Library for load balancing with SIMATIC S7-1500 (LLoadBal)

SIMATIC S7-1500 / TIA V16 / V1.0 / LLoadBal

https://support.industry.siemens.com/cs/ww/en/view/109794291

Siemens Industry Online Support

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG (“Siemens”). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.
Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.
By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.
The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens’ products and solutions constitute one element of such a concept.
Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.
For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.
Siemens’ products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer’s exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: https://www.siemens.com/industrialsecurity.

Table of contents

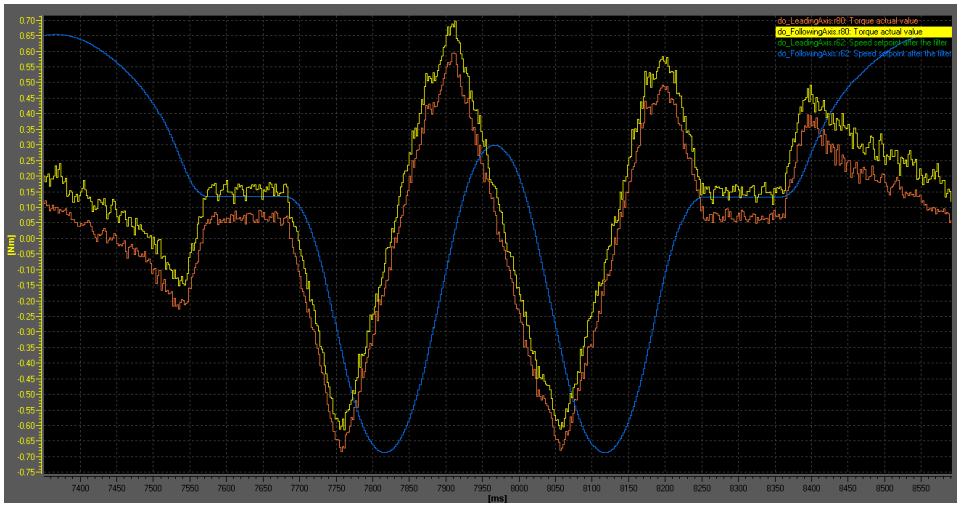
1 Introduction

1.1 Load balancing

If several motors operate together load balancing is required to distribute the complete mechanical load in a controlled fashion among the individual motors. If the motors are controlled without load balancing this can lead to the motors work against one another and even start oscillating. Their temperature then might rise significantly because of the asymmetrical load – or they reach their maximum torque and trip with a fault. To distribute the load across the motors involved a stiff mechanical connection is mandatory. This means that the motors must drive a common gearbox or need to be coupled through a shaft.

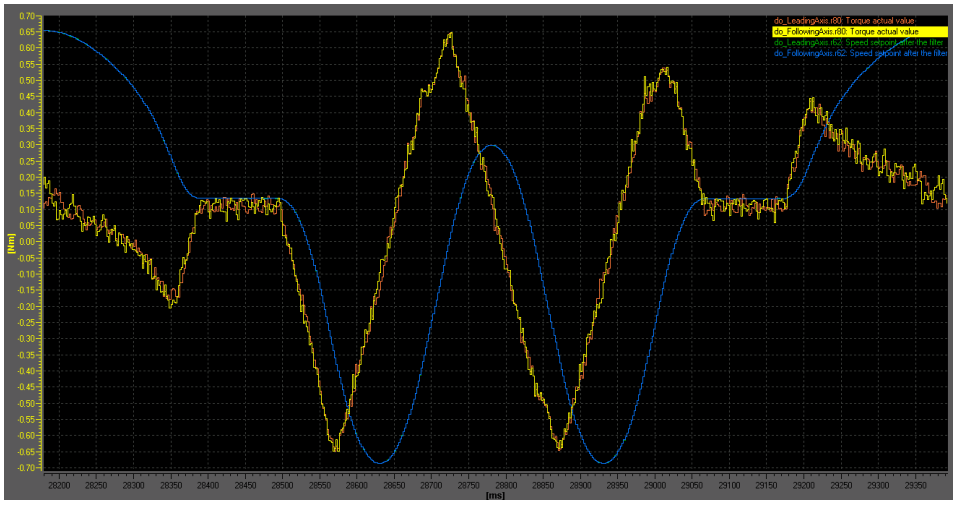
In Figure 1-1 the actual torque values of both motors are showing a difference. The same speed setpoint is forced by the mechanical coupling but both motors see different load.

Figure 1-1 Mechanically coupled motors **without** load balancing



In Figure 1-2 the actual torque values are identical. The motors work together and share the load symmetrically.

Figure 1-2 Mechanically coupled motors **with** load balancing



2 Solution

2.1 Principle of load balancing

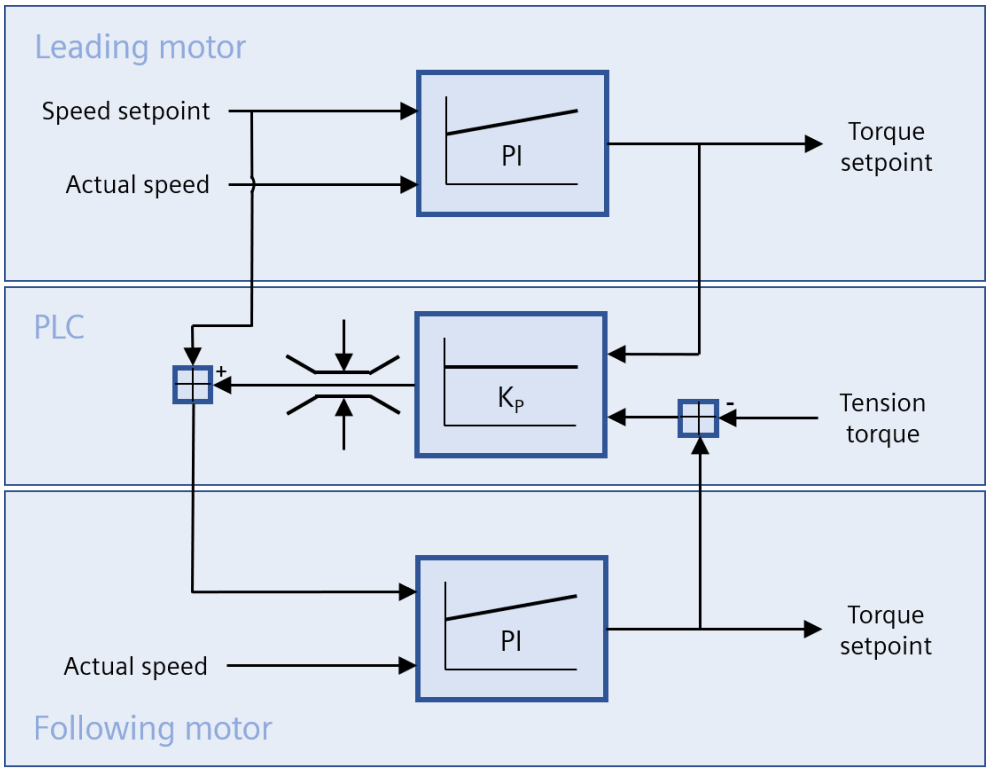
The principle of load balancing described in this document is based on comparing the torque setpoints of the motors that should operate together. Here the motors acting in a leading-following relation, whereas one leading motor can have several following motors assigned. For each following motor there is a load balancing controller running in PLC side which evaluates deviations in the torque setpoint compared to its leading motor. This evaluation results in an output value that is used as additional speed setpoint for the certain following motor to share the load symmetrically.

NOTE For the load balancing to work, basically the leading and following motor(s) must use the same speed setpoint. This can be ensured by using just one position control for both and distributing the leading motor speed setpoint to its certain following motor(s) (see chapter 5.1).

If the mechanical coupling is lost (e.g. due to material breakage or shaft breakage) the output of the load balancing controller is limited to the values defined by the user. This prevents the motor from uncontrollably accelerating up to its maximum speed.

If, for mechanical reasons (e.g. backlash compensation in the gearbox), a tension is desirable between leading and following motor, it's possible to set a static tension torque. The load balancing controller ensures that this torque is then applied as constant difference between the two motors.

Figure 2-1 Principle of load balancing

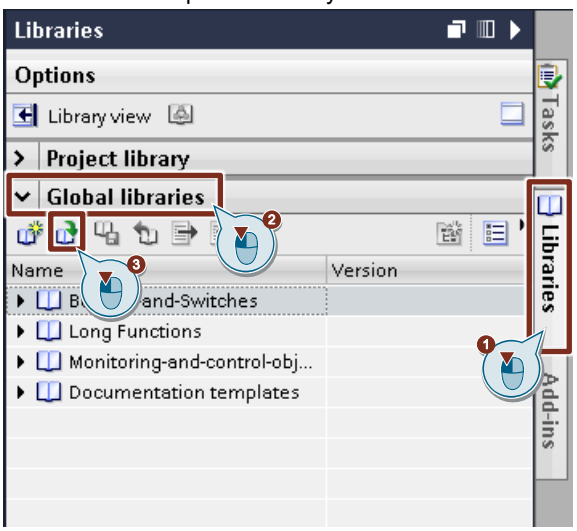
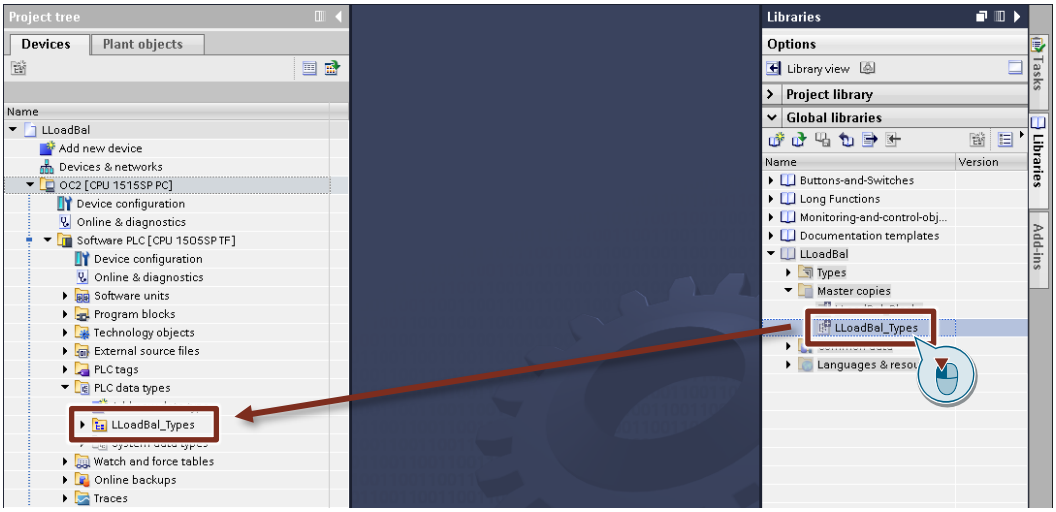


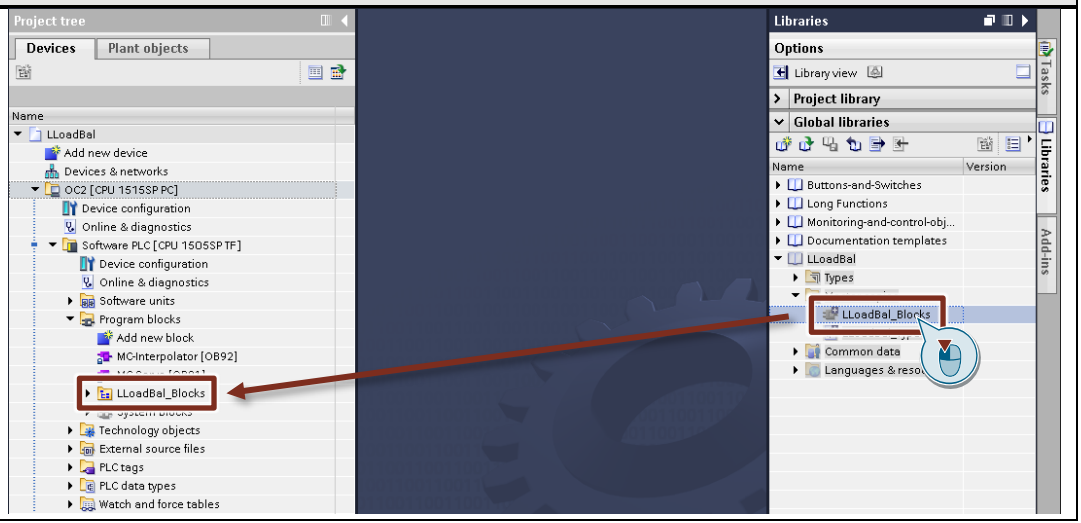
2.2 Integration into the user project

The functionality of distributing the load among two or several motors is provided by the library “LLoadBal”, which has been developed for S7-1500 PLCs.

For integrating the library into existing user projects, the following steps must be performed.

Table 2-1 Integration of the library “LLoadBal”

No.	Action
1.	Open the “Libraries” task card on the right side of TIA Portal and navigate to the “Global libraries” tab. Afterwards open the library “LLoadBal.zal16”. 
2.	Insert the PLC data types provided by the library by drag and drop the group “LLoadBal_Types” from the master copies to the “PLC data types” folder of the certain PLC. 
3.	Insert the program blocks provided by the library by drag and drop the group “LLoadBal_Blocks” from the master copies to the “Program blocks” folder of the certain PLC.

No.	Action
	

2.3 Hardware and software requirements

Requirements for this library

To be able to use the functionality of the library described here, the following hardware and software requirements must be met.

Hardware

- S7-1500(TF) / Software Controller

Software

- TIA Portal V16
- STEP 7 Professional

NOTE In general, it is possible to open a library with STEP 7 Basic, although STEP 7 Professional elements (e.g. SIMATIC S7-1500 controller) are included. In this case you will be informed with a message when opening the library.

