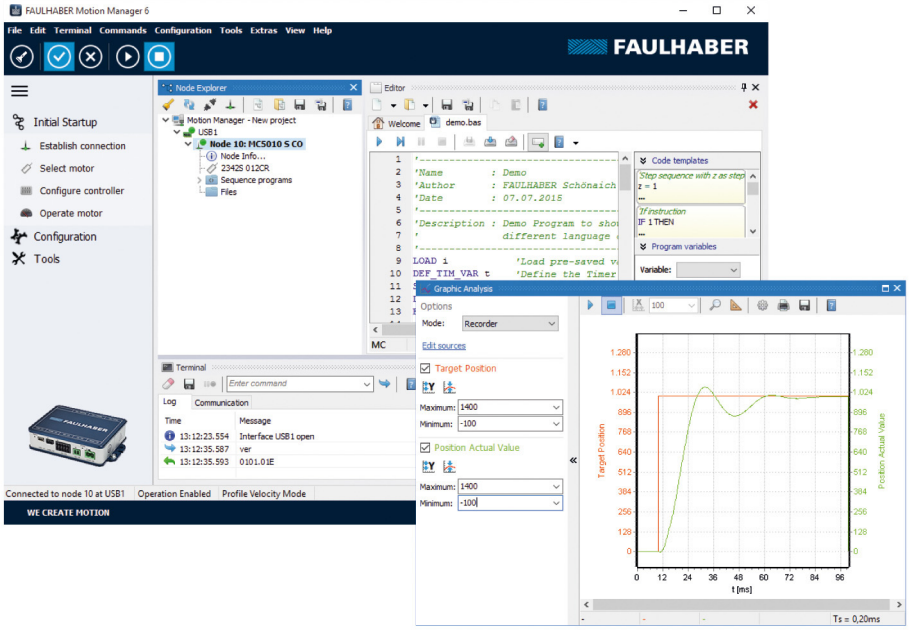
FAULHABER

Software Manual

Motion Manager 6



Imprint

Version:
6th edition, 8.10.2022

Software status:
V6.9

This document has been prepared with care.
Dr. Fritz Faulhaber GmbH & Co. KG cannot accept any liability for any errors in this document or for the consequences of such errors. Equally, no liability can be accepted for direct or consequential damages resulting from improper use of the equipment.

The relevant regulations regarding safety engineering and interference suppression as well as the requirements specified in this document are to be noted and followed when using the software.

Subject to change without notice.

The respective current version of this technical manual is available on FAULHABER's internet site:
www.faulhaber.com

Content

Content

Content

1 About this document

1.1 Validity of this document

This document describes the installation and use of the FAULHABER Motion Manager. The Motion Manager support the following controls:

MC V3.0 family

- MC5010 S RS/CO/ET
- MC5005 S RS/CO/ET
- MC5004 P RS/CO/ET
- MCS series
- MC 3001 B RS/CO
- MC 3001 P RS/CO
- MC 3603 RS/CO/ET

MC V2.0/V2.5 families

RS interface (RS232):

- MCDC 300x RS
- MCBL 300x RS
- MCLM 300x RS
- CS/CSD series

CF interface (CANopen with FAULHABER CAN):

- MCDC 300x CF
- MCBL 300x CF
- MCLM 300x CF
- CC/CCD series

CO interface (CANopen):

- MCDC 300x CO
- MCBL 300x CO
- MCLM 300x CO
- CO/COD series

SC family

- SCxxxx
- BX4 SC series
- BRC series
- BL Flat SC series
- BXT SC series

This document is intended for use by trained experts authorised to operate the supported products.

All data in this document relate to the standard versions of the series listed above. Changes relating to customer-specific versions can be found in the corresponding data sheet.

1.2 Further documents

During certain operations during commissioning and operation of FAULHABER products, additional information from further manuals is useful. These manuals can be downloaded in pdf format from the web page www.faulhaber.com/manuals.

1.3 Using this document

This document is an electronic document in the form of online help. It is supplied with the Motion Manager 6 and is called up via the respective buttons at suitable places or by pressing the F1 key.

- Read the document carefully before undertaking configuration of the communication, in particular the chapter "Safety" (see chap. 2, p. 9).
- Retain the document throughout the entire working life of the product.
- Keep the document accessible to the operating and, if necessary, maintenance personnel at all times.
- Pass the document on to any subsequent owner or user of the product.

1.4 List of abbreviations

Abbreviation	Meaning
ASCII	American Standard Code for Information Interchange
CAN	Controller Area Network
CiA	CAN in Automation e.V.
CSV	Comma-Separated Values
DLL	Dynamic Link Library
LSS	Layer Setting Service
NMT	CANopen network management
OD	Object dictionary
PDO	Process Data Object
RxPDO	Receive Process Data Object (PDO received from the drive)
SDO	Service Data Object
PLC	Programmable Logic Controller
TxPDO	Transmit Process Data Object (PDO sent from the drive)
USB	Universal Serial Bus
VB	Visual Basic

1.5 Symbols and designations

NOTICE!
Risk of damage.

- Measures for avoidance

Instructions for understanding or optimizing the operational procedures

✓ Pre-requirement for a requested action

1. First step for a requested action
   - Result of a step
2. Second step of a requested action

Result of an action

- Request for a single-step action

2 Safety

2.1 Intended use

The FAULHABER Motion Manager supports the configuration and commissioning of FAULHABER drive controls. The software is not designed for controlling drive systems in productive operation.

Depending on the model, drive controls can be connected to the PC via different interfaces. Depending on the features of the control, a programming adapter may also be necessary. This is available from FAULHABER on request.

The instructions listed in the operating instructions for parametrising and commissioning the respective drive controls must be observed.

The FAULHABER Motion Manager is designed for the following tasks:

- Configuration and commissioning of FAULHABER drive systems with Motion and Speed Controllers
- Communication to device controls via the supported interfaces
- Support during setup of the drive functions and controllers
- Operation of controls in the supported operating modes, by means of graphical interfaces or by command inputs
- Creation and execution of actuation sequences, using VB Script programmes or the integrated macro function
- Graphical analysis of the drive behaviour (trace function)
- Creation, transfer and management of sequence programmes to be executed in the device control (incl. debug options)
- Upload and download of parameter files

The functions listed above are not available on all models of controls, and are dependent on the features of the control to be configured.

The following interfaces are supported:

- RS232 (COMx)
- CAN (supported interfaces: IXXAT, Peak, ESD, EMS, others on request)
- USB

A programming adapter for RS232 or USB must be used for Speed Controllers of the SC family.

Motion Controllers with RS232 interface can alternatively be connected via a direct connection to COMx, and also via USB by means of a USB-to-serial adapter.

2.2 Safety instructions

NOTICE!
Incorrect settings of the drive can damage the Motion Controller.

- Comply with the instructions in this Software Manual.

3 General product description

Motion Manager can be used to easily access settings and parameters of the connected control.

Wizards assist during the commissioning of a control. Drive units detected on the selected interfaces are displayed in a tree view. The current interface and display settings can be saved in project files. User actions and communication flow are logged.

With the Motion Manager, sequence programmes for saving and execution can be created, edited, transferred and executed on the devices. Possibilities for error detection and monitoring the programme flow are also available.

The operation of a control and the execution of motion tasks are performed via:

- Graphical operating elements
- Command inputs
- Macro functions
- Programming of sequences using Visual Basic Script (VB Script)

A graphical analysis function allows recording of control parameters. Additional tools are available for the creation and optimisation of controller parameters.

