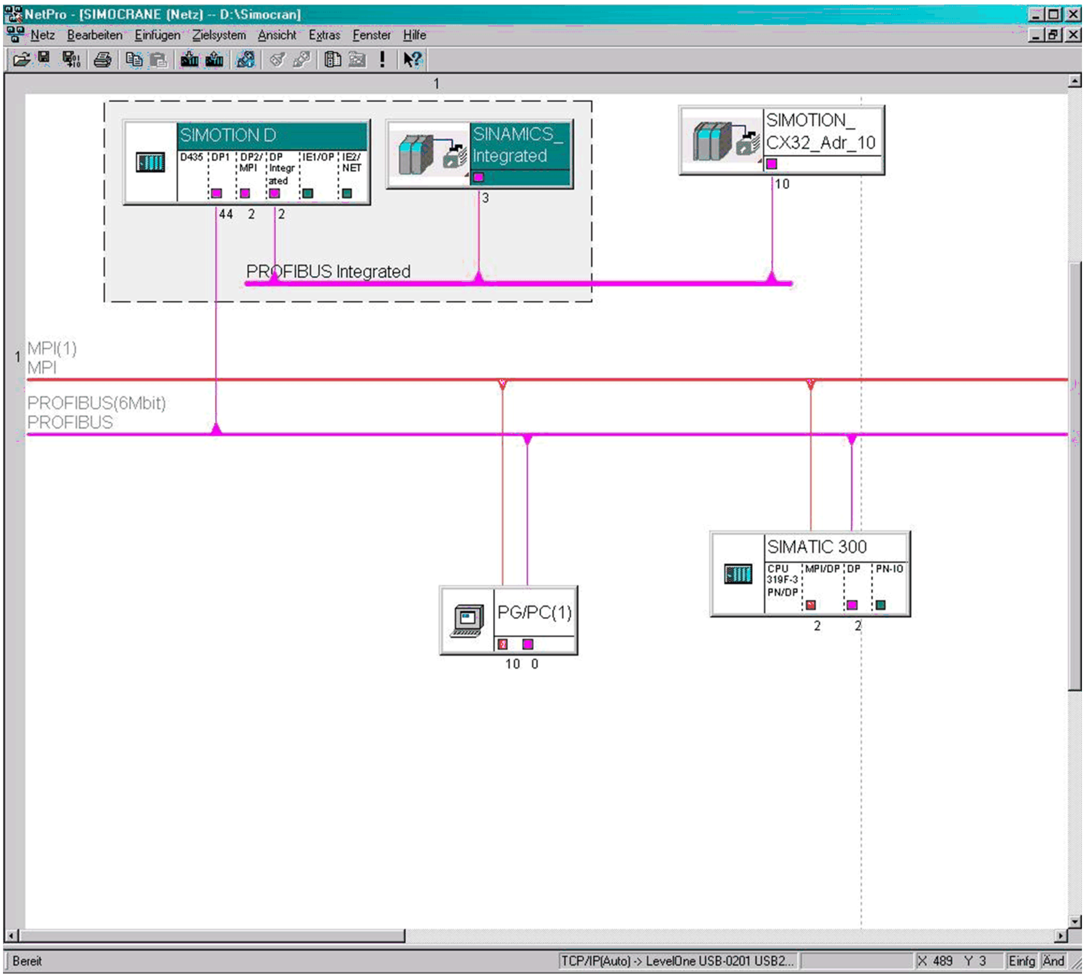
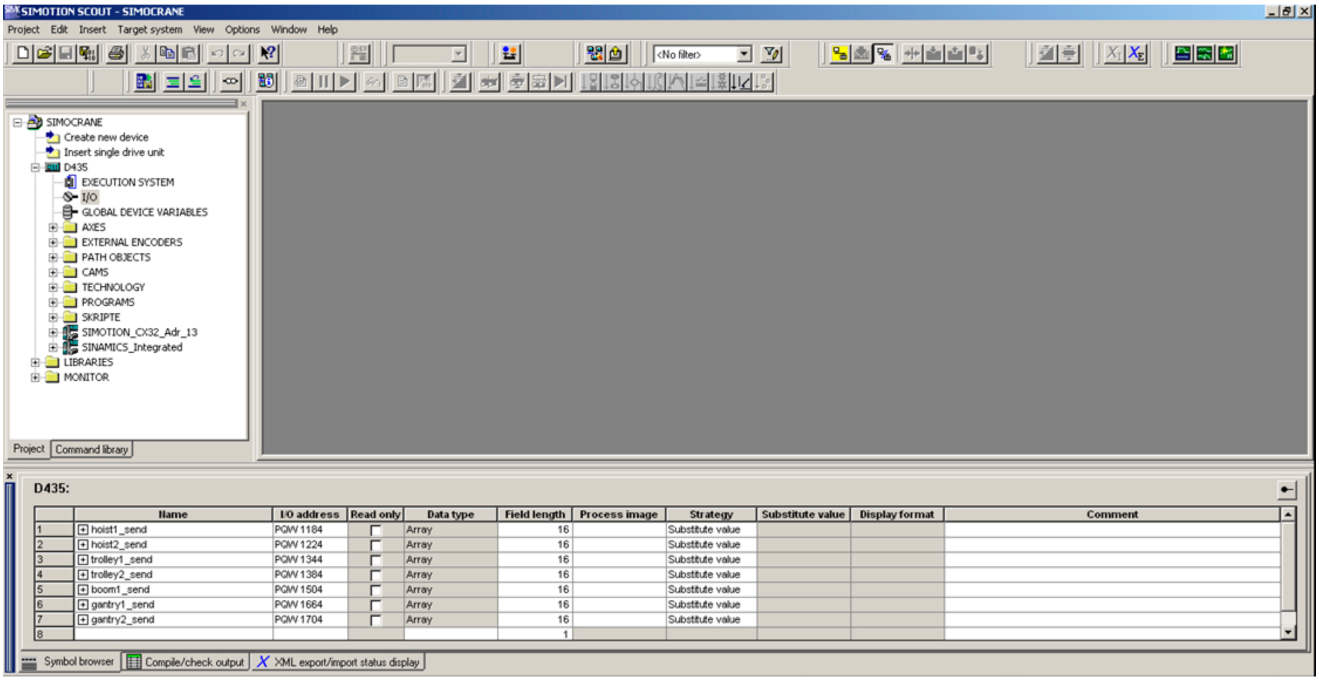


Figure 6-13 NetPro_Overview

The complete communication with the I/O or with the master via the I/O container is realized in SIMOTION. An array with 16 process data items must be created for receiving and sending each function module participating in communication.

The arrays must be integrated into the corresponding function modules as a user program with the function blocks "FB_TelegramS7ToSimotion" and "FB_TelegramSimotionToS7".



D435:

	Name	I/O address	Read only	Data type	Field length	Process image	Strategy	Substitute value	Display format	Comment
1	⊞ hoist1_send	PQW 1184	□	Array	16		Substitute value			
2	⊞ hoist2_send	PQW 1224	□	Array	16		Substitute value			
3	⊞ trolley1_send	PQW 1344	□	Array	16		Substitute value			
4	⊞ trolley2_send	PQW 1384	□	Array	16		Substitute value			
5	⊞ boom1_send	PQW 1504	□	Array	16		Substitute value			
6	⊞ gantry1_send	PQW 1664	□	Array	16		Substitute value			
7	⊞ gantry2_send	PQW 1704	□	Array	16		Substitute value			
8					1					

Figure 6-14 Overview of the I/O Container in SCOUT (communication between SIMATIC and SIMOTION)

PROFINET RT

The configuration of the PROFINET RT connection is similar to that for PROFIBUS DP and is described in the following documentation, refer to Ref. 14. Additional information can be taken from Ref. 15 and 16.

A configuring example is provided on CD, refer to Chapter STS tandem crane (ship-to-shore with four hoists) (Page 517).

6.2.2 Defining the telegram

An S7 module is used for the standard crane application; this can exchange 244 bytes of data between SIMATIC and SIMOTION. This is the reason that with PROFIBUS and 16 PZD per drive only seven drives can be implemented. If more drives are to be implemented under PROFIBUS, then the data exchange must be reduced from 16 PZD back to 12 PZD. Then you can implement ten drives with PROFIBUS; or otherwise you should use PROFINET. The telegram structure for PROFIBUS and PROFINET is the same.

Two function blocks (FB_TelegramS7ToSimotion, FB_TelegramSimotionToS7) are programmed in the basis technology to process communication telegrams (i.e. interconnect and convert signals) between SIMATIC S7 and SIMOTION. In addition, the corresponding function blocks and data blocks on the SIMATIC S7 side for defined communication are made available to users on the Software CD.

6.2.2.1 SIMATIC S7 → SIMOTION

Table 6- 2 SIMATIC S7 → SIMOTION

PZD	Signal name	Unit	Remarks
1	Control word_1_S7		See below, Table "Control word_1_S7"
2	Master switch_S7	as a % of the reference quantity	When the master switch is deflected, the velocity is entered from the master switch. The velocity setpoint can be specified either as a positive or negative value. The reference value is the maximum value - in the S7 as well as in SIMOTION and in SINAMICS. $n_{max} = n_{rated} + n_{Fw}$ = 100% 4000 hex = 16384 dec = 100%
3	Control word_2_S7		See below, Table "Control word_2_S7"
4	Automatic velocity	as a % of the reference quantity	In the AUTOMATIC operating mode, the S7 specifies an absolute velocity if the master switch is not deflected. If, during an active AUTOMATIC request, the master switch is deflected, the actual request is canceled and the velocity specified by the master switch is used (for safety-related reasons). The reference value is the maximum value - in the S7 as well as in SIMOTION and in SINAMICS. $n_{max} = n_{rated} + n_{Fw}$ = 100% 4000 hex = 16384 dec = 100%
5	Ramp-up time_S7	ms	A value range of 0 ms to 65,535 ms (UINT) is provided for the ramp-up and ramp-down time. The ramp-up and ramp-down times are valid for all operating modes. The acceleration and deceleration ramp [mm/s^2] are converted, referred to the maximum velocity. FFFF hex = 65535 dec
6	Ramp-down time_S7		
7	Setpoint for the start pulse_S7 / free	as a % of the reference quantity	Only for Hoist, Holding Gear, Closing Gear and Boom The setpoint for the start pulse is specified as normalized value in percent (%). 4000 hex = 16384 dec = 100%
8	Actual working radius / torque limit / free	mm / 0-100%	Actual working radius only for Slewing Gear or torque limit specification, for example, for the brake test
9	Application_control_word_1_S7	-	See below, Table "Application_control_word_1_S7"
10	Application_control_word_2_S7	-	See below, Table "Application_control_word_2_S7"
11	Target_position_S7	mm	In the AUTOMATIC, MANUAL and EASY_POSITIONING operating modes, the target position is specified in millimeters [mm]. S7 transfers the target position as DINT to the SIMOTION. The target position can be specified as either positive or negative value. The value range corresponds to the DINT, i.e. -2**31 to +2**31-1
12			
13	Start rounding-off time	ms	A value range 0 ms ... 65,535 ms (UINT) is provided for the initial and final rounding-off time. The initial rounding-off time is valid for all operating modes. FFFFhex= 65535dec
14	End rounding-off time	ms	A value range 0 ms ... 65,535 ms (UINT) is provided for the initial and final rounding-off time. The final rounding-off time is valid for all operating modes. FFFFhex= 65535dec

PZD	Signal name	Unit	Remarks
15	Material factor / steering setpoint / free	%	Material factor as an integer value for the Easy Closing function in the holding gear Additional setpoint for the Gantries in the RTG project Free for user-engineered applications
16	-	-	Free for user-engineered applications

PZD 8: Actual working radius

The actual working radius is specified in millimeters [mm] and transferred from the S7 to the SIMOTION as INT. The actual working radius can be specified as either **a positive or negative value**.

If the actual working radius cannot be used, then the PZD8 for the torque limit value (0% - 100%) can be used for the brake test.

Note

Monitoring the safety areas:

In the SIMOCRANE Basic Technology, SIMOTION is not used to monitor safety areas. This means that the SIMATIC S7 monitors the safety areas (e.g. operating limit switch, emergency limit switch); it then handles the following derived measures (e.g. disconnecting the power) and sends the safety-related positions to the SIMOTION.

Note

Monitoring the infeed

SIMATIC S7 controls (activation/deactivation) the infeed (ALM), refer to Ref. 6.

Note

Ramp-up and ramp-down times:

Minimum values for the ramp-up and ramp-down times are configured in the SIMOTION. If the S7 transfers a shorter ramp-up or ramp-down time to the SIMOTION, then the minimum value from SIMOTION is used.

The effective ramp-up and ramp-down times are increased when using initial and final rounding-off (smoothing). The following formula for the ramp-up and ramp-down times (equivalent to Masterdrives Vector Control) is obtained:
$t_{\text{effective}} = t_{\text{ramp-up time}} + (t_{\text{initial rounding-off}} + t_{\text{final rounding-off}})$

Note

Initial jerk time and final jerk time apply to acceleration and deceleration.

Table 6- 3 Control word_1_S7

Bit	Signal name	Remarks
0	On / Off1	The drive is powered up and powered down using On / Off1.
1	Off2	Off2 must be set otherwise the drive cannot be powered up. If input Off2 is withdrawn when the drive is powered up, the pulses must be inhibited and the motor coasts down.
2	Off3	Off3 must be set otherwise the drive cannot be powered up. If input Off 3 is withdrawn when the drive is powered up, the motor is stopped with the configured down ramp and is then powered down.
3	Pulse enable	The setpoint is only accepted and the travel command executed when input "pulse enable" is set.
4	Ramp-function generator enable	The ramp-function generator is enabled by setting the bit. When this bit is reset, the output of the ramp-function generator is set to zero.
5	Reserved	
6	Enable setpoint	When the input "Enable setpoint" is set, the speed setpoint in front of the ramp-function generator is enabled.
7	Acknowledge fault	When the input "Acknowledge fault" is set, a fault that is present is acknowledged as soon as the fault cause has been removed.
8	Enable speed controller	The speed controller is enabled by setting the bit. The speed control is inhibited with a low signal. This means that a travel command cannot be processed - but the motor remains magnetized
9	Open brake	The "Open brake" command can be set in the SINAMICS using the high signal. It can be used when the internal brake control of the SINAMICS is used and the extended brake control is activated.
10	Master control by PLC	The bit "Master ctrl by PLC" must be set, otherwise the higher-level controller does not accept any commands.
11	Reserved	
12	Used internally	Internally used: Closed-loop speed control = 0; closed-loop torque control = 1 Bit 12 cannot be influenced from the S7 in the standard application.
13-15	Reserved	

Table 6- 4 Control word_2_S7

Bit	Signal name	Comment
0	PositiveSuperimpose	A positive supplementary velocity is superimposed by setting the bits.
1	NegativeSuperimpose	A negative supplementary velocity is superimposed by setting the bits.
2	TandemHoming	A positive edge activates compensation of the offset between the tandem master and tandem slave.
3	TandemMode	TandemMode must be selected at the tandem master and the tandem slave if movement is to be performed in tandem mode. (This is true only for hoisting gear applications); refer to Offset mode between tandem master and tandem slave (TandemMode) (Page 478).
4	SlaveTandemMode	If the input is **not set**, the axis is the **Tandem-Master** and TandemMasterMode is activated. If the input is **set**, the axis is the **Tandem Slave** and SlaveTandemMode is activated. (This is true only for hoisting gear applications.)

Bit	Signal name	Comment
5	DriveMasterSuperimpose	True = the supplementary velocity is superimposed on the tandem-master. False = the supplementary velocity is superimposed on the tandem-slave. (This is true only for hoisting gear applications.)
6	SelectMasterAxis2	These inputs specify the master. 1. SelectMasterAxis2 = FALSE AND SelectMasterAxis3 = FALSE. This means the axis at TO_CheckbackMasterAxis is the master. 2. SelectMasterAxis2 = TRUE AND SelectMasterAxis3 = FALSE. This means the axis at TO_CheckbackMasterAxis2 is the master. 3. SelectMasterAxis2 = FALSE AND SelectMasterAxis3 = TRUE. This means the axis at TO_CheckbackMasterAxis3 is the master. 4. SelectMasterAxis2 = TRUE AND SelectMasterAxis3 = TRUE. This causes an error to be output.
7	SelectMasterAxis3	
8	SelectCheckbackSlave	If a slave is available and SelectCheckbackSlave is active, all checkback signals from the slave (boSelectcheckbackSlave) are monitored. (This is true only for hoisting gear applications.)
9	SelectCheckbackSlaveTandem	If a slave is available and SelectCheckbackSlaveTandem is active, all checkback signals from the tandem-slave (boSelectcheckbackSlaveTandem) are monitored. (This is true only for hoisting gear applications.)
10	SelectCheckbackSlaveSlaveTandem	If a slave is available and SelectCheckbackSlaveSlaveTandem is active, all checkback signals of the SlaveSlaveTandem (boSelectcheckbackSlaveSlaveTandem) are monitored.
11	boOffsetHoming	A rising edge activates the compensation of the offset between master and slave. (This is true only for hoisting gear applications); see Offset mode between the master and the slave (Page 479).
12	boOffsetMode	boOffsetMode must be selected on the slave when an offset should be active during travel. (This is true only for hoisting gear applications.)
13 - 15	Free	The bit is not used.

Note

For more information on STW2 bits 0/1/2/3/4/5/11/12, refer to Chapter Ship-to-shore tandem crane (STS tandem) (Page 470).

Table 6- 5 Application_control_word_1_S7

Bit	Signal name	1 signal	0 signal	Remarks
0	Enable ramp-function generator steering / free	Yes	No	This bit is currently used only for RTG projects, otherwise free.
1	Free	Yes	No	The bit is not used.

Bit	Signal name	1 signal	0 signal	Remarks
2	Enable start pulse	Yes	No	A high signal sets a constant or a speed setpoint issued from the S7 for a configurable time (method 1) or the I component of the speed controller is saved and set (method 2, see DCC_StartPulse (Page 150)). This decays according to an e function so that when the brake is opened, the load does not sag. **Note:** When the start pulse is set, the ramp-function generator enable must not be set (only for method 1).
3	Select heavy-duty operation	Yes	No	Heavy-duty operation is permitted with the high signal; heavy-duty operation is not permitted with a low signal. Velocity, acceleration and deceleration are reduced in heavy-duty operation.
4	Enable field-weakening	Yes	No	With a high signal, the drive is permitted to operate in field weakening beyond its rated speed
5	Reset load memory	Yes	No	The high signal resets the saved actual load of the calculation in the DCC_LoadDependingFieldWeak block.
6	Selecting master-slave operation	Yes	No	See below
7	Selecting synchronous operation	Yes	No	See below
8	Save offset	Yes	No	With a high signal, the actual offset in synchronous operation or master-slave operation between the master and slave is not equalized - instead, it is kept. This means the target position is adapted in the AUTOMATIC, MANUAL or EASY_POSITIONING operating modes. The new offset is saved with a positive edge. The offset value is only deleted if active synchronous operation or active master-slave operation is deselected.
9	Start AUTOMATIC operating mode	Yes	No	Using a high signal, the AUTOMATIC request is started with the specified values for position, velocity, acceleration and deceleration (positioning started). The accepted setpoints remain active • until the target position is reached • until the AUTOMATIC operating mode is exited • until the master switch is deflected • until the new edge "Start AUTOMATIC request" is set.
10	AUTOMATIC operating mode	Yes	No	See below
11	MANUAL operating mode	Yes	No	See below
12	EASY_POSITIONING operating mode	Yes	No	See below
13	SPEED_CONTROLLED operating mode (= JOGGING)	Yes	No	Control type: Speed-controlled with encoder This operating mode is selected only after operating mode interlocking; see below, "Selecting operating modes" section. Further, it is not permissible that the axis is in closed-loop control and it must be stationary (zero speed). In the SPEED_CONTROLLED operating mode, the axis only travels under closed-loop speed control with the specified velocity, ramp-up time and ramp-down time. The travel direction is derived from the polarity of the setpoint input.

Bit	Signal name	1 signal	0 signal	Remarks
14	SENSORLESS EMERGENCY operating mode	Yes	No	Control type: Closed-loop speed controlled without encoder or V/f characteristic This operating mode is selected only after operating mode interlocking; see below, "Selecting operating modes" section. Further, it is not permissible that the axis is in closed-loop control and it must be stationary (zero speed). In the SENSORLESS EMERGENCY operating mode, the axis travels with the control type "V/f open-loop control" or "closed-loop speed control without encoder" with a minimum velocity. The travel direction is derived from the polarity of the setpoint input. Note: For the V/f characteristic, parameter p1300 in the drive object (DO) must be set to parameterize the V/f control.
15	SWAYCONTROL operating mode	Yes	No	Control type: Speed-controlled with encoder This operating mode is selected only after operating mode interlocking; see below, "Selecting operating modes" section. Further, it is not permissible that the axis is in closed-loop control and it must be stationary (zero speed). In SWAYCONTROL operating mode, the velocity, ramp-up time, and ramp-down time are specified.

Bit 6: Selecting master-slave operation

Master-slave operation is selected with a high signal. This signal executes a leading / following drive function (rigid coupling).

The selection is described in detail in Chapter Master-slave operation control type (Page 253).

Bit 7: Selecting synchronous operation

Synchronous operation between the master and slave is selected using a high signal and both are then moved in speed or position synchronism (not a rigid coupling).

The selection is described in detail in Chapter Synchronous operation control type (Page 257).

Mode selection

Signal name	10	11	12	13	14	15
AUTOMATIC operating mode	1	0	0	0	0	0
MANUAL operating mode	0	1	0	0	0	0
EASY_POSITIONING operating mode	0	0	1	0	0	0
SPEED_CONTROLLED operating mode (= JOGGING)	0	0	0	1	0	0
SENSORLESS EMERGENCY operating mode	0	0	0	0	1	0
SWAYCONTROL operating mode	0	0	0	0	0	1

NOTICE

- It is only possible to switch over an operating mode if the plausibility check was successfully completed, the drive is not in closed-loop control and the drive is stationary (zero speed).
- When processing an AUTOMATIC request, the crane driver can interrupt the active request by operating the master switch (safety reasons!) and can continue to work with the master switch.

Bit 10: AUTOMATIC operating mode
Control type: Closed-loop position controlled with/without synchronous operation

In the AUTOMATIC operating mode, the axis travels closed-loop position-controlled to the specified position (with the specified velocity, ramp-up time and ramp-down time). The request is started with the specified values at standstill using the control bit "start automatic operation". If the master switch is deflected in the AUTOMATIC operating mode, the request is interrupted and the master switch specifies the velocity setpoint.

For further information on this operating mode refer to Chapter "Description of the operating modes (Page 224)".

Bit 11: MANUAL operating mode
Control type: Closed-loop position controlled with/without synchronous operation

In the MANUAL operating mode, the axis moves closed-loop position controlled. In so doing, the S7 specifies the target position, velocity, ramp-up time and ramp-down time. The travel direction is specified by the setpoint from the master switch.

Bit 12: EASY_POSITIONING operating mode
Control type: Closed-loop position controlled with/without synchronous operation

Position-controlled travel is used in the EASY_POSITIONING operating mode. The SIMATIC S7 so specifies the target position, velocity, ramp-up time and ramp-down time. The travel direction is specified by the setpoint from the master switch.

For further information on this operating mode refer to Chapter "Description of the operating modes (Page 224)".

Table 6- 6 Application_control_word_2_S7

Bit	Signal name	1 signal	0 signal	Remarks
0	Save and set the I component (start pulse)	Yes	No	Bit 0 is used to store and set the I component when method 2 is used for the start pulse (refer to Chapters DCC_StartPulse (Page 150) and DCC_StartPulse_1 (Page 154)). The high signal sets the saved I component; the low signal and saves the current I component.
1	Select velocity limitation, bit 1	Yes	No	Up to four different velocity limits can be selected using bit 1 and bit 2. If bit 1 and bit 2 are not selected, then the maximum velocity acts as limit.
2	Select velocity limitation, bit 2	Yes	No	

Bit	Signal name	1 signal	0 signal	Remarks
3	Select prelimit switch velocity	Yes	No	A high signal is used to change over to the configured prelimit switch velocity (e.g. 10% of the maximum velocity).
4	Enable slack rope controller	Yes	No	The slack rope controller is activated using a high signal and therefore the slack rope is wound up.
5	Command, save grab open	Yes	No	Using a high signal, the position actual value difference between Holding Gear and Closing Gear for grab open is saved.
6	Command, save grab closed	Yes	No	Using a high signal, the position actual value difference between Holding Gear and Closing Gear for grab closed is saved.
7	Select orange-peel bucket	Yes	No	Orange-peel bucket operation is activated using a high signal.
8	Enable current equalization controller	Yes	No	The current equalization controller is activated using a high signal.
9	Select grab change	Yes	No	The velocity setpoint of the holding gear is de-coupled from the closing gear with a high signal. This means that the ropes/cables can be adjusted independently of one another.
10	Homing	Yes	No	See below
11	Select torque limiting	Yes	No	The torque limiting is activated using a high signal and the drive torque is limited to the torque specified in the PZD8.
12	Select encoder switchover	Yes	No	The closed-loop position control is switched over from the motor encoder to an external encoder with a high signal. This switchover can only be made with the axis at a standstill (zero speed).
13	SlaveMode	Yes	No	With a low signal, the MasterMode is activated and with a high signal, the SlaveMode is activated. In MasterMode, the axis can be operated in all operating modes. In the SlaveMode, the axis can be operated in all operating modes with the exception of the AUTOMATIC operating mode. An AUTOMATIC request cannot be processed. The selection is accepted if the drive is not in closed-loop control and is stationary (zero speed). Example, refer to Description FB_OperationMode (Page 194)
14	BrakeTest	Yes	No	The brake test is selected using the high signal. The selection is accepted if the drive is not in closed-loop control and is stationary; see FB_OperationMode (Page 194).
15	changeTechnologyObject	Yes	No	This activates or deactivates the particular technology object. In order that the axis is active, a high signal must be available. For false, the axis is deactivated. For more detailed information see description of the FB_OperationMode (Page 194).

Bit 10: Homing

The axis is homed using a high signal. The axis can only be homed when it is at a standstill (zero speed). The reference point must be specified at the following input of function block "FB_OperationMode":

1. inHomingValuePLC_Bool = FALSE

   A fixed reference point (inHomingfixvalue_Lreal) is configured. The axis is set to this position by setting bit 10 of "Homing".

2. inHomingValuePLC_Bool = TRUE

   The SIMATIC S7 specifies a home position in the PZD for the target position (PZD 11 and PZD 12). The axis is set to this position by setting bit 10 of "Homing".

For additional information, refer to Chapter Homing (Page 252).

6.2.2.2 SIMOTION → SIMATIC S7

Table 6- 7 SIMOTION → S7

PZD	Signal name	Unit	Remarks
1	Status_word_1_S7	-	See below, Table "Status_word_1_S7"
2	Speed_actual_value_S7	as a % of the reference speed	The actual speed value is sent, referred to the reference value. The reference value is the maximum value. $n_{max} = n_{rated} + n_{Fw}$ = 100% 4000 hex = 16384 dec = 100%
3	Status_word_2_S7	-	See below, Table "Status_word_2_S7"
4	Fault and alarm numbers	-	The error messages should be interpreted as a UINT, 0 - 65535 See Application error messages and alarm messages (Page 339)
5	Current_actual_value_total_smoothed_S7	as a % of the reference current	The current actual value smoothed (r0068[1]) is sent, referred to the reference value. The reference value is the maximum value. $I_{max} = I_{rated} + I_{Overload}$ = 100% 4000 hex = 16384 dec = 100%
6	Load actual value_S7 / free	kg	The determined load actual value is sent (only for Hoist, e.g.: kg or tons). The determined load value is sent to the S7 from 0 to 65,535 (UINT). The normalization factor for the load current in physical units (kg) must be parameterized in function block "load-dependent field weakening".
7	Torque_actual_value_smoothed_S7	as a % of the reference torque	The current actual torque value smoothed (r0080[1]) is sent, referred to the reference value. The reference value is the maximum value. $M_{max} = M_{rated} + M_{Overload}$ = 100% 4000 hex = 16384 dec = 100%

PZD	Signal name	Unit	Remarks
8	Speed_setpoint_before_speed_controller_S7	as a % of the reference speed	The current speed setpoint before the speed controller (r0062) is sent, referred to the reference value. The reference value is the maximum value. $n_{max} = n_{rated} + n_{Fw} = 100\%$ 4000 hex = 16384 dec = 100%
9	Application_status_word_1_S7	-	See below, Table "Application_status_word_1_S7"
10	Application_status_word_2_S7		See below, Table "Application_status_word_2_S7"
11	Act. Position Val._S7	mm	The actual position (..positioningstate. actualposition) is sent as double word. The conversion from LREAL to DINT must be made in SIMOTION.
12			
13	Output voltage	As percentage of the reference voltage	The smoothed output voltage based on the reference value is sent. 4000 hex = 16384 dec = 100%
14	Torque_setpoint_S7 / open grab / acceleration setpoint / free	As percentage of the reference torque / 0 - 100%	The total torque setpoint (r0079) for the holding gear and the hoisting gear based on the reference value is sent. The torque setpoint is required for the brake test . The reference value is the maximum value. $M_{max}=M_{rated}+M_{overload}=100\%$ 4000 hex=16384 dec=100% / status message for the position for the closing gear gripper, 100% corresponds to grab open / acceleration setpoint of the trolley / free for user-engineered applications
15	-		Free for user-engineered applications
16	-		Free for user-engineered applications

PZD 4: Fault and alarm numbers

The fault and alarm numbers are sent. Only the positive value range is used which means that fault and alarm numbers from 0 to 65,535 (UINT) can be sent. The differentiation as to whether it involves a fault number from SIMOTION, SINAMICS or from a function block is displayed using bit 6 "SIMOTION", bit 7 "SINAMICS" and bit 8 "function block" in Application_status_word_2.

If no faults are present, the SINAMICS alarms are transferred to the S7 using the same process data item.

The following priority assignment is defined. Priority 1 has the highest priority and priority 4 the lowest priority:

Priority 1: SINAMICS error (AppZSW2 bit 7)
Priority 2: SIMOTION error (AppZSW2 bit 6)
Priority 3: Function block error (AppZSW2 bit 8)
Priority 4: SINAMICS alarm (ZSW1 bit 7)
Priority 5: Function block alarm (none of the above status bits)

Table 6- 8 Status_word_1_S7

Bit	Signal name	Remarks
0	Ready to switch-on	If a high signal is present at the bit, this indicates that the line contactor is OFF and switch-on (OFF1) is awaited
1	Ready	If a high signal is present at the bit, this indicates that voltage is present at the Line Module.
2	Enable operation	If a high signal is present at the bit, this indicates that the electronics and the pulses have been enabled and the axis has ramped up to the setpoint.
3	Fault present	If a high signal is present at the bit, this indicates that the drive has a fault and is therefore non-operational. After acknowledgement and the cause has been successfully removed, the drive goes into the power-up inhibit.
4	Coasting down active (OFF2)	If a high signal is present at the bit, this indicates that no OFF2 is active.
5	Fast stop active (OFF3)	If a high signal is present at the bit, this indicates that no OFF3 is active.
6	Power-on inhibit	If a high signal is present at the bit, this indicates that it is only possible to power up again using OFF1 followed by ON.
7	Alarm present	If a high signal is present at the bit, this indicates that the drive is still in operation and no acknowledgement is required.
8	Speed setpoint-actual value deviation in the tolerance range	If a high signal is present at the bit, this indicates that the actual value lies within the tolerance range.
9	Control requested to the PLC	If a high signal is present at the bit, this indicates that the automation system has been requested to accept control.
10	f or n comparison value reached or exceeded	If a high signal is present at the bit, this indicates that f or n comparison value has been reached or exceeded.
11	I, M or P limit reached or exceeded	If a high signal is present at the bit, this indicates that I, M or P limit has not been reached or was exceeded.
12	Holding brake open	If a high signal is present at the bit, this indicates that the holding brake has received the command to open.
13	Interpolator active (ramp-function generator active)	If a high signal is present at the bit, this indicates that the interpolator is active.
14	n_act >= 0	If a high signal is present at the bit, this indicates that the speed actual value is ≥ 0.
15	Alarm, drive converter thermal overload	If a high signal is present at the bit, this indicates that there is an alarm due to thermal overload of the converter.

Table 6- 9 Status_word_2_S7

Bit	Signal name	Remarks
0	Not used	The bit is not used.
1	Safe Torque Off active	A high signal indicates that the hardware triggered the Safe Torque Off state by means of input signals of the Control Unit or of the Motor Module.
2	TandemHoming active	The axis is moving in order to compensate for the offset between the tandem master and tandem slave. (Only in double-spreader operation)
3	TandemMode active	Tandem mode is activated. (Only for hoisting gear applications)

Bit	Signal name	Remarks
4	SlaveTandemMode active	Tandem-slave mode is activated. (Only for hoisting gear applications)
5	DriveMasterSuperimpo se	With a high signal, the tandem-master travels the offset; with a low signal, the tandem-slave travels the offset.
6	SelectMasterAxis2 active	These outputs indicate the active master. (Only for hoisting gear applications) 1. SelectMasterAxis2 = FALSE AND SelectMasterAxis3 = FALSE. This means the axis at TO_CheckbackMasterAxis is the master. 2. SelectMasterAxis2 = TRUE AND SelectMasterAxis3 = FALSE. This means the axis at TO_CheckbackMasterAxis2 is the master. 3. SelectMasterAxis2 = FALSE AND SelectMasterAxis3 = TRUE. This means the axis at TO_CheckbackMasterAxis3 is the master. 4. SelectMasterAxis2 = TRUE AND SelectMasterAxis3 = TRUE. This causes an error to be output.
7	SelectMasterAxis3 active	
8	CheckbackSlave active	The checkback signal of the slave (bocheckbackSlave) is activated and checkback signals are checked. (Only for hoisting gear applications)
9	CheckbackSlaveTande m active	The checkback signal of the tandem-slave (checkbackSlaveTandem) is activated and checkback signals are checked. (Only for hoisting gear applications)
10	CheckbackSlaveSlave Tandem active	The checkback signal of the tandem-slave-slave (checkbackSlaveSlaveTandem) is activated and checkback signals are checked. (Only for hoisting gear applications)
11	boOffsetHomingActive	The axis is moving in order to compensate the offset between master and slave.
12	boOffsetModeActive	Offset operation is active.
13 to 15	Free	These bits are not used.

Table 6- 10 Application_status_word_1_S7

Bit	Signal name	Remarks
0	Axis moves in the positive direction	With a high signal, the axis moves in the positive direction (...motionstatedata.commandvelocity ≥ 0)
1	Axis moves in the negative direction	With a high signal, the axis moves in the negative direction (...motionstatedata.commandvelocity < 0)
2	Message, drive stationary	With a high signal, the drive is stationary, i.e. the velocity = 0.
3	Message, current distribution monitoring responded	With a high signal, the current distribution monitoring has responded.
4	Message, field weakening enabled	With a high signal, the field weakening is enabled.
5	Message, AUTOMATIC operating mode, target position reached	With a high signal, the target position has been reached in the AUTOMATIC operating mode.
6	Message, master-slave operation active	With a high signal, master-slave operation is active.

Bit	Signal name	Remarks
7	Message, synchronous operation active	With a high signal, synchronous operation between the master and slave is active.
8	Message, offset active	With a high signal, an offset for synchronous operation between master and slave is active.
9	Message, AUTOMATIC request active	With a high signal, a request in the AUTOMATIC operating mode is active.
10	AUTOMATIC operating mode	A high signal is used to signal back that the AUTOMATIC operating mode is active.
11	MANUAL operating mode	A high signal is used to signal back that the MANUAL operating mode is active.
12	EASY_POSITIONING operating mode	A high signal is used to signal back that the EASY_POSITIONING operating mode is active.
13	SPEED_CONTROLLED operating mode (= JOGGING)	A high signal is used to signal back that the SPEED_CONTROLLED (= JOGGING) operating mode is active.
14	SENSORLESS EMERGENCY operating mode	A high signal is used to signal back that the SENSORLESS EMERGENCY operating mode is active.
15	SWAYCONTROL operating mode	A high signal is used to signal back that SWAYCONTROL operating mode is active.

Table 6- 11 Application_status_word_2_S7

Bit	Signal name	Remarks
0	Closed-loop torque control active (H) or grab ½ open (H)	"Grab ½ open" is needed only for the Closing Gear, otherwise "Closed-loop torque control active (H)"
1	Closed-loop speed control active	A high signal is used to signal back that the closed-loop speed control is active.
2	Closed-loop position control active (H) or grab 2/3 closed (H)	"Grab 2/3 closed (H)" is needed only for the Closing Gear, otherwise "Closed-loop position control active (H)"
3	Message a/v reduction	With a high signal, acceleration, deceleration or velocity have been reduced. Once detected, the signal remains down to standstill (zero speed).
4	Message, grab open	With a high signal, the grab is open.
5	Message, grab closed	With a high signal, the grab is closed.
6	SIMOTION fault	With a high signal, there is a fault/error present in SIMOTION at the technology object (priority 2).
7	Fault, SINAMICS	With a high signal, a fault/error is active in the SINAMICS drive object (priority 1).
8	Fault, function block	With a high signal, a fault/error is active at a function block (priority 3).
9	Message torque limiting active (H) or message torque limit exceeded (H)	"Message torque limit exceeded (H)" is needed only for the Closing Gear, otherwise "Message torque limiting active (H)"
10	Message, homed	With a high signal, the axis is homed, i.e. the homing position was accepted.

Bit	Signal name	Remarks
11	Position difference outside tolerance	The high signal indicates that the position difference between the master and slave is outside the tolerance limits.
12	Encoder switchover active	With a high signal, the encoder data set switchover has become active. The closed-loop position control now uses an external encoder and no longer the motor encoder.
13	SlaveModeActive	With a low signal, the MasterMode is active and with a high signal, the SlaveMode is active.
14	BrakeTestActive ... (H) or message "Grab touchdown" (H)	With a high signal, the BrakeTestActive is active or only for the Holding Gear with continuous load measurement the message "Grab touchdown (H)" is output.
15	TechnologyObjectActiv e (H)	With a high signal, the TechnologyObjectActive is active.

Bit 0: Closed-loop torque control active or grab 1/2 open
A high signal is used to signal back that closed-loop torque control is active. For a Closing Gear, the checkback signal "Closed-loop torque control active" is not returned, but the "Grab 1/2 open" instead.

Bit 2: Closed-loop position control active or grab 2/3 closed
A high signal is used to signal back that closed-loop position control is active. For a Closing Gear, the checkback signal "Closed-loop position control active" is not returned, but "Grab 2/3 closed" instead.

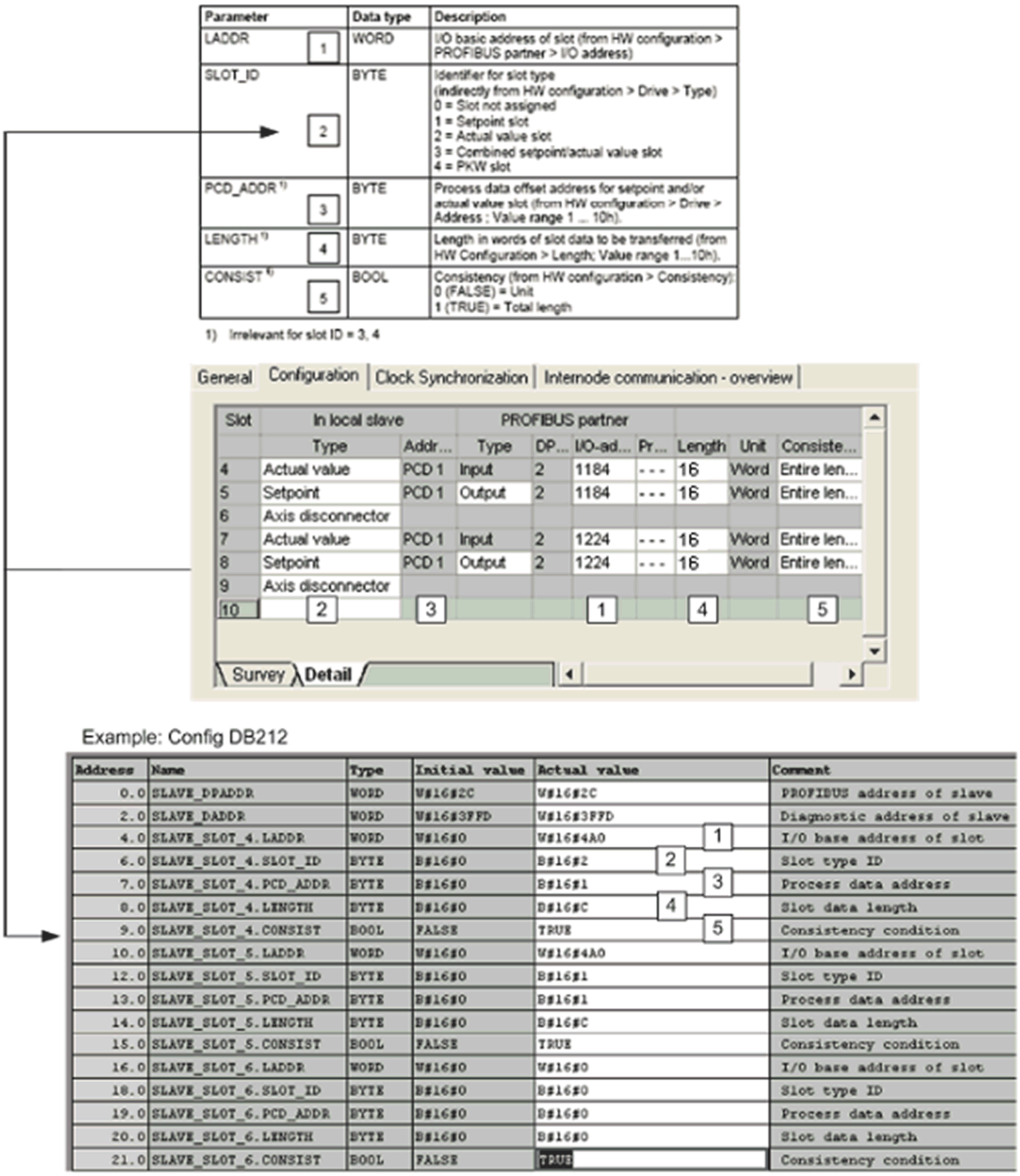
Bit 9: Message Torque limiting active or message Torque limit exceeded
Torque limiting is active with the high signal. For a Closing Gear, the checkback signal "Message Torque limiting active" is not returned, but "Message torque limit exceeded" instead.

6.2.3 Short description of the data blocks and function blocks in SIMATIC for communication

The blocks described below are used to control communication between SIMATIC and SIMOTION D.

Function	Name	Description
-FC1533	D_TOD_DTa	Date and TIME_OF_DAY convert to DATE_AND_TIME data format.
-FC1536	DT _DATEa	Form Date data format from the DATE_AND_ TIME format.
-FC1538	DT to TODa	Form TIME_OF_DAY data format from the DATE_AND_TIME format.
-FC1564	SD_DT_DTa	Subtraction of two DT values.
-FB1701	FB PCD_SEND	The block cyclically transfers the process data (control words, setpoints) from SIMATIC to the drive in accordance with consistency conditions.
-FB1702	FB PCD_RECV	The block cyclically transfers the process data (status words, actual values) from the drive to SIMATIC in accordance with consistency conditions.

Function	Name	Description
-FC1707	FC COM_STAT	The block calls system function SFC 51 RDSYSST to evaluate the SSL of the CPU and reports faulty or deactivated states of the slave to be processed.
-FB1720	SINAMICS_CY	The block manages cyclic communication between SIMATIC and a SIMOTION/SINAMICS drive. Block functions: • Sending process data (control words and setpoints) to the drive • Receiving process data (status words and actual values) from the drive • Checking the communication state; the block evaluates the SSL of the CPU and reports any faulty or deactivated states of the slave Cyclic communication is handled by standard SIMATIC-Drive-ES function blocks: FB1701 "PCD_SEND", FB1702 "PCD_RECV", FC1707 "COM_STAT"
-FC1732	Control word 1 SINAMICS/SIMOTION	FC1732 writes the value of the first control word of a SINAMICS or SIMOTION Control Unit.
-FC1733	Status word 1 SINAMICS/SIMOTION	FC1733 reads the value of the first status word of a SINAMICS or SIMOTION Control Unit.
-FC1734	Control word 2 SINAMICS/SIMOTION	FC1734 writes the value of the second control word of a SINAMICS or SIMOTION Control Unit.
-FC1735	Status word 2 SINAMICS/SIMOTION	FC1735 reads the value of the second status word of a SINAMICS or SIMOTION Control Unit.
-FC1736	Control word 1 Technology	FC1736 writes the value of the first application control word of a SIMOTION Control Unit.
-FC1737	Status word 1 Technology	FC1737 reads the value of the first application status word of a SIMOTION Control Unit.
-FC1738	Control word 2 Technology	FC1738 writes the value of the second application control word of a SIMOTION Control Unit.
-FC1739	Status word 2 Technology	FC1739 reads the value of the second application status word of a SIMOTION Control Unit.
-FC1740	TimeDateSIMOTION	FC1740 has the function to set a date and time in the SIMOTION from the comparison of the DT values of the PLC and SIMOTION.
-UDT31	User-defined data type SLOT_UDT	Predefined (user-defined) data type; used in FBs and DBs.
-DBxx2	Configuration data block	DBxx2 is the configuration data block within PROFIBUS, diagnostics, I/O addresses and slot type data. Each drive component must be assigned a separate configuration DB, e.g. DB212(HO-1), DB222(HO-2). These DBs contains the data structure shown in the diagram below.
-DBxx3	Replacement process DB	DBxx3 is multiple instance DB of FB1720 and is used for cyclic process data communication. Each drive component must be assigned a separate cyclic communication DB, e.g. DB213(HO-1), DB223(HO-2). These DBs contains the data structure described in Defining the telegram (Page 306). Data block DB213 for Hoist-1 contains, for example, Send process data: DB213 - DW36...DW66 Receive process data: DB213 - DW78...DW108
_DB4	Time data block	DB4 is used to store the time and date and is called in the OB1
DB21x = HO1; DB22x = HO2; DB31x = TR1; DB32x = TR2; DB41x = BH1; DB42x = BH2;DB51x = GA1; DB52x = GA2; ...		



Parameter	Data type	Description
LADDR	WORD	I/O basic address of slot (from HW configuration > PROFIBUS partner > I/O address)
SLOT_ID	BYTE	Identifier for slot type (indirectly from HW configuration > Drive > Type) 0 = Slot not assigned 1 = Setpoint slot 2 = Actual value slot 3 = Combined setpoint/actual value slot 4 = PKW slot
PCD_ADDR [1]	BYTE	Process data offset address for setpoint and/or actual value slot (from HW configuration > Drive > Address ; Value range 1 ... 10h).
LENGTH [1]	BYTE	Length in words of slot data to be transferred (from HW Configuration > Length; Value range 1...10h).
CONSIST [1]	BOOL	Consistency (from HW configuration > Consistency): 0 (FALSE) = Unit 1 (TRUE) = Total length

[1] Irrelevant for slot ID = 3, 4

Slot	In local slave		PROFIBUS partner						
	Type	Addr...	Type	DP...	I/O-ad...	Pr...	Length	Unit	Consiste...
4	Actual value	PCD 1	Input	2	1184	- - -	16	Word	Entire len...
5	Setpoint	PCD 1	Output	2	1184	- - -	16	Word	Entire len...
6	Axis disconnector								
7	Actual value	PCD 1	Input	2	1224	- - -	16	Word	Entire len...
8	Setpoint	PCD 1	Output	2	1224	- - -	16	Word	Entire len...
9	Axis disconnector								
10									

Example: Config DB212

Address	Name	Type	Initial value	Actual value	Comment
0.0	SLAVE_DPADDR	WORD	W#16#2C	W#16#2C	PROFIBUS address of slave
2.0	SLAVE_DADDR	WORD	W#16#3FFD	W#16#3FFD	Diagnostic address of slave
4.0	SLAVE_SLOT_4.LADDR	WORD	W#16#0	W#16#4A0	I/O base address of slot
6.0	SLAVE_SLOT_4.SLOT_ID	BYTE	B#16#0	B#16#2	Slot type ID
7.0	SLAVE_SLOT_4.PCD_ADDR	BYTE	B#16#0	B#16#1	Process data address
8.0	SLAVE_SLOT_4.LENGTH	BYTE	B#16#0	B#16#C	Slot data length
9.0	SLAVE_SLOT_4.CONSIST	BOOL	FALSE	TRUE	Consistency condition
10.0	SLAVE_SLOT_5.LADDR	WORD	W#16#0	W#16#4A0	I/O base address of slot
12.0	SLAVE_SLOT_5.SLOT_ID	BYTE	B#16#0	B#16#1	Slot type ID
13.0	SLAVE_SLOT_5.PCD_ADDR	BYTE	B#16#0	B#16#1	Process data address
14.0	SLAVE_SLOT_5.LENGTH	BYTE	B#16#0	B#16#C	Slot data length
15.0	SLAVE_SLOT_5.CONSIST	BOOL	FALSE	TRUE	Consistency condition
16.0	SLAVE_SLOT_6.LADDR	WORD	W#16#0	W#16#0	I/O base address of slot
18.0	SLAVE_SLOT_6.SLOT_ID	BYTE	B#16#0	B#16#0	Slot type ID
19.0	SLAVE_SLOT_6.PCD_ADDR	BYTE	B#16#0	B#16#0	Process data address
20.0	SLAVE_SLOT_6.LENGTH	BYTE	B#16#0	B#16#0	Slot data length
21.0	SLAVE_SLOT_6.CONSIST	BOOL	FALSE	TRUE	Consistency condition

Slot_4 = Actual Value Slot
Slot_5 = Setpoint Value Slot

Figure 6-15 Correlation between the SW Config and HW Config DBs

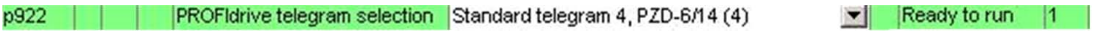
6.3 Communication, SIMOTION D – SINAMICS (internal SIMOTION D communication)

6.3.1 Configuring the connection

Communication between SIMOTION and SINAMICS is implemented using PROFIBUS Integrated via DRIVE-CLiQ.

The PROFIBUS data exchange rate is 12 Mbit/s. This means that an equidistant DP cycle is obtained from 1 ms onwards. The equidistant DP cycle depends on the number of connected nodes (devices).

Communication between SIMOTION and SINAMICS is configured using standard telegram 4 (6 process data items on the send side / 14 process data items on the receive side). Then, this standard telegram is extended to a free telegram (999) with 32 process data items on the send and receive sides.

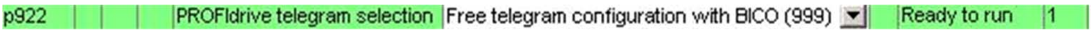


p922				PROFIdrive telegram selection	Standard telegram 4, PZD-6/14 (4)	Ready to run	1

Figure 6-16 Parameter p0922 to value 4

Note

If parameter p0922 has already been set to a free telegram (999) during configuration, it needs to be set to standard telegram 4 (4) before the script for communication interconnections is run (see below), and then reset to a free telegram (999) again afterwards. This ensures that the correct values are assigned to the data words at the send and receive sides of the telegram.



p922				PROFIdrive telegram selection	Free telegram configuration with BICO (999)	Ready to run	1

Figure 6-17 Parameter p0922 to value 999

The address ranges for communications between SIMOTION and SINAMICS should be configured as follows. These addresses can be adapted in the hardware configuration of SIMOTION.

Drives	I/O address area
Hoist 1 / Holding Gear 1	300 … 363
Hoist 2 / Closing Gear 1	364 … 427
Trolley 1/Boom 1/Luffing Gear 1	428 … 491
Trolley 2 / Boom 2	492 … 555
Gantry 1	556 … 619
Gantry 2	620 … 683
Slewing Gear 1	684 … 747
Slewing Gear 2	748 … 811
Hoist 3/Holding Gear 2	812 … 875
Hoist 4 / Closing Gear 2	876 … 939

The following additional information is transferred in the supplementary process data words:

1. Extension of the telegram SIMOTION → SINAMICS

PZD	Signal name	Module
7	Supplementary speed setpoint (start pulse)	Hoist, Trolley, Boom, Holding Gear, Closing Gear, Luffing Gear
8	Torque_limiting_upper	All
9	Torque_limiting_lower	All
10	Speed controller integrator setting value	All
11	Control word for integrator setting value in the speed controller	All

2. Extension of the telegram SINAMICS → SIMOTION

PZD	Signal name	Module
15	Torque-generating current setpoint	All
16	Total_current_actual_value_smoothed	All
17	Torque_actual_value_smoothed	All
18	Speed_controller_setpoint_actual_value_difference	All
19	Speed setpoint	All
20	Output voltage	All
21	Actual fault code	All
22	Actual alarm code	All
23	Total torque setpoint	All
24	Speed controller I torque output	All
25	Actual power value	All

Every standard application makes available scripts that can be used to automatically execute all the communication interconnections in SINAMICS. The particular script for the drive is located in the drive object under the script folder.

Note

For communication, see General (Page 361), "Script files" section.

6.3.2 Defining the telegram

Communication between SIMOTION D and SINAMICS units is established using integrated PROFIBUS DP with motion control expansion. The integrated DP must be configured for clock-cycle synchronous operation. The standard telegram 4 according to PROFIdrive profile V4.0 (/Ref.1/) is used and appropriately expanded up to max. 32 PZD.

Two function blocks (FB_TelegramSimotionToSinamics, FB_TelegramSinamicsToSimotion) to process communication telegrams (or interconnect and convert signals) between SIMOTION and SINAMICS is programmed in the basic technology, refer to Description of the function blocks (Page 163).

6.3.2.1 SIMOTION → SINAMICS

Table 6- 12 SIMOTION → SINAMICS

PZD	Signal name	Remarks
1	Control_word_1_SINAMICS	refer to the following table
2	Speed_setpoint_SINAMICS	The Speed_setpoint_SINAMICS is received as double word referred to the reference value. The double word assignment is a standard definition according to the PROFIdrive profile Standard. The internal interface between SIMOTION and SINAMICS was standardized according to the Standard. /Ref.1/
3		
4	Control_word_2_SINAMICS	Refer below, Table "Control_word_2_SINAMICS"
5	Control_word_encoder_1_SINAMICS	The Control_word_encoder_1_SINAMICS is described in the PROFIdrive profile. /Ref.1/
6	Control_word_encoder_2_SINAMICS	The Control_word_encoder_2_SINAMICS is described in the PROFIdrive profile. /Ref. 1/
7	Supplementary_speed_setpoint_SINAMICS	The supplementary speed setpoint is required for the start pulse for Hoist.
8	Torque_limiting_upper_SINAMICS	The torque can be used with the holding gear for the slack rope control. The required torque limitations can also be implemented using this channel.
9	Torque_limiting_lower_SINAMICS	
10	Setpoint I component speed controller	The torque setpoint for the I component of the speed controller can be specified here, e.g. from DCC block Startpulse, if the enableBrakeStoreValue option has been selected.
11	STW for speed controller	This control word transports the bit for setting the I component of the speed controller.
12-32	Not used	The word is not used.

Table 6- 13 Control_word_1_SINAMICS

Bit	Signal name	Remarks
0	On / Off1	The drive is powered up and powered down using On / Off1. p0840 = r2090.0
1	Off2	Off2 must be set - otherwise the drive cannot be powered up. If input Off2 is withdrawn when the drive is powered up, the pulses must be inhibited and the motor coasts down. p0844 = r2090.1
2	Off3	Off3 must be set - otherwise the drive cannot be powered up. If input Off3 is withdrawn when the drive is powered up, the motor is stopped with the configured down ramp and is then powered down. p848 = r2090.2
3	Pulse enable	The pulses are only enabled by setting input "pulse enable". p852 = r2090.3
4	Enable ramp function generator	The speed setpoint is only transferred to the speed controller after input "Enable ramp-function generator" has been set. p1140 = r2090.4
5	Reserved	
6	Enable setpoint	The speed setpoint is only transferred to the ramp-function generator after input "Enable setpoint" has been set. p1142 = r2090.6
7	Acknowledge fault	When the input "Acknowledge fault" is set, an existing fault is acknowledged if the cause of the fault has been removed. p2103 = r2090.7
8	Enable speed controller	The speed controller is enabled by setting the bit. p856 = r2090.8
9	Open brake	The "Open brake" command can be set in SINAMICS using the high signal. This can be used if an extended brake control is implemented via SINAMICS. p1224[1] = r2090.
10	Master control by PLC	This bit "Master control by PLC" must be set, otherwise the control does not accept any commands. p0854 = r2090.10
11	Reserved	
12	Select closed-loop torque control	Closed-loop speed control = 0; closed-loop torque control = 1 If bit 12 is not set, then the closed-loop speed control is active. If bit 12 is set, then the closed-loop torque control is active. p1501 = r2090.12
13-14	Reserved	
15	Intelligent overspeed signal	The bit "Intelligent overspeed signal" is only used for Hoist. In this case, the actual velocity is monitored for an overspeed condition. This bit is set if an overspeed condition is identified. Only for Hoist, ControlUnit P738 = MotorModule r2090.15

Note

The control in SIMOTION for bits 0 to 6 in control word 1 must be implemented using the command _enableAxis. Bits 8, 9, 11 to 15 must be controlled using the command _setaxisstw.

Table 6- 14 Control_word_2_SINAMICS

Bit	Signal name	Remarks
0	Drive data set selection DDS, bit 0	By selecting drive data set, it is possible to toggle between different drive data sets by controlling bits "DDS bit 0", "DDS bit 1" and "DDS bit 2". p0820 = r2093.0; p0821 = r2093.1; p0822 = r2093.2
1	Drive data set selection DDS, bit 1	
2	Drive data set selection DDS, bit 2	
3 to 6	Free	These bits are not used.
7	Acknowledge SINAMICS fault	By setting the input "Acknowledge SINAMICS fault", an existing fault is acknowledged if the cause of the fault was removed at the SINAMICS drive object. p2103 = r2093.7
8 to 11	Free	These bits are not used.
12	Master sign-of-life bit 0	Bits 12 to 15 are used as sign-of-life of the clock-cycle synchronous PROFIBUS master.
13	Master sign-of-life bit 1	
14	Master sign-of-life bit 2	
15	Master sign-of-life bit 3	

Note

The control in SIMOTION for bits 0 to 11 in control word 2 must be implemented using the command _setaxisstw.

Table 6- 15 Control word for speed controller

Bit	Signal name	Remarks
0	Enable motor blocking monitoring	The bit is used for disabling the blocking monitoring. A TRUE signal disables the monitoring. This is required for the brake test and can also be used for the slack rope control.
1	Speed controller Set integrator value	Bit for setting the I component of the speed controller. p1477 = r2094.1
2 to 15	Free	These bits are not used.

6.3.2.2 SINAMICS → SIMOTION

Table 6- 16 SINAMICS → SIMOTION

PZD	Signal name	Remarks
1	Status_word_1_SINAMICS	Refer to the table under "Status_word__1_SINAMICS" p2051.0 = r2089[0]
2	Speed_actual_value_SINAMICS	The speed actual value is sent as double word referred to the reference value. p2061.1 = r0063
3		
4	Status_word_2_SINAMICS	Refer to the table under "Status_word_2_SINAMICS" p2051.3 = r2089[1]
5	Status_word_encoder_1_SINAMICS	The status_word_encoder_1 is described in the PROFIdrive profile. /Ref.1/ p2051.4 = r0481[0]
6	Position_actual_value_1_encoder_1_SINAMICS	The position_actual_value_1_encoder_1 is described in the PROFIdrive profile. /Ref.1/ p2061.5 = r0482[0]
7		
8	Position_actual_value_2_encoder_1_SINAMICS	The position_actual_value_2_encoder_1 is described in the PROFIdrive profile. /Ref.1/ p2061.7 = r0483[0]
9		
10	Status_word_encoder_2_SINAMICS	The status_word_encoder_2 is described in the PROFIdrive profile. /Ref.1/ p2051.9 = r0481[1]
11	Position_actual_value_1_encoder_2_SINAMICS	The position_actual_value_1_encoder_2 is described in the PROFIdrive profile. /Ref.1/ p2061.10 = r0482[1]
12		
13	Position_actual_value_2_encoder_2_SINAMICS	The position_actual_value_2_encoder_2 is described in the PROFIdrive profile. /Ref.1/ p2061.12 = r0483[1]
14		
15	Current_setpoint_torque_generating_SINAMICS	The torque-generating current setpoint is sent as word, referred to the reference value. p2051.14 = r0077
16	Current_actual_value_total_smoothed_SINAMICS	The torque-generating and field-generating current actual value is smoothed (absolute current actual value) and sent as word, referred to the reference value. p2051.15 = r0068[1]
17	Torque_actual_value_smoothed_SINAMICS	The torque actual value is smoothed and sent as word, referred to the reference value. p2051.16 = r0080[1]
18	Setpoint-actual value difference_speed_controller_SINAMICS	The speed control difference at the speed controller is sent as word, referred to the reference value. p2051.17 = r0064
19	Speed_setpoint_SINAMICS	The speed setpoint is smoothed and sent as word, referred to the reference value. p2051.18 = r0062
20	Output voltage	The output voltage is smoothed and sent as word, referred to the reference value. p2051.19 = r0072
21	Actual fault code from SINAMICS	The actual fault code is sent as word. p2051.20 = r2131

PZD	Signal name	Remarks
22	Actual alarm code from SINAMICS	The actual alarm code is sent as word. p2051.21 = r2132
23	Torque setpoint	The torque setpoint is sent as word, referred to the reference value (p2003), as a % of the reference quantity (p2051.22 = r0079)
24	I component speed controller	The I component (torque setpoint) speed controller is smoothed and sent as word, referred to the reference value. p2051.23 = r1482
25	Actual power value	The torque actual value is smoothed and sent as word based on the reference value. p2051.24=r32
25-32	Free	These bits are not used.

PZD 23: Torque_setpoint_SINAMICS

The torque setpoint is sent as word, referred to the reference value (p2003).

Table 6- 17 Status_word_1_SINAMICS

Bit	Signal name	Remarks
0	Ready to switch-on	If a high signal is present at the bit, this indicates that the power supply is switched on and the electronics has been initialized. p2080.0 = r0899.0
1	Ready	If a high signal is present at the bit, this indicates that voltage is present at the Line Module. p2080.1 = r0899.1
2	Enable operation	If a high signal is present at the bit, this indicates that the electronics and the pulses have been enabled and the axis has ramped up to the setpoint. p2080.2 = r0899.2
3	Fault present	If a high signal is present at the bit, this indicates that the drive has a fault and is therefore non-operational. After acknowledgement and the cause has been successfully removed, the drive goes into the power-up inhibit. p2080.3 = r2193.3
4	Coasting down active (OFF2)	If a high signal is present at the bit, this indicates that no OFF2 is active. p2080.4 = r0899.4
5	Fast stop active (OFF3)	If a high signal is present at the bit, this indicates that no OFF3 is active. p2080.5 = r0899.5
6	Power-on inhibit	If a high signal is present at the bit, this indicates that it is only possible to power up again using OFF1 followed by ON. p2080.6 = r0899.6
7	Alarm present	If a high signal is present at the bit, this indicates that the drive is still in operation and no acknowledgement is required. p2080.7 = r2139.7
8	Speed setpoint-actual value deviation in the tolerance range	If a high signal is present at the bit, this indicates that the actual value lies within the tolerance range. p2080.8 = r2197.7

Bit	Signal name	Remarks
9	Control requested to the PLC	If a high signal is present at the bit, this indicates that the automation system has been requested to accept control. p2080.9 = r0899.9
10	f or n comparison value reached or exceeded	If a high signal is present at the bit, this indicates that f or n comparison value has been reached or exceeded. p2080.10 = r2199.1
11	I, M or P limit reached or exceeded	If a high signal is present at the bit, this indicates that the I, M, or P limit has been reached or exceeded. p2080.11 = r1407.7
12	Holding brake open	If a high signal is present at the bit, this indicates that the holding brake has received the "Open" command. p2080.12 = r0899.12
13	Alarm overtemperature motor	If a high signal is present at the bit, this indicates that the motor is signaling an overtemperature condition. p2080.13 = r2135.14
14	n_act >= 0	If a high signal is present at the bit, this indicates that the speed actual value is >=0. p2080.14 = r2197.3
15	Alarm, drive converter thermal overload	If a high signal is present at the bit, this indicates that there is an alarm due to thermal overload of the converter. p2080.15 = r2135.15

Table 6- 18 Status_word_2_SINAMICS

Bit	Signal name	Remarks
0	Drive data set DDS effective, bit 0	The effective drive data set bit 0 is signaled back using the high signal. p2081.0 = r0051.0
1	Drive data set DDS effective, bit 1	The effective drive data set bit 1 is signaled back using the high signal. p2081.1 = r0051.1
2	Drive data set DDS effective, bit 2	The effective drive data set bit 1 is signaled back using the high signal. p2081.2 = r0051.2
3 to 4	Not used	The bit is not used.
5	Safe Torque Off Control Unit active	The high signal indicates that Safe Torque Off (was: Safe Standstill) was initiated per hardware via the input signals of the Control Unit. p2081.5 = r9772.1
6	Safe Torque Off MotorModule active	The high signal indicates that Safe Torque Off (was: Safe Standstill) was initiated per hardware via the input signals of the Motor Module. p2081.6 = r9872.1
7	V/f control active	A high signal is used to signal back that the V/f control is active. p2081.8 = r1407.0
8	Encoderless operation active	A high signal is used to signal back that the closed-loop speed control is active. p2081.8 = r1407.1
9	Closed-loop torque control active	A high signal is used to signal back that the closed-loop torque control operating mode is active. p2081.9 = r1407.2
10	Closed-loop speed control active	A high signal is used to signal back that the closed-loop speed control is active. p2081.10 = r1407.3
11	Pulses enabled	A high signal is used to signal back that the pulses are enabled. p2081.11 = r899.11
12	Slave sign-of-life bit 0	A high signal is used to signal back the clock-cycle synchronous slave sign-of-life.
13	Slave sign-of-life bit 1	
14	Slave sign-of-life bit 2	
15	Slave sign-of-life bit 3	

Alarm, error and system messages 7

The monitoring functions for following error, positioning, and standstill are handled by the technology object in SIMOTION. Overspeed can be monitored by the drive object in SINAMICS (except for Hoist axes). A list of the messages can be found in Section Application error messages and alarm messages (Page 339).

The individual monitoring functions are described below; see also Ref. [2], Chapter 2.10 for more information. Information on the SIMOTION technology object alarms can be found in Ref. [3]. Information on the SINAMICS drive object alarms can be found in Ref. [4].

7.1 Overview - monitoring functions

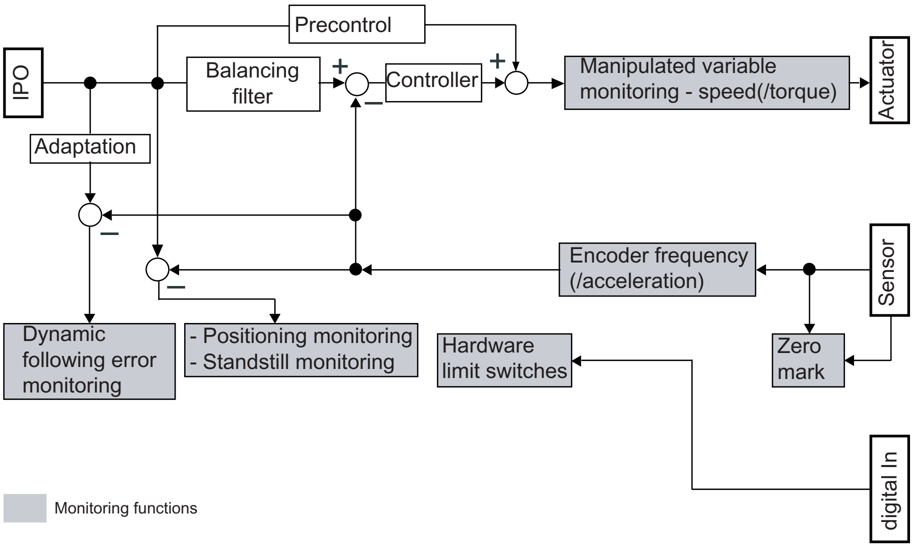


Figure 7-1 Block diagram of positioning axis monitoring functions

7.2 Dynamic monitoring of following errors

The following error monitoring on the position-controlled axis is performed on the basis of the calculated following error.