All elements (types and copy templates) can be used if they are supported by the hardware installed in the TIA Portal. If you try to copy elements with STEP 7 Basic from the library that are not supported (e.g. SIMATIC S7-1500 controller), an error message is displayed.

3 Content of the library

3.1 Function blocks

Table 3-1 Overview of function blocks

Overview of function blocks
LLoadBal_LeadingAxis
LLoadBal_FollowingAxis
LLoadBal_PIDController
LLoadBal_GetAxisConfiguation
LLoadBal_GetAxisStatus
LLoadBal_ReadLeadingAxisSetpointValues
LLoadBal_WriteFollowingAxisSetpointValues

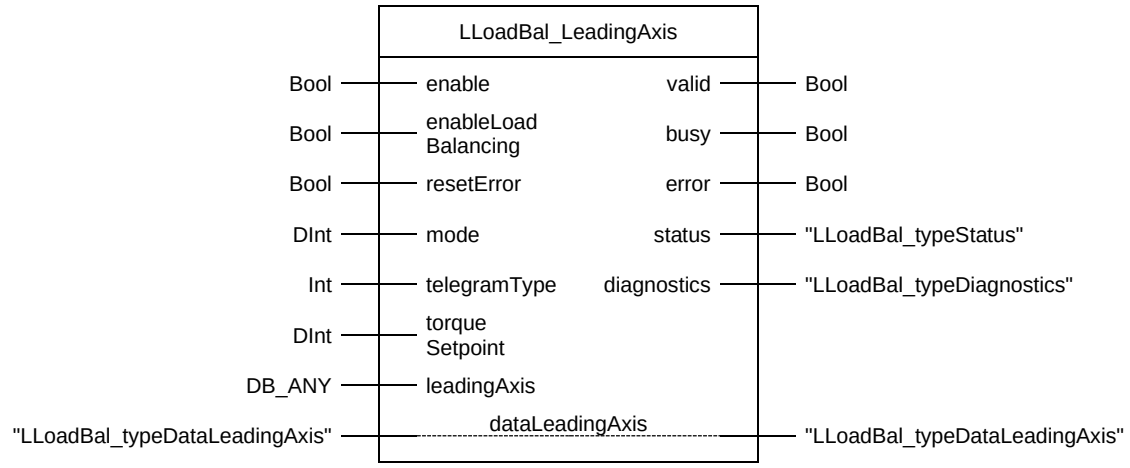
3.1.1 LLoadBal_LeadingAxis

The function block “LLoadBal_LeadingAxis” provides the functionality for a leading axis to balance the load between the participating axes in a leading-following relation.

This function block must be used in combination with the function block “LLoadBal_FollowingAxis”, which represents the counter part for the leading axis function block. The assignment of a following axis to its certain leading axis is done using the “dataLeadingAxis” variable, which must be connected to the corresponding variable at the function block “LLoadBal_FollowingAxis”. Each following axis can only have one leading axis, whereas one leading axis can have several following axes assigned.

If load balancing is enabled (“enableLoadBalancing”) the function block provides the necessary information for its following axes via the “dataLeadingAxis” variable, which mainly comprises the leading axis torque setpoint as well as information about its speed setpoint (NSOLL_B) and DSC[1] setpoint values (XERR, KPC).

Figure 3-1: LLoadBal_LeadingAxis



[1] Dynamic Servo Control

Table 3-2: Parameter of LLoadBal_LeadingAxis

Name	P-Type	Data Type	Comment
enable	IN	Bool	TRUE: Enable functionality of FB
enableLoad Balancing	IN	Bool	TRUE: Enable load balancing functionality; FALSE: Disable load balancing functionality (= MODE_IDLE)
resetError	IN	Bool	Rising edge: Resetting of FB errors
mode	IN	DInt	Mode of load balancing: 0 = MODE_IDLE, 1 = MODE_ADDITIONAL_SPEED_ SETPOINT, 2 = MODE_USERDEFINED
telegramType	IN	Int	Used telegram of drive connected to axis (i.e. telegram number)
torqueSetpoint	IN	DInt	Torque setpoint of drive connected to axis
leadingAxis	IN	DB_ANY	Technology object of leading axis (TO_SpeedAxis, TO_PositioningAxis, TO_SynchronousAxis)
valid	OUT	Bool	TRUE: Valid set of output values available at the FB
busy	OUT	Bool	TRUE: FB is not finished and new output values can be expected
error	OUT	Bool	TRUE: An error occurred during the execution of the FB
status	OUT	"LLoadBal_typeStatus"	16#0000 - 16#7FFF: Status of the FB, 16#8000 - 16#FFFF: Error identification
diagnostics	OUT	"LLoadBal_typeDiagnostics"	Diagnostics information of FB (optional)
dataLeadingAxis	IN_OUT	"LLoadBal_typeDataLeading Axis"	Data structure containing information of the leading axis (used for data exchange between leading and following axis)

Table 3-3: Status of LLoadBal_LeadingAxis

Name	Value	Comment
STATUS_NO_CALL	16#7000	No job being currently processed
STATUS_FIRST_CALL	16#7001	First call after incoming new job (rising edge enable)
STATUS_SUBSEQUENT_CALL	16#7002	Subsequent call during active processing without further details
ERR_UNDEFINED_STATE	16#8000	Error: due to an undefined state in the state machine
ERR_PARAMETERIZATION	16#8200	Error: during parameterization
ERR_PROCESSING_EXTERN	16#8400	Error: when processing from outside (e.g. wrong I/O signals, axis not referenced)
ERR_PROCESSING_INTERN	16#8600	Error: when processing internally (e.g. when calling a system function)

Table 3-4: Diagnostics.subfunctionStatus of LLoadBal_LeadingAxis

Name	Value	Comment
ERR_LEADING_AXIS_INVALID	16#8201	Error: Leading axis connected to input “leadingAxis” does not match the permitted types (i.e. TO_SpeedAxis, TO_PositioningAxis, TO_SynchronousAxis)
ERR_TELEGRAM_TYPE_INVALID	16#8202	Error: Telegram type (i.e. telegram number) interconnected to input “telegramType” does not match the permitted values (see chapter 4.1.2) or telegram does not support DSC while DSC is enabled for the axis
ERR_MODE_INVALID	16#8203	Error: Mode specified at input “mode” does not match the permitted values: • (0) MODE_IDLE • (1) MODE_ADDITIONAL_SPEED_SETPOINT • (2) MODE_USERDEFINED
ERR_READ_LEADING_AXIS_SETPOINT_VALUES	16#8601	Error: Reading setpoint values of the leading axis not possible

3.1.2 LLoadBal_FollowingAxis

The function block “LLoadBal_FollowingAxis” represents the counter part for the function block “LLoadBal_LeadingAxis” and provides the functionality for a following axis to balance the load between the participating axes in a leading-following relation.

Assigning the following axis to its certain leading axis is done by using the “dataLeadingAxis” variable, which is used as well for controlling the function block according to its leading axis actual status. Depending on the load balancing mode chosen at the leading axis (input “mode”) and enabling of the load balancing functionality itself (input “enableLoadBalancing”) an additional speed setpoint gets calculated for the certain following axis, which is the result of comparing the leading and following axis torque setpoint by a load balancing controller. As one leading axis can have several following axes assigned each following axis has its own load balancing controller running.

Figure 3-2: LLoadBal_FollowingAxis

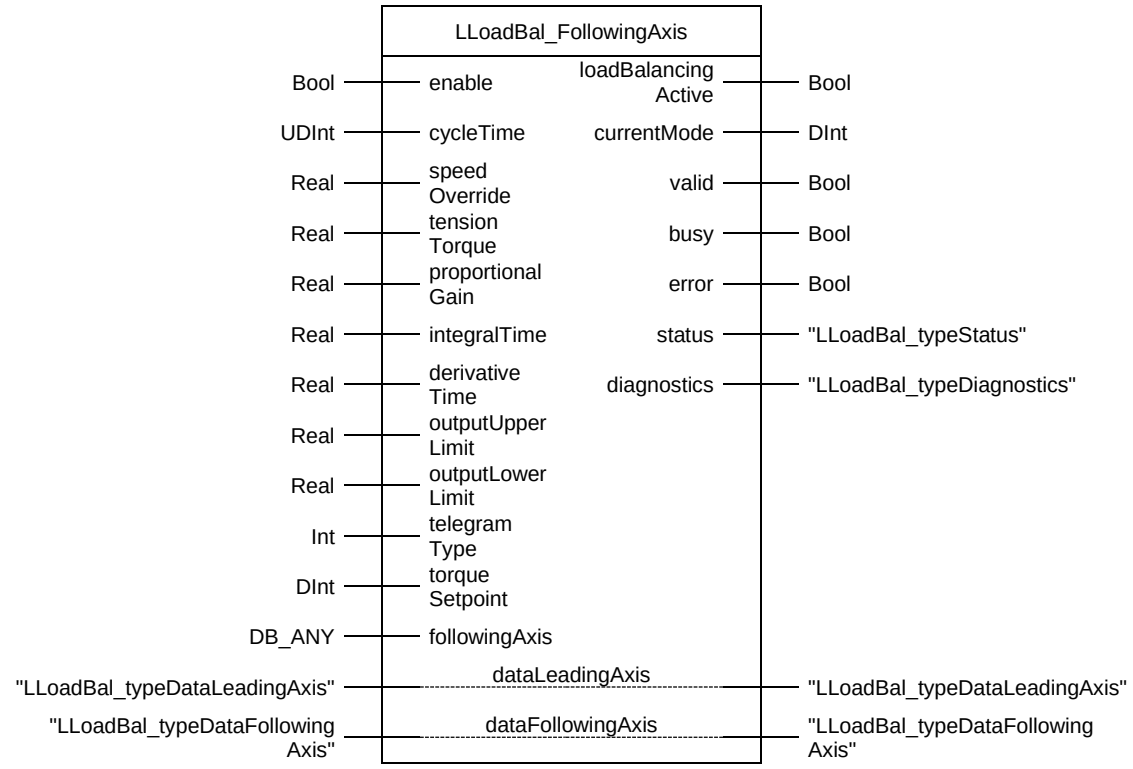


Table 3-5: Parameter of LLoadBal_FollowingAxis

Name	P-Type	Data Type	Comment
enable	IN	Bool	TRUE: Enable functionality of FB
cycleTime	IN	UDInt	Cycle time [ns] of the calling OB (variable #CycleTime of OB)
speedOverride	IN	Real	Speed override [n * speed setpoint of leading axis] for the following axis
tensionTorque	IN	Real	Torque value [Nm] (optional) used for tension between leading and following axis (if mode = (1) MODE_ADDITIONAL_SPEED_ SETPOINT))

Name	P-Type	Data Type	Comment
proportionalGain	IN	Real	Proportional gain [(% * reference speed of following axis) / Nm] for the load balancing controller (if mode = (1) MODE_ADDITIONAL_SPEED_ SETPOINT)
integralTime	IN	Real	Integral time [s] (optional) for the load balancing controller (if mode = (1) MODE_ADDITIONAL_SPEED_ SETPOINT)
derivativeTime	IN	Real	Derivative time [s] (optional) for the load balancing controller (if mode = (1) MODE_ADDITIONAL_SPEED_ SETPOINT)
outputUpperLimit	IN	Real	Upper limit [rpm] for the additional speed setpoint applied to following axis (if mode = (1) MODE_ADDITIONAL_SPEED_ SETPOINT)
outputLowerLimit	IN	Real	Lower limit [rpm] for the additional speed setpoint applied to following axis (if mode = (1) MODE_ADDITIONAL_SPEED_ SETPOINT)
telegramType	IN	Int	Used telegram of drive connected to axis (i.e. telegram number)
torqueSetpoint	IN	DInt	Torque setpoint of drive connected to axis
followingAxis	IN	DB_ANY	Technology object of following axis (TO_SpeedAxis, TO_PositioningAxis, TO_SynchronousAxis)
loadBalancing Active	OUT	Bool	TRUE: Load balancing functionality is active (e.g. if mode = (1) MODE_ADDITIONAL_SPEED_ SETPOINT calculation of additional speed setpoint for following axis is performed)
currentMode	OUT	DInt	Active mode of load balancing (0 = MODE_IDLE, 1 = MODE_ADDITIONAL_SPEED_ SETPOINT, 2 = MODE_USERDEFINED)
valid	OUT	Bool	TRUE: Valid set of output values available at the FB
busy	OUT	Bool	TRUE: FB is not finished and new output values can be expected
error	OUT	Bool	TRUE: An error occurred during the execution of the FB
status	OUT	"LLoadBal_typeStatus"	16#0000 - 16#7FFF: Status of the FB, 16#8000 - 16#FFFF: Error identification
diagnostics	OUT	"LLoadBal_typeDiagnostics"	Diagnostics information of FB (optional)
dataLeadingAxis	IN_OUT	"LLoadBal_typeDataLeading Axis"	Data structure containing information of the leading axis (used for data exchange between leading and following axis)

Name	P-Type	Data Type	Comment
dataFollowingAxis	IN_OUT	"LLoadBal_typeDataFollowing Axis"	Data structure containing information of the following axis (used for data exchange between leading and following axis)

Table 3-6: Status of LLoadBal_FollowingAxis

Name	Value	Comment
STATUS_NO_CALL	16#7000	No job being currently processed
STATUS_FIRST_CALL	16#7001	First call after incoming new job (rising edge enable)
STATUS_SUBSEQUENT_CALL	16#7002	Subsequent call during active processing without further details
ERR_UNDEFINED_STATE	16#8000	Error: due to an undefined state in the state machine
ERR_PARAMETERIZATION	16#8200	Error: during parameterization
ERR_PROCESSING_EXTERN	16#8400	Error: when processing from outside (e.g. wrong I/O signals, axis not referenced)
ERR_PROCESSING_INTERN	16#8600	Error: when processing internally (e.g. when calling a system function)

Table 3-7: Diagnostics.subfunctionStatus of LLoadBal_FollowingAxis

Name	Value	Comment
ERR_FOLLOWING_AXIS_INVALID	16#8201	Error: Following axis connected to input “followingAxis” does not match the permitted types (i.e. TO_SpeedAxis, TO_PositioningAxis, TO_SynchronousAxis)
ERR_TELEGRAM_TYPE_INVALID	16#8202	Error: Telegram type (i.e. telegram number) interconnected to input “telegramType” does not match the permitted values (see chapter 4.1.2) or telegram does not support DSC while DSC is enabled for the axis
ERR_LOAD_BALANCING_CONTROLLER	16#8601	Error: of instance of function block “LLoadBal_PIDController”
ERR_WRITE_FOLLOWING_AXIS_SETPOINT_VALUES	16#8602	Error: Writing setpoint values of the following axis not possible

3.1.3 LLoadBal_PIDController

The function block “LLoadBal_PIDController”, which is based on the system function block “PID_Compact” of TIA Portal, represents the load balancing controller for the following axis.