4 Installation

4.1 System requirements

- Operating system: Microsoft Windows 7 or higher version
- Required hard drive space: 150 MB

4.2 Install Motion Manager 6

1. Run the setup file.
2. Select the desired language version.
3. Confirm the installation process by clicking on ***Next***.
4. Read the complete End User Licence contract.
5. Click on ***Next*** to confirm acceptance of the licence agreement.
6. If necessary, modify the installation path for the software.
7. Click on ***Next*** to confirm the installation path.
8. Modify the programme links and create additional links.
9. Click on ***Next*** to confirm the selection.
10. Check the displayed summary of the settings.
11. Click on ***Installation*** to start the installation.
    - ↳ The installation process is being performed.
12. Select whether on completion of the installation the software should be started.
13. Click on ***Finish*** to finish the installation.

↳ The Motion Manager is now installed.

4.3 Update Motion Manager 6

4.3.1 Updating version 5.xx and earlier versions

Installing Motion Manager 6 has no effect on a pre-existing installed version 5.xx (or earlier) of Motion Manager. The two versions can coexist on the same PC system.

The installation process equals that for Motion Manager 6 (see chap. 4.2, p. 12).

4.3.2 Updating version 6.xx

If an earlier version of Motion Manager 6 is already on the system, installing a later version will update it to the newer version. With the **Options** item in the **Extras** menu you can activate an automatic or manual online update function, which requires an Internet connection to keep the software constantly updated to the latest version.

4.4 Uninstall Motion Manager 6

1. In the Windows system control, select "Programmes and Features" or "Software".
2. From the list presented, select "FAULHABER Motion Manager 6".
3. Click on **Uninstall** to select it and **Delete** to confirm it.
   - ↳ A question is displayed asking whether you really wish to uninstall Motion Manager 6.
4. Click on ***YES*** to confirm the query.

↳ Motion Manager is now uninstalled.

4.5 Call parameters

The Moman6.exe programme file contains different call parameters which can be used when starting the software by a command line or customised link. The call parameters can be specified individually or in combination with each other when calling up the Moman6.exe.

The programme file must be called up from the Motion Manager installation directory (working directory).

Parameter	Function
/ENG	Starts Motion Manager with an English user interface if German is set as the default.
/DEU	Starts Motion Manager with a German user interface if English is set as the default.
/CHS	Starts Motion Manager with a Chinese user interface if the Chinese language pack is installed.
/RUN:script.vbs	Runs the specified VB Script programme immediately after starting Motion Manager.
/H	Runs Motion Manager in the background together with /RUN:.
E	Closes Motion Manager in combination with /RUN: after execution of the script.

Example: Switch over from the English version to the German interface

Moman6.exe /DEU

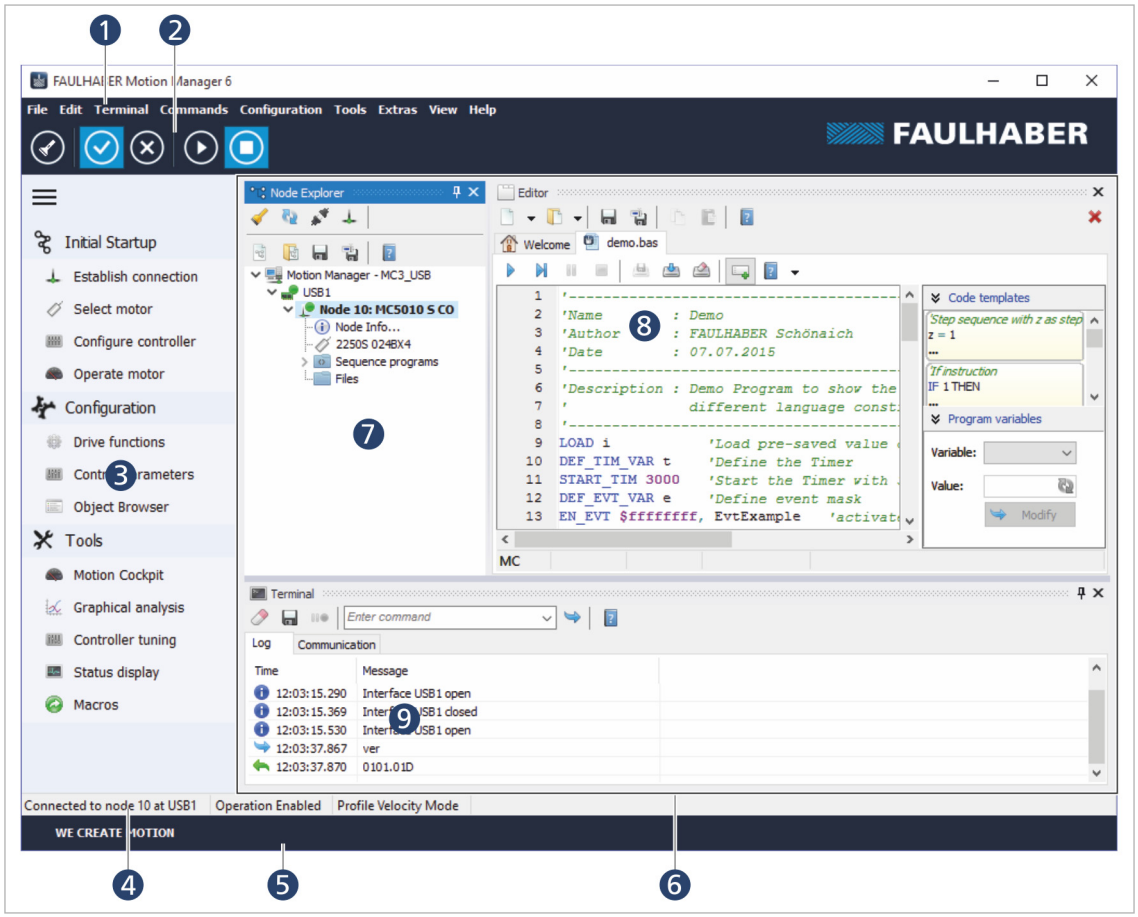
When creating a Windows link with call parameters, the call-up line in the **Target** properties field must be introduced in the following form (example):

"…\Faulhaber\Motion Manager 6\Moman6.exe" /ENG

Observe the quotation marks which include the programme name and its path.

5 User interface

The user interface consists of several windows for the various tasks. These are either free-standing or embedded within the main window.



The standard user interface consists of:

- Header with menu (1) and toolbar (2)
- Quick access (fold away, fold out) (3)
- Status bar (4)
- Footer (5)
- Docking area (6)

In the classic view (set via menu **View - Layout - Classic view**) the header is reduced and shows only the menu with the classic toolbar. The footer is hidden. This means there is more screen surface available for the working area.

Independent of which controller variant is connected, the following windows are always available:

- Node Explorer (7)
- Editor (8)
- Terminal (9)

Depending on the controller variant selected via the Node Explorer, other windows are now available, such as Graphical Analysis, Object Browser, Motion Cockpit.

The individual windows can be dragged using the mouse and docked in the Docking Area of the main window (Drag & Dock).

All the windows that can be docked are listed under the **View** menu item and can be called up from there. Windows can be arranged horizontally or vertically, or arranged as tabs. To do this, drag the window into the desired area that is displayed whilst dragging the window.