The permissible following error is dependent on the set velocity.

At velocities less than the specifiable minimum velocity, a programmable constant following error is monitored. Above this limit, the following error is increased proportionally to the set velocity up to a maximum difference. The maximum difference is reached at the maximum velocity.

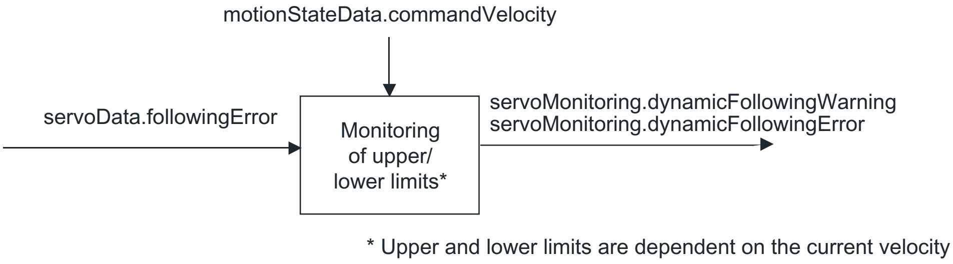
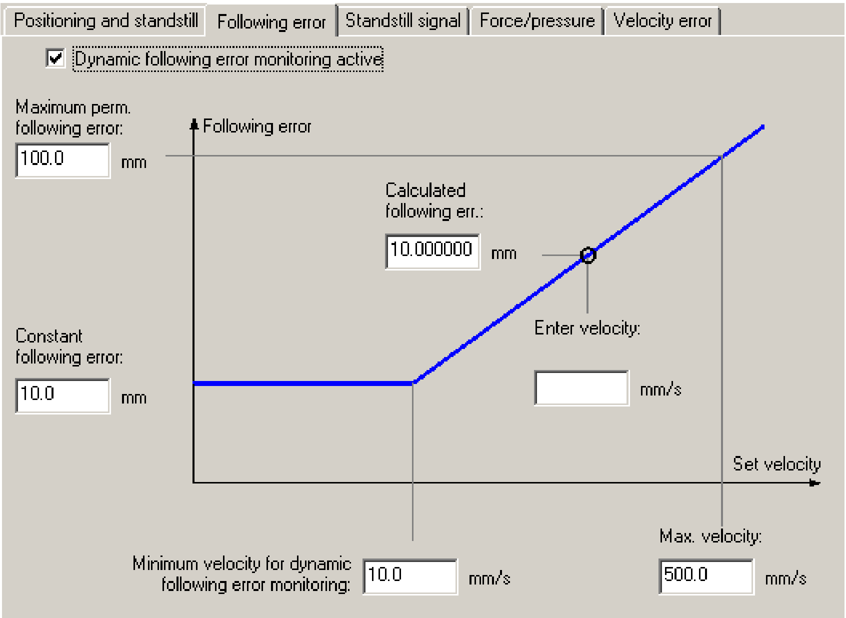

motionStateData.commandVelocity
servoData.followingError
Monitoring of upper/ lower limits*
servoMonitoring.dynamicFollowingWarning
servoMonitoring.dynamicFollowingError
* Upper and lower limits are dependent on the current velocity

Figure 7-2 Dynamic monitoring of following errors



☑ Dynamic following error monitoring active

Maximum perm. following error: 100.0 mm

Calculated following err.: 10.000000 mm

Enter velocity: mm/s

Constant following error: 10.0 mm

Minimum velocity for dynamic following error monitoring: 10.0 mm/s

Max. velocity: 500.0 mm/s

Figure 7-3 Function and parameters for following error monitoring

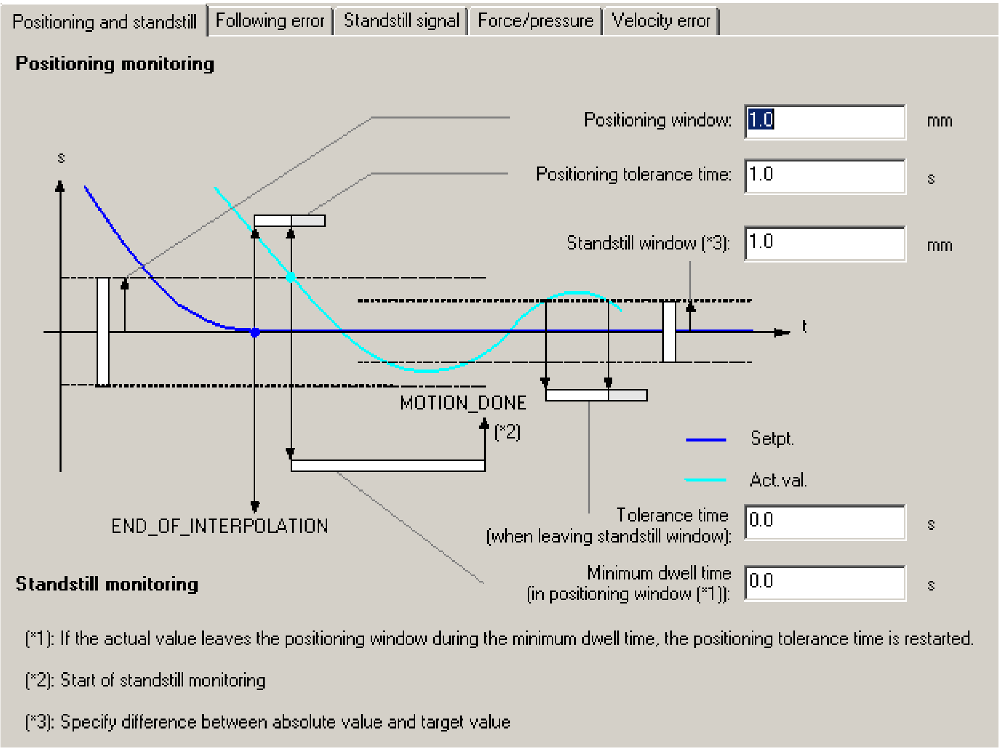
Determination of following errors

The following error is determined from the difference between the non-symmetrical setpoint prior to inclusion of the dynamic response adjustment and the actual value present in the control.

Therefore, the transfer times of the setpoint to the drive and the actual value to the control are included in the following error.

The following error must be determined after the speed controller and the positioning controller have been commissioned. This is because any changes to these two controllers can have an influence on the possible following error.

7.3 Position and standstill monitoring



Positioning monitoring

Positioning window: 1.0 mm

Positioning tolerance time: 1.0 s

Standstill window (*3): 1.0 mm

Tolerance time (when leaving standstill window): 0.0 s

Standstill monitoring

Minimum dwell time (in positioning window (*1)): 0.0 s

(*1): If the actual value leaves the positioning window during the minimum dwell time, the positioning tolerance time is restarted.

(*2): Start of standstill monitoring

(*3): Specify difference between absolute value and target value

Figure 7-4 Function and parameters for position monitoring

Position monitoring

The behavior of the actual position at the end of the setpoint interpolation is monitored.

This position-related monitoring does not distinguish whether the setpoint interpolation is ended as a result of reaching the target position from the setpoint side or due to a position-controlled stop during the motion performed by the interpolator (e.g., with a _stop() command). This monitoring is called position-related monitoring, although the position does not have to be the same as the target position. The tolerance window specified for this monitoring is called the positioning window.

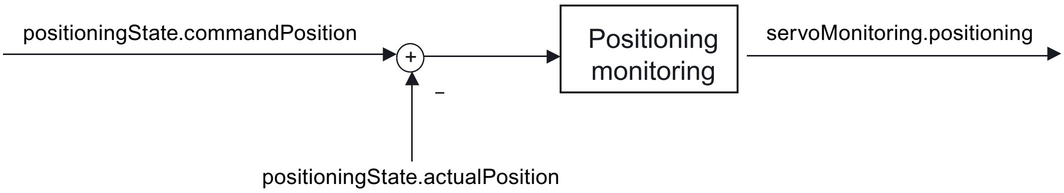

positioningState.commandPosition
+
–
Positioning monitoring
servoMonitoring.positioning
positioningState.actualPosition

Figure 7-5 Position monitoring

Sequence:

- At the end of the setpoint interpolation, the monitoring time **typeOfAxis.positionMonitoring.posWinTolTime** is started.
- If, before this time expires, the actual value reaches a definable window **typeOfAxis.positionMonitoring.tolerance** around the existing position at the end of the setpoint interpolation, monitoring of the minimum dwell time **typeOfAxis.positionMonitoring.posWinTolDelayTime** is initiated. The window is specified as a deviation in **typeOfAxis.standStillMonitoring.stillStandTolerance**, i.e. half of the window width is set.

  If the actual value does not reach the window within the monitoring time **typeOfAxis.positionMonitoring.posWinTolTime**, alarm 50106 (position monitoring) is issued.
- If the actual value leaves this window again during the minimum dwell time **typeOfAxis.positionMonitoring.posWinTolDelayTime**, the monitoring time **typeOfAxis.positionMonitoring.posWinTolTime** is restarted; each time it re-enters this window, the minimum dwell time **typeOfAxis.positionMonitoring.posWinTolDelayTime** is restarted.
- If the actual value remains in this window for the minimum dwell time **typeOfAxis.positionMonitoring.posWinTolDelayTime**, the MOTION_DONE status is set in the system variable **motionStateData.motionCommand** and the standstill monitoring is started.

The monitoring phases are also displayed in **servoMonitoring.positioningState** (V4.1.1 and higher):

- ACTUAL_VALUE_OUT_OF_POSITIONING_WINDOW = setpoint interpolation has been completed; actual value has not yet reached the positioning window
- ACTUAL_VALUE_INSIDE_POSITIONING_WINDOW = actual value is inside positioning window; standstill monitoring is not yet started

  The ACTUAL_VALUE_OUT_OF_POSITIONING_WINDOW is displayed when the actual value has left the positioning window again.
- STANDSTILL_MONITORING_ACTIVE = standstill monitoring is active; positioning to the position reached at the end of the setpoint interpolation has taken place.
- INACTIVE

Note

Position-related monitoring functions (e.g. following error monitoring or position monitoring) are disabled with the command to reduce the torque.

Standstill monitoring

Standstill monitoring is defined by the standstill window and the tolerance time during which the standstill window may be exited without alarm 50107 being triggered.

The standstill window is specified as a deviation in **typeOfAxis.standStillMonitoring.stillStandTolerance**, meaning that half of the window width is set.

The status of the standstill monitoring is displayed in **servoMonitoring.stillstand**. Standstill monitoring is not available on the speed-controlled axis.

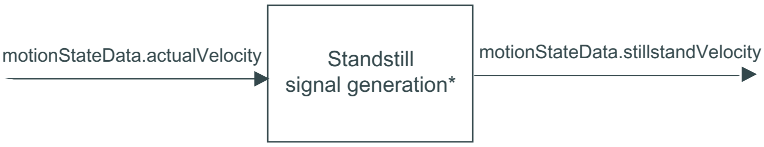
7.4 Standstill signal

The standstill signal **motionStateData.stillstandVelocity** is ACTIVE when the actual velocity is less than a configured velocity threshold for at least the duration of the delay time.

Note

Below this velocity, the motion is stopped in response to **_stopEmergency()** at zero setpoint without a preconfigured deceleration ramp.

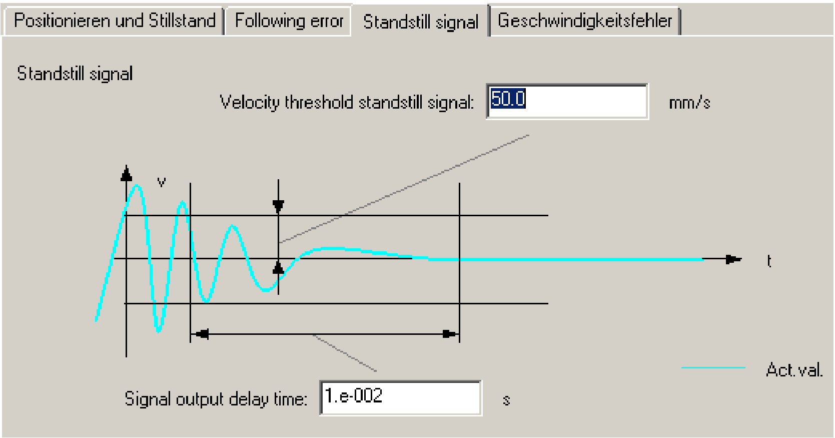
If the WHEN_MOTION_DONE step-enabling condition is set in speed-controlled mode for the speed-controlled axis and positioning axis, the command is ended when the standstill signal changes from INACTIVE to ACTIVE. The completion of commands with this setting for position-controlled motion is described in Position and standstill monitoring (Page 335).


motionStateData.actualVelocity
Standstill signal generation*
motionStateData.stillstandVelocity

* A standstill signal is generated when the actual velocity is less than the velocity set for the standstill signal during configuration.

Figure 7-6 Generating the standstill signal

The standstill signal is available on the positioning axis and speed-controlled axis.



Velocity threshold standstill signal: 50.0 mm/s

Signal output delay time: 1.e-002 s

Figure 7-7 Function and adjustable parameters for the standstill signal

Note

It is the standstill signal described here which is used in block FB_OperationMode and not the standstill signal specified by the drive! For this reason, take particular care when you set up the signal.
One of its functions, for example, is to ensure that switchover between operating modes takes place only when the drive is at a standstill.

1% of the maximum velocity can be used as a guide value.

7.5 Overspeed signal from SINAMICS

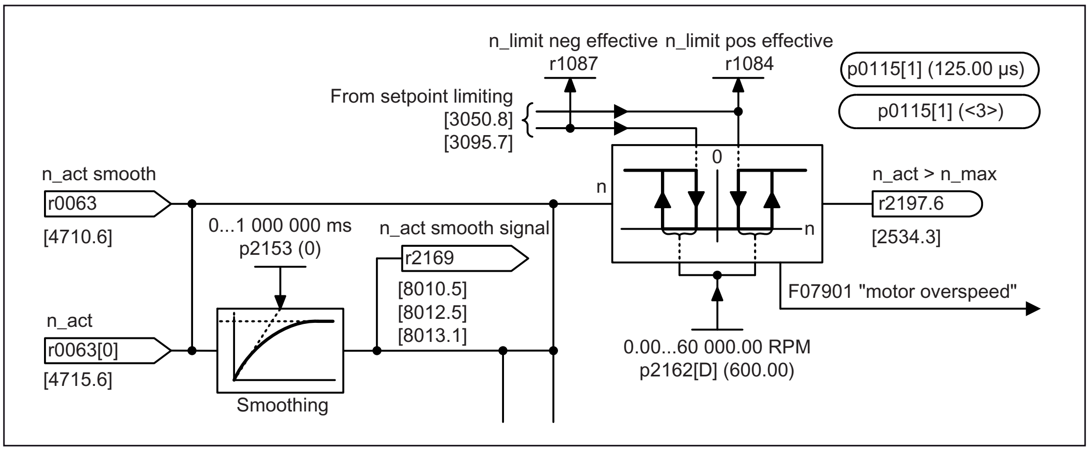


Figure 7-8 Overspeed signal

Positive and negative limits are derived from SINAMICS S parameter p1082 (maximum speed). The hysteresis is set at SINAMICS S parameter p2162.

Note

The complete function diagram (8010) is available in the SINAMICS S List Manual (Ref. [4]).

7.6 Application error messages and alarm messages

The function blocks (see Crane FB library (Page 163)) also return application fault messages and alarm messages: In the table, errors are identified with the letter "F" and alarms, with the letter "A".

Fault and alarm numbers are transferred over the same channel (PZD 4; see Table 6-7 SIMOTION → S7 (Page 315)) to the higher-level controller.

Because only one value can be transferred, a prioritization occurs (see FB_ErrorPriority (Page 187)). The fault or alarm number can be assigned as described below.

Table 7- 1 Fault or alarm number assignment

PZD4 (value not equal to 0) + AppZsw2 bit 7 (true)	=	SINAMICS fault
PZD4 (value not equal to 0) + AppZsw2 bit 6 (true)	=	SIMOTION fault
PZD4 (value not equal to 0) + AppZsw2 bit 8 (true)	=	FB fault
PZD4 (value not equal to 0) + Zsw1 bit 7 (true)	=	Drive alarm
PZD4 (value not equal to 0) + no further feedback	=	FB alarm

The first entry has the highest priority, the last entry the lowest priority.

Error No.	Description
0000	No error
F0001 - F0048	Description of the return value: Corresponds to the list of return values under the _move function in SIMOTION; see Ref. [3].
F2700	FB_ControlAxis: The technology object is neither a positioning axis, nor is it a speed axis. Remedy: Set up a positioning axis or a speed axis at the TO.
F3000	FB_OperationMode: The input variables "SelectSynchronousOperation" and "SelectMasterSlaveOperation" are selected at the same time. To correct or avoid errors: It is not permitted to set the "SelectSynchronousOperation" and "SelectMasterSlaveOperation" in parallel.
A3100	The technology object is not switched active. Remedy: Set bit 15 in application control word 2.
A3110	More than one operating mode is selected. Remedy: Check bits 10-15 in application control word 1 and set only one of them to TRUE.
A3140	An operating mode has been selected while POWER was active. Remedy: Switch off the power status message of the axis using STW1 bit 0 or bit 3.
F3200	FB_OperationMode: (with tandem mode not active) Slave feedback signal: Synchronous velocity or synchronous position is not active. Remedy: Check the selection for synchronous mode in the slave; refer to Synchronous operation control type (Page 257).

Error No.	Description
F3210	FB_OperationMode: (with tandem mode not active) Slave feedback signal: The two operating modes are not the same (AUTOMATIC, MANUAL, SPEED_CONTROLLED, SWAYCONTROL). Remedy: Assign the same operating mode to the master and the slave.
F3220	FB_OperationMode: (with tandem mode not active) Slave feedback signal: Master-slave operation or synchronous mode is not active. Remedy: Select master-slave operation or synchronous mode in the slave; refer to Master-slave operation control type (Page 253) and/or Synchronous operation control type (Page 257).
A3230	FB_OperationMode: (with tandem mode not active) Slave drive state is not active. Remedy: Check the ramp-function generator enable STW1 bit 4 and speed controller enable STW1 bit 8 in the slave.
A3240	FB_OperationMode: (with tandem mode not active) Slave power is not active. Remedy: Check STW1 (STW1 bit 0,1,2,3,8). These bits must be set in the slave.
A3250	FB_OperationMode: (with tandem mode not active) Slave setpoint is not active. Remedy: Activate slave setpoint (STW1 bit 6).
A3260	FB_OperationMode: Position synchronous operation not possible, as the slave is outside the tolerance range. Remedy: Bring the slave back into the tolerance range.
A3270	FB_OperationMode: Synchronous operation not possible, as the position from the synchronous object (FixGear) to the actual position is outside the tolerance. Remedy: Switch off the slave (STW1 bit3), save offset (AppSTW1 bit8) and switch on again.
F3300	FB_OperationMode: (with tandem mode active) Slave feedback signal: Synchronous velocity or synchronous position is not active. Remedy: Check the selection for synchronous mode in the slave; refer to Synchronous operation control type (Page 257).

Error No.	Description
F3310	FB_OperationMode: (with tandem mode active) Slave feedback signal: The two operating modes are not the same (AUTOMATIC, MANUAL, SPEED_CONTROLLED, SWAYCONTROL) Remedy: Assign the same operating mode to the master and the slave.
F3320	FB_OperationMode: (with tandem mode active) Slave feedback signal: Master-slave operation or synchronous mode is not active. Remedy: Select master-slave operation or synchronous mode in the slave; refer to Master-slave operation control type (Page 253) and/or Synchronous operation control type (Page 257).
A3330	FB_OperationMode: (with tandem mode active) The master switch has been deflected, but the slave drive state is not active. Remedy: Check the ramp-function generator enable STW1 bit 4 and speed controller enable STW1 bit 8 in the slave.
A3340	FB_OperationMode: (with tandem mode active) The master switch has been deflected, but slave power is not active. Remedy: Check STW1 (STW1 bit 0,1,2,3,8). These bits must be set in the slave.
A3350	FB_OperationMode: (with tandem mode active) The master switch has been deflected, but the slave setpoint is not active. Remedy: Activate slave setpoint (STW1 bit 6).
F3360	FB_OperationMode: (with tandem mode not active) Slave is active (by selecting master-slave or synchronous operation) and the slave has a TO, DO or FB error. Remedy: Eliminate the TO, DO, or FB error in the slave.
F3400	FB_OperationMode: (with tandem mode active) Tandem slave feedback signal: Synchronous velocity or synchronous position is not active. Remedy: Check the selection for synchronous mode in the tandem slave; refer to Synchronous operation control type (Page 257).
F3410	FB_OperationMode: (with tandem mode active) Tandem slave feedback signal: The two operating modes are not the same (AUTOMATIC, MANUAL, SPEED_CONTROLLED, SWAYCONTROL). Remedy: Assign the same operating mode to the tandem master and the tandem slave.

Error No.	Description
F3420	**FB_OperationMode:** (with tandem mode active) Tandem slave feedback signal: Master-slave operation or synchronous mode is not active. **Remedy:** Select master-slave operation or synchronous mode in the tandem slave; refer to Master-slave operation control type (Page 253) and/or Synchronous operation control type (Page 257).
A3430	**FB_OperationMode:** (with tandem mode active) The master switch has been deflected, but the tandem slave drive state is not active. **Remedy:** Check the ramp-function generator enable STW1 bit 4 and speed controller enable STW1 bit 8 in the tandem slave.
A3440	**FB_OperationMode:** (with tandem mode active) The master switch has been deflected, but the tandem slave power is not active. **Remedy:** Check STW1 (STW1 bit 0,1,2,3,8). These bits must be set in the tandem slave.
A3450	**FB_OperationMode:** (with tandem mode active) The master switch has been deflected, but the tandem slave setpoint is not active. **Remedy:** Activate tandem slave setpoint (STW1 bit 6).
F3500	**FB_OperationMode:** (with tandem mode active) Tandem slave slave feedback signal: Synchronous velocity or synchronous position is not active. **Remedy:** Check the selection for synchronous mode in the tandem slave slave; refer to Synchronous operation control type (Page 257).
F3510	**FB_OperationMode:** (with tandem mode active) Tandem slave slave feedback signal: The two operating modes are not the same (AUTOMATIC, MANUAL, SPEED_CONTROLLED, SWAYCONTROL). **Remedy:** Assign the same operating mode to the tandem slave and the tandem slave slave.
F3520	**FB_OperationMode:** (with tandem mode active) Tandem slave slave feedback signal: Master-slave operation or synchronous mode is not active. **Remedy:** Select master-slave operation or synchronous mode in the tandem slave slave; refer to Master-slave operation control type (Page 253) and/or Synchronous operation control type (Page 257).

Error No.	Description
A3530	FB_OperationMode: (with tandem mode active) The master switch has been deflected, but the tandem slave slave drive state is not active. Remedy: Check the ramp-function generator enable STW1 bit 4 and speed controller enable STW1 bit 8 in the tandem slave slave.
A3540	FB_OperationMode: (with tandem mode active) The master switch has been deflected, but tandem slave slave power is not active. Remedy: Check STW1 (STW1 bit 0,1,2,3,8). These bits must be set in the tandem slave slave.
F3541	FB_OperationMode: Slave mode or tandem slave mode is activated, although no valid technology object is available at input TO_checkbackMasterAxis. Remedy: Connect input TO_checkbackMasterAxis to a valid technology object or deselect slave mode or tandem slave mode again.
F3542	FB_OperationMode: Slave mode or tandem slave mode is activated, although no valid technology object is available at input TO_checkbackMasterAxis2. Remedy: Connect input TO_checkbackMasterAxis2 to a valid technology object or deselect slave mode or tandem slave mode again.
F3543	FB_OperationMode: Slave mode or tandem slave mode is activated, although no valid technology object is available at input TO_checkbackMasterAxis3. Remedy: Connect input TO_checkbackMasterAxis3 to a valid technology object or deselect slave mode or tandem slave mode again.
F3544	FB_OperationMode: Inputs boSelectMasterAxis2 and boSelectMasterAxis3 are set simultaneously (with the result that more than one master is selected). Remedy: Select just one master.
A3550	FB_OperationMode: (with tandem mode active) The master switch has been deflected, but the tandem slave slave setpoint is not active. Remedy: Activate tandem slave slave setpoint (STW1 bit 6).
F3560	FB_OperationMode: Slave available is active and a TO, DO, or FB error is present at the slave, tandem slave, or tandem slave slave. Remedy: Eliminate the TO, DO, or FB error in the slaves.

Error No.	Description
F3600	**FB_OperationMode:** Synchronous mode or master-slave mode is selected, master is moving, but not the slave. **Remedy:** Either deselect synchronous mode or master-slave operation, or select slave mode or tandem slave mode.
A3620	**FB_OperationMode:** The master switch was deflected, the drive receives a travel command and uses the extended brake control in SINAMICS, but the drive does not move. **Remedy:** Open the brake and check STW1 bit 9.
A3630	**FB_OperationMode:** The master switch has been deflected, but the master DriveState is not active. **Remedy:** Check the ramp-function generator enable STW1 bit 4 and speed controller enable STW1 bit 8 in the master.
A3640	**FB_OperationMode:** The master switch has been deflected, but master Power is not active. **Remedy:** Check STW1 (STW1 bit 0,1,2,3,8). These bits must be set in the master.
A3650	**FB_OperationMode:** The master switch has been deflected, but the master setpoint is not active. **Remedy:** Activate master setpoint (STW1 bit 6).
A3660	**FB_OperationMode:** No operating mode selected. **Remedy:** Select operating mode (AppSTW1 bit 10…15).
A3670	**FB_OperationMode:** Speed controller enable missing. **Remedy:** Set STW1 bit 8.
A3680	**FB_OperationMode:** Technology object is not active. **Remedy:** Check AppSTW2 bit15
A3690	**FB_OperationMode:** Closed-loop position controlled operating mode is not active. **Remedy:** Axis must be referenced (AppSTW2 bit10), state AppZSW2 bit10
F3700	**FB_OperationMode:** The technology object master is neither a positioning axis, nor is it a speed axis. **Remedy:** Refer to fault number 2700

Error No.	Description
F3710	**FB_OperationMode:** Closing Gear is active, slave mode or tandem slave mode is selected, but no valid master technology object is available. **Remedy:** Connect a valid technology object (position axis) as the master or deselect slave mode or tandem slave mode.
F3720	**FB_OperationMode:** In speed-controlled synchronous operation (SPEED_CONTROLLED, SENSORLESS EMERGENCY), a valid gear (TO_GearVel) or a master axis is expected. **Remedy:** Connect a valid gear (fixed gear).
F3730	**FB_OperationMode:** In position-controlled synchronous operation (AUTOMATIC or MANUAL), a valid gear (TO_GearPos) or a leading axis is expected. **Remedy:** Connect a valid gear or an axis as the master.
F3740	**FB_OperationMode:** For synchronous operation between the holding gear (master) and the closing gear (slave) or for a following-operation coupling, the master drive runs but the slave drive remains stationary. **Remedy:** At the slave drive, check whether all enable signals are present, whether the following operation coupling has been selected correctly, or whether error messages are present. If necessary, attempt to run the slave drive on its own with no coupling in order to establish that there are no faults with the drive.
F3800	**FB_OperationMode:** SWAYCONTROL and HeavyDuty are active simultaneously. **Remedy:** SWAYCONTROL cannot be used in heavy-duty operation. Deselect heavy-duty operation.
A3910	**FB_OperationMode:** SENSORLESS EMERGENCY is selected but the maximum permissible velocity for SENSORLESS EMERGENCY is zero. **Remedy:** Increase the limit for SENSORLESS EMERGENCY in the interface of the MCC unit.
A3940	**FB_OperationMode:** The master switch is deflected in the positive direction and the target position is smaller than the actual position in AUTOMATIC or MANUAL operating mode. **Remedy:** If the deflection direction is correct, the actual position and target position must be checked and corrected.
A3950	**FB_OperationMode:** The master switch is deflected in the negative direction and the target position is greater than the actual position in AUTOMATIC or MANUAL operating mode. **Remedy:** If the deflection direction is correct, the actual position and target position must be checked and corrected.

Error No.	Description
A3960	FB_OperationMode Internal axis changeover is selected but the assigned axis interface is **not** assigned EXCLUSIVE. Remedy: Change the configuration data for TypeOfAxis.SetpointDriverInfo.interfaceAllocation to 939.
A3970	FB_OperationMode External axis changeover is selected but the axis interface is not set to NON_EXCLUSIVE_AND_STARTUP_DEACTIVATED. Remedy: Change the configuration data for TypeOfAxis.SetpointDriverInfo.interfaceAllocation to 940 and, if required, TypeOfAxis.NumberOfEncoder.Encoder_1.interfaceAllocation to 940.
A3980	FB_OperationMode External axis changeover is selected but the axis sequence is **not** OK Remedy: Check the circuit in the subroutine call of the inputs in the "boChangeTO_Nr_1 ", "boChangeTO_Nr_2 ", "boChangeTO_Nr_3" mode block or read Chapter Technology object switchover (Page 214).
F3990	FB_OperationMode: Changing the target position in MANUAL operating mode decreases the distance-to-go to the new target. A check is made to determine whether the deceleration value can be increased to a level that allows the drive to decelerate within the shortened distance-to-go. If the deceleration required for this is greater than the maximum permissible deceleration value, this fault is generated. Remedy: Make sure that the selected distance-to-go is long enough to enable the drive to come to a standstill at the maximum deceleration.
F4000	FB_Cornering: Incorrect operating mode selected for cornering movement. Remedy: Select SPEED_CONTROLLED, JOGGING, or SENSORLESS_EMERGENCY operating mode.
F4010	FB_Cornering: Technology object **InnerAxis** is not a speed axis. Remedy: Refer to fault number 2700
F4020	FB_Cornering: Technology object **OuterAxis** is not a speed axis. Remedy: Refer to fault number 2700
F4040	FB_Cornering: The drives are not in synchronous operation. Remedy: Switch the drives to synchronous operation (refer to the section titled Synchronous operation control type (Page 257)). Interconnect the block's "boSynchronousOperationActive" input to the slave drive's FB_OperationMode "boSynchronousOperationActive" output.

Error No.	Description
F4050	FB_Cornering: The actual velocity is greater than the permissible curve velocity. Remedy: Check the velocity values at the block input.
F4070	FB_Cornering: Change in direction of travel above a sensor. Remedy: Continue to move in the original direction, pass the sensor completely, and then change the direction.
F4080	FB_Cornering: Parameterization fault, r64DistanceSmallRadius > r64DistanceLargeRadius. Remedy: Parameterize the above values correctly. The value at the input for the larger radius must be the larger of the two.
F4090 F4091	FB_Cornering: Permissible dynamic values exceeded or the curve entered too quickly (F4090: positive acceleration / F4091: negative acceleration). Remedy: Check the dynamic values and r64DistanceSensorCurve. Enter the corner more slowly.
F5000	FB_Monitoring: In master-slave operation, the actual velocity between the master and slave is outside the tolerance range. Remedy: • Check the mechanical function. • Check the reference data settings in S7 – SIMOTION – SINAMICS. • Increase the deceleration time. • Increase the velocity tolerance. • Scan the system for dead times (e.g. smoothing times).
F5100	FB_Monitoring: In synchronous operation in the AUTOMATIC / MANUAL operating mode, the actual position - without offset between the master and slave axis - is outside the tolerance range. To correct or avoid faults: Refer to fault 5000; also check for differences in the position control settings at both axes.
F5200	FB_Monitoring: In synchronous operation in the AUTOMATIC / MANUAL operating mode, the actual position - with offset between the master and slave axis - is outside the tolerance range. To correct or avoid faults: Refer to fault 5100
F5300	FB_Monitoring: In synchronous operation the actual position between the master and slave axis is outside the tolerance range in all operating modes with the exception of AUTOMATIC / MANUAL. To correct or avoid faults: Refer to fault 5000

Error No.	Description
F5400	**FB_Monitoring:** In synchronous operation or master-slave operation, the feedback signals of both technology objects (TO) are monitored. This monitors whether one of the two TOs is not switched in and the TO is moving. **To correct or avoid faults:** Check whether all enable signals are set or whether any one of the TOs is still moving.
F5500	**FB_Monitoring:** In synchronous operation or master-slave operation, the current setpoints or the actual currents respectively are outside the tolerance range. **Remedy:** • Check whether master-slave mode is actually activated. • Check the mechanical function. • Increase the deceleration time. • Increase the current tolerance. • Scan the system for dead times (e.g. smoothing times).
F5700	**FB_Monitoring:** The technology object **Master** is not a positioning axis and not a speed axis. **Remedy:** Refer to fault number 2700
F5800	**FB_Monitoring:** The technology object **Slave** is not a positioning axis and not a speed axis **Remedy:** Refer to fault number 2700
F5950	**FB_Monitoring:** The velocity difference between the motor encoder and the external encoder of the **master** is outside the tolerance range. **Remedy:** Check for axis fracture or adjust tolerance limits, increase deceleration time, check encoders and/or the signals "toDriveAxis.sensordata[1].velocity" and "toDriveAxis.sensordata[2].velocity".
F5960	**FB_Monitoring:** The velocity difference between the motor encoder and the external encoder of the **slave** is outside the tolerance range. **Remedy:** Refer to fault number 5950
F8750	**FB_ReferenceMode:** The technology object **master** is not a positioning axis. **Remedy:** Refer to fault number 2700
F8850	**FB_ReferenceMode:** The slave technology object **slave** is not a positioning axis. **Remedy:** Refer to fault number 2700
F8950	**FB_ReferenceMode:** No valid gear (TO_GearPos) available. **Remedy:** Connect a valid gear (fixed gear).

Error No.	Description
F9000	FB_AutoSettingFW: No valid technology object connected. Remedy: Connect valid technology object at the TO_Name input.
F9010	FB_AutoSettingFW: Speed controller deviation too high. Remedy: Optimize the speed control loop.
F9020	FB_AutoSettingFW: Interruption too long / pause between the measuring steps. Remedy: Repeat the current measuring step (Step 10 or Step 20).
F9030	FB_AutoSettingFW: The velocity during the constant travel differs too greatly from the rated velocity. Remedy: Set the correct rated velocity at the input or ensure that the rated velocity is overshot or undershot too much during the constant travel.
A9040	FB_AutoSettingFW: The minimum load value is larger than the maximum load value. Remedy: Check the entries. A reset is not necessary. The alarm is removed when the input values match again.
F9050	FB_AutoSettingFW: The number of sampling points is too small. Remedy: Check the entries and, if necessary, change the ramp-up time of the lower and/or upper measuring window.
F9060	FB_AutoSettingFW: The entered ramp-up time and real ramp-up time are too different. Remedy: Check the entries and, if necessary, change the ramp-up time.

Note

If you receive a fault message that is not included in the above list, then look here:

- for SINAMICS in Ref. [4]
- for SIMOTION in Ref. [3]

7.7 Troubleshooting

Certain settings or actions, such as software or project updates, may lead to errors, where the cause is not immediately obvious. To avoid and resolve problems such as these, see the link in the following note.

Note

You can find FAQs on the subject of troubleshooting in the Internet (http://support.automation.siemens.com/WW/view/en/10807397/130000). For other forms of contact, such as a service request, see Preface (Page 3).

Product support for SIMOCRANE

Use the addresses shown below to receive support for your SIMOCRANE products (under the keyword "Simocrane"):

- On the Internet:

  http://support.automation.siemens.com
- Europe hotline
  - Tel.: +49 (0) 911 895 7 222
  - Fax: +49 (0) 911 895 7 223
  - E-mail: support.automation@siemens.com
- America hotline
  - Tel.: +1 423 262 5710
  - Fax: +1 423 262 2231
  - E-mail: support.america.automation@siemens.com
- Asia/Pacific hotline
  - Tel.: +86 10 6475 7575
  - Fax: +86 10 6474 7474
  - E-mail: support.asia.automation@siemens.com

Commissioning 8

8.1 Preliminary remark

CAUTION
The polarity rules must be observed: Hoist: Hoisting velocity is POSITIVE, with ascending position values. Lowering velocity is NEGATIVE, with descending position values. Gantry: Motion to right, POSITIVE velocity with ascending position values. Motion to left, NEGATIVE velocity with descending position values. Trolley: Motion towards ship, POSITIVE velocity with ascending position values. Motion towards shore, NEGATIVE velocity with descending position values. Boom: Hoisting velocity is POSITIVE, with ascending position values. Lowering velocity is NEGATIVE, with descending position values. Holding Gear: Hoisting velocity is POSITIVE, with ascending position values. Lowering velocity is NEGATIVE, with descending position values. Closing Gear: Closing, velocity is POSITIVE, with ascending position values. Opening, velocity is NEGATIVE, with descending position values. (Position "zero", grab is fully open.) Slewing Gear: Slewing right, velocity is POSITIVE, with ascending position values. Slewing left, velocity is NEGATIVE, with descending position values. Luffing Gear: Hoisting velocity is POSITIVE, with ascending position values. Lowering velocity is NEGATIVE, with descending position values.

Two commissioning concepts:

1. The required functionality of the actual crane goes beyond the standard application. The user is very knowledgeable about SIMOTION / SINAMICS and is in a position to adapt the existing standard application "ready-to-apply" to the requirements.
2. The standard application covers the functional requirements of the actual crane. The user is not so knowledgeable about SIMOTION / SINAMICS and user is supplied with a "ready-to-run" standard configuration. This only has to be downloaded into SIMOTION D435 and then commissioning can start.

The "ready-to-run" procedure is described in this chapter.

The following diagram provides an overview regarding the commissioning sequence.

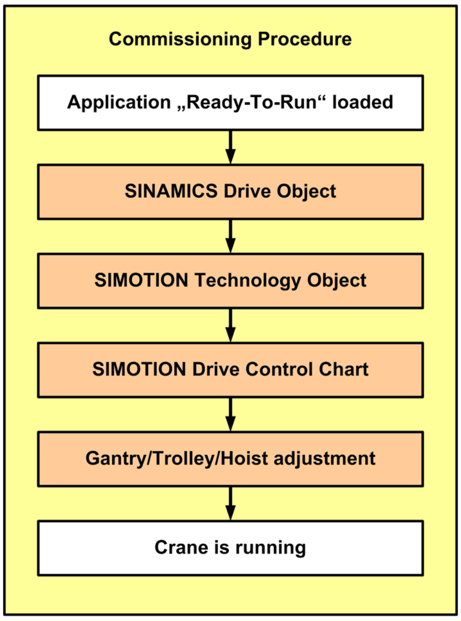

Commissioning Procedure
Application „Ready-To-Run“ loaded
SINAMICS Drive Object
SIMOTION Technology Object
SIMOTION Drive Control Chart
Gantry/Trolley/Hoist adjustment
Crane is running

Figure 8-1 Commissioning overview

Prerequisites:

⚠ WARNING

The system must be in a no-voltage state when checking the subsequently described prerequisites for commissioning

- All DRIVE-CLiQ nodes must be connected with one another precisely as was created in the project (reference and actual topology must match).
- The commissioning engineer is responsible for ensuring that the motors are correctly connected.
- The encoder must be correctly connected.

- Only connect in parallel drives of the same type and manufacturer in order to guarantee identical drive data and symmetrical load distribution.
- Master-slave-coupled motors must have the same reference variables, i.e. p2000, p2001, p2002, p2003 and r2004.

8.2 SIMOCRANE WebStart

SIMOCRANE WebStart is a Web-based tool, i.e. access to the tool is made using a Web browser, such as the Internet Explorer. The physical connection to the controller is established via Ethernet, e.g. using a crossover Ethernet cable for direct connection. A notebook that normally serves as client is used essentially only as terminal.

A crane lifecycle from the engineering through to the service comprises the following:

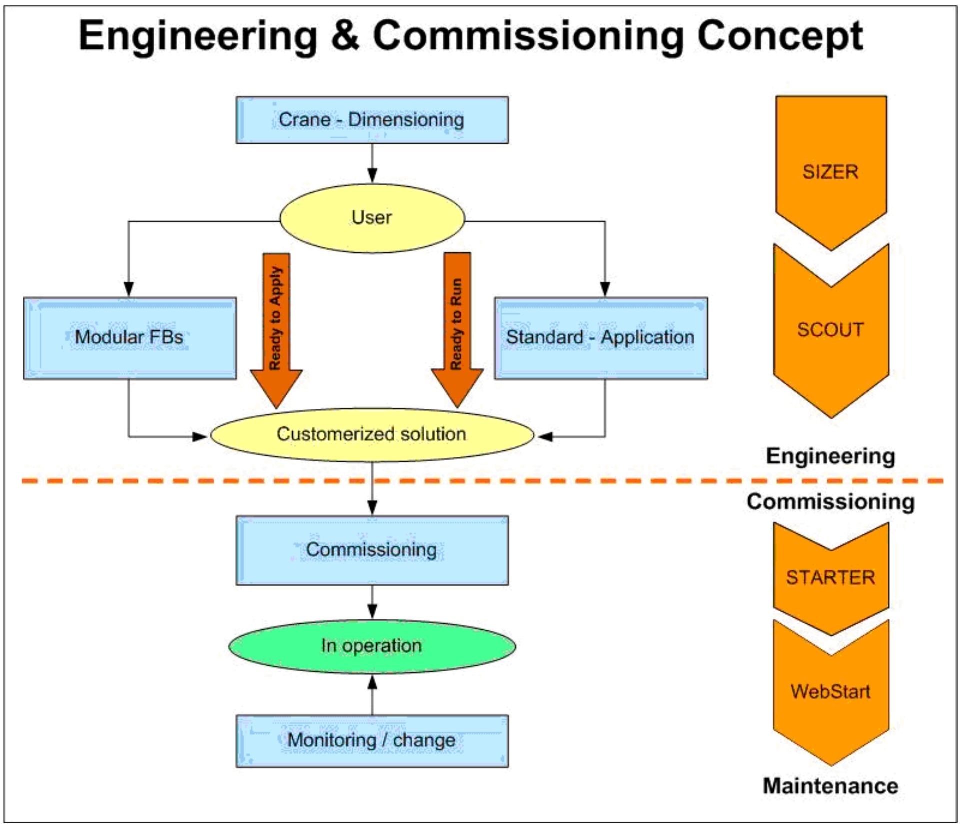


Figure 8-2 Concept for configuration and commissioning

1. Crane dimensioning (configuration phase in the office):

- Problem: The hardware configuration, the circuit diagrams and the topology of the interconnections are designed according to the crane specification.
- Tool: The design of the SINAMICS drive can be performed with the SIZER configuration tool.

2. Generation of the application software (configuration phase in the office):
   - For 'ready-to-run' customers: Implementation of smaller customizations (e.g. power, encoder types, topology) based on the standard application.
   - For 'ready-to-apply' customers: Implementation of customizations (hardware, software and communication) or in-house engineering with the existing Crane DCC/FB library.
   - Tool: SCOUT
3. Commissioning (locally on the crane):
   - Customization of the parameters and optimization of the control loops
   - Tool: STARTER and SIMOCRANE WebStart
   - SIMOTION SCOUT is required for changes to the application program.
4. Service and maintenance (locally on the crane):
   - Monitoring, diagnostics and trace and, if necessary, customization of the parameters.
   - Tool: SIMOCRANE WebStart

The SIMOCRANE Basic Technology with SIMOTION D controls the motion control and the cranes technology functions for the main drives. In most cases, the following customizations (hardware configuration, programs, DRIVE-CLiQ topology, etc.) can be performed during the configuration phase or in steps 1 and 2 (see above). In steps 3 and 4, only changes to parameters (system variables, parameters) and diagnostic options can be made locally. SIMOCRANE WebStart offers sufficient functions for this purpose.

The exact procedure for the configuration and commissioning is shown in the following figure:

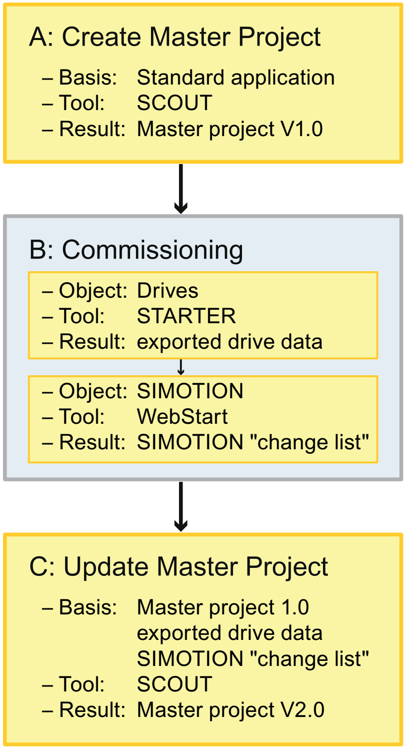


Figure 8-3 Use of WebStart during commissioning

A: Configuring phase:

A customer-specific SCOUT project is created on the basis of the standard application.

- For 'ready-to-run' customers: Implementation of smaller customizations (e.g. power, encoder types, topology) based on the standard application.
- For 'ready-to-apply' customers: Implementation of customizations (hardware, software and communication) or in-house engineering with the existing Crane DCC/FB library.
- SIMOTION SCOUT is required for this phase

Result: Creation of the master project V1.0 (original version)

B: Commissioning phase:

The commissioning of the drives is done with the STARTER. The data from the following components must be checked: Motor Module, Motor and Encoders. The control loops of the drives should then be optimized.

All changes must be saved and exported.

WebStart must be used to place the technology objects in the function module into operation:

1. Set the velocities, encoder data, controllers, etc.
2. Parameterize the crane technology functions in the prepared screen forms.

3. Parameterize the sequence control (limitation constants, default values, monitoring functions).
4. Save all the changes made in WebStart with the "SAVE" function.

C: Updating the master project:

The master project is updated automatically in SCOUT by importing the saved drive data and running a script file. This version can be used as a backup for end customers.

Result: Creating master project V2.0 (final version)

WebStart functional scope

Note

The following functions are **not** supported by WebStart:

- Program change
- Communications change
- Configuration change for DRIVE-CLiQ
- Drive commissioning

The following functions are contained in WebStart:

- All variable changes for SIMOTION, including MCC and DCC
- Display and acknowledgement of all alarm and error messages for SIMOTION, SINAMICS and SIMOCRANE
- Monitoring of the communication down to the bit level
  - SIMATIC ↔ SIMOTION D
  - SIMOTION ↔ SINAMICS
- Monitoring of the entire setpoint channel (from SIMATIC through to the drive)
- Trace function
- Watch table function
- Save function
- Integrated standard function 'IT-DIAG'
- Customer-specific function
  - Displaying and changing drive parameters
  - Creating a user-specific screen form

Implementation of WebStart in an existing project

SIMOCRANE WebStart can also be implemented in an existing SIMOCRANE Basic Technology project. In this case, the items mentioned below must be processed individually.

The necessary libraries can be found in the standard projects. The sources and standard projects are stored on the "SIMOCRANE Basic Technology" CD.

1. Import the LhdlUD library.
2. Import the xAdditional.st ST source.
3. Add the programs of the imported ST source file to the execution system:
   - The following programs must be added to the background task:

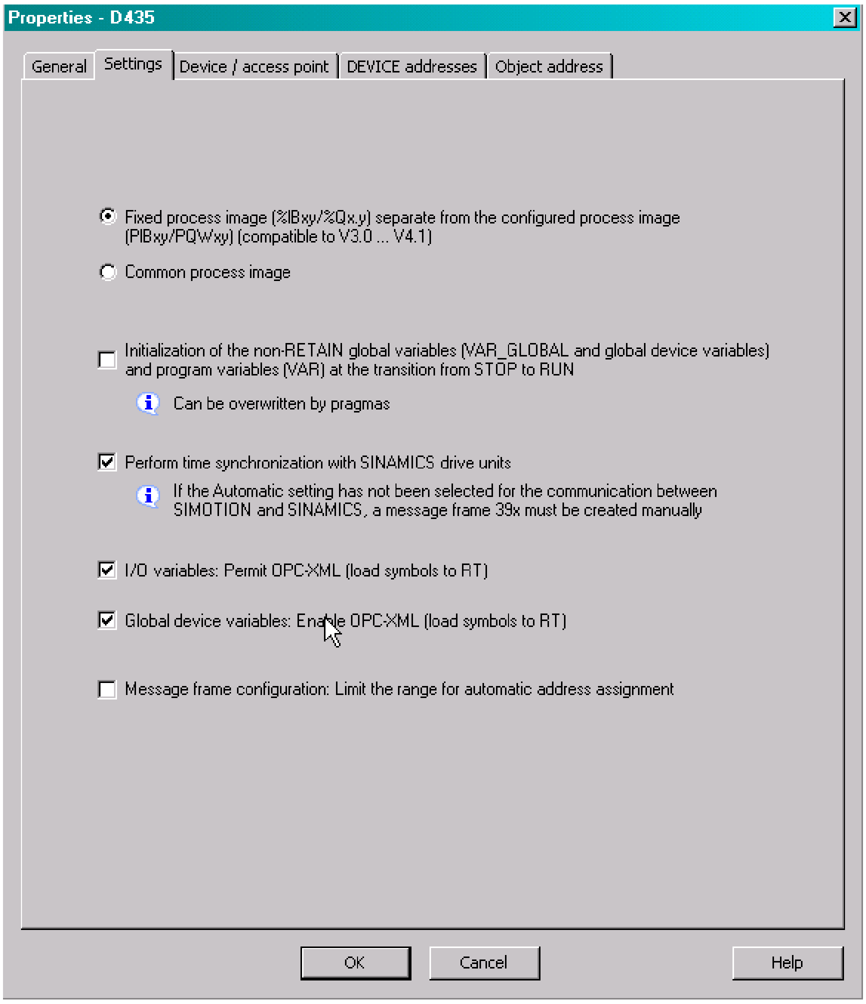
     pBackGroundSaveUnitDataSetp

     pBackGroundHandleBuffer

     pBackGroundPasswordCheck
   - The following programs must be added to the startup task:

     pStartUpLoadUnitDataSet
4. Copy the WebStart "WebStart.zip" file to the CF card.
   This makes available the required folders with the required files.

5. Open the "Settings" dialog (right-click "SIMOTION", "Properties" entry).



● Fixed process image (%IBxy/%Qx.y) separate from the configured process image (PIBxy/PQWxy) (compatible to V3.0 ... V4.1)

○ Common process image

□ Initialization of the non-RETAIN global variables (VAR_GLOBAL and global device variables) and program variables (VAR) at the transition from STOP to RUN

Can be overwritten by pragmas

☑ Perform time synchronization with SINAMICS drive units

If the Automatic setting has not been selected for the communication between SIMOTION and SINAMICS, a message frame 39x must be created manually

☑ I/O variables: Permit OPC-XML (load symbols to RT)

☑ Global device variables: Enable OPC-XML (load symbols to RT)

□ Message frame configuration: Limit the range for automatic address assignment

Figure 8-4 SIMOTION D435-2 properties

6. Place checkmarks for "Permit I/O variables: OPC-XML (load symbols into RT)" and for "Permit global device variables: OPC-XML (load symbols into RT)".

Call SIMOCRANE WebStart

Enter the following address in the Web browser to call SIMOCRANE WebStart:
http://<host>/WEBSTART.MBS

Enter the Internet address of the associated SIMOTION interface as <host>, for example 169.254.11.22.

In this case, the browser should be configured for direct access to the Internet.

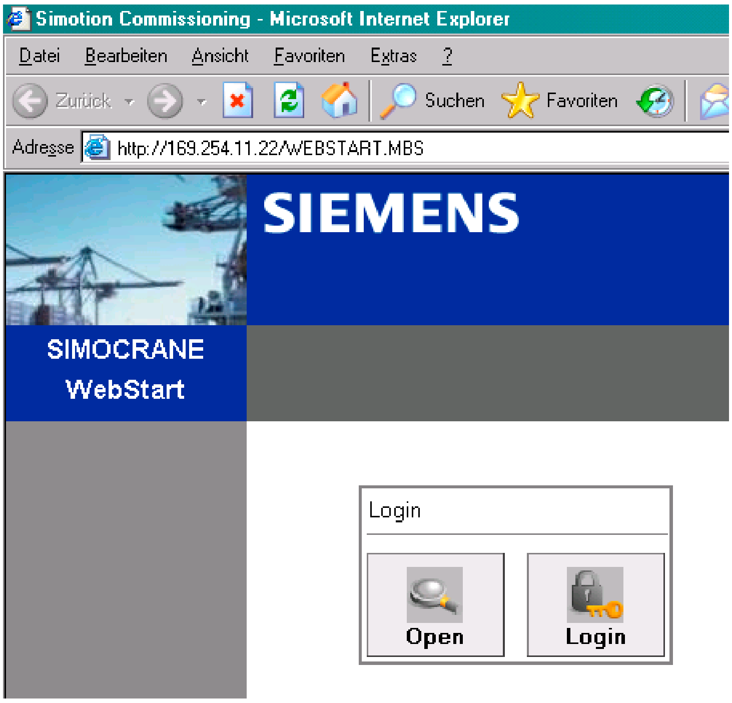
The following standard addresses should be used for the two Ethernet interfaces of the SIMOTION D435-2 DP/PN:

- X127: 169.254.11.22
- X150: 192.168.1.1

Example for direct address specification (standard Internet address of the controller):
http://169.254.11.22/WEBSTART.MBS

Password protection

The access to SIMOCRANE WebStart is password-protected. With access to SIMOCRANE WebStart, the user can browse all screen forms by clicking the "OPEN" button. A password must be entered when you click the "LOGIN" button.


Simotion Commissioning - Microsoft Internet Explorer

Adresse http://169.254.11.22/WEBSTART.MBS

Figure 8-5 WebStart start screen form

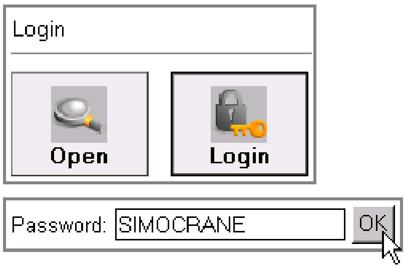

Password: SIMOCRANE

Figure 8-6 WebStart password

The default password "SIMOCRANE" can only be changed in the project. To do this, go to the "xAdditional" unit in the "pBackGroundPasswordCheck" program.

WebStart buttons

Place the SIMOTION into operation in the sequence described below. Start at far left in the commissioning bar.

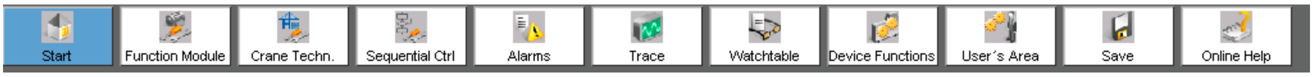


Figure 8-7 Commissioning screen-form icons

1. "Start" button

The start page shows an overview of the project:

- Configured SIMOTION and SINAMICS axes
- Switching option
- DCC blocks for the selected axis
- Version details for SIMOTION, SINAMICS and SIMOCRANE

2. "Function Module" button

First parameterize the corresponding function modules or technology axes in a specified screen form. This includes the encoders, communication, limitations and monitoring functions.

3. "Crane Technology" button

In the second step, parameterize the Crane Technology functions in the previously created screen forms. The individual technology is implemented as a block in DCC (Drive Control Chart), e.g. load-dependent field weakening for the hoist. The blocks are used for the setpoint processing of the function module.

4. "Sequential Control" button

Finally, the sequential control system must be parameterized. This includes some constants for the limitation, default values and monitoring functions.

Online help

Each SIMOCRANE WebStart settings page has a context-sensitive help page.

The online help is called by clicking the adjacent icon.

The "changelog.xml" commissioning file

The "changelog.xml" commissioning file contains all settings made with SIMOCRANE WebStart. This file is used every time the controller switches to the RUN mode (for example, after the power-off and on), in particular, also after restarting the crane.

After each commissioning, the SAVE function must be used to save the changes in the commissioning file on the SIMOTION.

After completion of the commissioning: Download the commissioning file from the controller and save. The commissioning file is located under the following path on the SIMOTION CF card:
[User]\Simotion\HMI\Files\changelog.xml

It can be stored on the local hard disk of the notebook or on any other suitable storage medium (e.g. a memory stick). This commissioning file should then be reimported into the SIMOTION SCOUT project or archived appropriately.

The commissioning file is executed with the "SimocraneOffline" script located in the project in the "Scripts" folder. If this is not done, the current commissioning data of the crane cannot be reproduced, for example, should a replacement memory card need to be created.

8.3 General

The scope of delivery is described in the chapter titled System overview (Page 11). The CD contains a number of standard applications for different crane types (see Standard applications (Page 497)). Before the user begins configuring the system, the setup file needs to be executed (see Setting up and updating the Crane_DCC_Library (Page 290)

It is then possible to open a suitable standard application with SIMOTION SCOUT. The following data needs to be checked and adapted if necessary:

SINAMICS:

- Motor Module data in the drive object
- Motor and encoder data in the drive object
- Reference data in the drive object
- Execution of script files for the relevant drive object

SIMOTION:

- Mechanical data in the technology object
- Limits in the technology object
- Monitoring functions in the technology object
- Encoder data in the technology object
- Reference data in the technology object

Crane-specific technologies (DCC):

- Adaptation of the master switch curve in the DCC block MasterSwitch
- Adaptation of the prelimit switch velocity in the DCC block PreLimitSwitch
- etc.

Adaptation of values in the MCC variables:

- Set the minimum velocity in the positive or negative direction for AUTOMATIC, MANUAL and EASY_POSITIONING operating modes
- Set the maximum velocity limit for the SENSORLESS EMERGENCY operating mode
- Set the reference current, reference torque, reference power and reference voltage
- etc.

The adapted standard application must be loaded to the CompactFlash (CF) Card once all necessary settings have been successfully adjusted. The user program can be loaded either directly to the CF card by means of a card reader or loaded into the SIMOTION D. If the second option is chosen, a RAM to ROM store operation must be performed afterwards to ensure that the user program is copied to the CF card for retentive storage.

When a project is being downloaded, there is an option to initialize RETAIN and non-RETAIN data. This selection can be made under Options → Settings → Download.

If the box next to **Initialization of the non-RETAIN technology object data** is checked, all technology objects are reinitialized during the download and, as a result, all alarms are cleared (including Power ON). No variable initialization will be performed for a download in RUN, even if this function is active here, because values in the user program cannot be changed during operation. If the box is not checked, only modified technology objects are loaded to the target system and initialized during the download in SCOUT. Pending alarms are acknowledged. If a Power ON alarm is pending on an unchanged technology object, this causes the download to be aborted. This does not affect the source data.

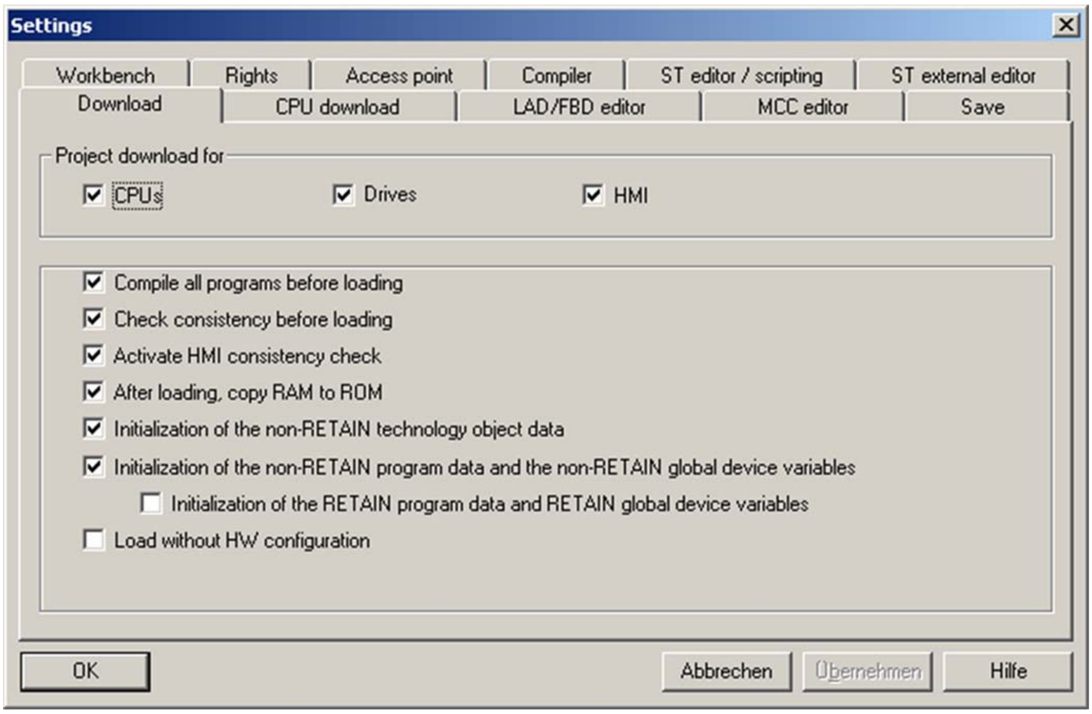
Check the box **Initialization of the non-RETAIN program data and the non-RETAIN global device variables** to initialize non-RETAIN program data and the non-RETAIN data of global device variables during a download. This does not affect the technology object data.

If box **Initialization of the RETAIN program data and the RETAIN global device variables** is checked, RETAIN program data and the RETAIN data of global device variables can be initialized during a download too. This does not affect the technology object data.

However, as a consequence all of the actual positions of the various axes and the retain variables under the global variables are reset to the initial value. As a consequence, for a grab crane, the grab must be re-adjusted and for the other axes, it may be necessary to re-reference them.

However, as a consequence, all of the actual positions of the various axes and the retain variables under the global variables are reset to the initial value. Specifically, this means the following:

- For a grab crane, the grab must be re-adjusted.
- It may be necessary to re-reference the other axes.



Project download for

☑ CPUs ☑ Drives ☑ HMI

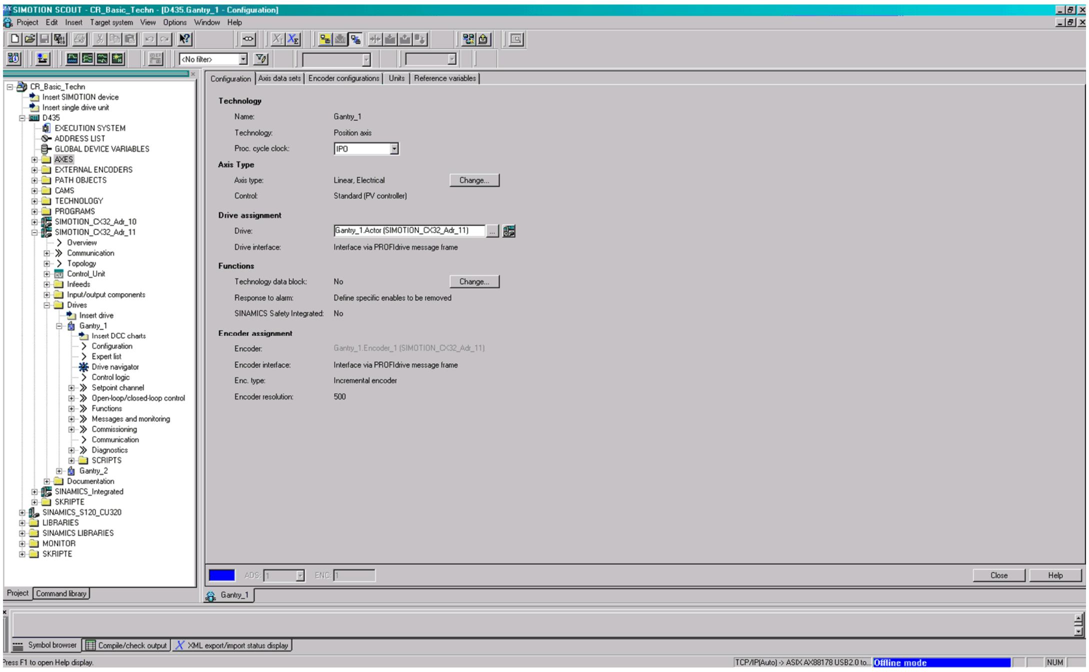
☑ Compile all programs before loading
☑ Check consistency before loading
☑ Activate HMI consistency check
☑ After loading, copy RAM to ROM
☑ Initialization of the non-RETAIN technology object data
☑ Initialization of the non-RETAIN program data and the non-RETAIN global device variables
□ Initialization of the RETAIN program data and RETAIN global device variables
□ Load without HW configuration

Figure 8-8 Download settings

Communication via PROFIBUS between the S7 and SIMOTION and between SIMOTION and SINAMICS is preconfigured in the standard projects; further information can be found in the chapter titled Communication (Page 293).

The commissioning procedure is described below.

Before users start the commissioning procedure, they should ensure that the motor data and encoder data in the SINAMICS drive object match those of their real system. This should be checked in the configuration of the SINAMICS drive object (see the figure below). The users must also ensure that the reference speed (p2000) and maximum speed (p1082) for SINAMICS match the maximum speed (TypeOfAxis.SetPointDriveInfo.DriveData.maxSpeed) for SIMOTION. In SINAMICS, the reference parameters p2000 to p2003 must be checked. In SIMOTION, reference parameter TypeOfAxis.SetPointDriveInfo.DriveData must be checked.



Technology

Name: Gantry_1

Technology: Position axis

Proc. cycle clock: IPO

Axis Type

Axis type: Linear, Electrical

Control: Standard (PV controller)

Drive assignment

Drive: Gantry_1.Actor (SIMOTION_CX32_Adr_11)

Drive interface: Interface via PROFIdrive message frame

Functions

Technology data block: No

Response to alarm: Define specific enables to be removed

SINAMICS Safety Integrated: No

Encoder assignment

Encoder: Gantry_1.Encoder_1 (SIMOTION_CX32_Adr_11)

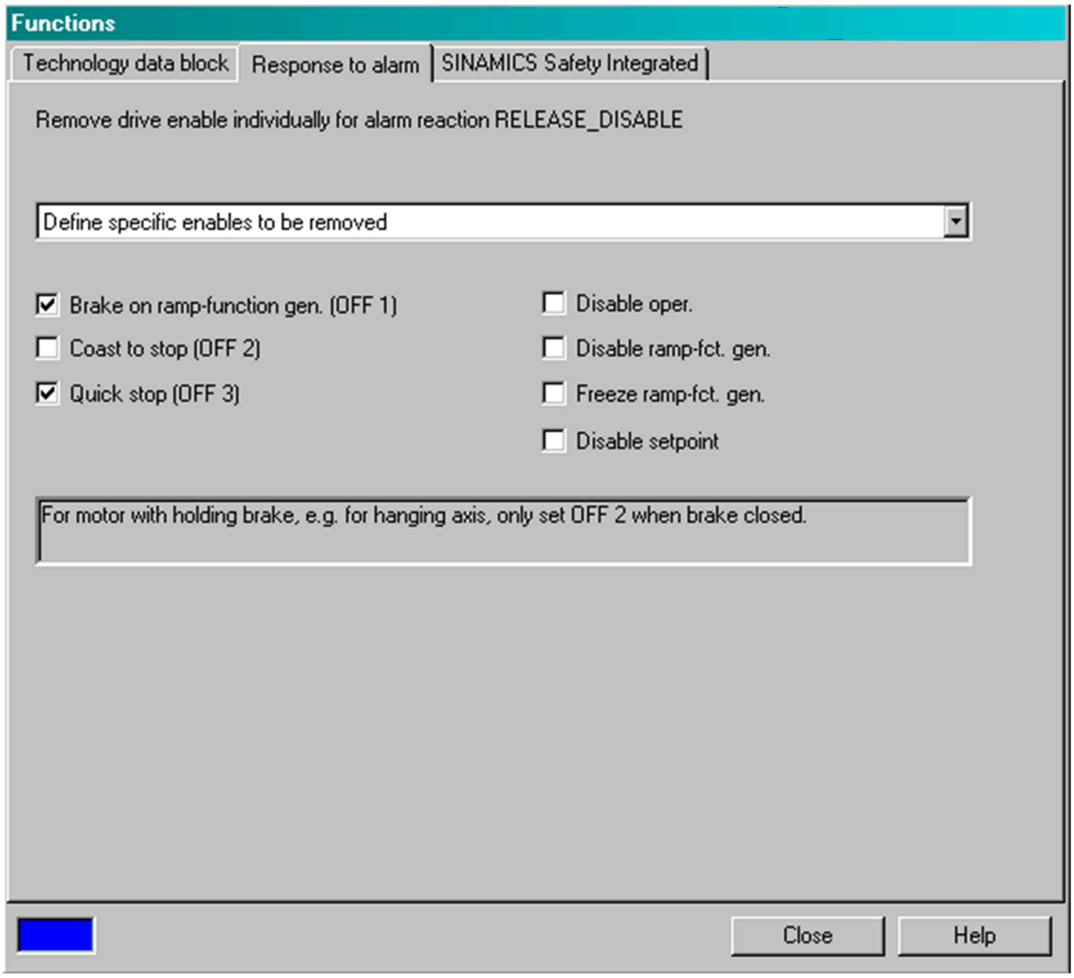
Encoder interface: Interface via PROFIdrive message frame

Enc. type: Incremental encoder

Encoder resolution: 500

Figure 8-9 TO Gantry_1 configuration file

The checkmark **Tolerate the encoder failure when it is not involved in the closed-loop control** must be set for each technology object data under **Encoder configuration** → **Settings for the Drive** in the technology object.