Figure 3-3: LLoadBal_PIDController

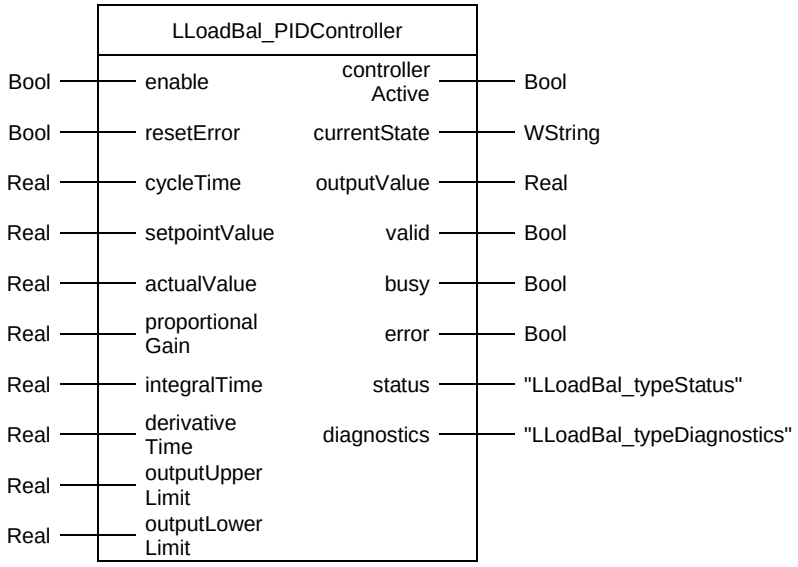


Table 3-8: Parameter of LLoadBal_PIDController

Name	P-Type	Data Type	Comment
enable	IN	Bool	TRUE: Enable functionality of FB
resetError	IN	Bool	Rising edge: Reset of FB errors
cycleTime	IN	Real	Cycle time [s] of the calling OB / FB
setpointValue	IN	Real	Setpoint value for PID_Compact
actualValue	IN	Real	Actual value for PID_Compact
proportionalGain	IN	Real	Proportional gain for PID_Compact
integralTime	IN	Real	Integral time [s] for PID_Compact
derivativeTime	IN	Real	Derivative time [s] for PID_Compact
outputUpperLimit	IN	Real	Upper limit for the output value of PID_Compact
outputLowerLimit	IN	Real	Lower limit for the output value of PID_Compact
controllerActive	OUT	Bool	TRUE: PID_Compact is active; FALSE: PID_Compact is inactive
currentState	OUT	WString	Active state of PID_Compact (INACTIVE, PRE_TUNING, FINE_TUNING, AUTOMATIC, MANUAL, SUBSTITUTE_OUTPUT_VALUE)
outputValue	OUT	Real	Output value of PID_Compact
valid	OUT	Bool	TRUE: Valid set of output values available at the FB
busy	OUT	Bool	TRUE: FB is not finished and new output values can be expected
error	OUT	Bool	TRUE: An error occurred during the

Name	P-Type	Data Type	Comment
			execution of the FB
status	OUT	"LLoadBal_typeStatus"	16#0000 - 16#7FFF: Status of the FB, 16#8000 - 16#FFFF: Error identification
diagnostics	OUT	"LLoadBal_typeDiagnostics"	Diagnostics information of FB (optional)

Table 3-9: Status of LLoadBal_PIDController

Name	Value	Comment
STATUS_NO_CALL	16#7000	No job being currently processed
STATUS_FIRST_CALL	16#7001	First call after incoming new job (rising edge enable)
STATUS_SUBSEQUENT_CALL	16#7002	Subsequent call during active processing without further details
ERR_UNDEFINED_STATE	16#8000	Error: due to an undefined state in the state machine
ERR_PARAMETERIZATION	16#8200	Error: during parameterization
ERR_PROCESSING_EXTERN	16#8400	Error: when processing from outside (e.g. wrong I/O signals, axis not referenced)
ERR_PROCESSING_INTERN	16#8600	Error: when processing internally (e.g. when calling a system function)

Table 3-10: Diagnostics.subfunctionStatus of LLoadBal_PIDController

Name	Value	Comment
ERR_OUTPUT_LIMITS_INVALID	16#8201	Error: Upper output limit < lower output limit
ERR_P_GAIN_INVALID	16#8202	Error: Proportional gain < 0.0
ERR_INTEGRAL_TIME_INVALID	16#8203	Error: Integral time < 0.0
ERR_DERIVATIVE_TIME_INVALID	16#8204	Error: Derivative time < 0.0
ERR_CYCLE_TIME_INVALID	16#8205	Error: Cycle time <= 0.0
ERR_PID_CONTROLLER	16#8601	Error: of system function PID_Compact (see ErrorBits for further details)

3.1.4 LLoadBal_GetAxisConfiguation

The function block “LLoadBal_GetAxisConfiguration” reads necessary configuration data of the specified axis and provides it at output “axisConfiguration”.

Figure 3-4: LLoadBal_GetAxisConfiguration

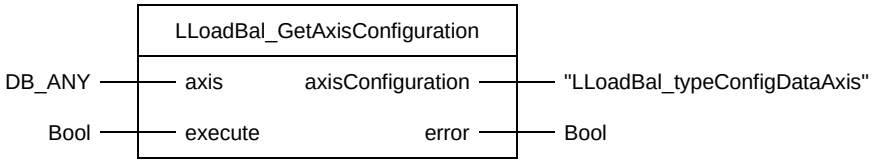


Table 3-11: Parameter of LLoadBal_GetAxisConfiguration

Name	P-Type	Data Type	Comment
axis	IN	DB_ANY	Axis specification
execute	IN	Bool	Rising edge: Execute functionality of FB
axisConfiguration	OUT	"LLoadBal_typeConfigDataAxis"	Configuration data of axis
error	OUT	Bool	TRUE: Error occured during execution (invalid axis specification)

3.1.5 LLoadBal_GetAxisStatus

The function block “LLoadBal_GetAxisStatus” evaluates necessary status data of the specified axis and provides it at output “axisStatus”.

Figure 3-5: LLoadBal_GetAxisStatus

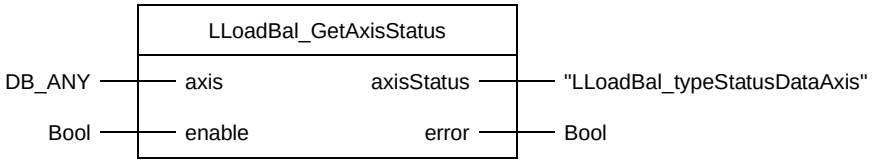


Table 3-12: Parameter of LLoadBal_GetAxisStatus

Name	P-Type	Data Type	Comment
axis	IN	DB_ANY	Axis specification
enable	IN	Bool	TRUE: Enable functionality of FB
axisStatus	OUT	"LLoadBal_typeStatusDataAxis"	Status data of axis
error	OUT	Bool	TRUE: Error occured during execution (invalid axis specification)

3.1.6 LLoadBal_ReadLeadingAxisSetpointValues

The function block “LLoardBal_ReadLeadingAxisSetpointValues” cyclically reads the speed setpoint value (NSOLL_B) as well as the DSC setpoint values (XERR, KPC) of the leading axis and copies them into the “dataLeadingAxis” structure.

Figure 3-6: LLoadBal_ReadLeadingAxisSetpointValues

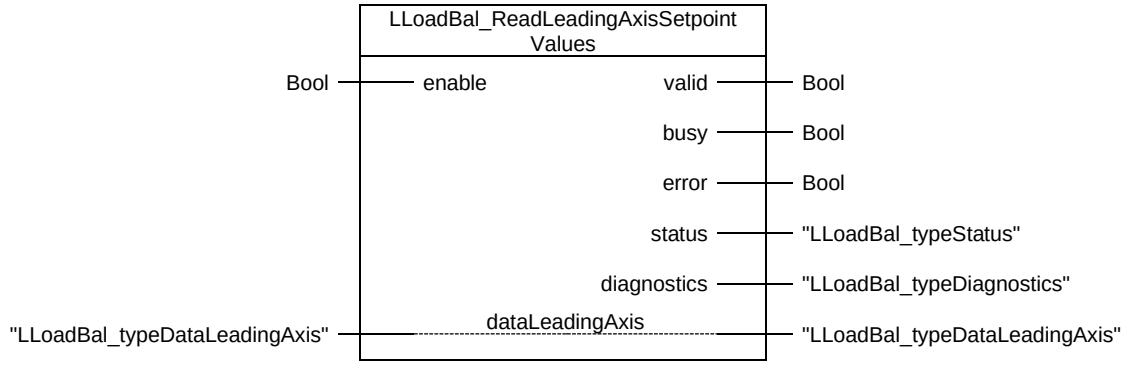


Table 3-13: Parameter of LLoadBal_ReadLeadingAxisSetpointValues

Name	P-Type	Data Type	Comment
enable	IN	Bool	TRUE: Enable functionality of FB
valid	OUT	Bool	TRUE: Valid set of output values available at the FB
busy	OUT	Bool	TRUE: FB is not finished and new output values can be expected
error	OUT	Bool	TRUE: An error occurred during the execution of the FB
status	OUT	"LLoadBal_typeStatus"	16#0000 - 16#7FFF: Status of the FB, 16#8000 - 16#FFFF: Error identification
diagnostics	OUT	"LLoadBal_typeDiagnostics"	Diagnostics information of FB (optional)
dataLeadingAxis	IN_OUT	"LLoadBal_typeDataLeadingAxis"	Data structure containing information of the leading axis (used for data exchange between leading and following axis)

Table 3-14: Status of LLoadBal_ReadLeadingAxisSetpointValues

Name	Value	Comment
STATUS_NO_CALL	16#7000	No job being currently processed
STATUS_FIRST_CALL	16#7001	First call after incoming new job (rising edge enable)
STATUS_SUBSEQUENT_CALL	16#7002	Subsequent call during active processing without further details
ERR_UNDEFINED_STATE	16#8000	Error: due to an undefined state in the state machine
ERR_PARAMETERIZATION	16#8200	Error: during parameterization
ERR_PROCESSING_EXTERN	16#8400	Error: when processing from outside (e.g. wrong I/O signals, axis not referenced)
ERR_PROCESSING_INTERN	16#8600	Error: when processing internally (e.g. when calling a system function)

Table 3-15: Diagnostics.subfunctionStatus of LLoadBal_ReadLeadingAxisSetpointValues

Name	Value	Comment
ERR_ACTOR_TYPE_INVALID	16#8401	Error: Actor type of axis is invalid (analog)
ERR_BLOCK_IS_OPTIMIZED	16#8402	Error: DB the axis is connected with is optimized
ERR_READING_BLOCK_DATA	16#8601	Error: due to reading attributes of DB the axis is connected with
ERR_READING_SETPOINT_VALUES	16#8602	Error: Reading setpoint values of leading axis is not possible

3.1.7 LLoadBal_WriteFollowingAxisSetpointValues

The function block “LLoadBal_WriteFollowingAxisSetpointValues” cyclically writes the speed setpoint value (NSOLL_B) as well as the DSC setpoint values (XERR / KPC, if DSC is enabled) provided within the “dataLeadingAxis” structure to the following axis.