Windows that are docked into the Docking Area at the edge of the screen can be hidden using the pin symbol. A tab for the hidden window is shown at the respective edge of the Docking Area. Clicking on the tab opens the window, which is then automatically hidden again once the mouse is used to select an element outside that window.

When the Motion Manager is closed it saves the current layout, which is then available again at restarting. The **View – Layout** menu also allows the current screen layout to be saved in a file which can then subsequently be called up.

5.1 Menu bar

The menu bar contains all the functions and commands necessary for operating the Motion Manager. Its content is dependent on the currently selected control, allowing various functions only to be available if they are also supported by the control.

The following functions are permanently present in the menu bar:

Function	Description
File	Default functions for management of programme files and project files
Edit	Default functions for editing files
Terminal	Functions for management of interfaces and connected nodes
Extras	Additional functions
View	Setting the window layout and the displayed windows
Help	Access to online Help and other support facilities

5.2 Toolbar

The toolbar is displayed below the menu bar. The toolbar contains buttons for quick access to frequently used functions.

Buttons

Button	Meaning	Description
	Node search	Searches the specified interfaces for connected drive nodes or network nodes
	Enable	Switches on the output stage of the control
	Disable	Switches off the output stage of the control
	Run	Loads and starts a sequence programme on the control
	Stop	Stops a running sequence programme on the control

5.3 Quick access

The quick access toolbar allows direct access to the most important functions. The quick access toolbar can be folded away or folded out. It is dynamically configured to match the connected control, and is divided into three areas:

Area	Contents
Commissioning	Wizards and dialogues for commissioning a drive control
Configuration	Dialogues for configuring and parametrising a drive unit to the respective drive task
Tools	Further tools for operating and analysing drives units

5.4 Node Explorer

The Node Explorer displays all controls found to which the Motion Manager was able to establish a successful connection. In addition, the project structure is displayed in the Node Explorer.

Buttons

Button	Meaning	Description
	Node search	Searches the specified interfaces for connected drive nodes or network nodes.
	Update active node	The active node is updated in the Node Explorer.
	Establish connection	Changing the connection settings (see chap. 6.1, p. 28).
	Breaking/establishing the connection	Existing connection is broken or re-established.
	New project	A new project is created.
	Open project	An existing project is opened.
	Save project	The current project is saved.
	Save project as	The current project is saved as a new file.

The display on the Node Explorer is divided into three hierarchical levels:

- Port level (COM, CAN, USB, NET)
- Node level (drive node or network node found)
- Project level (node information and associated files)

Right clicking the mouse opens up a context menu with additional functions.

The interfaces (ports) and nodes can have various states:

Fig. 1: Port and node statuses

1 Active and online
2 Not active
3 Active, offline
4 Fault state
5 Not found during scan

Double clicking on a network node selects and activates it. The user interface adjusts itself to match the properties of the node. By default, commands are sent with the node number of this node.

Double clicking on the **Node info** entry opens a window with further information about the selected node (e. g. software version or serial number).

If the control is already set to a FAULHABER motor, the respective motor name will be displayed. Otherwise the **Select motor** entry appears. Double clicking on this entry opens the wizard for motor selection, for setting and selecting a FAULHABER motor.

In the **Files** folder, links to any files that are connected to this device are managed. For instance these can be parameter files, data sheets or documentation. Files that are already open in the Motion Manager editor can be dragged into the file folder by drag & drop. The context menu can be used to add any files.

5.4.1 Project management

The Node Explorer manages the current programme settings in the project files.

When a project file is saved, the following information is recorded:

- Interface settings of all activated interfaces (protocol, port, channel, baud rate, scan range)
- Information on the node that is displayed (node number, serial number, name)
- Links to the added files for each node
- Programme settings for each node (e. g. trace settings)

When a project file is loaded, an attempt is made to recreate the saved information. If the node saved in the project file can no longer be found, it will be shown in the Node Explorer as a question mark. The nodes remain saved in the project file until they are deleted from the Node Explorer by means of the context menu or using the ***Delete*** key.

When a new project is created the communications interface must first be specified, with an associated search of the connected network nodes. Additional interfaces can be added if required (**Wizard for establishing a connection** or **Terminal - Manage connection...** menu). The node shown in Node Explorer will be written into the project file when it is saved.

The Motion Manager always starts with the most recently loaded project file.

5.4.2 Managing connections

The **Manage connections** dialogue permits management of communication connections that have previously been set up. This includes adding, removing and configuring connections.

The function can be called up either from the **Terminal** menu or from the Node Explorer toolbar.

A particularly easy way of setting up a connection is via the “Establish connection” wizard.

The window is divided into a navigation area in the form of a tree structure and an input area. When the window is opened, the current connection configuration is displayed initially. By selecting and deselecting boxes in the input area, the respective node can be added to the tree or removed from it.

The input area always shows the input options that suit the active node in the navigation area. By clicking on ***OK*** the settings for the current project are applied and the selected interfaces are then automatically searched for connected devices.

Supported interfaces

Display of all interface types supported by the Motion Manager:

- COM (serial RS232 interface)
- CAN
- USB
- NET

Available interface plug-ins

Display of all interface plug-ins available for the selected interface. An interface plug-in establishes the connection between the Motion Manager and interface driver.

The following drive connections are installed by default:

Interface	Plug-in name	Plug-in file	Function
COM	Standard COM	Mocom.dll	Connection to standard serial COM port
CAN	Ixxat VCI3	Ixxat_vci3.dll	Connection to HMS-IXXAT VCI3- / VCI4 driver
	PEAK PCAN	Peak_pcan.dll	Connection to PEAK PCAN driver
	EMS CPC	Ems_cpc.dll	Connection to EMS CPC driver
	ESD NTCAN	Esd_ntcan.dll	Connection to ESD NTCAN driver
USB	FAULHABER MC V3.x USB	MC3Usb.dll	Connection to FAULHABER MC V3.x USB driver
	FAULHABER SC USB	SCUsb.dll	Connection to FAULHABER SC-USB driver
NET	Moman Net Client	MoClient.dll	Connection to a remote Motion Manager server
	EtherCAT master	ET_Master.dll	Connection to an EtherCAT slave via PCAP network driver

COM: The Motion Manager supports the serial interfaces COM1 to COM256. The interfaces available in the system are detected automatically. A USB-to-serial adapter is required for access to the serial interface of the Motion Controller. In addition, the associated driver must be installed. The driver installed must always be up to date with the manufacturer's data.

CAN: For operating the Motion Manager with CAN interface, the driver of the CAN card used must be installed (see the CAN interface manufacturer's manual).

USB: In order to access controls via USB, an associated driver must be installed. The drivers for the FAULHABER controls are installed automatically with the Motion Manager.

NET: To set up a client-server network connection, see chap. 5.7.1, p. 25.

For the connection to an EtherCAT client, a PCAP network driver must be installed (e.g., *WinPcap* from the Motion Manager installation directory under \Drivers\PCAP)

Ports found

Display of the physical ports of the system, that were found via the selected interface plug-in. If a port that is present is not displayed, it was not detected by the system (see chap. 10.2, p. 78).

The assignment of the COM port number when a USB-to-serial adapter is used, can be viewed in the Windows Device Manager and changed if required.

Configuration of the selected port
Setting the connection parameters for the selected port.

The DLL file with the communication protocol of the respective device is selected as the protocol plug-in .