Remove drive enable individually for alarm reaction RELEASE_DISABLE

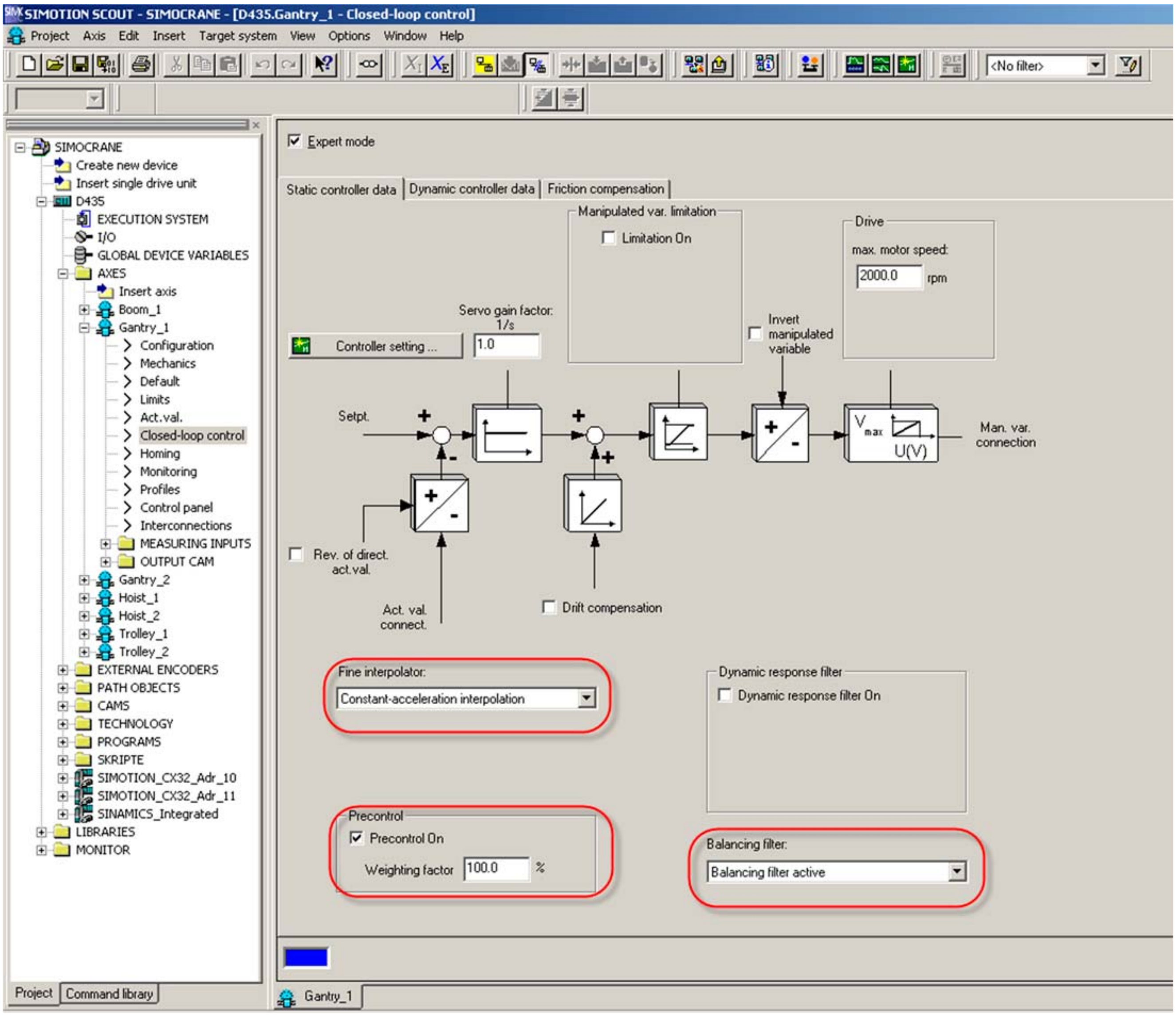
Define specific enables to be removed

☑ Brake on ramp-function gen. (OFF 1)
□ Coast to stop (OFF 2)
☑ Quick stop (OFF 3)
□ Disable oper.
□ Disable ramp-fct. gen.
□ Freeze ramp-fct. gen.
□ Disable setpoint

For motor with holding brake, e.g. for hanging axis, only set OFF 2 when brake closed.

Figure 8-10 Settings for the drive under DO Gantry_1

The user must then check the settings for maximum velocity, acceleration, deceleration and encoder data in the SIMOTION technology object. In SIMOTION, the reference data for velocity must be checked in the configuration data of the TO Gantry_1 under TypeOfAxis.SetPointDriverInfo.DriveData (see the section titled SINAMICS drive object (Page 400)).



☑ Expert mode

Manipulated var. limitation: □ Limitation On

Drive: max. motor speed: 2000.0 rpm

Servo gain factor: 1/s: 1.0

□ Invert manipulated variable

□ Rev. of direct. act.val.

□ Drift compensation

Fine interpolator: Constant-acceleration interpolation

Dynamic response filter: □ Dynamic response filter On

Precontrol: ☑ Precontrol On, Weighting factor 100.0 %

Balancing filter: Balancing filter active

Figure 8-11 Block diagram, closed-loop control under TO Gantry_1

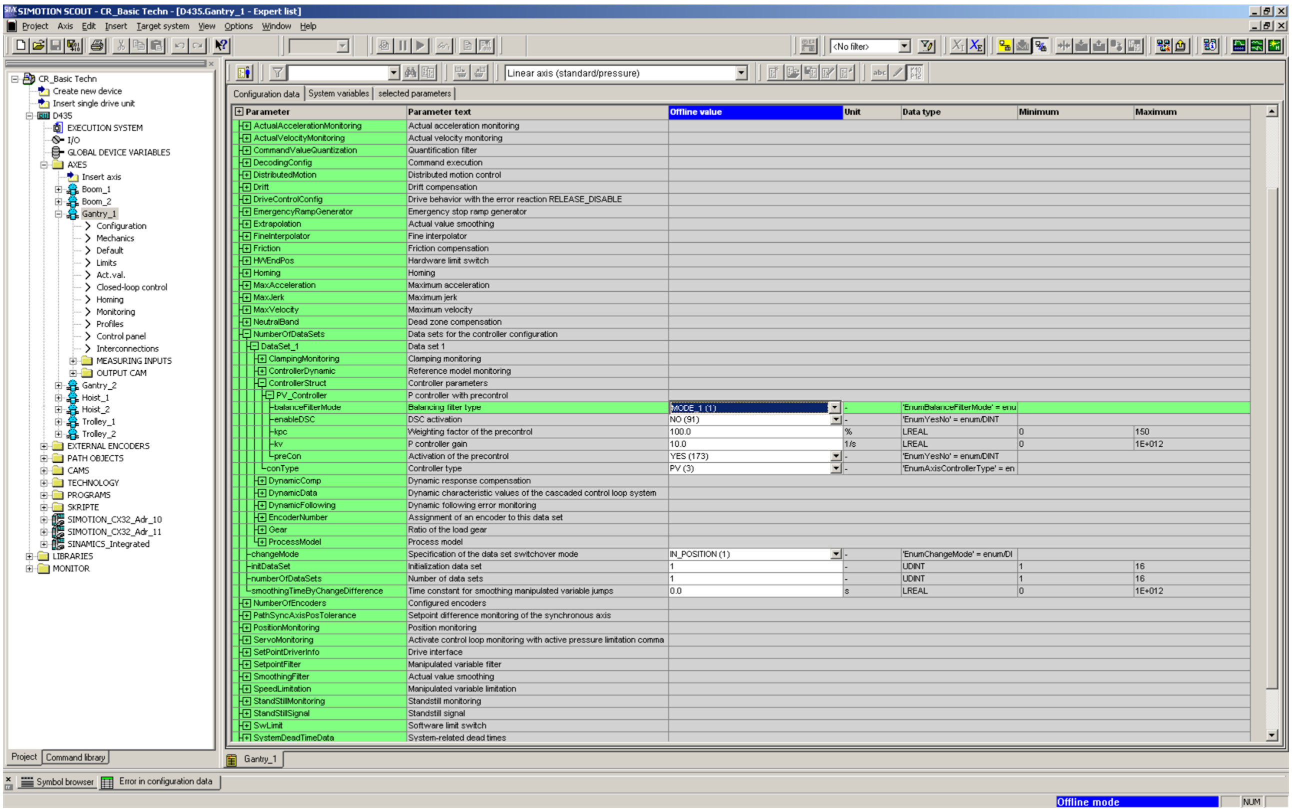
The speed pre-control also has to be activated: (Gantry_1.TypeOfAxis.NumberOfDataSets.DataSet_1.ControllerStruct.PV_Controller.preCon = YES)

Plus, MODE_1 must be set as the balancing filter type: (Gantry_1.TypeOfAxis.NumberOfDataSets.DataSet_1.ControllerStruct.PV_Controller.balanceFilterMode = MODE_1)

The interpolator (Gantry_1.TypeOfAxis.FineInterpolator._type) should be set to constant-acceleration interpolation (cubic).

NOTICE

The balancing filter must be set to Mode_1 to ensure proper operation of the position control.



Parameter	Parameter text	Offline value	Unit	Data type	Minimum	Maximum
ActualAccelerationMonitoring	Actual acceleration monitoring					
ActualVelocityMonitoring	Actual velocity monitoring					
CommandValueQuantization	Quantification filter					
DecodingConfig	Command execution					
DistributedMotion	Distributed motion control					
Drift	Drift compensation					
DriveControlConfig	Drive behavior with the error reaction RELEASE_DISABLE					
EmergencyRampGenerator	Emergency stop ramp generator					
Extrapolation	Actual value smoothing					
FineInterpolator	Fine interpolator					
Friction	Friction compensation					
HWEndPos	Hardware limit switch					
Homing	Homing					
MaxAcceleration	Maximum acceleration					
MaxJerk	Maximum jerk					
MaxVelocity	Maximum velocity					
NeutralBand	Dead zone compensation					
NumberOfDataSets	Data sets for the controller configuration					
DataSet_1	Data set 1					
ClampingMonitoring	Clamping monitoring					
ControllerDynamic	Reference model monitoring					
ControllerStruct	Controller parameters					
PV_Controller	P controller with precontrol					
balanceFilterMode	Balancing filter type	MODE_1 (1)	-	'EnumBalanceFilterMode' = enu		
enableDSC	DSC activation	NO (91)	-	'EnumYesNo' = enum/DINT		
kpc	Weighting factor of the precontrol	100.0	%	LREAL	0	150
kv	P controller gain	10.0	1/s	LREAL	0	1E+012
preCon	Activation of the precontrol	YES (173)	-	'EnumYesNo' = enum/DINT		
conType	Controller type	PV (3)	-	'EnumAxisControllerType' = en		
DynamicComp	Dynamic response compensation					
DynamicData	Dynamic characteristic values of the cascaded control loop system					
DynamicFollowing	Dynamic following error monitoring					
EncoderNumber	Assignment of an encoder to this data set					
Gear	Ratio of the load gear					
ProcessModel	Process model					
changeMode	Specification of the data set switchover mode	IN_POSITION (1)	-	'EnumChangeMode' = enum/DI		
initDataSet	Initialization data set	1	-	UDINT	1	16
numberOfDataSets	Number of data sets	1	-	UDINT	1	16
smoothingTimeByChangeDifference	Time constant for smoothing manipulated variable jumps	0.0	s	LREAL	0	1E+012
NumberOfEncoders	Configured encoders					
PathSyncAxisPosTolerance	Setpoint difference monitoring of the synchronous axis					
PositionMonitoring	Position monitoring					
ServoMonitoring	Activate control loop monitoring with active pressure limitation comma					
SetPointDriverInfo	Drive interface					
SetpointFilter	Manipulated variable filter					
SmoothingFilter	Actual value smoothing					
SpeedLimitation	Manipulated variable limitation					
StandStillMonitoring	Standstill monitoring					
StandStillSignal	Standstill signal					
SwLimit	Software limit switch					
SystemDeadTimeData	System-related dead times					

Figure 8-12 Block diagram, configuration data Gantry_1

Script files

The Com_Drive script file is located in the script folder of each drive. Start this script with **Accept and execute**. The script file is performed and all necessary communication connections created.

Note

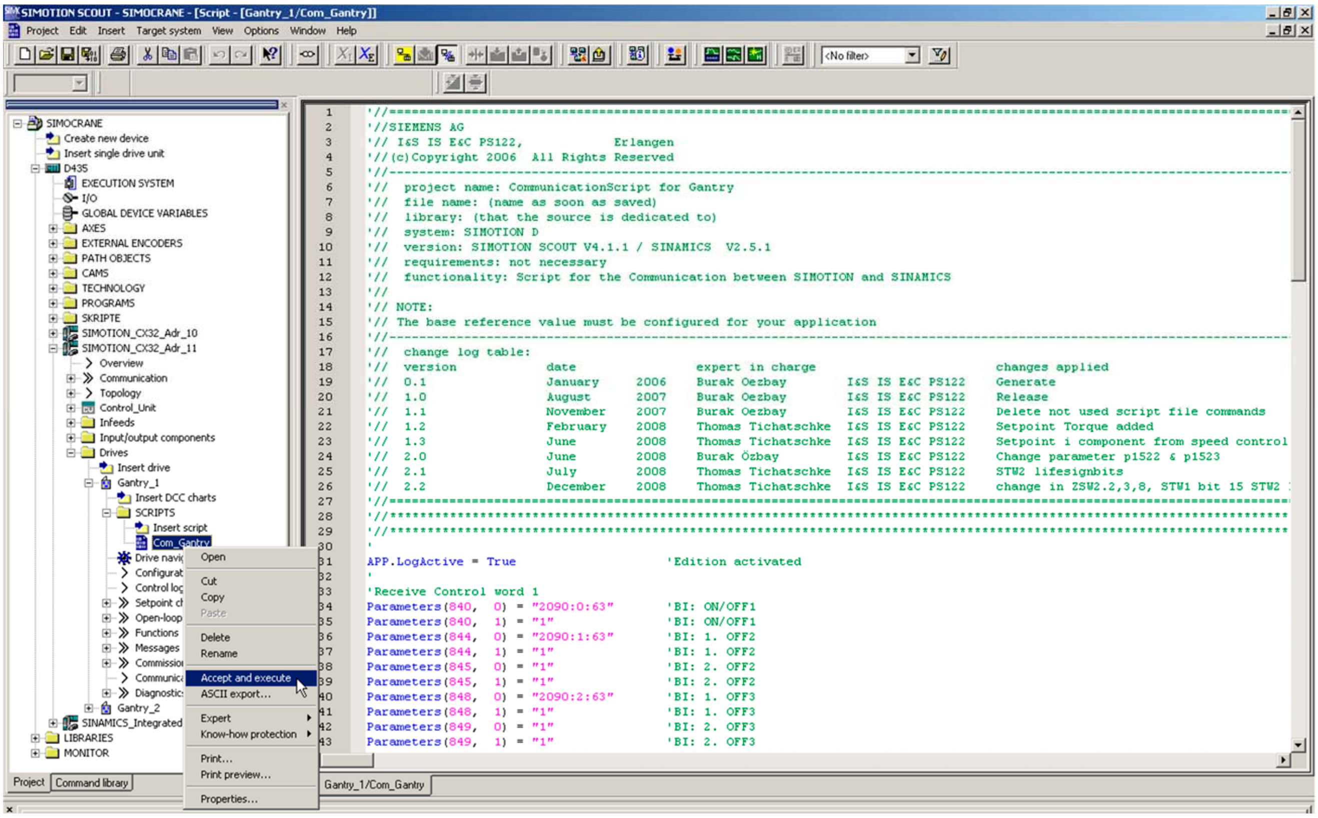
The script file should be executed offline. The changes then have to be downloaded into the control system.

CAUTION

The reference values (p2000, p2001, p2002, p2003, r2004) are checked. They must be the same for the master and slave drives and may need to be customized.

In addition, you must yourself for the slave (e.g. Gantry_2) connect the torque setpoint r79 of the master (e.g. Gantry_1) to the drive parameter p1503; the script does not perform this change.

The current status of the script execution is shown in the output window.



```
'//==========================================================================================================
'//SIEMENS AG
'// I&S IS E&C PS122,            Erlangen
'//(c)Copyright 2006  All Rights Reserved
'//----------------------------------------------------------------------------------------------------------
'//  project name: CommunicationScript for Gantry
'//  file name: (name as soon as saved)
'//  library: (that the source is dedicated to)
'//  system: SIMOTION D
'//  version: SIMOTION SCOUT V4.1.1 / SINAMICS  V2.5.1
'//  requirements: not necessary
'//  functionality: Script for the Communication between SIMOTION and SINAMICS
'//
'// NOTE:
'// The base reference value must be configured for your application
'//----------------------------------------------------------------------------------------------------------
'//  change log table:
'//  version            date                expert in charge                        changes applied
'//  0.1                January    2006     Burak Oezbay        I&S IS E&C PS122    Generate
'//  1.0                August     2007     Burak Oezbay        I&S IS E&C PS122    Release
'//  1.1                November   2007     Burak Oezbay        I&S IS E&C PS122    Delete not used script file commands
'//  1.2                February   2008     Thomas Tichatschke  I&S IS E&C PS122    Setpoint Torque added
'//  1.3                June       2008     Thomas Tichatschke  I&S IS E&C PS122    Setpoint i component from speed control
'//  2.0                June       2008     Burak Özbay         I&S IS E&C PS122    Change parameter p1522 & p1523
'//  2.1                July       2008     Thomas Tichatschke  I&S IS E&C PS122    STW2 lifesignbits
'//  2.2                December   2008     Thomas Tichatschke  I&S IS E&C PS122    change in ZSW2.2,3,8, STW1 bit 15 STW2 
'//==========================================================================================================
'//**********************************************************************************************************
'//**********************************************************************************************************
'
APP.LogActive = True                    'Edition activated
'
'Receive Control word 1
Parameters(840,  0) = "2090:0:63"       'BI: ON/OFF1
Parameters(840,  1) = "1"               'BI: ON/OFF1
Parameters(844,  0) = "2090:1:63"       'BI: 1. OFF2
Parameters(844,  1) = "1"               'BI: 1. OFF2
Parameters(845,  0) = "1"               'BI: 2. OFF2
Parameters(845,  1) = "1"               'BI: 2. OFF2
Parameters(848,  0) = "2090:2:63"       'BI: 1. OFF3
Parameters(848,  1) = "1"               'BI: 1. OFF3
Parameters(849,  0) = "1"               'BI: 2. OFF3
Parameters(849,  1) = "1"               'BI: 2. OFF3
```

Figure 8-13 Accepting and executing the script

Note

Before starting any travel, the torque limits should be checked in SINAMICS for boom operation (r1526: Torque limit, upper And r1527: Torque limit, lower).

CraneAlarm

Alarms not necessary for the basic technology should be suppressed. This is done using the script file "CraneAlarm". The script is located in the script folder under SIMOTION D435.

The script procedure for SINAMICS corresponds to the procedure for SIMOTION. The communication script is located in the script folder under the drive object.

Parameter_Transfer

Parameter_Transfer is used to match the reference data between the technology object, the drive object and the unit-global variables. The script is executed as for CraneAlarm (right-click to mark the script and select "Accept and execute"). The Parameter_Transfer script is located in the script folder under SIMOTION D435.

Use this script to fetch the associated drive parameters (p2000, p2001, p2002, p2003, r2004) and enter in the reference values of the relevant MCC source in the X_nominalVelocity, X_nominalCurrent, X_nominalTorque, X_nominalVoltage, X_nominalPower variables. In addition, the reference variable of the associated axis is set to the value entered in p2000.

8.4 Controlling the crane taking the PROFIdrive profile into account

There follows an example of how to control the Active Line Module (Crane ON OFF) and SPEED_CONTROLLED operating mode (SPEED_CONTROLLED ON OFF).

8.4.1 Switching on the crane

Make the crane ready to start by proceeding as follows:

- Switch on the crane, the infeed, and then the line contactor.

If the infeed is precharged, the pulse enable will be activated. "Standby" is signaled back after one program cycle and the drive is ready for a drive command to be selected.

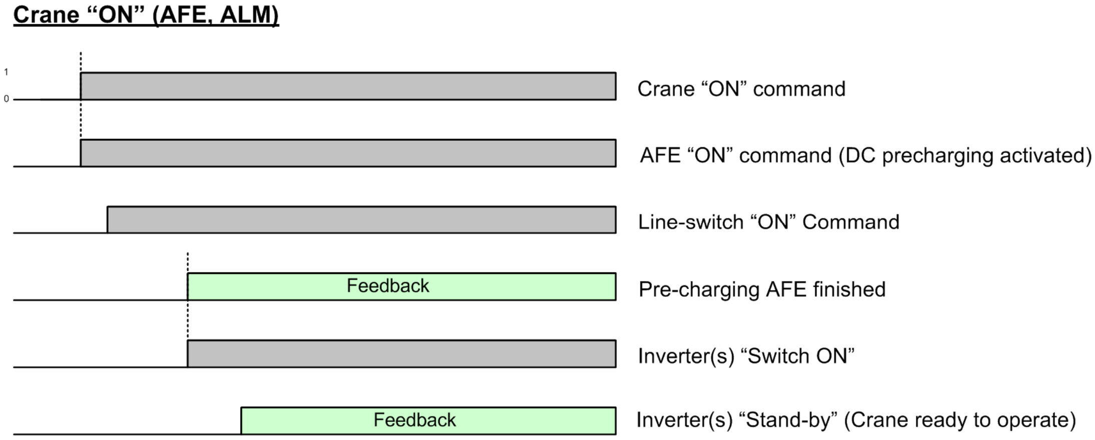

Crane "ON" (AFE, ALM)
1
0
Crane "ON" command
AFE "ON" command (DC precharging activated)
Line-switch "ON" Command
Feedback
Pre-charging AFE finished
Inverter(s) "Switch ON"
Feedback
Inverter(s) "Stand-by" (Crane ready to operate)

Figure 8-14 Switching on the crane

8.4.2 Switching off the crane

Switch the crane off as follows:

- "Crane off" command and shutdown of converters using OFF1 and OFF2

After one program cycle a signal is output indicating that the crane is ready to start again. The infeed is then shut down, followed by the line contactor. After one further cycle, the infeed is ready to start again.

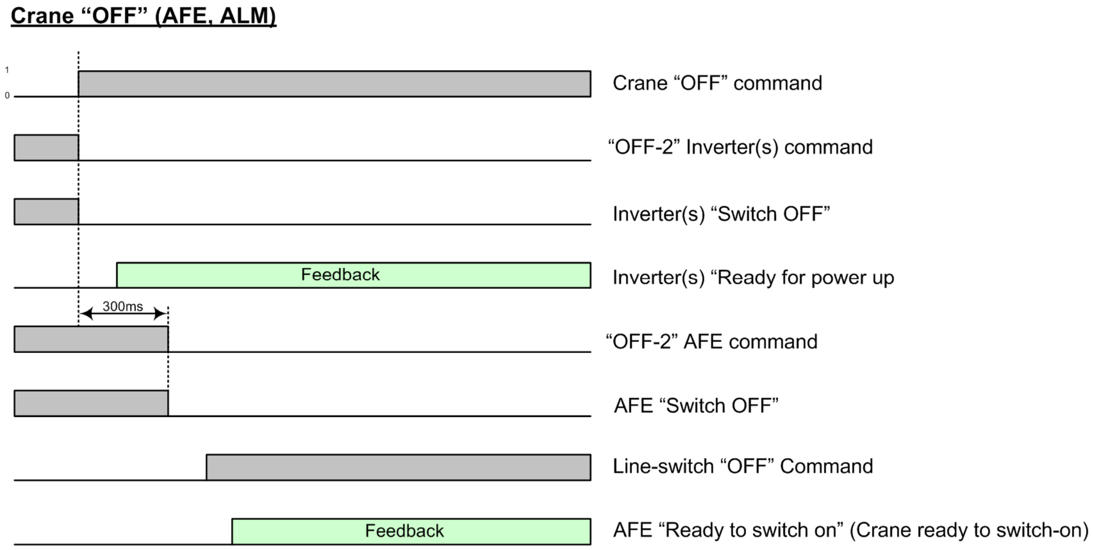

Crane “OFF” (AFE, ALM)
1
0
Crane “OFF” command
“OFF-2” Inverter(s) command
Inverter(s) “Switch OFF”
Feedback
Inverter(s) “Ready for power up
300ms
“OFF-2” AFE command
AFE “Switch OFF”
Line-switch “OFF” Command
Feedback
AFE “Ready to switch on” (Crane ready to switch-on)

Figure 8-15 Switching off the crane

8.4.3 SPEED_CONTROLLED ON

Select a motion task as follows:

- Select SPEED_CONTROLLED operating mode.
- Select OFF2 and OFF3.

The operating mode is signaled back. OFF2 and OFF3 checkback signals are output after one program cycle. "Ready to power up" is also output. These signals switch OFF1. After one further cycle, "Ready" is signaled.

The drive command, the pulse enable, and the speed controller enable magnetize the motor. As soon as magnetization is complete, operation is enabled and the brake is released.

If "Brake released" is signaled back, the ramp-function generator and the setpoint can be enabled.

A positive or negative command causes the drive motion to start after one program cycle and a signal is sent back indicating that the axis is in motion.

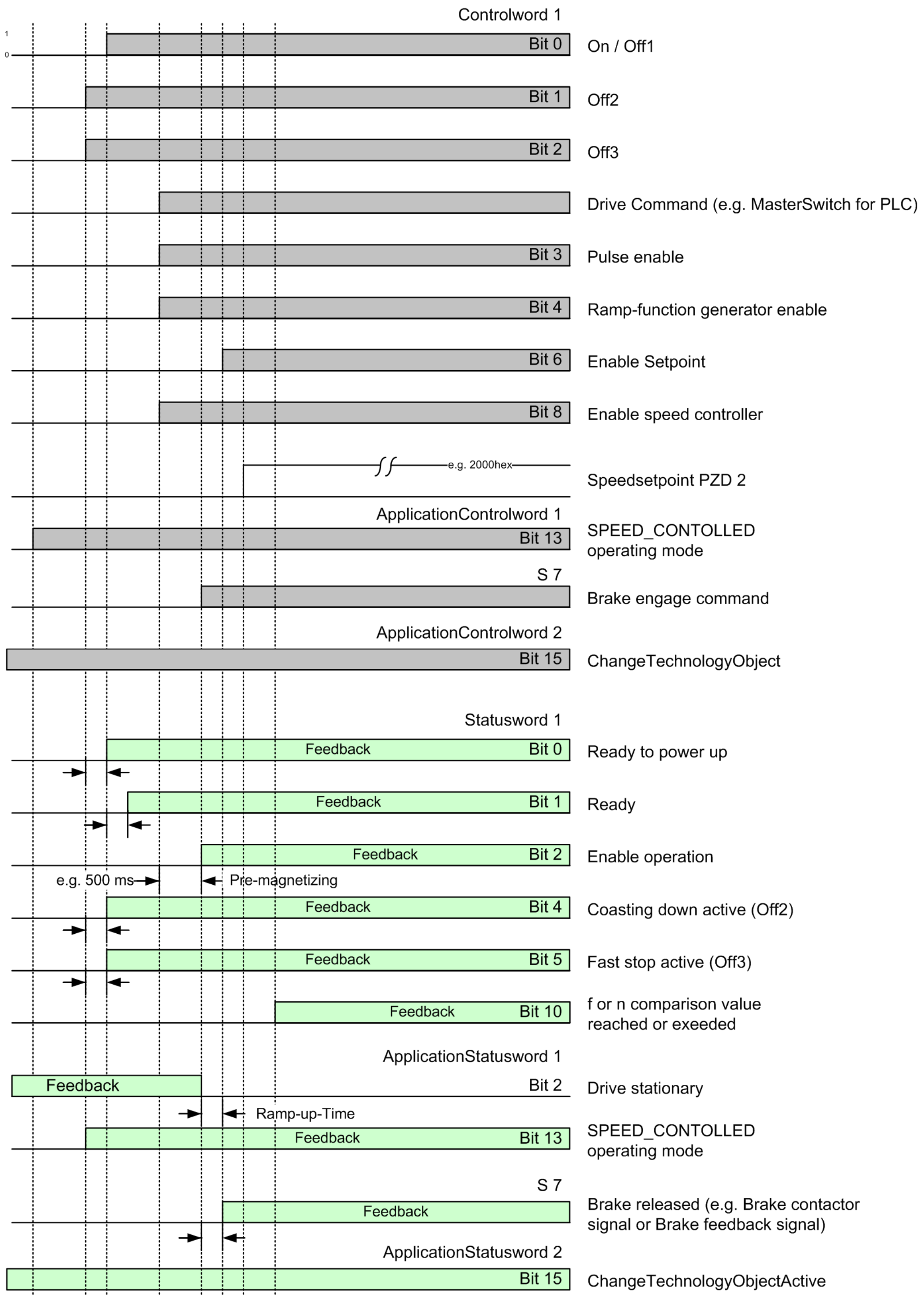


Figure 8-16 SPEED_CONTROLLED ON

8.4.4 SPEED_CONTROLLED OFF

Deselect a motion task as follows:

- Cancel the positive or negative command and the motion task.

Once the ramp-down time has elapsed, the checkback signals "Direction of travel positive or negative" and "f or n comparison value reached or exceeded" are reset. The brake closes.

When the "Brake closed" message is output, the pulse enable is canceled.

After one program cycle, a low signal is issued back at "Enable operation" and OFF1 is canceled.

After one further cycle the drive is no longer ready to start.

Deactivating OFF2 and OFF3 disables the "ready to start" condition and checkback signals OFF2 and OFF3 switch to the low level.

The SPEED_CONTROLLED operating mode can then be deselected.

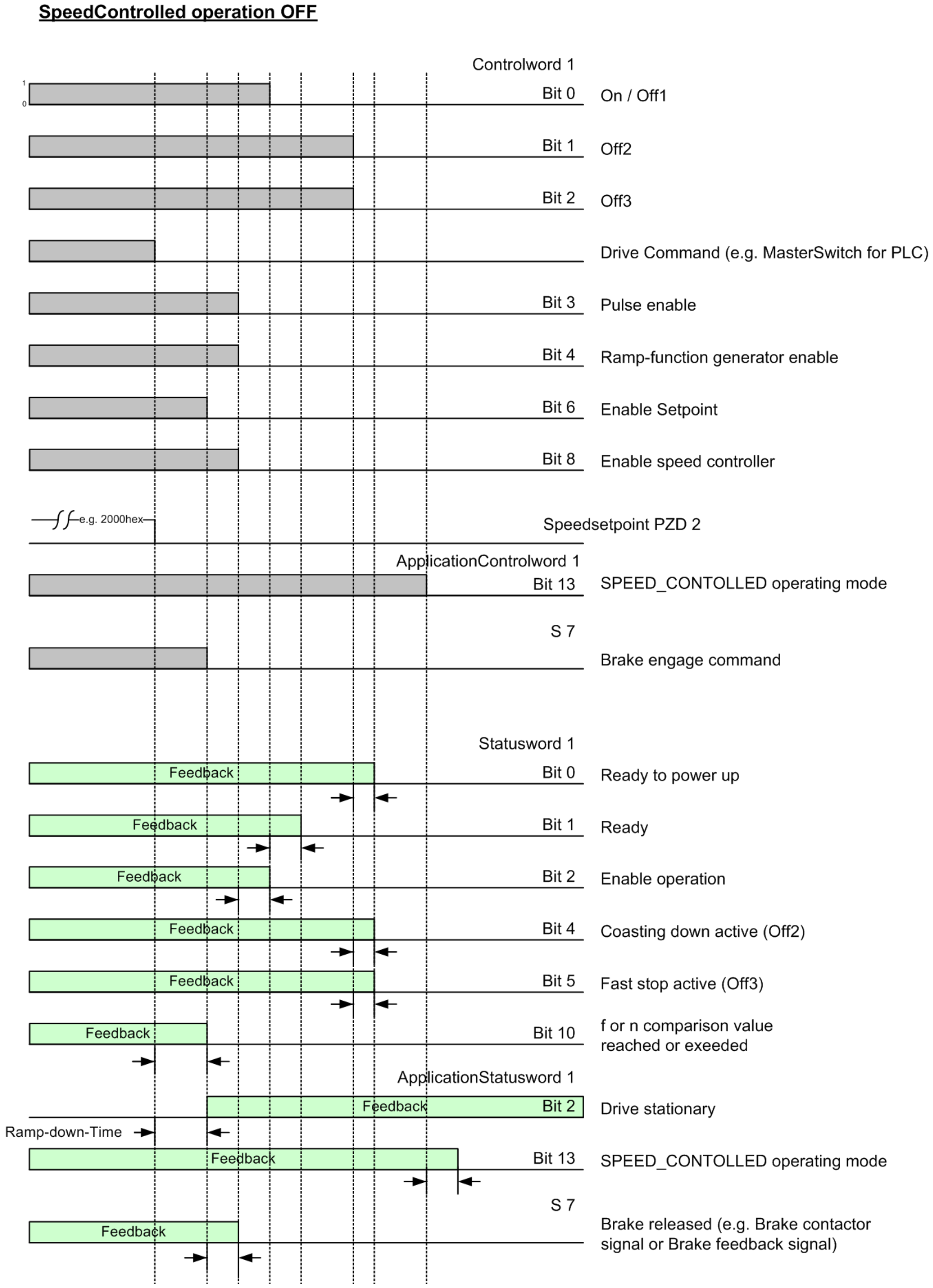


Figure 8-17 SPEED_CONTROLLED OFF

8.4.5 Mechanical motion stop

The S7 transmits the signal "Mechanical stop". At the same time, the commands OFF2, "Close brake", "Disable ramp-function generator", and "Inhibit pulse enable" are activated.

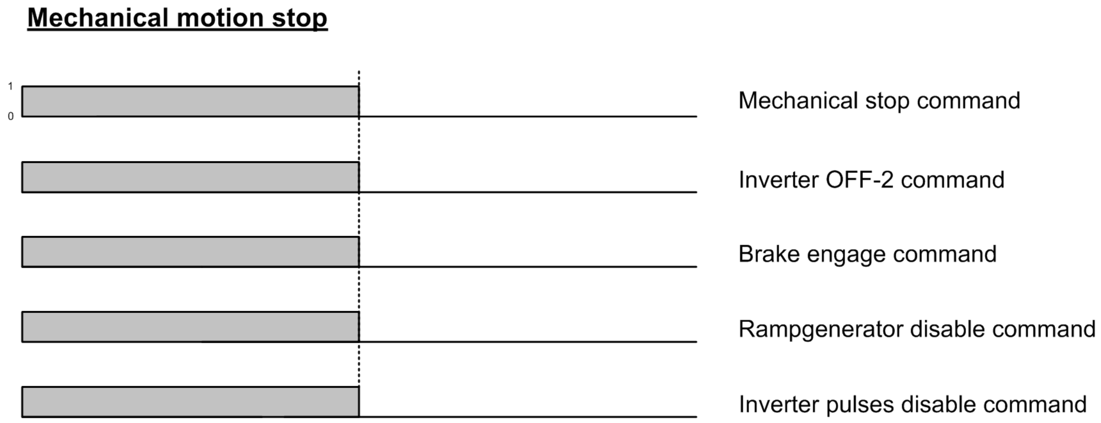


Figure 8-18 Mechanical stop

8.5 Application examples for the S7 control

For the application examples of the S7 controller, the following flags are used.

A distinction is made between the various main drives using the thousands and hundreds range of the flags.

- Trolley_1 is the flag address area 2100.0 to 2117.7
- Boom_1 is the flag address area 4100.0 to 4117.7
- HoldingGear is the flag address area 1100.0 to 1117.7
- ClosingGear is the flag address area 1300.0 to 1317.7

Table 8- 1 Control word 1

M2100.0	=	On / Off1
M2100.1	=	Off2
M2100.2	=	Off3
M2100.3	=	Pulse enable
M2100.4	=	Ramp-function generator enable
M2100.6	=	Setpoint enable
M2100.7	=	Acknowledge fault
M2101.0	=	Speed controller enable
M2101.2	=	Master control by PLC

Table 8- 2 Application control word 1

M2104.7	=	Selecting synchronous operation
M2105.2	=	AUTOMATIC operating mode
M2105.3	=	MANUAL operating mode
M2105.5	=	SPEED_CONTROLLED operating mode
M2105.6	=	SENSORLESS_EMERGENCY operating mode
M2105.7	=	SWAY_CONTROL operating mode

Table 8- 3 Application control word 2

M2106.4	=	Enable slack rope controller
M2106.5	=	Command, save "Grab open"
M2106.6	=	Command, save "Grab closed"
M2106.7	=	Select orange-peel bucket
M2107.0	=	Enable current equalization controller
M2107.1	=	Select grab change
M2107.2	=	Homing
M2107.5	=	SlaveMode
M2107.7	=	ChangeTechnologyObject

Table 8- 4 Status word 1

M2110.0	=	Ready to switch-on
M2110.1	=	Ready
M2110.2	=	Operation enabled
M2110.3	=	Fault present
M2110.4	=	Coast down active
M2110.5	=	Quick stop active
M2111.1	=	Control requested to the PLC
M2111.2	=	f or n comparison value reached

Table 8- 5 Application status word 1

M2114.0	=	Axis moves in the positive direction
M2114.1	=	Axis moves in the negative direction
M2114.7	=	Message, "synchronous operation active"
M2115.2	=	AUTOMATIC operating mode
M2115.3	=	MANUAL operating mode
M2115.5	=	SPEED_CONTROLLED operating mode
M2115.6	=	SENSORLESS_EMERGENCY operating mode
M2115.7	=	SWAY_CONTROL operating mode

Table 8- 6 Application status word 2

M2116.0	=	Grab 1/2 open
M2116.2	=	Grab 2/3 closed
M2116.4	=	Message, "grab open"
M2116.5	=	Message, "grab closed"
M2117.2	=	Message, "homed"
M2117.5	=	SlaveModeActive
M2117.7	=	TechnologyObjectActive

8.5.1 Controlling the drive

The command sequences that the controller uses to switch-on and switch-off the main drive are shown in the following example.

Note

- In this example, the brake control is not considered; the user must program this himself. Another precondition is that there is no fault present.
- "Flag" is abbreviated with the letter M.

NOTICE

To activate the technology object, bit 15 "ChangeTechnologyObject" in application control word 2 must be "TRUE". The technology object is deactivated with "FALSE".

8.5.1.1 Switching-on a drive

The drive is switched-off and there is no fault/error present. The drive is now switched-on in steps.

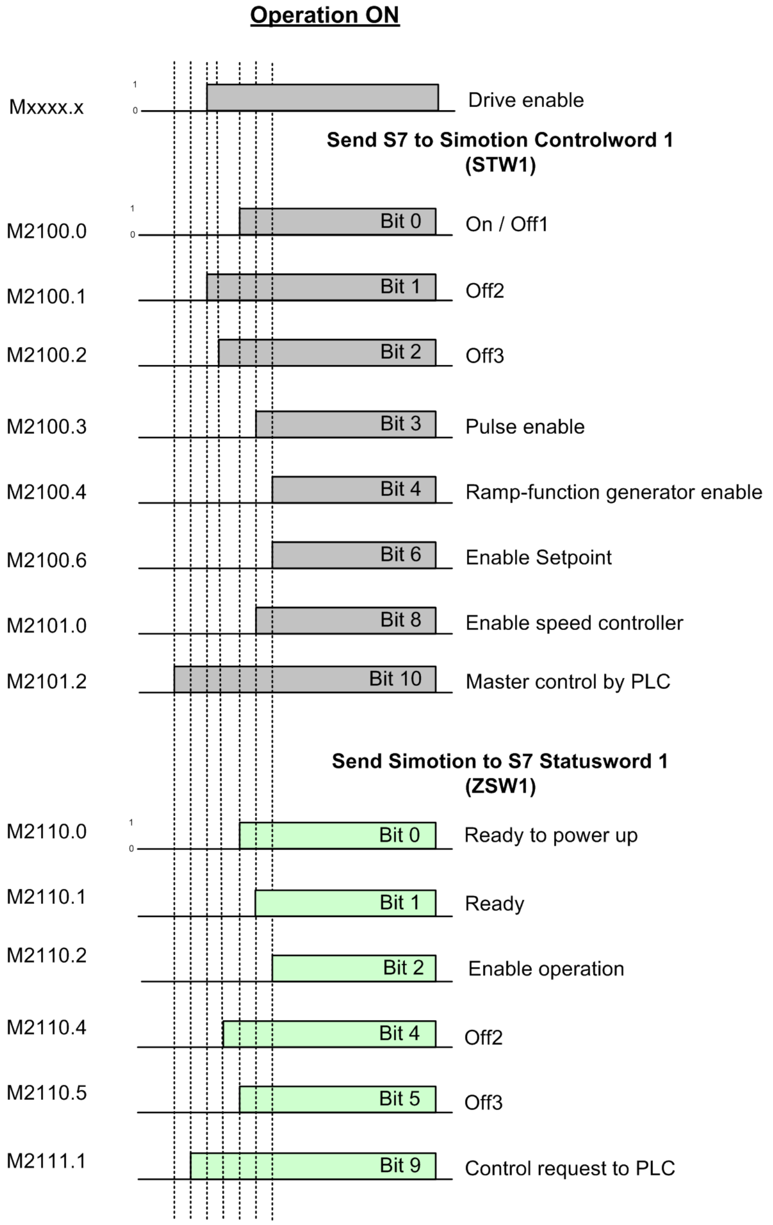


Figure 8-19 Switching-on a drive

- The flag 2101.2 "Master control by the PLC" must be set so that the drive accepts all of the commands from the higher-level control. A check is then made as to whether flag 2111.1 "Control requested to the PLC" is set.
- The switch-on command is received from the higher-level control. In this case, flag 2100.1 "Off2" must be set. A check is then made as to whether flag 2110.4 "Coast down active" has been reset.

- The flag 2100.2 "Off3" must be set. A check is then made as to whether flag 2110.4 "Quick stop active" has been reset.
- The flag 2100.0 "Off1" must be set. A check is then made as to whether flag 2110.0 "Ready to switch-on" and flag 2110.1 "Ready" are set.
- The flag 2100.3 "Pulse enable" and flag 2101.0 "Speed control enable" must be simultaneously set. A check is then made as to whether flags 2110.0 "Ready to switch-on", 2110.1 "Ready" and 2110.2 "Enable operation" are set.
- The flag 2100.4 "Ramp-function generator enable" and flag 2100.6 "Setpoint enable" must be simultaneously set. A setpoint can then be entered.

8.5.1.2 Switching-off a drive

The drive is moving and the setpoint from the master switch is set to 0.

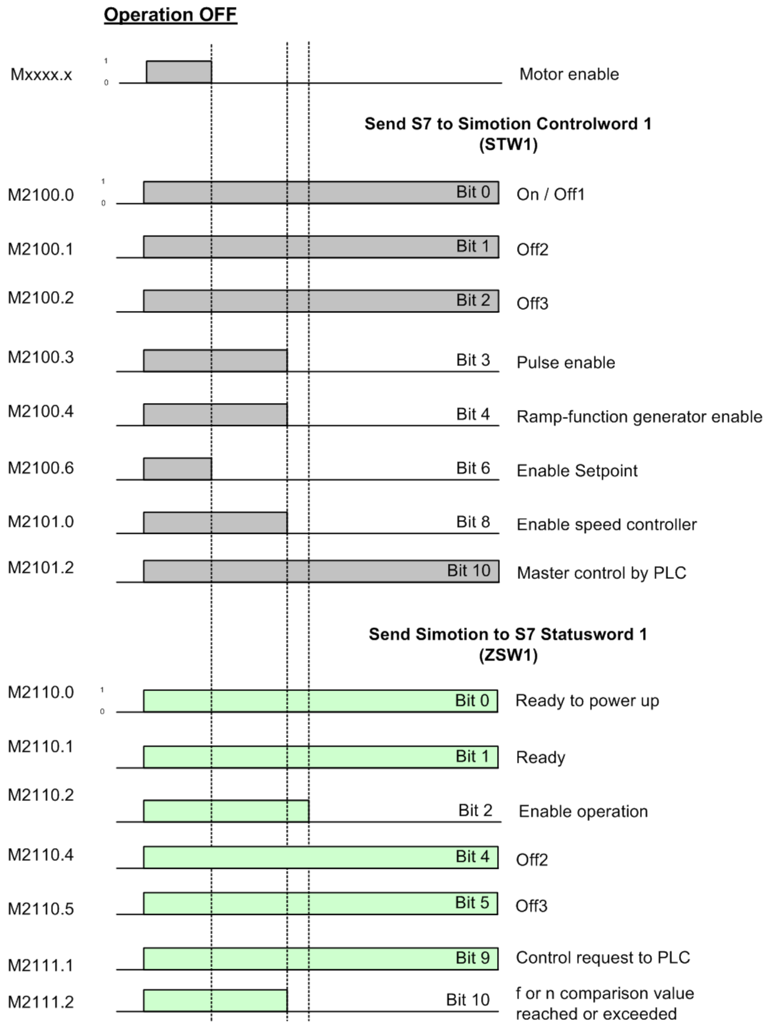


Figure 8-20 Switching-off a drive

- The switch-on command from the higher-level control is reset. The flag 2100.6 "Setpoint enable" must be reset. A check is then made as to whether flag 2111.2 "f or n comparison value reached" has been reset.
- The flags 2100.3 "Pulse enable", 2100.4 "Ramp function generator enable" and 2100.8 "Speed controller enable" must be simultaneously reset. A check is then made as to whether flags 2110.0 "Ready to switch-on" and 2110.1 "Ready" are set and whether flag 2110.2 "Enable operation" is reset.

8.5.2 Selecting and deselecting operating modes

This chapter describes using an example how the various operating modes for a main drive are selected and deselected.

Note

To switch-on or switch-off a drive, refer to Chapter Controlling the drive (Page 376).

NOTICE
To activate the technology object, bit 15 "ChangeTechnologyObject" in application control word 2 must be "TRUE". The technology object is deactivated with "FALSE".

8.5.2.1 Selecting the AUTOMATIC operating mode

The drive is switched-off and no operating mode has been selected.

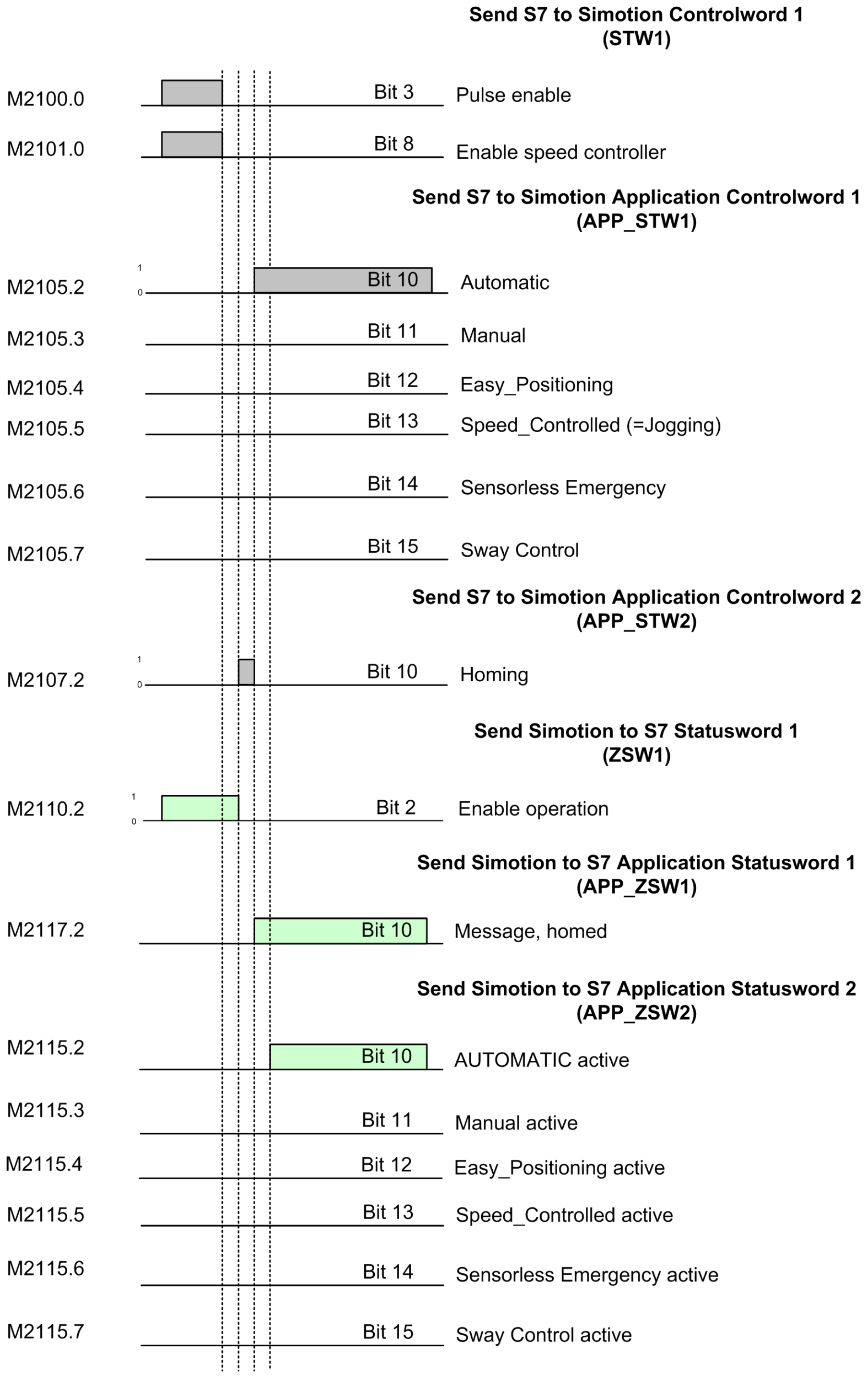


Figure 8-21 Selecting the AUTOMATIC operating mode

- Flag 2110.2 "Enable operation" is not set.
- Flag 2107.2 "Homing" must be set using a positive signal edge. A check is then made whether flag 2117.2 "Homed message" has been set.
- Flag 2105.2 "AUTOMATIC operating mode" is set.
- It is not permissible that flags 2105.3 "MANUAL operating mode", 2105.4 "Easy-Positioning operating mode", 2105.5 "SPEED_CONTROLLED operating mode", 2105.6 "SENSORLESS_EMERGENCY" and 2105.7 "SWAY_CONTROL operating mode" are set.
- A check is then made whether flag 2115.2 "AUTOMATIC operating mode" is set and whether flags 2115.3 "MANUAL operating mode", 2115.4 "Easy-Positioning operating mode", 2115.5 "SPEED_CONTROLLED operating mode", 2115.6 "SENSORLESS_EMERGENCY" and 2115.7 "SWAY_CONTROL operating mode" are not set.

8.5.2.2 Deselecting the AUTOMATIC operating mode

The drive is switched-off and the AUTOMATIC operating mode is active.

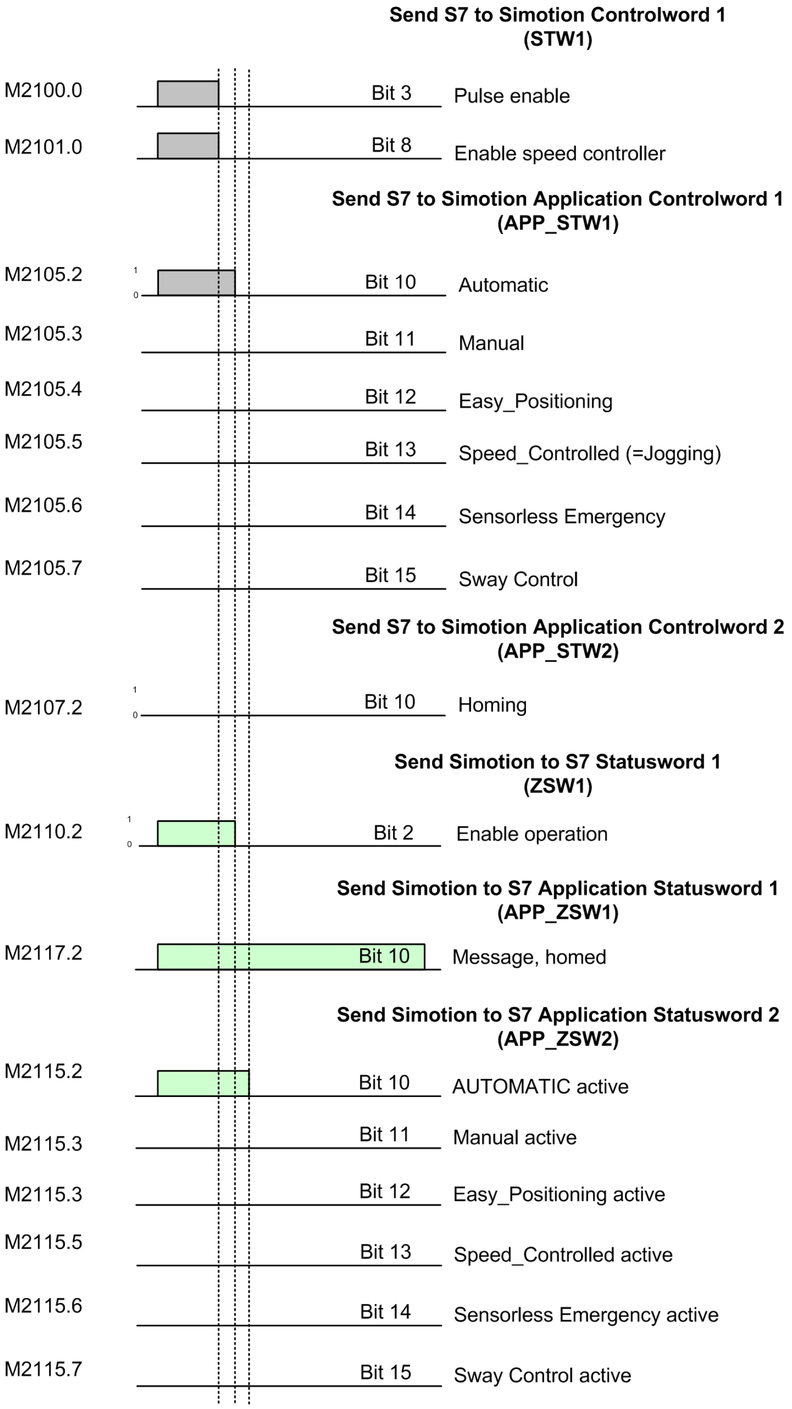


Figure 8-22 Deselecting the AUTOMATIC operating mode

- Flag 2110.2 "Enable operation" is not set.
- Flag 2105.2 "AUTOMATIC operating mode" is reset.
- It is not permissible that flags 2105.3 "MANUAL operating mode", 2105.4 "Easy-Positioning operating mode", 2105.5 "SPEED_CONTROLLED operating mode", 2105.6 "SENSORLESS_EMERGENCY" and 2105.7 "SWAY_CONTROL operating mode" are set.
- A check is then made whether flag 2115.2 "AUTOMATIC operating mode" is reset and whether the flags 2115.3 "MANUAL operating mode", 2115.4 "Easy-Positioning operating mode", 2115.5 "SPEED_CONTROLLED operating mode", 2115.6 "SENSORLESS_EMERGENCY" and 2115.7 "SWAY_CONTROL operating mode" are not set.

8.5.3 Externally switching over from trolley to boom and vice versa

Using an example, a description will be given on switching over from trolley to boom and vice versa.

Note

- In this example, the brake control will not be considered; the user must program this himself.
- The "SPEED_CONTROLLED" operating mode has already been selected.

8.5.3.1 Switching over from trolley to boom

The drive is at a standstill (zero speed) and is switched-off.

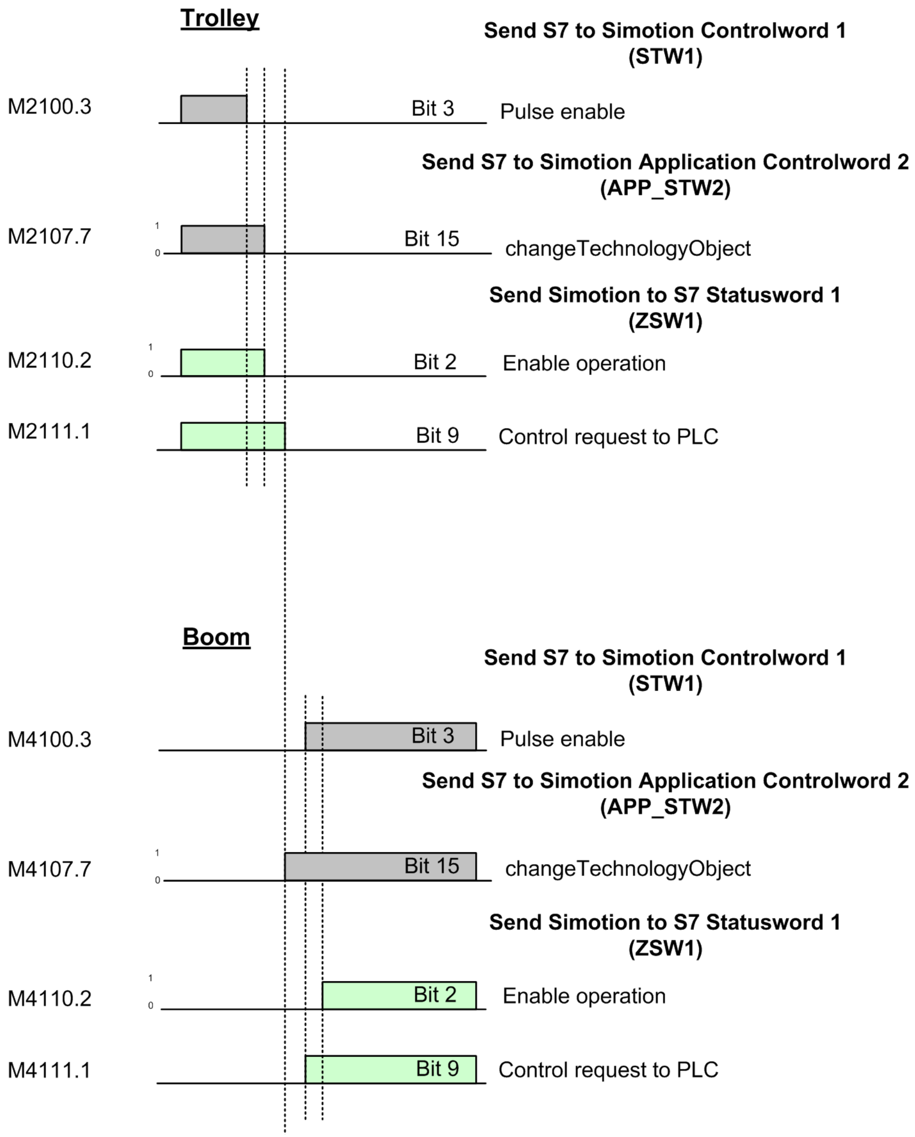

Trolley
Send S7 to Simotion Controlword 1 (STW1)
M2100.3
Bit 3
Pulse enable
Send S7 to Simotion Application Controlword 2 (APP_STW2)
M2107.7
1
0
Bit 15
changeTechnologyObject
Send Simotion to S7 Statusword 1 (ZSW1)
M2110.2
1
0
Bit 2
Enable operation
M2111.1
Bit 9
Control request to PLC
Boom
Send S7 to Simotion Controlword 1 (STW1)
M4100.3
Bit 3
Pulse enable
Send S7 to Simotion Application Controlword 2 (APP_STW2)
M4107.7
1
0
Bit 15
changeTechnologyObject
Send Simotion to S7 Statusword 1 (ZSW1)
M4110.2
1
0
Bit 2
Enable operation
M4111.1
Bit 9
Control request to PLC

Figure 8-23 Switching over from trolley to boom

- The flag 2110.2 "Enable operation" for trolley is not set.
- The flag 2107.7 "ChangeTechnologyObject" must be reset. A check is then made as to whether flag 2111.1 "Control requested to the PLC" for trolley is reset.
- The flag 4107.7 "ChangeTechnologyObject" must be set. A check is then made as to whether flags 4111.1 "Control requested to the PLC" and 4117.7 " TechnologyObjectActive" for the boom are set.

8.5.3.2 Switching over from boom to trolley

The drive is at a standstill (zero speed) and is switched-off.

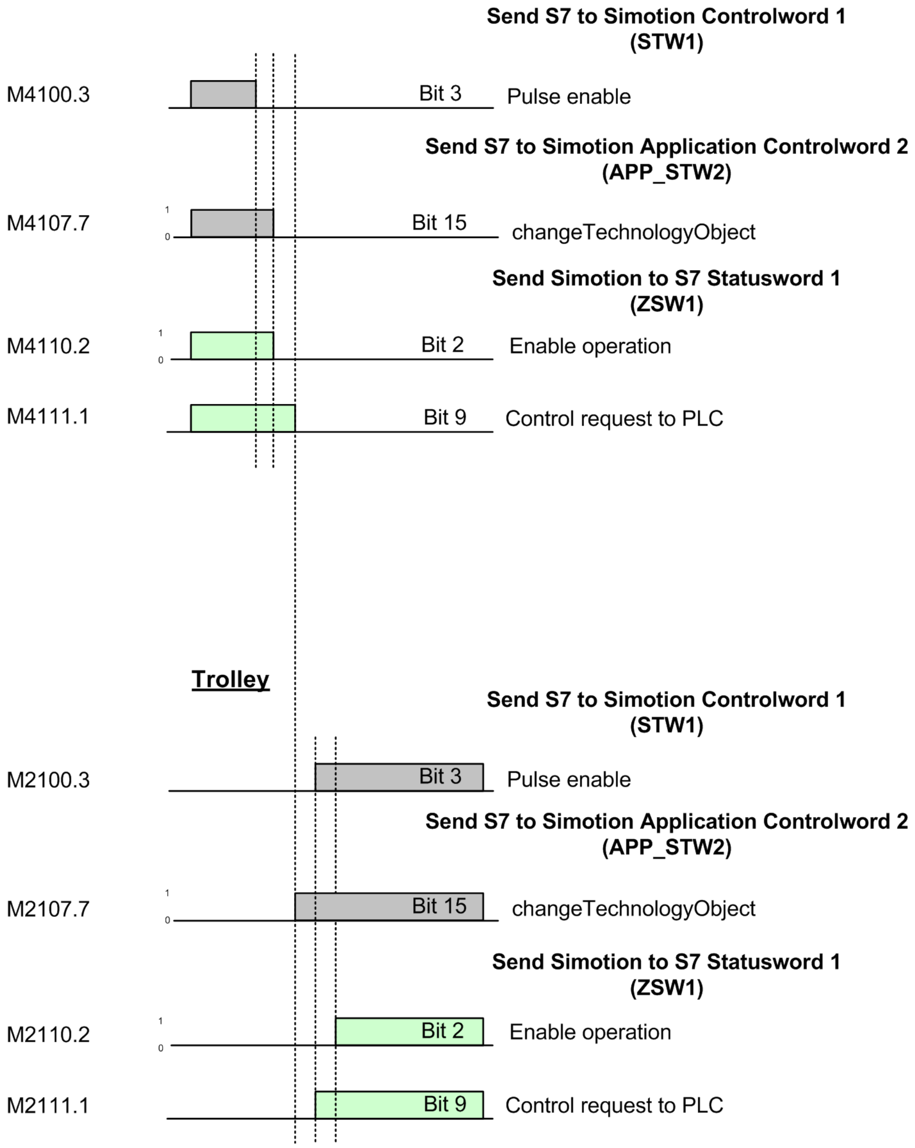


Figure 8-24 Switching over from boom to trolley

- The flag 4110.2 "Enable operation" for boom is not set.
- The flag 4107.7 "ChangeTechnologyObject" must be reset. A check is then made as to whether flag 4111.1 "Control requested to the PLC" for boom is reset.
- The flag 2107.7 "ChangeTechnologyObject" must be set. A check is then made as to whether flag 2111.1 "Control requested to the PLC" for trolley is set.

8.5.4 Grab crane

This chapter briefly describes the most important grab functions.

- In the holding gear, flag 1107.5 "SlaveMode" is not set - and therefore the "MasterMode" is active.
- Flag 1105.5 "SPEED_CONTROLLED operating mode" is set.
- In the closing gear, flag 1307.5 "SlaveMode" is set - and therefore the "SlaveMode" is active.
- Flag 1305.5 "SPEED_CONTROLLED operating mode" is set.
- In the closing gear, flag 1304.7 "Select synchronous operation" is set.
- The feedback flag 1314.7 "Synchronous operation active message" is set.

Note

In this example, the brake control is not considered; the user must program this himself.

Switching-on and switching-off the drive was already described in Chapter Controlling the drive (Page 376).

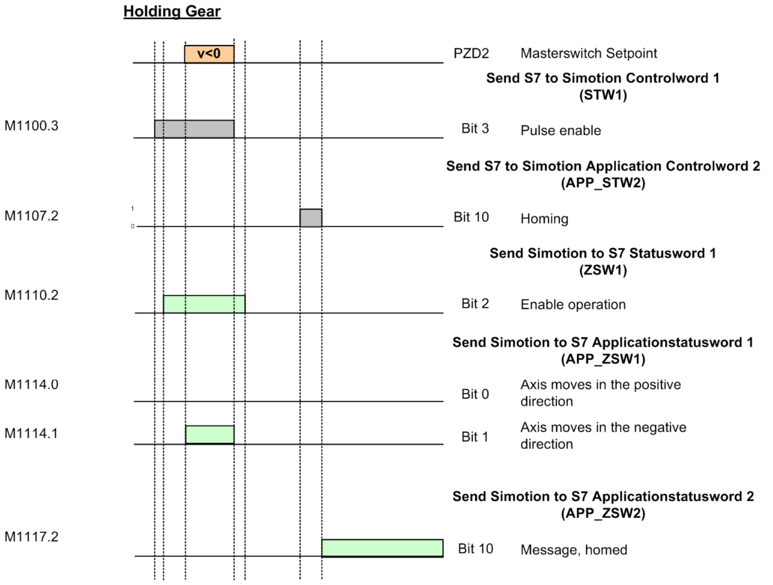
Selecting and deselecting the operating modes was already described in Chapter Selecting and deselecting operating modes (Page 379).

NOTICE

To activate the technology object, bit 15 "ChangeTechnologyObject" in application control word 2 must be "TRUE". The technology object is deactivated with "FALSE".

8.5.4.1 Adjusting the grab

Holding gear and closing gear are switched-on. The ropes have already been connected and the grab is attached to the ropes.


