

A man in a light blue shirt is seen from the side, holding a tablet. He is in a factory environment with various industrial equipment and a clock in the background. Overlaid on the scene are several digital graphics: a '24/7' icon with a circular arrow, a 'NEWS' section with a person icon, a 'Home' button, and a network diagram with three people icons. The overall theme is industrial digitalization and online support.

**SIEMENS**

*Ingenuity for life*

## **SINAMICS G: Load sharing for two coupled drives**

SINAMICS G / Load sharing / V1.2

<https://support.industry.siemens.com/cs/ww/en/view/109774232>

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

|                                                       |           |
|-------------------------------------------------------|-----------|
| <b>Legal information .....</b>                        | <b>2</b>  |
| <b>1 Introduction .....</b>                           | <b>4</b>  |
| 1.1 Components Used .....                             | 5         |
| 1.2 Integration into the user project .....           | 6         |
| <b>2 Control modes .....</b>                          | <b>9</b>  |
| 2.1 Torque coupling .....                             | 9         |
| 2.2 Speed override with torque limit .....            | 10        |
| 2.3 Droop and compensation .....                      | 10        |
| <b>3 Configuring the drives .....</b>                 | <b>13</b> |
| 3.1 Telegram .....                                    | 13        |
| 3.1.1 Structure .....                                 | 13        |
| 3.1.2 Interconnection .....                           | 13        |
| 3.2 Parameter .....                                   | 14        |
| <b>4 Description of the library .....</b>             | <b>16</b> |
| 4.1 Contents .....                                    | 16        |
| 4.2 LLoadSharing_Control .....                        | 17        |
| 4.2.1 Description .....                               | 17        |
| 4.2.2 Calling OBs .....                               | 18        |
| 4.2.3 Input parameters .....                          | 18        |
| 4.2.4 Output parameters .....                         | 18        |
| 4.3 LLoadSharing_DriveCommunication .....             | 19        |
| 4.4 PLC data types .....                              | 19        |
| 4.4.1 LLoadSharing_typeDriveSettings .....            | 19        |
| 4.4.2 LLoadSharing_typeDriveStatus .....              | 20        |
| 4.4.3 LLoadSharing_typeDriveInfo .....                | 20        |
| 4.5 MasterDriveInfo and SlaveDriveInfo .....          | 21        |
| 4.6 LLoadSharing_CyclicInterrupt .....                | 21        |
| <b>5 Selecting the appropriate control mode .....</b> | <b>23</b> |
| 5.1 Reference run .....                               | 23        |
| 5.2 Traversing with load sharing .....                | 24        |
| 5.3 Torque coupling .....                             | 24        |
| 5.4 Speed overdrive with torque limit .....           | 24        |
| 5.4.1 Opportunities for optimization .....            | 26        |
| 5.5 Droop and compensation .....                      | 28        |
| <b>6 Operation .....</b>                              | <b>30</b> |
| 6.1 Single traversing .....                           | 30        |
| 6.2 Load sharing .....                                | 30        |
| 6.2.1 Speed difference monitoring .....               | 30        |
| 6.2.2 Changing load sharing factor .....              | 31        |
| 6.3 Error handling .....                              | 32        |
| <b>7 Appendix .....</b>                               | <b>33</b> |
| 7.1 Error codes from DPWR_DAT and DPRD_DAT .....      | 33        |
| 7.2 Load sharing control word .....                   | 33        |
| 7.3 Importing trace configuration .....               | 34        |
| 7.4 Service and support .....                         | 35        |
| 7.5 Links and literature .....                        | 36        |
| 7.6 History .....                                     | 36        |

# 1 Introduction

Load sharing must be implemented whenever two or more motors are mechanically coupled with one another. A mechanical coupling can be rigid, flexible or separable. Common types of coupling are for example:

- shared shaft - can be rigid or torsionable
- belt - can be oscillating
- mating roll - firmly coupled through friction contact from material web, possible slippage or lose connection
- vehicle with at least two drives

If the motors are controlled without load sharing, they can work against each other and even start oscillating.

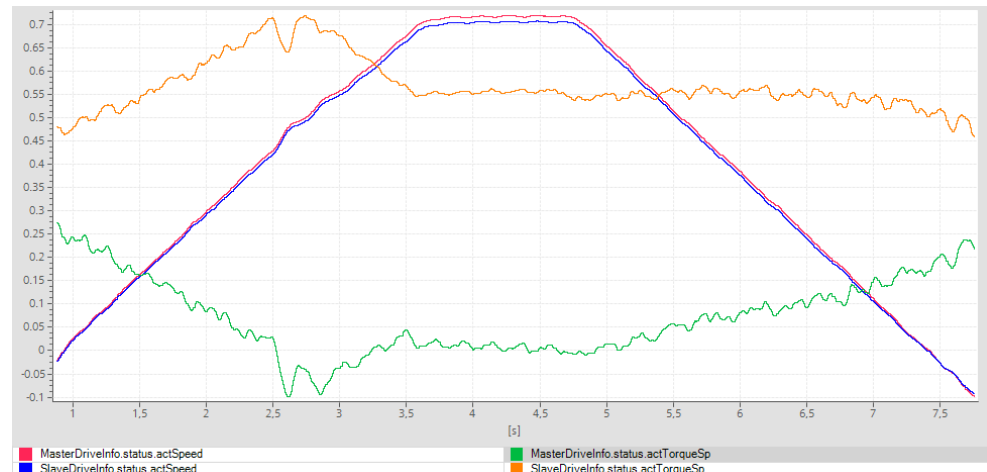
The "LLoadSharing\_Control" function block is used for controlling two mechanically coupled motors. The function block supports not only traversing of both drives with a configurable load sharing but also traversing of one individual drive.

The following control modes are available for load sharing:

- Torque coupling
- Speed override with torque limit
- Droop and compensation

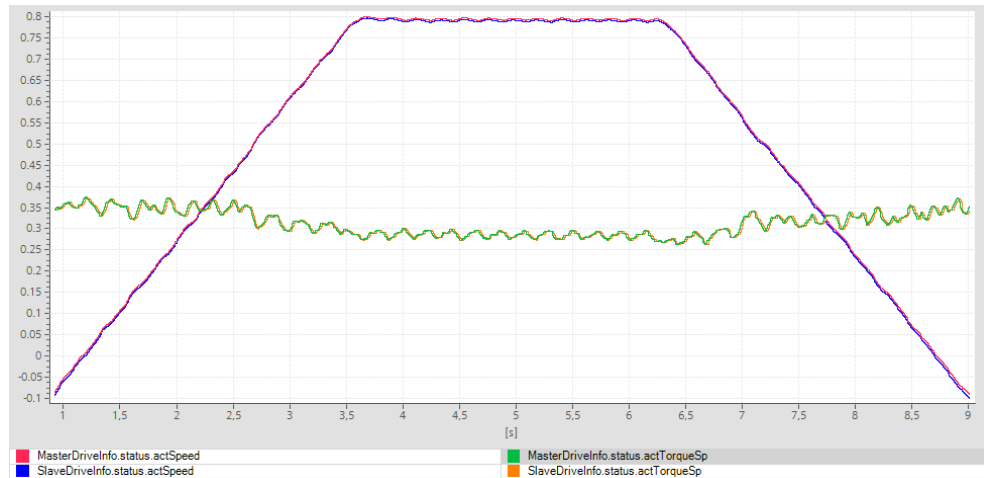
In [Figure 1-1](#) the torques of the drives are opposite to each other. The same speed setpoint is forced by the mechanical coupling but both drives work against each other.

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



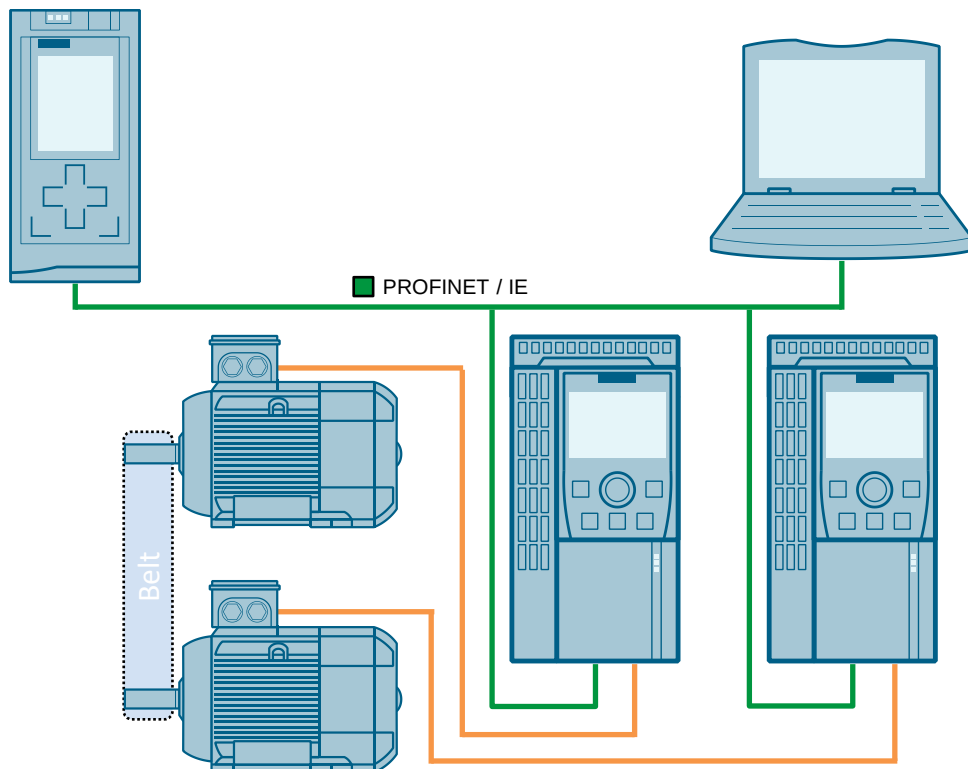
In [Figure 1-2](#) both torques are identical. The drives work together and share the load.

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



## 1.1 Components Used

Figure 1-3 Hardware configuration



The library "LLoadSharing" has been developed for S7-1200/S7-1500 CPUs and SINAMICS G120 drives.

This application example was created with the following soft- and hardware components:

Table 1-1 Used hardware components

| Components                  | Quantity | Article number      |
|-----------------------------|----------|---------------------|
| SIMATIC S7 1513-1 PN        | 1        | 6ES7 513-1AL01-0AB0 |
| SINAMICS G120 CU240E-2 PN-F | 2        | 6SL3244-0BB13-1FA0  |
| SINAMICS PM340              | 2        | 6SL3210-1SB14-0UA0  |
| Asynchronous motor          | 2        | 1LA7060-4AB10       |

**Note**

Other SINAMICS G Control Units may also be used for the application example, although not all support droop feedback. Verify that parameter p1488 "Droop input source" is available. If not, the control mode "Droop and compensation" is not usable in such case.