Interface	Protocol plug-in	Meaning
COM	MC2RS	Serial protocol for Motion Controllers of the family MC V2.x
	SCRS	Protocol for the Speed Controller via RS232
	CO_RS232	CO protocol via RS232 for Motion Controllers of the family MC V3.x
CAN	CO_CAN	Standard CANopen-Protocol via CAN
USB	CO_USB	CO protocol via USB for Motion Controllers of the family MC V3.x
	SC_USB	Protocol for the Speed Controller via USB
NET	CO_NET	CO protocol via network for Motion Controllers of the family MC V3.x and MC V2.5 CO

It is permissible to use multiple protocols over a single interface if protocol conflicts can be excluded. When scanning the network, all listed protocol plug-ins are loaded successively in order to search for supported nodes.

Transfer rate
Setting the speed for data transfer (baud rate). A check must first be made whether all devices to be addressed at this interface also support the desired baud rate.

Scan range
Setting the range of node numbers within which connected devices will be sought.

5.4.3 Configuration of a network

Some drives can be addressed by the Motion Manager both individually and also within a network. During configuration of a network, reference must be made to the communication manual for the controls. In general the following points must be satisfied:

- All nodes must have the same transfer rate
- Every node must have a unique node number

▸ If in doubt, first configure all controls individually and then connect them to each other.

5.4.4 Search for nodes

During node search, all selected interfaces which satisfy the parameter settings are searched. After the scan , even nodes that were not found but are listed in the project file are retained. If desired, nodes that are no longer required can be deleted from the project with the ***Del*** button or via the context menu.

5.5 Terminal

The terminal is used for manually inputting commands. It records the data exchange, user actions and status messages.

Operating element	Description
	Delete content of the active tab
	Save the contents of the active tab in a file
	Pause or continue recording in the active tab
	Field for manually command inputs
	Send the entered command

The individual messages are displayed with a time stamp and symbol:

Symbol	Meaning
	Node search started
	Status information
	Error information
	Command sent
	Receive data asynchronously
	Receive data synchronously
	Asynchronous error message received
	Error during synchronous data exchange
	Output from script programme

5.5.1 Command input

The command input field allows Motion Manager commands to be input. These are sent after conversion into the corresponding protocol to the control (see command reference in chap. 10.1, p. 72).

For devices of the MC V2.x family with interface RS or CF, the specified commands are sent either directly as ASCII characters or as a CAN telegram via the FAULHABER channel. Thus all the commands supported by the control can be specified here.

Devices of the SC family do not support any command input.

Commands sent via the command input are logged in the activity log.

5.5.2 Activity log

The activity log records user actions and status information in the **Log** tab.

Active operating elements:

Operating element	Function
	Delete activity log.
	Save activity log in a file (CSV format).

5.5.3 Communication history

The communication history logs the entire data exchange via the active interface and status information in the **Communication** tab.

- The **Data** column displays the data and commands that were sent and received as they were input or interpreted by the Motion Manager.
- The **Telegram** column shows the data bytes of the individual telegrams that were sent and received as hexadecimal values.

Active operating elements:

Operating element	Function
	Delete communication history.
	Save communication history in a file (CSV format).
	Pause recording. The data exchange is no longer recorded until recording is restarted by releasing the button.

5.5.4 Search function

The search function can be used to search the entire content of the log and communication history for a specific term.

Operating element	Function
	Field for entering the search term. Any string of characters can be entered as the search term. ▪ "**x**" deletes the entered search term. ▪ Pressing the ***Enter*** key selects the most recent item containing the search term.
	Used to select the previous and next item containing the search term.

The number of lines in which matching character strings were found is additionally displayed next to the selection arrows.

5.6 Editor

The built-in editor allows various file formats to be edited and executed:

- **Sequence programmes:**

  Programmes that are saved and executed on the control:

  - Motion Control file MC V3.x (*.bas)
  - Motion Control file MC V2.x (*.mcl)

- **Parameter files:**

  MC V2.x parameter file for transferring read parameter sets to the control or for saving them:

  - Parameter file MC V2.x (*.mcp)

- **VB Script programmes:**

  Programmes that run on the PC within the Motion Manager:

  - VB Script file (*.vbs)

- **Text files:**

  Files with any content, e.g. for documentation:

  - Text file (*.txt)

Each new or opened file is shown in a separated tab. When a new programme file is created, a comment header is generated automatically. For VBS files , the body of the main function is inserted. Tabs with programme files have an additional toolbar by which the programme can be started, stopped, uploaded and downloaded or used in other ways. The individual symbols in this toolbar and the procedure for programming and debugging are described in the respective programming chapters or in the respective programming manual.

Pressing the ***Extras*** button (see chap. 8.1.2, p. 62) in the programme files toolbar allows the display of additional functions such as monitoring/changing code templates or variables. Code templates contain popular programming constructs which can be dragged into the programme code and customised there. Conversely a highlighted section of code can be dragged from the Editor window into the code templates toolbar, thereby creating a new template. Added templates can be deleted again by pressing the ***Del*** key on the keyboard.

Buttons

Button	Description
	Create new document
	Open saved document
	Save document
	Save document as
	Copy selected content on to the clipboard
	Insert content from the clipboard
	Call up context-sensitive Help
	Close the open document

5.7 Configuration of the Motion Manager

The **Extras - Options** menu item contains general settings for the Motion Manager. These include:

Setting	Description
General	Options for online updates of the Motion Manager software
Data exchange	Messages filter (CAN only): The Motion Manager has a messages filter which allows certain asynchronous CAN messages to be filtered out. By default all messages from all network participants, other than heartbeat telegrams, are displayed in the communication history of the terminal window. The following filter options are available: ▪ **All of unselected nodes**: Only the asynchronous messages of the selected node are displayed. The asynchronous messages of other network participants are filtered out. ▪ **TxPDOs of selected node**: The PDO messages from the selected node are filtered out. ▪ **Use exceptions**: Messages with one of the specified COB-IDs are excluded from the filtering. **Note:** Heartbeat telegrams from the active node are evaluated only if the NMT state changes. Otherwise they are filtered out by the message filter.
Network client	Management of connection profiles to network servers
Network server	Settings for the use of the Motion Manager as network server

5.7.1 Client-server network functionality

The Motion Manager client-server network functionality allows a remote PC to access a Motion Manager in the same network. The remote PC is connected to the Motion Manager network server via the network as a Motion Manager network client. The Motion Manager network server is directly connected to FAULHABER Motion Controllers via CAN, RS232 or USB.

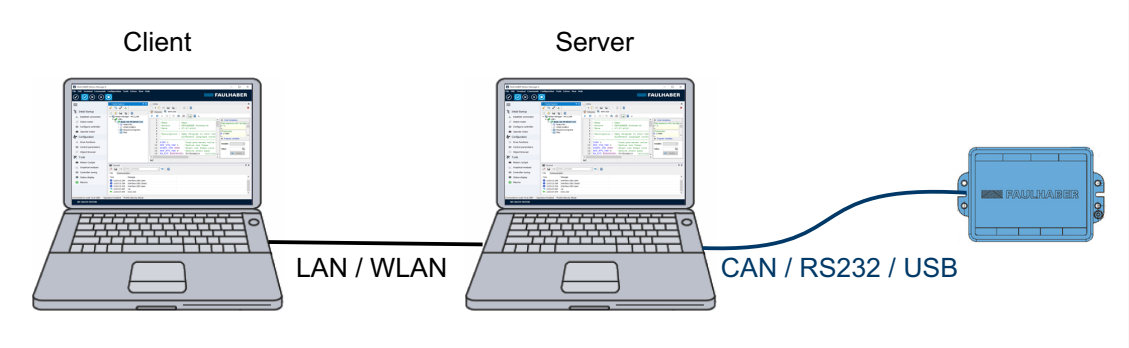


Fig. 2:

5.7.1.1 Configuring Motion Manager network server

1. Establish direct connections (CAN, RS232 or USB) between server PC and the desired Motion Controllers.
2. On the server PC, select the **Network Server** tab in the **Extras - Options** menu of the Motion Manager.
3. Activate **Motion Manager as network server**.
4. Enter any desired name and any desired password for the server PC.
5. Confirm the entries using the ***OK*** button.