Holding Gear
v<0
PZD2 Masterswitch Setpoint
Send S7 to Simotion Controlword 1 (STW1)
M1100.3
Bit 3 Pulse enable
Send S7 to Simotion Application Controlword 2 (APP_STW2)
M1107.2
1
0
Bit 10 Homing
Send Simotion to S7 Statusword 1 (ZSW1)
M1110.2
Bit 2 Enable operation
Send Simotion to S7 Applicationstatusword 1 (APP_ZSW1)
M1114.0
Bit 0 Axis moves in the positive direction
M1114.1
Bit 1 Axis moves in the negative direction
Send Simotion to S7 Applicationstatusword 2 (APP_ZSW2)
M1117.2
Bit 10 Message, homed

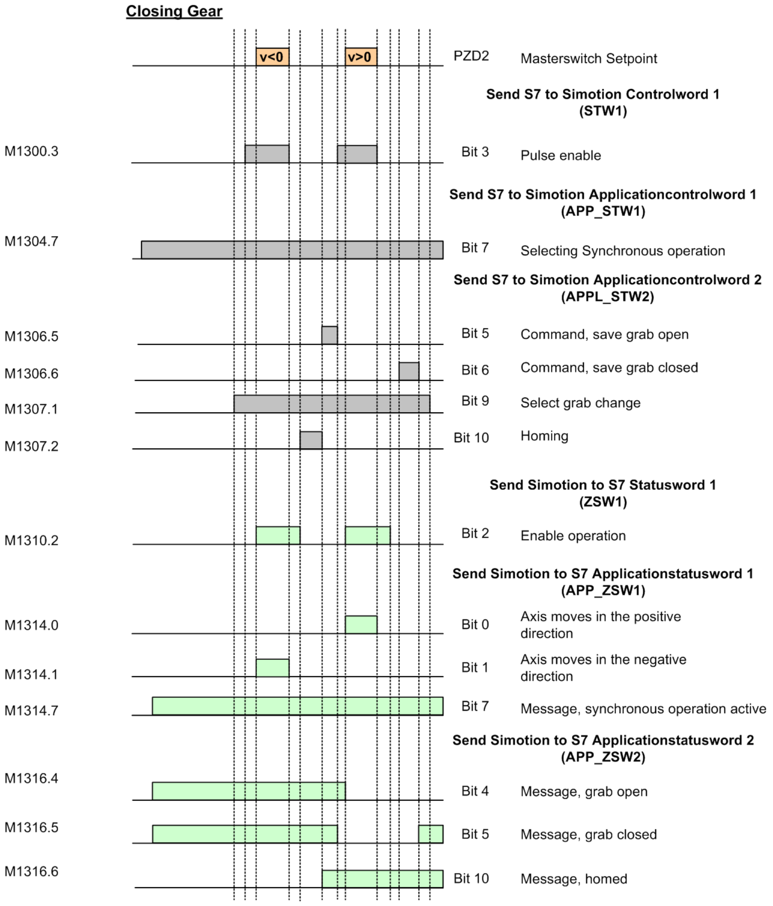

Closing Gear
v<0
v>0
PZD2 Masterswitch Setpoint
Send S7 to Simotion Controlword 1 (STW1)
M1300.3
Bit 3 Pulse enable
Send S7 to Simotion Applicationcontrolword 1 (APP_STW1)
M1304.7
Bit 7 Selecting Synchronous operation
Send S7 to Simotion Applicationcontrolword 2 (APPL_STW2)
M1306.5
Bit 5 Command, save grab open
M1306.6
Bit 6 Command, save grab closed
M1307.1
Bit 9 Select grab change
M1307.2
Bit 10 Homing
Send Simotion to S7 Statusword 1 (ZSW1)
M1310.2
Bit 2 Enable operation
Send Simotion to S7 Applicationstatusword 1 (APP_ZSW1)
M1314.0
Bit 0 Axis moves in the positive direction
M1314.1
Bit 1 Axis moves in the negative direction
M1314.7
Bit 7 Message, synchronous operation active
Send Simotion to S7 Applicationstatusword 2 (APP_ZSW2)
M1316.4
Bit 4 Message, grab open
M1316.5
Bit 5 Message, grab closed
M1316.6
Bit 10 Message, homed

Figure 8-25 Adjusting the grab

- The flags 1110.2 "Enable operation" for the holding gear and 1310.2 "Enable operation" for the closing gear are set.
- The flag 1116.4 "Grab open" and 1116.5 "Grab closed" are set, as the grab has still not been adjusted.
- The crane operator gives the command "Lift grab" until the limit switch switches-off the holding gear.
- The flag 1307.1 "Select grab change" is set.
- Using the master switch, the crane operator issues the command "Open grab" until the required position "Grab open" is reached.
- Using homing, the holding and closing gear are set to the position 0 mm; for this purpose, the flags 1107.2 "Homing" for the holding gear and 1307.2 "Homing" for the closing gear are set, each with a positive signal edge.
- The flag 1306.5 "Command, save grab open" is set. A check is then made as to whether flag 1316.4 "Grab open" is set and flag 1316.5 "Grab closed" is not set.
- Using the master switch, the crane operator issues the command "Close grab" until the grab is closed.
- When the grab is closed, flag 1306.6 "Command, save grab closed" is set. A check is then made as to whether flag 1316.4 "Grab open" is not set and flag 1316.5 "Grab closed" is set.
- The flag 1307.1 "Select grab change" is reset.

8.5.4.2 Slack rope controller function

The opened grab is on the material. The message "Grab touchdown" can be independently generated.

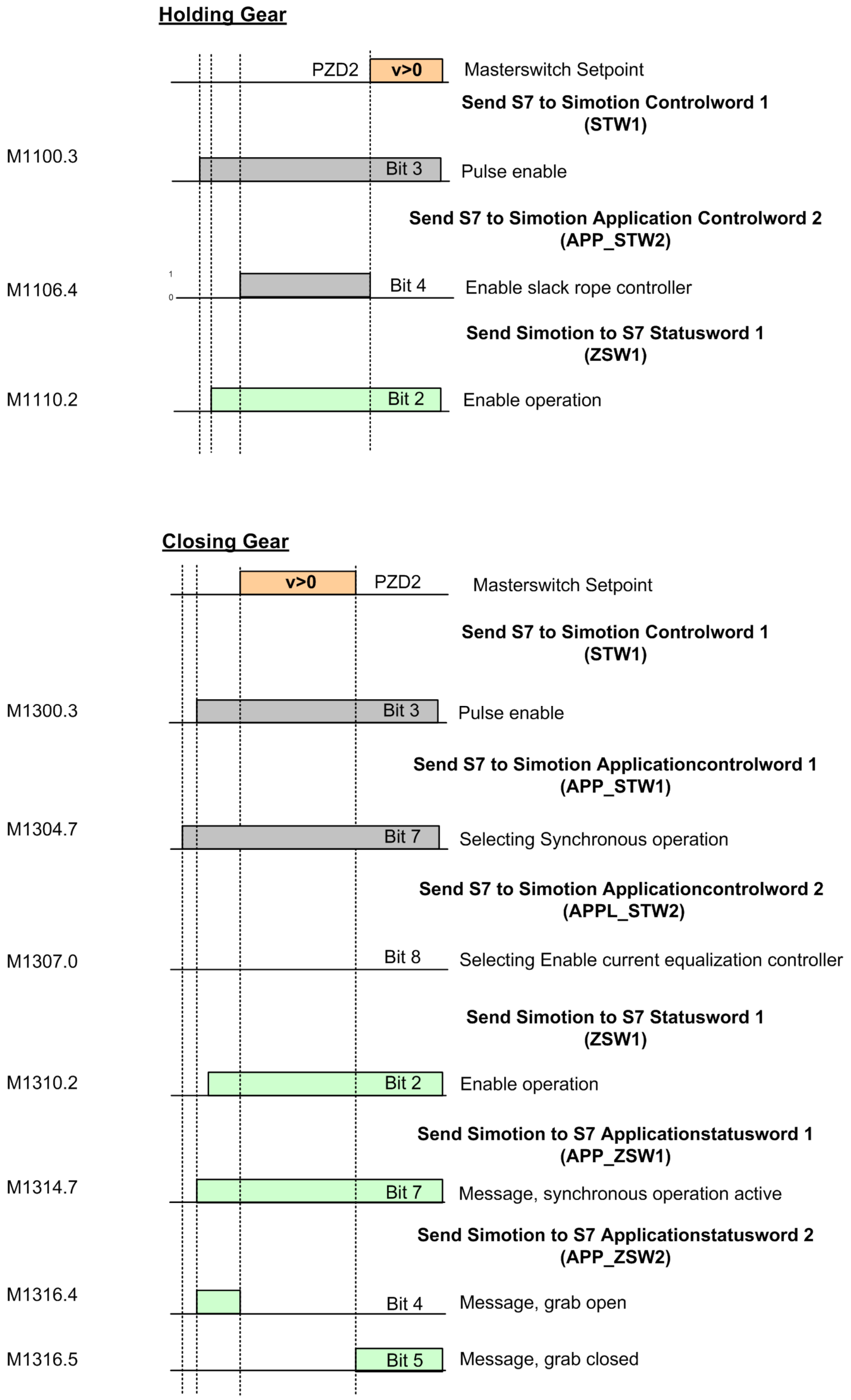

Holding Gear
PZD2
v>0
Masterswitch Setpoint
Send S7 to Simotion Controlword 1 (STW1)
M1100.3
Bit 3
Pulse enable
Send S7 to Simotion Application Controlword 2 (APP_STW2)
M1106.4
1
0
Bit 4
Enable slack rope controller
Send Simotion to S7 Statusword 1 (ZSW1)
M1110.2
Bit 2
Enable operation
Closing Gear
v>0
PZD2
Masterswitch Setpoint
Send S7 to Simotion Controlword 1 (STW1)
M1300.3
Bit 3
Pulse enable
Send S7 to Simotion Applicationcontrolword 1 (APP_STW1)
M1304.7
Bit 7
Selecting Synchronous operation
Send S7 to Simotion Applicationcontrolword 2 (APPL_STW2)
M1307.0
Bit 8
Selecting Enable current equalization controller
Send Simotion to S7 Statusword 1 (ZSW1)
M1310.2
Bit 2
Enable operation
Send Simotion to S7 Applicationstatusword 1 (APP_ZSW1)
M1314.7
Bit 7
Message, synchronous operation active
Send Simotion to S7 Applicationstatusword 2 (APP_ZSW2)
M1316.4
Bit 4
Message, grab open
M1316.5
Bit 5
Message, grab closed

Figure 8-26 Slack rope controller function

- The flags 1110.2 "Enable operation" for the holding gear and 1310.2 "Enable operation" for the closing gear are set.
- It is not permissible that flag 1307.04 "Enable current equalization controller" for the closing gear is set.
- The master switch gives the command "Close grab". With this command, flag 1106.4 "Enable slack rope controller" for the holding gear is set.
- The flag 1106.4 "Enable slack rope controller" remains set for the holding gear until the master switch gives the command "Lift grab".

8.5.4.3 Current equalization controller function

Holding gear and closing gear are switched-on.

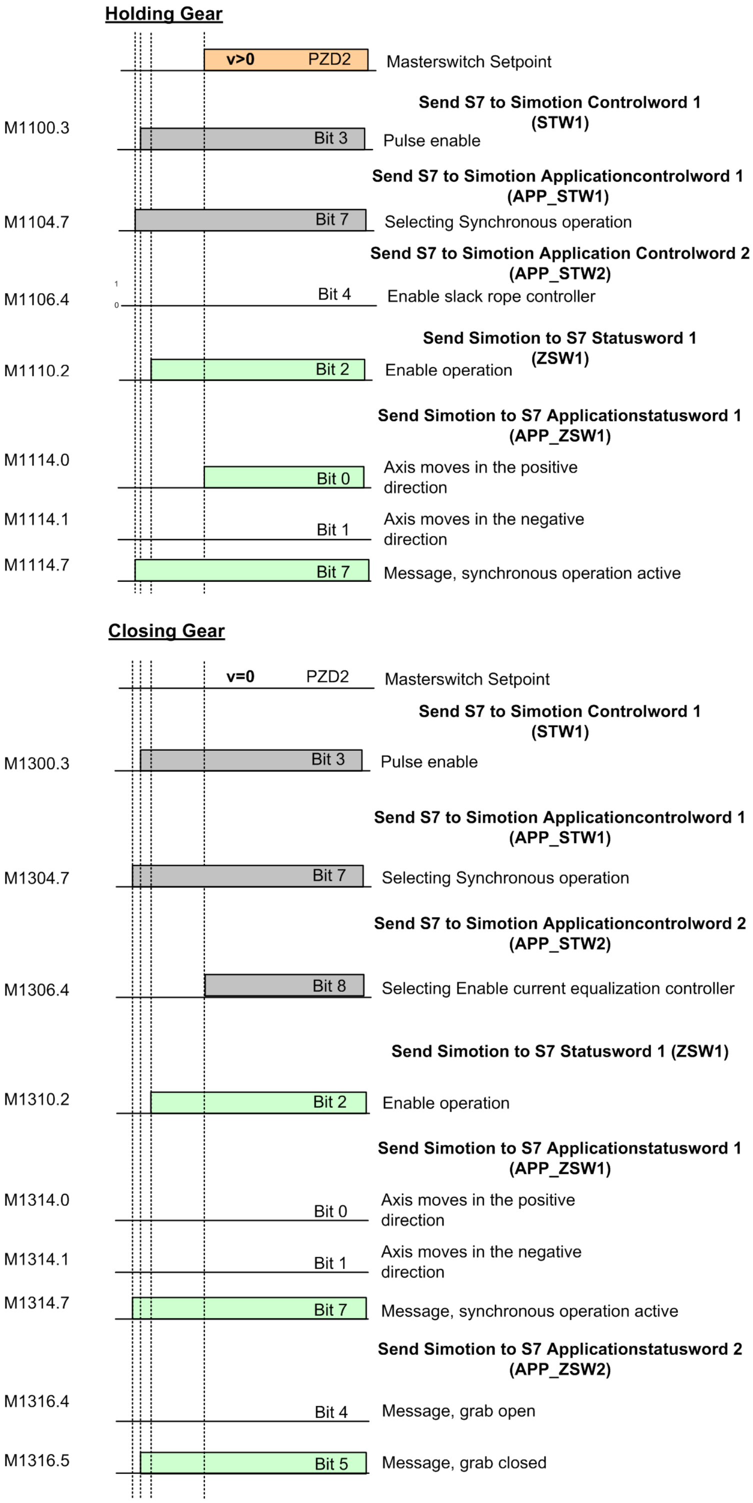

Holding Gear
v>0
PZD2
Masterswitch Setpoint
Send S7 to Simotion Controlword 1 (STW1)
M1100.3
Bit 3
Pulse enable
Send S7 to Simotion Applicationcontrolword 1 (APP_STW1)
M1104.7
Bit 7
Selecting Synchronous operation
Send S7 to Simotion Application Controlword 2 (APP_STW2)
M1106.4
1
0
Bit 4
Enable slack rope controller
Send Simotion to S7 Statusword 1 (ZSW1)
M1110.2
Bit 2
Enable operation
Send Simotion to S7 Applicationstatusword 1 (APP_ZSW1)
M1114.0
Bit 0
Axis moves in the positive direction
M1114.1
Bit 1
Axis moves in the negative direction
M1114.7
Bit 7
Message, synchronous operation active
Closing Gear
v=0
PZD2
Masterswitch Setpoint
Send S7 to Simotion Controlword 1 (STW1)
M1300.3
Bit 3
Pulse enable
Send S7 to Simotion Applicationcontrolword 1 (APP_STW1)
M1304.7
Bit 7
Selecting Synchronous operation
Send S7 to Simotion Applicationcontrolword 2 (APP_STW2)
M1306.4
Bit 8
Selecting Enable current equalization controller
Send Simotion to S7 Statusword 1 (ZSW1)
M1310.2
Bit 2
Enable operation
Send Simotion to S7 Applicationstatusword 1 (APP_ZSW1)
M1314.0
Bit 0
Axis moves in the positive direction
M1314.1
Bit 1
Axis moves in the negative direction
M1314.7
Bit 7
Message, synchronous operation active
Send Simotion to S7 Applicationstatusword 2 (APP_ZSW2)
M1316.4
Bit 4
Message, grab open
M1316.5
Bit 5
Message, grab closed

Figure 8-27 Current equalization controller

- The flags 1110.2 "Enable operation" for the holding gear and 1310.2 "Enable operation" for the closing gear are set.
- It is not permissible that flag 1106.4 "Enable slack rope controller" for the holding gear is set.
- If the flag 1316.5 "Grab closed message" is set, and the grab is to be lifted or lowered using the master switch, then flag 1307.04 "Enable current equalization controller" is set for the closing gear.
- The flag 1307.04 "Enable current equalization controller" for the closing gear remains set until the command "Open grab" is issued by the master switch - or until the flag 1311.2 "f or n comparison value reached" for the closing gear is not set.

8.5.4.4 Lifting the grab

Holding gear and closing gear are switched-on.

The current equalization controller is switched-on if the grab is closed and the holding gear is raised or lowered.

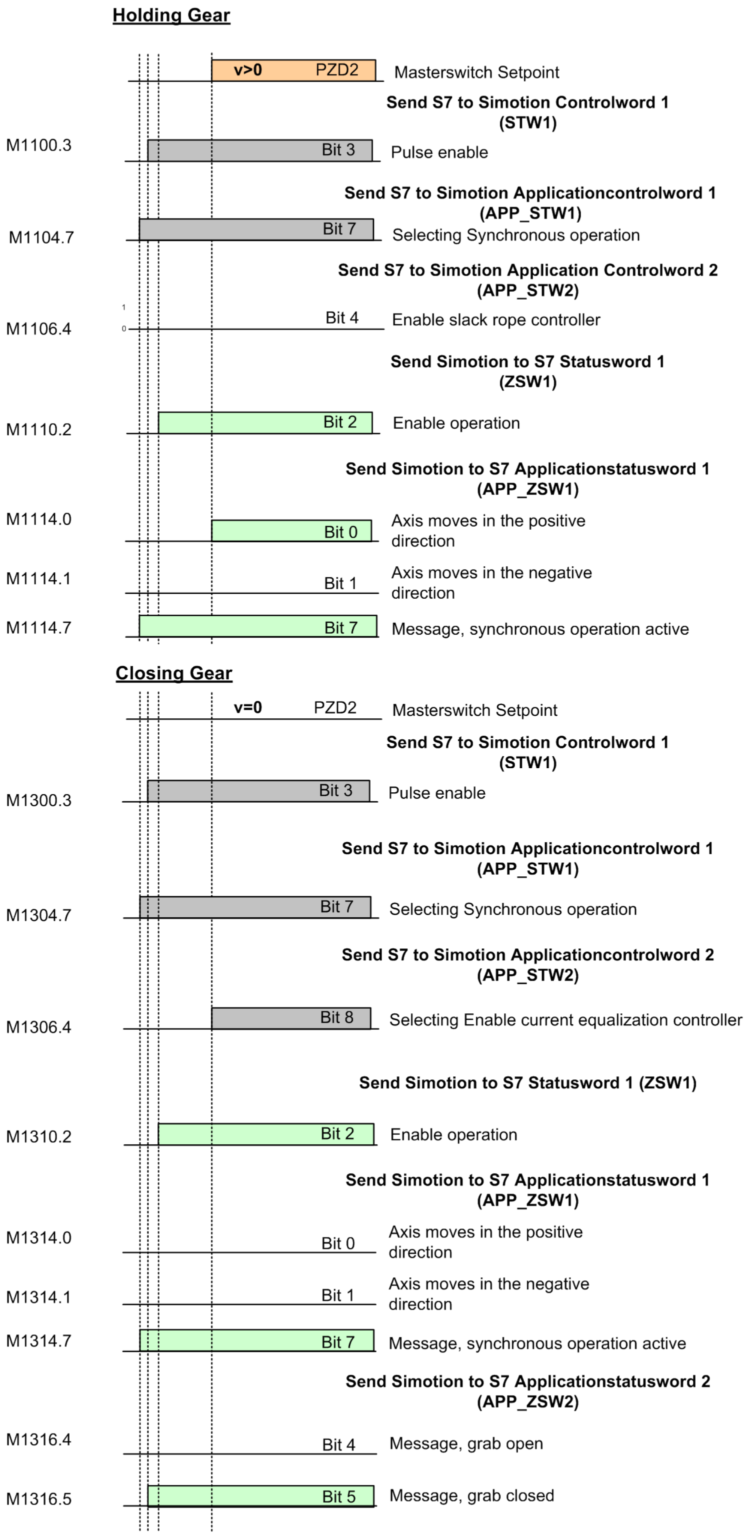

Holding Gear
v>0
PZD2
Masterswitch Setpoint
Send S7 to Simotion Controlword 1 (STW1)
M1100.3
Bit 3
Pulse enable
Send S7 to Simotion Applicationcontrolword 1 (APP_STW1)
M1104.7
Bit 7
Selecting Synchronous operation
Send S7 to Simotion Application Controlword 2 (APP_STW2)
1
0
M1106.4
Bit 4
Enable slack rope controller
Send Simotion to S7 Statusword 1 (ZSW1)
M1110.2
Bit 2
Enable operation
Send Simotion to S7 Applicationstatusword 1 (APP_ZSW1)
M1114.0
Bit 0
Axis moves in the positive direction
M1114.1
Bit 1
Axis moves in the negative direction
M1114.7
Bit 7
Message, synchronous operation active
Closing Gear
v=0
PZD2
Masterswitch Setpoint
Send S7 to Simotion Controlword 1 (STW1)
M1300.3
Bit 3
Pulse enable
Send S7 to Simotion Applicationcontrolword 1 (APP_STW1)
M1304.7
Bit 7
Selecting Synchronous operation
Send S7 to Simotion Applicationcontrolword 2 (APP_STW2)
M1306.4
Bit 8
Selecting Enable current equalization controller
Send Simotion to S7 Statusword 1 (ZSW1)
M1310.2
Bit 2
Enable operation
Send Simotion to S7 Applicationstatusword 1 (APP_ZSW1)
M1314.0
Bit 0
Axis moves in the positive direction
M1314.1
Bit 1
Axis moves in the negative direction
M1314.7
Bit 7
Message, synchronous operation active
Send Simotion to S7 Applicationstatusword 2 (APP_ZSW2)
M1316.4
Bit 4
Message, grab open
M1316.5
Bit 5
Message, grab closed

Figure 8-28 Lifting the grab

- The flags 1110.2 "Enable operation" for the holding gear and 1310.2 "Enable operation" for the closing gear are set.
- The grab is lifted with a positive master switch setpoint at the holding gear.

8.5.4.5 Lowering the grab

Holding gear and closing gear are switched-on.

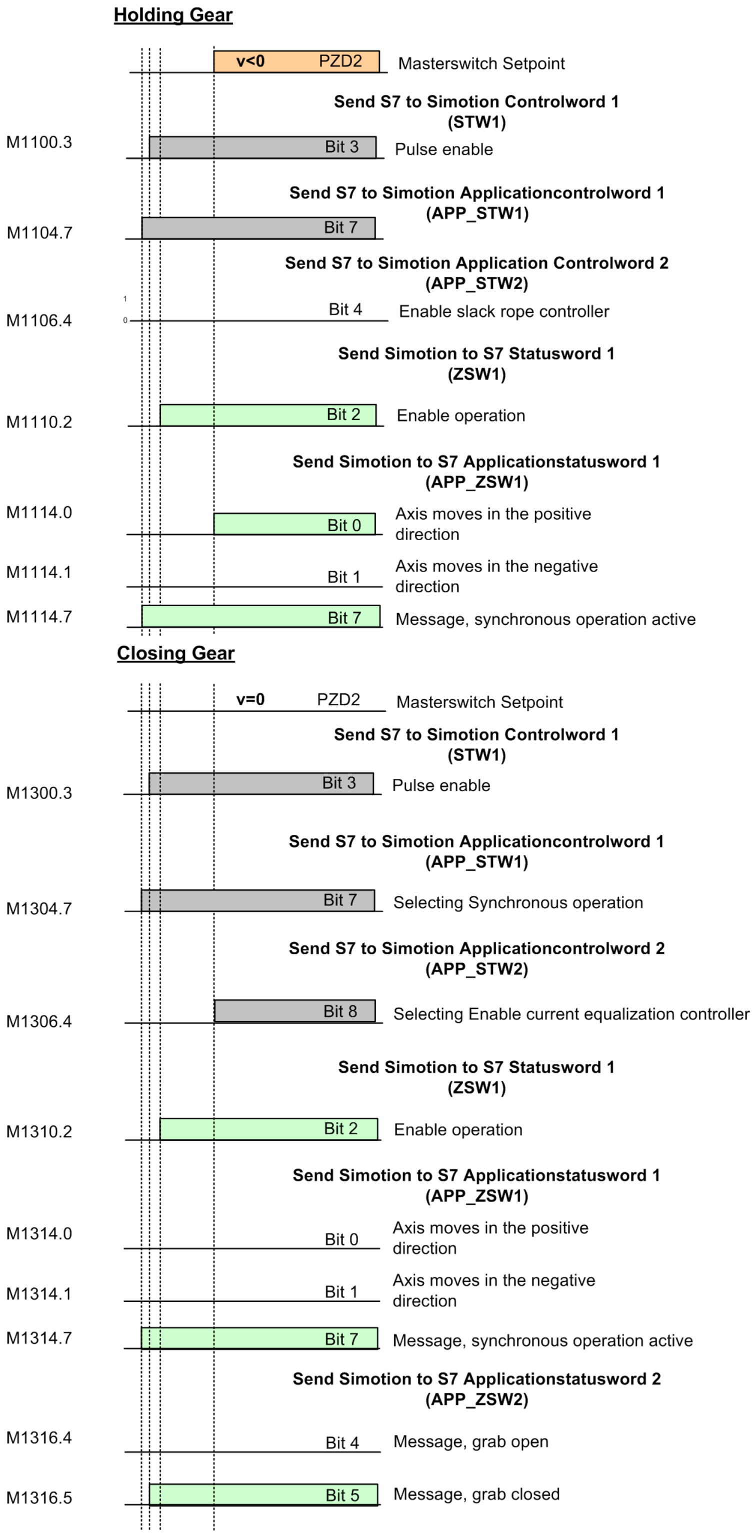

Holding Gear
v<0
PZD2
Masterswitch Setpoint
Send S7 to Simotion Controlword 1
(STW1)
M1100.3
Bit 3
Pulse enable
Send S7 to Simotion Applicationcontrolword 1
(APP_STW1)
M1104.7
Bit 7
Send S7 to Simotion Application Controlword 2
(APP_STW2)
M1106.4
1
0
Bit 4
Enable slack rope controller
Send Simotion to S7 Statusword 1
(ZSW1)
M1110.2
Bit 2
Enable operation
Send Simotion to S7 Applicationstatusword 1
(APP_ZSW1)
M1114.0
Bit 0
Axis moves in the positive direction
M1114.1
Bit 1
Axis moves in the negative direction
M1114.7
Bit 7
Message, synchronous operation active
Closing Gear
v=0
PZD2
Masterswitch Setpoint
Send S7 to Simotion Controlword 1
(STW1)
M1300.3
Bit 3
Pulse enable
Send S7 to Simotion Applicationcontrolword 1
(APP_STW1)
M1304.7
Bit 7
Selecting Synchronous operation
Send S7 to Simotion Applicationcontrolword 2
(APP_STW2)
M1306.4
Bit 8
Selecting Enable current equalization controller
Send Simotion to S7 Statusword 1
(ZSW1)
M1310.2
Bit 2
Enable operation
Send Simotion to S7 Applicationstatusword 1
(APP_ZSW1)
M1314.0
Bit 0
Axis moves in the positive direction
M1314.1
Bit 1
Axis moves in the negative direction
M1314.7
Bit 7
Message, synchronous operation active
Send Simotion to S7 Applicationstatusword 2
(APP_ZSW2)
M1316.4
Bit 4
Message, grab open
M1316.5
Bit 5
Message, grab closed

Figure 8-29 Lowering the grab

- The flags 1110.2 "Enable operation" for the holding gear and 1310.2 "Enable operation" for the closing gear are set.
- The grab is lowered with a negative master switch setpoint at the holding gear.

8.5.4.6 Opening the grab

Holding gear and closing gear are switched-on.

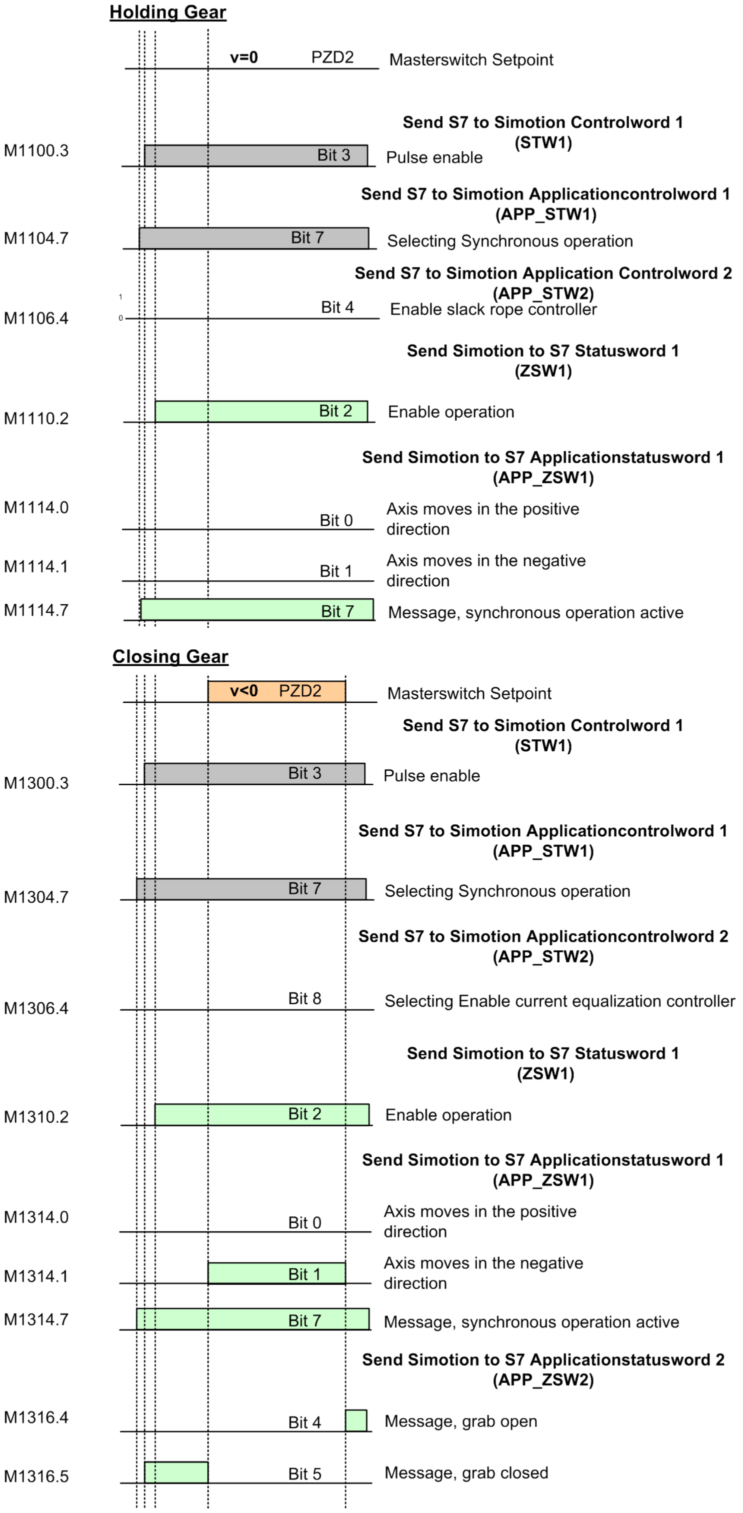

Holding Gear
v=0
PZD2
Masterswitch Setpoint
Send S7 to Simotion Controlword 1 (STW1)
M1100.3
Bit 3
Pulse enable
Send S7 to Simotion Applicationcontrolword 1 (APP_STW1)
M1104.7
Bit 7
Selecting Synchronous operation
Send S7 to Simotion Application Controlword 2 (APP_STW2)
M1106.4
1
0
Bit 4
Enable slack rope controller
Send Simotion to S7 Statusword 1 (ZSW1)
M1110.2
Bit 2
Enable operation
Send Simotion to S7 Applicationstatusword 1 (APP_ZSW1)
M1114.0
Bit 0
Axis moves in the positive direction
M1114.1
Bit 1
Axis moves in the negative direction
M1114.7
Bit 7
Message, synchronous operation active
Closing Gear
v<0
PZD2
Masterswitch Setpoint
Send S7 to Simotion Controlword 1 (STW1)
M1300.3
Bit 3
Pulse enable
Send S7 to Simotion Applicationcontrolword 1 (APP_STW1)
M1304.7
Bit 7
Selecting Synchronous operation
Send S7 to Simotion Applicationcontrolword 2 (APP_STW2)
M1306.4
Bit 8
Selecting Enable current equalization controller
Send Simotion to S7 Statusword 1 (ZSW1)
M1310.2
Bit 2
Enable operation
Send Simotion to S7 Applicationstatusword 1 (APP_ZSW1)
M1314.0
Bit 0
Axis moves in the positive direction
M1314.1
Bit 1
Axis moves in the negative direction
M1314.7
Bit 7
Message, synchronous operation active
Send Simotion to S7 Applicationstatusword 2 (APP_ZSW2)
M1316.4
Bit 4
Message, grab open
M1316.5
Bit 5
Message, grab closed

Figure 8-30 Opening the grab

- The flags 1110.2 "Enable operation" for the holding gear and 1310.2 "Enable operation" for the closing gear are set.
- The grab is opened with a negative master switch setpoint at the closing gear.

8.5.4.7 Close grab

Holding gear and closing gear are switched-on.

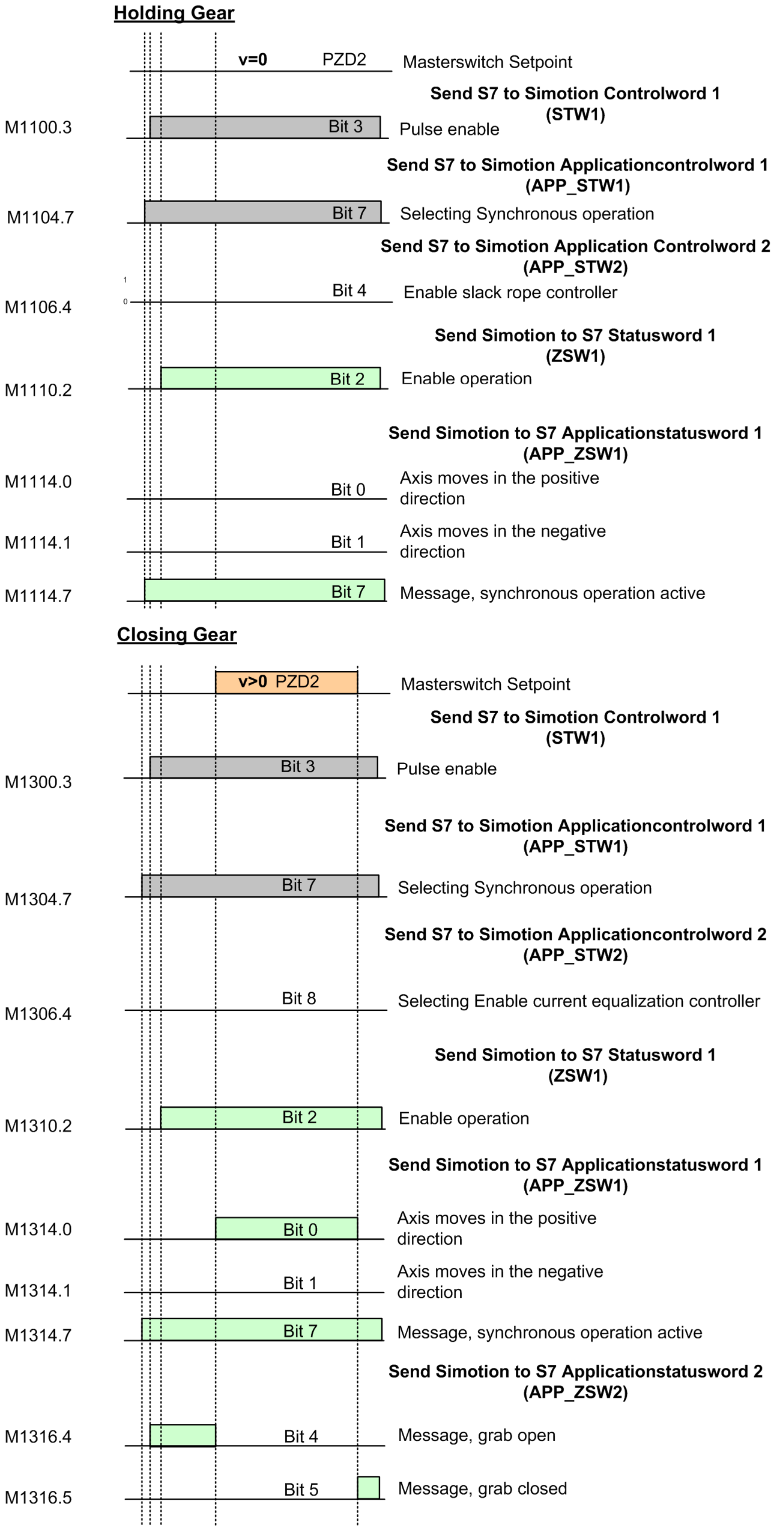

Holding Gear
v=0
PZD2
Masterswitch Setpoint
Send S7 to Simotion Controlword 1 (STW1)
M1100.3
Bit 3
Pulse enable
Send S7 to Simotion Applicationcontrolword 1 (APP_STW1)
M1104.7
Bit 7
Selecting Synchronous operation
Send S7 to Simotion Application Controlword 2 (APP_STW2)
1
0
M1106.4
Bit 4
Enable slack rope controller
Send Simotion to S7 Statusword 1 (ZSW1)
M1110.2
Bit 2
Enable operation
Send Simotion to S7 Applicationstatusword 1 (APP_ZSW1)
M1114.0
Bit 0
Axis moves in the positive direction
M1114.1
Bit 1
Axis moves in the negative direction
M1114.7
Bit 7
Message, synchronous operation active
Closing Gear
v>0 PZD2
Masterswitch Setpoint
Send S7 to Simotion Controlword 1 (STW1)
M1300.3
Bit 3
Pulse enable
Send S7 to Simotion Applicationcontrolword 1 (APP_STW1)
M1304.7
Bit 7
Selecting Synchronous operation
Send S7 to Simotion Applicationcontrolword 2 (APP_STW2)
M1306.4
Bit 8
Selecting Enable current equalization controller
Send Simotion to S7 Statusword 1 (ZSW1)
M1310.2
Bit 2
Enable operation
Send Simotion to S7 Applicationstatusword 1 (APP_ZSW1)
M1314.0
Bit 0
Axis moves in the positive direction
M1314.1
Bit 1
Axis moves in the negative direction
M1314.7
Bit 7
Message, synchronous operation active
Send Simotion to S7 Applicationstatusword 2 (APP_ZSW2)
M1316.4
Bit 4
Message, grab open
M1316.5
Bit 5
Message, grab closed

Figure 8-31 Close grab

- Flags 1110.2 "Enable operation" for the holding gear and 1310.2 "Enable operation" for the closing gear are set.
- The grab is closed with a positive master switch setpoint at the closing gear.

8.6 SINAMICS drive object

Basic procedure

- Enter the motor data according to the rating plate
- Enter the following drive data for any number (n_items) of drives wired in parallel:
  - Rated voltage: p0304 ≤ V_{rated}
  - Rated current: p0305 ≤ I_{rated} • n_items
  - Rated power: p0307 ≤ P_{rated} • n_Stck.
  - Rated power factor: p0308 ≤ cos_φ_{rated}
  - Rated efficiency: p0309 ≤ η_{rated}
  - Rated frequency: p0310 ≤ f_{rated}
  - Rated speed: p0311 ≤ n_{rated}
- As of SINAMICS V2.6 for motors connected in parallel, enter the following motor values for a parallel connection:
  - Rated voltage: p0304 ≤ V_{rated}
  - Rated current: p0305 ≤ I_{rated}
  - Number of motors connected in parallel: n_motors
  - Rated power: p0307 ≤ P_{rated}
  - Rated power factor: p0308 ≤ cos_φ_{rated}
  - Rated efficiency: p0309 ≤ η_{rated}
  - Rated frequency: p0310 ≤ f_{rated}
  - Rated speed: p0311 ≤ n_{rated}
- In order to set up the drive model with maximum precision, enter the SINAMICS parameters p0230, p0233 and p0235 for each drive in accordance with the plant configuration.

NOTICE
If this value is incorrectly preset for a motor with a separate winding system, extreme torque surges will occur when the drive is switched on.

- Measure the total line resistance (Motor Module to drive) for each drive and enter the result in SINAMICS parameter p0352.
- Enter the motor series reactor (if installed) in SINAMICS parameter p0353.
- If the drives are in a master-slave coupling, r0079 from the master should be interconnected to p1503 for the slave!

Note

When commissioning the system from the control panel of the drive object, a changeover must be made from command data set 0 (command data set selection CDS p0810) to command data set 1. After moving the axis from the control panel of the drive object, it must be reset otherwise operator control from the S7 controller is not possible.

Stationary Measurement

After having checked all prerequisites, carry out a stationary measurement in SINAMICS for each Motor Module (i.e. Hoist, Gantry, Trolley, etc.).

NOTICE

Prior to standstill measurement:
After having entered all data, go online to the corresponding drive object and verify the data specified above or adapt these accordingly. Set the initial value 2 at SINAMICS parameter p0340. After parameter p0340 has returned to 0 state, set it once again to value 3

The following values are determined in the stationary measurement:

- Equivalent circuit diagram data
- Stator resistance
- Leakage inductances:
- IGBT on-state voltage or compensation for the IGBT lockout times

Tips after stationary measurement

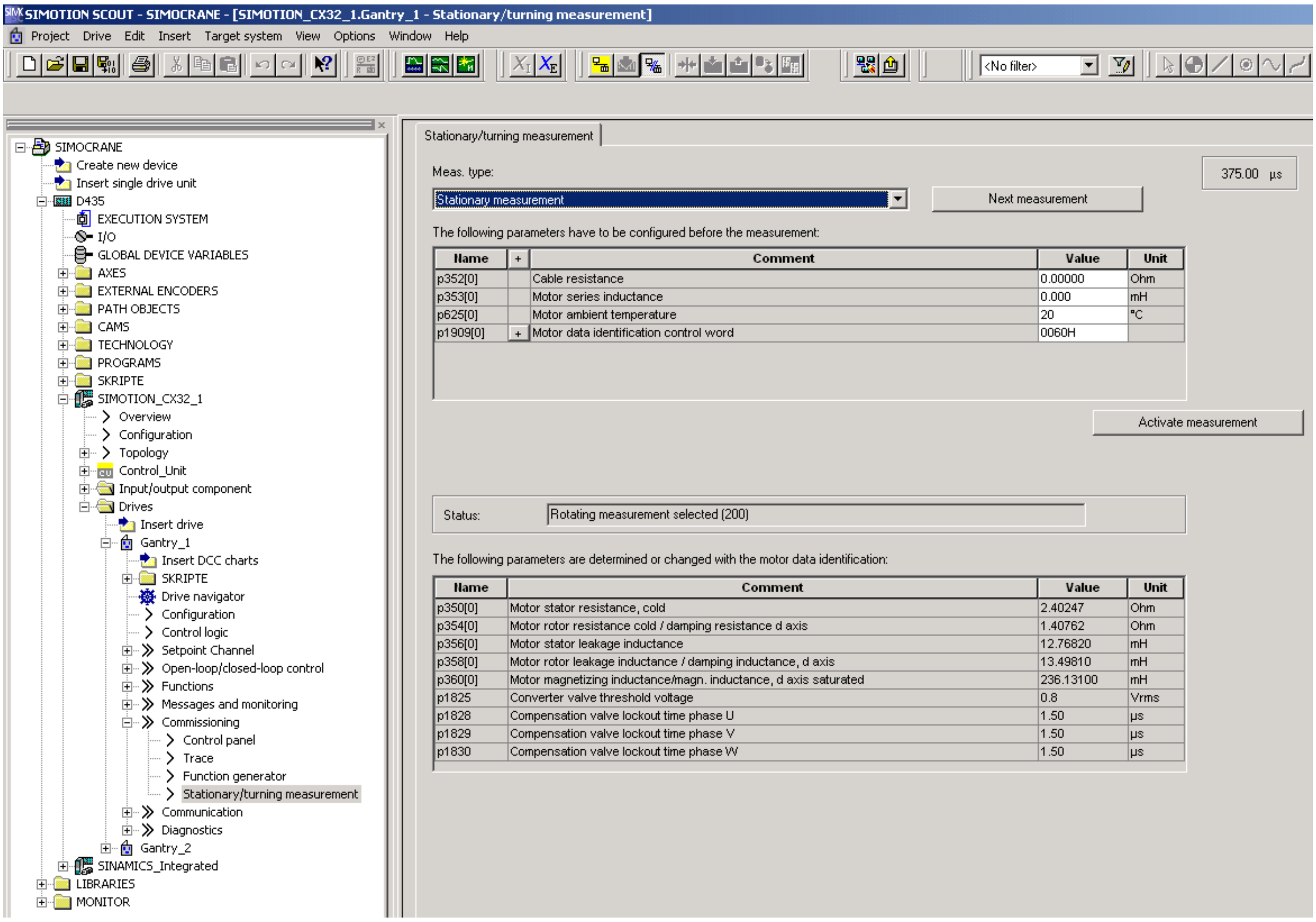
- Check the actual operational flow:
  - After the correct rating plate data has been set and after successfully settled MOT ID, the actual flow (r0084) is 100% below field weakening.
  - Check the entire speed range up to field weakening, e.g. during a ramp-up operation.
- If a rotating measurement cannot be performed, correct the magnetizing current and magnetizing inductance manually if necessary:
  - Correct the rated magnetizing current (p0320) until an actual flow (r0084) in the entire speed range until the field weakening attains 100%. Differences particularly in the range to 30% n_{rated} will become evident if the the rated magnetizing current has not been corrected.
  - If the drive can be rotated freely (without or with slight load), the corrected value of the magnetizing current (r0029) can be read when operating in the speed range 60% n_{rated} to 80% n_{rated} and be entered in p0320.
  - The magnetizing inductance (p0360) cannot be corrected directly. The drive retains a constant product p0320 • p0360 (rated flow) and changes p0360 depending on p0320. The value r1787 = 0 is displayed (correction value Lh adaptation) if the correct rated magnetizing current was set (p0320).

Note

After having carried out the standstill measurement, do not forget to copy RAM data to ROM and to transfer all data to the offline project. Select **Upload to PG** accordingly and then save the project.

The stationary measurement is available in the Project Navigator at SINAMICS **Drive object** (e.g. Hoist_1) in menu item **Commissioning.**

NOTICE
Check the torque and current limits after each stationary measurement!



Meas. type: Stationary measurement

375.00 µs

The following parameters have to be configured before the measurement:

Name	+	Comment	Value	Unit
p352[0]		Cable resistance	0.00000	Ohm
p353[0]		Motor series inductance	0.000	mH
p625[0]		Motor ambient temperature	20	°C
p1909[0]	+	Motor data identification control word	0060H	

Status: Rotating measurement selected (200)

The following parameters are determined or changed with the motor data identification:

Name	Comment	Value	Unit
p350[0]	Motor stator resistance, cold	2.40247	Ohm
p354[0]	Motor rotor resistance cold / damping resistance d axis	1.40762	Ohm
p356[0]	Motor stator leakage inductance	12.76820	mH
p358[0]	Motor rotor leakage inductance / damping inductance, d axis	13.49810	mH
p360[0]	Motor magnetizing inductance/magn. inductance, d axis saturated	236.13100	mH
p1825	Converter valve threshold voltage	0.8	Vrms
p1828	Compensation valve lockout time phase U	1.50	µs
p1829	Compensation valve lockout time phase V	1.50	µs
p1830	Compensation valve lockout time phase W	1.50	µs

Figure 8-32 Stationary measurement

Checking the direction of rotation

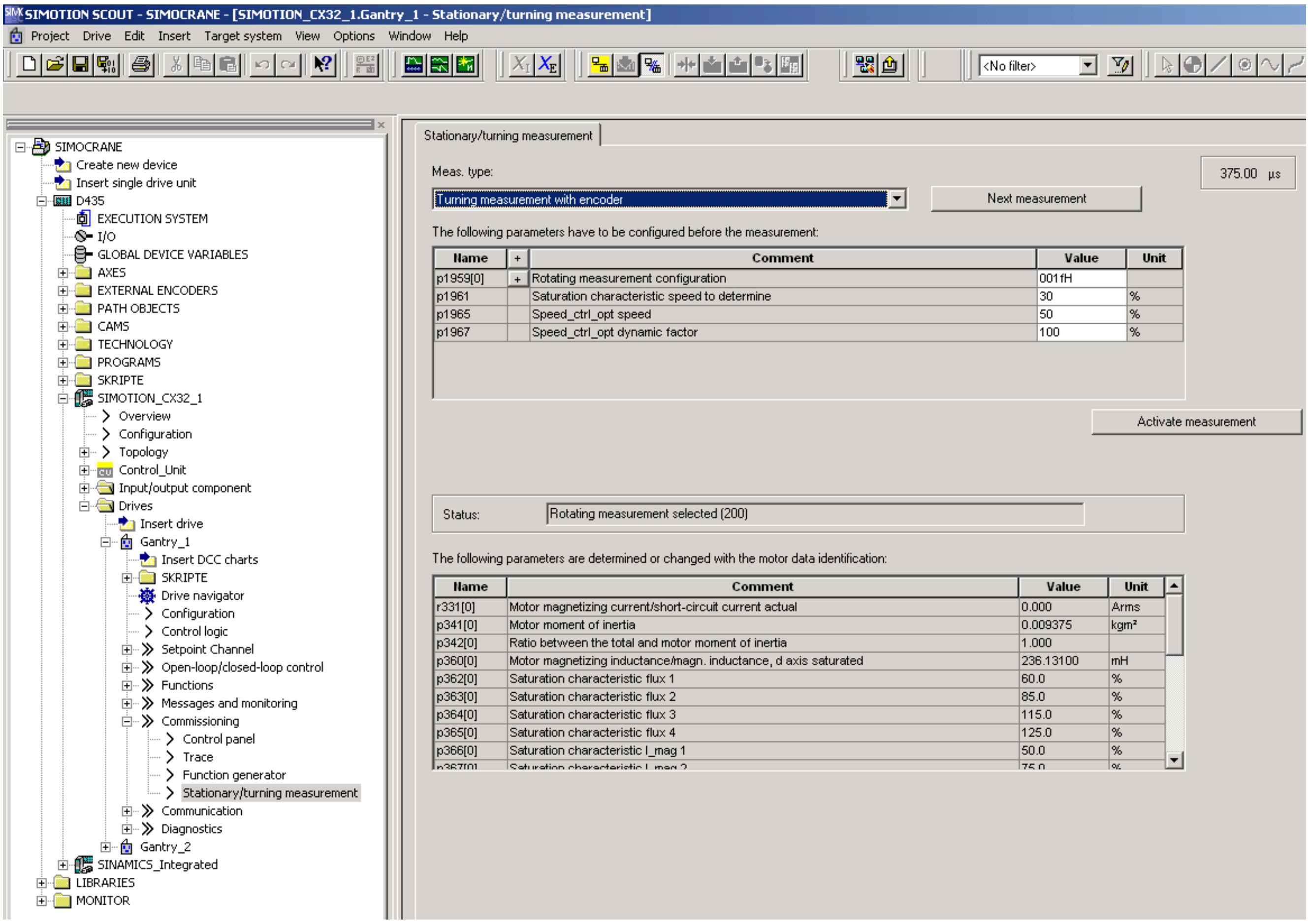
After the stationary measurement has been successfully completed, check the direction of rotation. If the motors have still not been installed, the SINAMICS control panel should be used to check the direction of rotation. Otherwise, the direction of rotation should be checked in the SPEED_CONTROLLED operating mode, i.e. when the motor is rotating in the positive direction. The encoder must then supply positive values. If it is no longer possible to interchange the phases, the direction must be reversed in SINAMICS using parameter p1821. Please note that for several motor data sets (Drive Data Set), this change must be made for all of the data sets involved.

More detailed information about using the control panel can be found in Chapter 4.3 "Using the operator control panel" in Ref. [5].

Rotating measurement

The rotating measurement with speed controller optimization should then be carried out. The rotating measurement function should only be initiated if the motor can freely rotate. Otherwise, it is not permissible that this function is executed. The following values are determined in the rotating measurement:

- The magnetizing characteristic is measured
- The magnetizing current is measured
- The speed controller is optimized
- The acceleration pre-control is set
- The ratio between the total moment of inertia and the motor is set



Meas. type: Turning measurement with encoder

375.00 µs

The following parameters have to be configured before the measurement:

Name	+	Comment	Value	Unit
p1959[0]	+	Rotating measurement configuration	001fH	
p1961		Saturation characteristic speed to determine	30	%
p1965		Speed_ctrl_opt speed	50	%
p1967		Speed_ctrl_opt dynamic factor	100	%

Status: Rotating measurement selected (200)

The following parameters are determined or changed with the motor data identification:

Name	Comment	Value	Unit
r331[0]	Motor magnetizing current/short-circuit current actual	0.000	Arms
p341[0]	Motor moment of inertia	0.009375	kgm²
p342[0]	Ratio between the total and motor moment of inertia	1.000	
p360[0]	Motor magnetizing inductance/magn. inductance, d axis saturated	236.13100	mH
p362[0]	Saturation characteristic flux 1	60.0	%
p363[0]	Saturation characteristic flux 2	85.0	%
p364[0]	Saturation characteristic flux 3	115.0	%
p365[0]	Saturation characteristic flux 4	125.0	%
p366[0]	Saturation characteristic I_mag 1	50.0	%
p367[0]	Saturation characteristic I_mag 2	75.0	%

Figure 8-33 Rotating measurement

If the rotating measurement cannot be initiated, then the speed control loop should either be optimized by re-calculating the control parameters or manually using the controller. If the moment of inertia was entered, the control parameters are calculated using the automatic parameterization (p0340 = 4). The controller parameters are defined in accordance with the symmetrical optimum as follows:

- $T_n = 4 \cdot T_s$
- $K_p = 0.5 \cdot r0345 / T_s \rightarrow K_p = 2 \cdot r0345 / T_n$ (T_s = total of short deceleration times)
  - Sum of the short deceleration times, vector operation with encoder:

    Actual speed value smoothing time p1441 and speed controller, actual speed value smoothing time p1442
  - Sum of the short deceleration times, vector operation without encoder:

    Speed controller, actual speed value smoothing time p1452

> **Note**
>
> As a general rule, the lowest possible actual speed smoothing values should be selected for normal operation. When the encoders are well shielded, a smoothing value of 0 can be entered.
> If a setting of 0 is entered, the above formulas no longer apply. In this case, T_n should be set to 0, K_p to overshoot-free, and then T_n increased until the desired controller performance is obtained.

If vibration develops with these settings, reduce speed controller gain K_p manually. Actual speed value smoothing can also be increased (standard procedure for gearless or high-frequency torsional vibration) and the controller calculation performed again because this value is also used to calculate K_p and T_n. The following relationships apply for optimization:

- If K_p is increased, the controller becomes faster and overshoot is increased.
- If T_n is reduced, the controller also becomes faster and overshoot is also increased.

When setting the speed control manually, you are advised to define the dynamic response via K_p (and actual speed value smoothing) first, so that the integral time can subsequently be reduced as much as possible. Please note that the closed-loop control must also remain stable in the field-weakening range.

Reducing the controller gain will generally dampen any oscillations in the closed-loop speed control. Another possible option is to increase the smoothing time setting in p1452. However, the accuracy of the control will diminish if the smoothing setting is too high. The integral output of the speed controller can be monitored via r1482 and the limited controller output (torque setpoint) via r1508.

If the motor is to operate under closed-loop position control, the control parameters must be set according to the absolute value optimum. Make sure that the speed controller is free of overshoots!

You can also design the speed controller for the relevant motor according to the symmetrical optimum and utilize the reference model. When correctly set, the reference model can achieve a response to disturbances according to the symmetrical optimum and a response to reference variables according to the absolute value optimum. For further information about the reference model, refer to Ref. [4], Chapter 4.5.

NOTICE
Check the torque and current limits after each stationary measurement!

Programming tips

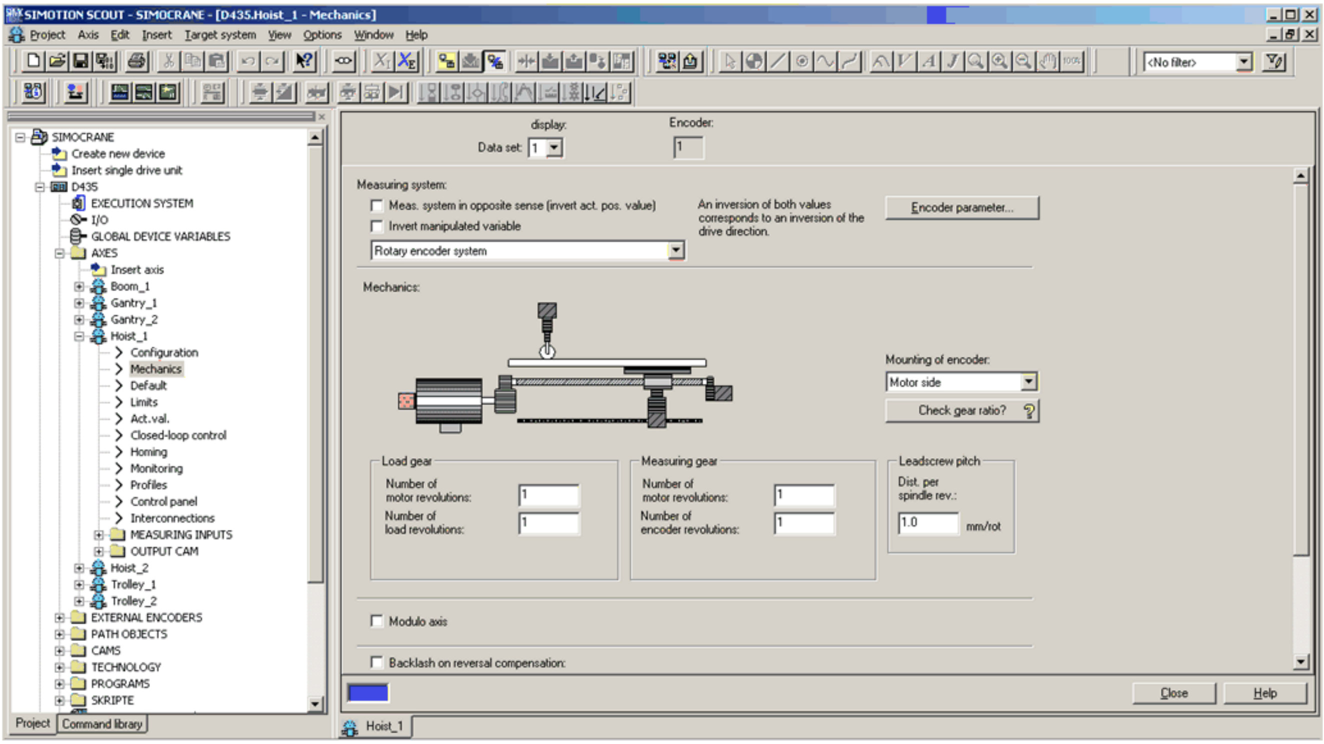
- Adjusting the switchover from current model ("controlled") to voltage model ("regulated"):
  - p1752 = to approx. 30% of p0311 "Rated motor speed"
  - p1756 = 20 [%]: Hysteresis

  Both value were proven useful, however, these can be adapted to suit your requirements.
- Conclude commissioning by testing the entire speed range and verifying smooth operation in order to exclude universal shaft/gear problems.
- The trolleys excite inherent resonance of the crane frame at rapid changes of the speed.
  This is not in the control loop of the trolley; it represents the reference system.
  In order to avoid excitation of vibration: K_p low, T_n high, adaptation
  - The integrator has a tendency towards saturation which leads to overshoot of the end position and subsequent retraction
  - It is possible that rounding-off could be helpful.
- Acceleration precontrol for known masses
  Problem: Mass variation with/without container; set a precontrol value approx. 80% – 90%
  Relieve the I component as far as possible with precontrol: Also observe r1482; r1481; r0080 und r0078 in the trace.
  - The precontrol function is available in the Project Navigator at SINAMICS **Drive object** (e.g. Hoist_1) at **Open-loop/closed-loop control** > **speed setpoint filter** > **Precontrol**
  - The precontrol function can be set accordingly at p0342 **Ratio of total/motor moment of inertia** and p1496 **Acceleration precontrol scaling**.
- Setpoint/actual value smoothing – Balancing
  - Actual value smoothing should be kept at minimum; high values increase overshoot
    Setpoint smoothing provides a more constant velocity due to the motion profile (master switch setpoint)

8.7 SIMOTION technology object

Setting the travel path

Once the speed control loop has been optimized successfully, the mechanical connection between the motor revolutions and the travel path should be configured next. You can find these under technology object (e.g.: Hoist_1) in the menu item **Mechanics**.



display: Data set: 1

Encoder: 1

Measuring system:

□ Meas. system in opposite sense (invert act. pos. value)

□ Invert manipulated variable

An inversion of both values corresponds to an inversion of the drive direction.

Rotary encoder system

Mechanics:

Mounting of encoder: Motor side

Load gear
Number of motor revolutions: 1
Number of load revolutions: 1

Measuring gear
Number of motor revolutions: 1
Number of encoder revolutions: 1

Leadscrew pitch
Dist. per spindle rev.: 1.0 mm/rot

□ Modulo axis

□ Backlash on reversal compensation:

Figure 8-34 Setting the mechanical connection

The value for the maximum velocity is calculated using the formula below:

$$V_{max}\,[{}^{mm}/_{s}] = \frac{\text{number of load revolutions}\,[{}^{1}/_{min}]}{\text{number of motor revolutions}\,[{}^{1}/_{min}]} \cdot \frac{\text{rated speed}}{60}\,[{}^{1}/_{s}] \cdot \text{dis. per spindle revolution}\,[{}^{mm}/_{rot}]$$

The reference velocity must be normalized to the maximum speed in the expert list under **TypeOfAxis.SetPointDriver Info.DriveData.speedReference**. The maximum speed – maxSpeed – setting must be identical to the reference speed (p2000) of the drive object.

If the relationship between the motor speed and the travel path is to be checked in advance, it can be helpful to set the mechanics as follows:

- Number of motor revolutions: 1
- Number of load revolutions: 60
- Distance per spindle revolution: 1

This setting is used to check the velocity interface, because one motor revolution per minute is equal to one millimeter of distance travelled per second (1 1/min ≙ 1 mm/s). This means that the velocity setpoint and the actual value channel can be compared without the need for further conversions.

Checking the travel path

If the maximum velocity has been set and the relationship between the drive object and the technology object is correct, the travel path can be checked. In this case, move the drive slowly in closed-loop speed controlled operation in the SPEED_CONTROLLED operating mode to check the position actual value. It is important that the specified distance per motor revolution precisely corresponds with what happens in reality.

Note

Verify proper optimization of the speed control loop before you start tuning the position control loop.

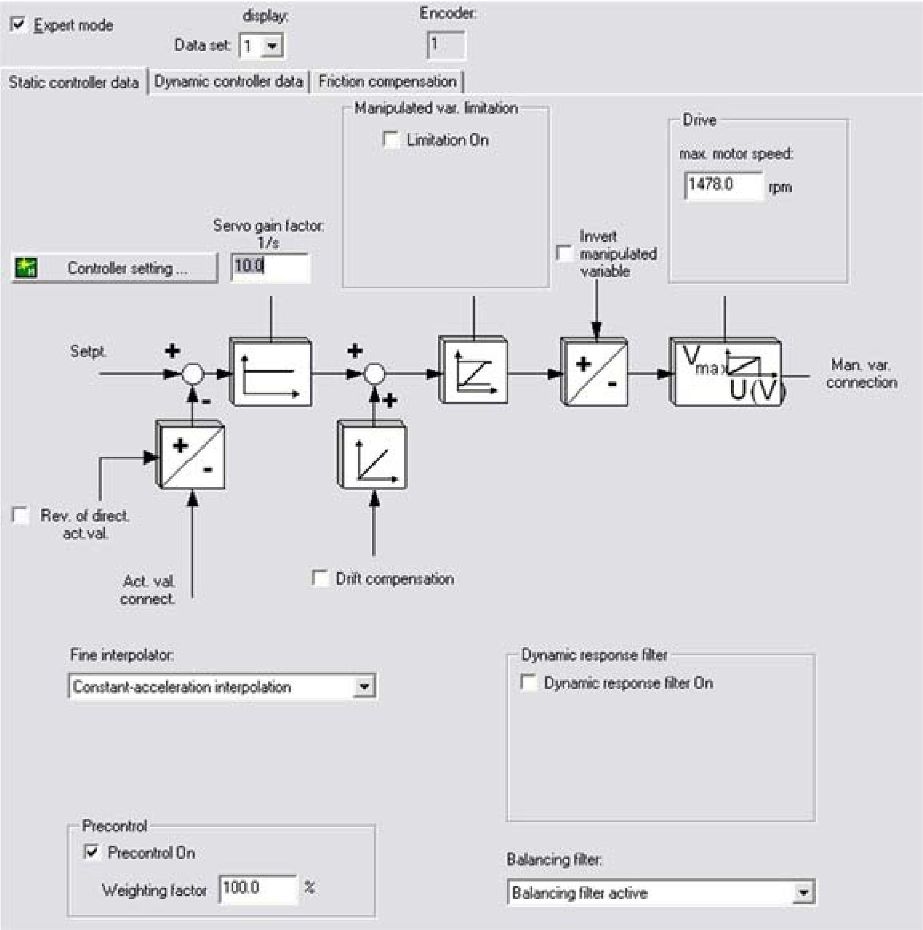
If the speed actual value is affected by high ripple, a position controller gain Kv higher than "1" should be avoided, as this will increase vibration.

Position control optimization

The position control loop should then be optimized. For further information, please make sure that you read Ref. 2, Chapter 2.12 "Commissioning the position controllers of positioning axes". Chapters 2.10 and 2.11 in Ref. 2 also contain useful information.

The explanation of the position controller optimization process given below is intended only as a guide and is a summary of the content of document Ref. 2, Chapter 2.12. The task of a position controller for positioning applications is to ensure that the drive precisely reaches the target. The speed pre-control must be disabled in order to optimize the position controller. To do this, set the weighting factor (Hoist_1.TypeOfAxis.NumberOfDataSets.DataSet_1.ControllerStruct.PV_Controller.kpc) to 0 % and the equivalent time of the subordinate drive v_{TC} (Hoist_1.TypeOfAxis.NumberOfDataSets.DataSet_1.DynamicData.velocityTimeConstant) to 0 s. Then traverse the drive to different positions in one of the modes AUTOMATIC or MANUAL. The position setpoint (Hoist_1.servoData.symmetricCommandPosition) and the position actual value (Hoist_1.servoData.actualPosition) must be recorded using the SIMOTION trace function. The position actual value must precisely track the position setpoint during the complete positioning operation. Make sure that the axis reaches the constant velocity phase during this operation. The difference between the position setpoint and position actual value is known as the following error (Hoist_1.servodata.followingError).

The following error is changed by changing the Kv factor (Hoist_1.TypeOfAxis.NumberOfDataSets.DataSet_1.ControllerStruct.PV_Controller.kv). A low following error during the complete travel motion - without position actual value overshoot at the target - signifies a good position controller setting.



☑ Expert mode

display: Data set: 1

Encoder: 1

Manipulated var. limitation: □ Limitation On

Drive: max. motor speed: 1478.0 rpm

Servo gain factor: 1/s 10.0

□ Invert manipulated variable

□ Rev. of direct. act.val.