Tabelle 1-2 Used software components

| Components                      | Quantity | Article number     |
|---------------------------------|----------|--------------------|
| STEP 7 Professional V15.1       | 1        | 6ES7822-1..05-..   |
| SINAMICS Startdrive Basic V15.1 | 1        | 6SL3072-4FA02-0XA0 |

This application example consists of the following software components:

Table 1-3 Components of the Application Example

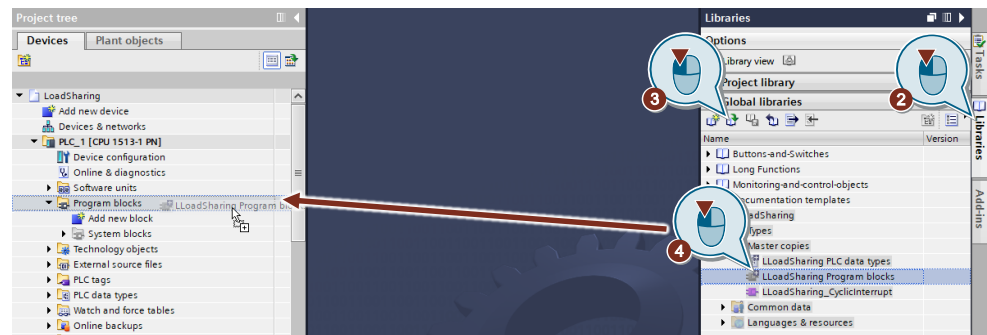
| Components                                     | File name                             |
|------------------------------------------------|---------------------------------------|
| Documentation                                  | 109774232_LoadSharing_DOC_V111_en.pdf |
| Function library                               | 109774232_LoadSharing_LIB_V11.zip     |
| Trace configuration (see <a href="#">7.3</a> ) | 109774232_LoadSharing_TRACES.zip      |

## 1.2 Integration into the user project

Proceed as follows to integrate the load sharing function blocks into a user program:

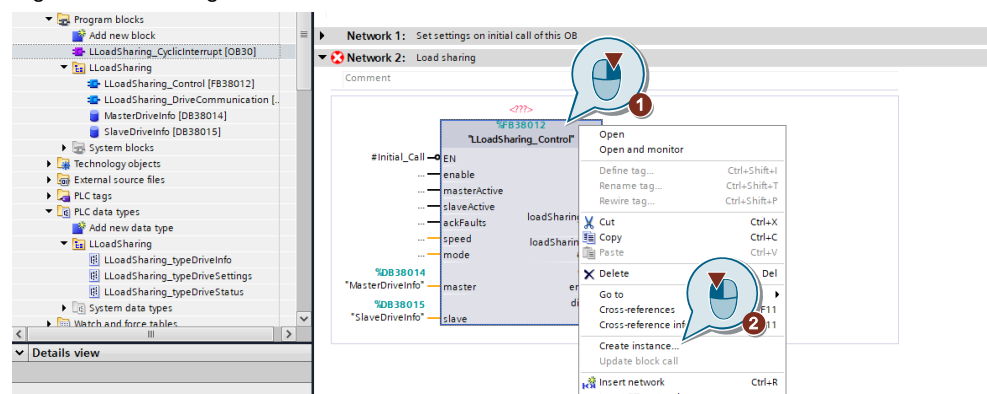
1. Download the "LLoadSharing" library (see [12](#)).
2. Open the "Libraries" task card.
3. Open the library in the palette "Global libraries".
4. Drag and drop the group "LLoadSharing Program Blocks" from the master copies into the program blocks folder of the opened project.

Figure 1-4 Integrating function blocks into the project



5. Drag and drop the group "LLoadSharing PLC data types" from the master copies into the "PLC data types" folder.
6. Optionally, you can insert the template of a cyclic interrupt organization block by dragging and dropping from the master copies into the Program blocks folder. The "LLoadSharing\_CyclicInterrupt" organization block is preconfigured with a cycle time of 8000  $\mu$ s.
7. Now open the inserted organization block and create an instance of the function block "LLoadSharing\_Control" and select "Create instance" in the context menu.

Figure 1-5 Creating an instance of the function block



8. Enter parameters for the necessary settings of both drives in Network 1 of the organization block:
  - maxSpeed: Maximum speed [p1082]
  - refSpeed: Reference speed [p2000]
  - refTorque: Reference torque [p2003]
  - initialRoundingOffTime: Ramp-function generator initial rounding-off time [p1130]

Figure 1-6 Parameterize settings

▼ Network 1: Set settings on initial call of this OB

Comment

```

1 // set the master/slave settings on initial call of this organization block
2 IF #Initial_Call THEN
3   "MasterDriveInfo".settings.HWIDSTW := "G120_1~PROFINET_interface~Free_telegram";
4   "MasterDriveInfo".settings.HWIDZSW := "G120_1~PROFINET_interface~Free_telegram";
5   "MasterDriveInfo".settings.refSpeed := 1500.0;
6   "MasterDriveInfo".settings.refTorque := 1.7;
7   "MasterDriveInfo".settings.initialRoundingOffTime := T#100ms;
8
9   "SlaveDriveInfo".settings.HWIDSTW := "G120_2~PROFINET_interface~Free_telegram";
10  "SlaveDriveInfo".settings.HWIDZSW := "G120_2~PROFINET_interface~Free_telegram";
11  "SlaveDriveInfo".settings.refSpeed := 1500.0;
12  "SlaveDriveInfo".settings.refTorque := 1.7;
13  "SlaveDriveInfo".settings.initialRoundingOffTime := T#100ms;
14 END_IF;

```

**NOTE**

Changes in this network only become effective after changing from STOP to RUN

**HWIDSTW and HWIDZSW parameter**

The "HWIDSTW" and "HWIDZSW" parameters must reference the hardware ID of the telegram used for the drive. This application uses a free telegram of length 6/6. In the auto-complete area (Ctrl + Space), select the automatically generated system constants of the drive's telegram.

Figure 1-7 Interconnecting the telegram slot

```

1 // set the master/slave settings on initial call of this organization block
2 IF #Initial_Call THEN
3   "MasterDriveInfo".settings.HWIDSTW := G120_1~PROFINET_interface~Free_telegram
4   "MasterDriveInfo".settings.HWIDZSW := G120_1~PROFINET_interface~Free_telegram
5   "MasterDriveInfo".settings.refSpeed := 1500.0;
6   "MasterDriveInfo".settings.refTorque := 1.7;
7   "MasterDriveInfo".settings.initialRoundingOffTime := T#100ms;
8

```

|                                              |              |
|----------------------------------------------|--------------|
| "G120_1~PROFINET_interface"                  | Hw_Interface |
| "G120_1~PROFINET_interface~Free_telegram"    | Hw_SubModule |
| "G120_1~PROFINET_interface~IODevice"         | Hw_Device    |
| "G120_1~PROFINET_interface~ModuleAccessP..." | Hw_SubModule |

**Note**

Whenever you make changes to the interconnection of telegrams, always compile and load the hardware and software configuration of the SIMATIC controller, so that the system constants in use remain consistent with the drive configuration.

## 2 Control modes

The "LLoadSharing\_Control" function block supports the following control modes:

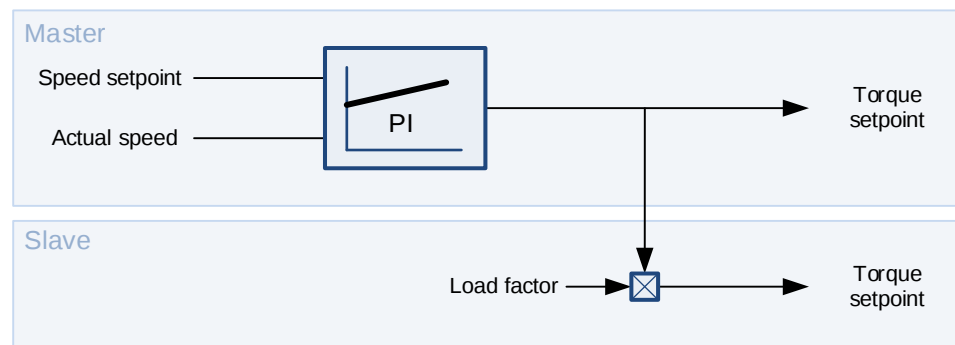
- Torque coupling
- Speed override with torque limit
- Droop and compensation

### 2.1 Torque coupling

In torque coupling, the master drive is speed-controlled and the slave drive is torque-controlled.

The torque setpoint of the master drive is sent to the slave drive as torque setpoint.

Figure 2-1 Torque coupling



The slave drive immediately follows the changes from the master drive torque. If the slave drive's mechanical connection comes loose, the master drive must operate with the full load and the torque setpoint of the master drive increases. The slave drive then follows the torque setpoint without a load and the drive could “spin free”. However, this is appropriately mitigated in the application.

#### CAUTION Speed monitoring

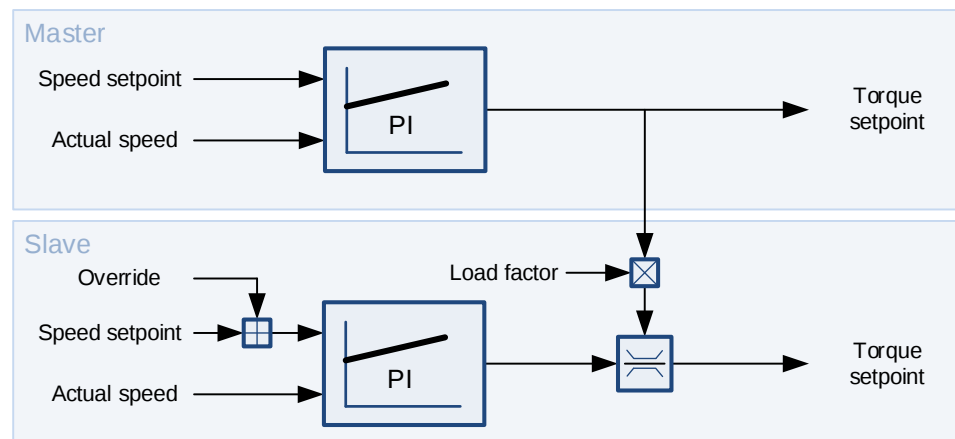
Because it is necessary to use speed monitoring for this control mode, the "LLoadSharing\_Control" function block will shut the drives off if the speeds of the drives are outside of a configurable tolerance (see [6.2.1](#)).