Figure 3-7: LLoadBal_WriteFollowingAxisSetpointValues

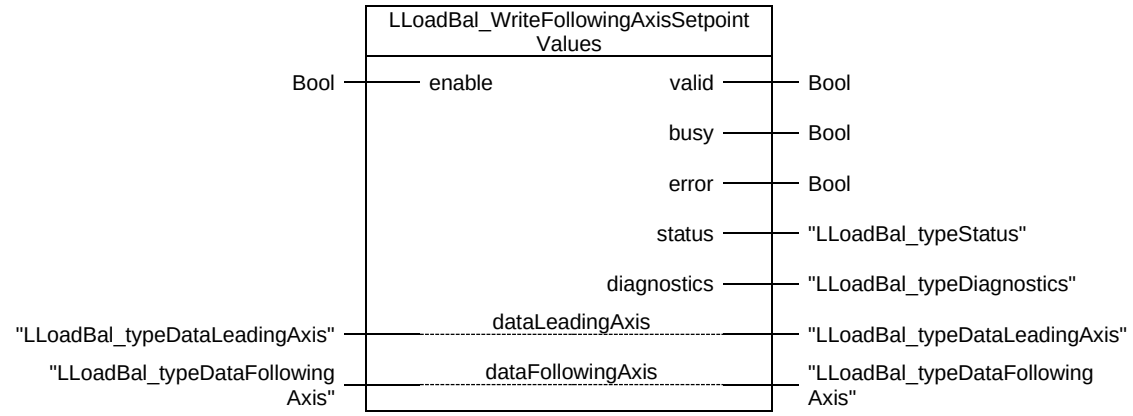


Table 3-16: Parameter of LLoadBal_WriteFollowingAxisSetpointValues

Name	P-Type	Data Type	Comment
enable	IN	Bool	TRUE: Enable functionality of FB
valid	OUT	Bool	TRUE: Valid set of output values available at the FB
busy	OUT	Bool	TRUE: FB is not finished and new output values can be expected
error	OUT	Bool	TRUE: An error occurred during the execution of the FB
status	OUT	"LLoadBal_typeStatus"	16#0000 - 16#7FFF: Status of the FB, 16#8000 - 16#FFFF: Error identification
diagnostics	OUT	"LLoadBal_typeDiagnostics"	Diagnostics information of FB (optional)
dataLeadingAxis	IN_OUT	"LLoadBal_typeDataLeadingAxis"	Data structure containing information of the leading axis (used for data exchange between leading and following axis)
dataFollowingAxis	IN_OUT	"LLoadBal_typeDataFollowingAxis"	Data structure containing information of the following axis (used for data exchange between leading and following axis)

Table 3-17: Status of LLoadBal_WriteFollowingAxisSetpointValues

Name	Value	Comment
STATUS_NO_CALL	16#7000	No job being currently processed
STATUS_FIRST_CALL	16#7001	First call after incoming new job (rising edge enable)
STATUS_SUBSEQUENT_CALL	16#7002	Subsequent call during active processing without further details
ERR_UNDEFINED_STATE	16#8000	Error: due to an undefined state in the state machine

Name	Value	Comment
ERR_PARAMETERIZATION	16#8200	Error: during parameterization
ERR_PROCESSING_EXTERN	16#8400	Error: when processing from outside (e.g. wrong I/O signals, axis not referenced)
ERR_PROCESSING_INTERN	16#8600	Error: when processing internally (e.g. when calling a system function)

Table 3-18: Diagnostics.subfunctionStatus of LLoadBal_WriteFollowingAxisSetpointValues

Name	Value	Comment
ERR_ACTOR_TYPE_INVALID	16#8401	Error: Actor type of axis is invalid (analog)
ERR_BLOCK_IS_OPTIMIZED	16#8402	Error: DB the axis is connected with is optimized
ERR_READING_BLOCK_DATA	16#8601	Error: due to reading attributes of DB the axis is connected with
ERR_WRITING_SETPOINT_VALUES	16#8602	Error: Writing setpoint values of following axis is not possible

3.2 Functions

Table 3-19 Overview of functions

Function	Description
LLoadBal_CheckTelegramType	Checks if the specified telegram type is valid and if the telegram supports using DSC. In case the telegram type is invalid (see chapter 4.1.2) or does not support DSC (while DSC is enabled) the error output is set.
LLoadBal_GetNameOfConstant	Returns the name of a constant variable in string format.
LLoadBal_UnpubConfigurationSpeedAxis	Returns the configuration of the specified axis (TO_SpeedAxis).
LLoadBal_UnpubConfigurationPosAxis	Returns the configuration of the specified axis (TO_PositioningAxis).
LLoadBal_UnpubConfigurationSyncAxis	Returns the configuration of the specified axis (TO_SynchronousAxis).
LLoadBal_UnpubStatusSpeedAxis	Returns the status of the specified axis (TO_SpeedAxis).
LLoadBal_UnpubStatusPosAxis	Returns the status of the specified axis (TO_PositioningAxis).
LLoadBal_UnpubStatusSyncAxis	Returns the status of the specified axis (TO_SynchronousAxis).
LLoadBal_GetByteOffsetValues	Returns the positions (byte offset) of the PZDs NSOLL_B, XERR and KPC within the specified telegram type.
LLoadBal_ScaleValue	Converts an unscaled value to a scaled value using the specified reference value.
LLoadBal_UnscaleValue	Converts a scaled value to an unscaled value using the specified reference value.

3.3 Data types

Table 3-20 Overview of data types

Data type	Description
LLoadBal_typeDataLeadingAxis	Data of the leading axis.
LLoadBal_typeDataFollowingAxis	Data of the following axis.
LLoadBal_typeConfigDataAxis	Configuration data of the axis.
LLoadBal_typeStatusDataAxis	Status data of the axis.
LLoadBal_typeSetpointDataAxis	Setpoint data of the axis.
LLoadBal_typeActorData	Actor data of the axis.
LLoadBal_typeControllerData	Data of the PID controller.
LLoadBal_typeControllerConfig	Configuration data of the PID controller.
LLoadBal_typeDiagnostics	Diagnostics data of the function blocks.
LLoadBal_typeStatus	Status data of the function blocks.

4 Commissioning

4.1 Preconditions

The following preconditions must be observed when using the library “LLoadBal”.

4.1.1 Supported axis types

The function blocks “LLoadBal_LeadingAxis” and “LLoadBal_FollowingAxis” each need to have a certain axis TO[2] assigned, which is supposed to be the leading or the following axis in terms of load balancing.

The following axis types are supported:

- Speed axis (“TO_SpeedAxis”)
- Positioning axis (“TO_PositioningAxis”)
- Synchronous axis (“TO_SynchronousAxis”)

4.1.2 Supported drive telegrams

To be able to read the leading axis setpoint values (i.e. speed setpoint and DSC setpoint[3]) and distribute them to its certain following axes, the function blocks “LLoadBal_LeadingAxis” and “LLoadBal_FollowingAxis” must know which telegram is used for the communication between the assigned axis TO and its interconnected drive.

The following drive telegrams are supported:

- Standard telegram 2 / 3 / 4 / 5 / 6
- Siemens telegram 102 / 103 / 105 / 106 / 116 / 118 / 125 / 126 / 136 / 138 / 139 / 166
- Free telegram configuration via BICO (which is based on standard or Siemens telegrams listed above)

4.1.3 Configuration of axes, drives and motors

CAUTION	**Axes, drives and motors, which should act in a leading-following relation using the load balancing functionality provided by library “LLoadBal” must be configured identically!**

This comprises the following:

- If DSC is enabled for the leading axis also the following axes must have DSC enabled (and vice versa).
- Same mechanical settings for leading axis and the following axes (e.g. gear ratio, encoder configuration, direction of rotation[4], etc.)

[2] Technology object
[3] XERR, KPC
[4] Change of direction by p1821 in the expert list of the drive (if needed)

- Direction of speed and torque must be the same in the corresponding drives.
- Motors must be identical (i.e. same reference speed (p2000) / torque (p2003), same maximum speed (p1082[0]) / torque (p1520)).

NOTE The control of the drives must be optimized in advance!

4.1.4 Mechanical system

The motors involved in the load balancing must be connected via a suitable mechanical system. In general, this means that a stiff mechanical connection between the motors is mandatory. This can be achieved either by coupling the motors through a shaft or driving a common gearbox.

4.1.5 Exception handling

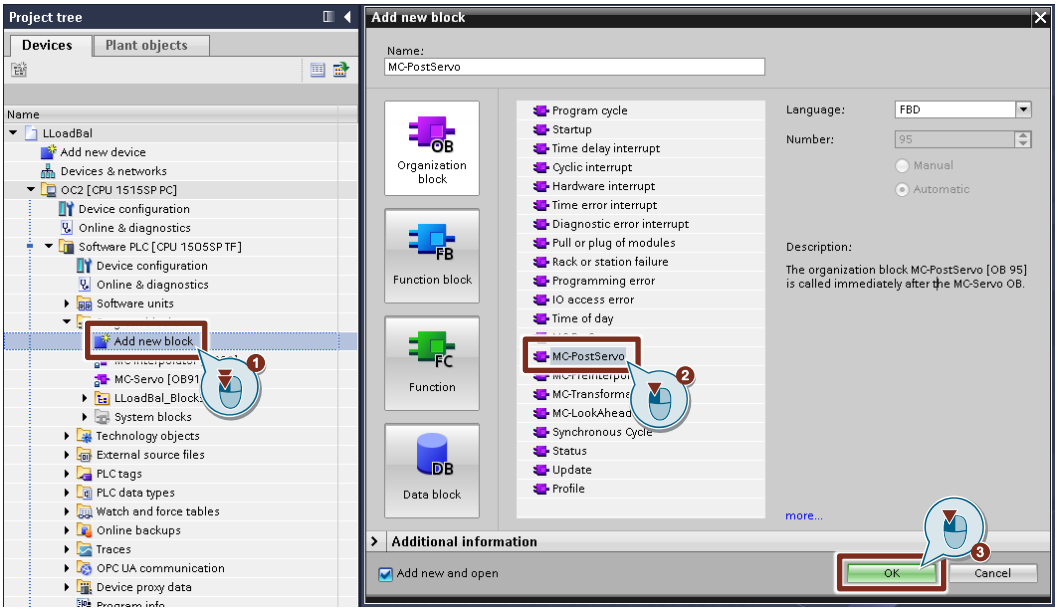
NOTICE **Capturing and reacting to errors of the corresponding axes is not part of the library.**

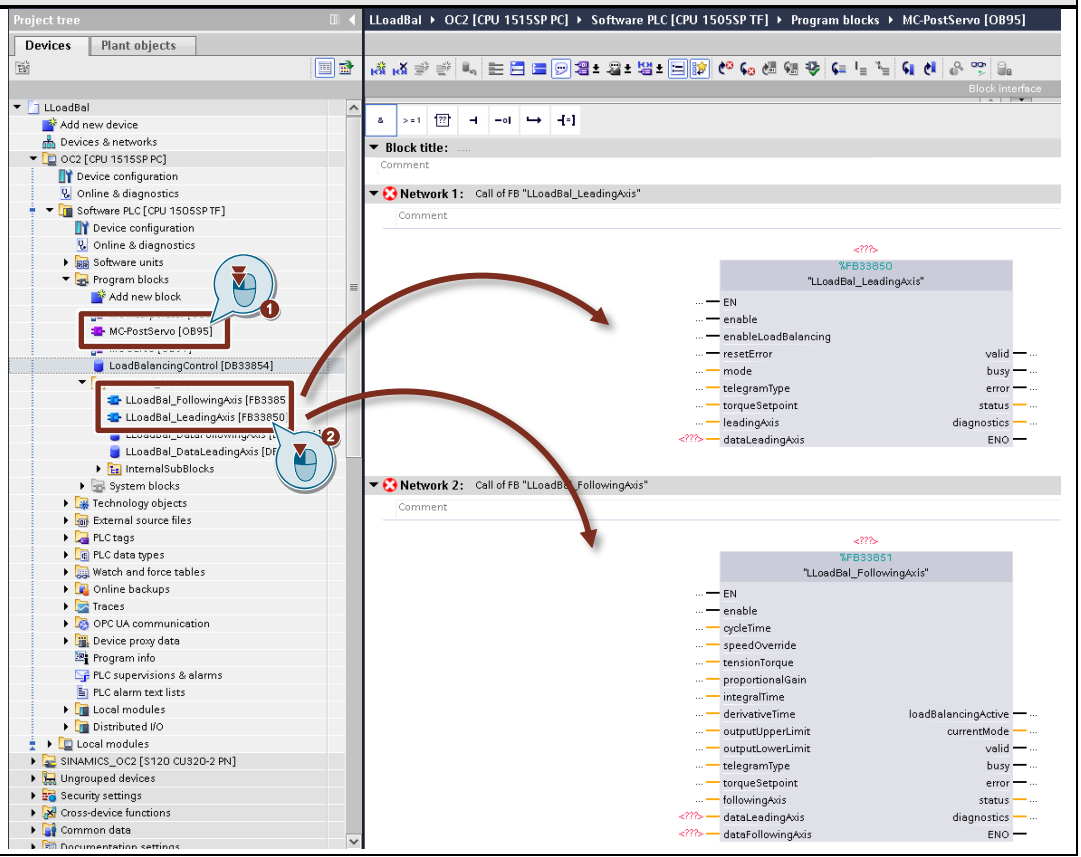
Due to the variety of possibilities an appropriate exception handling must be covered within the certain application using the functionality provided by the library.

4.2 Calling the function blocks

The function blocks “LLoadBal_LeadingAxis” and “LLoadBal_FollowingAxis” must be cyclically called in the MC-PostServo (OB95) organization block.