□ Drift compensation

Fine interpolator: Constant-acceleration interpolation

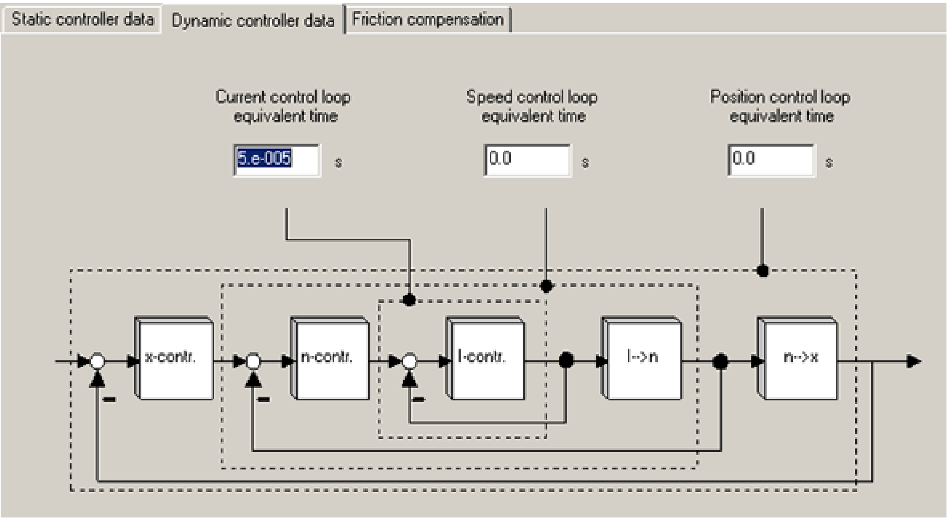
Dynamic response filter: □ Dynamic response filter On

Precontrol: ☑ Precontrol On; Weighting factor 100.0 %

Balancing filter: Balancing filter active

Figure 8-35 Limiting the manipulated variable in static controller data

After the optimum Kv factor was determined, the weighting factor for the speed pre-control must be again set to 100 %. After this, several positioning operations should be carried out. In so doing, the position setpoint and position actual value should again be checked and the approach to the target recorded using the trace function. The following applies: The following error must be low during the complete travel motion. It is not permissible that the position actual value overshoots the target. In order to achieve this, the precise equivalent time constant of the speed control loop v_{TC} (Hoist_1.TypeOfAxis.NumberOfDataSets.DataSet_1.DynamicData.velocityTimeConstant) must be determined.



Current control loop equivalent time: 5.e-005 s

Speed control loop equivalent time: 0.0 s

Position control loop equivalent time: 0.0 s

Figure 8-36 Dynamic controller data

In so doing, this time constant must be increased until the target is optimally approached without any overshoot. Reduce the calculated Kv factor slightly if overshoot still develops; refer to Ref.2, Chapters 2.11 and 2.12.

Adapting monitoring limits

The following monitoring limits must be adapted after the optimum control loop settings have been determined.
Following error, positioning and standstill monitoring as well as standstill signal.
These are specified in chapter Overview - monitoring functions (Page 333) or in Ref. 3.

The settings are determined using various positioning operations. The position setpoint, position actual value and following error are recorded using SIMOTION trace. The maximum following error that occurs can be determined using the trace function. A low safety margin should be added and then entered into the corresponding variable as maximum permissible following error; Dynamic monitoring of following errors (Page 333).

Positioning window, positioning tolerance time, standstill window and standstill signals should be determined using various approaches to the target. A low safety margin should be added to these values for calculation and be entered into the corresponding variables; refer to Position and standstill monitoring (Page 335) and Standstill signal (Page 337).

Adjust limitations

The next step is to adjust the limitations. In this case, it is particularly important to recheck the maximum velocity which can be achieved as a result of the mechanical components configured in the axis. It is also important to check on the "Dynamic response" tab whether the maximum settings for acceleration and jerk have been adjusted according to the configured mechanical system, i.e. that these maximum values cannot result in mechanical damage to the crane.

8.8 Crane DCC blocks

The DCC blocks must now be configured for the various tasks; for a description of the DCC blocks, see Crane DCC library (Page 71).

1. "MasterSwitch" or "MasterSwitch_1" block

   The master switch curve must be saved if the "MasterSwitch" block is used.
2. "TractionControl" block (only relevant to Gantry and Trolley)

   The velocity reduction and acceleration reduction curves must be adapted if the "TractionControl" block is used.
3. "PreLimitSwitch" block

   The velocity limits must be adapted in the "PreLimitSwitch" block.
4. "ChangeOverHDFW" block

   The acceleration and deceleration times must be adapted in the "ChangeOverHDFW" block and when selecting heaving duty load.
5. "CurrentDistribution" block (only relevant for Trolley and Hoist)

   The deceleration time and the parameter threshold must be adapted as required in block "CurrentDistribution".
6. "HeavyDuty" block (relevant for Hoist, Holding Gear, and Closing Gear)

   The heavy duty load velocity must be adapted in the "HeavyDuty" block.
7. "StartPulse" block (relevant for Boom, Hoist, Holding Gear, Closing Gear, and Luffing Gear)

   The procedure must be defined in the "StartPulse" block. The decay time and the evaluation for hoisting and lowering must be adapted if necessary; see Start pulse (Page 451)
8. "LoadDependingFW" block (only relevant for Hoist and Holding Gear)

   The load-dependent curve must be entered in the "LoadDependingFW" block and the factors must be determined in the commissioning phase; refer to Field weakening (LDFW) (Page 414).
9. "OverSpeed" block (relevant for Hoist, Holding Gear, and Luffing Gear)

   When required, the parameter threshold is adapted in the "OverSpeed" block in order to identify an overspeed condition.
10. "AccelChangeSlewGear" block (relevant for Slewing Gear and Luffing Gear)

    The acceleration limits must be adapted in the "AccelChangeSlewGear" block.
11. "VelocityChangeSlewGear" block (relevant for Slewing Gear and Luffing Gear)

    The velocity limits must be adapted in the "VelocityChangeSlewGear" block.
12. "ContLoadMeasurement" or "ContLoadMeasurement_1" block (relevant for Holding Gear)

    The velocities must be adapted in the "ContLoadMeasurement" block; see Continuous load measurement (CLM) (Page 434)

13. "CurrentEqualControl" or "CurrentEqualControl_1" block (relevant for Closing Gear)

    The smoothing times must be adapted in the "CurrentEqualControl" block.

14. "GrabMonitor" block (relevant for Closing Gear)

    The smoothing time for the torque has to be adapted in the "GrabMonitor" block.

15. "Monitoring" block

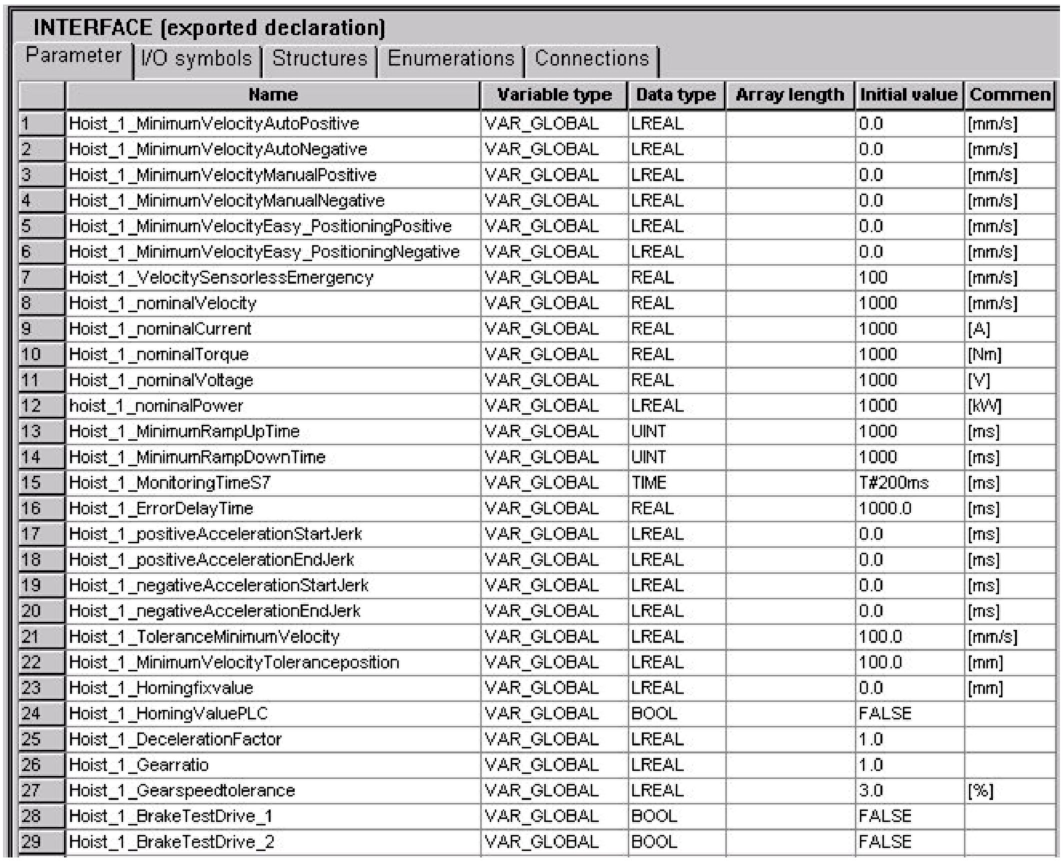
    No modifications are necessary in the "Monitoring" block.

16. "SlackRopeControl" or "SlackRopeControl_1" block (relevant for Holding Gear)

    The current values and velocities must be adapted in the "SlackRopeControl" block.

8.9 User program (MCC program)

The following variables must be configured for every function module in each user program:



	Name	Variable type	Data type	Array length	Initial value	Commen
1	Hoist_1_MinimumVelocityAutoPositive	VAR_GLOBAL	LREAL		0.0	[mm/s]
2	Hoist_1_MinimumVelocityAutoNegative	VAR_GLOBAL	LREAL		0.0	[mm/s]
3	Hoist_1_MinimumVelocityManualPositive	VAR_GLOBAL	LREAL		0.0	[mm/s]
4	Hoist_1_MinimumVelocityManualNegative	VAR_GLOBAL	LREAL		0.0	[mm/s]
5	Hoist_1_MinimumVelocityEasy_PositioningPositive	VAR_GLOBAL	LREAL		0.0	[mm/s]
6	Hoist_1_MinimumVelocityEasy_PositioningNegative	VAR_GLOBAL	LREAL		0.0	[mm/s]
7	Hoist_1_VelocitySensorlessEmergency	VAR_GLOBAL	REAL		100	[mm/s]
8	Hoist_1_nominalVelocity	VAR_GLOBAL	REAL		1000	[mm/s]
9	Hoist_1_nominalCurrent	VAR_GLOBAL	REAL		1000	[A]
10	Hoist_1_nominalTorque	VAR_GLOBAL	REAL		1000	[Nm]
11	Hoist_1_nominalVoltage	VAR_GLOBAL	REAL		1000	[V]
12	hoist_1_nominalPower	VAR_GLOBAL	LREAL		1000	[kW]
13	Hoist_1_MinimumRampUpTime	VAR_GLOBAL	UINT		1000	[ms]
14	Hoist_1_MinimumRampDownTime	VAR_GLOBAL	UINT		1000	[ms]
15	Hoist_1_MonitoringTimeS7	VAR_GLOBAL	TIME		T#200ms	[ms]
16	Hoist_1_ErrorDelayTime	VAR_GLOBAL	REAL		1000.0	[ms]
17	Hoist_1_positiveAccelerationStartJerk	VAR_GLOBAL	LREAL		0.0	[ms]
18	Hoist_1_positiveAccelerationEndJerk	VAR_GLOBAL	LREAL		0.0	[ms]
19	Hoist_1_negativeAccelerationStartJerk	VAR_GLOBAL	LREAL		0.0	[ms]
20	Hoist_1_negativeAccelerationEndJerk	VAR_GLOBAL	LREAL		0.0	[ms]
21	Hoist_1_ToleranceMinimumVelocity	VAR_GLOBAL	LREAL		100.0	[mm/s]
22	Hoist_1_MinimumVelocityToleranceposition	VAR_GLOBAL	LREAL		100.0	[mm]
23	Hoist_1_Homingfixvalue	VAR_GLOBAL	LREAL		0.0	[mm]
24	Hoist_1_HomingValuePLC	VAR_GLOBAL	BOOL		FALSE	
25	Hoist_1_DecelerationFactor	VAR_GLOBAL	LREAL		1.0	
26	Hoist_1_Gearratio	VAR_GLOBAL	LREAL		1.0	
27	Hoist_1_Gearspeedtolerance	VAR_GLOBAL	LREAL		3.0	[%]
28	Hoist_1_BrakeTestDrive_1	VAR_GLOBAL	BOOL		FALSE	
29	Hoist_1_BrakeTestDrive_2	VAR_GLOBAL	BOOL		FALSE	

Figure 8-37 Symbol browser extract from the MCC program Hoistp_1

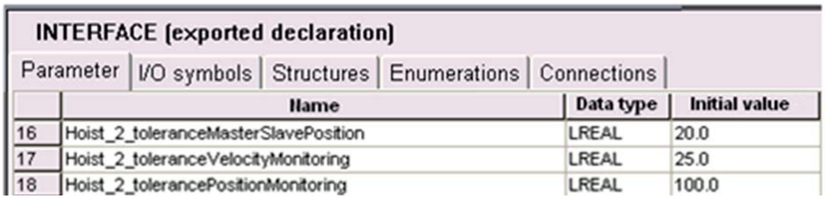
- Minimum velocity for the positive and negative directions, each for the AUTOMATIC, MANUAL and EASY_POSITIONING operating modes.
- Velocity for the SENSORLESS EMERGENCY operating mode

- Rated velocity; this means that the system identifies when the drive is in the field weakening mode.
- The following information must be entered:
  - Reference current
  - Reference torque
  - Reference voltage
  - Reference power
  - Rated velocity

  This information must be available so that the corresponding values can be converted.
- Fixed homing position if no HomingValuePLC selection has been made.
- An initial and final rounding-off; this permits soft starting and stopping of a traversing motion.
- DecelerationFactor; decreases the deceleration.
- Determination of BreakTestDrive; see Selecting a function module (Page 210)
- Minimum acceleration and deceleration times; this guarantees that values below the configured times are not accepted.
- SIMATIC S7 selects the homing position if S7 specifies the homing position.
- For a rigid connection between Hoist 1 and Hoist 2, master-slave operation is active. A certain position tolerance must not be exceeded.
- The position, velocity and current tolerances are monitored in the monitoring block.
- ToleranceMinimumVelocity or MinimumVelocityTolerancposition are provided for the changeover point of the minimum velocity for the AUTOMATIC, MANUAL and EASY_POSITIONING operating modes.

Note

There are a couple of additional variables for the function module Hoist 2, which is also operated as a slave of Hoist 1.



	Name	Data type	Initial value
16	Hoist_2_toleranceMasterSlavePosition	LREAL	20.0
17	Hoist_2_toleranceVelocityMonitoring	LREAL	25.0
18	Hoist_2_tolerancePositionMonitoring	LREAL	100.0

Figure 8-38 Symbol browser extract from the MCC program Hoistp_2

8.10 Field weakening (LDFW)

8.10.1 General description

For closed-loop control drives, an attempt is made to increase the speed above the rated speed for various operating cases. This is used, for example, for Hoist of cranes to increase the handling capacity.

However, the maximum speed in the rated speed range that can be achieved with rated flow is determined by the maximum permissible voltage. This means that an increase above this speed cannot be achieved by increasing the voltage; both for DC and AC drives, this is only possible using field weakening.

However, with field weakening the available torque is reduced. This means that field weakening is only possible if also a reduced torque is demanded. For Hoist, this is for example only possible for partial loads less than the rated load.

Calculate the maximum valid speed ω for this speed increase in the rated range at (maximum) rated power $P = P_N$ by means of the equation $P = M \bullet \omega$, with initially constant flow and depending on torque requirements.

In this case, the upper speed limit is the maximum permissible speed for the motor.

A measure for required load torque in the rated speed range is the current in the steady-state phase which is directly proportional to load torque.

However, in order to achieve a seamless transition from the rated speed range into the field-weakening range while accelerating, it is necessary to determine this supplementary speed setpoint while accelerating in the rated speed range and to enter this before reaching the transition point from the rated speed range into the field weakening range.

However, in the acceleration phase, in addition to the steady-state (static) load current, this current also includes the dynamic components to accelerate the load and the rotating masses.

In order to determine the steady-state (static) component of the current already while accelerating, it is necessary to calculate the dynamic components while accelerating and to subtract these from the total current in order to obtain the steady-state (static) component corresponding to the load. Only this steady-state (static) component can be used to generate a load-dependent supplementary speed setpoint for constant power.

While accelerating, the DCC block calculates the steady-state (static) component of the current - which must be provided for the prevailing torque - from the total current. To calculate this current, the acceleration component of the load and the acceleration component of the rotating masses is taken into account.

For reasons of precision, the arithmetic average value is generated when making the calculation. Further, dependent on the calculated torque, the supplementary speed setpoint for field weakening, permissible for $P = P_N$ is formed using the motor power characteristic.

8.10.2 Theoretical basics and equations

The steady-state load torque is calculated using the following torque equation.

$M_{Motor} = M_{Load} + M_{ALoad} + M_{ARot} + M_{Friction}$

M_{motor}	Motor torque
M_{Load}	Load torque (scanned):
M_{ALoad} :	Load acceleration torque (proportional to acceleration):
M_{ARot} :	Acceleration torque as a function of gear unit inertia, for example (proportional to acceleration)
$M_{Friction}$:	Frictional torque

$$M_{Motor} = M_{Load} + M_{Load} \bullet k_{ALoad} \bullet \frac{dn}{dt} + k_{ARot} \bullet \frac{dn}{dt} + M_{Friction}$$

k_{Aload} and k_{Arot} are proportional factors whose quantity must be derived from measurement. $M_{friction}$ must also be determined by measurement.

$$M_{Motor} = M_{Load} \bullet \left(1 + k_{ALoad} \bullet \frac{dn}{dt} \right) + k_{ARot} \bullet \frac{dn}{dt} + M_{Friction}$$

$$M_{Load} = \frac{M_{Motor} - M_{Friction} - k_{ARot} \bullet \frac{dn}{dt}}{\left(1 + k_{ALoad} \bullet \frac{dn}{dt}\right)}$$

This equation only applies to hoisting and positive dn/dt when accelerating in the hoisting direction. Torque can be derived from the constant current flow in the rated speed range of the drive, as current is directly proportional to torque.

$$\text{Load} \sim M_{Load} \sim I_{Load} = \frac{I_{Motor} - I_{Friction} - k_{ARot} \bullet \frac{dn}{dt}}{\left(1 + k_{ALoad} \bullet \frac{dn}{dt}\right)}$$

To determine the load, it is necessary to measure the speed change dn/dt. This is realized using a differentiating element in the DCC block "DCC_LoadDependingFieldWeak". The actual dn/dt values and the actual current value of the motor I_{motor} is continually transferred to SIMOTION and is measured there. The measurement must have been completed within the rated speed range of the machine in order to determine the corresponding load and therefore to calculate the maximum field weakening speed. The measurement is made at 40% to 90% of the rated speed range of the machine; refer to DCC_LoadDependingFieldWeak (Page 106).

⚠ WARNING
This setting should only be carried out by qualified personnel. Particular attention must be paid to the reduced breakdown torque of the drive within the field weakening range. Allowances must be made for appropriate safety reserve in terms of the data provided by the machine manufacturer.

8.10.3 Commissioning information

8.10.3.1 Generating the measurement quantities

Sufficient measurements must be made when determining the average acceleration and the average current while accelerating in order to obtain a realistic value. Assuming an acceleration time of 3 s for reaching maximum speed (field weakening speed), approx. 1.5 s (N_{Fw} = 2 x N_{rated}) will remain until the rated speed has been reached.

The arithmetic average value is generated by integrating all of the values obtained in a measuring window for each sampling operation.

If the measurement window is 40-90 % of the rated speed, then approx. 750 ms is available for the measurement. Approx. 25-30 measurements should be made within this 750 ms. The result is a valid maximum cycle time of 20 ms for calculating the DCC block. All dn/dt values and current actual values are added during the measurement phase. After the measurement phase has been completed, this sum is divided by the number of measurements and therefore the average value generated.

The number of measurements can be read-out in "rCountsMeasurRange". A least 25 measurements must be made, otherwise the measured load is declared as non-valid.

The drive must cover the entire measurement window for new measurements and the previous measurement must be deleted (by means of a positive edge at boResetLoadCurrent), for otherwise it is not possble to determine any additional velocities. If the old measurement is not deleted, then the values last determined are available at the output of the DCC block.

For the hoisting direction, the following always applies:

- POSITIVE acceleration (100% at the rated ramp-up time).
- POSITIVE actual speed (100% at the field weakening speed)
- Current actual value (100% at the rated motor current) must be POSITIVE.

Note

At maximum acceleration and at the end of the measuring window,

- the dn/dt average value (rAverage_dn_dt) should be approx. 100%; the measurement window limits must then be adapted accordingly.
- the number of measurements (rCountsMeasurRange) must be at least 25.

If a new measurement is to be made, then it is absolutely necessary that a positive edge is connected to connection "boResetLoadCurrent".

All measurements must always be taken in SPEED_CONTROLLED mode.

⚠WARNING

If the rated motor current is not used as reference, then the subsequent values here (rFact_AccCurrent_Load, rFact_AccCurrent_RotMass, rFrictionCurrent) must be correspondingly adapted.

Example:
"rFact_AccCurrent_Load" should be between 5 % and 10 % if the current (rCurrentSetpoint) is referred to the rated motor current (rCurrentNominal). However, if for example, the current is referred to twice the rated current, then this value must be converted and should now be between 2.5 % and 5 %. All of the values in the following table are empirical and only serve as a guide.
The values at the actual crane system can deviate from the values specified here.

Table 8- 7 Standard assignment of the setting connections before commissioning

Name	Value	Description
rFact_AccCurrent_Load	0.0 %	Adaptation factor (referred to the rated load) for the acceleration component of the load Rough order of magnitude: 5 %- 10 % This corresponds to the current value to be compensated.
rFact_AccCurrent_RotMass	0.0 %	Adaptation factor (referred to the rated load) for the acceleration component of the rotating masses Rough order of magnitude: 20 % - 40 % This corresponds to the current value to be compensated.
rFrictionCurrent	0.0 %	Frictional component (referred to the rated load) for the current Rough order of magnitude: 1 % - 3 % This corresponds to the current value to be compensated.

Name	Value	Description
rEfficiency_eta	100 %	Adaptation factor (referred to the rated load) for the difference in efficiency when raising and lowering Rough order of magnitude: 110 % - 120 %
rLoadNormFactor	1.0	Adaptation factor to convert into a physical quantity The conversion is made using the formula: Load [t] = measured value * rLoadNormFactor + rAddCurrentBeforePLI
rAddCurrentBeforePLI	0.0	Offset to determine the load: Load[t] = measured value * rLoadNormFactor + rAddCurrentBeforePLI
rLowLimitMeasRange	40.0 %	Start measurement window is already referenced to the rated speed. No other conversion is required.
rUpLimitMeasRange	90.0 %	End measurement window is already referenced to the rated speed. No other conversion is required
rSmoothTime_dn_dt	24 ms	Smoothing time for the velocity setpoint applied to the derivative action element. Order of magnitude: 0 ms – 150 ms
rCurrentSmoothTime	120 ms	Smoothing time for the current [ms] Order of magnitude: 0 ms – 200 ms

Note

The orders of magnitude specified in the above table are guide values.

Table 8- 8 Important outputs of the DCC block for commissioning

Name	Value	Description
rCountsMeasurRange	-	Number of sampling operations in the measuring window min. 25
rAverageSetpointCurrent	%	Arithmetic average value, current setpoint
rAverage_dn_dt	%	Arithmetic average value dn/dt (approx. 100 %)
rJBrot	%	Acceleration component of the rot. masses
rJBLoad	%	Acceleration component of the load
rMm_minusM_Friction	%	Total current - frictional current
rMm_minus_JBrot_Mf	%	I_M - I_{ARot} - $I_{Friction}$
rLoadCurrent_with_eta	%	Load current with efficiency correction
rLoadCurrent_without_eta	%	Load current without efficiency correction
rLoadCurrent_plus_Frict	%	Total current + frictional current
rLoadCurrent_before_PLI	%	Load current before the power characteristic
rLoadWithNormFactor	-	Evaluated load actual value
rSetpointAfter_PLI	%	Auxiliary speed setpoint relative to rMaximumVelocity after PLI
rAddSetpoint_PLI	%	Auxiliary speed setpoint with offset relative to rMaximumVelocity after PLI

Name	Value	Description
rOutSetpointVelocity	-	Auxiliary speed setpoint logically linked with the master switch setpoint
boIact_greater_In	-	Actual current higher than current setpoint with hysteresis (False)
boResetFieldWeak	-	Reset field weakening
boOutEnableFieldWeak	-	Enable field-weakening

8.10.3.2 Commissioning support for LDFW with FB_AutoSettingFW

Note

The use of the FB_AutoSettingFW block is optional. If you use this block, you do not need to take Chapters Compensating the frictional current (Page 423) to Correcting the efficiency (Page 425) into account.

Only when you use this block can the ropes be tightened early and the container spreader / grab fully mounted. Otherwise, observe the procedures Compensating the frictional current (Page 423) to Correcting the efficiency (Page 425).

Note

- The rated velocity must not be exceeded during the commissioning with this block. The acceleration and deceleration should have the values with which the crane is operated later.
- The hoisting and lowering actions should always run without interruption. Always traverse at the rated velocity.
- The motors should be warmed up prior to the commissioning.

The FB_AutoSettingFW simplifies the commissioning so that at the end all important input parameters for the DCC_LoadDependFieldweak_1 are issued at this block output. The commissioning engineer then only needs to accept them at the inputs of the LDFW block.

If the DCC block DCC_ContLoadMeasurement_1 is also required on the crane, you do not need to repeat the measurements. This block determines automatically the appropriate inputs for both DCC blocks.

Two measuring steps must be performed during the commissioning. These two measuring steps should be executed together without a significant pause in between.

The following inputs have already been linked in the standard projects. If a block is added later, the links must be augmented in accordance with the following table.

Table 8- 9 Subsequent links

Name of the input	Logic operation
TO_Name	Link with the hoist drive - master axis (hoist or holding gear)
boReset	This input can be used to reset the measuring Step10 or Step20 or all measured values. • boAutoSettingFWStep10 = TRUE and boReset Only the measured values from Step10 are deleted. • boAutoSettingFWStep20 = TRUE and boReset Only the measured values from Step20 are deleted. • Keep boReset pressed for three seconds. All measured values are reset.
r64CurrentNominal	Enter rated current
r64CurrentSetpoint	Link with the current setpoint of the master
r64SmoothTimeCurrentSetpoint	Smoothing time for the current setpoint [ms] Change only if the current fluctuates excessively.
r64VelocityNominal	Enter rated velocity [mm/s]
r64RampUpTimeNominal	Ramp-up time to the rated velocity [ms]
r64UpLimitMeasRange	Upper measuring window limit [%] Change this value only if the measurement window does not suffice during the acceleration process or when the measuring range should be moved.
r64LowLimitMeasRange	Lower measuring window limit [%] Change this value only if the measurement window does not suffice during the acceleration process or when the measuring range should be moved.
r64MinLoad	Load value without load but with container spreader / grab [in % or physical unit (kg or t), depending on what is present at the LDFW block (Loadnormfactor)] **Note:** The load with which the crane may travel with maximum velocity is required here. [r64MinLoad = Maxvelocity]
r64MaxLoad	Load value with rated load, container spreader / grab [in % or physical unit (kg or t), depending on what is present at the LDFW block (Loadnormfactor)] **Note:** The maximum load with which the crane may travel with the rated velocity is required here. [r64MaxLoad = Nominalvelocity]

Measuring step 1: Measurement without load

Prerequisites:

- The ropes are tightened and the container spreader / grab is fully mounted.
- Limit switches, rated and maximum velocity should be set and all standardizations completed. The crane must be available for several test runs.
- Container spreader / grab must not accept any load and all measurements must be performed without load.
- boStartAutoSettingFW = TRUE → FB_AutoSettingFW is enabled.
- boAutoSettingFWStep10 = TRUE → first measuring step is started.
- boAutoSettingFWStep20 = FALSE

Procedure:

Perform ten uninterruptible hoisting operations. The rated velocity must be reached for each hoist operation and travel constant for the time "r64MinTimeConstantDrive". "r64MinTimeConstantDrive" is available as output value.

A hoist operation is considered to have completed successfully when the "boSingleMeasFinished" output changes to TRUE. The hoist can then be lowered again in order to start a new hoist operation. If this output does not change to TRUE in time, the hoist operation was not valid. An intervention is not currently required and a new hoist operation can be started. After ten valid hoist operations, the "boStep10MeasFinished" output is set to TRUE.

If "boSingleMeasFinished" never changes to TRUE, the required sampling operations during the positive or negative acceleration or during the constant travel will not be reached. A check should be made whether constant travel is made long enough (at least "r64MinTimeConstantDrive") and possibly decrease or increase the measurement window "r64UpLimitMeasRange"/"r64LowLimitMeasRange" slightly.

If measuring step 1 must be repeated, you can use "boReset" to delete the measurement. Ensure, however, that "boAutoSettingFWStep10" is set to TRUE.

Note

If "boReset" is pressed for three seconds, all measuring steps will be reset irrespective of "boAutoSettingFWStep10/20".

Measuring step 2: Measurement with load

Prerequisites:

- The ropes are tightened and the container spreader / grab is fully mounted.
- Limit switches, rated and maximum velocity should be set and all standardizations completed. The crane must be available for several test runs.
- Container spreader / grab must now accept the rated load and all measurements must be performed with the rated load.

- boStartAutoSetting FW = TRUE - FB_AutoSettingFW is enabled
- boAutoSettingFWStep10 = FALSE
- boAutoSettingFWStep20 = TRUE - the second measuring step is started.

Procedure:

Perform ten uninterruptible hoist operations again. The same conditions for measuring step 1 now also apply to measuring step 2. After ten valid hoist operations, "boStep20MeasFinished" is now set to TRUE.

If measuring step 2 must be repeated, you can use "boReset" to delete the measurement. Ensure, however, that "boAutoSettingFWStep20" is set to TRUE.

Note

If "boReset" is pressed for three seconds, all measuring steps will be reset irrespective of "boAutoSettingFWStep10/20".

You should avoid any unnecessary waiting times in the associated measuring steps. If the waiting time exceeds three minutes, the associated measuring step is deleted and must be repeated. This serves to ensure that the measurements are performed with a constant motor temperature.

Completion

If "boStep10MeasFinished" and "boStep20MeasFinished" are both TRUE, the following outputs are not equal to 0. The values can now be accepted for LDFW (and, if required, also for CLM).

- r64AutoFrictionCurrent
- r64AutoFactAccCurrRotMa
- r64AutoFactAccCurrentLoad
- r64Eta_LDFW (only for the LDFW block)
- r64Eta_lift_CLM (only for the CLM block)
- r64Eta_drop_CLM (only for the CLM block)

If a value is present at the "r64MinLoad" or "r64MaxLoad" input, the following output values are also output automatically:

- r64X1ParameterLoad
- r64Y1ParameterVelocity
- r64X2ParameterLoad
- r64Y2ParameterVelocity
- r64X3ParameterLoad
- r64Y3ParameterVelocity
- r64X4ParameterLoad
- r64Y4ParameterVelocity

- r64X5ParameterLoad
- r64Y5ParameterVelocity
- r64X6ParameterLoad
- r64Y6ParameterVelocity

If an error occurs, "boErrorFunctionBlock" output is TRUE and "i32ErrorIDFunctionBlock" sends the error ID. If "i32ErrorIDFunctionBlock" is not equal to 0 but "boErrorFunctionBlock" = FALSE, this is an alarm. Chapter Application error messages and alarm messages (Page 339) contains a complete list and description of all error and alarm messages of the function blocks.

Note

After completion of all measurements, you can delete a specific measuring step. This must be activated beforehand (Step10 or Step20). "boReset" can then be used to delete the activated measuring step. This does not affect the other measuring step and the outputs are formed again taking the two measuring steps into account. This means both measuring steps do not need to be repeated in the event of an error. Do not press "boReset" longer than three seconds, otherwise the two measuring steps will be reset.

Note

Finally, a plausibility check should also be performed. Empirical values can be found at Table 8-7 Standard assignment of the setting connections before commissioning (Page 417).

8.10.3.3 Compensating the frictional current

$$\text{Load} \sim M_{\text{Load}} \sim I_{\text{Load}} = \frac{I_{\text{Motor}} - \boxed{I_{\text{Friction}}} - k_{\text{ARot}} \bullet \frac{dn}{dt}}{\left(1 + k_{\text{ALoad}} \bullet \frac{dn}{dt}\right)}$$

Setting instructions:

Move the empty drum at constant rated speed (n_{rated}) in the hoisting direction. The inputs "rCurrentNominal" and "rCurrentSetpoint" must be appropriately supplied with the ST variables "hoist_X_setpoint_current_for_dcc" and "hoist_X_currentnominal_for_dcc". 120 ms has proven itself as a good value for "rCurrentSmoothTime". However, this value can be individually adapted. The friction current value can be read out as a percentage using "rSmoothSetpointCurrent".

Repeat the measurement several times in the hoisting direction and enter the average value (with the motor in the warm state and the gearbox oil warm) in "rFrictionCurrent" (approx. 1% to 3%).

This means that the frictional current will be taken into account at output "rmm_minusm_friction".

8.10.3.4 Compensating the acceleration torque of the rotating masses

$$\text{Load} \sim M_{\text{Load}} \sim I_{\text{Load}} = \frac{I_{\text{Motor}} - I_{\text{Friction}} - \boxed{k_{\text{ARot}} \bullet \frac{dn}{dt}}}{\left(1 + k_{\text{ALoad}} \bullet \frac{dn}{dt}\right)}$$

Setting instructions:

At the input of the DCC block, set "rLowLimitMeasRange" to 40 % and "rUpLimitMeasRange" to 90 %. These values are automatically referenced to the rated speed and do not require any additional conversion. However, if 100% is not reached at output "rAverage_dn_dt", there are two ways of remedying this:

- Change the smoothing time (rSmoothTime_dn_dt) for the velocity setpoint applied to the derivative action element to correct minor deviations
  or
- Change of measuring window limits (rUpLimitMeasRange or rLowLimitMeasRange).

  Tip:
  If the value reached is significantly higher than 100%, reduce "rUpLimitMeasRange".
  If the value reached is significantly lower than 100%, reduce "rLowLimitMeasRange".

Accelerate the empty drum, hoisting (raising) up to the rated speed. At the end of the measurement phase (boOut_EnableFieldWeak is true) the value at the output of the DCC block "rMm_minus_JBrot_Mf" (I_{Motor} - I_{Friction} - k_{ARot} • dn/dt) should be 0 %, as in this case, no acceleration component of the load is included in the total current. Correspondingly vary using the connection "rFact_AccCurrent_RotMass" (approx. 20 % to 30 %). Confirm these values by making several acceleration trials.

Note

Connect a positive edge at input "boResetLoadCurrent" after every measurement. "rAverage_dn_dt" must be + 100%.

Note

After you have made this setting, you can attach the rope.

8.10.3.5 Compensating the acceleration torque of the load

$$\text{Load} \sim M_{\text{Load}} \sim I_{\text{Load}} = \frac{I_{\text{Motor}} - I_{\text{Friction}} - k_{\text{ARot}} \bullet \frac{dn}{dt}}{\boxed{\left(1 + k_{\text{ALoad}} \bullet \frac{dn}{dt}\right)}}$$

Setting instructions:

Set connection "rEfficiency_eta" to 100 %. Approximately the rated load (=max. load) should be suspended on the ropes. Accelerate hoisting to the valid speed.

After the measurement phase has been completed, connection "rLoadCurrent_plus_Frict" should, as far as possible, correspond to the value that the drive draws after the acceleration phase (only in the hoisting direction) when the load in a steady-state (rSmoothSetpointCurrent). Correspondingly vary connection "rFact_AccCurrent_Load" (approx. 5 % to 10 %). Confirm these values by making several acceleration trials.

Note

Connect a positive edge at input "boResetLoadCurrent" after every measurement. "rAverage_dn_dt" must be + 100%.

⚠WARNING

For safety reasons, AppSTW1 bit4 "enable field weakening" should be FALSE. Accelerate hoisting only to the speed specified for rated loads (maximum load).

8.10.3.6 Correcting the efficiency

The efficiency correction should be used to ensure that with the same load, the velocity when lowering is identical to the velocity when hoisting (raising). For hoisting and lowering, in order to determine the same load actual value at different current values, it is necessary to multiply the value present when lowering "rLoadCurrent_without_eta" by the factor $1/\eta^2$.

The setting is made with "rEfficiency_eta". The connection "rLoadCurrent_with_eta" should neither change when hoisting nor when lowering.

$$\eta^2 = \frac{\text{rLoad\_with\_eta}_{\text{Hoisting}}}{\text{rLoad\_with\_eta}_{\text{Lowering}}} \bullet 100\ \%$$

Calculation example:

- Displayed value with rated load, hoisting:
  $\text{rLoad\_with\_eta}_{\text{hoisting}}$ = 98%
- Displayed value with rated load, lowering:
  $\text{rLoad\_with\_eta}_{\text{lowering}}$ = 82%

$$\eta^2 = \frac{98\ \%}{82\ \%} \cdot 100\ \% = 119{,}51$$

Calculated parameter (rEfficiency_eta_drop) = 119.51%
After setting "rEfficiency_eta_drop": set "rEfficiency_eta_lift" accordingly.

Setting instructions:

1. Move the rated load in hoisting direction and then in lowering direction at constant rated speed (n_{rated} motor), assuming rEfficiency_eta = 100% (default setting).
2. Repeat your measurement several times.
3. Note down the mean value of "rLoadCurrent_with_etaHoisting" and "rLoadCurrent_with_etaLowering".
4. Calculate the rEfficiency_eta corresponding to the specified example.

> **Note**
>
> Connect a positive edge at input "boResetLoadCurrent" after every measurement.

5. Check the settings by hoisting and lowering the rated load (up to the rated speed) several times.

   In so doing "rLoadCurrent_with_eta" and "rLoadWithNormFactor" should no longer change when hoisting and lowering.
6. Adapt the factor rEfficiency_eta (approx. 115 %) to keep the change between hoisting and lowering as low as possible.

> **Note**
>
> If the load when lowering is determined to be higher than the actual load, a higher value of "rLoadCurrent_with_eta" while lowering with respect to hoisting ensures that the drive is not overloaded when braking the load.

This setting is only necessary for the case that the load should be determined while lowering (e.g. at the start of lowering). For normal applications, it is sufficient if the determined load is only re-calculated again if it has changed. Normally, this is only possible after the spreader has taken-up a new load. This means that the signals "Twistlocks open" and "Twistlocks closed" can be used to set a positive edge of the signal "boResetLoadCurrent". This allows the load to be recalculated.

8.10.3.7 Calculating the physical size of the load

The determined load can be converted to a physical quantity (e.g. tons) using connections "rLoadNormFactor" and "rAddCurrentBeforePLI".

Calculation example:

When hoisting a rated load of 45 t, "rLoadWithNormFactor" indicates a value of 85 %.

$$\text{rLoadNormFactor} = \frac{45}{85} = 0{,}5294$$

A constant load can be added (e.g. in tons) using "rAddCurrentBeforePLI".

8.10.4 Procedure for drives with a switchable gear

In the case of drives with a switchable gear, a separate DCC_LoadDependingFieldWeak block and a separate DCC_Overspeed block must be available for each gear ratio. The blocks must be put into operation with the corresponding ratio and the relevant parameters need to be adjusted to the modified behavior. These steps also need to be taken for any slave drives (slave, Hoist 2) which may be present.

Furthermore, the selection of the blocks' output signals needs to be linked to the selection of the gear stage. To do this, corresponding selector switches (NSW, BSW) must be connected to the block outputs. These must be controlled by the S7 via a shared switchover bit.

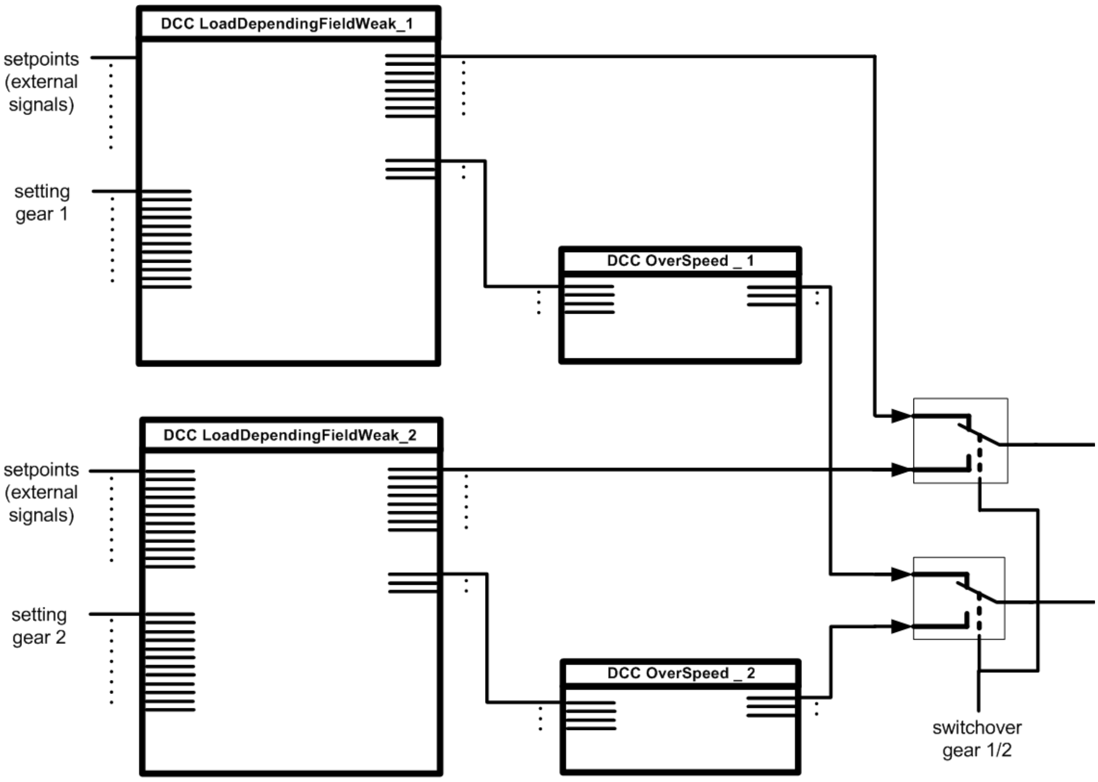

DCC LoadDependingFieldWeak_1
setpoints (external signals)
setting gear 1
DCC OverSpeed _ 1
DCC LoadDependingFieldWeak_2
setpoints (external signals)
setting gear 2
DCC OverSpeed _ 2
switchover gear 1/2

Figure 8-39 Use of two LoadDependingFieldWeak blocks

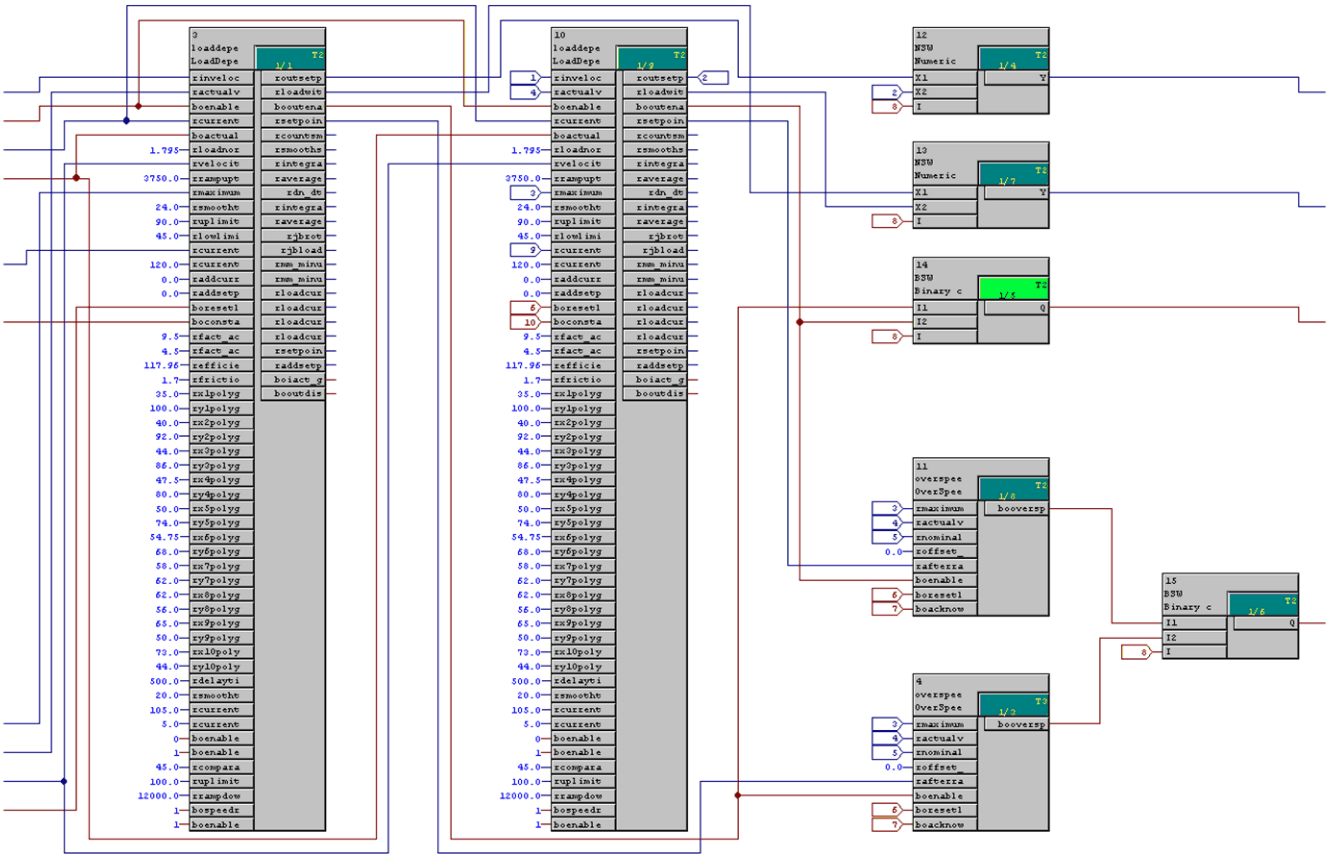


Figure 8-40 Two LoadDependingFieldWeak blocks and selector switches (NSW, BSW) in a DCC chart

8.10.5 Criteria for enabling field weakening

The criteria for enabling field weakening are described in this section. Field weakening is not enabled or is inhibited as long as "boEnableFieldWeak" is set to false.

While field weakening is not enabled, the master switch setpoint is limited to the rated speed. As soon as field weakening is enabled (boOutEnableFieldWeak = True), the master switch setpoint is limited to the calculated load-dependent speed limiting value.

The criteria to enable field weakening – dependent on the actual speed value and speed setpoint – are described in the following:

1. The actual velocity (rActualVelocity) is scanned at the DCC block input as to whether the upper limit of the measurement window (UpLimitMeasRange) has been exceeded and whether at least 25 measurements (rCountsMeasurRange) have been performed – otherwise the measured load is considered to be invalid.

2. Further, the lower measurement window limit (rLowLimitMeasRange) must have been passed through in order to declare the measurement valid. This means that at the start of the measurement, the actual velocity value must be below the measurement window and then – as described in Point 1 – the upper measurement window limit exceeded. This means that the measurement window has been completely run-through once.
3. The user can select the precise conditions for the enabling the field weakening using the master switch. DCC_LoadDependingFieldWeak and DCC_LoadDependFieldWeak_1 have two or three possibilities, respectively; see DCC_LoadDependFieldWeak_1 (Page 113), "Mode of operation" section.

8.10.6 Reducing the supplementary setpoint for an incorrect calculation

After calculating the supplementary speed setpoint, the ramp-function generator - that is connected as motor potentiometer here is set to this calculated value.

Further, the current actual value is monitored in the steady-state phase (boConstantMovement is true), if "boEnableCurrentMonitor" was set to 1.

This monitoring checks the calculation of the load-dependent supplementary setpoint. If the current actual value exceeds the rated current (I_N = 100 %), during the steady-state phase, then the calculation of the supplementary setpoint is incorrect. In this case, the drive is neither shut down via the closed-loop control monitoring nor via the overcurrent monitoring (the current limit is normally 180%), then the motor would be overloaded from a power perspective.

This is followed by a time-delayed correction of the supplementary setpoint. The time delay should be entered into "rDelayTimeConstantMove". It is required for the stabilization of the current actual value.

If, for reasons previously described, the speed setpoint is incorrectly calculated, then the current actual value can increase above the rated current in the steady-state phase. In this case, the speed setpoint is reduced via the ramp-function generator – and more precisely, until the current actual value re-enters the hysteresis loop or the rated speed is reached.

"rCurrentInterval" and "rCurrentHysteresis" should be set together corresponding to the required response threshold of the current monitoring.

Setting instructions:

The required response threshold of the current monitoring should be defined or determined when commissioning:

Example:

rCurrentInterval + rCurrentHysteresis = 105 % + 5 % = 110 % "on"

rCurrentInterval + rCurrentHysteresis = 105 % - 5 % = 100 % "off"

rCurrentInterval:	Current response threshold, e.g. 105 %
rCurrentHysteresis:	Current hysteresis, e.g. 5 % i.e. between 100 % and 110 %
rUpLimit_RampGen:	100 % = maximum speed
rRampDownTime_RampGen:	Ramp-down time when reducing the speed setpoint (experience value: 12 seconds)
boSpeedRampToLowLim:	to 1 (enable signal to reduce the speed setpoint)
boEnableCurrentMonitor:	to 1 (to enable the current monitoring)

Note

For Basic Technology, 100 % current corresponds to the maximum current!

Note

If "boSpeedRampToLowLim" is not True and current monitoring is active (boIact_greater_In = False), the setpoint is not stepped down and field weakening is disabled directly. This limits acceleration to the rated speed until a new measurement was initiated (positive edge at boResetLoadCurrent).

8.10.7 Examples of load curves for load-dependent field weakening

If current values were converted into a physical quantity using the factor "rLoadNormFactor", then the value "Load (A)" (refer to the subsequent tables) is directly specified in tons instead of – as before – as a percentage of current.

8.10.7.1 Example 1: Constant power

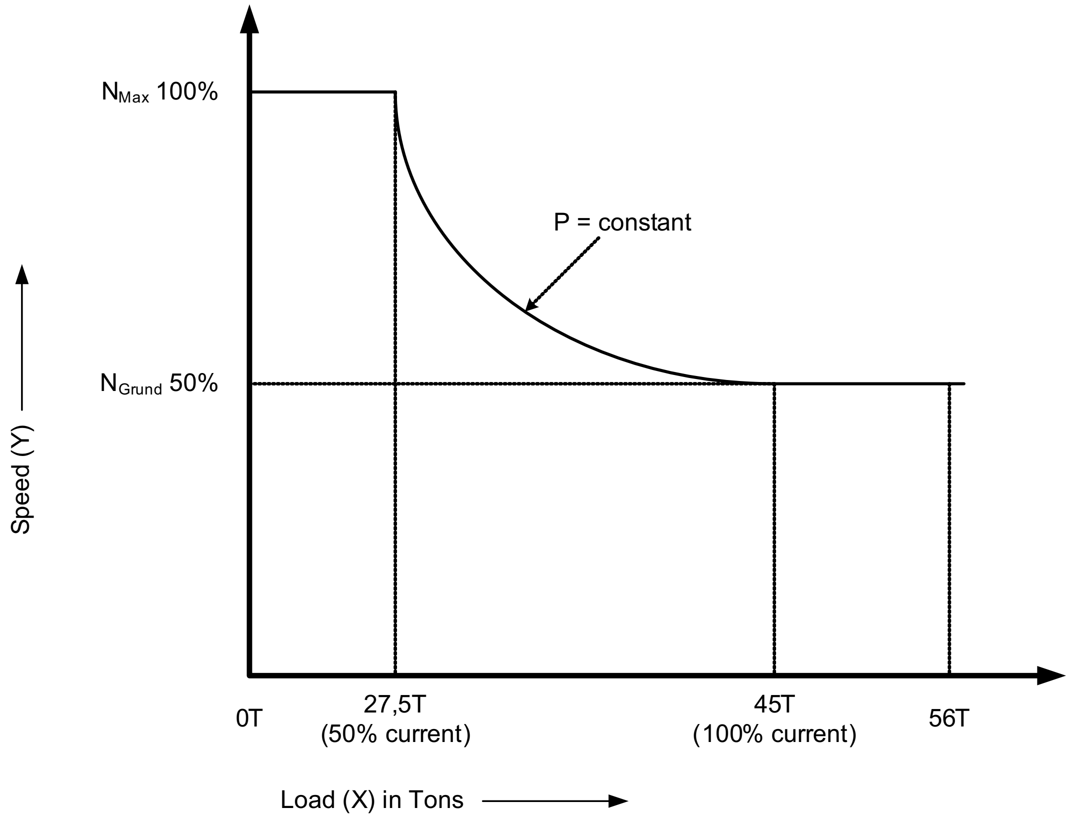


Figure 8-41 Load-dependent field weakening, example 1

Load (X)	Velocity (Y)
X1 = 50%	Y1 = 100%
X2 = 55%	Y2 = 90.9%
X3 = 60%	Y3 = 83.3%
X4 = 65%	Y4 = 76.9%
X5 = 70%	Y5 = 71.4%
X6 = 75%	Y6 = 66.6%
X7 = 80%	Y7 = 62.5%
X8 = 85%	Y8 = 58.8%
X9 = 90%	Y9 = 55.5%
X10 = 100%	Y10 = 50%

Formula:

$Y = \frac{100}{X} \cdot 50$ $X = \frac{100}{60} \cdot 50 = 83{,}\overline{3}$

⚠WARNING

- If excessively high current values occur during constant-velocity travel, then the velocity (Y2-Y9) must be reduced.
- The initial points X1 (load) and Y1 (velocity) as well as end points X10 and Y10 are fixed. The load points (X2-X9) can assume any intermediate values. The corresponding intermediate values (Y2-Y9) are calculated according to the formula above.
- We recommend that the first points along the curve are selected close to one another as the velocity is inversely proportional to the load.
- If using a linearized drive curve, make sure that its interpolation points (X1 to X10; Y1 to Y10) remain below the constant power curve.

8.10.7.2 Example 2: Curve is specified (no constant power)

This example is valid under the following conditions:

1. It is not possible to utilize drive performance.
2. The customer has delivered a load curve which specifies technical requirements.

0 – 15 T	→	Velocity 100%
15 – 40 T	→	Velocity, refer to curve
40 – 56 T	→	Velocity 50 %

Start by determining the current required for lifting 40 tons at rated speed (no field weakening). The current is directly proportional to the load. This means that for a current measurement of 86% at 40 tons, then at 15 tons 32% will be necessary.

The curve can be calculated using the following formula:

$$n = \frac{1}{Q} \bullet X + Y$$

n: Velocity
Q: Load
X: Constant
Y: Constant

Synchronization with data specified above:

$$100 = \frac{1}{32} \bullet X + Y$$

$$50 = \frac{1}{86} \bullet X + Y$$

$$X = \frac{50}{\left(\frac{1}{32} - \frac{1}{86}\right)}$$ and $$Y = 50 - \frac{X}{86}$$

Result after inserting (constant):
X = 2548.148151 and Y = 20.37037028

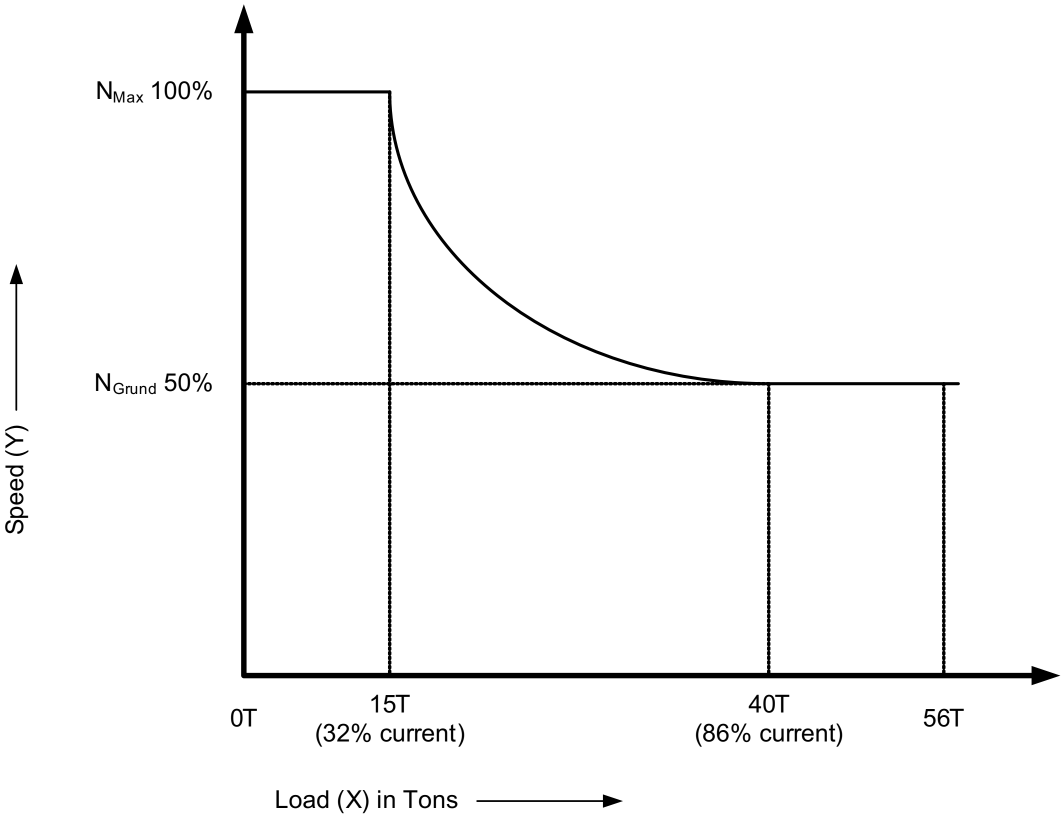


Figure 8-42 Load-dependent field weakening, example 2

The following table is calculated with the formula:

$$n = \frac{1}{Q} \cdot 2548{,}15 + 20{,}37$$

Load (X)	Velocity (Y)
X1 = 32 %	Y1 = 100 %
X2 = 35 %	Y2 = 93.2 %
X3 = 38 %	Y3 = 87.4 %
X4 = 41 %	Y4 = 82.5 %
X5 = 45 %	Y5 = 77.0 %
X6 = 50 %	Y6 = 71.3 %
X7 = 55 %	Y7 = 66.7 %
X8 = 65 %	Y8 = 59.6 %
X9 = 75 %	Y9 = 54.3 %
X10 = 86 %	Y10 = 50.0 %

WARNING

- If excessively high current values occur during constant-velocity travel, then the velocity (Y2-Y9) must be reduced.
- The initial points X1 (load) and Y1 (velocity) as well as end points X10 and Y10 are fixed. The load points (X2-X9) can assume any intermediate values. The corresponding intermediate values (Y2-Y9) are calculated according to the formula above.
- We recommend that the first points along the curve are selected close to one another as the velocity is inversely proportional to the load and the offset. Check - T
- If using a linearized drive curve, make sure that its interpolation points (X1 to X10; Y1 to Y10) remain below the constant power curve.

8.11 Continuous load measurement (CLM)

8.11.1 General description

Using the DCC_ContLoadMeasurement and DCC_ContLoadMeasurement_1 blocks, for the holding gear, a continuous load measurement is performed. The crane driver can always see the weight of the hoisted load.

In the case of dredgers, the crane driver cannot see the submerged grab bucket. Guided by the continuous load measurement function, the driver can lift a fully loaded bucket out of the water every time. A "Grab touchdown" signal is also displayed.

The DCC_ContLoadMeasurement and DCC_ContLoadMeasurement _1 blocks should be set above water if at all possible.

8.11.2 Theoretical basics and equations

The steady-state load torque is calculated using the following torque equation:

$M_{Motor} = M_{Load} + M_{ALoad} + M_{ARot} + M_{Friction}$

M_{motor} Drive torque

M_{Load} Load torque (value to be calculated)

M_{ALoad} : Load acceleration torque (proportional to acceleration):

M_{ARot} : Acceleration torque as a function of gear unit inertia, for example (proportional to acceleration)

$M_{Friction}$: Frictional torque

$$M_{Motor} = M_{Load} + M_{Load} \cdot k_{ALoad} \cdot \frac{dn}{dt} + k_{ARot} \cdot \frac{dn}{dt} + M_{Friction}$$

k_{Aload} and k_{Arot} are proportional factors whose quantity must be derived from measurement. $M_{friction}$ must also be determined by measurement.

$$M_{Motor} = M_{Load} \bullet \left(1 + k_{ALoad} \bullet \frac{dn}{dt} \right) + k_{ARot} \bullet \frac{dn}{dt} + M_{Friction}$$

$$M_{Load} = \frac{M_{Motor} - M_{Friction} - k_{ARot} \bullet \frac{dn}{dt}}{\left(1 + k_{ALoad} \bullet \frac{dn}{dt} \right)}$$

This equation only applies to hoisting and positive dn/dt when accelerating in the hoisting direction.

To determine the load, it is necessary to measure the speed change dn/dt. This measurement is performed by a differentiating element in the DCC block "DCC_ContLoadMeasurement". The current dn/dt value and the current torque value are continually transferred to SIMOTION and are measured there.

WARNING

This setting may only be made by qualified personnel. Pay particular attention to the reduced breakdown torque of the drive within the field weakening range! Add a safety reserve to the data provided by the machine manufacturer.

8.11.3 Commissioning information

For the hoisting direction, the following always applies:

- POSITIVE acceleration (100% at the rated ramp-up time).
- POSITIVE actual speed (100% at the field weakening speed)
- POSITIVE current torque value (100% at the rated motor torque).

NOTICE

At maximum acceleration, the dn/dt average value (rAccelerationMeanValue) at the end of the measuring window must be approximately +100%; adjustments must then be made accordingly via "rDifferentiationTimeCons" and "rSmoothTime_dn_dt".

DCC_ContLoadMeasurement

⚠ WARNING

If the rated motor torque is not applied as a reference, the following values must be adjusted:

- "rFact_AccTorque_Load1" or "…2"
- "rFact_AccTorque_RotMass1" or "…2"
- All friction torques

Example:
"rFact_AccTorque_Load1" should be between 5% and 10% if the torque (rSetpointTorqueHG) refers to the rated motor torque (rTorqueNominal). However, if for example, the torque is to be referred to twice the rated torque, then this value must be converted and should now be between 2.5% and 5%.

All of the values in the following table are empirical and only serve as a guide. The values on the actual crane system can deviate from the values specified here.

Table 8- 10 Default assignments of setting connections prior to commissioning for gear stage 1

Name	Value	Description
rFact_AccTorque_Load1	0.0%	Adaptation factor (referred to the rated load) for the acceleration component of the load: Rough order of magnitude: 5%- 10 % This corresponds to the torque value to be compensated.
rFact_AccTorque_RotMass1	0.0%	Adaptation factor (referred to the rated load) for the acceleration component of the rotating masses Rough order of magnitude: 20% - 40% This corresponds to the torque value to be compensated.
rY1_Friction_SGST1 to rY10Friction_SGST1	0.0%	Frictional component (referred to the rated load) of total torque Rough order of magnitude: 1% - 3% This corresponds to the torque value to be compensated.
rEfficiency_eta_lift	100%	Adaptation factor (referred to the rated load) for the difference in efficiency when hoisting
rEfficiency_eta_drop	100%	Adaptation factor (referred to the rated load) for the difference in efficiency when lowering
rLimitToStartSawToothGen	10%	The breakaway torque should be overcome (10% is referred to rActualVelocity).
rLimitToStopSawToothGen	30.0	Stop limit from which the sawtooth generator begins counting from the beginning again.
rLoadNormFactor	1.0	Adaptation factor to convert into a physical quantity The conversion is performed using the formula: Load [t] = measured value * rLoadNormFactor

Name	Value	Description
rStartMeasurement	2.0	Start limit from which the sawtooth generator starts the load measurement. (boStartMeasurement)
rStopMeasurement	30.0	Stop limit from which the sawtooth generator stops the load measurement. (boStopMeasurement)
rIntegrationTimeSawTooth	1.0	Integration time for the sawtooth generator [ms]

Table 8- 11 Important outputs of the DCC block for commissioning

Name	Value	Description
rSavedCounts	-	Number of sampling operations in the measuring window
rSetTorqueMeanValue	%	Arithmetic mean of the torque setpoint
rAccelerationMeanValue	%	Arithmetic mean of the acceleration (100%)
rJBrot	%	Acceleration component of the rotating masses
rJBLoad	%	Acceleration component of the load
rMm_minusM_Friction	%	Total torque – friction torque (M_{tot} - $M_{Friction}$)
rMm_minus_JBrot_Mf	%	M_M - M_{ARot} - $M_{Friction}$
rLoad_with_eta	%	Load torque with efficiency correction
rLoad_without_eta	%	Load torque without efficiency correction
rLoad_eta_plus_Frict	%	Total torque + friction torque (M_{tot} + $M_{Friction}$)
rLoad	-	Evaluated actual load value
boGrabTouchdown	-	Message "Grab touchdown"

DCC_ContLoadMeasurement_1

⚠ WARNING

If the rated motor torque is not applied as a reference, the following values must be adjusted:

- rFact_AccTorque_Load
- rFact_AccTorque_RotMass
- All friction torques

Example:
"rFact_AccTorque_Load" should be between 5% and 10% if the torque (rSetpointTorqueHG) refers to the rated motor torque (rTorqueNominal). However, if for example, the torque is to be referred to twice the rated torque, then this value must be converted and should now be between 2.5% and 5%.

All of the values in the following table are empirical and only serve as a guide. The values on the actual crane system can deviate from the values specified here.

Table 8- 12 Standard assignment of the setting connections before commissioning

Name	Value	Description
rFact_AccTorque_Load	0.0%	Adaptation factor (referred to the rated load) for the acceleration component of the load: Rough order of magnitude: 5%- 10% This corresponds to the torque value to be compensated.
rFact_AccTorque_RotMass	0.0%	Adaptation factor (referred to the rated load) for the acceleration component of the rotating masses Rough order of magnitude: 20% - 40% This corresponds to the torque value to be compensated.
rY1_Friction to rY6Friction	0.0%	Frictional component (referred to the rated load) of total torque Rough order of magnitude: 1% - 3% This corresponds to the torque value to be compensated.
rEfficiency_eta_lift	100%	Adaptation factor (referred to the rated load) for the difference in efficiency when hoisting
rEfficiency_eta_drop	100%	Adaptation factor (referred to the rated load) for the difference in efficiency when lowering
rLoadNormFactor	1.0	Adaptation factor for converting into a physical quantity The following formula is used for the conversion: Load[t] = measured value • rLoadNormFactor
iNumberOfCycles_MVS	10	Number of computation cycles, over which a sliding mean value generation (MVS) is realized for acceleration and torque

Table 8- 13 Important outputs of the DCC block for commissioning

Name	Value	Description
rTorque_MVS	%	Mean value, torque setpoint
rAccel_MVS	%	Mean value, acceleration
rJBrot	%	Acceleration component of the rotating masses
rJBLoad	%	Acceleration component of the load
rMm_minusM_Friction	%	Total torque - friction torque (M_{tot} - $M_{Friction}$)
rMm_minus_JBrot_Mf	%	M_M - M_{ARot} - $M_{Friction}$
rLoad_with_eta	%	Load torque with efficiency correction
rLoad_without_eta	%	Load torque without efficiency correction
rLoad_eta_plus_Frict	%	Total torque + friction torque (M_{tot} + $M_{Friction}$)
rLoad	-	Evaluated actual load value
boGrabTouchdown	-	Message "Grab touchdown"

Name	Value	Description
boLVM_Load_QL	-	Limit monitor: Load actual value < load limit value (rGrabWeight); important for the message "Grab touchdown"
boLVM_Speed_QU	-	Limit monitor: Velocity actual value > velocity limit value (rSpeedAverLimitValueMon); important for the message "Grab touchdown"
boLVM_Rope_QU	-	Limit monitor: Rope length > rope length limit value (rUnderWater); important for the message "Grab touchdown"

8.11.3.1 Commissioning support for CLM with FB_AutoSettingFW

Note

The use of the FB_AutoSettingFW block is optional. If this block is used, Chapters Friction torque compensation (Page 443) to Correcting the efficiency (Page 446) do not need to be taken into account.