↳ The Motion Manager network server now runs on the PC and can be accessed from a remote PC. "Netserver running" is displayed in the status bar.

5.7.1.2 Configuring the Motion Manager network client

1. On the client PC, select the **Network Client** tab in the **Extras - Options** menu of the Motion Manager.
2. Press the ***New*** button to create a new connection profile.
3. Enter the following values of the server PC (see **Network Server** tab):
   - IP address (server IP)
   - Name (ServerName)
   - Password
4. Confirm the entries using the ***OK*** button.
5. If more than one connection profile is displayed, select the desired connection profile.
6. Confirm the selection using the ***OK*** button.

   A selected connection profile can be changed using the ***Edit*** button and deleted using the ***Delete*** button.

5.7.1.3 Establishing client-server network connection

1. On the client PC, select the ***Establish connection*** button on the quick access toolbar or via the **Terminal** menu.
   - ↳ The connection wizard starts.
2. Select the "NET" interface.
   - ↳ All accessible servers of the selected connection profile are displayed in the list of available ports.
3. Select the desired connection.
4. Press the Search button.
   - ↳ The devices that are connected to the PC are displayed.
5. Confirm the selection using the ***Finished*** button.

The devices that are connected to the server PC are displayed in the Node Explorer under a NET node and can be operated in the same way as locally connected devices.

The following restrictions apply with respect to a direct connection:

- Only SDO and EMCY services are supported (not NMT, not PDO).
- For the graphical analysis, only the trace recorder function is supported (not trace logger).
- The communication parameters of the devices cannot be accessed.
- Access to devices of the MC V2.x controller family with RS232 interface is not supported.

Only one network adapter may be activated in the client PC.

If a network connection cannot be established, other network adapters may need to be disabled through the Windows Control Panel.

6 Commissioning the control

The commissioning of FAULHABER controls is supported by various wizards and dialogues which are available in the Motion Manager. The functions of the Motion Manager that are used for the respective steps in commissioning are listed in the individual sub-chapters.

6.1 Establish connection

In order to communicate with the control in question, a connection with the control must be established by the PC on which the Motion Manager is installed. A wizard is available for setting up the communication connection via a one of the supported interfaces. The wizard appears automatically when a new project is created. It can be called up at any time by pressing the ***Establish connection*** button in the quick access toolbar or via the **Terminal** menu.

Extended options for setting up connections can be called up via the **Manage terminal connections** menu.

After a connection to the control has been established, additional functions are available for customising the connected drive unit.

6.2 Selecting the motor

So that the connected motor can be reliably operated, the drive and controller parameters of the control must be customised to suit the motor and the sensor system that is used. An additional wizard is available for this task. It can be called up using the ***Select motor*** button in the quick access toolbar or via the **Configuration – Commissioning** menu.

Once all the steps in the wizard have been completed successfully, the drive parameters and the motor controller are adapted to the properties of the motor and to the sensor systems used, so that the motor can be operated in the idle state (i.e. without any additional load).

The actual configuration of the drive parameters is controller-specific. Therefore the structure of the wizard is adjusted to the different device families.

Motor selection	MC V3.x	MC V2.x	SC
Wizard for motor selection	✓	✓	✓

6.3 Setting up controller

After basic configuration of the current controller, speed controller and position controller by selecting the motor, the controllers must be adapted to the properties of the system in question. For this purpose, information is required about the system. This information can be used for optimising the settings of the controller parameters. Wizards and dialogues, which have been adapted to the properties of the device families, are provided for this purpose.

The controller parameters can also be configured via the configuration dialogue for the drive functions or via the window for dynamic setting of the controller parameters. It is however strongly recommended that the basic setting is first performed using the wizard.

Controller setting	MC V3.x	MC V2.x	SC
Wizard for controller configuration	✓	–	–
Controller setting via the wizard for motor selection	–	✓	✓
Configuration dialogue for the drive functions	✓	✓	✓
Dynamic configuration of the controller parameters	✓	✓	–
Controller tuning	✓	✓	–

6.4 Operating the motor

After the drive and controller parameters have been configured for the application, the motor can be operated. The Motion Manager offers many facilities, which can be used for various purposes.

Simple operation	MC V3.x	MC V2.x	SC
Dialogue for motor operation	✓	✓	–
Motion Cockpit	✓	✓	–
Motor operation via the configuration dialogue	–	–	✓
Controller tuning	✓	✓	–

Extended operation	MC V3.x	MC V2.x	SC
Macros	✓	✓	–
Sequence programmes	✓	(✓)	–
VB script	✓	✓	–
Individual commands	✓	✓	–

6.5 Extended functions

The overall scope of functions of the FAULHABER Motion and Speed Controller is described in the manuals for the controls (www.faulhaber.com/manuals). Most of these functions can be configured using the configuration dialogue for the drive parameters of the respective control. Further dialogues and tools are available for special functions and for analysis of the drives.

Configuration	MC V3.x	MC V2.x	SC
Connection parameters	✓	✓	–
Drive functions	✓	✓	✓
CANopen standard functions	(✓)	(✓)	–
Firmware update	✓	✓	✓

Analysis	MC V3.x	MC V2.x	SC
Graphical analysis	✓	✓	–
Status display	✓	✓	–

7 Controller-specific functions

Controller-specific functions are available only when the controller is connected. The items in the main menu and for quick access are extended depending on the controller that is connected.

7.1 MC V2.x / SC controller family

The functions described in this chapter are available for the Motion Controller of the MC V2.x family and the Speed Controller of the SC series (see chap. 1.1, p. 6).

7.1.1 Wizard for motor selection

The wizard for motor selection can be called up via the ***Select motor*** button in the quick access toolbar.

This wizard permits customisation of an external control to the connected motor. For this, the connected FAULHABER motor must be selected from a list.

The motor data, current limitation values and additionally calculated controller parameters must be set for the selected motor. This function can also be used for integrated units.

The wizard for motor selection is divided into 5 sections and is dynamically matched to the selected configuration. It may be possible that not all listed configuration facilities are available for the control in question.

Step 1: Select motor
FAULHABER catalogue motors that are supported by the control selected in the Node Explorer are available for selection.

The upper list field contains the motor types supported by the control. The two fields below list the Product Code that can be found in the label of the motor.

Motors not included in the list can be added to the list using the ***Create*** button. In this case, the data sheet values of the motor must be input manually.

In addition, the sensor type (Hall sensors, incremental encoders, etc.) used for the motor must be selected. Here a selection of sensor types suitable for the control is listed as well.

Speed Controllers have no check available to determine which motor types and sensor types are supported by the connected control. Therefore all possible configurations are displayed here for selection.

The motor voltage applied (voltage value set at the power supply unit for the motor power supply) is either read automatically by the control (Motion Controller) or must be input manually (Speed Controller). It is not mandatory that this value matches the nominal voltage of the motor.