Table 4-1 Calling the function blocks

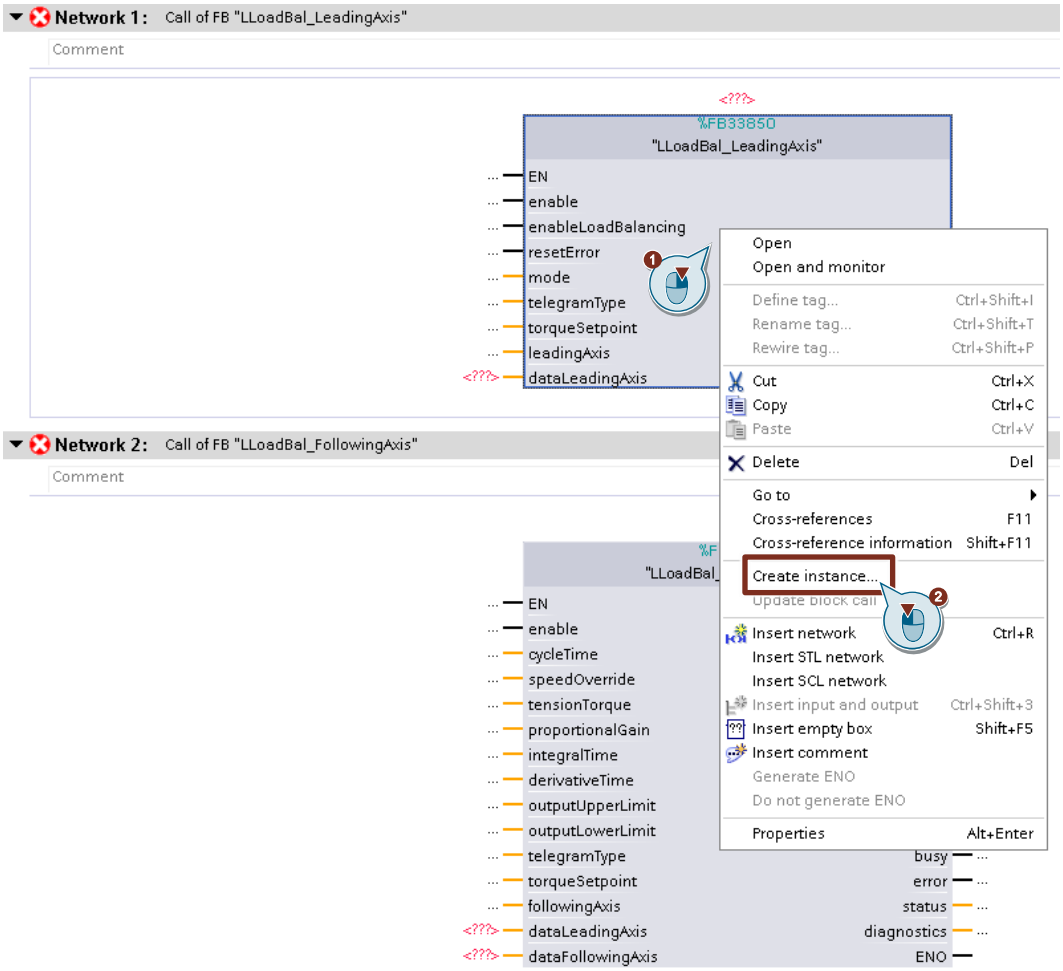
No.	Action
1.	If not existing in the project add the OB95 to the “Program blocks” folder of the certain PLC. 
2.	Open the OB95 and insert the function block “LLoadBal_LeadingAxis” by drag and drop to the first empty network (e.g. “Network 1”). For inserting the function block “LLoadBal_FollowingAxis” proceed in the same way.

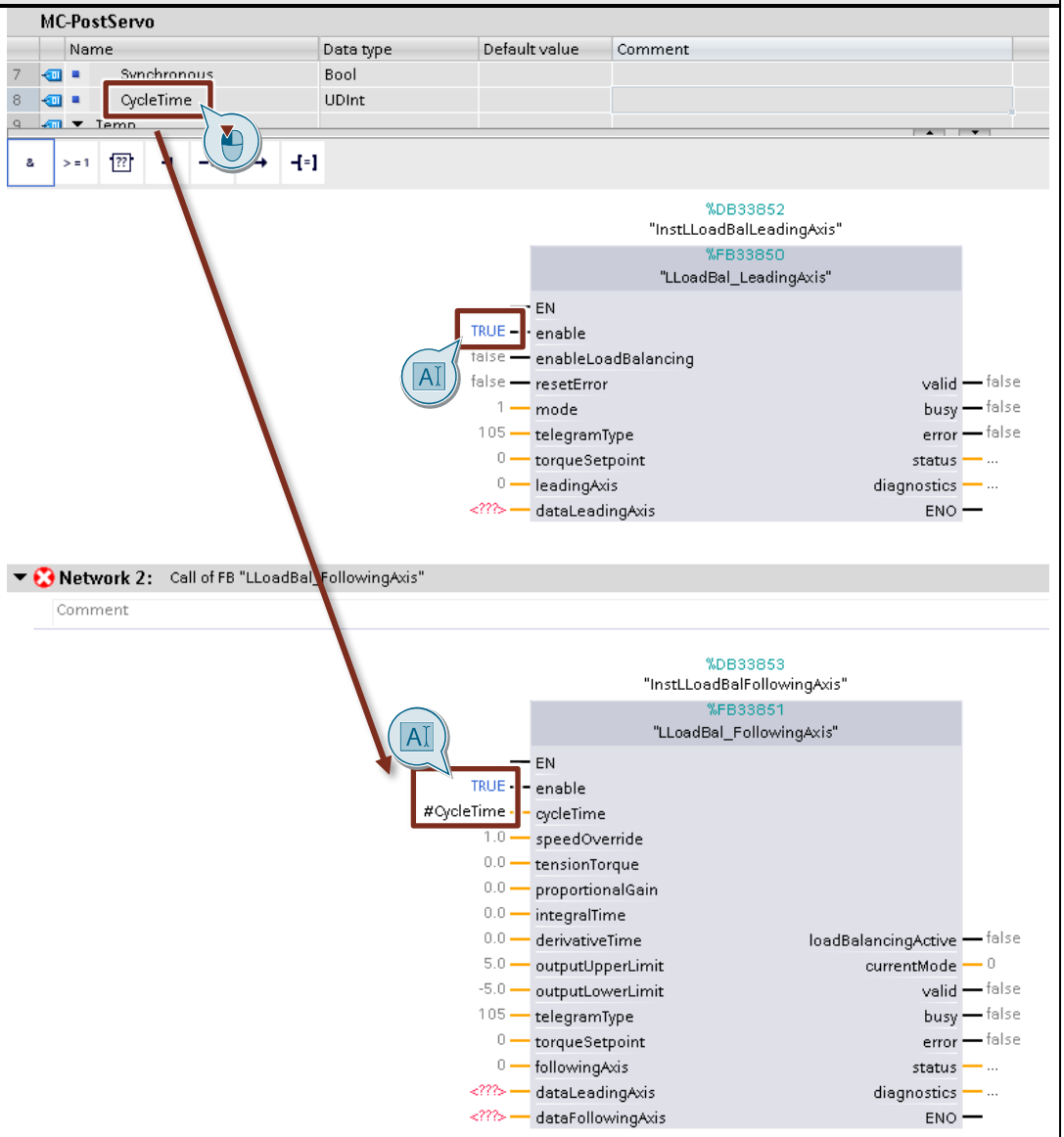
No.	Action
	

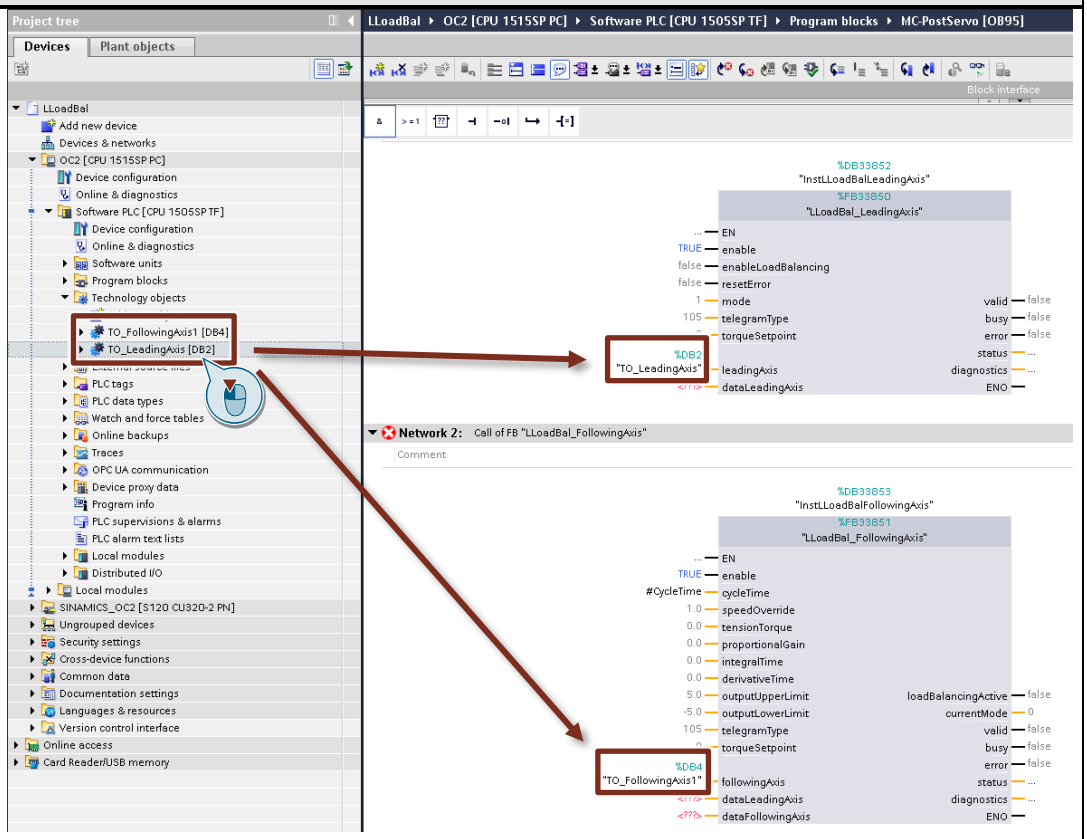
NOTE

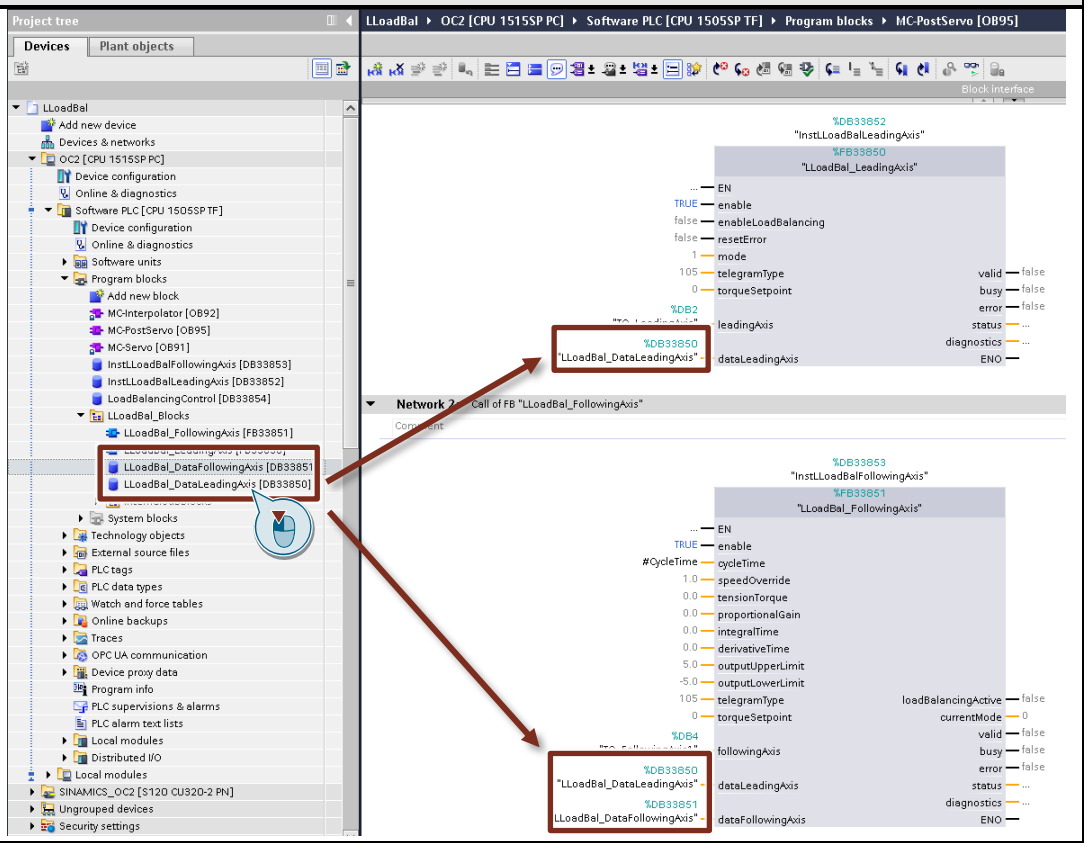
If more than one leading / following axis being part of a leading-following relation the certain function block must be inserted as often as required.

Please make sure to verify the correct order of calling the function blocks. All instances of the function block “LLoadBal_LeadingAxis” must be called before the instances of the function block “LLoadBal_FollowingAxis”.

No.	Action
3.	After inserting the required number of function blocks a separate instance for each must be created. To do so right-click on the function block in the certain network and choose the option “Create instance”, specify a unique name and confirm your choice. 
4.	By sending a (static) TRUE signal to input “enable” the certain function block gets enabled in general. For internal purpose the function block “LLoadBal_FollowingAxis” must know the cycle time it gets called. Therefore OB MC-PostServo provides this information in the start information variable “#CycleTime”, which must be assigned to the input “cycleTime” of the function block.

No.	Action
	
5.	The technology object of the axis (see chapter 4.1.1) representing the leading axis must be assigned to the input “leadingAxis” of the function block “LLoadBal_LeadingAxis”. Further the axis representing the following axis must be assigned to the input “followingAxis” of the function block “LLoadBal_FollowingAxis”.

No.	Action
	
6.	Linking of both function blocks and therefore both axes is done by the data structure “LLoadBal_DataLeadingAxis” (part of the library), which must be assigned to the input “dataLeadingAxis” of function block “LLoadBal_LeadingAxis” as well as “LLoadBal_FollowingAxis”. If more than one leading axis is existing the data structure “LLoadBal_DataLeadingAxis” must be copied and renamed, so that for each leading axis a unique data structure is available, which then can be distributed to the corresponding following axes. The data structure “LLoadBal_DataFollowingAxis” (part of the library), which must be connected to input “dataFollowingAxis” of function block “LLoadBal_FollowingAxis” is mandatory but is not further used in terms of load balancing. For every following axis a separate data structure is necessary.

No.	Action
	

NOTE

"Linking" of the function blocks means the assignment of the certain following axis to its leading axis (i.e. defining the group of axes where the load should be distributed symmetrically).

Each following axis must only be assigned to one leading axis (1-1 relation), whereas one leading axis can have several following axes assigned (1-n relation).