## 2.2 Speed override with torque limit

In speed override with torque limit, both the master drive and the slave drive are speed-controlled.

The speed setpoint of the slave drive is overridden by a configurable value (default 5%). The slave drive thus receives a higher speed setpoint than the master drive. In order to prevent the slave drive from working against the master drive, the torque setpoint of the master drive is interconnected symmetrically to the torque limits of the slave drive.

Figure 2-2 Speed overdrive and torque limit



Due to the persistent control deviation, the slave drive is permanently working at the torque limit of the master drive. If a higher override is selected, a greater difference between setpoint and actual value will result at the speed control. If the slave drive is not yet at the torque limit, the torque limit will be reached faster due to the greater control deviation.

In contrast to torque coupling, the slave drive remains in speed-control in this control mode.

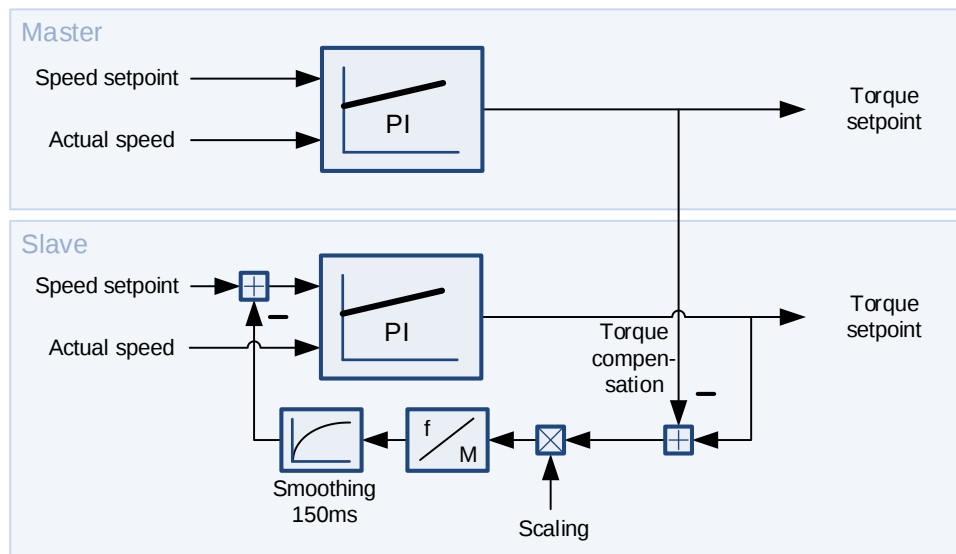
### Note

In the event that the mechanical connection between the drives comes loose, the slave drive will not spin free but will instead attempt to approach its overridden speed.

## 2.3 Droop and compensation

In this control mode the slave torque is compared with the master torque. If differences occur, the slave speed setpoint is adjusted in such a way that the same torques are achieved again. The strength of the droop feedback can be set via the droop scaling factor.

Figure 2-3 Droop and compensation



The scaling and smoothing compensates for spikes and vibrations. The conversion from torque to speed is based on the reference speed and the reference torque of the motor.

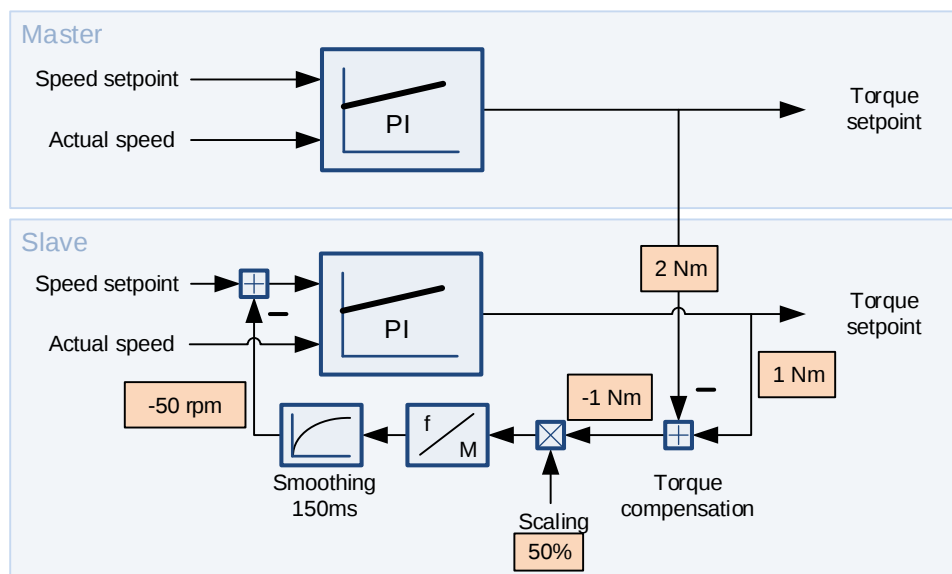
In this control mode, both the master drive and the slave drive are operating speed-controlled.

### Example

Both drives have a speed setpoint of 1500 rpm. The master drive has a torque setpoint of 2 Nm, which is subtracted from the current torque setpoint of the slave drive. At a scaling of 50%, the result is -1 Nm. After conversion to rpm and subsequent smoothing, for example, the result is a value of -50 rpm.

The speed setpoint of the slave drive is thus increased by -50 rpm from 1500 rpm to 1550 rpm. The slave tries to accelerate and provides a higher torque for this. The master drive thereby relieved of some of the load and the load sharing takes effect.

Figure 2-4 Droop and compensation example calculation



## 3 Configuring the drives

The following description shows the parameterization of the drives in SINAMICS Startdrive. If the drives are integrated via GSDML, then the following parameterizations must each be performed in the expert lists in STARTER/SCOUT.

### NOTE

In this application example identical motors and drives are used to allow the change of master and slave role!

### 3.1 Telegram

A free telegram configuration based on SIEMENS telegram 352 will be used for communication with the controller.

#### 3.1.1 Structure

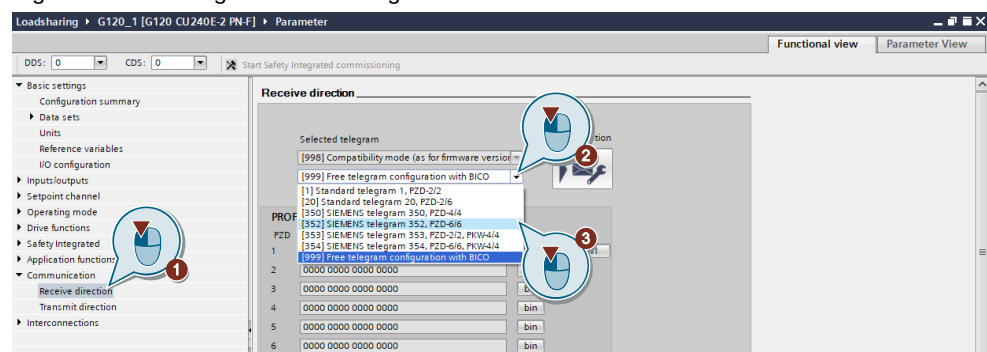
Table 3-1 Telegram structure

| PZD  | Receive direction                                                  | Send direction                             |
|------|--------------------------------------------------------------------|--------------------------------------------|
| PZD1 | STW1                                                               | ZSW1                                       |
| PZD2 | p1070[0] - Speed setpoint                                          | r63[1] - Actual speed value, smoothed      |
| PZD3 | Load sharing control word                                          | r79 - Torque setpoint                      |
| PZD4 | p1523[0] - Lower torque limit                                      | r80[0] - Actual torque value, not smoothed |
| PZD5 | p1522[0] - Upper torque limit                                      | r2132 - Current alarm code                 |
| PZD6 | p1486[0] - Droop compensation torque<br>p1503[0] - Torque setpoint | r2131 - Current fault code                 |

#### 3.1.2 Interconnection

Because the telegram is based on SIEMENS telegram 352, set the receive and send direction to "[352] SIEMENS Telegram 352, PZD-6/6".

Figure 3-1 Selecting SIEMENS telegram 352



After the switchover, the PZD is automatically interconnected to the prescribed parameters. Now reset the telegram to "[999] Free telegram configuration with BICO" so that the PZD can be interconnected in what follows.

### Receive direction

Interconnect the receive direction as follows:

Table 3-2 Parameters

| PZD  | Parameter | Parameter text                                              |
|------|-----------|-------------------------------------------------------------|
| PZD3 |           | No interconnection - reserved for load sharing control word |
| PZD4 | p1523     | Lower torque limit                                          |
| PZD5 | p1522     | Upper torque limit                                          |
| PZD6 | p1486     | Droop compensation torque                                   |
|      | p1503     | Torque setpoint                                             |

### Send direction

Interconnect the send direction as follows:

Table 3-3 Parameters

| PZD  | Parameter | Parameter text                    |
|------|-----------|-----------------------------------|
| PZD3 | r79       | Torque setpoint                   |
| PZD4 | r80[0]    | Torque value, <b>not smoothed</b> |
| PZD5 | r2132     | Current alarm code                |
| PZD6 | r2131     | Current fault code                |

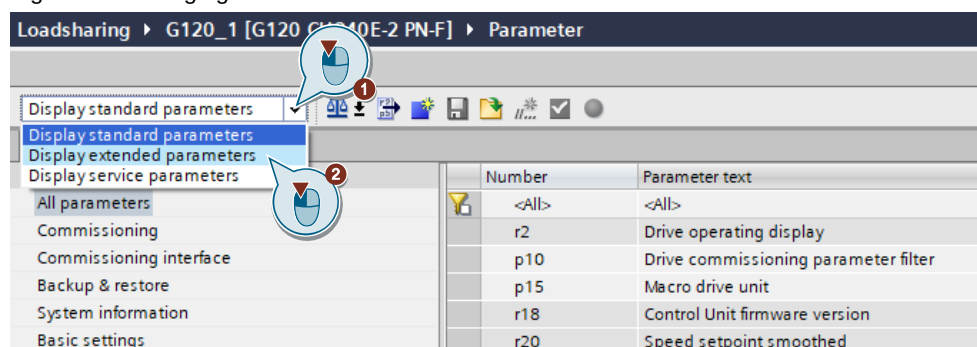
## 3.2 Parameter

#### Note

The configuration will be set up identically for both drives.

Open the parameter editor for the respective drive and, before setting the parameters, set the access level to "Display extended parameters" in the parameter view.

Figure 3-2 Changing access level



### 3 Configuring the drives

---

Set the parameters to shown in [Table 3-4](#).