NOTICE!
Applying a voltage value that is too high, or inputting an incorrect voltage value can lead to material damage.

- Comply with the permissible voltage ranges for the respective control (see the control data sheet).

Step 2: Load transmission
Different types of load transmission (gearbox, spindle, …) can be selected in this section. The settings that are made, affect the design of the controller within the control in question.

Step 3: Factor of inertia
The factor of inertia K_J is used in the calculation of the motor controller parameters. This factor of inertia is calculated by the following formula from the mass moment of inertia J_M of the motor and the mass moment of inertia J_L of the coupled load:

$$K_J = \frac{J_M + J_L}{J_M}$$

The mass moment of inertia of the motors is referred to as the rotor moment of inertia on the data sheet. The mass moment of inertia of the load must be determined or estimated. The value can be set via a slider or input into the respective input field. For linear motors, the mass must be input instead of the mass moment of inertia.

The inertia factor is limited to 30 for calculating the controller parameters.

Step 4: Controller setting
A further pre-setting for the calculation of the controller parameters is the data whether the controller should be optimised for quiet running or for high dynamic response.

In some circumstances the desired operating speed is requested after the ***Next*** button has been pressed.

Step 5: Overview
A summary of the controller parameters determined for the selected motor/sensor combination is displayed on the closing page of the wizard.

Incorrect settings can be adjusted by pressing the ***Back*** button. By pressing the ***Finish*** button, the motor data and the controller parameters that were determined are transferred to the control.

Speed Controller:
For Speed Controllers, the data cannot be sent directly. They are used for pre-assigning new settings in the configuration dialogue. In order to transfer the new data to the control, the ***Send*** button must be pressed in the configuration dialogue.

Motion Controller:
For Motion Controllers, the data are sent directly by the respective command. If the parameters should be retained even after the control is switched on again, a SAVE command must also be performed. Confirmation of this is then requested by an appropriate message.

After a new brushless motor with analogue Hall sensors has been connected to a Motion Controller, the Hall sensor signals must be optimised after transfer of the motor and controller parameters. To do this, press the ***Configuration drive functions*** button or menu item to start the configuration dialogue.

In the **Basic setting** tab, the adjustment can be performed via the ***Optimising to the connected motor*** button (see chap. 7.1.2, p. 33).

7.1.2 Configuration of the Motion Controller

7.1.2.1 Drive functions

The commands for the changed settings are loaded. The new setting is active immediately and remains in force until the power supply to the control is switched off.

If the new setting should be saved permanently, the ***EEPSAV*** or ***SAVEAPP*** buttons must then be pressed. This loads the current parameters to the non-volatile memory in the drive unit.

Motion Controller with CAN interface CF series

For Motion Controllers with CAN interface CF series or older the parameter commands for the changed settings are sent via the FAULHABER channel to PDO2. The configuration dialogues are accessible only in the *Operational* state. The drive unit must previously be set to the appropriate state via the Command Network Management menu (**Start Remote Node**). Here the **Basic settings** tab is available only in FAULHABER mode (OPMOD-1), since this is done using FAULHABER-specific configurations which are not supported by the CANopen standard.

In the **Basic settings** tab, the function **Optimising to the connected motor** is also available for the BL and LM controllers. At first, switch from another operating mode into FAULHABER mode (OPMOD-1 in the **Mode** tab) in order to perform the optimisation (see Optimising to the connected motor (MCBL / MCLM) in this chapter).

Motion Controllers with CAN interface CO series

For Motion Controllers with CAN interface CO series, the parametrising is performed via the CANopen object dictionary. The changed parameters are sent via SDO communication.

Optimising to the connected motor (MCBL / MCLM)

If a new BL motor or linear motor was connected to the controller, an adjustment of the Hall sensor signals should be performed after the new motor parameters have been set. In the **Basic settings** tab of the configuration dialogue there is a button available for this purpose.

Hall sensor signals that are not optimised can lead to rough running by the motor in the first seconds after switching on, and to lower accuracy.

In order to adapt an MCBL controller even better to the connected motor, an additional optimisation of the phase angle of the sinus commutation can be performed. A phase angle that is not optimised leads to a higher power consumption and poorer efficiency.

Make sure that during both optimisations the motor is free for several seconds and can move without load.

After the ***Optimising to the connected motor*** button has been pressed, you will be guided through the automatic optimisation of the Hall sensor signals and the phase angle.

After optimisation has been completed, the system parameters that were determined must be permanently saved in the control by pressing the SAVE command.

If the connected motor cannot be operated at maximum speed in idle state, because e. g. a gearhead is flanged to it, it may not be possible to use automatic setting of the phase angle. If this is the case there is still the option of correcting the phase angle manually:

- The desired output voltage should be set on the respective screen.

  The setting can be performed most accurately at 100% (equivalent to 15,000 min^{-1} or no-load speed of the drive). With certain connected drive units it is recommended that the value is reduced, in order to not to exceed the maximum gear input speed, for instance.
- Then move the slider for the phase angle in one or the other direction and monitor the current value. The phase angle is then set to the optimum when the smallest current value is displayed.
- Click on ***Next*** to exit this page, after you have successfully performed manual adjustment. The Hall sensor signals must possibly be optimised once more. Finally, perform another SAVE command, in order to permanently save the system parameters that were determined in the control.

7.1.2.2 Controller parameters

Dynamic setting of the controller parameters

As well as the possibilities offered by the motor selection wizard and the controller tuning wizard, a separate dialogue is available for setting the controller parameters. Under the **Configuration - controller parameters** menu item, the controller parameters can be changed online by moving the arrows in the input fields or by inputting a value with the keyboard. As soon as the value of an input field has been changed, this is automatically sent to the drive. For keyboard inputs the respective field remains grey until the field is exited or the ***Enter*** button is pressed. Only then is the value sent to the drive.

This permits dynamic adjustment of the parameters, by a technique similar to using a potentiometer.

The controller parameters dialogue is a non-modal dialogue which can be opened in addition to other open windows. Thus the controller parameters can be changed whilst the graphical analysis is displayed, and the effects on speed stability or position stability can be monitored.

After setting the optimum controller parameters, perform ***SAVE*** or **EEPSAV** so that the parameters remain saved even after the control has been switched off. See "Setting the controller parameters" in the Communications and Functional Manual.

7.1.2.3 Connection parameters

As well as a dialogue for configuring the connection parameters of the Motion Manager, a dialogue for the connection parameters of the control is also available. It is accessed via the **Configuration** menu. This dialogue allows configuration of the node number and if necessary the transfer rate of the control. After confirmation of the changes, the connection parameters of the Motion Manager are updated. For controls with CAN interface, additional inputs are necessary (see chap. 7.3.4, p. 58).

When operating in a network, make sure that no node addresses have been assigned more than once and that all nodes are operating at the same transfer rates.

7.1.3 Configuration of the Speed Controller

For the Speed Controllers, the change in configuration is performed by a firmware download.

In order to immediately check the effects of the change, you can press the ***Run*** button after download, which switches the drive unit from configuration mode into operating mode. No configuration changes can be performed in operating mode. To perform further changes, the ***Stop*** button must be pressed. This switches the drive unit back into configuration mode.

The control can be switched into configuration mode only after the power supply has been switched on. If during configuration the connection is interrupted, press ***Run***/***Stop*** to return to configuration mode.