Only when you use this block can the ropes be tightened early and the container spreader / grab fully mounted. Otherwise, observe the procedures Friction torque compensation (Page 443) to Correcting the efficiency (Page 446).

Note

- The rated velocity must not be exceeded during the commissioning with this block. The acceleration and deceleration should have the values with which the crane is operated later.
- Hoisting and lowering operations should always run without interruption at the rated velocity.
- The motors should be warmed up prior to the commissioning.

The FB_AutoSettingFW simplifies the commissioning by all important input parameters for the DCC_ContLoadMeasurement_1 being calculated and issued to the output at the end. The commissioning engineer then only needs to accept them at the inputs of the CLM block.

If the DCC_LoadDependFieldweak_1 block is also required on the crane, you do not need to repeat the measurements. This block determines automatically the appropriate inputs for both DCC blocks.

Two measuring steps must be performed during the commissioning. These two measuring steps should be executed together without a significant pause in between.

The following inputs have already been linked in the standard projects. If this block was added later, these links must be augmented in accordance with the following table.

Table 8- 14 Subsequent links

Name of the input	Logic operation
TO_Name	Link with the hoist drive - master axis (hoist or holding gear)
boReset	This input can be used to reset the measuring Step10 or Step20 or all values. • boAutoSettingFWStep10 = TRUE and boReset Only the measured values from Step10 are deleted. • boAutoSettingFWStep20 = TRUE and boReset Only the measured values from Step20 are deleted. • Keep boReset pressed for three seconds. All measured values are reset.
r64CurrentNominal	Enter rated current
r64CurrentSetpoint	Link with the current setpoint of the master
r64SmoothTimeCurrentSetpoint	Smoothing time for the current setpoint [ms] Change only if the current fluctuates excessively.
r64VelocityNominal	Enter rated velocity [mm/s]
r64RampUpTimeNominal	Ramp-up time to the rated velocity [ms]
r64UpLimitMeasRange	Upper measuring window limit [%] Change this value only if the measurement window does not suffice during the acceleration process or when the measuring range should be moved.
r64LowLimitMeasRange	Lower measuring window limit [%] Change this value only if the measurement window does not suffice during the acceleration process or when the measuring range should be moved.
r64MinLoad	Not required for DCC_ContLoadMeasurement_1, important only for DCC_LoadDependFieldWeak_1.
r64MaxLoad	Not required for DCC_ContLoadMeasurement_1, important only for DCC_LoadDependFieldWeak_1.

Measuring step 1: Measurement without load

Prerequisites:

- The ropes are tightened and the container spreader / grab is fully mounted.
- Limit switches, rated and maximum velocity should be set and all standardizations completed. The crane must be available for several test runs.
- Container spreader / grab must not accept any load and all measurements must be performed without load.
- boStartAutoSetting FW = TRUE → FB_AutoSettingFW is enabled
- boAutoSettingFWStep10 = TRUE → first measuring step is started.
- boAutoSettingFWStep20 = FALSE

Procedure:

Perform ten uninterruptible hoisting operations. The rated velocity must be reached for each hoist operation and travel constant for the time "r64MinTimeConstantDrive". "r64MinTimeConstantDrive" is available as output value.

A hoist operation is considered to have completed successfully when the "boSingleMeasFinished" output changes to TRUE. The hoist can then be lowered again in order to start a new hoist operation. If this output does not change to TRUE in time, the hoist operation was not valid. An intervention is not currently required and a new hoist operation can be started. After ten valid hoist operations, the "boStep10MeasFinished" output is set to TRUE.

If "boSingleMeasFinished" never changes to TRUE, the required sampling operations during the positive or negative acceleration or during the constant travel will not be reached. A check should be made whether constant travel is made long enough (at least "r64MinTimeConstantDrive") and possibly decrease or increase the measurement window "r64UpLimitMeasRange"/"r64LowLimitMeasRange" slightly.

If measuring step 1 must be repeated, you can use "boReset" to delete the measurement. Ensure, however, that "boAutoSettingFWStep10" is set to TRUE.

Note

If "boReset" is pressed for three seconds, all measuring steps will be reset irrespective of "boAutoSettingFWStep10/20".

Measuring step 2: Measurement with load

- The ropes are tightened and the container spreader / grab is fully mounted.
- Limit switches, rated and maximum velocity should be set and all standardizations completed. The crane must be available for several test runs.
- Container spreader / grab must now accept the rated load and all measurements must be performed with the rated load.
- boStartAutoSetting FW = TRUE → FB_AutoSettingFW is enabled.
- boAutoSettingFWStep10 = FALSE
- boAutoSettingFWStep20 = TRUE → the second measuring step is started.

Procedure:

Perform ten uninterruptible hoist operations again. The same conditions from measuring step 1 now also apply to measuring step 2. After ten valid hoist operations, "boStep20MeasFinished" is now set to TRUE.

If measuring step 2 must be repeated, you can use "boReset" to delete the measurement. Ensure, however, that "boAutoSettingFWStep20" is set to TRUE.

Note

If "boReset" is pressed for three seconds, all measuring steps will be reset irrespective of "boAutoSettingFWStep10/20 ".

You should avoid any unnecessary waiting times in the associated measuring steps. If the waiting time exceeds three minutes, the associated measuring step is deleted and must be repeated. This serves to ensure that the measurements are performed with a constant motor temperature.

Completion:

If "boStep10MeasFinished" and "boStep20MeasFinished" are both TRUE, the following outputs are not equal to 0. The values can now be accepted for CLM (and, if required, also for LDFW).

- r64AutoFrictionCurrent
- r64AutoFactAccCurrRotMa
- r64AutoFactAccCurrentLoad
- r64Eta_LDFW (only for the LDFW block)
- r64Eta_lift_CLM (only for the CLM block)
- r64Eta_drop_CLM (only for the CLM block)

If a value is present at the "r64MinLoad" and "r64MaxLoad" input, the following output values are also output automatically: They are not necessary for CLM. If LDFW is not used, the two "r64MinLoad" and "r64MaxLoad" inputs can be set to 0. The following outputs then also return 0:

- r64X1ParameterLoad
- r64Y1ParameterVelocity
- r64X2ParameterLoad
- r64Y2ParameterVelocity
- r64X3ParameterLoad
- r64Y3ParameterVelocity
- r64X4ParameterLoad
- r64Y4ParameterVelocity
- r64X5ParameterLoad
- r64Y5ParameterVelocity
- r64X6ParameterLoad
- r64Y6ParameterVelocity

If an error occurs, "boErrorFunctionBlock" output is TRUE and "i32ErrorIDFunctionBlock" sends the error ID. If "i32ErrorIDFunctionBlock" is not equal to 0 but "boErrorFunctionBlock" = FALSE, this is an alarm.

Note

After completion of all measurements, you can delete a specific measuring step. This must be activated beforehand (Step10 or Step20). "boReset" can then be used to delete the activated measuring step. This does not affect the other measuring step and the outputs are formed again taking the two measuring steps into account. This means both measuring steps do not need to be repeated in the event of an error. Do not press "boReset" longer than three seconds, otherwise the two measuring steps will be reset.

Note

Finally, a plausibility check should also be performed. Empirical values can be found at Table 8-10 Default assignments of setting connections prior to commissioning for gear stage 1 (Page 436).

8.11.3.2 Friction torque compensation

DCC_ContLoadMeasurement

$$\text{Load} \sim M_{Load} = \frac{M_{Motor} - \boxed{M_{Friction}} - k_{ARot} \cdot \frac{dn}{dt}}{\left(1 + k_{ALoad} \cdot \frac{dn}{dt}\right)}$$

Setting instructions:

Move the empty drum at constant base speed (n_{rated}) in the hoisting direction. Positive values must be passed accordingly to inputs "rTorqueNominal", "rSetpointTorqueHG" and "rSetpointTorqueCG". You can set the friction torque by means of the friction characteristic ("rY1_Friction_SGST1" to "rY10_Friction_SGST1"). The friction torque can be displayed as a percentage via "rSetpointTorquePercent".

DCC_ContLoadMeasurement_1

You can set the friction torque by means of the friction characteristic ("rY1_Friction" to "rY6_Friction"). The friction torque can be displayed as a percentage via "rSetpointTorquePercent".

Note

Always enter the friction torque value at rated speed in the friction characteristic for the commissioning phase. This makes the rest of the commissioning procedure easier. You can assign friction torques to the relevant speeds later on.

8.11.3.3 Compensating the acceleration torque of the rotating masses

$$\text{Load} \sim M_{\text{Load}} = \frac{M_{\text{Motor}} - M_{\text{Friction}} - \boxed{k_{\text{ARot}} \cdot \frac{dn}{dt}}}{\left(1 + k_{\text{ALoad}} \cdot \frac{dn}{dt}\right)}$$

Setting instructions:

Note

100% must be reached at output "rAccelerationMeanValue" for every measurement; otherwise you will need to make adjustments using inputs "rDifferentiationTimeCons" and "rSmoothTime_dn_dt".

Note

DCC_ContLoadMeasurement_1
After every acceleration attempt, rated speed 100% should be reached at output "rAccel_MVS", otherwise you must make adjustments using the inputs "rDifferentiationTimeCons" and "rSmoothTime_dn_dt".

- Accelerate the empty drum up to rated speed in the hoisting direction.
- On completion of the measurement, the value at the DCC block output "rMm_minus_JBrot_Mf" must be 0% (0% is the ideal scenario), as the total torque does not include a load acceleration component in this case.
- Try different values using parameter "rFact_AccTorque_RotMass1".

- Verify that you have identified the correct parameter setting by performing several acceleration trial runs.
- When doing so, remember the following formula:

$$M_{Motor} - M_{Friction} - k_{Arot} \bullet \frac{dn}{dt}$$

Note

At the beginning, the factor at the input of block "rFact_AccTorque_RotMass1" or "rFact_AccTorque_RotMass" for DCC_ContLoadMeasurement_1 is still 0. Enter the value of output "rMm_minus_JBrot_Mf" as the start value. Then adjust the factor "rFact_AccTorque_RotMass1" or "rFact_AccTorque_RotMass" for the DCC_ContLoadMeasurement_1 by iteration until the output "rMm_minus_JBrot_Mf" shows a value of 0.

NOTICE
After you have made this setting, you can attach the ropes.

8.11.3.4 Compensating the acceleration torque of the load

$$\text{Load} \sim M_{Load} = \frac{M_{Motor} - M_{Friction} - k_{ARot} \bullet \frac{dn}{dt}}{\boxed{\left(1 + k_{ALoad} \bullet \frac{dn}{dt}\right)}}$$

Setting instructions

- Set the parameters "rEfficiency_eta_drop" and "rEfficiency_eta_lift" to 100%.
- Approximately the rated load = maximum load should be suspended on the ropes.
- Accelerate up to rated speed in the hoisting direction.
- On completion of the measurement, parameter "rLoad_eta_plus_Friction" should match as closely as possible the value that the drive draws (rSetpointTorquePercent) after the acceleration phase (only in the hoisting direction) when the load is in a steady-state.
- Try various values with the "rFact_AccTorque_Load1" or "rFact_AccTorque_Load" parameter for the DCC_ContLoadMeasurement_1.
- Verify that you have identified the correct parameter setting by performing several acceleration trial runs.

8.11.3.5 Correcting the efficiency

The efficiency correction should be used to ensure that with the same load, the velocity when lowering is identical to the velocity when hoisting (raising). To then obtain the same load actual value for hoisting and lowering at different current values, you need to set the efficiency correction factor for hoisting (rEfficiency_eta_lift) and lowering (rEfficiency_eta_drop) as follows:

$$\text{Load}_{\text{average}} = \frac{\text{Load}_{\text{lifting}} + \text{Load}_{\text{dropping}}}{2}$$

$$\text{rEfficiency\_eta\_lift} = \frac{\text{Load}_{\text{average}}}{\text{Load}_{\text{lifting}}} \bullet 100\ \%$$

$$\text{rEfficiency\_eta\_drop} = \frac{\text{Load}_{\text{average}}}{\text{Load}_{\text{dropping}}} \bullet 100\ \%$$

As a result, the efficiency correction value will always be ≤ 100% for hoisting and ≥ 100 % for lowering. This will be illustrated in the following example calculation:

Example calculation

Load value shown for hoisting	250
Load value shown for lowering	150
Calculated average value for load	200

$$\text{rEfficiency\_eta\_lift} = \frac{200}{250} \bullet 100\ \% = 80\ \%$$

$$\text{rEfficiency\_eta\_drop} = \frac{200}{150} \bullet 100\ \% = 133{,}33\ \%$$

The efficiency correction is set with "rEfficiency_eta_drop" and "rEfficiency_eta_lift". The connection "rLoadCurrent_with_eta" should neither change when hoisting nor when lowering.

The measurements can be taken at no-load and the correction set. The values should then be checked with load and fine-tuned if necessary.

Setting instructions:

- Move the rated load in hoisting direction and then in lowering direction at constant rated speed (n_{rated} motor), provided
  rEfficiency_eta = 100% (default)
- Repeat your measurement several times. Note down the mean value of "$\text{rLoadCurrent\_with\_eta}_{Hoisting}$" and "$\text{rLoadCurrent\_with\_eta}_{Lowering}$".
- Calculate the rEfficiency_eta corresponding to the specified example.

Note

Connect a positive edge at input "boResetLoadCurrent" after every measurement.

- Check the settings by hoisting and lowering the rated load (up to the rated speed) several times. In so doing "rLoadCurrent_with_eta" and "rLoadWithNormFactor" should no longer change when hoisting and lowering.
- Adapt the factor rEfficiency_eta (approx. 115 %) to keep the change between hoisting and lowering as low as possible.

Note

When lowering, if a load higher than the actual load is determined, a higher value of rLoadCurrent_with_eta while lowering with respect to hoisting ensures that the drive is not overloaded when braking the load.

This setting is only necessary for the case that the load should be determined while lowering (e.g. when starting with lowering). For normal applications, it is sufficient if the determined load is only re-calculated again if the load has changed.

8.11.3.6 Compensating the rope weight

Using "rActualRopeLength" (rope length) combined with "rCompensationRopeFactor", you can compensate the rope load of the uncoiled rope in order to calculate the difference in weight between the coiled and uncoiled rope from the load. The current rope length can be determined in numerous different ways, one of which is shown below by way of an example:

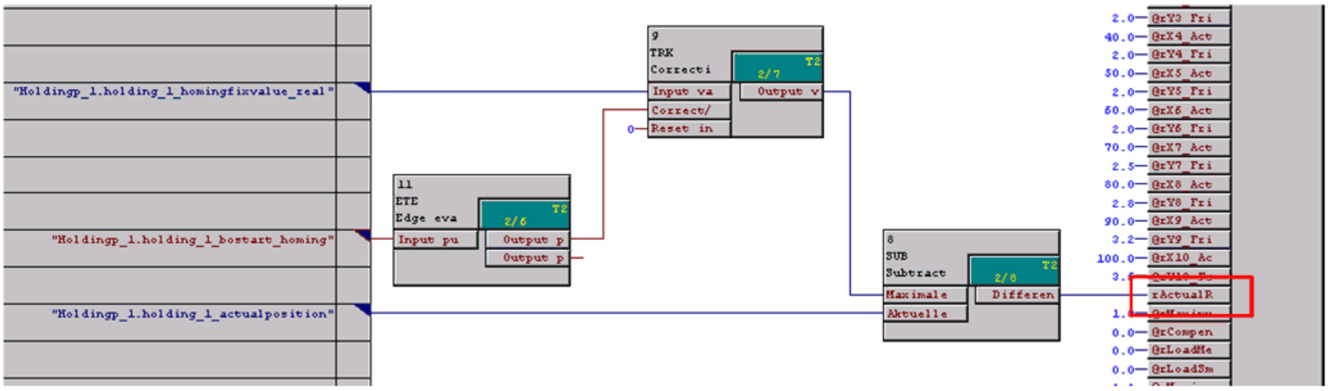


Figure 8-43 Determining the current rope length

The homing signal is routed to a memory block (TRK) by means of an edge evaluation (ETE). As a rule, the grab should be homed to the highest position. The top position and, thus, the maximum position value are saved in the memory block together with the homing signal. When the grab is lowered, the position decreases, since lowering equates to negative velocity. The current position is now subtracted from the maximum position (SUB) and the result is the current rope length.

Note

The description above is simply an example; it is not the only method of determining the current rope length.

8.11.3.7 Load from load measuring cell

You can input a load directly by means of "rloadMeasurementCell". In this case, you must set "boSwitchToLoadMeasCell" to True. The load value from the measuring cell will then be applied instead of the calcualted load value.

8.11.3.8 Calculating the physical size of the load

Using input parameter "rLoadNormFactor" combined with outputs "rLoad_with_eta_rope" and "rLoad", you can convert the calculated load to a physical quantity (e.g. tonnes).

Calculation example:

When hoisting a rated load of 45 t, "rLoad_with_eta_rope" and "rLoad" indicate a value of 85 %.

$$\text{rLoadNormFactor} = \frac{45\ \text{t}}{85\ \%} = 0{,}5294\ \frac{\text{t}}{\%}$$

Output "rLoad" should now show 45t and "rLoad_with_eta_rope" 85 %.

8.11.3.9 Output "Grab touchdown" (boGrabTouchDown = True)

The "boGrabTouchDown" output will change to true if the inputs of the DCC block fulfill the following conditions:

- "rActualRopeLength" is greater than the value entered in "rRopeAverLimitValueMon" AND
- "rLoad" is less than the value entered in "rLoadAverLimitValueMon" AND
- The value of "rActualVelocity" is greater than the value entered in "rSpeedAverLimitValueMon" AND
- the "rOnDelayTimeLimitMonitor" timer has expired AND
- "rActualVelocity" is less than zero (lowering).

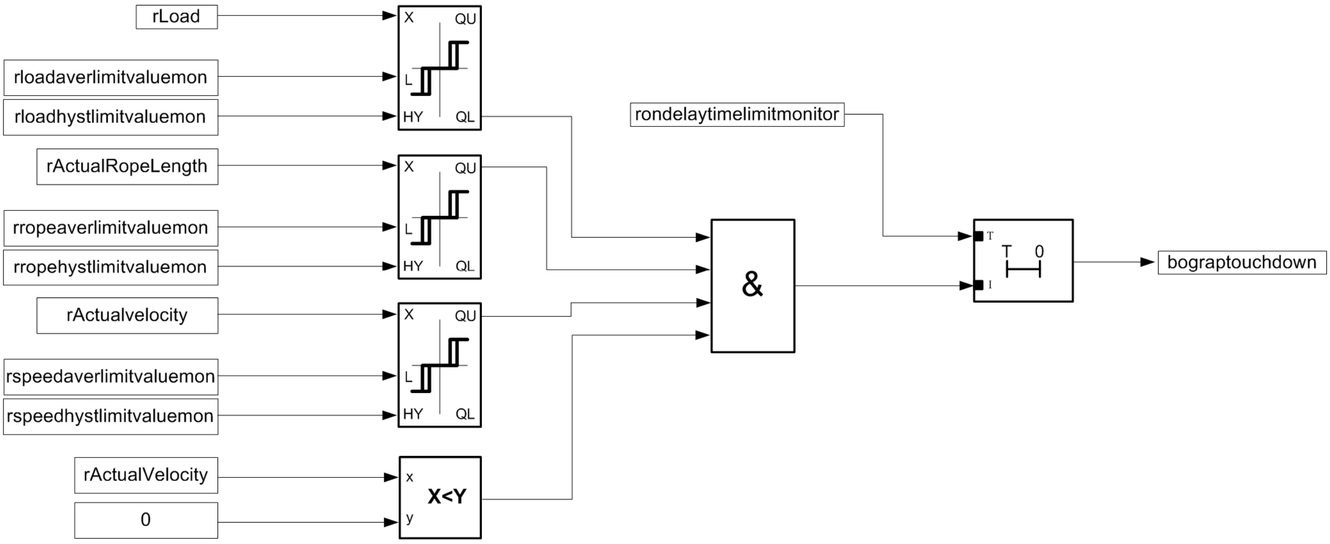


Figure 8-44 Overview of "Grab touchdown"

DCC_ContLoadMeasurement_1

The "boGrabTouchDown" output will change to TRUE if the inputs of the DCC block fulfill the following conditions:

- "rActualRopeLength" is greater than the value entered in "rUnderWater" AND
- "rLoad" is less than the value entered in "rGrabWeight" AND
- the absolute value of "rActualVelocity" is greater than the value entered in "rSpeedAverLimitValueMon" AND
- the "rOnDelayTimeLimitMonitor" timer has expired AND
- "rActualVelocity" is less than zero (lowering).

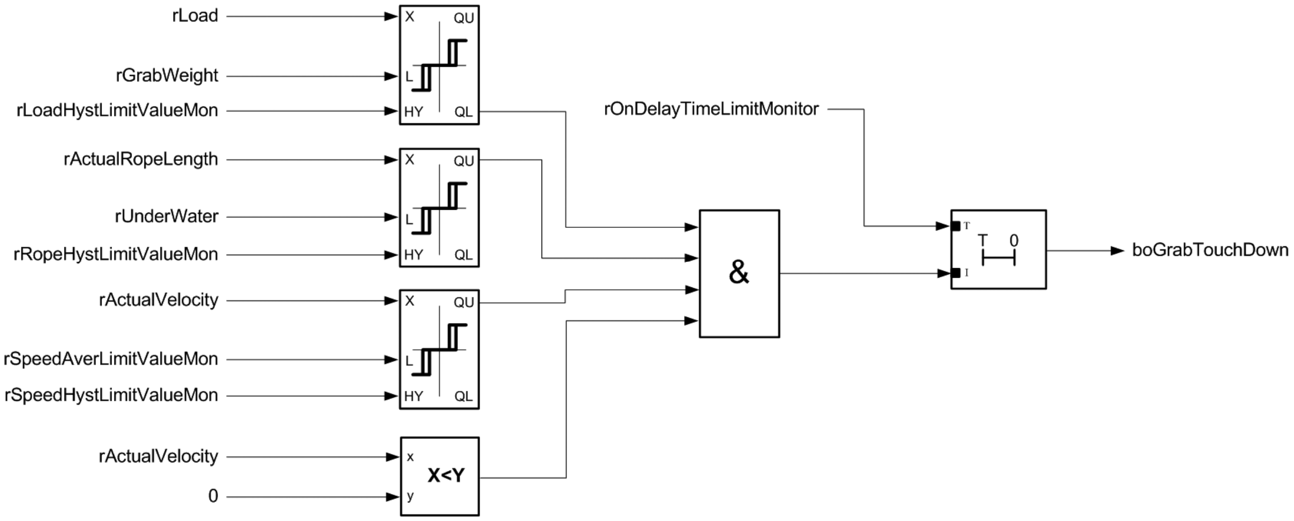


Figure 8-45 Overview "Grab touchdown" for DCC_ContLoadMeasurement_1

8.12 Start pulse

8.12.1 General description

In DCC block StartPulse the I component of the speed controller is stored when the brakes are closed, and set again when they are opened. Chapter DCC_StartPulse (Page 150), "2. Methods" paragraph contains a detailed description.

8.12.2 Commissioning

The block DCC_StartPulse can also be tested during commissioning in just a few steps without saving the I component, whereby the relevant value is sent by the S7.

The setpoint is sent to SIMOTION in PZD 7 "Setpoint for start pulse_S7" (see SIMATIC S7 → SIMOTION (Page 307)). This value can be connected to the input "rTorqueValueRetain" of the DCC block StartPulse in the DCC chart under the name "..._startpulse_receive". It is, however, advisable to connect a numerical selector switch in front of this input. This enables the user to toggle between the saved I component and the setpoint from the S7.

When commissioning of the block is complete, the numerical selector switch should be cleared again and the input "rTorqueValueRetain" connected to the variable "…_i_component_for_dcc".

Commissioning the integration component of the speed controller

The I/O variables "DO-name_setpoint_torque" and "DO-name_i_component" are needed to be able to send the I component of the speed controller (SINAMICS parameter r1482) to SIMOTION. The torque value is made available via function block FB_TelegramSinamicsToSimotion (r64TorqueSetpointSpeedController) (refer to Chapter FB_TelegramSinamicsToSimotion (Page 164)). It is interconnected with variable "rTorqueValueRetain". When the "Close brake" command is issued (must come from S7; AppSTW2 bit 0 – state 1→0), the torque value of variable "rTorqueValueRetain" is stored in the non-volatile memory and is immediately available at output "rSetTorqueValue" (if boSinamicsComReady = True and boCommandOpenBrake = False).

When the "Open brake" command is issued (must also come from S7; AppSTW2 bit 0 – state 0→1), bit "boSetTorque" is set and the saved I component is transferred to the speed controller. For this purpose, the variable "rSetTorqueValue" and the bit "boSetTorque" are interconnected with the corresponding variables (r64TorqueSetpoint and b16AdditionalSTW) of block FB_TelegramSimotionToSinamics. This means that the torque that was saved when the brake was closed is present when starting to move, which prevents the load from sagging. Please ensure that the particular nominal torque of the drive is interconnected to"r64standfactortorquesetpoint" from FB_TelegrammSimotionToSinamics and to "r64standfactortorquesinamics" for FB_TelegrammSinamicsToSimotion.

To enable this process to work correctly, a TRUE signal must be present at the following inputs:

- boEnableStartpulse
- boEnableBrakeStoreValue
- boSinamicsComReady

A FALSE signal at "boSinamicsComReady" resets the output signal "boSetTorque" if the pulse time has not yet elapsed. If the brakes issue the checkback signals "Brake open" and "Brake closed", the inputs "boCheckbackBrakeOpened" and "boCheckbackBrakeClosed" can (X? Note Chen: "The two variables are no longer present for Block_1. This page should be checked and revised. We now have a "boCommandOpenBrake" input, please use it) be provided at the start pulse (only DCC_StartPulse, not DCC_Startpulse_1). If the command "Close brake" is issued, followed by the checkback signal "Brake closed", or if "Open brake" is present, followed by "Brake open", this will also cause the output signal "boSetTorque" to be reset.

--- The following description is valid only for DCC_StartPulse not DCC_Startpulse_1 ---

The command "close brake" can also be derived from the command "open brake" (negated), if this does not come from the S7. In this particular case, a time delay is required for the "close brake" command. A PDE DCC block can be used for this purpose. The call cycle (T2 ≙ IPO cycle, e.g. 9 ms) should be selected as minimum delay time.

If checkback signals are used, they must actually come from the brakes, and not be generated indirectly via the control commands "Open brake" and "Close brake", as in this case the torque is only present during the period between setting the command "Open brake" and when the checkback signal "Brake opened" is received. This prevents the load from sagging while the brake is opening.

If no checkback signals are present, the two inputs "boCheckbackBrakeOpened" and "boCheckbackBrakeClosed" must both be set to FALSE. This causes the output signal "boSetTorque" to be reset either when a new close command is issued or, at the latest, when the pulse time "rPulseTime" has elapsed.

⚠WARNING

Incorrect start pulse values can lead to injury as well as material damage.

8.13 Temporary travel at overspeed

CAUTION

Basically, the maximum speed should not be increased by any more than 10 percent; this is because increasing the speed puts mechanical and electrical components under more strain and may result in the system sustaining damage.

During commissioning it may be necessary to run the drive at a speed that is faster than the maximum motor speed in order to check mechanical limits and, if required, to adjust them or to test with the overspeed switch.

There are two situations in which the maximum speed may need to be increased:

1. The maximum motor speed (p0322) is preset to 0.0.

   No further consideration has to be given to the maximum motor speed (p0322) in this case. The maximum speed is increased in parameter p1082, with the value rising by 10 percent. Since the S7 defines the speed setpoint as a percentage value referred to the reference speed (p2000), this needs to be increased by 10 percent too.
2. The maximum motor speed (p0322) is preset to a value greater than 0.0.
   The maximum motor speed (p0322) and the maximum speed (p1082) must be increased by 10 percent. Please note that parameter p1082 cannot be greater than p0322. The reference speed (p2000) has to be increased by 10 percent too.

Both cases do not require any changes to be made to the technology objects, provided that they have previously been set to the correct reference values for the system. When the maximum motor speed (p0322) is changed, alarm "Incorrect friction characteristic" may appear. Make sure that the value in p0322 is greater than or equal to the friction characteristic p3820 to p3829.

If travel with overspeed is not to be used any longer, parameters p0322, p1082, and p2000 must be reset to the correct values.

8.14 Grab Ship Unloader (GSU)

8.14.1 General description

In order to commission a GSU crane correctly and quickly, please take note of the following points. We will also refer to grab dredgers in this context; these devices are identical to GSU cranes in terms of their gripper functionality, but they do have certain special features from a technical programming point of view.

8.14.2 Procedure

The procedure for commissioning a grab crane (grab ship unloader or GSU) is described in bullet points and applies equally to all types of commissioning.

Differences between GSU cranes and grab dredgers are highlighted.

8.14.2.1 Checking the hardware

- What infeeds, drives, and converters have been installed?
- What other hardware is connected (SMC, TM, hub, etc.)?

8.14.2.2 Checking the configuration in the project

- Are the connected devices also available in the project in exactly the same format?
- Do the topologies match?
- Have the axes and gears been set up accordingly?
- Are the MCC/DCC programs available for the relevant drives? Are the libraries available?
- Can the project be translated and compiled without errors?
- If a drive does not have an encoder, delete the associated axis and recreate it as a drive axis with the same name (when creating the axis, uncheck the box next to "Positioning").

8.14.2.3 Running a script for each drive

- A script folder containing a script should be located under "Drives"; see General (Page 361), "Script files" section.
- If no script folder is present, you can create it as follows: Right-click on Drive → Expert → Insert script folder.
- Import the script for the corresponding drive from a standard project into the script folder and run it in order to establish the interconnection required for communication.

8.14.2.4 Adjusting the direction of rotation

Depending on how the motors and encoders of the holding gear and the closing gear are arranged, you may need to invert both the direction of rotation and the encoder evaluation of a drive.

Possible arrangement 1 – Both drives are arranged identically.

In this case, the direction of one of the two drives must be reversed; this ensures that, when working with the same setpoint, both drives wind the ropes onto or off the drum simultaneously. A positive direction means the ropes must be wound onto the drum: Reverse the direction on the drive where this is not the case.

- Set p0010 to "Encoder commissioning (4)" (drive commissioning parameter filter).
- Set p0410[0].0 and p0410[0].1 to "Yes" (encoder inversion actual value)
- Set p0010 to "Motor commissioning (3)".
- Set p1820[0] to "On" (direction reversal output phases).
- Reset p0010 to "Ready (0)".

Note

Instead of setting parameter p1820[0] to "On", you can invert the electric power cable at the motor and leave p1820[0] set to "Off". The direction can also be reversed via the reciprocal swapping of two phases. For reasons of safety, however, it is more advisable to reverse the direction by means of the electric power cable.

Similarly, the direction should preferably be reversed on the encoder via the hardware and not via the software.

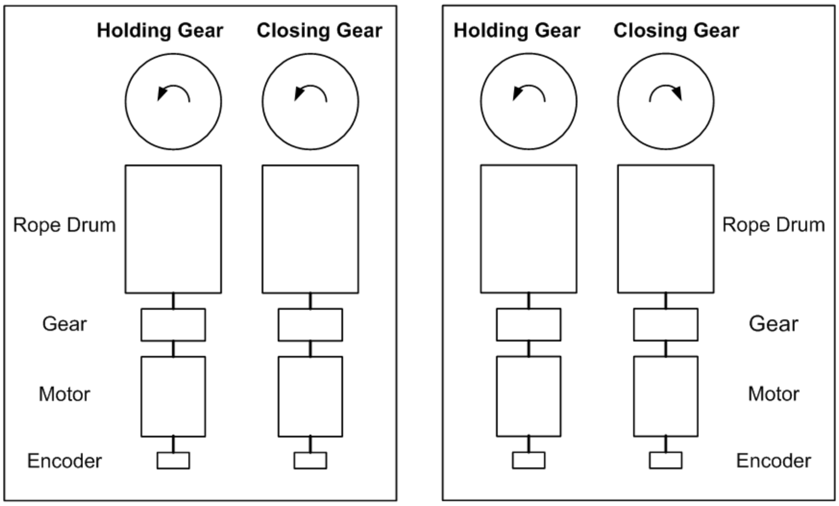


Figure 8-46 Left with no correction (incorrect), right with correction (correct)

Possible arrangement 2 – Both drives are arranged in opposition to one another.

The mechanical arrangement means that there is no need to perform any adjustment via software. When working with the same setpoint, the drives would rotate simultaneously in such a away that the ropes would either be wound onto or off the drum.

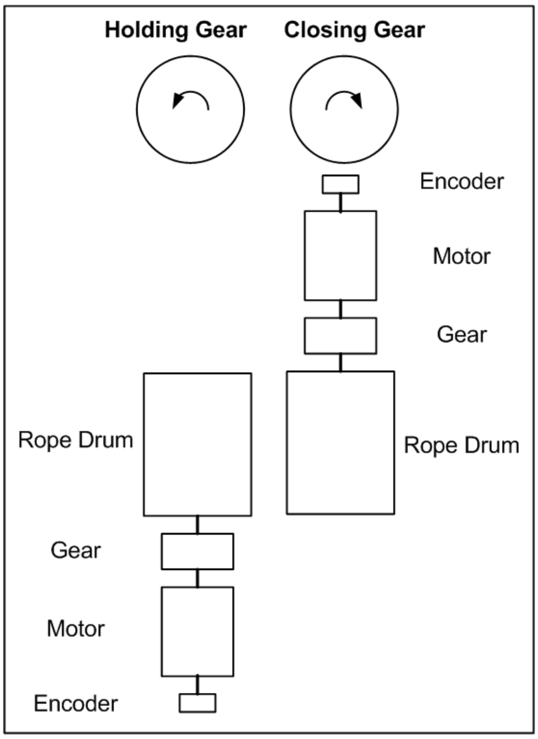


Figure 8-47 No correction necessary; directions of rotation are correct

8.14.2.5 Stationary measurement and rotating measurement

Perform a stationary and a rotating measurement for all drives. For the rotating measurement the drive shaft must not be coupled and the motor must be able to move freely. Parameter p0856 must be set to 1 for the rotating measurement. Once the rotating measurement has been taken, reset parameter p0856 to its original value or run the script again.

Note

In the case of data sets (DDS/MDS) which have already been set up, these measurements must be repeated.

When taking the measurements, pay attention to the data set which is currently active (r0051).

For further information on stationary and rotating measurements, see the section titled SINAMICS drive object (Page 400).

8.14.2.6 Setting the reference and maximum parameters

- In SINAMICS, set parameters p2000 (reference value) and p1082 (maximum value) under the relevant drive. In SIMOTION, set TypeOfAxis → SetpointDriverInfo → Drivedata → maxspeed and nominalspeed under the relevant axis to the same value as that set in SINAMICS p2000. Check TypeOfAxis → MaxVelocity → maximum (unit here should be [mm/s], not [rpm]).
- Parameters p2001 and p2003 must be scaled too. If, for example, parameter p2001 = 500 V during transmission of the current voltage, SINAMICS sends the value in a data word, referred to 500 V. So, 500 V = 100% = 4000 hex = 16384. If the current voltage is 250 V, for example, a data word is sent with the value 8192.
- It is recommended that parameter p2002 is set to the maximum value from p0640.

Note

You can also set the reference parameters using a script, see General (Page 361), "Script files" section.

8.14.2.7 Reference parameters in the interface of the MCC unit

The reference values must be entered in every MCC source. These include:

- X_nominalVelocity (reference speed, converted into [mm/s] or [rpm])
- X_nominalVoltage (reference voltage; see parameter in drive p2001)
- X_nominalCurrent (reference current; see parameter in drive p2002)
- X_nominalTorque (reference torque; see parameter in drive p2003)
- X_nominalPower (reference power; see parameter in drive r2004)
- (X = Holdingp_1, Closingp_1, Trollp_1, etc.)

Note

You can also set the reference parameters using a script, see General (Page 361), "Script files" section.

If the drive has an encoder, the reference speed must be converted into [mm/s]. The formula for this is as follows:

$$v\,[{}^{mm}/_{s}] = \frac{\text{number of load revolutions } [{}^{1}/_{min}]}{\text{number of motor revolutions } [{}^{1}/_{min}]} \cdot \frac{\text{rated speed}}{60}[{}^{1}/_{s}] \cdot \text{dis. per spindle revolution } [{}^{mm}/_{rot}]$$

For encoderless drives, which are often trolley drives on grab dredgers and small GSU cranes, the reference speed must be converted into rpm. The formula for this is as follows:

$$v\,[{}^{1}/_{min}] = \frac{\text{number of load revolutions } [{}^{1}/_{min}]}{\text{number of motor revolutions } [{}^{1}/_{min}]} \cdot \text{rated speed } [{}^{1}/_{min}]$$

The following three variables can be found in the axes under "Configuration data".

- Number of load revolutions:
  TypeOfAxis.NumberOfDataSets.DataSet_1.Gear.denFactor
- Number of motor revolutions:
  TypeOfAxis.NumberOfDataSets.DataSet_1.Gear.numFactor
- Leadscrew pitch:
  LeadScrew.pitchVal

You can find the reference speed in the drive under p0311[0].

8.14.2.8 Setting the closing velocity

The closing velocity of the closing gear is controlled using the "r64KP_ChangeOverPoint" variable. The closing velocity generally controls the maximum permissible velocity setpoint until the end position is approached (grab open or grab closed). The velocity setpoint can be limited at any time by deflecting the master switch. Shortly before the end position is reached, the "r64KP_ChangeOverPoint" variable accepts the velocity setpoint and approaches the end position.

"r64KP_ChangeOverPoint" is required for the P element. The root function keeps the setpoint lower than with a P element in the event of significant positional differences. When the end position is reached and the positional difference is, therefore, minor, the route function is replaced by the P element, since its characteristic gets very steep towards the end. Here, the P element becomes active and ensures a linear path.

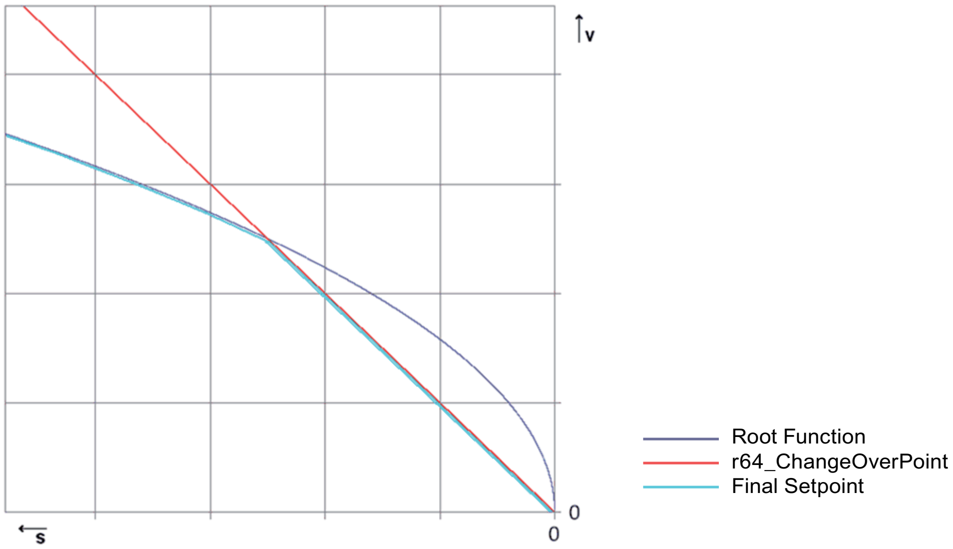


Figure 8-48 Theoretical characteristic of K_P variables and results

Note

Experience has shown that the value 0.5 (default value for "r64KP_ChangeOverPoint") works very well in practice.

The gain factor defines the length of the constant velocity phase and the start of the deceleration phase. The specified ramp-up time is not changed, but only the ramp-down time. As a consequence, the total closing time can be influenced.

NOTICE
The slope of the deceleration ramp determined by the K_P variable must be set so that the predefined ramp-down time (set by the higher-level controller, e.g. S7) is not undershot. This is to stop the drive overshooting the target.

If "r64KP_ChangeOverPoint" is too small, the drive will go straight into the deceleration phase during ramp-up without having reached the constant velocity phase because the K_P factor is too weak and the drive must start to brake earlier.

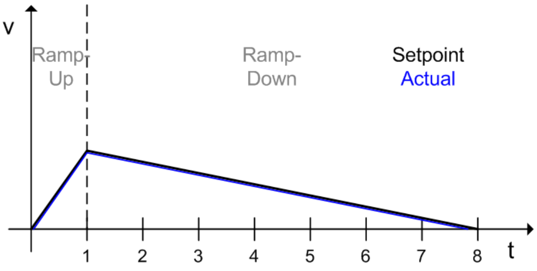


Figure 8-49 K_P factor too weak → long closing time

A closing time close to requirements with reserves can achieved by increasing "r64KP_ChangeOverPoint". The ramp-down time is somewhat longer than the ramp-up time. The closing time could be reduced further. The purpose of a reserve is to allow the predefined ramp-down time (set by the higher-level controller) to be increased without having to adapt the K_P factor.

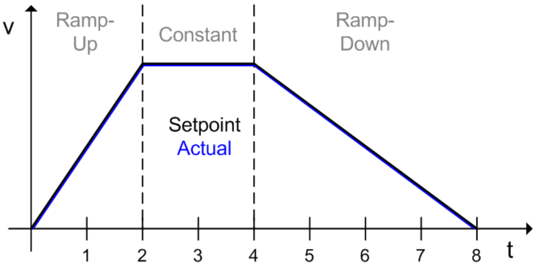


Figure 8-50 Increased K_P factor with reserves → good closing time

If the K_P factor is optimized in terms of time, the deceleration ramp will match the predefined deceleration ramp. In this context, it is important that the predefined deceleration is not undershot. This is to stop the drive overshooting the target. The predefined ramp-down time cannot be increased further without adapting the K_P factor accordingly.

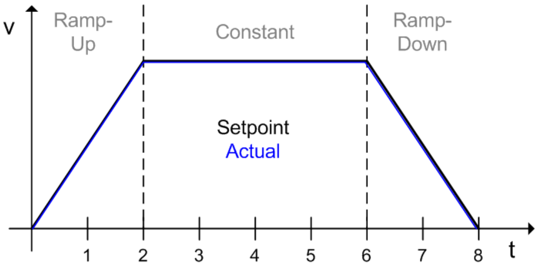


Figure 8-51 Time-optimized K_P setting

Setting a K_P factor which is too high shortens the ramp-down time and makes the ramp steeper than would be the case with the predefined ramp-down time. The drive will overshoot the target in this case.

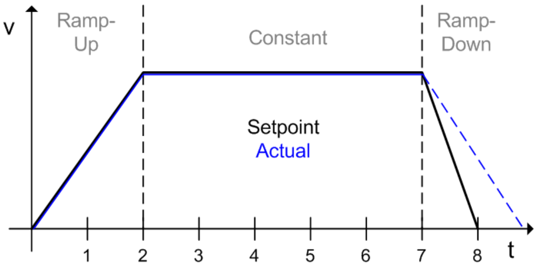


Figure 8-52 K_P factor too high → target overshoot

Note

Ensure that the grab does not close suddenly but steadily; if this is not the case, the K_P factor is too large.

8.14.2.9 Deactivating control blocks in the DCC for the time being

At the beginning, deactivate the DCC_SlackRopeControl (AppStw2 Bit 4) and DCC_CurrentEqualControl (AppStw2 Bit 8) and DCC_Startpulse (AppStw1 Bit 2) blocks. These blocks must be activated again at a later time; as long as they are not adjusted correctly, they can cause faults. Only if the holding gear and closing gear operate together at the same time: Selectively switch-in the DCC blocks.

8.14.2.10 Checking the interface between the S7 and SIMOTION

Move the drives via the S7 and send a couple of signals to the SIMOTION. You can check the signals which have been received and sent in the I/O area (symbol browser) in SIMOTION.

8.14.2.11 Settings for load-dependent field weakening and continuous load measurement

If the two DCC blocks DCC_LoadDependingFieldWeak and DCC_ContLoadMeasurement are used, some values such as the frictional torque and acceleration torque of the rotating loads must be determined before the ropes are attached (see the sections titled Field weakening (LDFW) (Page 414) and Continuous load measurement (CLM) (Page 434)). As a rule, only DCC_LoadDependingFieldWeak is used for standard GSU cranes. Grab dredgers are not usually operated with the load-dependent field weakening function active. However, a continuous load measurement is performed so that the dredger operator knows whether the grab is full and when the grab has touched down. Only once the relevant values have been determined can the ropes be attached.

Re continuous load measurement:
At the start of the measurements, set "rLoadNormFactor" to 1.0 and "rCompensationRopeFactor" to 0.0 (default values).

The system-specific settings for these two inputs should not be made until the end, because making them at the beginning would distort the results; see the section titled Continuous load measurement (CLM) (Page 434) for more information.

8.14.2.12 Control word circuit

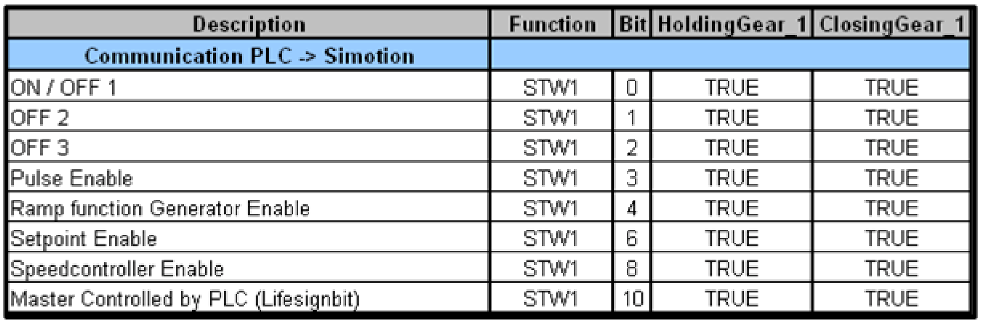
Note

Definition of the "Grab open" and "Grab closed" positions:

- Grab open: The positional difference between holding gear and closing gear is ZERO.
- Grab closed: The positional difference between holding gear and closing gear is a positive value.

Hoisting or lowering the holding gear and closing gear with an open grab

The tables below show how the control words are set for hoisting and lowering with the grab open. Hoisting equates to a positive setpoint and lowering to a negative one. As hoisting and lowering involves movement of both the holding gear and the closing gear, all enables for both drives must be available in control word 1.

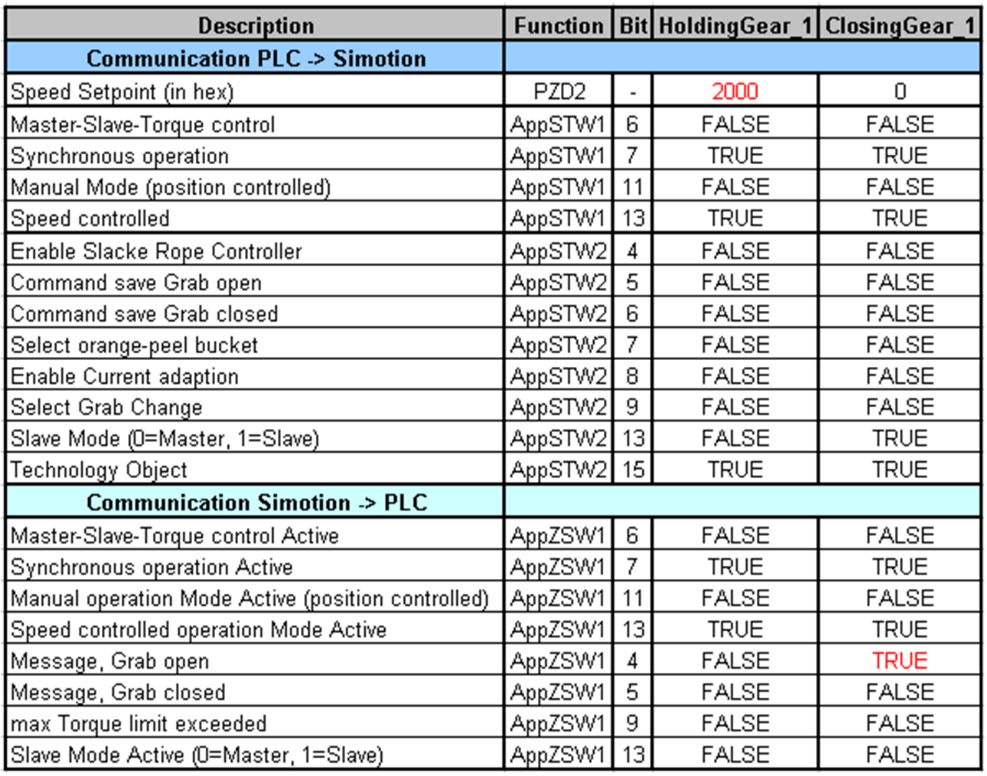


Description	Function	Bit	HoldingGear_1	ClosingGear_1
Communication PLC -> Simotion				
ON / OFF 1	STW1	0	TRUE	TRUE
OFF 2	STW1	1	TRUE	TRUE
OFF 3	STW1	2	TRUE	TRUE
Pulse Enable	STW1	3	TRUE	TRUE
Ramp function Generator Enable	STW1	4	TRUE	TRUE
Setpoint Enable	STW1	6	TRUE	TRUE
Speedcontroller Enable	STW1	8	TRUE	TRUE
Master Controlled by PLC (Lifesignbit)	STW1	10	TRUE	TRUE

Figure 8-53 Control word 1 for hoisting and lowering with an open grab

Contrary to control word 1, the setpoint input for hoisting and lowering in PZD2 is different. To hoist the open grab, **only** the holding gear needs to move in the positive direction. The closing gear uses the velocity setpoint of the holding gear for the purpose of moving. This means that a setpoint does **not** have to be entered for the closing gear.

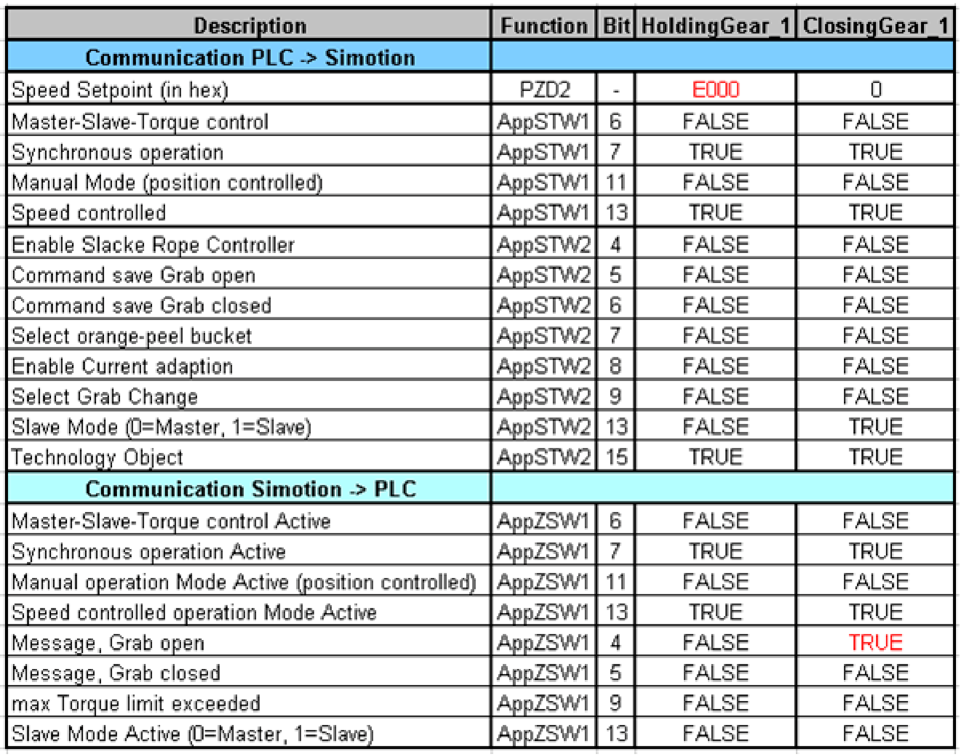
The "Grab open" signal is only fed back in application status word 1, bit 4 of the closing gear.



Description	Function	Bit	HoldingGear_1	ClosingGear_1
Communication PLC -> Simotion				
Speed Setpoint (in hex)	PZD2	-	2000	0
Master-Slave-Torque control	AppSTW1	6	FALSE	FALSE
Synchronous operation	AppSTW1	7	TRUE	TRUE
Manual Mode (position controlled)	AppSTW1	11	FALSE	FALSE
Speed controlled	AppSTW1	13	TRUE	TRUE
Enable Slacke Rope Controller	AppSTW2	4	FALSE	FALSE
Command save Grab open	AppSTW2	5	FALSE	FALSE
Command save Grab closed	AppSTW2	6	FALSE	FALSE
Select orange-peel bucket	AppSTW2	7	FALSE	FALSE
Enable Current adaption	AppSTW2	8	FALSE	FALSE
Select Grab Change	AppSTW2	9	FALSE	FALSE
Slave Mode (0=Master, 1=Slave)	AppSTW2	13	FALSE	TRUE
Technology Object	AppSTW2	15	TRUE	TRUE
Communication Simotion -> PLC				
Master-Slave-Torque control Active	AppZSW1	6	FALSE	FALSE
Synchronous operation Active	AppZSW1	7	TRUE	TRUE
Manual operation Mode Active (position controlled)	AppZSW1	11	FALSE	FALSE
Speed controlled operation Mode Active	AppZSW1	13	TRUE	TRUE
Message, Grab open	AppZSW1	4	FALSE	TRUE
Message, Grab closed	AppZSW1	5	FALSE	FALSE
max Torque limit exceeded	AppZSW1	9	FALSE	FALSE
Slave Mode Active (0=Master, 1=Slave)	AppZSW1	13	FALSE	FALSE

Figure 8-54 Activation of hoisting with an open grab

To lower the grab, the setting of control word 1 remains the same. A negative setpoint for the holding gear must be entered in PZD2. Thus, hoisting and lowering with the grab open only differ in terms of the setpoint input for the holding gear.

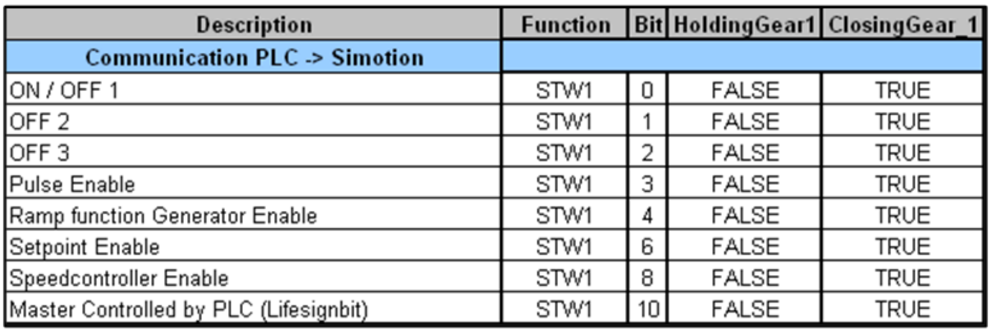


Description	Function	Bit	HoldingGear_1	ClosingGear_1
Communication PLC -> Simotion				
Speed Setpoint (in hex)	PZD2	-	E000	0
Master-Slave-Torque control	AppSTW1	6	FALSE	FALSE
Synchronous operation	AppSTW1	7	TRUE	TRUE
Manual Mode (position controlled)	AppSTW1	11	FALSE	FALSE
Speed controlled	AppSTW1	13	TRUE	TRUE
Enable Slacke Rope Controller	AppSTW2	4	FALSE	FALSE
Command save Grab open	AppSTW2	5	FALSE	FALSE
Command save Grab closed	AppSTW2	6	FALSE	FALSE
Select orange-peel bucket	AppSTW2	7	FALSE	FALSE
Enable Current adaption	AppSTW2	8	FALSE	FALSE
Select Grab Change	AppSTW2	9	FALSE	FALSE
Slave Mode (0=Master, 1=Slave)	AppSTW2	13	FALSE	TRUE
Technology Object	AppSTW2	15	TRUE	TRUE
Communication Simotion -> PLC				
Master-Slave-Torque control Active	AppZSW1	6	FALSE	FALSE
Synchronous operation Active	AppZSW1	7	TRUE	TRUE
Manual operation Mode Active (position controlled)	AppZSW1	11	FALSE	FALSE
Speed controlled operation Mode Active	AppZSW1	13	TRUE	TRUE
Message, Grab open	AppZSW1	4	FALSE	TRUE
Message, Grab closed	AppZSW1	5	FALSE	FALSE
max Torque limit exceeded	AppZSW1	9	FALSE	FALSE
Slave Mode Active (0=Master, 1=Slave)	AppZSW1	13	FALSE	FALSE

Figure 8-55 Activation of lowering with an open grab

Opening and closing the closing gear (after grab adjustment)

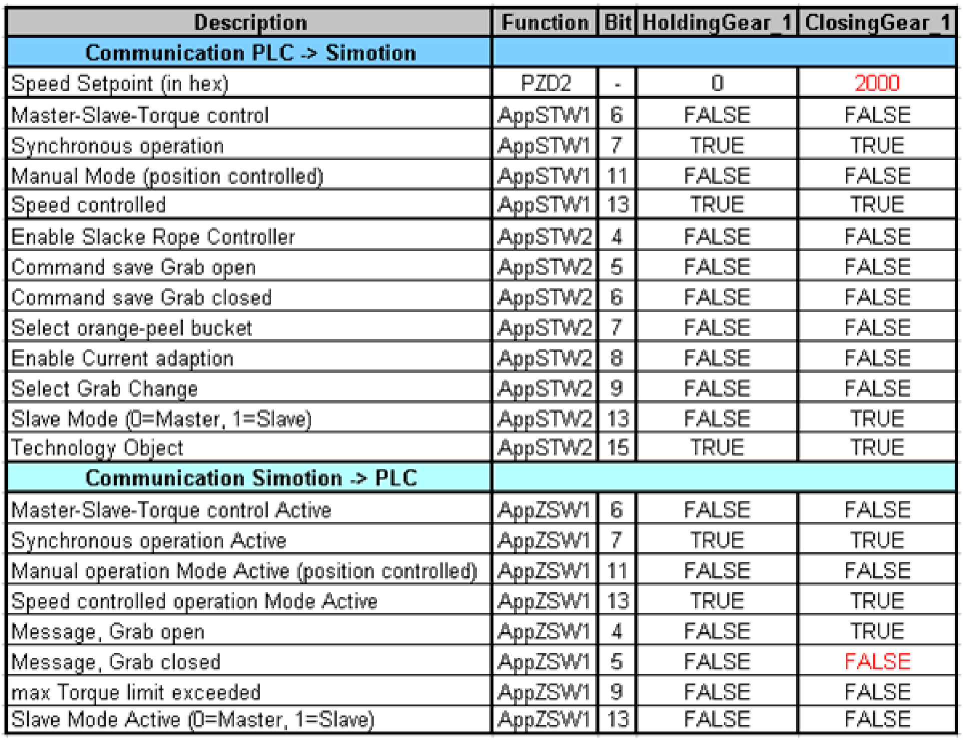
The tables below show how the control words are set for opening and closing the grab in the air. Closing equates to a positive setpoint and opening to a negative one. Opening and closing involves only the closing gear. As a result, all enable signals for the holding gear can be deselected in control word 1.
See also Chapter Correct grab adjustment (Page 469).



Description	Function	Bit	HoldingGear1	ClosingGear_1
Communication PLC -> Simotion				
ON / OFF 1	STW1	0	FALSE	TRUE
OFF 2	STW1	1	FALSE	TRUE
OFF 3	STW1	2	FALSE	TRUE
Pulse Enable	STW1	3	FALSE	TRUE
Ramp function Generator Enable	STW1	4	FALSE	TRUE
Setpoint Enable	STW1	6	FALSE	TRUE
Speedcontroller Enable	STW1	8	FALSE	TRUE
Master Controlled by PLC (Lifesignbit)	STW1	10	FALSE	TRUE

Figure 8-56 Control word 1 for opening and closing the grab in the air

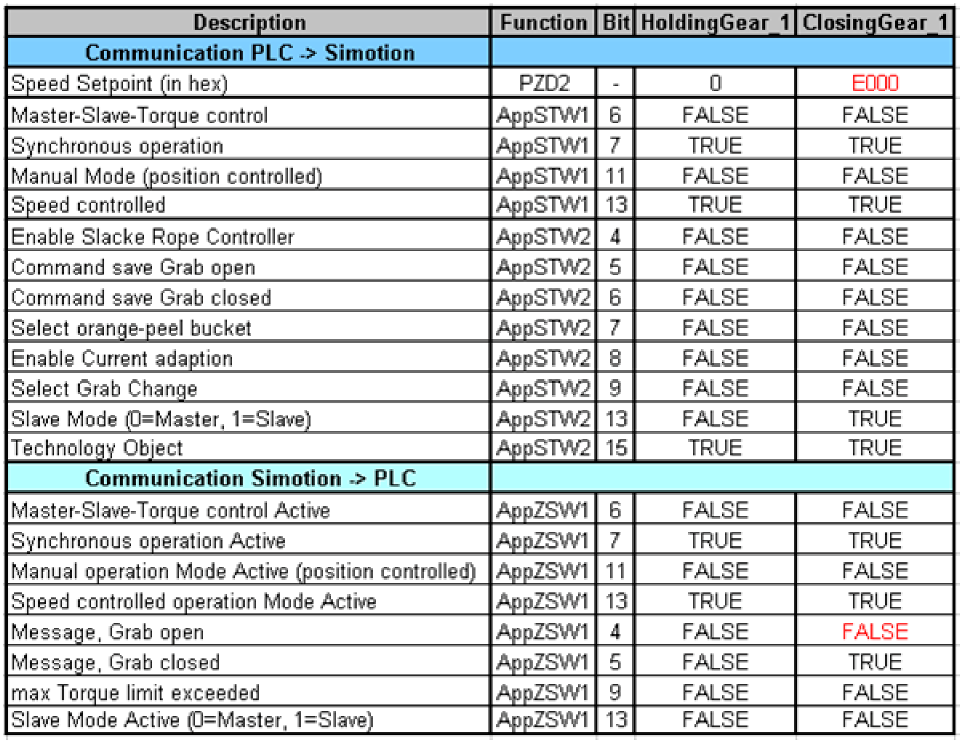
To close the grab, the closing gear must be moved with a positive setpoint. At the start of the closing operation, the checkback signal in application status word 1 indicates that the grab has still not been closed.



Description	Function	Bit	HoldingGear_1	ClosingGear_1
Communication PLC -> Simotion				
Speed Setpoint (in hex)	PZD2	-	0	2000
Master-Slave-Torque control	AppSTW1	6	FALSE	FALSE
Synchronous operation	AppSTW1	7	TRUE	TRUE
Manual Mode (position controlled)	AppSTW1	11	FALSE	FALSE
Speed controlled	AppSTW1	13	TRUE	TRUE
Enable Slacke Rope Controller	AppSTW2	4	FALSE	FALSE
Command save Grab open	AppSTW2	5	FALSE	FALSE
Command save Grab closed	AppSTW2	6	FALSE	FALSE
Select orange-peel bucket	AppSTW2	7	FALSE	FALSE
Enable Current adaption	AppSTW2	8	FALSE	FALSE
Select Grab Change	AppSTW2	9	FALSE	FALSE
Slave Mode (0=Master, 1=Slave)	AppSTW2	13	FALSE	TRUE
Technology Object	AppSTW2	15	TRUE	TRUE
Communication Simotion -> PLC				
Master-Slave-Torque control Active	AppZSW1	6	FALSE	FALSE
Synchronous operation Active	AppZSW1	7	TRUE	TRUE
Manual operation Mode Active (position controlled)	AppZSW1	11	FALSE	FALSE
Speed controlled operation Mode Active	AppZSW1	13	TRUE	TRUE
Message, Grab open	AppZSW1	4	FALSE	TRUE
Message, Grab closed	AppZSW1	5	FALSE	FALSE
max Torque limit exceeded	AppZSW1	9	FALSE	FALSE
Slave Mode Active (0=Master, 1=Slave)	AppZSW1	13	FALSE	FALSE

Figure 8-57 Activation of closing the grab in the air

Contrary to closing, opening requires a negative setpoint. The checkback signal in application control word 1 indicates that the grab is not yet fully open.

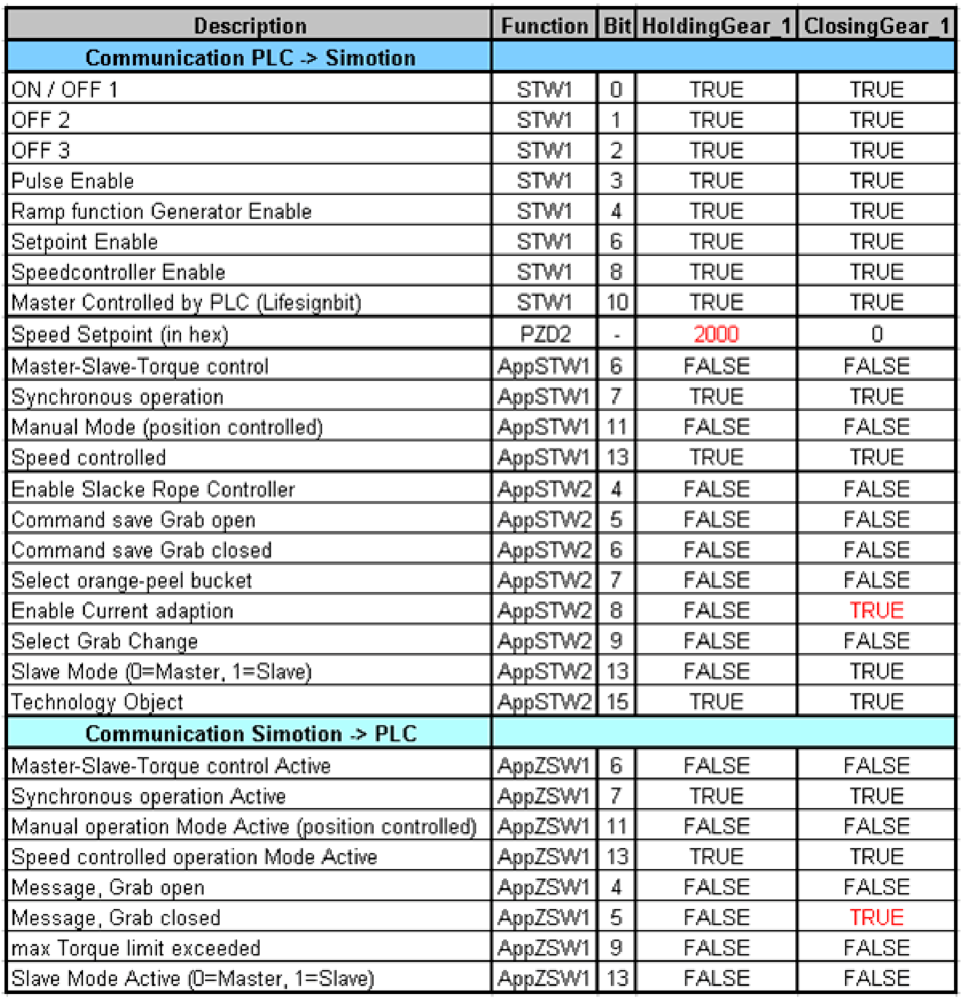


Description	Function	Bit	HoldingGear_1	ClosingGear_1
Communication PLC -> Simotion				
Speed Setpoint (in hex)	PZD2	-	0	E000
Master-Slave-Torque control	AppSTW1	6	FALSE	FALSE
Synchronous operation	AppSTW1	7	TRUE	TRUE
Manual Mode (position controlled)	AppSTW1	11	FALSE	FALSE
Speed controlled	AppSTW1	13	TRUE	TRUE
Enable Slacke Rope Controller	AppSTW2	4	FALSE	FALSE
Command save Grab open	AppSTW2	5	FALSE	FALSE
Command save Grab closed	AppSTW2	6	FALSE	FALSE
Select orange-peel bucket	AppSTW2	7	FALSE	FALSE
Enable Current adaption	AppSTW2	8	FALSE	FALSE
Select Grab Change	AppSTW2	9	FALSE	FALSE
Slave Mode (0=Master, 1=Slave)	AppSTW2	13	FALSE	TRUE
Technology Object	AppSTW2	15	TRUE	TRUE
Communication Simotion -> PLC				
Master-Slave-Torque control Active	AppZSW1	6	FALSE	FALSE
Synchronous operation Active	AppZSW1	7	TRUE	TRUE
Manual operation Mode Active (position controlled)	AppZSW1	11	FALSE	FALSE
Speed controlled operation Mode Active	AppZSW1	13	TRUE	TRUE
Message, Grab open	AppZSW1	4	FALSE	FALSE
Message, Grab closed	AppZSW1	5	FALSE	TRUE
max Torque limit exceeded	AppZSW1	9	FALSE	FALSE
Slave Mode Active (0=Master, 1=Slave)	AppZSW1	13	FALSE	FALSE

Figure 8-58 Activation of opening the grab in the air

Hoisting a closed grab

As mentioned above, both the holding gear and the closing gear have to move during hoisting and, as a result, their enable signals have to be available in control word 1. The hoisting motion task is started with a positive setpoint input at the holding gear; the closing gear does not receive a setpoint from the PLC. However, because the grab is closed and has to remain closed during hoisting, the current compensatory controller (AppSTW2 bit 8) has to be activated in the closing gear. This prevents the grab opening unintentionally. The lowering of the closed grab is controlled in a similar way (not illustrated here).