Table 3-4 Parameters

| Parameter | Parameter text                                   | Value                                  |
|-----------|--------------------------------------------------|----------------------------------------|
| p96       | Application class                                | [0] Expert                             |
| p1300     | Open-loop/closed-loop control operating mode     | Speed control with or without encoder  |
| p1488     | Droop function input source                      | [1] Droop from torque setpoint         |
| p1492     | Droop feedback enable                            | PROFIdrive received PZD3 Bit 1 r2092.1 |
| p1496     | Acceleration pre-control scaling                 | 0.0%                                   |
| p1501     | BI: Switch between speed/torque feedback control | PROFIdrive received PZD3 Bit 0 r2092.0 |

## 4 Description of the library

### 4.1 Contents

The "LLoadSharing" library is composed of program blocks and user-defined PLC data types.

Two function blocks are used for load sharing:

- LLoadSharing\_Control  
Function block to control the load sharing
- LLoadSharing\_DriveCommunication  
This function block is called in LLoadSharing\_Control and is used for communication with the drive.

For drive data storage, the following user-defined PLC data types are used:

- LLoadSharing\_typeDriveSettings  
Contains necessary reference values for the drive
- LLoadSharing\_typeDriveStatus  
Contains drive status values
- LLoadSharing\_typeDriveInfo  
Composed of one instance each of the types LLoadSharing\_typeDriveSettings and LLoadSharing\_typeDriveStatus

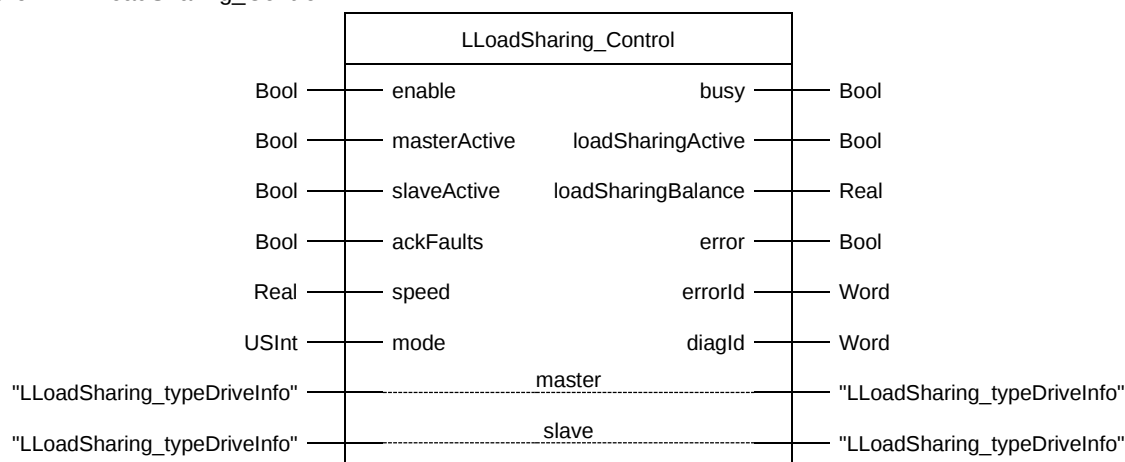
The library also contains the two data blocks of the type LLoadsharing\_typeDriveInfo:

- MasterDriveInfo
- SlaveDriveInfo

The organization block "LLoadSharing\_CyclicInterrupt" serves as a template for use of the library. The function block "LLoadSharing\_Control" is already called there.

## 4.2 LLoadSharing\_Control

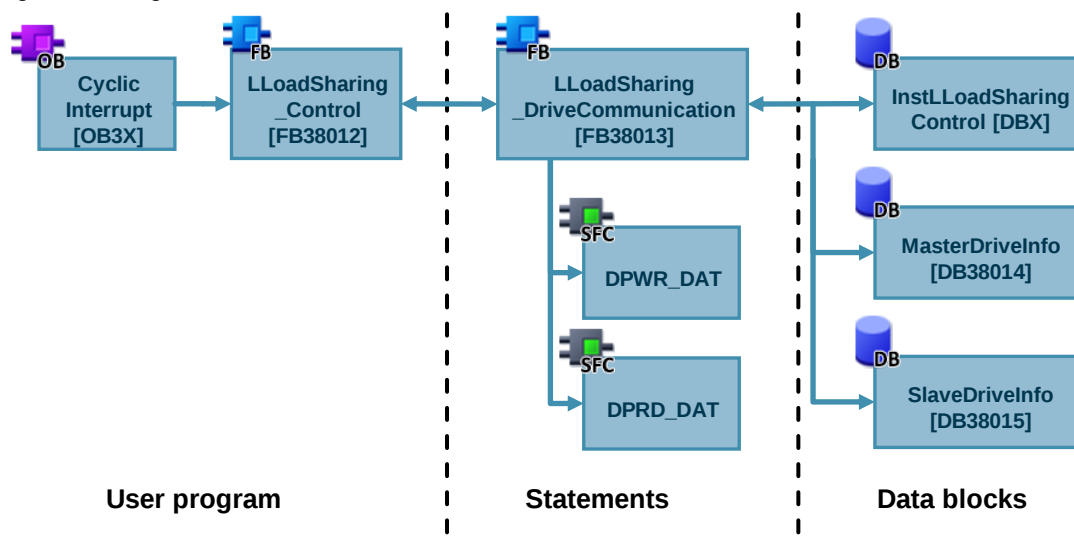
Figure 4-1: LLoadSharing\_Control



### 4.2.1 Description

The function block "LLoadSharing\_Control [FB38012]" is used for controlling the master and slave drive. The function block supports not only traversing of one individual drive but also traversing of both drives with a configurable load sharing.

Figure 4-2 Program overview



For cyclic communication to a SINAMICS drive, the function block uses the following system function blocks:

- **DPWR\_DAT** (see [/3/](#))  
write consistent data of a DP standard slave
- **DPRD\_DAT** (see [/4/](#))  
read consistent data of a DP standard slave

These instructions ensure that consistency is maintained over the entire process data, i.e. that all elements of the process data of a device originate from the same bus cycle or are transferred in one bus cycle.

### 4.2.2 Calling OBs

The function block must be called cyclically, in a cyclic interrupt OB (e.g. OB30). For the cycle time, select a multiple of the inverter cycle time of 4 ms. The cycle time should be selected according to the necessary control mode dynamic.

### 4.2.3 Input parameters

Table 4-1: Input parameters of LLoadSharing\_Control

| Name         | Type                         | Comment                                                                 |
|--------------|------------------------------|-------------------------------------------------------------------------|
| enable       | Bool                         | Enable load sharing or single drive                                     |
| masterActive | Bool                         | Master drive is active                                                  |
| slaveActive  | Bool                         | Slave drive is active                                                   |
| ackFaults    | Bool                         | Acknowledge faults                                                      |
| speed        | Real                         | Speed setpoint [rpm]                                                    |
| mode         | USInt                        | 1 = Torque coupling<br>2 = Speed override<br>3 = Droop and compensation |
| master       | "LLoadSharing_typeDriveInfo" | Master drive info                                                       |
| slave        | "LLoadSharing_typeDriveInfo" | Slave drive info                                                        |

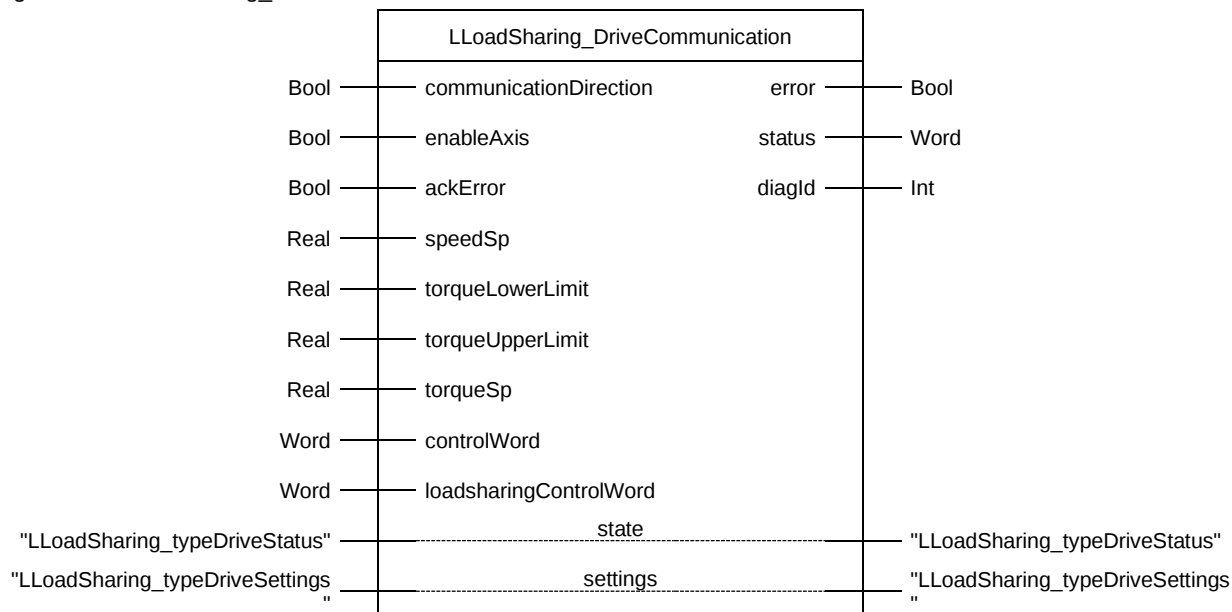
### 4.2.4 Output parameters

Table 4-2: Output parameters of LLoadSharing\_Control

| Name               | Type | Comment                                                                                                        |
|--------------------|------|----------------------------------------------------------------------------------------------------------------|
| busy               | Bool | Load sharing or single drive operation active                                                                  |
| loadSharingActive  | Bool | Load is shared between both drives                                                                             |
| loadSharingBalance | Real | Difference between actual torque of the master and slave drive                                                 |
| error              | Bool | Error                                                                                                          |
| errorId            | Word | Error description (see chapter <a href="#">6.3</a> )                                                           |
| diagId             | Word | Communication error codes of the system function blocks DPWR_DAT / DPRD_DAT (see chapter <a href="#">7.1</a> ) |

## 4.3 LLoadSharing\_DriveCommunication

Figure 4-3 LLoadSharing\_DriveCommunication



The function block "LLoadSharing\_DriveCommunication" is only used for communication to the drives in the function block „LLoadSharing\_Control“ and needs to be inserted in the project by the user (see [Figure 4-2](#)).