Exiting the configuration dialogue also switches the drive unit into operating mode. The drive unit then immediately starts up with the configuration that has been set. If the drive should not be started immediately, the power supply must be switched off first.

7.1.4 Controller tuning

The controller tuning wizard can be called up by pressing the ***Controller tuning*** button in the quick access toolbar.

This wizard offers functions to accept and evaluate step responses using graphical analysis and permits manual optimisation of the controller parameters.

Motion Controllers with CAN interface must be in the *Operational* NMT state in order to open the graphical analysis.

The buttons in the toolbar permit activation and deactivation of the drive (enable/disable the output stage), setting the intervals for the step sequences and the size of the target corridor.

NOTICE!

Material damage due to collisions.

When performing step sequences, the drive moves according to the values that were input. If there are obstacles within the movement range, this can lead to collisions.

- Make sure that when performing step sequences the drive is free to move within the values that were input.

The following steps should be performed for optimising the controller parameters. For positioning tasks it is recommended that the speed controller is optimised first, followed by the position controller.

For Motion Controllers of the CO series, the object TxPDO4 is required for recording the analysis data, which is temporarily reconfigured for this purpose. When the wizard is closed, the original mapping of the object TxPDO4 is restored.

Step 1: Record step response
In the first step the desired set values for speed and position must be set. After the ***Start*** button has been clicked, the set values are sent to the controller alternately at the set intervals.

Depending on the setting, one or more step responses for the selected size are executed and displayed.

During the execution of a continuous sequence, the slider for the control gain is changed whilst the effects on the speed signal and position signal are monitored.

Step 2: Evaluate the step response
After at least one complete step response has been recorded, it can be evaluated using the **Analysis** tab. If several steps are performed within a (continuous) sequence, the last step response is displayed.

With the ***Scrolling arrow keys*** you can switch back and forth between analysed step responses in order to compare results. The ***Recycle bin*** button deletes the displayed analysis result from the list of step responses.

Tab. 1: Values of the analysis

Value	Meaning
Control rise time	Period between exit from the starting corridor and first entry into the target corridor.
Control settling time	Period between exit from the starting corridor and last entry into the target corridor.
Overshoot	Maximum deviation of the actual value from the set value.

Step 3: Optimising the controller parameters
If further optimisation of the drive behaviour is required, the slider in the **Speed controller** tab or **Position controller** tab can be used to modify the gain of the respective controller; another sequence can then be started.

The last step response of the newly recorded sequence can be evaluated once again using the **Analysis** tab, and compared with the step previously recorded.

With the ***Apply*** button this setting can be loaded once again to the drive, and after confirming the security query, it is saved permanently.

Use the motor selection wizard for the basic setting of the controller.

For positioning tasks, first set the speed controller to the most dynamic setting possible, i.e. with the least overshooting and short rise time and transient time, and then optimise the position controller.

Not all controller parameters can be set using the controller tuning wizard. For extended settings, use the **Controller parameters** form under the **Configuration - controller parameters** menu item. This form can also be displayed together with the trace window for graphical analysis, in order to monitor the effects of changes to the control behaviour.

7.1.5 Trace function

The trace function allows recording of up to 2 parameters of the control in logging mode. The parameter values are continuously requested and read. For drives with CAN interface, the PDO communication service is used.

Motion Controller with RS or CF interface:
For each curve a selection list is available by which the data sources to be logged can be selected. The lists contain specified data sources supported by the respective control. There is also the option to add parameter numbers supported by the control. Once a data source has been selected, further setting options for scaling are available.

Motion Controller with CO interface:
The **Source** selection allows one of the available transmit PDOs for the control to be selected. The parameters assigned to the respective PDO by PDO mapping can be used as data sources. The PDO mapping dialogue can be used to assign any mappable parameters from the object dictionary to a trace PDO (e. g. TxPDO4), which then are available as data sources (see chap. 7.3.3.2, p. 57). Every available data source contains its own area for deactivating and activating the scaling, and for setting it.

Motion Controllers with CAN interface must be in the Operational NMT state to allow running of the trace function.

Display and scaling:

- Switch automatic scaling on and off. For manual scaling the Y-axis has the values listed below for the axis minimum and maximum.
- Activate or deactivate synchronisation with other Y-axes. A change made to an axis setting is applied to the other axes.
- Centring curve. If automatic scaling is active, the current axis setting is entered in the input fields. Otherwise the axis minimum and axis maximum are shifted and the spacing remains unchanged.

7.1.5.1 Trace settings

Trigger tab
The following settings can be specified for the **Signal recording** area:

Continuous:	The recording is performed continuously.
Single shot:	If the selected trigger source exceeds the set limit for the trigger threshold, the recording will be stopped.
Types of trigger:	Selection from the specified types of trigger and an optional trigger delay time (not for controls of the CO series).

Buffer tab
The following settings can be specified for the data acquisition area:

Number of data packets per request:	The specified number of data packets per request is sent to the PC by the control.
Time resolution:	The time resolution can optionally be specified by the controller or by the PC. If specified by the PC, the desired sampling interval can be set, otherwise the data exchange takes place as quickly as the available communications method permits.
Total buffer size:	The number of values specified here is saved temporarily during the recording. During longer recordings the older values are overwritten (ring buffer)
Fixed X axis:	If the "fixed X-axis" setting is active, a fixed period of time is shown in the display window.

Curve tab
The following settings can be performed in the display area:

Curve:	Parameters for the display of the curve can be set.
Physical unit/conversion factor:	The raw data supplied by the controller can be converted into a definable unit, if required.

The controller always supplies raw data. The conversion is performed on the PC.

7.1.6 Status display

The status display can be started via the **Tools - Status display** menu, providing this function is supported by the selected control.

Changes to the listed values are indicated by displaying and hiding a tick before the value designations. The display is updated every 500 ms.

7.1.7 Sequence programmes

Motion Controllers which support the saving and execution of sequence programmes offer functions for editing, loading, debugging and managing these sequence programmes.

Loading an existing programme sequence:
Existing programme sequences can be loaded to a file editor window via **File - open**.

File format:
Motion Control files MC V2.x have by default the ending *.mcl. Since the mcl files are saved in ASCII format, files can be read irrespective of which text editor they were created with.

Create sequence programme:
To create a new sequence programme from the context menu, select the ***Motion Control file MC V2.x*** item in the **File - New** menu.

Now the code can be entered. When programme edit mode (**Edit - programme file** menu) is activated, the commands in the command menu can be loaded directly into the programme code.

A toolbar is available under the File tab which allows sequence programmes to be loaded, executed and debugged.

Syntax explanation:

- Each line contains a command, which may be followed by an argument in the form of a number (e.g.: LA1000).
- Spaces at the start of the line and between the command and argument are ignored. Only alphanumeric characters are sent.
- In addition to commands, comments can be entered. Comments are introduced by a semicolon (;) and can be written at the end of a command line or in a separate line (e.g.: HO ;Define home position).
- Comments are not sent to the drive, their function is purely to document the programme saved on the PC.
- In principle, any letter and any number of a programme line up until the occurrence of a semicolon will be sent to the Motion Controller. The Motion Controller saves each programme line that contains a valid command.
- The commands **PROGSEQ** and **END** need not be input, since these are sent automatically by the **Load programme file** function.

To exit the programme edit mode, select the **Edit - programme file** menu item again to deactivate it. Now commands from the command menu can once against be sent directly to the drive.