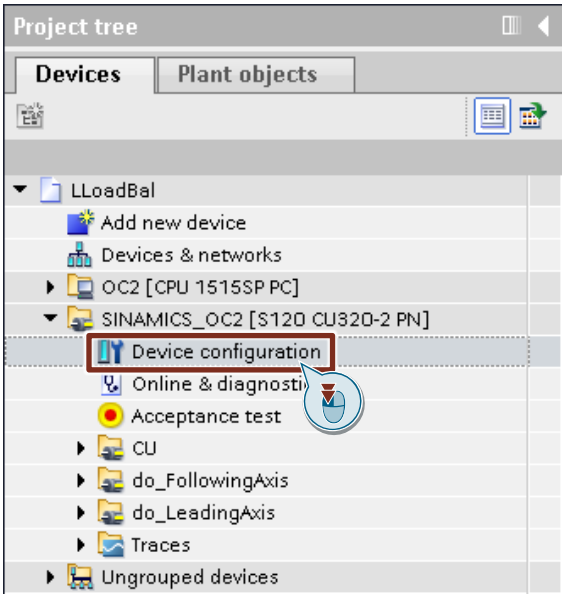
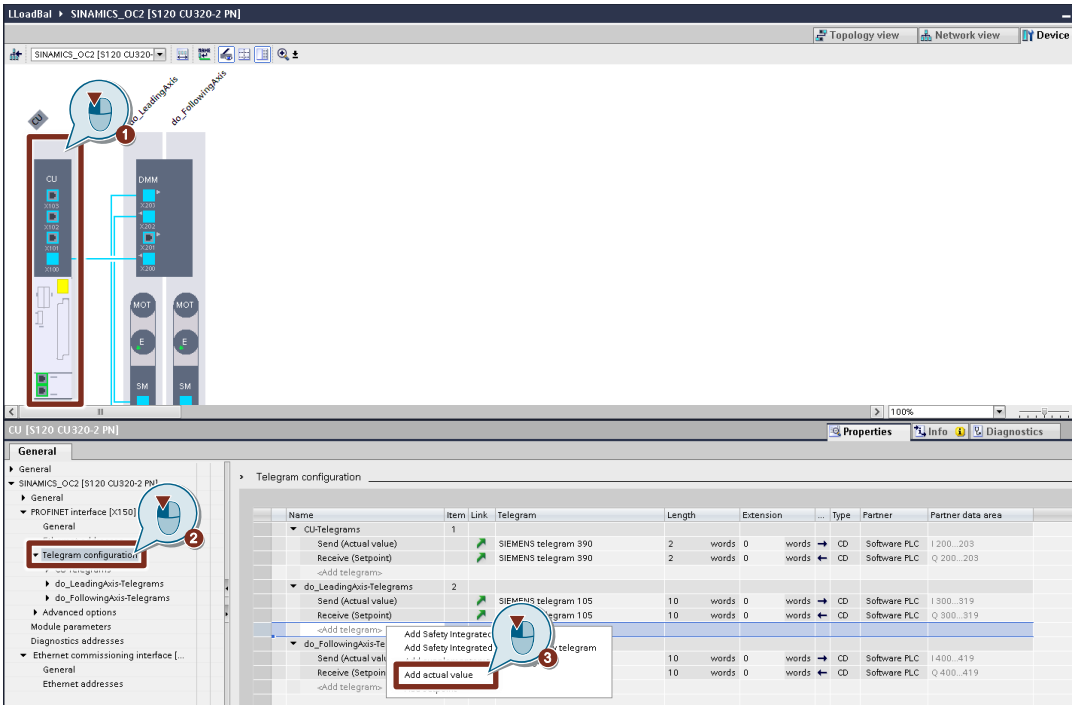
This relation must be observed by the user when setting up the program!

4.3 Adapting the drive telegram

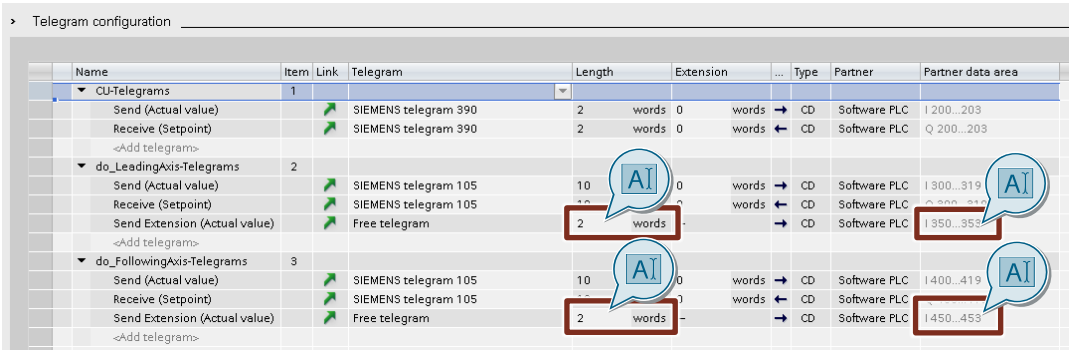
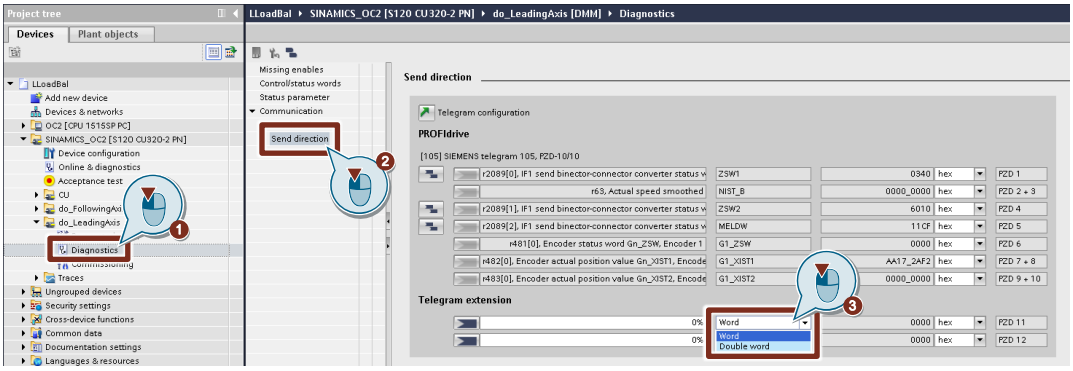
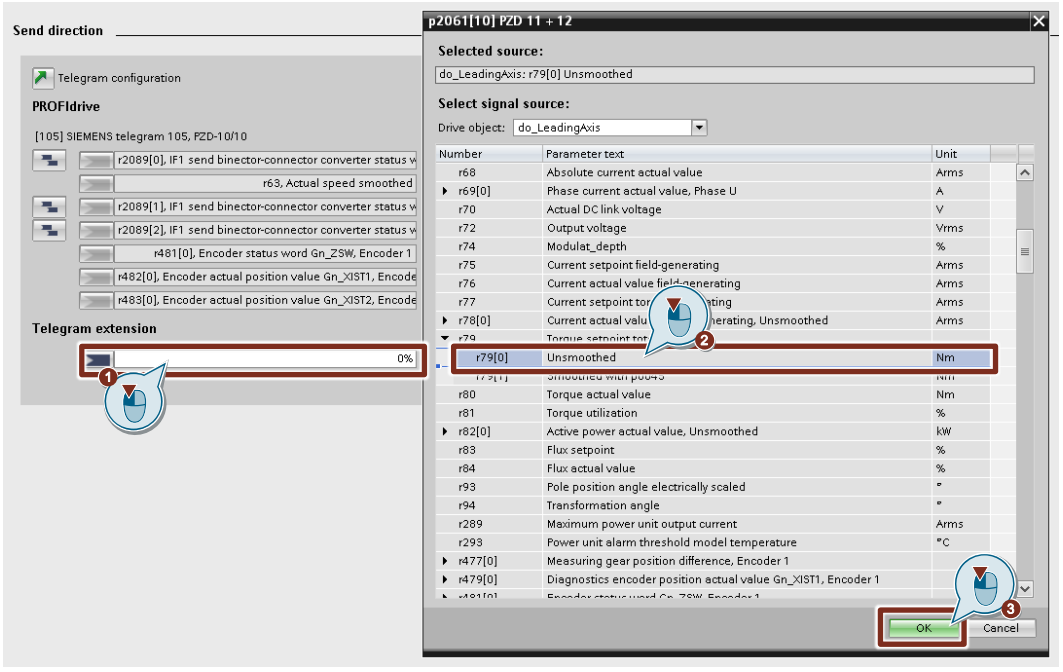
As the load balancing is based on comparing the torque setpoints of the motors that should operate together the setpoint values must be present in the PLC for further processing.

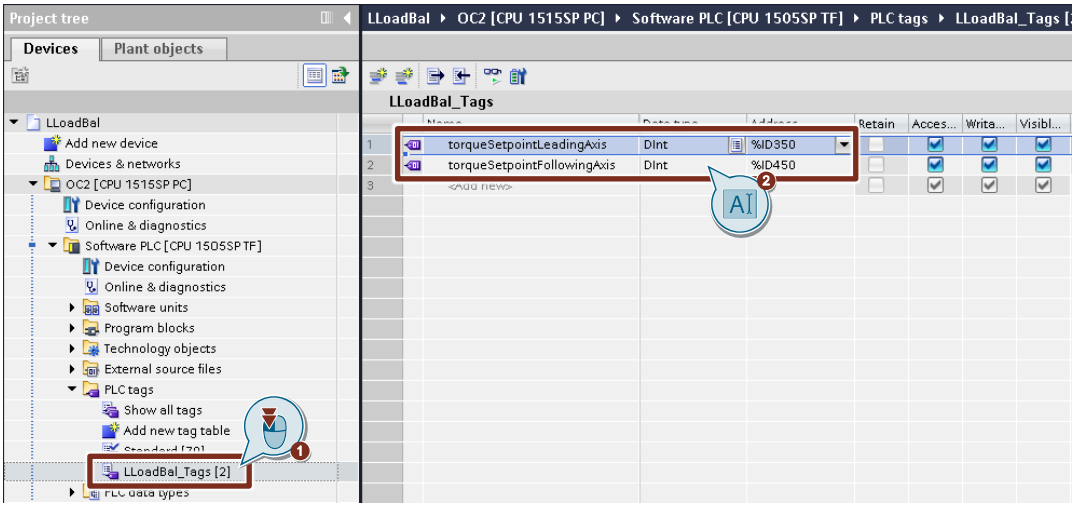
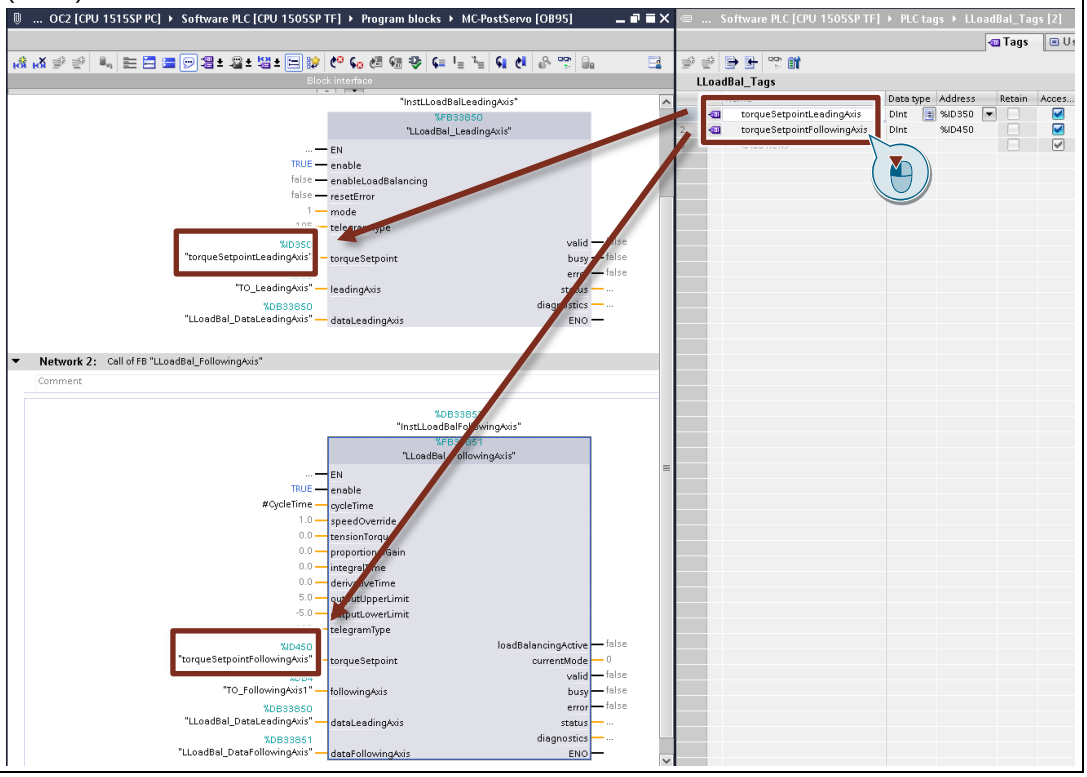
For this purpose, the certain drive telegram must be extended by an additional actual value.

Table 4-2 Adapting the drive telegram

No.	Action
1.	Open the device configuration of the SINAMICS control unit in TIA Portal. 
2.	In the properties of the control unit navigate to the option “Telegram configuration” and add a free telegram (actual value) to the leading DO[5] as well as to the following DOs being involved in the load balancing. 