## 4.4 PLC data types

### 4.4.1 LLoadSharing\_typeDriveSettings

The PLC data type "LLoadSharing\_typeDriveSettings" contains the hardware IDs which are needed for communication with the drive, as well as reference values with which the received data from the drive are scaled.

Table 4-3: Parameters of LLoadSharing\_typeDriveSettings

| Name       | Type  | Comment                                             |
|------------|-------|-----------------------------------------------------|
| refSpeed   | Real  | Reference speed [p2000] in [rpm]                    |
| refTorque  | Real  | Reference torque [p2003] in [Nm]                    |
| maxSpeed   | Real  | Maximum speed [p1082] in [rpm]                      |
| rampUpTime | Time  | Ramp-function generator ramp-up time [p1120] in [s] |
| HWIDSTW    | HW_IO | Hardware identifier set point slot                  |
| HWIDZSW    | HW_IO | Hardware identifier actual value slot               |

### Assigning reference values parameters

The corresponding reference values must be specified for the standardization of the setpoints and actual values:

Table 4-4 Reference values input parameters

| Input parameters | Data type | Meaning          | Parameters in SINAMICS drives |
|------------------|-----------|------------------|-------------------------------|
| refSpeed         | Real      | Reference speed  | p2000                         |
| refTorque        | Real      | Reference torque | p2003                         |

**Note**

The reference values can be fixed or read out by the SINAMICS drive using acyclic communication. To do this, you can use the function block "SinaPara" from the Startdrive options package. If several SINAMICS drives are communicated with simultaneously, the LAcycCom standard library can also be used for coordinated acyclic work (see [15](#)).

#### 4.4.2 LLoadSharing\_typeDriveStatus

The PLC data type "LLoadSharing\_typeDriveStatus" represents the drive's send telegram. In addition to state word 1 and the actual values for speed and torque, it also contains the current warning word and error word.

Table 4-5: Parameters of LLoadSharing\_typeDriveStatus

| Name        | Type | Comment                        |
|-------------|------|--------------------------------|
| axisEnabled | Bool | Drive is enabled               |
| error       | Bool | Drive has error                |
| lockout     | Bool | Drive lockout active           |
| statusWord  | Word | Status word 1 of the drive     |
| actSpeed    | Real | Smoothed actual speed in [rpm] |
| actTorqueSp | Real | Actual torque setpoint in [Nm] |
| actTorque   | Real | Actual torque in [Nm]          |
| warnCode    | UInt | Warn code of the drive         |
| faultCode   | UInt | Fault code of the drive        |

#### 4.4.3 LLoadSharing\_typeDriveInfo

The PLC data type "LLoadSharing\_typeDriveInfo" contains an instance of the data type "LLoadSharing\_typeDriveStatus" and "LLoadSharing\_typeDriveSettings". An instance of this PLC data type is used for each drive for communication with the drive and for calculating the setpoint values in the "LLoadSharing\_Control" function block.

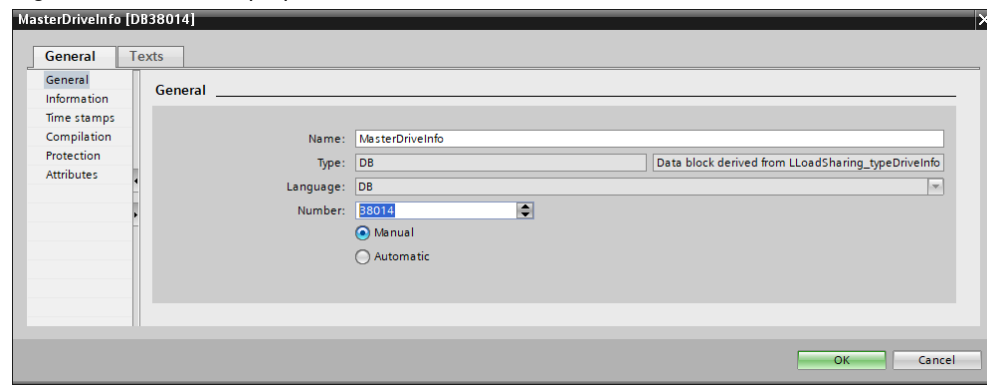
Table 4-6: Parameters of LLoadSharing\_typeDriveInfo

| Name     | Type                             | Comment                |
|----------|----------------------------------|------------------------|
| status   | "LLoadSharing_typeDriveStatus"   | Drive status           |
| settings | "LLoadSharing_typeDriveSettings" | Drive reference values |

## 4.5 MasterDriveInfo and SlaveDriveInfo

The data blocks "MasterDriveInfo [DB38014]" and "SlaveDriveInfo [DB38015]" are derived from the PLC data type "LLoadSharing\_typeDriveInfo". The data blocks are interconnected to the respective drive and passed to the "LLoadSharing\_Control" function block. They can also be used to analyze the status values of the drive.

Figure 4-4 Data block properties



The status of the drive in the corresponding data block can be analyzed in the user program.

Figure 4-5: DriveInfo data block structure

| MasterDriveInfo |             |                                  |             |               |                                  |
|-----------------|-------------|----------------------------------|-------------|---------------|----------------------------------|
|                 | Name        | Data type                        | Start value | Monitor value | Comment                          |
| 1               | Static      |                                  |             |               |                                  |
| 2               | status      | "LLoadSharing_typeDriveStatus"   |             |               |                                  |
| 3               | axisEnabled | Bool                             | false       | TRUE          | Drive is enabled                 |
| 4               | error       | Bool                             | false       | FALSE         | Drive has error                  |
| 5               | lockout     | Bool                             | false       | FALSE         | Drive lockout active             |
| 6               | statusWord  | Word                             | 16#0        | 16#EF37       | Status word 1 of the drive       |
| 7               | actSpeed    | Real                             | 0.0         | 1499.725      | Smoothed actual speed in [rpm]   |
| 8               | actTorqueSp | Real                             | 0.0         | 0.08508302    | Actual torque setpoint in [Nm]   |
| 9               | actTorque   | Real                             | 0.0         | 0.08082886    | Actual torque in [Nm]            |
| 10              | warnCode    | UInt                             | 0           | 0             | Warn code of the drive           |
| 11              | faultCode   | UInt                             | 0           | 0             | Fault code of the drive          |
| 12              | settings    | "LLoadSharing_typeDriveSettings" |             |               |                                  |
| 13              | refSpeed    | Real                             | 0.0         | 1500.0        | Reference speed [p2000] in [rpm] |
| 14              | refTorque   | Real                             | 0.0         | 1.7           | Reference torque [p2003] in [Nm] |

## 4.6 LLoadSharing\_CyclicInterrupt

The organization block "LLoadSharing\_CyclicInterrupt [OB30]" serves as a template for the correct use of the data blocks and function blocks. It is configured with a cycle time of 8000 µs.

### Network 1

When the organization block is called for the first time in network 1, the reference values for the two drives are written once to the respective data block. You can find information on interconnecting "HWIDSTW" and "HWIDZSW" in the chapter [HWIDSTW and HWIDZSW parameter](#).

## 4 Description of the library

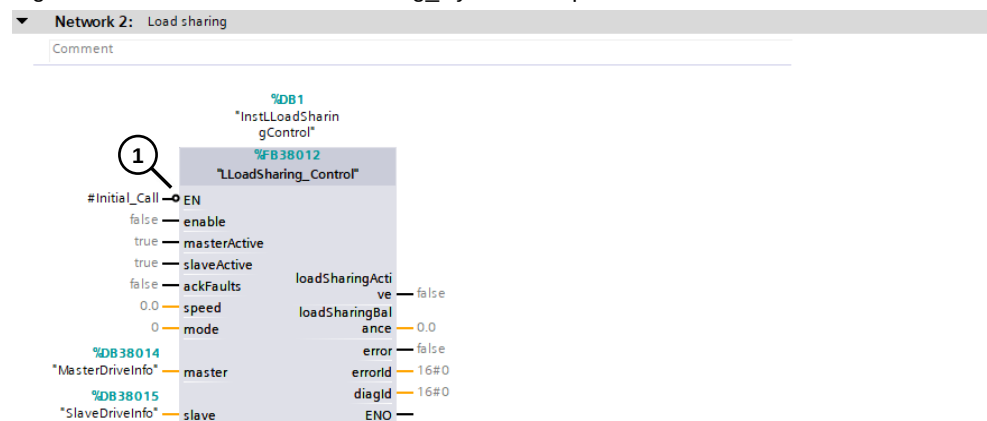
Table 4-7: Network 1 of LLoadSharing\_CyclicInterrupt

|    |                                                                             |
|----|-----------------------------------------------------------------------------|
| 1  | // set the master/slave settings on initial call of this organization block |
| 2  | IF #Initial_Call THEN                                                       |
| 3  | "MasterDriveInfo".settings.HWIDSTW := <HWIDSTW Master>;                     |
| 4  | "MasterDriveInfo".settings.HWIDZSW := <HWIDZSW Master>;                     |
| 5  | "MasterDriveInfo".settings.maxSpeed := 1500.0;                              |
| 6  | "MasterDriveInfo".settings.refSpeed := 1500.0;                              |
| 7  | "MasterDriveInfo".settings.refTorque := 1.7;                                |
| 8  | "MasterDriveInfo".settings.rampUpTime := 5.0;                               |
| 9  |                                                                             |
| 10 | "SlaveDriveInfo".settings.HWIDSTW := <HWIDSTW Slave>;                       |
| 11 | "SlaveDriveInfo".settings.HWIDZSW := <HWIDZSW Slave>;                       |
| 12 | "SlaveDriveInfo".settings.maxSpeed := 1500.0;                               |
| 13 | "SlaveDriveInfo".settings.refSpeed := 1500.0;                               |
| 14 | "SlaveDriveInfo".settings.refTorque := 1.7;                                 |
|    | "SlaveDriveInfo".settings.rampUpTime := 5.0;                                |
|    | END_IF;                                                                     |

### Network 2

The "LLoadSharing\_Control" function block is called in network 2. Please note that the function block is only called after the second call (1) of the organization block. This prevents the function block from being run with invalid reference values.

Figure 4-6: Network 2 of LLoadSharing\_CyclicInterrupt



## 5 Selecting the appropriate control mode

The function block "LLoadSharing\_Control" supports up to three control modes (see chapter 2). Depending on the application, it must be investigated which control mode is the most suitable.