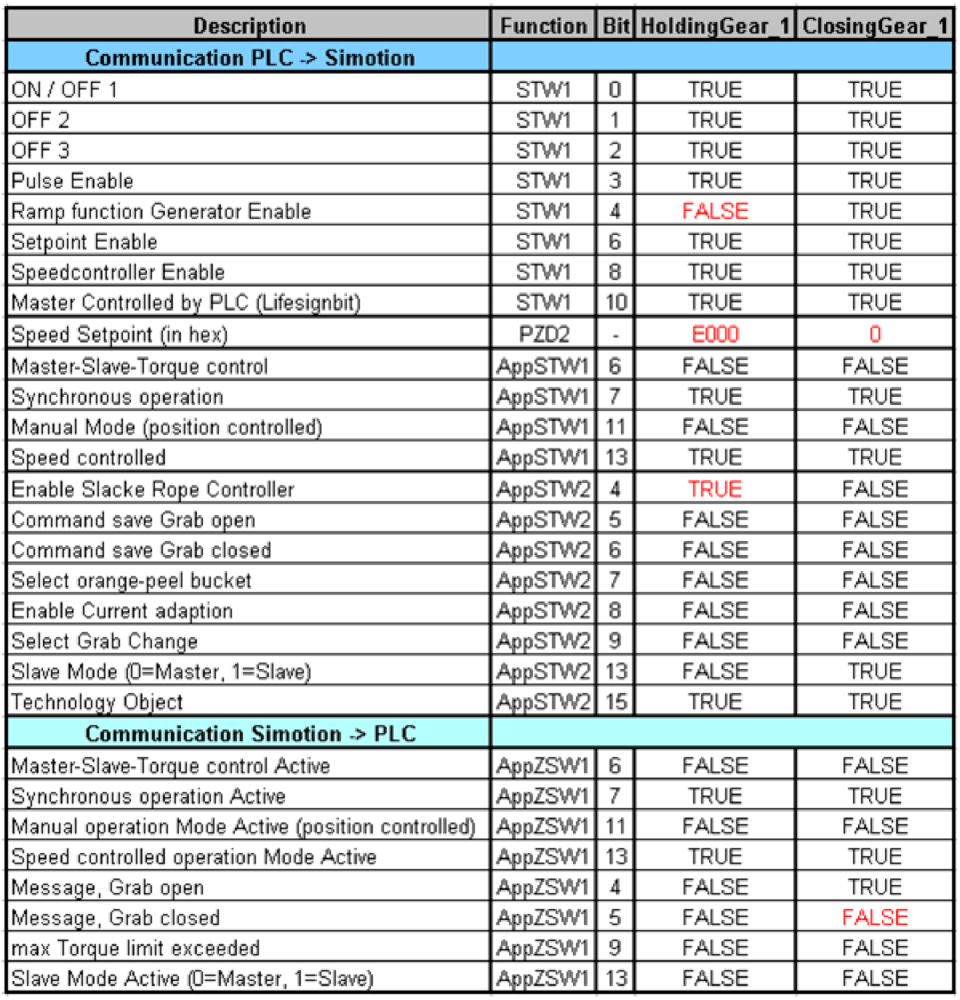


Description	Function	Bit	HoldingGear_1	ClosingGear_1
Communication PLC -> Simotion				
ON / OFF 1	STW1	0	TRUE	TRUE
OFF 2	STW1	1	TRUE	TRUE
OFF 3	STW1	2	TRUE	TRUE
Pulse Enable	STW1	3	TRUE	TRUE
Ramp function Generator Enable	STW1	4	TRUE	TRUE
Setpoint Enable	STW1	6	TRUE	TRUE
Speedcontroller Enable	STW1	8	TRUE	TRUE
Master Controlled by PLC (Lifesignbit)	STW1	10	TRUE	TRUE
Speed Setpoint (in hex)	PZD2	-	2000	0
Master-Slave-Torque control	AppSTW1	6	FALSE	FALSE
Synchronous operation	AppSTW1	7	TRUE	TRUE
Manual Mode (position controlled)	AppSTW1	11	FALSE	FALSE
Speed controlled	AppSTW1	13	TRUE	TRUE
Enable Slacke Rope Controller	AppSTW2	4	FALSE	FALSE
Command save Grab open	AppSTW2	5	FALSE	FALSE
Command save Grab closed	AppSTW2	6	FALSE	FALSE
Select orange-peel bucket	AppSTW2	7	FALSE	FALSE
Enable Current adaption	AppSTW2	8	FALSE	TRUE
Select Grab Change	AppSTW2	9	FALSE	FALSE
Slave Mode (0=Master, 1=Slave)	AppSTW2	13	FALSE	TRUE
Technology Object	AppSTW2	15	TRUE	TRUE
Communication Simotion -> PLC				
Master-Slave-Torque control Active	AppZSW1	6	FALSE	FALSE
Synchronous operation Active	AppZSW1	7	TRUE	TRUE
Manual operation Mode Active (position controlled)	AppZSW1	11	FALSE	FALSE
Speed controlled operation Mode Active	AppZSW1	13	TRUE	TRUE
Message, Grab open	AppZSW1	4	FALSE	FALSE
Message, Grab closed	AppZSW1	5	FALSE	TRUE
max Torque limit exceeded	AppZSW1	9	FALSE	FALSE
Slave Mode Active (0=Master, 1=Slave)	AppZSW1	13	FALSE	FALSE

Figure 8-59 Activation of hoisting the closed grab

Use the slack rope controller to close the grab once it has touched down.

If the grab is to take up a load (e.g. bulk goods), the open grab is set down on the material and then closed (positive setpoint at the closing gear). During closing, the grab plunges into the material; this can cause slackness in the rope at the holding gear. This slackening of the rope can be compensated by enabling the slack rope controller (AppSTW2 bit 4). As the holding gear drive moves during compensation, the enables have to be available (as illustrated in the figure below).



Description	Function	Bit	HoldingGear_1	ClosingGear_1
Communication PLC -> Simotion				
ON / OFF 1	STW1	0	TRUE	TRUE
OFF 2	STW1	1	TRUE	TRUE
OFF 3	STW1	2	TRUE	TRUE
Pulse Enable	STW1	3	TRUE	TRUE
Ramp function Generator Enable	STW1	4	FALSE	TRUE
Setpoint Enable	STW1	6	TRUE	TRUE
Speedcontroller Enable	STW1	8	TRUE	TRUE
Master Controlled by PLC (Lifesignbit)	STW1	10	TRUE	TRUE
Speed Setpoint (in hex)	PZD2	-	E000	0
Master-Slave-Torque control	AppSTW1	6	FALSE	FALSE
Synchronous operation	AppSTW1	7	TRUE	TRUE
Manual Mode (position controlled)	AppSTW1	11	FALSE	FALSE
Speed controlled	AppSTW1	13	TRUE	TRUE
Enable Slacke Rope Controller	AppSTW2	4	TRUE	FALSE
Command save Grab open	AppSTW2	5	FALSE	FALSE
Command save Grab closed	AppSTW2	6	FALSE	FALSE
Select orange-peel bucket	AppSTW2	7	FALSE	FALSE
Enable Current adaption	AppSTW2	8	FALSE	FALSE
Select Grab Change	AppSTW2	9	FALSE	FALSE
Slave Mode (0=Master, 1=Slave)	AppSTW2	13	FALSE	TRUE
Technology Object	AppSTW2	15	TRUE	TRUE
Communication Simotion -> PLC				
Master-Slave-Torque control Active	AppZSW1	6	FALSE	FALSE
Synchronous operation Active	AppZSW1	7	TRUE	TRUE
Manual operation Mode Active (position controlled)	AppZSW1	11	FALSE	FALSE
Speed controlled operation Mode Active	AppZSW1	13	TRUE	TRUE
Message, Grab open	AppZSW1	4	FALSE	TRUE
Message, Grab closed	AppZSW1	5	FALSE	FALSE
max Torque limit exceeded	AppZSW1	9	FALSE	FALSE
Slave Mode Active (0=Master, 1=Slave)	AppZSW1	13	FALSE	FALSE

Figure 8-60 Activation of the grab for closing with slack rope controller

- If the holding gear needs to be moved independently of the closing gear, then synchronous operation must be deselected in the holding gear (AppSTW2 bit 7 = false).
- To move the closing gear independently of the holding gear, deselect slave operation in the closing gear or activate grab change (AppSTW2 bit 13 = false or AppSTW2 bit 9 = true).
- It is only advisable to operate the holding gear and closing gear independently of one another during commissioning and when grabs are being changed.

8.14.2.13 Grab monitoring

The block DCC_GrabMonitor can be used for grab cranes. This block issues a high signal at the output if a set torque is exceeded; see also the section titled DCC_GrabMonitor (Page 102).

8.14.2.14 Continuous load measurement

When the ropes are attached, the measurements described in the section titled Continuous load measurement (CLM) (Page 434) can be taken. One of the DCC_ContLoadMeasurement or DCC_ContLoadMeasurement_1 blocks issues the "Grab touchdown" message, among others. This message is important for activating the slack rope controller. However, to use it a full grab is needed, which only works with a functioning slack rope controller. To be able to fill the grab despite this, you must use the holding gear to manually create a slack rope; this enables the grab to bury itself into the material even without a slack rope controller and the measurements performed with the rated load (i.e. with a full grab).

When calculating differences in weight in the event of a deviating rope length for the variable "rCompensationRopeFactor", you must ensure that the rope difference is as large as possible. If the rope difference is small (< 20 m), it is perfectly possible that the load shown will swing too much with a deviating rope length. In this case, leave "rCompensationRopeFactor" at 0.0, if this means the output of the block will not fluctuate as much.

8.14.2.15 Setting control blocks in the DCC

The current compensatory controller is used to distribute the load as evenly as possible between the holding gear and the closing gear. A corresponding additional setpoint is then added/subtracted at the closing gear in order to adjust the current consumption in the closing gear. The current compensatory controller must only be activated with the encoder closed during a hoisting or lowering procedure; it ensures that the grab is closed (for information on the current compensatory controller, see the section titled DCC_CurrentEqualControl (Page 95)).

Practical experience has shown the best course of action is to use the current compensatory controller only as a proportional controller. To do this, set the input "boHoldIntegration" to 1 (true). If operating the crane with current compensatory control activated leads to vibrations, the gain "rPGainCurrentEqualCtrl" should be reduced. The smoothing time for the current and the setpoint velocity also need to be taken into account.

The slack rope controller ensures that the ropes in the holding gear are kept tight, but do not hoist the grab. This enables the grab to bury into the material to be moved. The slack rope controller should only be activated once the grab has touched down and a closing command is issued. The slack rope controller is deactivated when the closing gear has closed or when the holding gear switches to hoisting operation (for information on slack rope controllers, see the section titled DCC_SlackRopeControl (Page 137)).

With the slack rope controller you can store a characteristic curve of how the holding gear should move in relation to the current, or you can also use the integrated proportional controller, which is the most frequently used option. Here too, the gain "rp_gain" must be set in such a way that the ropes are pulled tight and are free of vibration, but the closing gear is not prevented from sagging. Experience has shown that 1 to 2% is a good value for the comparison value "rCompareValueSetpointSRC". This is just a guide value, however, which may be set lower or higher, depending on how the slack rope controller is to function.

Note

The current compensatory controller and the slack rope controller must not both be active at the same time.

8.14.2.16 Correct grab adjustment

To adjust the grab, proceed as follows:

1. Move the grab to the upper end position until the limit switch (mechanical, if possible) switches the holding gear off.
2. Activate grab change (AppSTW2 bit 9 = True) and open the grab fully.
3. Home the holding gear and closing gear. Both will then have the actual position 0 mm or another identical position. The important thing is that the difference between the holding gear and the closing gear must be 0 mm.
4. Next, save the position as "Grab open" (positive edge in AppSTW2 bit 5).
5. Close the grab until it is shut but the ropes are still tight.
6. Next, save the position as "Grab closed" (positive edge in AppSTW2 bit 6).
7. Deactivate grab change (AppSTW2 bit 9 = False).
8. Test the opening and closing operations. The grab must automatically slow down and become stationary when the "Open" or "Closed" positions are reached.
9. The velocity of the closing gear can be set via the "KP_ChangeOverPoint" variable; see Setting the closing velocity (Page 458).
10. Finally, save the data.

Grab adjustment is now complete.

8.15 Ship-to-shore tandem crane (STS tandem)

This chapter describes tandem mode with four drives. Terminal operators always want to increase their cargo handling volumes when loading or unloading ships. This increase ensures that ships' docking times are reduced. Cargo handling volumes can be increased through the use of double spreaders.

Two spreaders are installed on a container unloading crane. Each spreader can pick up either one 40-foot or two 20-foot containers. Each spreader is driven by 2 motors and 2 converters, which serve to hoist or lower it. In tandem mode, the two spreaders work in velocity gearing or in position synchronous operation, depending on the selected operating mode.

The following rules apply for tandem mode:

- Hoisting-gear motor 1 = master 1 (tandem master)
- Hoisting-gear motor 2 = slave 1
- Hoisting-gear motor 3 = master 2 (tandem slave)
- Hoisting-gear motor 4 = slave 2

Master 1 and slave 1 can be connected to one another via a rigid connection (master-slave operation) or a flexible connection (synchronous operation). The same also applies to master 2 and slave 2. For information on how to select synchronous or master-slave operation for master 1 and slave 1 or master 2 and slave 2, see FB_OperationMode (Page 194).

In tandem mode, master 1 and master 2 run in synchronous operation. The slaves follow their respective master. This ensures that both spreaders are hoisted and/or lowered at the same time.

8.15.1 Getting started

The initial stages of commissioning described above in Grab Ship Unloader (GSU) (Page 454) can also be used as a guide for the STS tandem crane.

8.15.2 Interconnections

In order to give a clearer picture of the tasks associated with each drive, as well as the "master" and "slave" designations, in tandem mode there are also the "tandem master" and "tandem slave" designations. These describe the connection between the two masters. The master selected as the tandem master takes on the master role. The other master, which then has to be selected as the tandem slave, follows the tandem master.

The figure below depicts the exact relationships between the drives in tandem mode.

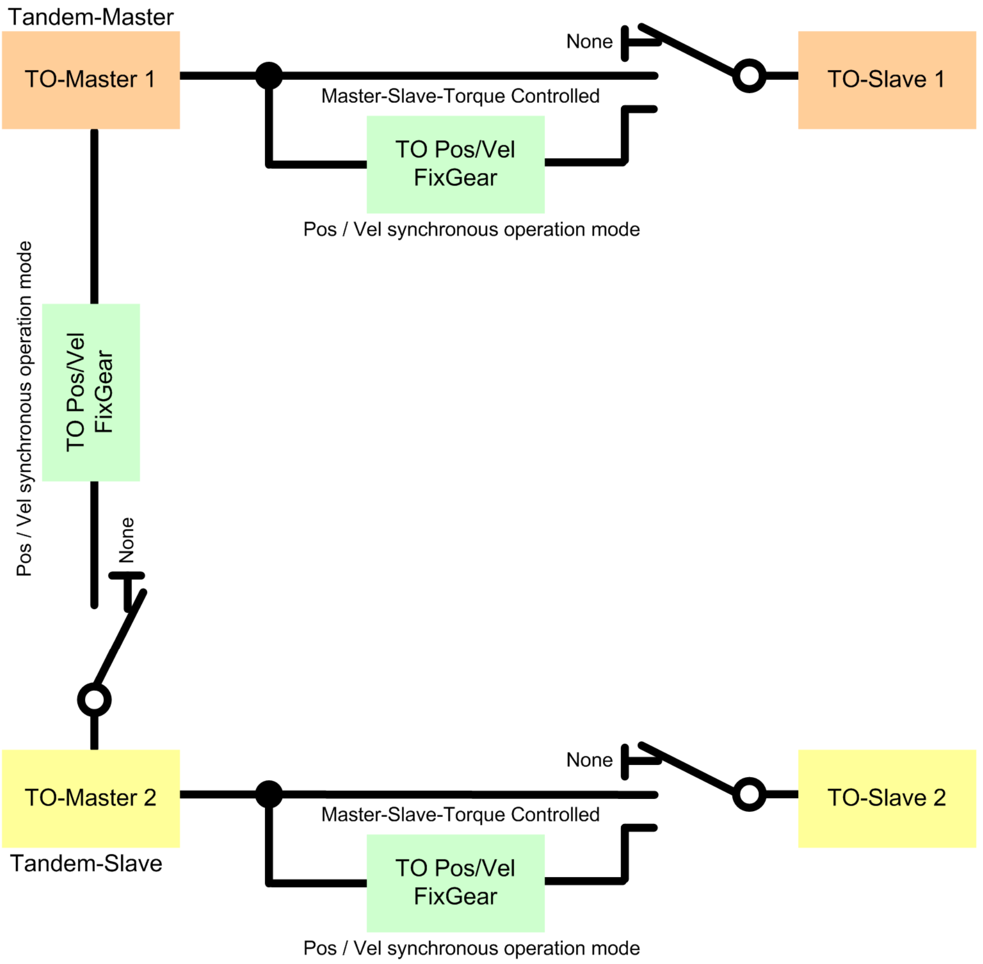

Tandem-Master
TO-Master 1
None
Master-Slave-Torque Controlled
TO-Slave 1
TO Pos/Vel FixGear
Pos / Vel synchronous operation mode
Pos / Vel synchronous operation mode
TO Pos/Vel FixGear
None
TO-Master 2
Tandem-Slave
None
Master-Slave-Torque Controlled
TO-Slave 2
TO Pos/Vel FixGear
Pos / Vel synchronous operation mode

Figure 8-61 Representation of tandem mode

Fixed gear connections are needed for synchronous operation between the drives. A gear connection is not required for master-slave operation. The table below shows the six gears and their motion inputs.

A connection is configured for each of the position and velocity setpoints from Hoist_1 to Hoist_2 and from Hoist_3 to Hoist_4. The connection for tandem mode is implemented via Tandem_Position and Tandem_Velocity.

Table 8- 15 Gear connections

Fixed gear	Motion input
Hoist_Position_TM	D435 – Hoist_1 - Setpoint
Hoist_Velocity_TM	D435 – Hoist_1 - Setpoint
Hoist_Position_TS	D435 – Hoist_3 - Setpoint
Hoist_Velocity_TS	D435 – Hoist_3 - Setpoint
Tandem_Position	D435 – Hoist_1 - Setpoint
Tandem_Velocity	D435 – Hoist_1 - Setpoint

The motion outputs of the gears are interconnected with the motion inputs of the axes.

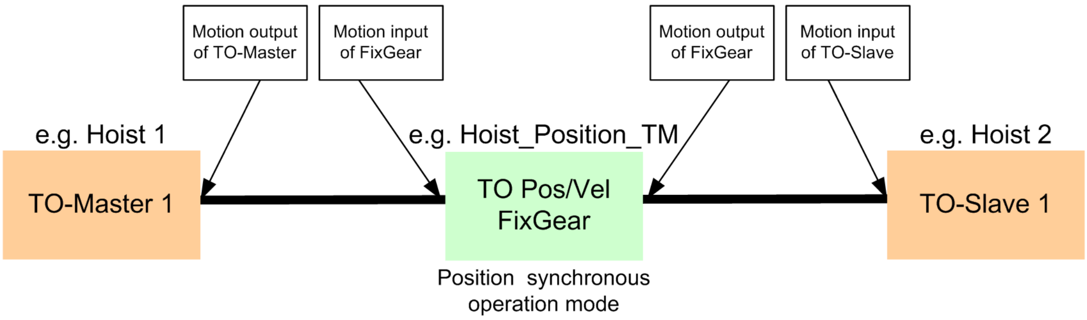


Figure 8-62 Diagram showing position synchronous operation with hoist 1 and hoist 2

8.15.3 Interface description

The six bits listed below are sent from the S7 to the SIMOTION in order for the full range of functions to be used in tandem mode.

Table 8- 16 Control word_2_S7 (S7 → SIMOTION)

Bit	Signal name	Remarks
0	boPositiveSuperimpose	Using this bit a positive offset is established between the tandem-master and tandem-slave in the tandem mode or between the master and slave in the offset mode. The bit is connected with the FB_OperationMode block from the tandem-slave or slave and is only required there, no matter whether the offset is moved with the master or with the slave.
1	boNegativeSuperimpos e	Using this bit a negative offset is established between the tandem-master and tandem-slave in the tandem mode or between the master and slave in the offset mode. The bit is connected with the FB_OperationMode block from the tandem-slave or slave and is only required there, no matter whether the offset is moved with the master or with the slave.

Bit	Signal name	Remarks
2	boTandemHoming	This signal can be used to automatically cancel the offset that has been created. The operating mode (speed-controlled or position-controlled) does not have any effect here. The velocity at which the offset is canceled is the same applied to create it. The offset can be canceled on the fly or at a standstill by the tandem master or tandem slave. However, this bit is only required in the tandem-slave in the FB_OperationMode block and interconnected. The homing procedure is started simply by evaluating the positive edge. Homing remains active until the offset has been canceled, stopped by a new homing sequence or until it is interrupted by reactivating the direction signals (STW2 bit 0/1) (tandem slave). When the current actual position values of the tandem master and the tandem slave are identical, the offset has been canceled. NOTICE: Homing is active where an offset is being traveled through. Homing is also performed if a tandem-slave is activated for superimposed movement (see STW2 bit 5 "boDriveMasterSuperimpose").
3	boTandemMode	In order to select tandem mode, the bit must be set for the tandem master and the tandem slave. The bit is interconnected with FB_OperationMode. This indicates to the two masters that they should run in synchronous operation.
4	boSlaveTandemMode	The input is only interconnected for the tandem slave FB_OperationMode. The signal state of this bit (TRUE) signals to the master that it must now operate as tandem slave. Although the tandem slave is actually a master in its own right, it follows the tandem master in the tandem mode.
5	boDriveMasterSuperimpose	This bit is used to define which drive should travel the offset. If the bit = false, then the tandem-slave performs the superimposed movement. If the bit = true, then the tandem-master performs the superimposed movement. The homing can then also be performed by the corresponding drive. This bit only has to be interconnected for the tandem-slave. Note: This bit should only be changed at standstill.

The superimposed movement to establish and cancel an offset can be performed both by the tandem master as well as by the tandem slave. However, regarding communication, nothing changes. For the tandem-master, only bit 3 "boTandemMode" and bit 4 "boSlaveTandemMode" are required. For the tandem-slave, all six bits are required (bit 0 – bit 5).

Note

All offset and equalization travels are controlled by the tandem-master and tandem-slave. This is the reason that for the tandem-slave all control bits are required, and for the tandem-master, only bit 3.

The master drive must always monitor the feedback signals from the slave. This means: Hoist 1 (tandem-master) monitors the signals from hoist 2; hoist 3 (tandem-slave) monitors the signals from hoist 4. This means control word 2, bit 8 must be set to TRUE for hoist 1 and hoist 3.

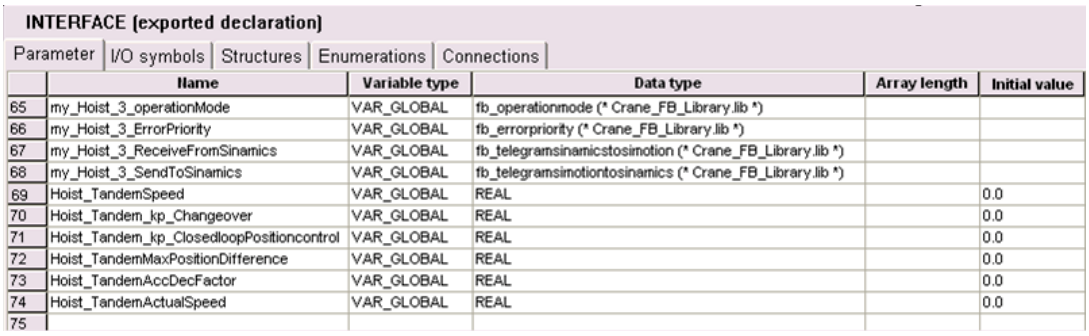
In addition, control word 2 bit 9/10 must be TRUE for hoist 1 (tandem-master) because this as the tandem-master must monitor the tandem-slave (hoist 3) and the tandem-slave-slave (hoist 4).

The checkback signals (SIMOTION → S7) are listed in the table below.

Table 8- 17 Status word_2_S7 (SIMOTION → S7)

Bit	Signal name	Remarks
2	boTandemHomingActive	This bit is True for as long as the homing motion task is active. Once the homing procedure has been completed or interrupted, this bit is false. The feedback signal only comes from the slave.
3	boTandemModeActive	The drives on which tandem mode has been activated and still remains active use this bit to signal their state.
4	boSlaveTandemModeActive	The master which is to act as the tandem slave and, as such, follows the tandem master, signals True via this bit.
5	boDriveMasterSuperimposeActive	This bit returns who performed the offset travel and homing. If the feedback is true, then the tandem-master moves through the offset. If the feedback is false, then the tandem-slave moves through the offset. The feedback signal only comes from the slave.

The following input signals must be set for the tandem-slave. For this purpose, the inputs are already interconnected with unit-global variables and can therefore be sent in the MCC unit from the tandem-slave (see the following diagram as example):



INTERFACE (exported declaration)

	Name	Variable type	Data type	Array length	Initial value
65	my_Hoist_3_operationMode	VAR_GLOBAL	fb_operationmode (* Crane_FB_Library.lib *)		
66	my_Hoist_3_ErrorPriority	VAR_GLOBAL	fb_errorpriority (* Crane_FB_Library.lib *)		
67	my_Hoist_3_ReceiveFromSinamics	VAR_GLOBAL	fb_telegramsinamicstosimotion (* Crane_FB_Library.lib *)		
68	my_Hoist_3_SendToSinamics	VAR_GLOBAL	fb_telegramsimotiontosinamics (* Crane_FB_Library.lib *)		
69	Hoist_TandemSpeed	VAR_GLOBAL	REAL		0.0
70	Hoist_Tandem_kp_Changeover	VAR_GLOBAL	REAL		0.0
71	Hoist_Tandem_kp_ClosedloopPositioncontrol	VAR_GLOBAL	REAL		0.0
72	Hoist_TandemMaxPositionDifference	VAR_GLOBAL	REAL		0.0
73	Hoist_TandemAccDecFactor	VAR_GLOBAL	REAL		0.0
74	Hoist_TandemActualSpeed	VAR_GLOBAL	REAL		0.0
75					

Table 8- 18 Function block OperationMode

Name	Connection type	Default setting	Data type	Meaning
r64TandemSlaveSpeed (In the MCC unit: To_Name_Tandemspeed)	IN	0.0	LREAL	This value specifies what percentage of the maximum velocity should be superimposed on the tandem-master or tandem-slave as an additional setpoint when a superimposed motion is to be performed [%] (max. 50%). This input only has to be supplied for the tandem-slave.
r64AccDecFactorTandem (In the MCC unit: To_Name_TandemAccDecFactor)	IN	0.01	LREAL	This value specifies what percentage of the actual acceleration/deceleration value should be applied as the superimposed value when a superimposed motion is to be performed [%] (max. 50%). This input only has to be supplied for the tandem-slave.
r64maxTandemPositiondifference (In the MCC unit: To_Name_TandemMaxPositionDifference)	IN	0.0	LREAL	Specifies the maximum offset which can be applied in the positive or negative direction [mm]. This input only has to be supplied for the tandem-slave.
r64kpTandemChangeOverPoint (In the MCC unit: To_Name_Tandem_kp_changeover)	IN	1.0	LREAL	This value is the gain factor for path control, which is required for homing and offset motion in speed-controlled operation. The higher the value, the faster the deceleration when the homing position is reached. The gain factor is only needed in speed-controlled synchronous operation. (The section titled Setting the closing velocity (Page 458) describes how to set the gain factor.) This input only has to be supplied for the tandem-slave. **Caution:** The gain should be selected such that this rate of deceleration can also be achieved with the current ramp-down time.
r64ActualTandemSlaveSpeed (In the MCC unit: To_Name_TandemActualSpeed)	OUT	0.0	LREAL	Actual additional velocity for the superimposed motion of the tandem slave This output must be read from the tandem-slave.

8.15.4 Procedure

Settings for master 1 (tandem master):

- The "boTandemMode" bit must be active (STW2 bit 3).
- Select bit "select synchronous operation" (AppSTW1 bit 7) or "Select master-slave operation" (AppSTW1 bit 6) (depending on which coupling is required between master 1 and slave 1), so that slave 1 can check the checkback signals.
- Independent of which coupling is used between master 1 and slave 1, for the synchronous operation coupling between the tandem-master and tandem-slave, in tandem operation, synchronous operation must always be selected (AppSTW1 bit 7). Only for the tandem operation is it permitted – when required – to simultaneously select synchronous operation and master-slave operation.
- Select the required operating mode. (In position-controlled operation, home all the drives.)
- The "boselectcheckbackslave", "boselectcheckbackslavetandem" and "boselectcheckbackslaveslavetandem" inputs (STW2 bits 8,9,10) must be set to true so that the tandem-master can monitor all of the drives.

Settings for master 2 (tandem slave):

- The "boTandemMode" bit must be active (STW2 bit 3).
- The "boSlaveTandemMode" bit must be active (STW2 bit 4).
- The "boTandemHoming" bit is only set in the event of a need for further offset compensation (STW2 bit 2).
- Select bit "select synchronous operation" (AppSTW1 bit 7) or "Select master-slave operation" (AppSTW1 bit 6) (depending on which coupling is required between master 2 and slave 2), so that master 2 can check the checkbox signals from slave 2.
- Independent of which coupling is used between master 2 and slave 2, for the synchronous operation coupling between the tandem-master and tandem-slave, in tandem operation, synchronous operation must always be selected (AppSTW1 bit 7). Only for the tandem operation is it permitted – when required – to simultaneously select synchronous operation and master-slave operation.
- Select the same operating mode as for master 1.
- The "boselectcheckbackslave" input (STW2 bit 8) must be set to true so that the tandem-slave can monitor the slave drive.

Settings for slave 1:

- Select the "boSlaveMode" bit to ensure that slave 1 actually works as a slave (AppSTW2 bit 13).
- Select the same operating mode as for master 1.
- Select same coupling as for master 1: Synchronous operation coupling (AppSTW1 bit 7) or master-slave coupling (AppSTW1 bit 6).

Settings for slave 2:

- Select the "boSlaveMode" bit to ensure that slave 2 actually works as a slave (AppSTW2 bit 13).
- Select the same operating mode as for master 1.
- Select same coupling as for master 2: Synchronous operation coupling (AppSTW1 bit 7) or master-slave coupling (AppSTW1 bit 6)

8.15.5 Moving in tandem mode

A motion will not start until all checkback signals are correct. Certain conditions, including the following, must be met for this to be the case:

- There must be no error message present at the technology object, the drive object, or the function block.
- Power and DriveState must be active (control word_1_S7, bits 3, 4, 6 and 8) and the feedback from the drive (status word_1_S7, bit 2) should also be available, before a travel command or a direction bit (STW2 bit 0/1) is issued.
- S7 must also have previously set the enable signals above – also for homing. If homing is to be performed, and the enable signals have still not been set, then the homing signal must be appropriately delayed.
- The operating mode of the master must be the same as that of the slave.
- Master-slave or synchronous operation is active for every Hoist group.
- When position-controlled tandem mode is activated for the first time, any existing offset must be taken into account. To achieve this - for the tandem-slave - once synchronous operation has been activated (AppSTW1 bit 7), the offset must be saved (AppSTW1 bit 8) before switching on the drives (e.g. STW1 bit 3). It is essential that this is done, otherwise jumps may occur for the tandem-slave in tandem operation.

Note

If the tandem-master has reached its target position, then an offset can still always be traveled using the tandem-slave. The following applies:
End position, tandem-slave = end position, tandem-master + maximum offset

The difference between the end position of the tandem-master and the limit switch should be at least the value of the maximum tandem offset.

Note

Both drives must always be switched on, even if at standstill only one drive should travel an offset. For instance, if the tandem-master should travel through an offset at standstill, then the tandem-master and the tandem-slave must be activated and all enable signals set. However, for the tandem-slave, the brakes can be kept closed.

Note

For closed-loop speed controlled tandem mode (SPEED_CONTROLLED operating mode), for homing or when traveling through an offset, the position isn't so precisely approached as for position-controlled tandem mode (MANUAL operating mode). Deviations of up to a millimeter can occur in closed-loop speed controlled operation.

Note

If the tandem master and the tandem slave are traveling with the maximum velocity, they cannot travel any faster to compensate for an offset. Instead, the drive that is leading, must be braked in order to compensate the offset.

Note

In the tandem mode, all checkback signals are sent to the FB_OperationMode block of master 1. The following name convention is observed when reading into FB_OperationMode of master 1:

- Slave 1:

  bocheckback**Slave**ErrorTO, etc.
  r64checkback**Slave**offsetSave
- Master 2:

  bocheckback**SlaveTandem**ErrorTO, etc.
  r64checkback**SlaveTandem**offsetsave
- Slave 2:

  bocheckback**SlaveSlaveTandem**ErrorTO, etc.
  r64checkback**SlaveSlaveTandem**offsetsave

If a required checkback signal is no longer received or if an error occurs during tandem movement, then tandem operation is stopped via the ramp-down-time ramp.

8.15.6 Offset mode between tandem master and tandem slave (TandemMode)

In tandem mode, the four hoisting-gear motors or both masters and slaves are controlled via a master switch. This is the master switch of the tandem master. Direction signals (STW2 Bit 0/1) are available for offset travel; these superimpose a positive or negative velocity setpoint in addition to the basis motion.

An offset can be created or canceled (on the fly, if necessary) between the tandem master and the tandem slave. In order to control the offset on the fly during operation, STW2 bit 5 should be used to define whether the tandem-master or the tandem-slave is allocated the velocity superimposition. The set, maximum velocity also cannot be exceeded for velocity superimposition. If the tandem-master and tandem-slave of already traveling at the maximum velocity, then one of the two must be slowed down if an offset is to be traveled through.

The maximum offset between the two must be defined during the configuration or commissioning phase. The offset that has been set is not exceeded and can be traveled through by either the tandem-master or tandem-slave.

The offset is increased until the maximum offset is reached or until the direction bit (STW2 Bit 0/1) is no longer selected.

A positive edge of the "boTandemHoming" signal causes the offset that has been created to be canceled again automatically.

The action remains active until

- The motion task is complete and the offset is zero
  or
- Homing is interrupted by the direction signal (STW2 bit 0/1),
  or
- Homing is interrupted by a new homing signal.

Note

If the load-dependent field weakening is used, this then specifies a maximum velocity dependent on the load. If the maximum velocity is reduced because of the load that is already being moved by the master, the slave must not move faster. The permissible velocity is monitored and, if necessary, the supplementary velocity for tandem or offset limited accordingly.

If no load-dependent field weakening is used, the "boCheckLDFWLimitation" input must be set to FALSE.

8.15.7 Offset mode between the master and the slave

Offset mode is comparable with tandem mode in terms of its functionality. In tandem mode, an offset is applied between the tandem master and the tandem slave. Both of these drives act as masters. In the offset mode, however, an offset is applied between the master and the slave, which run with a synchronous-mode coupling. Contrary to the tandem mode, in the offset mode, only the slave can travel through an offset.

To travel through an offset, the same direction bits are used as for the tandem mode (STW2 bit 0/1). Just as in tandem mode, here too the offset can be created or canceled on the fly, a maximum offset can be defined, and an existing offset can be canceled again automatically via "boOffsetHoming". Homing also remains active in this mode until the offset has been canceled, a direction bit is present for new offset motion or the homing signal is activated again. All motion superimposition is issued to the slave.

Note

The tandem mode and offset mode mutually exclude one another. If hoist 1 and hoist 3 are moved in the tandem mode, then it is not permissible that the respective slaves, hoist 2 and hoist 4 operate in the offset mode

Note

If the load-dependent field weakening is used, this then specifies a maximum velocity dependent on the load. If the maximum velocity is reduced because of the load that is already being moved by the master, the slave must not move faster. The permissible velocity is monitored and, if necessary, the supplementary velocity for tandem or offset limited accordingly.

If no load-dependent field weakening is used, the "boCheckLDFWLimitation" input must be set to FALSE.

Interface description

Offset mode is essentially controlled with the three bits described below.

Table 8- 19 Control word_2_S7 (S7 → SIMOTION)

Bit	Signal name	Remarks
11	boOffsetHoming	This signal can be used to automatically cancel the offset that has been created. The operating mode (speed-controlled or position-controlled) does not have any effect here. The velocity at which the offset is canceled is the same as when establishing an offset. The offset can be canceled on the fly or at a standstill. Since the offset can only be created or canceled by the slave, this bit is only interconnected in the slave, in block FB_OperationMode. Only the positive edge is evaluated to start homing. Homing remains active until the offset has been canceled or until it is interrupted by the direction signal being activated again or a new homing signal. When the current actual position values of the master and the slave are identical, the offset has been canceled.
12	boOffsetMode	In order to select offset mode, this bit must only be set for the slave. This indicates to the drive that an offset can be applied.

Table 8- 20 Status word_2_S7 (SIMOTION → S7)

Bit	Signal name	Remarks
11	boOffsetHomingActive	This bit is True for as long as the homing motion task is active. As soon as the homing procedure is completed or interrupted, this bit switches to False. The feedback signal only comes from the slave.
12	boOffsetModeActive	The drive that applies the offset indicates that this is the case by signaling back via this bit. The feedback signal only comes from the slave.

The following input signals must be set at FB_OperationMode of the slave:

Table 8- 21 Function block OperationMode

Name	Connection type	Default setting	Data type	Meaning
r64OffsetSlaveSpeed (In the MCC unit: To_Name_Offsetspeed)	IN	0.0	LREAL	Specifies what percentage of the maximum velocity should be superimposed on the slave as an additional setpoint when a superimposed motion is to be performed [%] (max. 50%). This input only has to be supplied for the slave.
r64AccDecFactorOffset (In the MCC unit: To_Name_OffsetAccDecFactor)	IN	0.01	LREAL	Specifies what percentage of the current acceleration/deceleration value should be applied as the superimposed value when a superimposed motion is to be performed [%] (max. 50%). This input only has to be supplied for the slave.
r64maxOffsetPositiondifference (In the MCC unit: To_Name_OffsetMaxPositionDifference)	IN	0.0	LREAL	Specifies the maximum offset which can be applied in the positive or negative direction [mm]. This input only has to be supplied for the slave.
r64kpOffsetChangeOverPoint (In the MCC Unit: To_Name_Offset_kp_changeover)	IN	1.0	LREAL	This value is the gain factor for path control, which is required for homing and offset motion in speed-controlled operation. The higher the value, the higher the rate of deceleration when reaching the homing position. The gain factor is only required in closed-loop speed controlled synchronous operation. (Setting the gain factor is described in Chapter Setting the closing velocity (Page 458).) This input only has to be supplied for the slave. **Caution:** Select the gain factor such that this rate of deceleration can also be achieved with the current ramp-down time.
r64ActualOffsetSlaveSpeed (In the MCC unit: To_Name_OffsetActualSpeed)	OUT	0.0	LREAL	Actual additional velocity for the superimposed motion for the slave This output must be read from the slave.

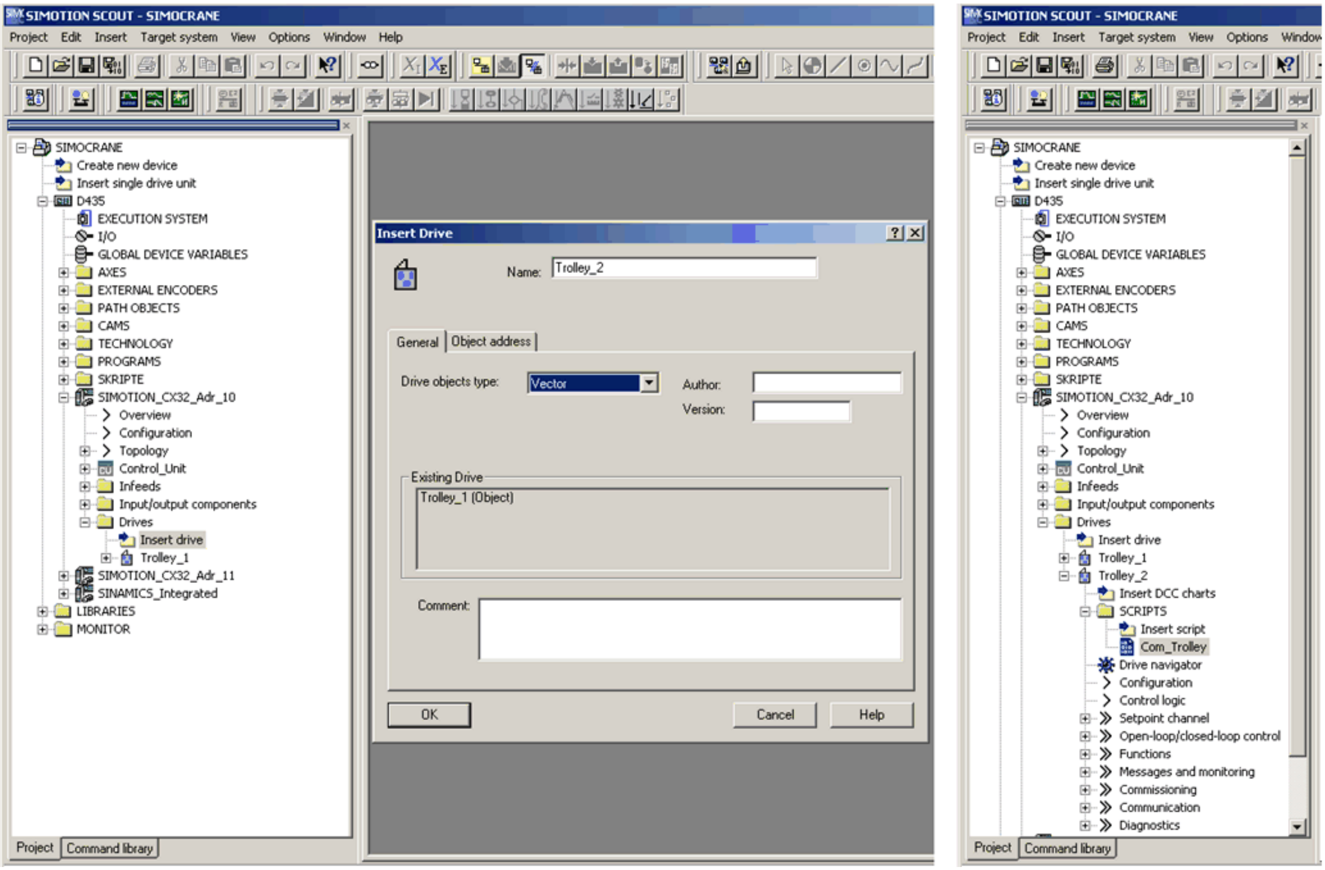
Possibility of adaptation

9

The possibility of adaptation of the standard project is described in the following using as an example an expansion from one trolley to two trolleys. The entire connection is described in full detail for the MCC level and DCC level in Ref. [11].

9.1 Inserting a drive object

To add a second Trolley to the project, drive Trolley_2 needs to be created and the script file copied from Trolley_1.



Insert Drive

Name: Trolley_2

General | Object address

Drive objects type: Vector

Author:

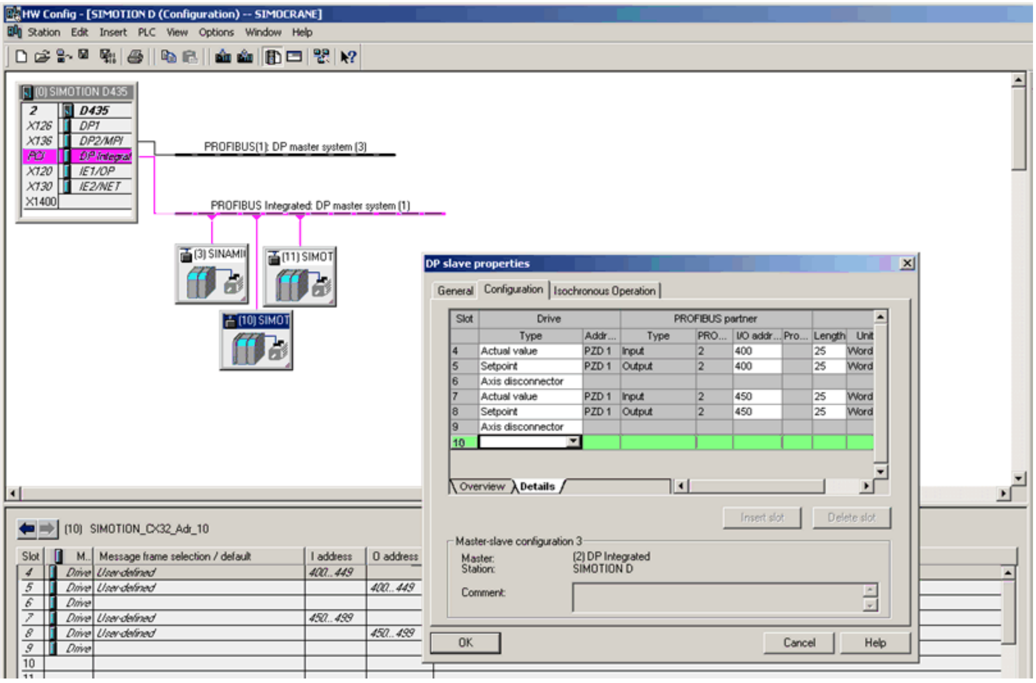
Version:

Existing Drive: Trolley_1 (Object)

Comment:

Figure 9-1 Inserting Trolley_2 and copying Com_Trolley from Trolley_1

Then add the addresses for SIMOTION – SINAMICS communication into the hardware configuration; see Communication, SIMOTION D – SINAMICS (internal SIMOTION D communication) (Page 323).


HW Config - [SIMOTION D (Configuration) -- SIMOCRANE]
Station Edit Insert PLC View Options Window Help
(0) SIMOTION D435
2 D435
X126 DP1
X136 DP2/MPI
PCI DP Integrated
X120 IE1/OP
X130 IE2/NET
X1400
PROFIBUS(1): DP master system (3)
PROFIBUS Integrated: DP master system (1)
(3) SINAMI
(11) SIMOT
(10) SIMOT

DP slave properties

General | Configuration | Isochronous Operation

Slot	Drive		PROFIBUS partner					
	Type	Addr ...	Type	PRO...	I/O addr ...	Pro...	Length	Unit
4	Actual value	PZD 1	Input	2	400		25	Word
5	Setpoint	PZD 1	Output	2	400		25	Word
6	Axis disconnector							
7	Actual value	PZD 1	Input	2	450		25	Word
8	Setpoint	PZD 1	Output	2	450		25	Word
9	Axis disconnector							
10								

Overview | Details

Insert slot | Delete slot

Master-slave configuration 3
Master: (2) DP Integrated
Station: SIMOTION D
Comment:

OK | Cancel | Help

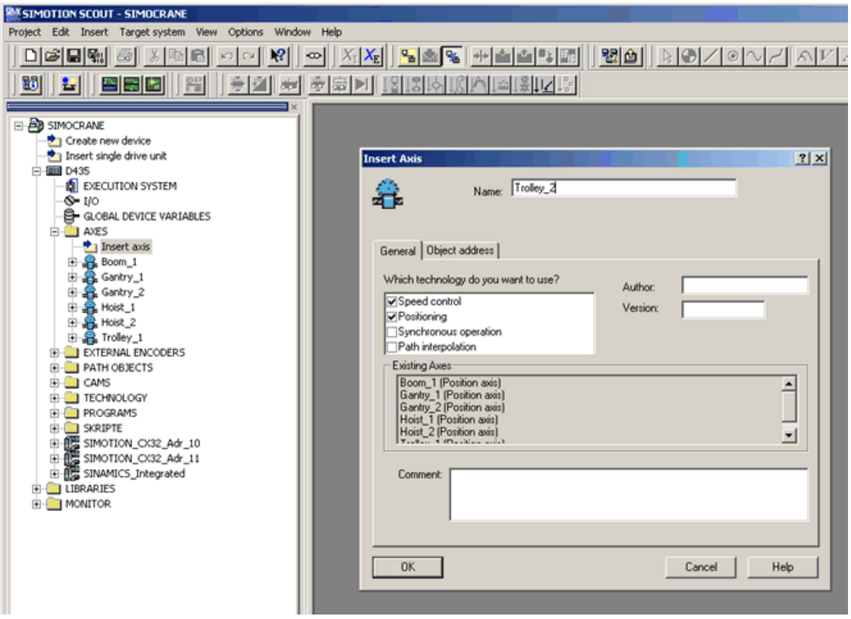
(10) SIMOTION_CX32_Adr_10

Slot	M...	Message frame selection / default	I address	O address
4	Drive	User-defined	400...449	
5	Drive	User-defined		400...449
6	Drive			
7	Drive	User-defined	450...499	
8	Drive	User-defined		450...499
9	Drive			
10				

Figure 9-2 Inserting addresses for Trolley_2

9.2 Inserting a technology object and a fixed gear

In the next step you can insert and configure the technology object Trolley_2. Then you can connect Trolley_1 to Trolley_2 via a fixed gear.



Insert Axis

Name: Trolley_2

General | Object address

Which technology do you want to use?
☑ Speed control
☑ Positioning
☐ Synchronous operation
☐ Path interpolation

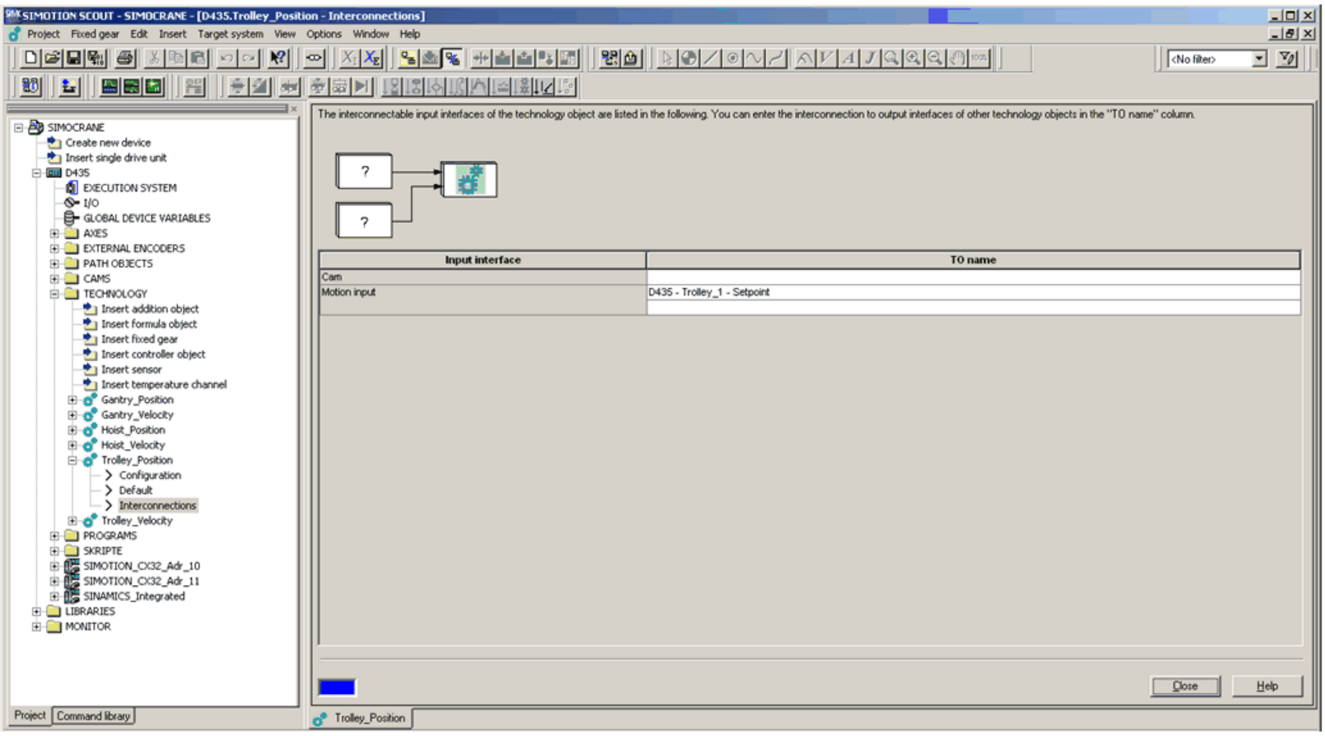
Author:
Version:

Existing Axes
Boom_1 (Position axis)
Gantry_1 (Position axis)
Gantry_2 (Position axis)
Hoist_1 (Position axis)
Hoist_2 (Position axis)

Comment:

OK | Cancel | Help

Figure 9-3 Inserting axis Trolley_2



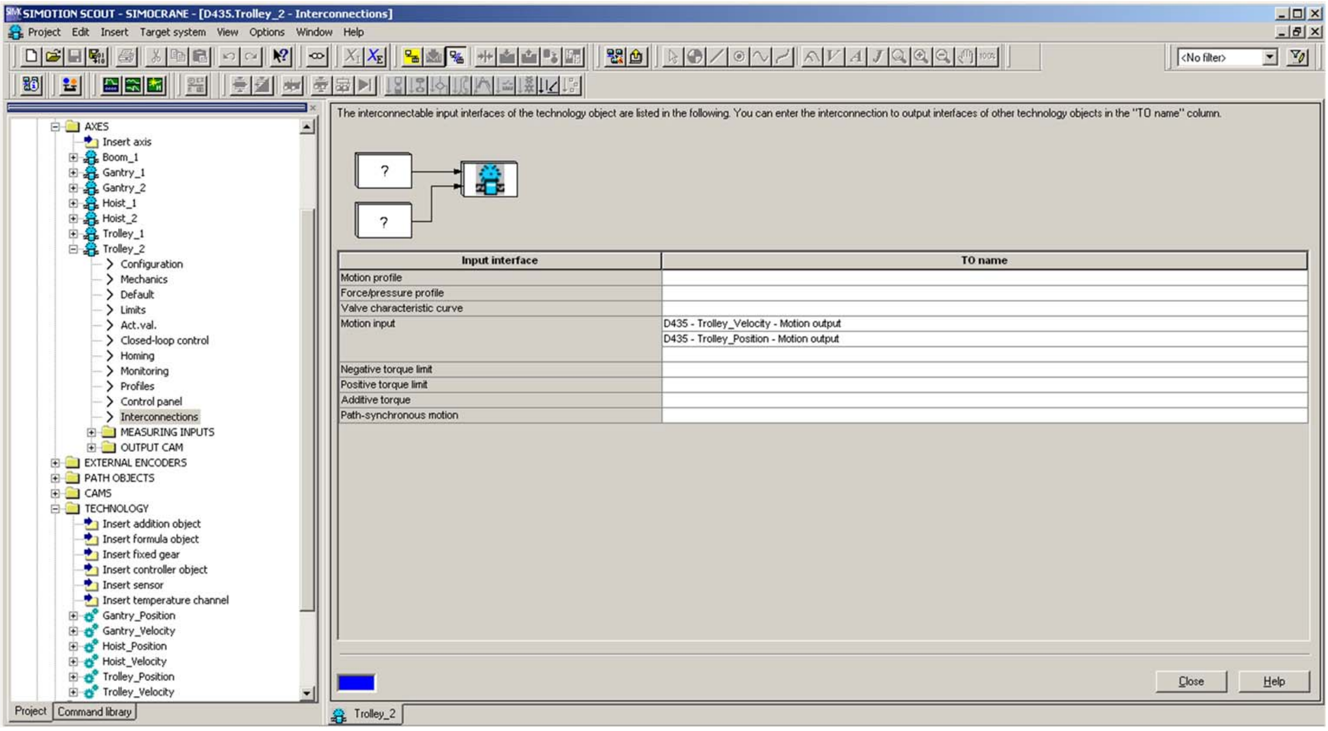
The interconnectable input interfaces of the technology object are listed in the following. You can enter the interconnection to output interfaces of other technology objects in the "TO name" column.

Input interface	TO name
Cam	
Motion input	D435 - Trolley_1 - Setpoint

Figure 9-4 Fixed gear for synchronous operation

Set up the gears "Trolley_Position" and "Trolley_Velocity" for synchronous operation between Trolley_1 and Trolley_2. In the configuration, only check "Ignore position" for the "Trolley_Velocity" gear.

Then connect the gears to the motion input of the slave axis (Trolley_2 in this case).



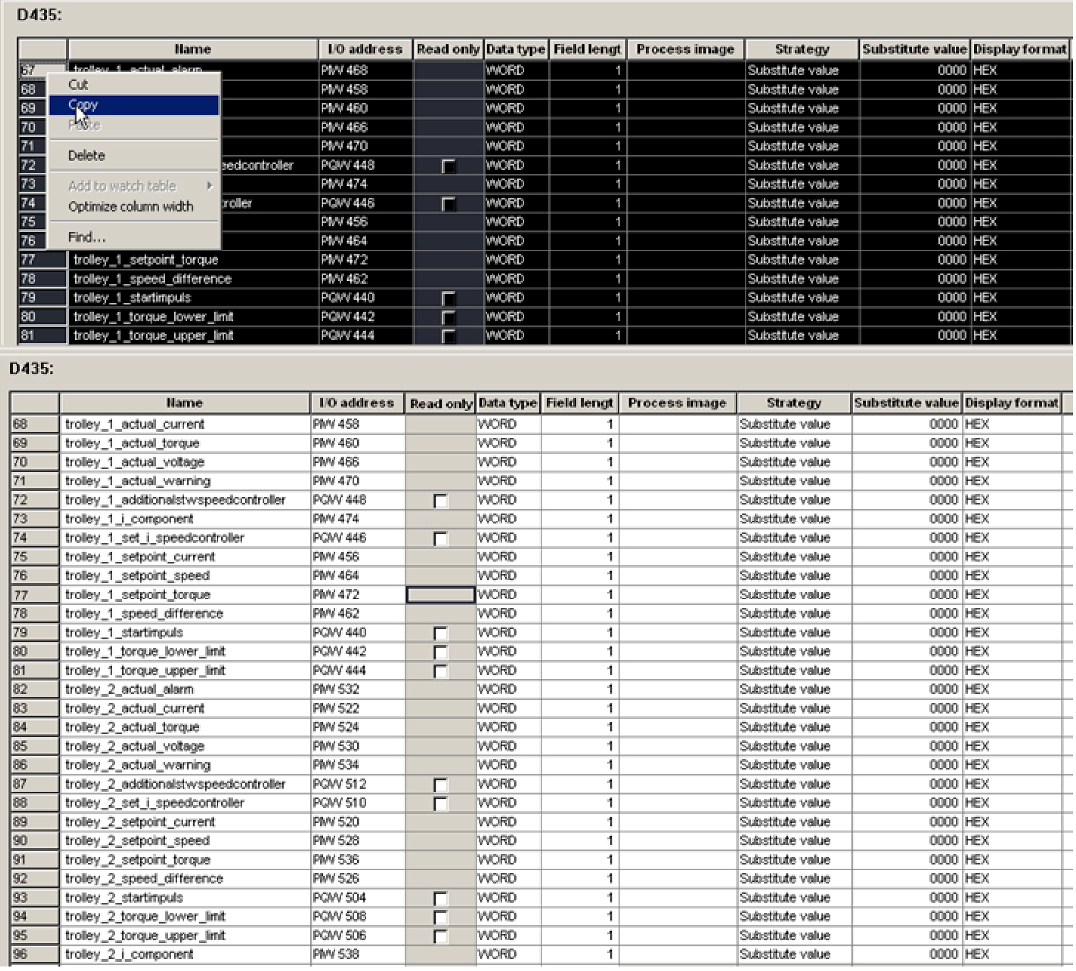
The interconnectable input interfaces of the technology object are listed in the following. You can enter the interconnection to output interfaces of other technology objects in the "TO name" column.

Input interface	TO name
Motion profile	
Force/pressure profile	
Valve characteristic curve	
Motion input	D435 - Trolley_Velocity - Motion output D435 - Trolley_Position - Motion output
Negative torque limit	
Positive torque limit	
Additive torque	
Path-synchronous motion	

Figure 9-5 Gear connection to slave

9.3 Creating input and output addresses

For SIMOTION – SINAMICS and S7 – SIMOTION communication, you still need to create variables in the I/O container. Select and copy all variables called "trolley_1_…", paste them in, and rename them "trolley_2_…". Modify the addresses by adding 64 to them.



D435:

	Name	I/O address	Read only	Data type	Field lengt	Process image	Strategy	Substitute value	Display format
67	trolley_1_actual_alarm	PIW 468		WORD	1		Substitute value	0000	HEX
68		PIW 458		WORD	1		Substitute value	0000	HEX
69		PIW 460		WORD	1		Substitute value	0000	HEX
70		PIW 466		WORD	1		Substitute value	0000	HEX
71		PIW 470		WORD	1		Substitute value	0000	HEX
72	eedcontroller	PQW 448	□	WORD	1		Substitute value	0000	HEX
73		PIW 474		WORD	1		Substitute value	0000	HEX
74	roller	PQW 446	□	WORD	1		Substitute value	0000	HEX
75		PIW 456		WORD	1		Substitute value	0000	HEX
76		PIW 464		WORD	1		Substitute value	0000	HEX
77	trolley_1_setpoint_torque	PIW 472		WORD	1		Substitute value	0000	HEX
78	trolley_1_speed_difference	PIW 462		WORD	1		Substitute value	0000	HEX
79	trolley_1_startimpuls	PQW 440	□	WORD	1		Substitute value	0000	HEX
80	trolley_1_torque_lower_limit	PQW 442	□	WORD	1		Substitute value	0000	HEX
81	trolley_1_torque_upper_limit	PQW 444	□	WORD	1		Substitute value	0000	HEX

D435:

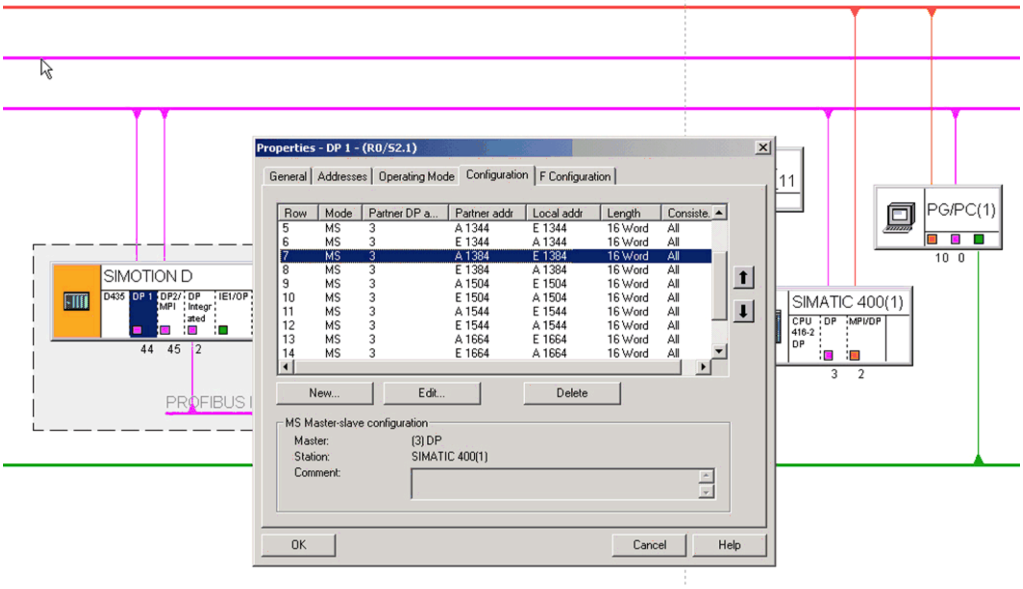
	Name	I/O address	Read only	Data type	Field lengt	Process image	Strategy	Substitute value	Display format
68	trolley_1_actual_current	PIW 458		WORD	1		Substitute value	0000	HEX
69	trolley_1_actual_torque	PIW 460		WORD	1		Substitute value	0000	HEX
70	trolley_1_actual_voltage	PIW 466		WORD	1		Substitute value	0000	HEX
71	trolley_1_actual_warning	PIW 470		WORD	1		Substitute value	0000	HEX
72	trolley_1_additionalstwspeedcontroller	PQW 448	□	WORD	1		Substitute value	0000	HEX
73	trolley_1_i_component	PIW 474		WORD	1		Substitute value	0000	HEX
74	trolley_1_set_i_speedcontroller	PQW 446	□	WORD	1		Substitute value	0000	HEX
75	trolley_1_setpoint_current	PIW 456		WORD	1		Substitute value	0000	HEX
76	trolley_1_setpoint_speed	PIW 464		WORD	1		Substitute value	0000	HEX
77	trolley_1_setpoint_torque	PIW 472		WORD	1		Substitute value	0000	HEX
78	trolley_1_speed_difference	PIW 462		WORD	1		Substitute value	0000	HEX
79	trolley_1_startimpuls	PQW 440	□	WORD	1		Substitute value	0000	HEX
80	trolley_1_torque_lower_limit	PQW 442	□	WORD	1		Substitute value	0000	HEX
81	trolley_1_torque_upper_limit	PQW 444	□	WORD	1		Substitute value	0000	HEX
82	trolley_2_actual_alarm	PIW 532		WORD	1		Substitute value	0000	HEX
83	trolley_2_actual_current	PIW 522		WORD	1		Substitute value	0000	HEX
84	trolley_2_actual_torque	PIW 524		WORD	1		Substitute value	0000	HEX
85	trolley_2_actual_voltage	PIW 530		WORD	1		Substitute value	0000	HEX
86	trolley_2_actual_warning	PIW 534		WORD	1		Substitute value	0000	HEX
87	trolley_2_additionalstwspeedcontroller	PQW 512	□	WORD	1		Substitute value	0000	HEX
88	trolley_2_set_i_speedcontroller	PQW 510	□	WORD	1		Substitute value	0000	HEX
89	trolley_2_setpoint_current	PIW 520		WORD	1		Substitute value	0000	HEX
90	trolley_2_setpoint_speed	PIW 528		WORD	1		Substitute value	0000	HEX
91	trolley_2_setpoint_torque	PIW 536		WORD	1		Substitute value	0000	HEX
92	trolley_2_speed_difference	PIW 526		WORD	1		Substitute value	0000	HEX
93	trolley_2_startimpuls	PQW 504	□	WORD	1		Substitute value	0000	HEX
94	trolley_2_torque_lower_limit	PQW 508	□	WORD	1		Substitute value	0000	HEX
95	trolley_2_torque_upper_limit	PQW 506	□	WORD	1		Substitute value	0000	HEX
96	trolley_2_i_component	PIW 538		WORD	1		Substitute value	0000	HEX

Figure 9-6 Variables for Trolley_2

The 16 process data items used for exchanging data between the S7 and SIMOTION must be set up in the I/O container (see Configuring the connection (Page 297)).