[5] Drive object

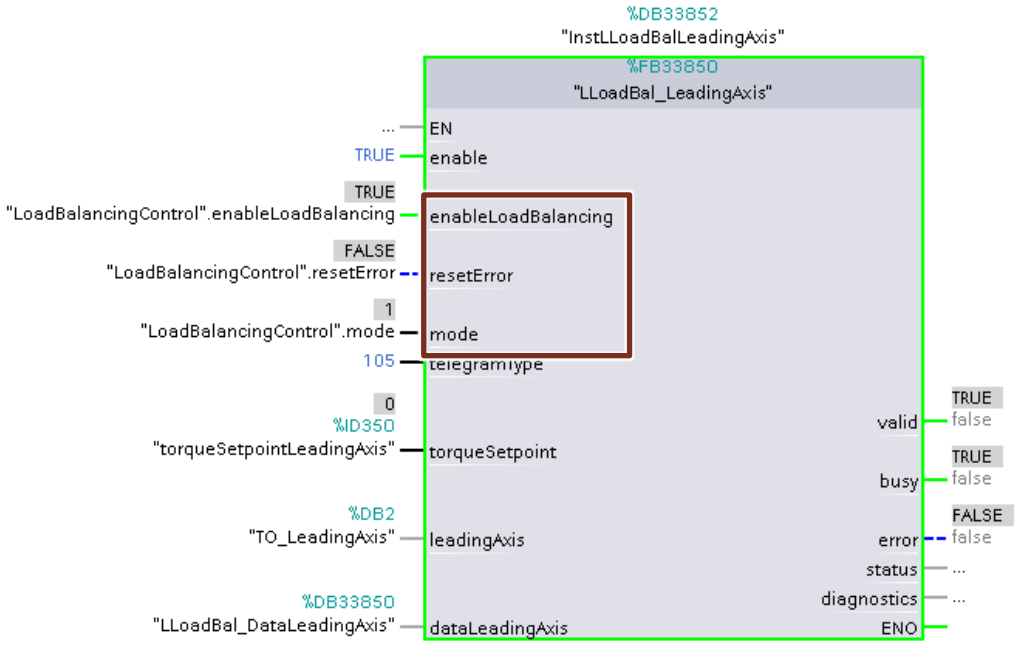
No.	Action
3.	The length of the free telegram must be changed to two words. If needed the starting address of the telegram can be adapted (optional). 
4.	In the “Diagnostics” overview of the certain DO the two words added before must be combined to one double word. 
5.	The double word then must be interconnected to parameter r79[0] (Torque setpoint total, Unsmoothed) of the certain DO. 

No.	Action
6.	For accessing the torque setpoint values by the PLC program add a PLC tag for each value in the “PLC tags” folder, using the starting address of the free telegram added in the device configuration of the control unit before. The data type of the PLC tags must be changed to “Dint”. 
7.	The PLC tags must then be assigned to the input “torqueSetpoint” of the certain instance of the function blocks “LLoadBal_LeadingAxis” and “LLoadBal_FollowingAxis” in OB MC-PostServo (OB95). 

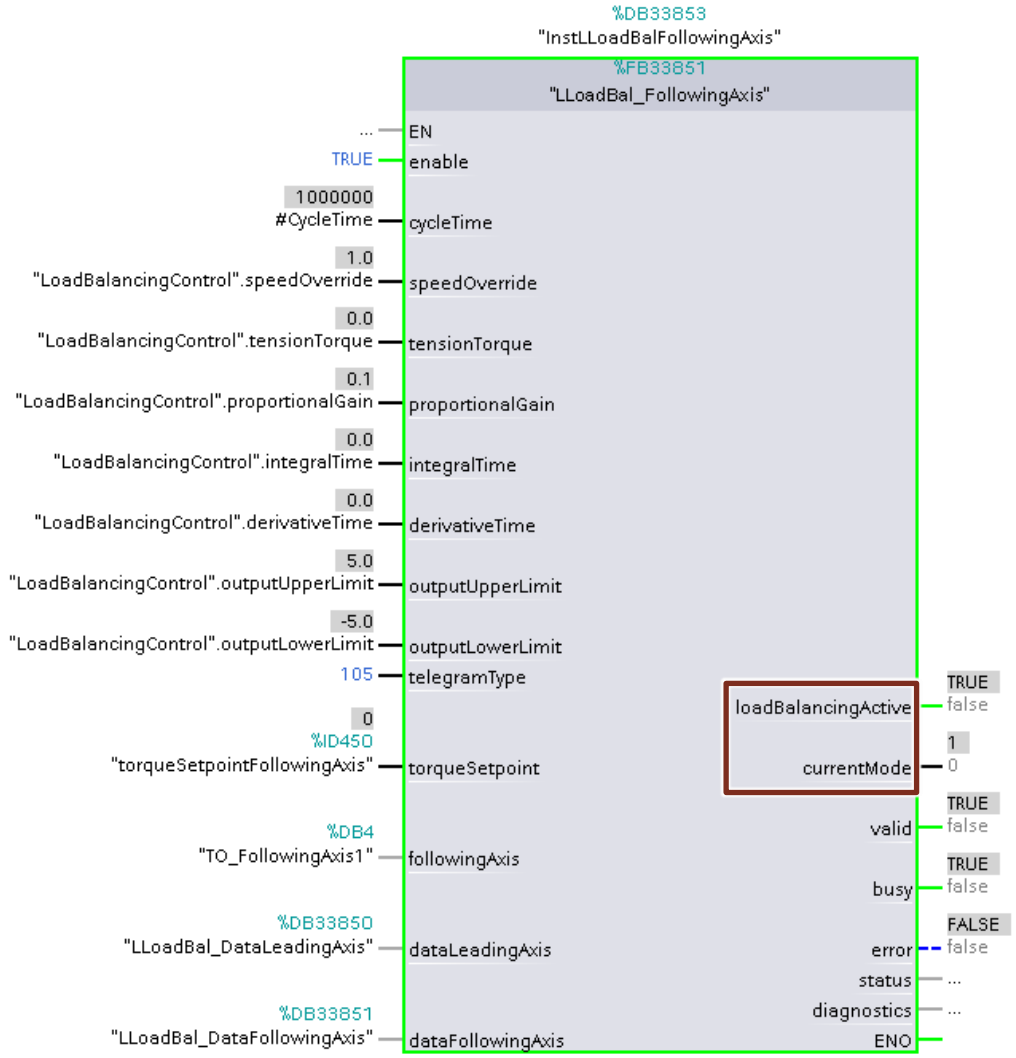
4.4 Enabling and adjusting the load balancing

After finishing the network configuration in OB MC-PostServo (OB95) (see chapter 4.2) the load balancing must be enabled for evaluating a suitable setting of the load balancing controller parameters.

Table 4-3 Enabling and adjusting the load balancing

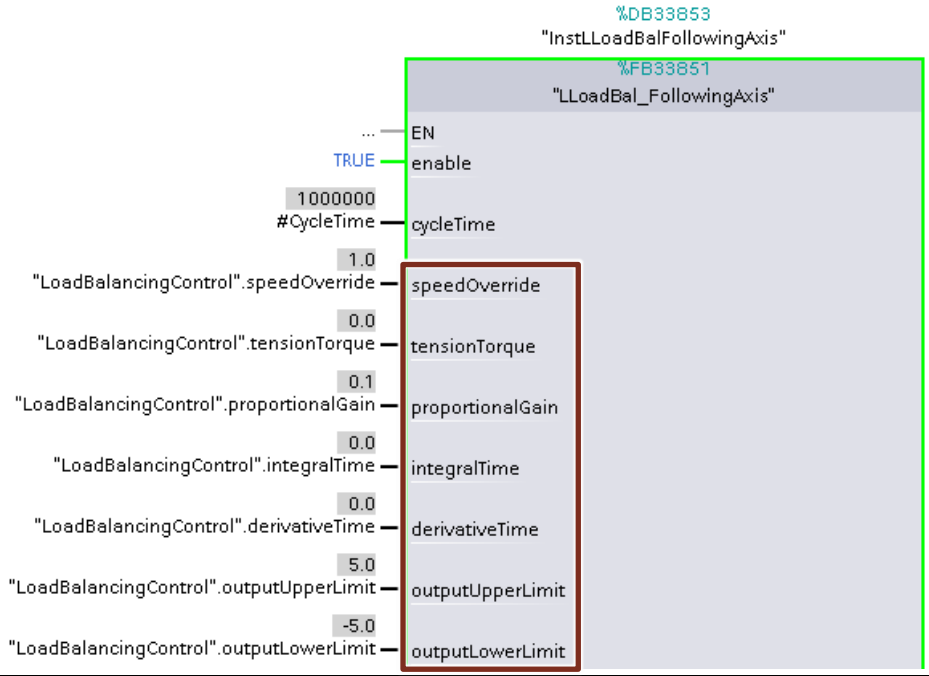
No.	Action
1.	Enabling / Disabling the load balancing as well as changing the mode is only possible if the leading axis is in standstill. If the leading axis is moving changing the input state has no effect on the current processing of the function block. However, changes on input side will be recognized and get effective as soon as the leading axis reaches standstill state. Enabling / Disabling is done via input “enableLoadBalancing”. Depending on the mode that has been specified via input “mode” (preset value = 1) all following axis function blocks linked to the certain leading will be forced to change to the selected mode. For resetting errors of the leading axis function block and its assigned following axis function blocks a rising edge on input “resetError” is required. 

No.	Action
2.	If the load balancing is active for the certain following axis this is signalized by the output “loadBalancingActive” of the following axis function block. The load balancing gets active if the correct mode has been selected (i.e. “mode” = 1), a proportional gain > 0.0 has been set, no errors are pending, and the leading and following axis are both enabled.

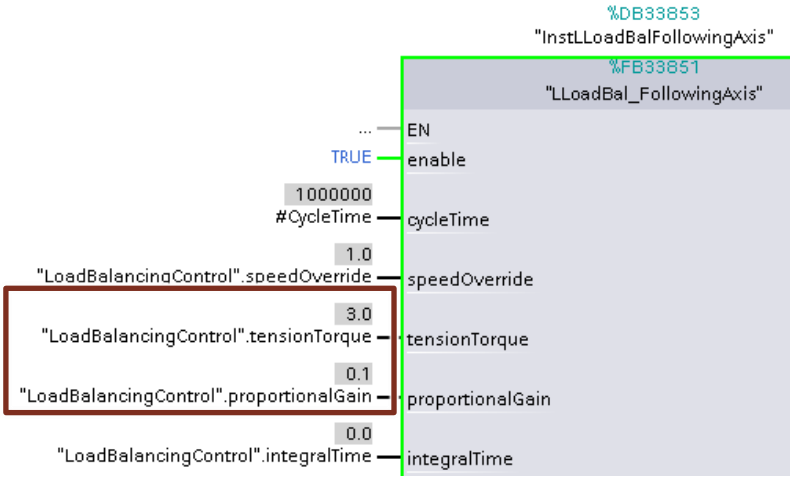


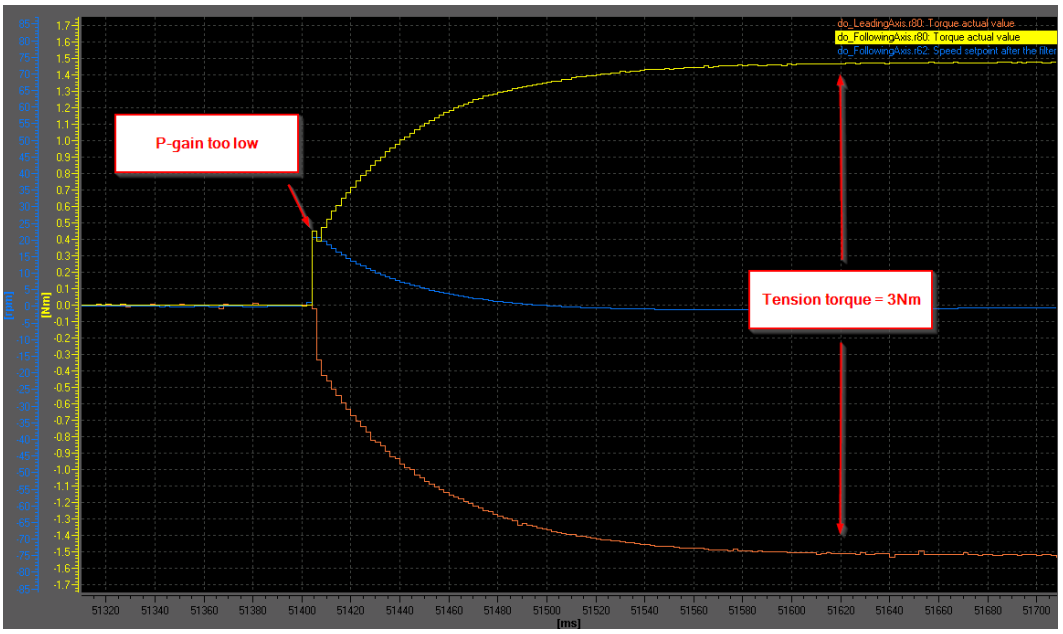
The current mode of the certain following axis function block is shown at output “currentMode”.

- **Idle (0)**
  The function block waits for the command to change to another mode. The load balancing functionality is disabled. This mode is accessed if the load balancing has been disabled (“enableLoadBalancing” = FALSE), mode “Idle” (“mode” = 0) has been specified or an error (“error” = TRUE) occurred at the leading axis function block.
- **Additional speed setpoint (1)**
  If the load balancing is active (“loadBalancingActive” = TRUE) the leading axis setpoints (speed and DSC setpoint values) get distributed to the certain following axis and an additional speed setpoint gets calculated, which is based on deviations between the torque setpoint values of leading and following axis. This mode is accessed if mode “Additional speed setpoint” (“mode” = 1) has been specified and load balancing has been enabled (“enableLoadBalancing” = TRUE).
- **User defined (2)**
  Spare mode that can be adapted by the user (if required). The load balancing functionality is disabled by default. This mode is accessed if mode “User defined” (“mode” = 2) has been specified and load balancing has been enabled (“enableLoadBalancing” = TRUE).