### Note

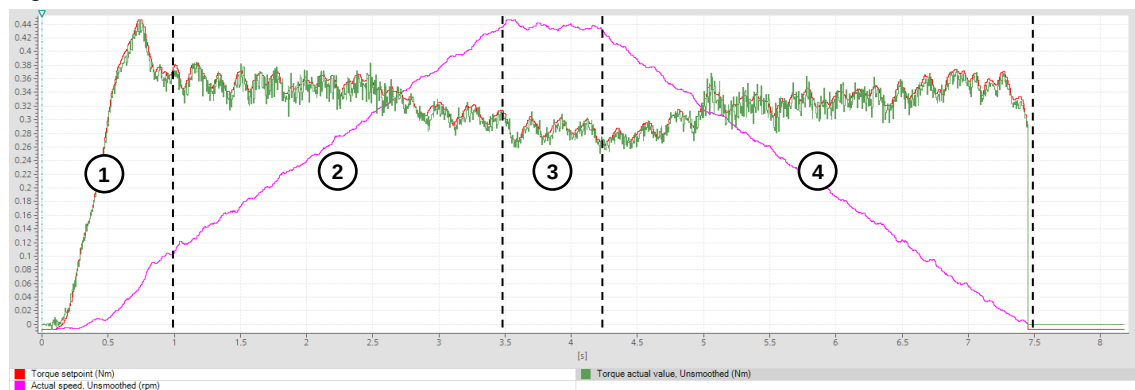
The test of a control mode should always be performed for the entire traverse range and under usual conditions.

### 5.1 Reference run

Before determining the appropriate control mode, it is helpful to analyze and understand the traversing profile of the drives.

In this application example, two drives have been coupled with each with a belt. [Figure 5-1](#) shows a reference run for the slave drive, analyzed below. For traversing of an individual drive, follow the steps in chapter 6.1.

Figure 5-1 Reference run



Because this example traverses in speed control without an encoder, **segment 1** shows the switchover from open-loop operation to closed-loop operation. This can be seen in the torque peak. After the change to closed-loop operation, the torque is steadily reduced at constant acceleration in **segment 2**. In **segment 3** the speed setpoint has been reached and the torque stays constant for a constant traverse. During the deceleration leading up to standstill in **segment 4** the torque rises again due to the resistance from the belt and from the drive shaft of the master drive.

With the reference run having been recorded and the traversing profile analyzed, the various control options can be compared in the next step.

### Note

If the traversing profile differs significantly from the reference run at uniform load sharing, check the correct interconnection/parameter entry for the drives according to chapter 3.

## 5.2 Traversing with load sharing

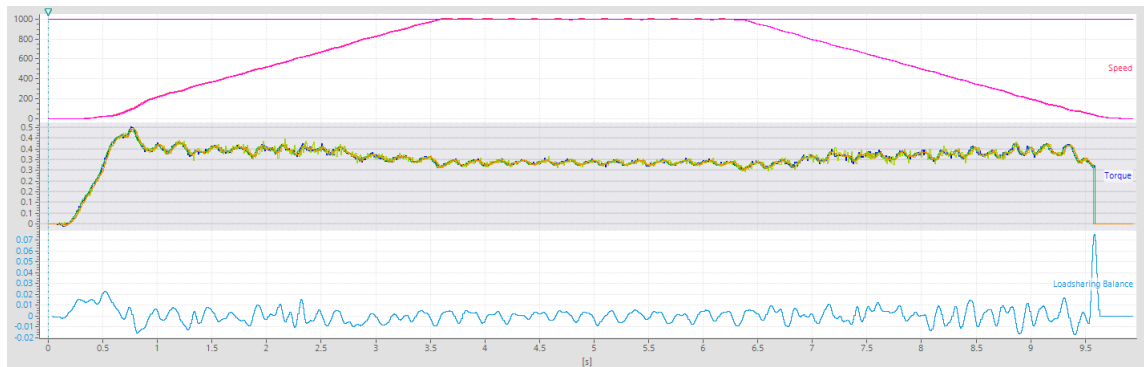
Choose a control mode at the "mode" input parameter (see [4.2.3](#)) and traverse the drives.

While traversing the drives, record a measurement with the actual torque values from both drives and check the difference of both torques (signal "Loadsharing Balance").

## 5.3 Torque coupling

A measurement with the "Torque coupling" mode has been created in [Figure 5-2](#). At a 1:1 load sharing between the two drives, an average difference of 0.005 Nm is measured (see signal "Loadsharing Balance").

Figure 5-2 Reference run with torque coupling



### Note

The trace configurations of the graphs shown can be downloaded under [I2](#) and imported as described in chapter [7.3](#).

## 5.4 Speed overdrive with torque limit

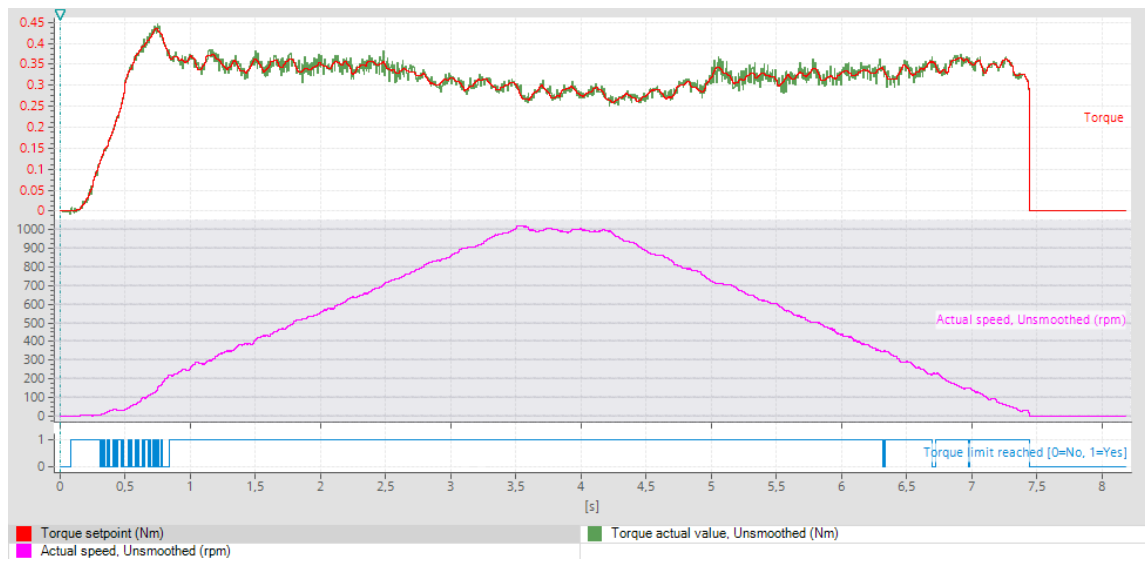
The speed setpoint of the slave drive is overdriven by 5% by default. In this mode it is important that the slave drive is always at the torque limit. If the closed-loop control performance after a reference run is unsatisfactory, then a faster feedback response, and thus the torque limit, can be achieved with an increase in the override factor.

Therefore, you should also record the speed controller state word (r1407) in a reference run and analyze whether bit 7 "Torque limit reached" is set throughout the entire reference run.

### Note

For speed control without an encoder, the torque limit may not be reached at the start of the acceleration ramp and at the end of the deceleration ramp due to the switchover from open-loop control to closed-loop control.

Figure 5-3 Torque limit reached



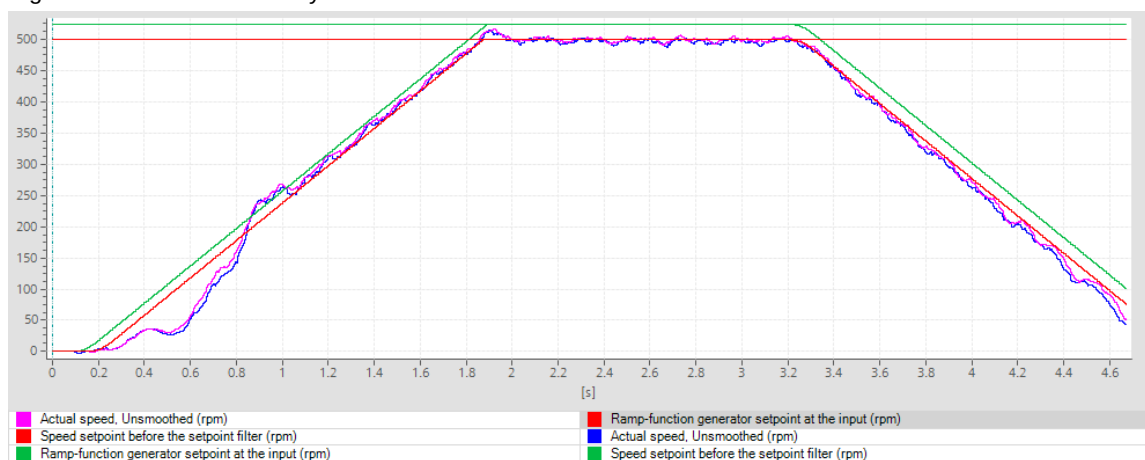
Because an additive speed override is not easily possible in SINAMICS G converters due to a limited parameter set, the master drive is run at a delay to the slave drive during speed override with torque limit.

In this way, the ramp-function generator of the slave drive's speed regulator is always ahead of the master drive's. Due to the restricted torque limit, however, the slave drive cannot start before the master drive. The slave drive is thus at the torque limit.

The master drive is delayed by delay time configured in DriveInfo initialRoundingOffTime at the start of the traverse.

Figure 5-4 shows the traversing profile of the master and slave drive with the measurements superimposed. The green signal represents the speed setpoint as well as the ramp-function generator slope for the slave drive. In contrast to the master drive's red signal it can be seen that the master drive generates the ramp-function generator slope, delayed by the programmed time. Because the master drive has not yet achieved any torque, however, the slave drive is at the torque limit from the very start.