Note

Observe the following information when inserting variables into the I/O Container: The newly inserted variables must be assigned to the strategy "Substitute value". Furthermore, the addresses for S7 – SIMOTION communication also have to be set in NetPro. The address 1384 is reserved for Trolley_2.



Properties - DP 1 - (R0/S2.1)

General | Addresses | Operating Mode | Configuration | F Configuration

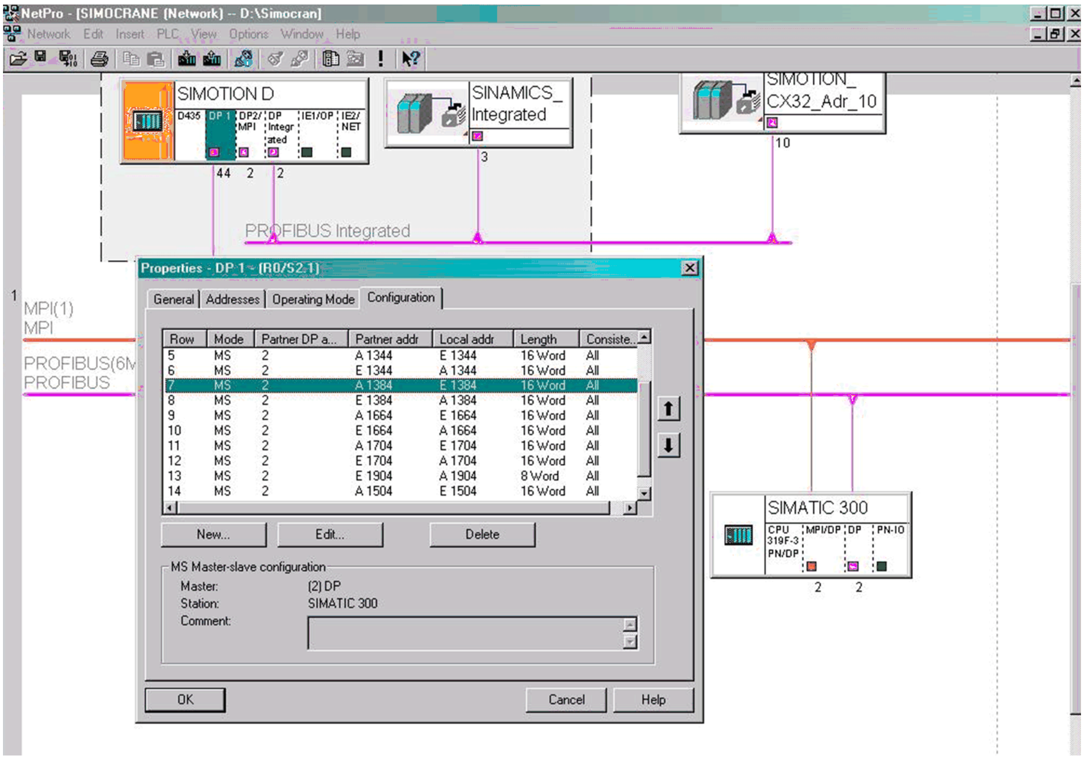
Row	Mode	Partner DP a...	Partner addr	Local addr	Length	Consiste.
5	MS	3	A 1344	E 1344	16 Word	All
6	MS	3	E 1344	A 1344	16 Word	All
7	MS	3	A 1384	E 1384	16 Word	All
8	MS	3	E 1384	A 1384	16 Word	All
9	MS	3	A 1504	E 1504	16 Word	All
10	MS	3	E 1504	A 1504	16 Word	All
11	MS	3	A 1544	E 1544	16 Word	All
12	MS	3	E 1544	A 1544	16 Word	All
13	MS	3	A 1664	E 1664	16 Word	All
14	MS	3	E 1664	A 1664	16 Word	All

New... | Edit... | Delete

MS Master-slave configuration
Master: (3) DP
Station: SIMATIC 400(1)
Comment:

OK | Cancel | Help

Figure 9-7 Addresses in NetPro for SIMATIC S7-400



Properties - DP 1 - (R0/S2.1)

General | Addresses | Operating Mode | Configuration

Row	Mode	Partner DP a...	Partner addr	Local addr	Length	Consiste...
5	MS	2	A 1344	E 1344	16 Word	All
6	MS	2	E 1344	A 1344	16 Word	All
7	MS	2	A 1384	E 1384	16 Word	All
8	MS	2	E 1384	A 1384	16 Word	All
9	MS	2	A 1664	E 1664	16 Word	All
10	MS	2	E 1664	A 1664	16 Word	All
11	MS	2	A 1704	E 1704	16 Word	All
12	MS	2	E 1704	A 1704	16 Word	All
13	MS	2	E 1904	A 1904	8 Word	All
14	MS	2	A 1504	E 1504	16 Word	All

New... | Edit... | Delete

MS Master-slave configuration
Master: (2) DP
Station: SIMATIC 300
Comment:

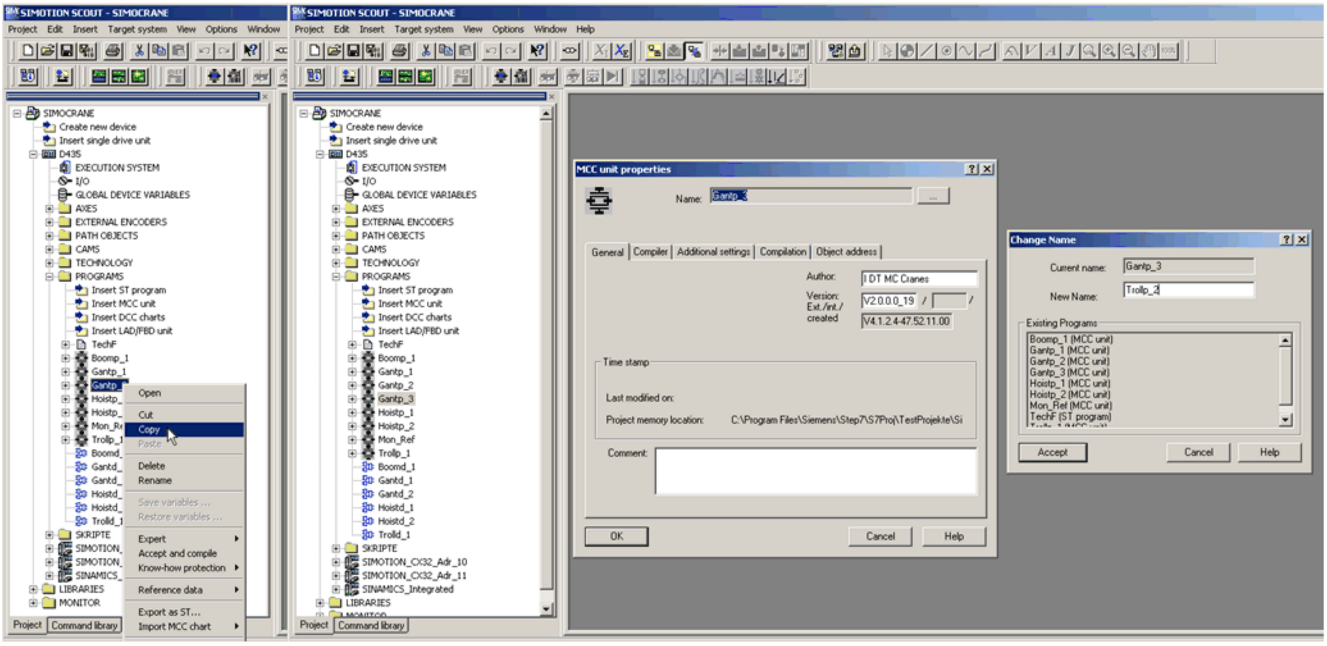
OK | Cancel | Help

Figure 9-8 Addresses in NetPro for SIMATIC S7-300

9.4 Setting up the program

9.4.1 MCC level

You must copy the MCC source Gantp_2 to set up the MCC program Trollp_2. The MCC unit Gantp_2 has the same functionality as the MCC unit Trollp_2.



MCC unit properties

Name: Gantp_3

General | Compiler | Additional settings | Compilation | Object address

Author: I DT MC Cranes

Version: Ext./int./ created: V2.0.0.0_19 / / V4.1.2.4-47.52.11.00

Time stamp

Last modified on:

Project memory location: C:\Program Files\Siemens\Step7\S7Proj\TestProjekte\Si

Comment:

OK | Cancel | Help

Change Name

Current name: Gantp_3

New Name: Trollp_2

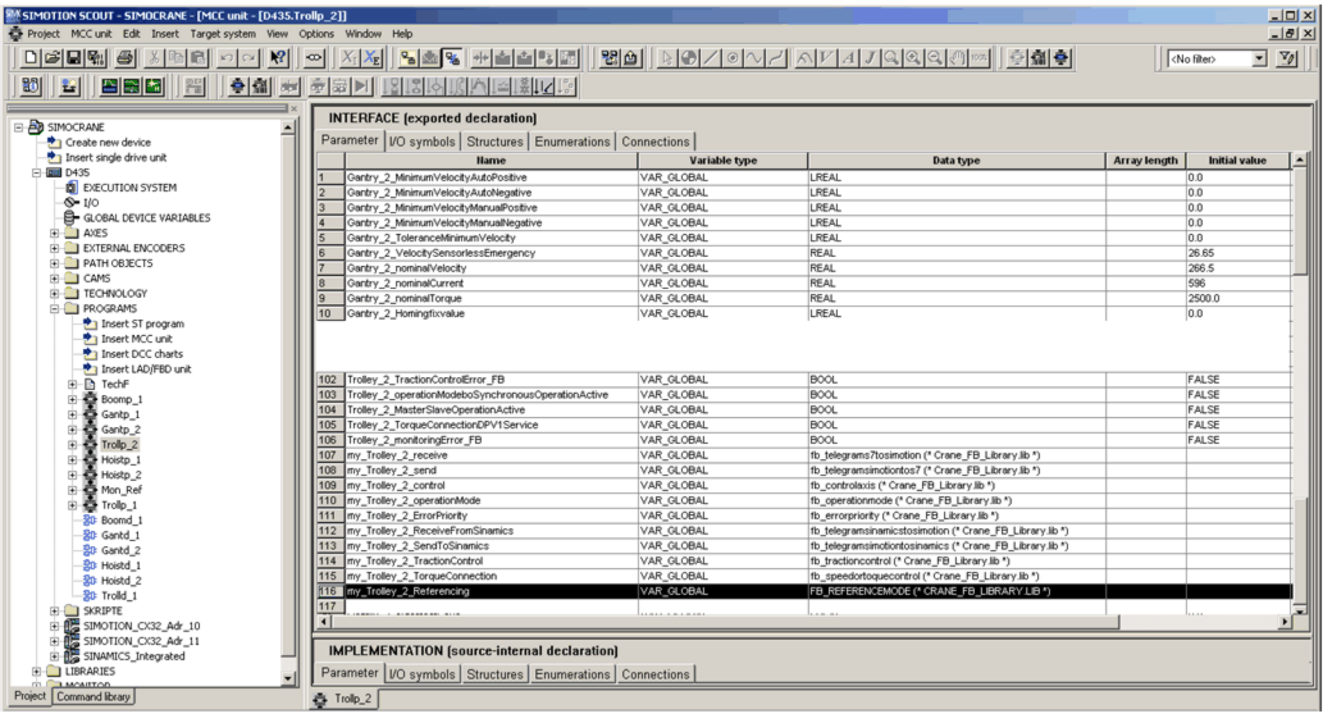
Existing Programs

Boomp_1 (MCC unit)
Gantp_1 (MCC unit)
Gantp_2 (MCC unit)
Gantp_3 (MCC unit)
Hoistp_1 (MCC unit)
Hoistp_2 (MCC unit)
Mon_Ref (MCC unit)
TechF (ST program)

Accept | Cancel | Help

Figure 9-9 Copying, pasting, and renaming the MCC program Gantry_2

- After copying, insert the MCC source under **Programs**.
- Under **MCC unit properties**, rename the MCC unit as "Trollp_2" and the MCC programs "gantrymcc_2_s_1" and "gantrymcc_2_f_1" as "trolleymcc_2_s" and "trolleymcc_2_f".
- Open the interface area of Trollp_2 and rename all "Gantry_2_xxx" variables as "Trolley_2_xxx".
- Add the global variable "my_Trolley_2_Referencing" of type "fb_referencemode(*Crane_FB_Library-lib*)" too.



INTERFACE (exported declaration)

Parameter | I/O symbols | Structures | Enumerations | Connections

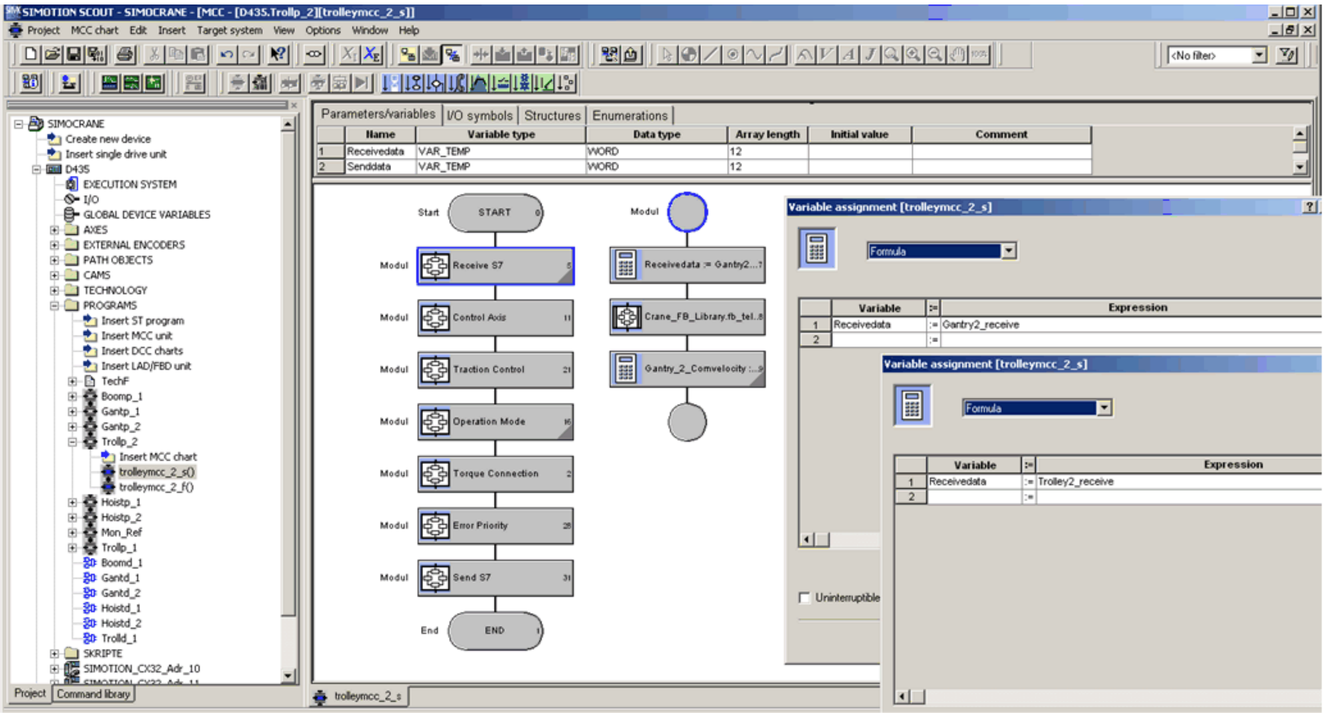
	Name	Variable type	Data type	Array length	Initial value
1	Gantry_2_MinimumVelocityAutoPositive	VAR_GLOBAL	LREAL		0.0
2	Gantry_2_MinimumVelocityAutoNegative	VAR_GLOBAL	LREAL		0.0
3	Gantry_2_MinimumVelocityManualPositive	VAR_GLOBAL	LREAL		0.0
4	Gantry_2_MinimumVelocityManualNegative	VAR_GLOBAL	LREAL		0.0
5	Gantry_2_ToleranceMinimumVelocity	VAR_GLOBAL	LREAL		0.0
6	Gantry_2_VelocitySensorlessEmergency	VAR_GLOBAL	REAL		26.65
7	Gantry_2_nominalVelocity	VAR_GLOBAL	REAL		266.5
8	Gantry_2_nominalCurrent	VAR_GLOBAL	REAL		596
9	Gantry_2_nominalTorque	VAR_GLOBAL	REAL		2500.0
10	Gantry_2_Homingfixvalue	VAR_GLOBAL	LREAL		0.0
102	Trolley_2_TractionControlError_FB	VAR_GLOBAL	BOOL		FALSE
103	Trolley_2_operationModeboSynchronousOperationActive	VAR_GLOBAL	BOOL		FALSE
104	Trolley_2_MasterSlaveOperationActive	VAR_GLOBAL	BOOL		FALSE
105	Trolley_2_TorqueConnectionDPV1Service	VAR_GLOBAL	BOOL		FALSE
106	Trolley_2_monitoringError_FB	VAR_GLOBAL	BOOL		FALSE
107	my_Trolley_2_receive	VAR_GLOBAL	fb_telegrams7tosimotion (* Crane_FB_Library.lib *)		
108	my_Trolley_2_send	VAR_GLOBAL	fb_telegramsimotiontos7 (* Crane_FB_Library.lib *)		
109	my_Trolley_2_control	VAR_GLOBAL	fb_controlaxis (* Crane_FB_Library.lib *)		
110	my_Trolley_2_operationMode	VAR_GLOBAL	fb_operationmode (* Crane_FB_Library.lib *)		
111	my_Trolley_2_ErrorPriority	VAR_GLOBAL	fb_errorpriority (* Crane_FB_Library.lib *)		
112	my_Trolley_2_ReceiveFromSinamics	VAR_GLOBAL	fb_telegramsinamicstosimotion (* Crane_FB_Library.lib *)		
113	my_Trolley_2_SendToSinamics	VAR_GLOBAL	fb_telegramsimotiontosinamics (* Crane_FB_Library.lib *)		
114	my_Trolley_2_TractionControl	VAR_GLOBAL	fb_tractioncontrol (* Crane_FB_Library.lib *)		
115	my_Trolley_2_TorqueConnection	VAR_GLOBAL	fb_speedortoquecontrol (* Crane_FB_Library.lib *)		
116	my_Trolley_2_Referencing	VAR_GLOBAL	FB_REFERENCEMODE (* CRANE_FB_LIBRARY.LIB *)		
117					

IMPLEMENTATION (source-internal declaration)

Parameter | I/O symbols | Structures | Enumerations | Connections

Figure 9-10 Renaming interface variables

- Open the MCC program "trolleymcc_2_s", the module "Receive S7", and then the **Variable assignment**. Change all variable names from "Gantry_2_xxx" to "Trolley_2_xxx".
- This also needs to be done for all other user programs and instances of function blocks.



	Name	Variable type	Data type	Array length	Initial value	Comment
1	Receivedata	VAR_TEMP	WORD	12		
2	Senddata	VAR_TEMP	WORD	12		

Variable assignment [trolleymcc_2_s]

Formula

	Variable	:=	Expression
1	Receivedata	:=	Gantry2_receive
2		:=	

□ Uninterruptible

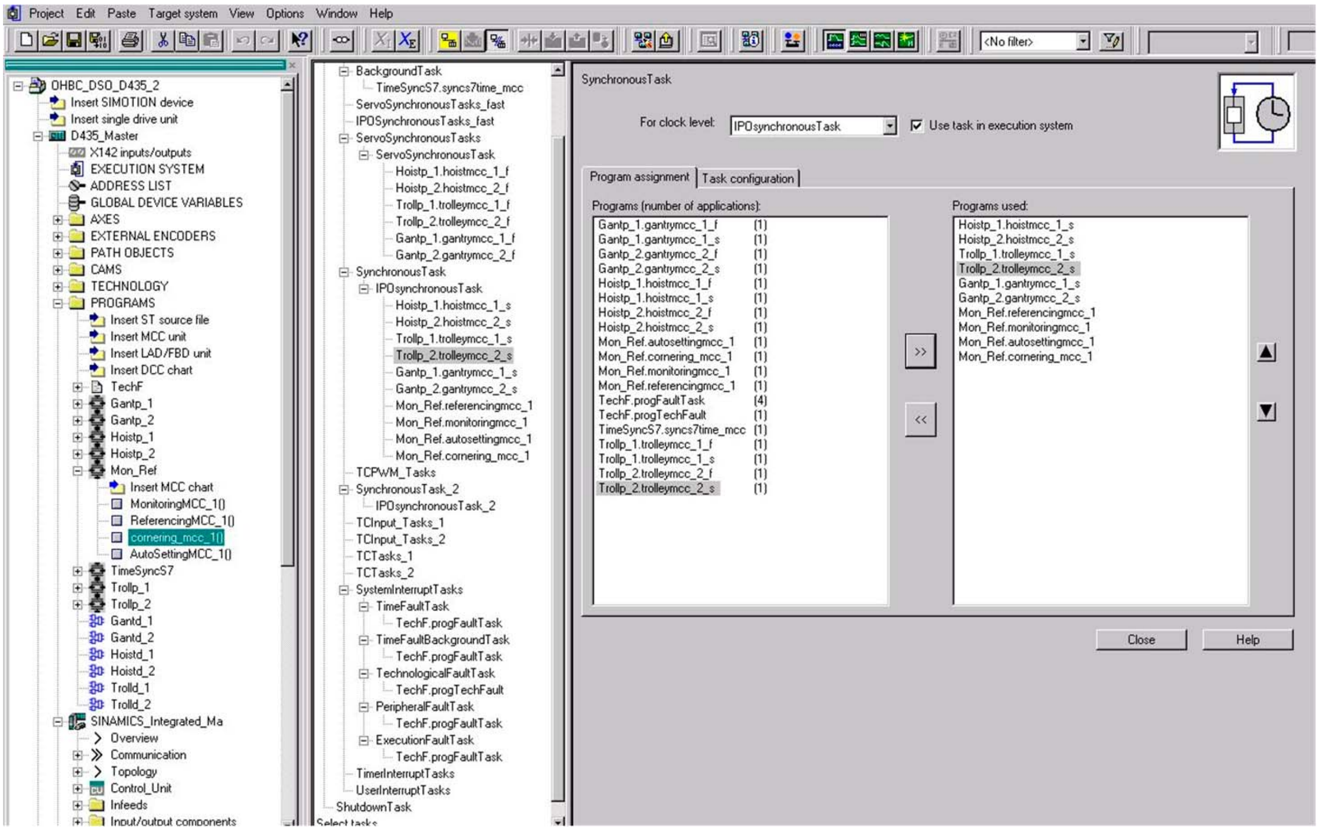
Variable assignment [trolleymcc_2_s]

Formula

	Variable	:=	Expression
1	Receivedata	:=	Trolley2_receive
2		:=	

Figure 9-11 Changing variable names

- Finally, link the user program "Trollp_2" to the execution system. This means that all user programs are executed in a cyclic task. The Trollp_2.trolleymcc_2_f and Trollp_2.trolleymcc_2s programs are called in different tasks. Trollp_2.trolleymcc_2_f must be processed twice as fast as Trollp_2.trolleymcc_2_s.
- Trollp_2.trolleymcc_2_s and Trollp_2.trolleymcc_2_f in the execution system



SynchronousTask

For clock level: IPOsynchronousTask ☑ Use task in execution system

Programs (number of applications):

Gantp_1.gantrymcc_1_f	(1)
Gantp_1.gantrymcc_1_s	(1)
Gantp_2.gantrymcc_2_f	(1)
Gantp_2.gantrymcc_2_s	(1)
Hoistp_1.hoistmcc_1_f	(1)
Hoistp_1.hoistmcc_1_s	(1)
Hoistp_2.hoistmcc_2_f	(1)
Hoistp_2.hoistmcc_2_s	(1)
Mon_Ref.autosettingmcc_1	(1)
Mon_Ref.cornering_mcc_1	(1)
Mon_Ref.monitoringmcc_1	(1)
Mon_Ref.referencingmcc_1	(1)
TechF.progFaultTask	(4)
TechF.progTechFault	(1)
TimeSyncS7.syncs7time_mcc	(1)
Trollp_1.trolleymcc_1_f	(1)
Trollp_1.trolleymcc_1_s	(1)
Trollp_2.trolleymcc_2_f	(1)
Trollp_2.trolleymcc_2_s	(1)

Programs used:

Hoistp_1.hoistmcc_1_s
Hoistp_2.hoistmcc_2_s
Trollp_1.trolleymcc_1_s
Trollp_2.trolleymcc_2_s
Gantp_1.gantrymcc_1_s
Gantp_2.gantrymcc_2_s
Mon_Ref.referencingmcc_1
Mon_Ref.monitoringmcc_1
Mon_Ref.autosettingmcc_1
Mon_Ref.cornering_mcc_1

Figure 9-12 Example: Trollp_2.trolleymcc_2_s and Trollp_2.trolleymcc_2_f in the execution system

9.4.2 DCC level

Note

If, in the case of the DCC block, the inputs and outputs which are not interconnected need to be visible in the symbol browser of SIMOTION SCOUT, they must be published by inserting an @ character in the comment line of the required input or output (Ref. [10], Chapter 3.1.3.4).

Proceed as follows for DCC to generate and integrate the Trolley_2 DCC chart:

- Select "Gantd_2" and click on it with the right-hand mouse key.
- Select **Copy** from the context menu.
- Select **Programs** and click on it with the right-hand mouse key.

- Select **Paste** from the context menu.
- Select the newly added chart "Gantd_3" and click on it with the right-hand mouse key.
- Select **Rename**.
- Rename "Gantd_3" as "Trolld_2".

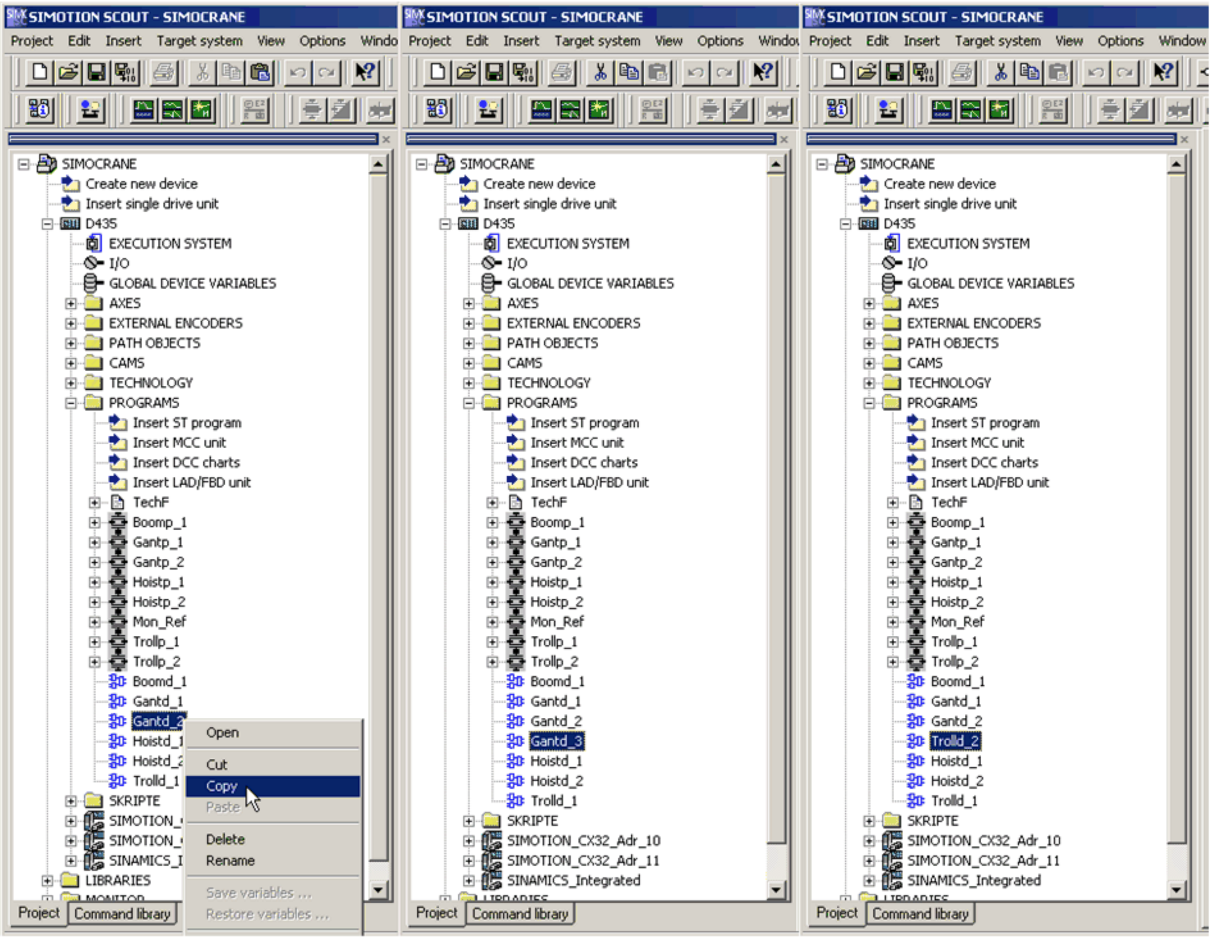
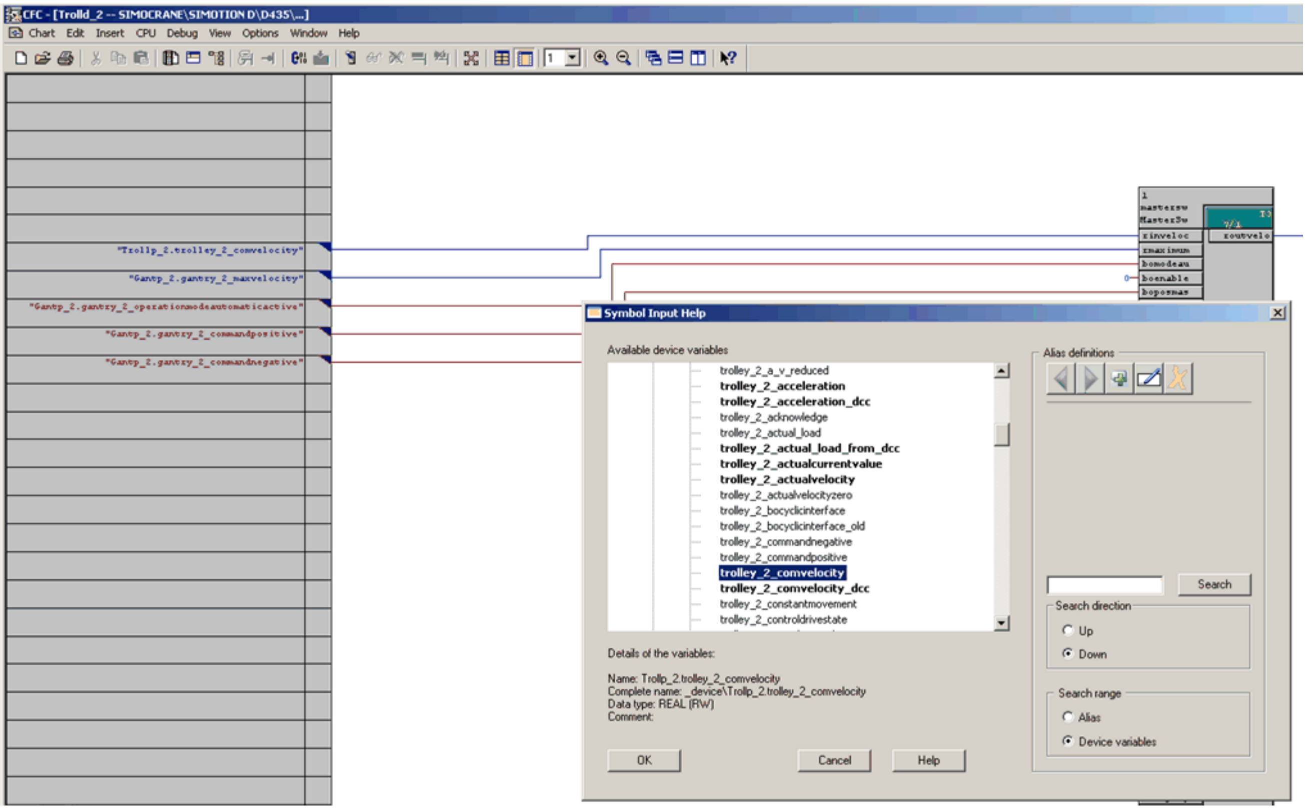


Figure 9-13 Create the DCC Trolld_2 (Copy, Paste und Rename)

- Open the DCC chart "Trollyd_2"

All variables of the blocks must be newly linked with the MCC program Trollp_2 as these still have a current link to Gantp_2

- Double click on the variable.
- Under **Programs** select the particular program, in this case "Trollp_2".
- Then search for the corresponding variable that is to be replaced. In the diagram, for example "Trollp_2.trolley_2_comvelocity".



"Trollp_2.trolley_2_comvelocity"
"Gantp_2.gantry_2_maxvelocity"
"Gantp_2.gantry_2_operationmodeautomaticactive"
"Gantp_2.gantry_2_commandpositive"
"Gantp_2.gantry_2_commandnegative"

trolley_2_a_v_reduced
trolley_2_acceleration
trolley_2_acceleration_dcc
trolley_2_acknowledge
trolley_2_actual_load
trolley_2_actual_load_from_dcc
trolley_2_actualcurrentvalue
trolley_2_actualvelocity
trolley_2_actualvelocityzero
trolley_2_bocyclicinterface
trolley_2_bocyclicinterface_old
trolley_2_commandnegative
trolley_2_commandpositive
trolley_2_comvelocity
trolley_2_comvelocity_dcc
trolley_2_constantmovement
trolley_2_controldrivestate

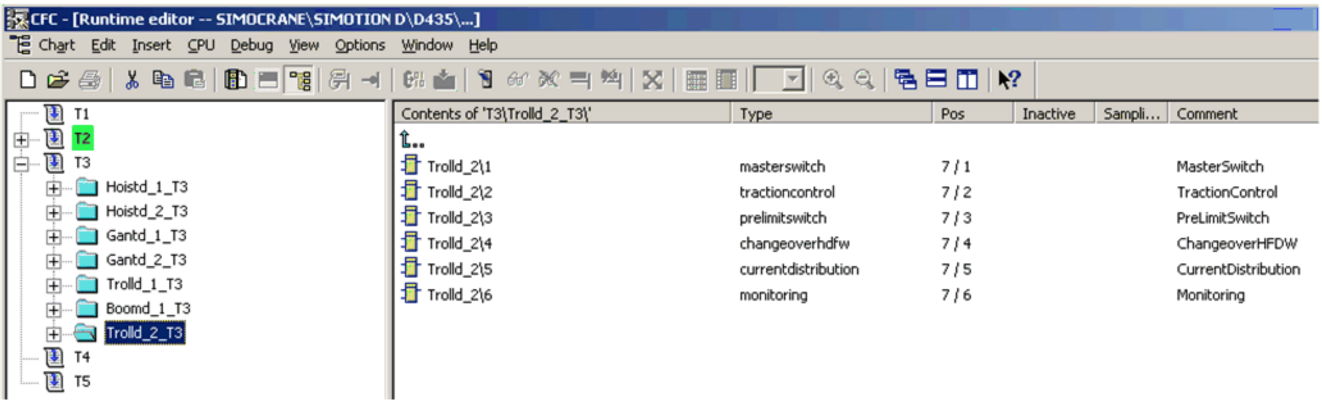
Details of the variables:

Name: Trollp_2.trolley_2_comvelocity
Complete name: _device\Trollp_2.trolley_2_comvelocity
Data type: REAL (RW)
Comment:

Figure 9-14 Example: Linking variables in the DCC chart of Trolld_2

Proceed in the same way with all variables that were taken from the MCC program Gantp_2. You can identify the MCC variables in the DCC by the fact that they have a blue triangle by the variable name.

After having interconnected the I/O of the DCC blocks with the correct MCC variable, check the runtime system to determine whether all DCC blocks were integrated in time slice T3 in the DCC chart folder of Trolld_2.


CFC - [Runtime editor -- SIMOCRANE\SIMOTION D\D435\...]
Chart Edit Insert CPU Debug View Options Window Help
T1
T2
T3
Hoistd_1_T3
Hoistd_2_T3
Gantd_1_T3
Gantd_2_T3
Trolld_1_T3
Boomd_1_T3
Trolld_2_T3
T4
T5

Contents of 'T3\Trolld_2_T3\'	Type	Pos	Inactive	Sampli...	Comment
t..					
Trolld_2\1	masterswitch	7 / 1			MasterSwitch
Trolld_2\2	tractioncontrol	7 / 2			TractionControl
Trolld_2\3	prelimitswitch	7 / 3			PreLimitSwitch
Trolld_2\4	changeoverhdfw	7 / 4			ChangeoverHFDW
Trolld_2\5	currentdistribution	7 / 5			CurrentDistribution
Trolld_2\6	monitoring	7 / 6			Monitoring

Figure 9-15 View of the runtime system in the DCC of Trolld_2

Finally, the **Compile** button from the symbol bar must be selected. The **Compile** window opens. In this window, click on the button **OK**.

Note

You can also open another standard project, which contains Trolley2; then first copy Trollp_2 and then Trolld_2.

Standard applications 10

The standard applications are arranged according to crane type. This means that we are offering an initial basis both for ready-to-run and ready-to-apply applications.

Linking the digital output signals standard application

Table 10- 1 Linking the digital output signals in SIMOTION

Drives	Terminal	CU output	Signal
Hoist	-X142.3	DO 0	Hoist 1&2 velocity less than 70%
Trolley	-X142.6	DO 2	Trolley velocity less than 70%
Boom	-X142.9	DO 4	Boom velocity less than 70%

Table 10- 2 Linking the digital output signals in SINAMICS Integrated CU

Drives	Terminal	CU output	Signal
Hoist 1 / Holding Gear 1	-X122.9	DO 8	Hoist 1 / Holding Gear 1: Pulses enabled
Hoist 1 / Holding Gear 1	-X122.10	DO 9	Hoist 1 / Holding Gear 1: Intelligent overspeed
Hoist 2 / Closing Gear 1	-X122.12	DO 10	Hoist 2 / Closing Gear 1: Pulses enabled
Hoist 2 / Closing Gear 1	-X122.13	DO 11	Hoist 2 / Closing Gear 1: Intelligent overspeed
Boom	-X132.9	DO 12	Boom: Pulses enabled
Boom	-X132.10	DO 13	Boom: Intelligent overspeed

Linking the digital output signals tandem standard application

Table 10- 3 Linking the tandem digital output signals in SIMOTION

Drives	Terminal	CU output	Signal
Hoist (tandem master)	-X142.3	DO 0	Hoist 1&2 velocity less than 70%
Hoist (tandem slave)	-X142.4	DO 1	Hoist 3&4 velocity less than 70%
Trolley	-X142.6	DO 2	Trolley velocity less than 70%
Boom	-X142.9	DO 4	Boom velocity less than 70%

Table 10- 4 Linking the tandem digital output signals in SINAMICS Integrated CU

Drives	Terminal	CU output	Signal
Hoist 1 / Holding Gear 1	-X122.9	DO 8	Hoist 1 / Holding Gear 1: Pulses enabled
Hoist 1 / Holding Gear 1	-X122.10	DO 9	Hoist 1 / Holding Gear 1: Intelligent overspeed
Hoist 2 / Holding Gear 2	-X122.12	DO 10	Hoist 2 / Holding Gear 2: Pulses enabled
Hoist 2 / Holding Gear 2	-X122.13	DO 11	Hoist 2 / Holding Gear 2: Intelligent overspeed
Hoist 3 / Closing Gear 1	-X132.9	DO 12	Hoist 3 / Closing Gear 1: Pulses enabled
Hoist 3 / Closing Gear 1	-X132.10	DO 13	Hoist 3 / Closing Gear 1: Intelligent overspeed
Hoist 4 / Closing Gear 2	-X132.12	DO 14	Hoist 4 / Closing Gear 2: Pulses enabled
Hoist 4 / Closing Gear 2	-X132.13	DO 15	Hoist 4 / Closing Gear 2: Intelligent overspeed

Table 10- 5 Linking the tandem digital output signals on SINAMICS CX32-2 CU

Drives	Terminal	CU output	Signal
Boom	-X122.12	DO 10	Boom: Pulses enabled
Boom	-X122.13	DO 11	Boom: Intelligent overspeed

10.1 STS crane (ship-to-shore)

The functionality and scope of the standard application for the STS Crane type are grouped as follows.

There are three variants of this standard application:

- STS Crane
- STS Crane DCM

  STS Crane DCM is intended for hoisting gear with DC motors and AC current on the other axes
- STS Crane Backup

  STS Crane Backup is based on the use of a single backup converter and only one trolley drive.

Quantity structure

Function module	Number of axes	Control mode	Operating mode	Switchover
Hoist	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation	• AUTOMATIC • MANUAL • SPEED_CONTROLLED • SWAYCONTROL • EASY_POSITIONING	All drives (application)
Trolley	2 (1 for STS Crane Backup)	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation (For STS Crane Backup only "closed-loop speed controlled" and "positioning" because only one trolley)	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • SWAYCONTROL • EASY_POSITIONING	Boom (standard) All drives (application)
Gantry	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	Boom (application) All drives (application)
Boom	1	1. Speed-controlled	• SPEED_CONTROLLED	Trolley (standard) All drives (application)

Note

"Standard" for switching means: The user does not need to become active explicitly.

"Application" for switching means: The user must create the drive data sets and determine the selection using the function modules.

See Technology object switchover (Page 214), "External TO switchover" section

Application of crane-specific technology

Function module	Crane-specific technology
Hoist	• Time-optimized positioning • Basic positioning • Master-slave closed-loop torque control • Synchronous operation • Load-dependent field weakening • Start pulse • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Optional: Advanced Technology (Sway Control, 2D trajectory) • Brake test
Trolley	• Time-optimized positioning • Basic positioning • Master-slave closed-loop torque control (except for "STS Crane Backup" because only one trolley) • Synchronous operation with offset compensatory control (except for "STS Crane Backup" because only one trolley) • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Optional: Advanced Technology (Sway Control, 2D trajectory) • Flying homing • Brake test

Function module	Crane-specific technology
Gantry	• Time-optimized positioning • Basic positioning • Master-slave closed-loop torque control • Synchronous operation with offset compensatory control • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Cornering movement • Flying homing
Boom	• Non-linear setpoint of the master switch • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Start pulse • Brake test

Description of functions

For information about each function module, refer to Chapter Function modules (Page 37) .

Commissioning

For information on commissioning the standard application, refer to Chapter Commissioning (Page 351)

10.2 RMG Crane (Rail-mounted Gantry)

Quantity structure

Function module	Number of axes	Control mode	Operating mode	Switchover
Hoist	1	1. Speed-controlled 2. Positioning	• AUTOMATIC • MANUAL • SPEED_CONTROLLED • SWAYCONTROL • EASY_POSITIONING	
Trolley	1	1. Speed-controlled 2. Positioning	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • SWAYCONTROL • EASY_POSITIONING	
Gantry	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	
Slewing Gear	1	1. Speed-controlled 2. Positioning	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	

Application of crane-specific technology

Function module	Crane-specific technology
Hoist	• Time-optimized positioning • Load-dependent field weakening • Basic positioning • Start pulse • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Optional: Advanced Technology (Sway Control, 2D trajectory) • Brake test
Trolley	• Time-optimized positioning • Basic positioning • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Optional: Advanced Technology (Sway Control, 2D trajectory) • Flying homing • Brake test

Function module	Crane-specific technology
Gantry	• Time-optimized positioning • Basic positioning • Master-slave closed-loop torque control • Synchronous operation with offset compensatory control • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Flying homing • Cornering movement
Slewing Gear	• Time-optimized positioning • Basic positioning • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Velocity dependent on working radius • Ramp-up / ramp-down time dependent on working radius and velocity • Brake test

Description of functions

For information about each function module, refer to Chapter Function modules (Page 37)

Commissioning

For information on commissioning the standard application, refer to Chapter Commissioning (Page 351)

10.3 RTG crane (Rubber-Tired Gantry)

There are two versions of this standard application:

- RTG Crane
- RTG Crane virtual S7

  No real S7 exists for the RTG Crane virtual S7 standard application; program in the SIMOTION performs the control.

Quantity structure

Function module	Number of axes	Control mode	Operating mode	Switchover
Hoist	1	1. Speed-controlled 2. Positioning	• AUTOMATIC • MANUAL • SPEED_CONTROLLED • EASY_POSITIONING	
Trolley	1	1. Speed-controlled 2. Positioning	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	
Gantry	4	1. Speed-controlled	• SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	

Application of crane-specific technology

Function module	Crane-specific technology
Hoist	• Time-optimized positioning • Basic positioning • Load-dependent field weakening • Start pulse • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Brake test
Trolley	• Time-optimized positioning • Basic positioning • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Flying homing • Brake test
Gantry	• Time-optimized positioning • Basic positioning • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Non-linear setpoint of the master switch • Prelimit switch

Description of functions

For information about each function module, refer to Chapter Function modules (Page 37)

Commissioning

For information on commissioning the standard application, refer to Chapter Commissioning (Page 351)

10.4 GSU (grab ship unloader) crane

There are two versions of this standard application:

- GSU Crane
- GSU Crane DCM

  GSU Crane DCM is intended for holding and closing gear with a DC machine.

Quantity structure

Function module	Number of axes	Control mode	Operating mode	Switchover
Holding Gear	1	1. Speed-controlled 2. Synchronous operation between holding gear and closing gear	• SPEED_CONTROLLED • SWAYCONTROL • EASY_POSITIONING	
Closing Gear	1	1. Speed-controlled 2. Synchronous operation between holding gear and closing gear	• SPEED_CONTROLLED • SWAYCONTROL	
Trolley	1	1. Speed-controlled 2. Positioning	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • SWAYCONTROL • EASY_POSITIONING	Boom (standard) All drives (application)
Long travel (Gantry)	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	All drives (application)
Boom	1	1. Speed-controlled	• SPEED_CONTROLLED	Trolley (standard) All drives (application)

Note

"Standard" for switching means: The user does not need to become active explicitly.

"Application" for switching means: The user must create the drive data sets and determine the selection using the function modules.

See Technology object switchover (Page 214), "External TO switchover" section

Application of crane-specific technology

Function module	Crane-specific technology
Holding Gear	• Time-optimized positioning • Basic positioning • Synchronous operation between holding gear and closing gear • Load-dependent field weakening • Constant field weakening • Continuous load measurement • Start pulse • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Slack rope control • Prelimit switch • Non-linear setpoint of the master switch • Optional: Advanced Technology (Sway Control, 2D trajectory) • Brake test
Closing Gear	• Time-optimized positioning • Synchronous operation between holding gear and closing gear • Constant field weakening • Start pulse • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Current equalization control • Prelimit switch • Non-linear setpoint of the master switch • Optional: Advanced Technology (Sway Control, 2D trajectory) • Brake test

Function module	Crane-specific technology
Trolley	• Time-optimized positioning • Basic positioning • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Optional: Advanced Technology (Sway Control, 2D trajectory) • Flying homing • Brake test
Long travel (Gantry)	• Time-optimized positioning • Basic positioning • Synchronous operation with offset compensatory control • Master-slave closed-loop torque control • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Flying homing • Cornering movement
Boom	• Non-linear setpoint of the master switch • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Start pulse • Brake test

Description of functions

For information about each function module, refer to Chapter Function modules (Page 37)

Commissioning

For information on commissioning the standard application, refer to Chapter Commissioning (Page 351)

Note

On the closing gear, only speed control is possible in grab operation. In hook operation, positioning is also supported.

10.5 LSC (Luffing Slewing Crane)

Quantity structure

Function module	Number of axes	Control mode	Operating mode	Switchover
Holding Gear	1	1. Speed-controlled 2. Synchronous operation between holding gear and closing gear	• SPEED_CONTROLLED • EASY_POSITIONING	
Closing Gear	1	1. Speed-controlled 2. Synchronous operation between holding gear and closing gear	• SPEED_CONTROLLED	
Slewing Gear	1	1. Speed-controlled 2. Positioning	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	
Luffing Gear (Boom)	1	1. Speed-controlled	• SPEED_CONTROLLED	
Gantry	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	

Application of crane-specific technology

Function module	Crane-specific technology
Holding Gear	• Time-optimized positioning • Basic positioning • Synchronous operation between holding gear and closing gear • Constant field weakening • Start pulse • Continuous load measurement • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Slack rope control • Prelimit switch • Non-linear setpoint of the master switch • Brake test
Closing Gear	• Time-optimized positioning • Synchronous operation between holding gear and closing gear • Constant field weakening • Start pulse • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Current equalization control • Prelimit switch • Non-linear setpoint of the master switch • Brake test
Slewing Gear	• Time-optimized positioning • Basic positioning • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Velocity dependent on working radius • Ramp-up / ramp-down time dependent on working radius and velocity • Brake test

Function module	Crane-specific technology
Luffing Gear (Boom)	• Load-dependent field weakening • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Velocity dependent on working radius • Ramp-up / ramp-down time dependent on working radius and velocity • Prelimit switch • Start pulse • Non-linear setpoint of the master switch • Brake test
Gantry	• Time-optimized positioning • Basic positioning • Master-slave closed-loop torque control • Synchronous operation with offset compensatory control • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Flying homing • Cornering movement

Description of functions

For information about each function module, refer to Chapter Function modules (Page 37)

Commissioning

For information on commissioning the standard application, refer to Chapter Commissioning (Page 351)

10.6 OHBC (Overhead Bridge Crane)

Quantity structure

Function module	Number of axes	Control mode	Operating mode	Switchover
Hoist	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation	• AUTOMATIC • MANUAL • SPEED_CONTROLLED • EASY_POSITIONING	
Trolley	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	
Gantry	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	

Application of crane-specific technology

Function module	Crane-specific technology
Hoist	• Time-optimized positioning • Basic positioning • Load-dependent field weakening • Start pulse • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Synchronous operation with offset mode • Brake test
Trolley	• Time-optimized positioning • Basic positioning • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Synchronous operation with offset mode • Flying homing • Brake test
Gantry	• Time-optimized positioning • Basic positioning • Master-slave closed-loop torque control • Synchronous operation with offset compensatory control • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Flying homing

Description of functions

For information about each function module, refer to Chapter Function modules (Page 37)

Commissioning

For information on commissioning the standard application, refer to Chapter Commissioning (Page 351)

10.7 GD Crane (Grab Dredger)

Quantity structure

Function module	Number of axes	Control mode	Operating mode	Switchover
Holding Gear	1	1. Speed-controlled 2. Synchronous operation between holding gear and closing gear	• SPEED_CONTROLLED • EASY_POSITIONING	
Closing Gear	1	1. Speed-controlled 2. Synchronous operation between holding gear and closing gear	• SPEED_CONTROLLED	
Trolley	1	1. Speed-controlled 2. Positioning	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	

Application of crane-specific technology

Function module	Crane-specific technology
Holding Gear	• Time-optimized positioning • Basic positioning • Synchronous operation between holding gear and closing gear • Load-dependent field weakening • Constant field weakening • Start pulse • Continuous load measurement • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Slack rope control • Prelimit switch • Non-linear setpoint of the master switch • Brake test
Closing Gear	• Time-optimized positioning • Synchronous operation between holding gear and closing gear • Constant field weakening • Start pulse • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Current equalization control • Prelimit switch • Non-linear setpoint of the master switch • Brake test
Trolley	• Time-optimized positioning • Basic positioning • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Flying homing • Brake test

Description of functions

For information about each function module, refer to Chapter Function modules (Page 37)

Commissioning

For information on commissioning the standard application, refer to Chapter Commissioning (Page 351)

10.8 STS tandem crane (ship-to-shore with four hoists)

There are two versions of this standard application:

- STS Tandem Crane DP

  Communication between the S7 and SIMOTION via PROFIBUS
- STS Tandem Crane PN

  Communication between the S7 and SIMOTION via PROFINET

Quantity structure

Function module	Number of axes	Control mode	Operating mode	Switchover
Hoist	4	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation	• AUTOMATIC • MANUAL • SPEED_CONTROLLED • SWAYCONTROL • EASY_POSITIONING	
Trolley	1	1. Speed-controlled 2. Positioning	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • SWAYCONTRO • EASY_POSITIONING L	Boom (standard)
Gantry	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	
Boom	1	1. Speed-controlled	• SPEED_CONTROLLED	Trolley (standard)

Note

"Standard" for switching means: The user does not need to become active explicitly.

For switchover, see Technology object switchover (Page 214), "External TO switchover" section

Application of crane-specific technology

Function module	Crane-specific technology
Hoist	• Time-optimized positioning • Basic positioning • Master-slave closed-loop torque control • Synchronous operation with offset mode • Tandem mode • Load-dependent field weakening • Start pulse • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Optional: Advanced Technology (Sway Control, 2D trajectory) • Brake test
Trolley	• Time-optimized positioning • Basic positioning • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Optional: Advanced Technology (Sway Control, 2D trajectory) • Flying homing • Brake test

Function module	Crane-specific technology
Gantry	• Time-optimized positioning • Basic positioning • Master-slave closed-loop torque control • Synchronous operation with offset compensatory control • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Flying homing • Cornering movement
Boom	• Non-linear setpoint of the master switch • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Start pulse • Brake test

Description of functions

For information about each function module, refer to Chapter Function modules (Page 37)

The STS Tandem Crane standard application is stored in two versions on the product CD: One DP version with 12 PZD and one PN version with 16 PZD; refer to Chapter Communication (Page 293).

Commissioning

For information on commissioning the standard application, refer to Chapter Commissioning (Page 351)

10.9 Easy RTG crane (simple rubber-tired gantry)

Quantity structure

Table 10- 6 Gantry2

Function module	Number of axes	Control mode	Operating mode	Switchover
Hoist	1	1. Speed-controlled	• SPEED_CONTROLLED • EASY_POSITIONING	
Trolley	1	1. Speed-controlled	• SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	
Gantry	2	1. Speed-controlled	• SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	

Application of crane-specific technology

Function module	Crane-specific technology
Hoist	• Load-dependent field weakening • Basic positioning • Start pulse • Prelimit switch • Monitoring overspeed
Trolley	• Prelimit switch • Basic positioning
Gantry	• Prelimit switch • Basic positioning

Description of functions

For information about each function module, refer to Chapter Function modules (Page 37)

Commissioning

For information on commissioning the standard application, refer to Chapter Commissioning (Page 351)

10.10 GSU Tandem Crane (GSU with two closing gear and two holding gear)

Quantity structure

Function module	Number of axes	Control mode	Operating mode	Switchover
Holding Gear	2	1. Speed-controlled 2. Synchronous operation between holding gear and closing gear	• SPEED_CONTROLLED • SWAYCONTROL • EASY_POSITIONING	
Closing Gear	2	1. Speed-controlled 2. Synchronous operation between holding gear and closing gear	• SPEED_CONTROLLED • SWAYCONTROL	
Trolley	1	1. Speed-controlled 2. Positioning	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • SWAYCONTROL • EASY_POSITIONING	Boom (standard) All drives (application)
Long travel (Gantry)	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	All drives (application)
Boom	1	1. Speed-controlled	• SPEED_CONTROLLED	Trolley (standard) All drives (application)

Note

"Standard" for switching means: The user does not need to become active explicitly.

"Application" for switching means: The user must create the drive data sets and determine the selection using the function modules.

See Technology object switchover (Page 214), "External TO switchover" section

Application of crane-specific technology

Function module	Crane-specific technology
Holding Gear	• Time-optimized positioning • Basic positioning • Synchronous operation between holding gear and closing gear • Load-dependent field weakening • Constant field weakening • Continuous load measurement • Start pulse • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Slack rope control • Prelimit switch • Non-linear setpoint of the master switch • Optional: Advanced Technology (Sway Control, 2D trajectory) • Brake test • Tandem Mode
Closing Gear	• Time-optimized positioning • Basic positioning • Synchronous operation between holding gear and closing gear • Constant field weakening • Start pulse • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Current equalization control • Prelimit switch • Non-linear setpoint of the master switch • Optional: Advanced Technology (Sway Control, 2D trajectory) • Brake test • Tandem Mode
Trolley	• Time-optimized positioning • Basic positioning • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Optional: Advanced Technology (Sway Control, 2D trajectory) • Flying homing • Brake test

Function module	Crane-specific technology
Long travel (Gantry)	• Time-optimized positioning • Basic positioning • Synchronous operation with offset compensatory control • Master-slave closed-loop torque control • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Flying homing • Cornering movement
Boom	• Non-linear setpoint of the master switch • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Start pulse • Brake test

Description of functions

For information about each function module, refer to Chapter Function modules (Page 37) .

Commissioning

For information on commissioning the standard application, refer to Chapter Commissioning (Page 351)

Note

On the closing gear, only speed control is possible in grab operation. In hook operation, positioning is also supported.

10.11 OHBC (Overhead Bridge Crane distributed synchronous operation over two SIMOTION D)

Quantity structure for the first SIMOTION D

Function module	Number of axes	Control mode	Operating mode	Switchover
Hoist	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation 5. Synchronous operation with hoist (second SIMOTION D)	• AUTOMATIC • MANUAL • SPEED_CONTROLLED • EASY_POSITIONING	
Trolley	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation 5. Synchronous operation with trolley (second SIMOTION D)	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	
Gantry	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation 5. Synchronous operation with gantry (second SIMOTION D)	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	

CAUTION

This standard project implements the interconnection of technology objects between two SIMOTION D, including the software extension, in order to activate the distributed synchronous operation. The control with which the interface to the distributed synchronous operation should be activated, as well as the error response and the monitoring functions must be implemented by the user.

The distributed synchronous operation is possible only when position control is active (operating mode: AUTOMATIC or MANUAL).

Quantity structure for the second SIMOTION D

Function module	Number of axes	Control mode	Operating mode	Switchover
Hoist	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation 5. Synchronous operation with hoist (first SIMOTION D)	• AUTOMATIC • MANUAL • SPEED_CONTROLLED • EASY_POSITIONING	
Trolley	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation 5. Synchronous operation with trolley (first SIMOTION D)	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	
Gantry	2	1. Speed-controlled 2. Positioning 3. Master-slave operation 4. Synchronous operation 5. Synchronous operation with gantry (first SIMOTION D)	• AUTOMATIC • MANUAL • SENSORLESS EMERGENCY • SPEED_CONTROLLED • EASY_POSITIONING	

Application of crane-specific technology

Function module	Crane-specific technology
Hoist	• Time-optimized positioning • Basic positioning • Load-dependent field weakening • Start pulse • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Synchronous operation with offset mode • Brake test
Trolley	• Time-optimized positioning • Basic positioning • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Synchronous operation with offset mode • Flying homing • Brake test
Gantry	• Time-optimized positioning • Basic positioning • Master-slave closed-loop torque control • Synchronous operation with offset compensatory control • Switchover of the ramp-function generator in the field-weakening range and when selecting heavy-duty operation • Prelimit switch • Non-linear setpoint of the master switch • Traction control • Flying homing

Description of functions

For information about each function module, refer to Chapter Function modules (Page 37)

Commissioning

For information on commissioning the standard application, refer to Chapter Commissioning (Page 351)

Spare parts/Accessories 11

Spare parts

- SIMOTION D435 (order no. 6AU1435-0AA00-0AA1)
- SIMOTION D435-2 (order no. 6AU1435-2AD00-0AA0)
- Memory card for SIMOTION D435 (order no. 6AU1400-2PA02-0AA0)
- Memory card for SIMOTION D435-2 (order no. 6AU1400-2PA22-0AA0)
- To request licenses, please contact
  - Hotline (refer to the preface)
  - Siemens A&D Web License Manager at http://www.siemens.com/automation/license

Note

For questions relating to SIMOCRANE, such as application support, please contact Siemens I DT Application Support (e-mail: Cranesapplications.cranes.aud@siemens.com) or your local sales.

Supplementary components

Depending on the application, the following components can be required:

- SINAMICS Controller Extension SIMOTION CX32 (order no. 6SL3040-0NA00-0AA0)
- SINAMICS Controller Extension SIMOTION CX32-2 (order no. 6AU1432-2AA00-0AA0)
- SINAMICS DRIVE-CLiQ Hub Module DMC20 (order no. 6SL3055-0AA00-6AA0)
- SINAMICS Sensor Module Cabinet-Mounted SMC30 (order no. 6SL3055-0AA00-5CA2)
- SINAMICS Terminal Module TM31 (order no. 6SL3055-0AA00-3AA1)
- CBE30 (order no. 6FC5312-0FA00-0AA0)
- CBE30-2 (order no. 6FC5312-0FA00-2AA0)
- SIMOTION SCOUT V4.1.5 (order no. 6AU1810-1BA41-5XA0)
- Drive Control Chart DCC V2.0.5 option package (order no. 6AU1810-1JA20-5XA0)
- SIMOTION SCOUT V4.1 SP5 upgrade DVD (order no. 6AU1810-1BA41-5XE0)
- SIMOTION SCOUT V4.3.1 (order no. 6AU1810-1BA43-1XA0)
- SIMOTION SCOUT V4.3 SP1 upgrade DVD (order no. 6AU1810-1CA43-1XE0)

- Drive Control Chart DCC V2.2.1 option kit
  (order no. 6AU1810-1JA22-1XA0)
- Drive Control Chart DCC V2.2.1 option kit, upgrade license
  (order no. 6AU1810-1JA22-1XE0)

Additional information is provided in the following Catalogs to select and order supplementary components:

- Catalog CR1, 2012 – Drive and Control Components for Hoisting Gear
- Catalog PM21, 2011 – SIMOTION, SINAMICS S120 and Motors for Production Machines
- Distributed I/O, PROFIBUS:
  Catalog IK PI (Industrial Communication)

Note

Drive systems, motors and connection systems are not included in the package. These components must be ordered separately.

Appendix A

A.1 References

Ref. 1 PROFIBUS and PROFINET, PROFIdrive Profile Drive Technology
PROFIBUS User Organization e. V.
Haid-und-Neu-Straße 7, D-76131 Karlsruhe
http://www.profibus.com
order no. 3.172 Version 4.0 August 2005

Ref. 2 SIMOTION, Motion Control, TO Axis Electric/Hydraulic, External Encoder, Function Manual, Edition 02/2012

Ref. 3 SIMOTION, Technology Packages Alarms, Diagnostics Manual, Edition 02/2012

Ref. 4 SINAMICS S120, S150 List Manual, Edition 01/2012

Ref. 5 SINAMICS S120 Getting Started for the STARTER Commissioning Tool, Edition 11/2009

Ref. 6 CraneSoft-7 V0.6: Interface Description SINAMICS / SIMOTION

Ref. 7 SIMOTION, Technology Packages System Functions, List Manual, Edition 02/2012

Ref. 8 SIMOTION, System Functions/Variables Devices, List Manual, Edition 05/2009

Ref. 9 SIMOTION, Motion Control Supplementary Technology Objects, Function Manual, Edition 02/2012

Ref. 10 SINAMICS/SIMOTION Editor Description DCC, Edition 02/2012

Ref. 11 The Detailed Description of Connections of the Function modules in SIMOCRANE Basic Technology V1.0, Edition 05/2008

Ref. 12 SIMOCRANE SC Integrated, Operating Instructions, Edition 04/2011

Ref. 13 SIMOTION Function Diagram, Edition 10/2010

Ref. 14 SIMOTION FAQ Profinet IO/RT RT I-Device Coupling SIMATIC/SIMOTION, Edition 05/2008

Ref. 15 SIMOTION, Communication, System Manual, Edition 02/2012

Ref. 16 SIMATIC, Communication with SIMATIC, System Manual, Edition 09/2006

A.2 Abbreviations

AppSTW	Applikationssteuerwort/Application Control Word
AppZSW	Applikationszustandswort/Application Status Word
CLM	Continuous Load Measurement
DCC	Drive Control Chart
DDS	Drive Data Set
DO	Drive Object for SINAMICS, e.g. Motor Module
DP-V1	Supplement to PROFIBUS DP in order to carry out acyclic communication.
FBD	The function block diagram (FBD) is one of the three programming languages for STEP 5 and STEP 7. FBD uses the logic boxes known from Boolean algebra for mapping the logic.
FW	Field Weakening
GSU	Grab Ship Unloader
HG	Holding Gear
IPO	Interpolator
LDFW	Load-dependent Field Weakening
MCC	Motion Control Chart
MM	Motor Module
MPI	Multiple Point Interface
OHBC	Overhead Bridge Crane
PLI	Drive curve (polygon characteristic for current)
PWM	Pulse-Width Modulation
PZD	PROFIBUS process parameter
RMG	Rail-Mounted Gantry
RTG	Rubber-Tired Gantry
SG	Slewing Gear
ST	Structured Text; text-based high-level language for SIMOTION that has been extended with motion control and other language commands. These are integrated as functions or function blocks.
STS	Ship-to-Shore
STW	Steuerwort/Control Word
TO	Technology Object in SIMOTION; symbol for a moving axis
ZSW	Zustandswort/Status Word

A.3 Terminology (German/English)

englisch	deutsch
Acceleration	Beschleunigung
Actual value	Istwert
Application	Applikation
Automatic Mode	Betriebsart Automatik
Axis	Achse
Basic Technology	Basistechnologie
Block name	Bausteinname
Boom (Boom-Hoist)	Ausleger (Einziehwerk)
Closing Gear	Schließwerk, Schließwinde
Comparator	Vergleicher
Commissioning	Inbetriebnahme
Connection	Anschluss
Control word	Steuerwort
Cross Travel (CT)	siehe Trolley
Current	Strom
Current distribution	Stromverteilung
Current equalization control	Stromausgleichsregelung
Data type	Datentyp
Deceleration	Verzögerung
Error Priority	Fehler-Priorisierung
Gantry (Long Travel)	Fahrwerk
Heavy Duty	Schwerlast
Hoist	Hubwerk
Holding Gear	Haltewerk, Haltewinde
Homing	Referenzieren
Jogging Mode	Betriebsart Tippen
Library	Bibliothek
Load depending field weakening	lastabhängige Feldschwächung
Long Travel (LT)	siehe Gantry
Luffing Gear	Wippwerk
Manual Mode	Betriebsart Hand
Master Switch	Masterschalter
Monitoring	Überwachung
Offset	Versatz
Operation mode	Betriebsart
Overspeed	Überdrehzahl
Speed_Controlled Mode	Betriebsart Drehzahlgeregelt
Prelimit switch	Vorendschalter
Procedure	Vorgehensweise
Receive from	Empfangen von
Send to	Senden an
Sensorless Emergency	geberloser Notbetrieb
Setpoint	Sollwert
Ship to Shore (STS)	Containerkaikran
Slack rope	Schlaffseil
Slew Gear	Drehwerk
Speed	Drehzahl
Start pulse	Startimpuls
Status word	Zustandswort
Switch over	Umschaltung
Synchronous operation	Gleichlauf
Torque	Drehmoment
Traction control	Traktionskontrolle (Antischlupfregelung)
Trolley (Cross Travel)	Katze, Katzfahrwerk
User program	Anwenderprogramm
Velocity	Geschwindigkeit
working radius	Ausladung