No.	Action
3.	The following axis function block provides the possibility to specify the load balancing controller parameters via the inputs “proportionalGain”, “integralTime” and “derivativeTime”, whereas the integral and derivative part of the controller is optional to use. In addition, the controller result (i.e. additional speed setpoint for the following axis) can be limited via the inputs “outputUpperLimit” and “outputLowerLimit”. The limits are preset with values of +5 / -5 rpm and must be adapted during set up of the load balancing controller (if necessary). Further it’s possible to set a static tension torque via input “tensionTorque”, which is applied as constant torque difference between leading and following axis. The tension torque is used as well for setting up the load balancing controller like described in the following. In case of need the speed setpoint of the following axis can be adapted using the input “speedOverride”. Default setting of the override is 1.0, which means: speed setpoint of the following axis = 1.0 * speed setpoint of the leading axis. 
4.	Setting up the load balancing controller basically means adjusting the proportional gain so that deviations between the torque setpoint values of leading and following axis are regulated fast enough. Usually, using a pure P-controller is sufficient for this purpose. At the beginning a small proportional gain should be set (e.g. **0.1** or even less) to get a first impression of the current situation. The value must be adjusted step by step until a sufficient setting has been found. To evaluate the adjustment a SINAMICS trace should be configured, which contains the following parameters: • LeadingAxis.r80: Actual torque value • FollowingAxis.r80: Actual torque value • FollowingAxis.r62: Speed setpoint

No.	Action
5.	While the leading and following axis being enabled and possible motor holding (or external) brakes are opened, a setpoint jump must be applied to the load balancing controller. This is possible by using input “tensionTorque” of the following axis function block.





In the trace you can see, that a setpoint jump is given to the following axis in terms of a **speed setpoint value** being the output of the load balancing controller. The tension torque of 3Nm applied is reached approximately after 250ms, having a symmetrical distribution of +/- 1.5Nm of the **leading** and **following axis actual torque**. Though, the proportional gain can still be increased to shorten the regulation time. The goal is to have the regulation time as short as possible while the jump of the following axis actual torque at the beginning does not exceed the final value when reaching the tension torque applied.

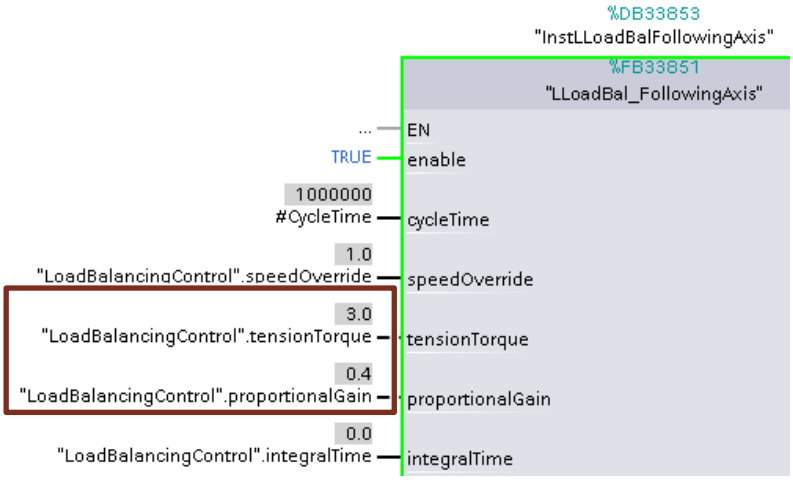
NOTE

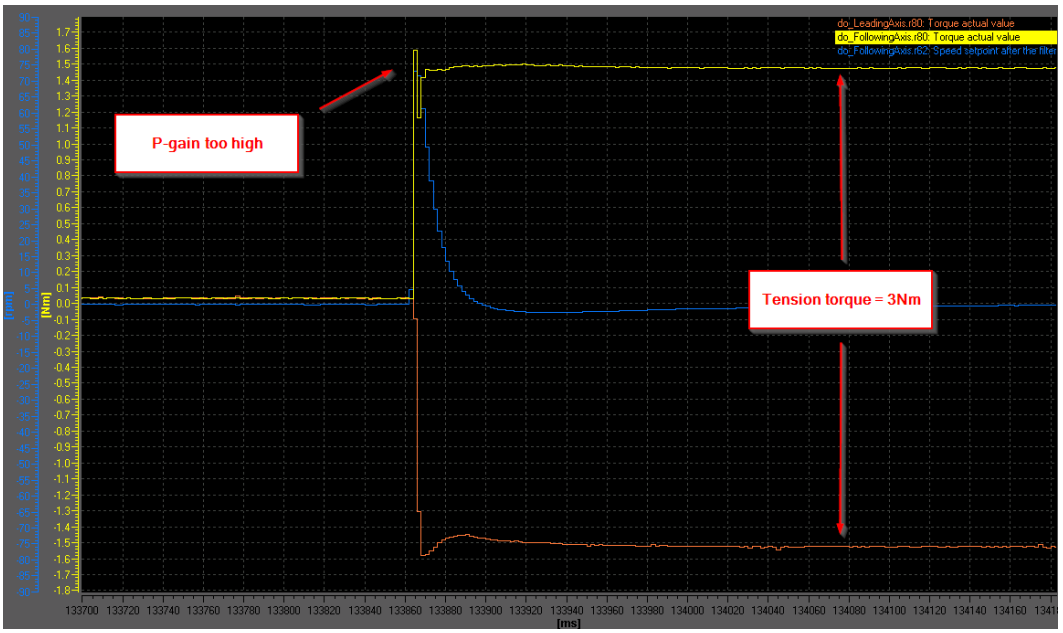
The tension torque applied when setting up the load balancing controller is depending on the size of the motors being involved. The bigger the motors, the higher the tension torque can be. The mechanical connection must be considered as well.

While applying / resetting the tension torque a slight movement of the corresponding axes might be possible. As soon as the tension torque is reached the axes should remain in standstill.

Before adjusting the proportional gain, the tension torque should always be reset to 0.0Nm before!

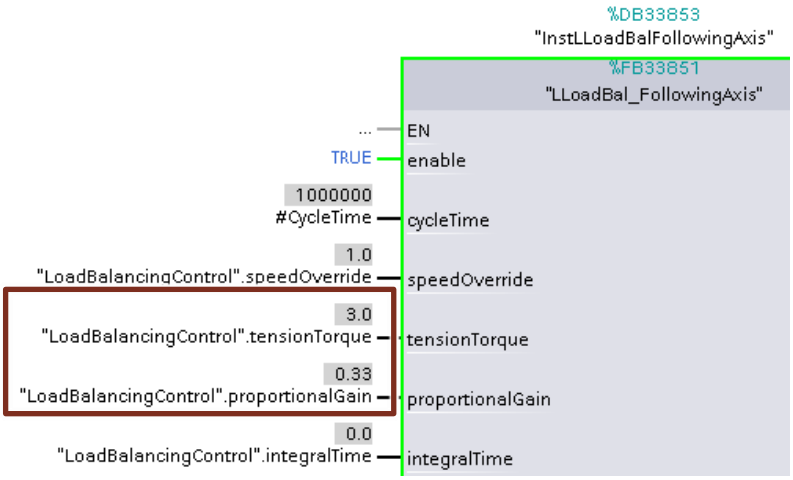
No.	Action
6.	In a next step the proportional gain is increased to 0.4.

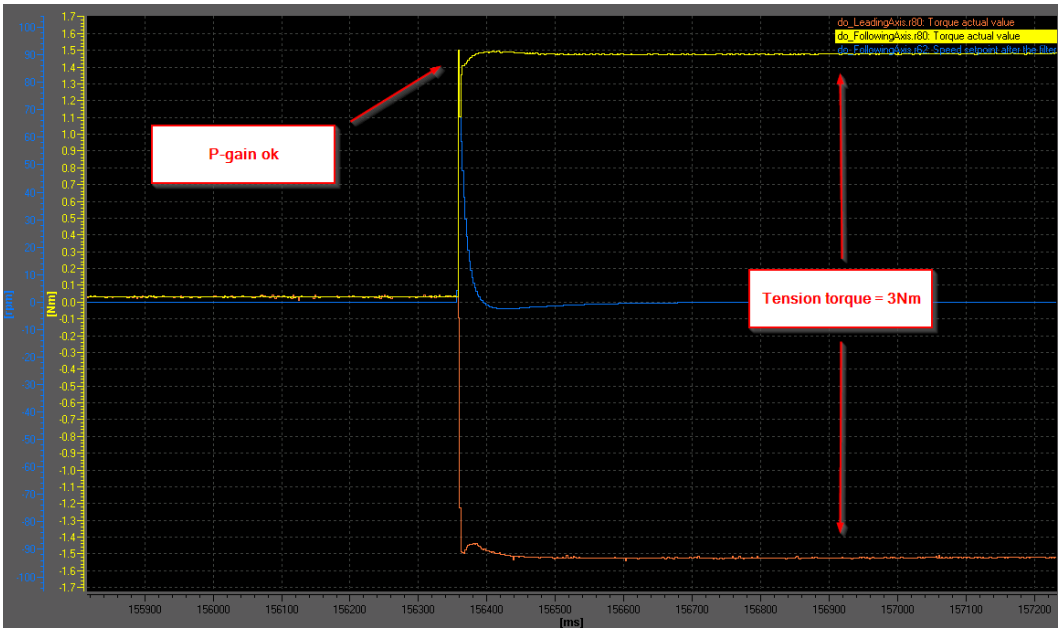




The **speed setpoint value** applied to the following axis is higher, therefore the regulation time is decreased a lot (~ 25ms). Though the proportional gain is too high, as the jump of the **following axis actual torque** value at the beginning exceeds the final value of 1.5Nm.

No.	Action
7.	The proportional gain must be slightly decreased. Therefore, the next value will be 0.33.





The regulation time is still short enough (~ 50ms), while the jump of the **following axis actual torque** does not exceed the final value. The proportional gain is suitable.

NOTE

If more than one following axis is assigned to a leading axis the steps described above must be repeated for the remaining following axes as well.

After finishing the adjustment of each load balancing controller, the results should be verified by operating the motors together. The actual torque values should be identical, sharing the load symmetrically.

5 Operation

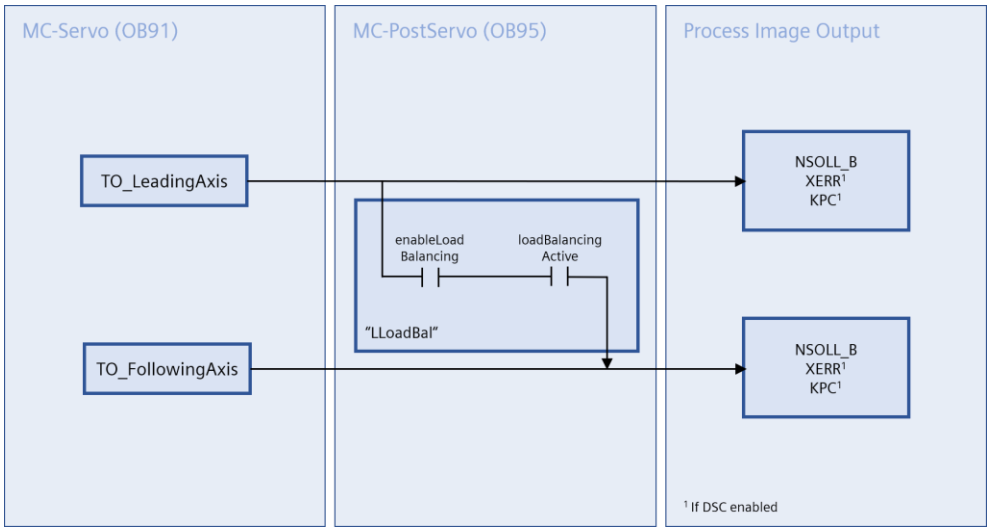
5.1 Distribution of setpoint values

In the MC-Servo (OB91) the position controller of each axis is calculated. If the load balancing functionality is disabled, the setpoint values are directly send to the corresponding drive.

If the load balancing functionality gets enabled (“enableLoadBalancing”), the setpoint values of the following axis are overwritten in the MC-PostServo (OB95), resulting in the leading axis setpoint values being send to the following axis drive as soon as load balancing gets active (“loadBalancingActive”).

The setpoint values being distributed comprise the speed setpoint value (NSOLL_B) as well as the setpoint values for DSC[6] (if enabled). Here the speed setpoint value of the following axis consists of the leading axis speed setpoint and the additional speed setpoint being output by the load balancing controller.

Figure 5-1: Distribution of setpoint values



NOTE

If the following axis gets enabled position-controlled with following error monitoring being active this will lead to an axis error in case of load balancing is enabled and the axis starts to move by receiving the leading axis speed setpoint value.

In this case the actual position of the following axis increases while the setpoint position doesn’t change, which results in the maximum following error of the axis getting exceeded. To prevent this situation there are two possible solutions:

1. Enabling of the following axis non-position-controlled (MC_Power > StartupMode = 0)
2. Use (1:1) gearing with the leading axis (MC_GearIn)
   Advantage: Following axis keeps on moving even if load balancing functionality faults.

[6] Dynamic Servo Control

6 Appendix

6.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:

support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts.

Please send queries to Technical Support via Web form:

support.industry.siemens.com/cs/my/src

SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:

siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:

support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for iOS and Android:

support.industry.siemens.com/cs/ww/en/sc/2067

6.2 Industry Mall

The Siemens Industry Mall is the platform on which the entire siemens Industry product portfolio is accessible. From the selection of products to the order and the delivery tracking, the Industry Mall enables the complete purchasing processing – directly and independently of time and location:
mall.industry.siemens.com

6.3 Application support

Siemens AG
Digital Factory Division
Factory Automation
Production Machines
DF FA PMA APC
Frauenauracher Str. 80
91056 Erlangen, Germany

mailto: tech.team.motioncontrol@siemens.com

6.4 Links and literature

Table 6-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to this entry page of this application example https://support.industry.siemens.com/cs/ww/en/view/109794291

6.5 Change documentation

Table 6-2

Version	Date	Modifications
V1.0	03/2021	First version