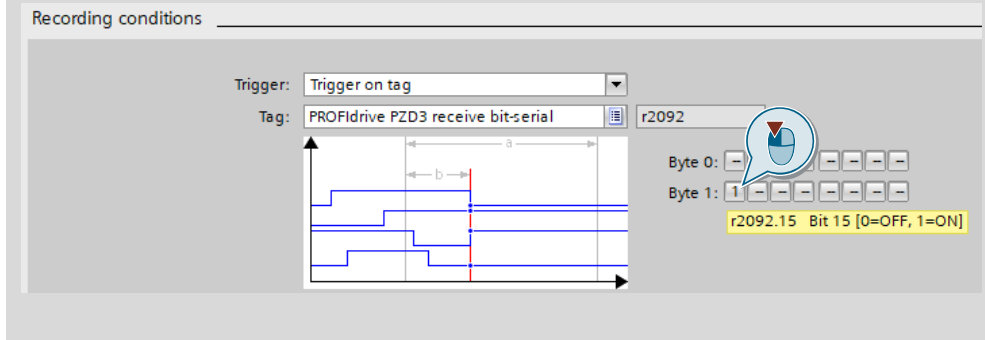
Figure 5-4 Master drive delay



## 5 Selecting the appropriate control mode

### Note

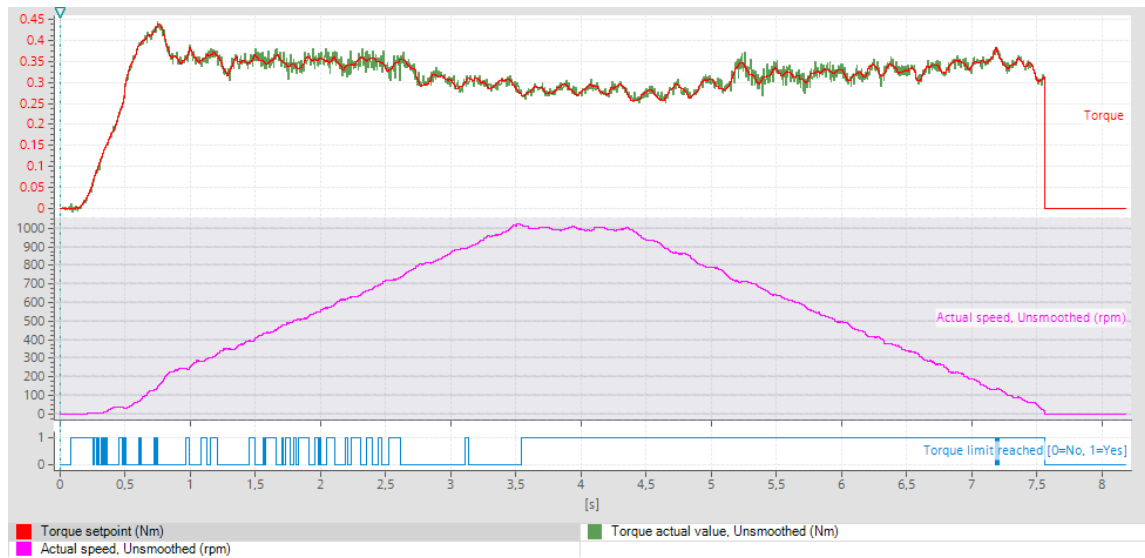
A synchronous start of two drive traces is possible with parameter r2092 bit 15. Parameter r2092 corresponds to the load balancing control word.



### 5.4.1 Opportunities for optimization

If the torque limit is not consistently reached during the reference run, this can negatively impact the load sharing. In [Figure 5-5](#) the bit "Torque limit reached" [r1407.7] was recorded for the analyses.

Figure 5-5 Torque limit not reached



### Increasing the override

If the torque limit was not reached during the reference run, it can be helpful to increase the override.

In order to increase the override, open the instance block of the "LLoadSharing\_Control" function block and set the value for the static tag "statSpeedOverrideFactor" to a higher value.

Figure 5-6 Adjusting speed override factor

| InstLLoadSharingControl |                              |           |             |                          |
|-------------------------|------------------------------|-----------|-------------|--------------------------|
|                         | Name                         | Data type | Start value | Retain                   |
| 1                       | Input                        |           |             | <input type="checkbox"/> |
| 2                       | Output                       |           |             | <input type="checkbox"/> |
| 3                       | InOut                        |           |             | <input type="checkbox"/> |
| 4                       | Static                       |           |             | <input type="checkbox"/> |
| 5                       | statSlippageDetectionEnabled | Bool      | TRUE        | <input type="checkbox"/> |
| 6                       | statSpeedOverrideFactor      | Real      | 5.0         | <input type="checkbox"/> |
| 7                       | statState                    | USInt     | 0           | <input type="checkbox"/> |
| 8                       | statEnableOld                | Real      | 6.1e        | <input type="checkbox"/> |

#### Note

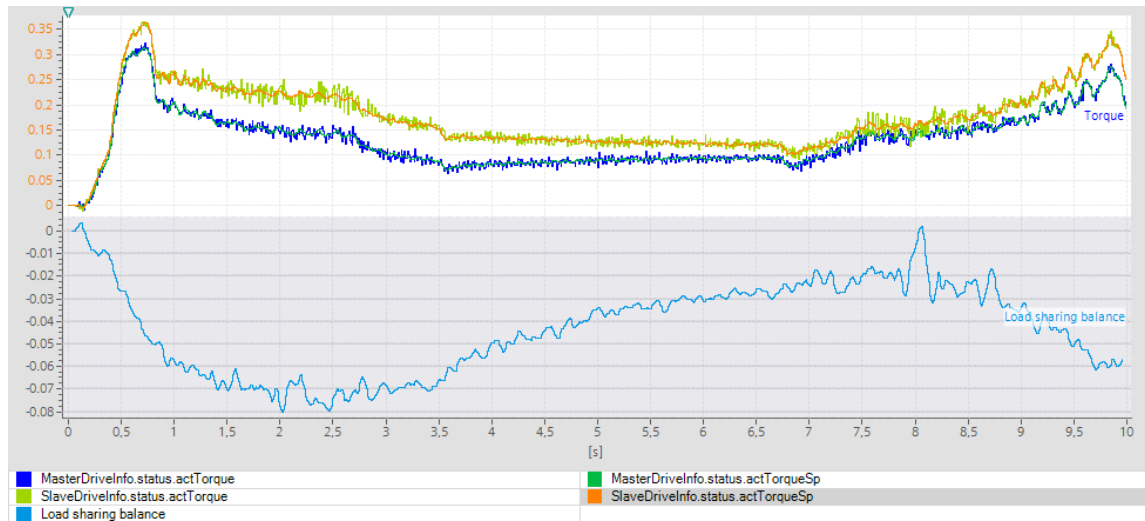
An excessive override factor can cause the slave drive to oscillate!

## 5.5 Droop and compensation

The difference from the compensation torque of the master drive and the torque setpoint of the slave drive is scaled prior to feedback with the factor p1489 "Droop scaling".

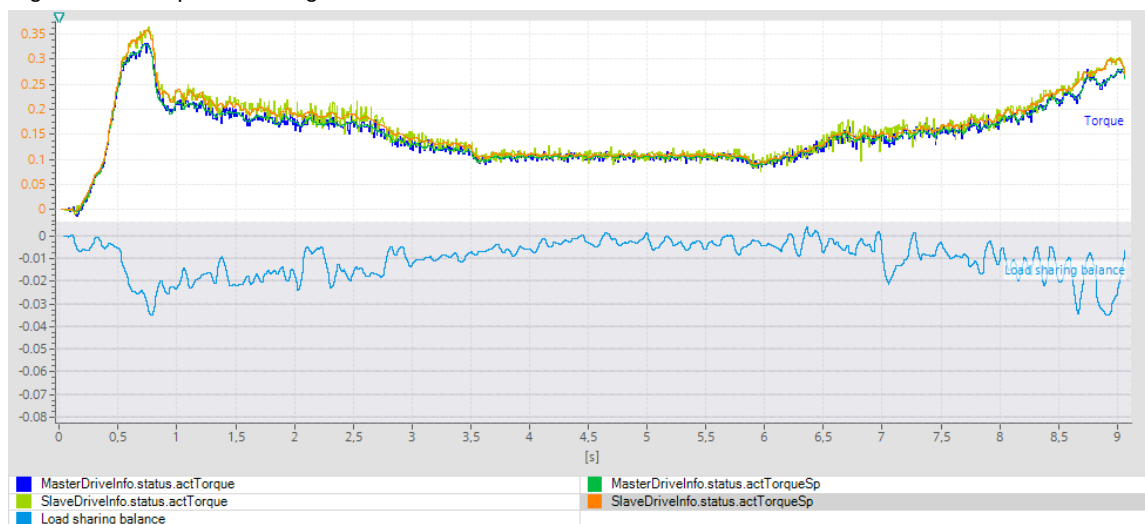
By default the feedback is only 5%.

Figure 5-7 Droop scaling with 5%



Thus, for example, torque spikes or vibrations from the master drive are not conveyed directly to the slave drive. In [Figure 5-7](#) it can be seen that in this manner a 1:1 load sharing is only gradually attained. This behavior can be changed by increasing the scaling.

Figure 5-8 Droop and scaling with 25%



### Droop source integral share, speed regulator

In the slave drive, the torque setpoint of the master drive has been connected as the droop source. If a higher control accuracy is needed, then the integral output from the speed regulator of the master drive can be used as a droop source.

Follow these steps:

- In the **slave** drive, set parameter p1488 to "[3] Droop from integral output, speed regulator".
- In the **master** drive, interconnect PZD3 of the send telegram to parameter r1482 "speed regulator I-torque output".
- Load the changes to the drives.

#### CAUTION

#### Changing control modes is no longer possible!

Changing control modes is no longer possible after modification of the send telegram!

## 6 Operation

### 6.1 Single traversing

The "LLoadSharing\_Control" function block offers the ability to traverse a drive individually. This can be necessary if one drive has a fault.

Follow these steps:

1. If there is a mechanical coupling between the drives, it may have to be released (except for a reference run).
2. Acknowledge pending faults with the "ackFaults" input parameter.
3. At the input parameter "masterActive" or "slaveActive", set the drive to be traversed individually to the value "true" and the other to "false".
4. If necessary, reduce the speed at the input parameter "speed".
5. Start the traversing movement with a positive edge at the input parameter "enable".

The active drive will now run with 100% load.

### 6.2 Load sharing

1. Set "masterActive" and "slaveActive" to "true" to activate load sharing.
2. Choose a control mode at the "mode" input parameter (see [4.2.3](#)).
3. Specify a speed setpoint at the input parameter "speed".
4. Acknowledge pending faults with the "ackFaults" input parameter.
5. Start the traversing movement with load sharing by setting the input parameter "enable" to "true".

The difference between the torques of the master and slave drive is output at the output parameter "loadSharingBalance".

Stop the drives by setting the input parameter "enable" to "false".

#### 6.2.1 Speed difference monitoring

Because both drives should run with the same speed, minimal differences in speed will occur during normal operation. If, for example, the mechanical connection of the slave drive breaks, then the motor may "spin free" as described in chapter [2.1](#). In order to prevent this, the difference between both actual speed values is calculated and monitored in the function block "LLoadSharing\_Control". If the difference exceeds a predefined limit, both drives will coast off (OFF2).

#### Configuration

The speed difference monitoring can be configured in the interface description of the "LLoadSharing\_Control" function block.

- **statActualSpeedDetectionEnabled**  
Activate speed difference monitoring (default: TRUE)
- **SPEED\_DIFF\_DETECTION\_FACTOR**  
Threshold value of the difference in percent. 2.0 means that the speed difference is not permitted to be greater than 2% of the speed setpoint (speed).

Figure 6-1 Setting up speed difference monitoring

| LLoadSharing_Control |                                 |              |               |                                                                       |
|----------------------|---------------------------------|--------------|---------------|-----------------------------------------------------------------------|
|                      | Name                            | Data type    | Default value | Comment                                                               |
| 1                    | Input                           |              |               |                                                                       |
| 2                    | Output                          |              |               |                                                                       |
| 3                    | InOut                           |              |               |                                                                       |
| 4                    | Static                          |              |               |                                                                       |
| 5                    | statActualSpeedDetectionEnabled | Bool         | TRUE          | Will detect a difference in master and slave actual speed             |
| 6                    | statSpeedOverrideFactor         | Real         | 5.0           | Speed override factor [%]                                             |
| 7                    | statState                       | USInt        | 0             |                                                                       |
| 8                    | statEnableOld                   | Bool         | false         |                                                                       |
| 9                    | statAckFaultsOld                | Bool         | false         |                                                                       |
| 22                   | slaveSettings                   | *LLoadSha... |               |                                                                       |
| 23                   | Temp                            |              |               |                                                                       |
| 24                   | Constant                        |              |               |                                                                       |
| 25                   | LOAD_SHARING_FACTOR             | Real         | 1.0           | Load sharing factor, 1.0 = load is shared equally amongst both drives |
| 26                   | SPEED_DIFF_DETECTION_FACTOR     | Real         | 2.0           | Threshold of the speed difference detection [%]                       |
| 27                   | STANDSTILL_DETECTION_THRESHOLD  | Real         | 1.0           | Below this value the drive is in standstill [rpm]                     |

### 6.2.2 Changing load sharing factor

The default setting for the load sharing factor is 1.0. This means that the torque of the master drive is relayed to the slave drive at 100%. The load sharing factor can be set in the constant "LOAD\_SHARING\_FACTOR" of the function block "LLoadSharing\_Control". A factor of 0.5 means that only half of the torque of the master drive is relayed to the slave drive. So the slave takes one third and the master two thirds of the load.

Figure 6-2 Setting load ratio

| LLoadSharing_Control |                                |          |             |                                                                       |
|----------------------|--------------------------------|----------|-------------|-----------------------------------------------------------------------|
|                      | Name                           | Datentyp | Defaultwert | Kommentar                                                             |
| 3                    | InOut                          |          |             |                                                                       |
| 4                    | Static                         |          |             |                                                                       |
| 5                    | Temp                           |          |             |                                                                       |
| 6                    | Constant                       |          |             |                                                                       |
| 7                    | LOAD_SHARING_FACTOR            | Real     | 1.0         | Load sharing factor, 1.0 = load is shared equally amongst both drives |
| 8                    | SPEED_DIFF_DETECTION_FACTOR    | Real     | 2.0         | Threshold of the speed difference detection [%]                       |
| 9                    | STANDSTILL_DETECTION_THRESHOLD | Real     | 1.0         | Below this value the drive is in standstill [rpm]                     |

## 6.3 Error handling

The collective error "error" is set as soon as one of the following conditions is fulfilled:

- SINAMICS drive reports fault
- SINAMICS drive switch-on lock is active
- Invalid values were specified at an input parameter or they were changed while the drive was running.
- A speed difference was detected during load sharing
- Error when reading/writing telegram data (SFC error see "diagID")

Table 8 Error codes

| ErrorId | Meaning                                                                              | Note                                                                                                                |
|---------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| 16#0000 | No error                                                                             |                                                                                                                     |
| 16#7005 | Warning on drive active                                                              | Drive warning number is output in <code>&lt;DriveInfo&gt;.status.warnCode</code> (see <a href="#">Figure 4-5</a> ). |
| 16#7013 | Several reference values are invalid                                                 | Specify a valid reference value at the respective input parameter. (see <a href="#">Table 4-4</a> )                 |
| 16#8001 | Reference speed is invalid, operation not possible                                   |                                                                                                                     |
| 16#8002 | Reference torque is invalid, operation not possible                                  |                                                                                                                     |
| 16#8010 | No drive active                                                                      | Select the drives to be run with input parameter "masterActive" and "slaveActive".                                  |
| 16#8011 | The input parameter "masterActive" or "slaveActive" was changed in active operation. | Changing the active drives only permitted in idle.                                                                  |
| 16#8012 | Speed difference between master and slave drive                                      | Check whether one of the drives is blocked or spinning freely.                                                      |
| 16#8013 | Invalid torque limit<br>Lower torque limit is greater than the upper limit.          | Only for speed override with torque limit.                                                                          |
| 16#8202 | Invalid control mode selected                                                        | Select a valid control mode at the "mode" input parameter (see <a href="#">4.2.3</a> ).                             |
| 16#8401 | Drive fault active                                                                   | Check drive fault, acknowledge if necessary via "ackFaults".                                                        |
| 16#8402 | Drive switch-on lock active                                                          | Check for axis / encoder parked, safety functions active, parameter <code>p10</code> $\neq$ 0.                      |
| 16#8600 | Error when reading the telegram data                                                 | Evaluate error code from <code>DPRD_DAT</code> or <code>DPWR_DAT</code> in "diagId" (see <a href="#">7.1</a> ).     |
| 16#8601 | Error while writing the telegram data                                                |                                                                                                                     |
| 16#8800 | Drive fault                                                                          | Drive error number is output in <code>&lt;DriveInfo&gt;.status.faultCode</code> (see <a href="#">Figure 4-5</a> ).  |
| 16#7FFF | Internal block error - invalid state in state machine                                |                                                                                                                     |

## 7 Appendix

### 7.1 Error codes from DPWR\_DAT and DPRD\_DAT

The following table shows the error codes of the system blocks DPWR\_DAT and DPRD\_DAT that are relevant for the library "LLoadSharing" and can occur at the output parameter "diagId".

For a complete overview of all error codes, refer to the function description for DPWR\_DAT (see [3](#)) and DPRD\_DAT (see [4](#)).

Table 7-1 Error codes

| Error code (W#16#...) | Explanation                                                                                                                                                                                                                                                                                    | Remedy                                                                                             |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| 0000                  | No error occurred.                                                                                                                                                                                                                                                                             |                                                                                                    |
| 8090                  | <ul style="list-style-type: none"> <li>You have not configured a module for the specified HW identifier, or</li> <li>You have ignored the restriction on the length of the consistent data, or</li> <li>You have not specified a HW ID as the address at parameter HWIDSTW/HWIDZSW.</li> </ul> | Connect a valid HW identifier (see <a href="#">HWIDSTW</a> and <a href="#">HWIDZSW</a> parameter). |
| 8093                  | For the HW ID specified under HWIDSTW/HWIDZSW, there is no DP module / PROFINET IO device from which you can read consistent data.<br>If the module addressed via HWIDSTW/HWIDZSW does not have inputs, this error code also occurs.                                                           |                                                                                                    |
| 80A0<br>80A1          | An access error was detected when accessing the periphery.                                                                                                                                                                                                                                     | Check the hardware configuration of the TIA project.                                               |
| 80B1                  | The length of the specified target area at parameter RECORD is shorter than the configured user data length.                                                                                                                                                                                   | Check telegram configuration.                                                                      |

### 7.2 Load sharing control word

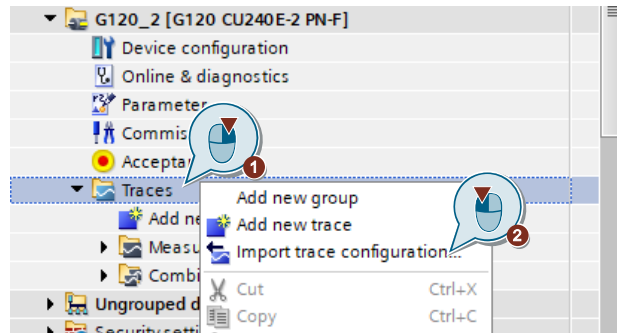
Table 7-2 Structure of load sharing control word

| Bit   | Meaning                                                                                                                   |
|-------|---------------------------------------------------------------------------------------------------------------------------|
| 0     | 0 = Speed control active                                                                                                  |
|       | 1 = Torque control active                                                                                                 |
| 1     | 0 = Droop disabled                                                                                                        |
|       | 1 = Droop enabled                                                                                                         |
| 2..14 | Reserved                                                                                                                  |
| 15    | Tracetrigger<br>Set as soon as load sharing or single run is started. Can be used as a trigger condition for drive trace. |

### 7.3 Importing trace configuration

1. Download the trace configurations (see [V2](#)).
2. Right click on the "Traces" folder.
3. Select the context menu item "Import trace configuration".
4. Select a trace configuration in the dialog window.

Figure 7-1 Importing trace configuration



## 7.4 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:

<https://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:

[www.siemens.com/industry/supportrequest](http://www.siemens.com/industry/supportrequest)

### 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:

[www.siemens.com/sitrain](http://www.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:

<https://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:

<https://support.industry.siemens.com/cs/ww/en/sc/2067>

## 7.5 Links and literature

Table 7-3 Links and literature

| No. | Topic                                                                                                                                                                                                   |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \1\ | Siemens Industry Online Support<br><a href="https://support.industry.siemens.com">https://support.industry.siemens.com</a>                                                                              |
| \2\ | Link to the article page of the Application Example<br><a href="https://support.industry.siemens.com/cs/ww/en/view/109774232">https://support.industry.siemens.com/cs/ww/en/view/109774232</a>          |
| \3\ | DPWR_DAT function description<br><a href="https://support.industry.siemens.com/cs/ww/en/view/109755202/95340509067">https://support.industry.siemens.com/cs/ww/en/view/109755202/95340509067</a>        |
| \4\ | DPRD_DAT function description<br><a href="https://support.industry.siemens.com/cs/ww/en/view/109755202/95340531851">https://support.industry.siemens.com/cs/ww/en/view/109755202/95340531851</a>        |
| \5\ | Standard library LAcycCom in Siemens Industry Online Support<br><a href="https://support.industry.siemens.com/cs/ww/en/view/109479553">https://support.industry.siemens.com/cs/ww/en/view/109479553</a> |

## 7.6 History

Table 7-4 Revision history

| Version | Date    | Modifications                                                         |
|---------|---------|-----------------------------------------------------------------------|
| V1.0    | 05/2020 | First edition                                                         |
| V1.1    | 06/2020 | Fixed error in slippage detection                                     |
| V1.1.1  | 02/2022 | Correction of initial setup parameters (Section <a href="#">1.2</a> ) |
| V1.2    | 12/2022 | Correction of slippage detection calculation                          |