Load the sequence programme to the control:
The programme that was input or loaded can be sent to the control by the **Commands - Load file** menu item following the selection of **Sequence programme**.

Alternatively the sequence programme can also be transferred by pressing the ***Start*** button on the File tab. It will then start immediately.

Compare sequence programmes:
The programme code from the Edit file window can be compared to the programme code saved in the control, to check that it matches. This is done by the **Commands - Compare files** menu item, followed by the selection of **Sequence programme**.

If the sequence programme is loaded via the Debug toolbar, it is automatically checked with an associated syntax error display before the programme is started.

Load the sequence programme from the control into the Motion Manager:
A sequence programme saved in the control can be loaded to the Motion Manager by means of the **Commands - Receive file** menu item, followed by the selection of **Sequence programme**. The programme code is now displayed in a new file tab, under which it can be edited, saved, printed and also loaded again.

Start sequence programme:
After the programme has been sent to the control, it can be started via the **ENPROG** command or by clicking on the ***Execute sequence programme*** button.

When the sequence programme is loaded via the debug toolbar it starts automatically.

Debug sequence programme:
An additional toolbar is available in the file editor window for the Motion Control file format V2.x:

Symbol	Function
	**Load and execute sequence programme:** After the sequence programme has been loaded it is first read back and checked for syntax errors. If syntax errors are present the command line in question cannot be interpreted by the Motion Controller and the defective programme line is then shown in the editor window in red. If the programme is found to be error-free after loading, the entire file editor window is shown in grey. In this state the programme will run on the controller, and debug mode is activated.
	**Single step:** The displayed sequence programme is loaded to the Motion Controller and compared. If the programme is found to be error-free after loading, the entire file editor window is shown in grey, and the first programme line is shown in green. The sequence programme is now at the displayed programme line and by pressing the button can be advanced one line at a time.
	**Pause sequence programme:** If the programme is running in debug mode, the programme that is running can be interrupted by pressing this button. The current programme line is then shown in green in the editor window.
	**Stop sequence programme:** If the programme is running in debug mode, pressing this button stops the programme and exits debug mode. The file editor window switches back into Edit mode, so that once again changes can be made to the programme code.

Examples of sequence programmes can be found in the installation directory under **Motion Manager 6\Examples**.

7.1.8 Parameter files

In addition to the file functions for sequence programmes there are functions for loading, receiving and comparing parameter files and for complete configuration of parameter files and sequence programmes.

Load an existing parameter file:
Existing parameter files (with and without sequence programme) can be loaded into the file editor window by means of **File - open**.

Motion Controller parameter files are text files and have by default the ending *.mcp.

Transfer parameter file:
A loaded parameter set can be sent to the control via the **Commands - Load parameter file** or **Commands - Load file** menu items followed by the selection of **Parameter file**.

If a configuration file with programme sequence and parameter listing is loaded, this can be loaded by means of **Parameter file with sequence programme**.

Receive parameter file:
An image of the drive configuration saved in the control can be created by means of the **Commands - Receive parameter file** or **Commands - Receive file** menu items, followed by the selection of **Parameter file** or **Parameter file with sequence programme**.

The parameter configuration that was read (with and without the sequence programme) is now displayed in a new file tab, under which it can be edited, saved, printed and also loaded again.

Compare parameter file:
A check whether the loaded configuration file matches the current configuration of the control can be performed using the **Commands - Compare parameter file** or **Commands - Compare file** menu items, followed by the selection of **Parameter file** or **Parameter file with sequence programme**.

Edit parameter file:
For documentation purposes, parameter files can be provided with comments before they are saved.

7.2 MC V3.x controller family

The functions described in this chapter are available for the Motion Controller of the MC V3.x family (see chap. 1.1, p. 6).

7.2.1 Wizard for motor selection

This wizard can be called up by means of the ***Select motor*** button in the quick access toolbar or by means of the **Configuration – Commissioning** menu.

The wizard permits customisation of a FAULHABER control to the connected motor and the sensor system that is used. The configuration is divided into several steps and is dynamically matched to the selected control. Thus it may happen that not all configuration facilities listed here are available for the control in question.

NOTICE!
Damage to the motor may result if the wrong motor is selected or incorrect data sheet values are input.

- Make sure that the correct motor is selected and the correct motor data are input.

1. Select the motor used from the list of FAULHABER catalogue motors. If the motor used is not in the list:
   - Input the data sheet values of the motor and create a new motor based on these values.
   - The newly created motor is then permanently available and can if required be edited or removed.
2. Input the sensor systems that are used.
   - Often the motor already has an integrated or attached sensor system (see product description of the motor).
   - If additional sensor systems are used, the selection lists must be used to specify which sensor system is connected to which connection of the Motion Controller.
3. Assign functions to the specified sensor systems.
4. Check the specified configuration in the overview and load it to the control.

As well as performing the configuration of the motor parameters and sensor systems, the wizard also configures other features in the background:

- The controller and profile parameters are configured so that the drive can be run in idle state.
- The factor group (configuration of the user unit for speed and position) is reset.
- The speed and position windows are reset.
- Current and speed pre-control are deactivated.

5. If analogue Hall sensors are used:
   a) Make sure that the motor is free and can move without load
   b) Adjusting Hall sensor signals.

   ↳ Hall sensor signals that are not optimised can lead to rough running of the motor and to lower accuracy.

7.2.2 Wizard for controller configuration

For stable control, the controller must be configured to the application. The wizard can be called up by means of the ***Configure controller*** button in the quick access toolbar or by means of the **Configuration – Commissioning** menu.

1. Determine system parameters.

   In order to match the controller parameters to the system to be controlled (controlled system), the parameters of the systems must be determined in the following way:

   Manual input

   The factor of inertia K_J is used in the calculation of the motor controller parameters. This factor of inertia is calculated by the following formula from the mass moment of inertia J_M of the motor and the mass moment of inertia J_L of the coupled load:

   $$K_J = \frac{J_M + J_L}{J_M}$$

   The mass moment of inertia of the motors is referred to as the rotor moment of inertia on the data sheet. The mass moment of inertia of the load must be determined or estimated. The value can be set via a slider or input into the respective input field. For linear motors, the mass must be input instead of the mass moment of inertia.

   Automatic identification

   Automatic identification can be used only if a speed sensor is fitted to the motor.

   NOTICE!
   When automatic identification is performed, the drive will move. Obstacles within the movement range can lead to collisions and material damage.

   - When performing automatic identification make sure that the drive is free to move within the input values.

     For automatic identification, the system parameters are determined by a specific measurement process. For the determination of the mass of inertia, the drive is activated for a short time in current controller mode at the rated torque of the motor to deliver constant acceleration up to a certain speed. The maximum permissible speed and specified range of movement are complied with during this operation. The measurement can be cancelled at any time.

     If during the identification process the motor is restrained by mechanical limitations, identification is invalid. In this case, the start position of the drive must be corrected using the function provided and the identification process repeated.

     After identification, the identified parameters should be viewed and corrected if necessary.

     ↳ When the required parameters are to hand, the controller parameters are determined and displayed on the last page of the wizard.

2. Check the controller parameters and load them to the control.

7.2.3 Setting the drive functions

A comprehensive graphical configuration dialogue is available for the connected control. This permits easy drive configuration and parametrising (***Drive functions*** button in the quick access toolbar or **Configuration** menu).

The dialogue is tailored to the functions of the control and, unlike the wizards, offers extended configuration functionality. The tree structure on the left side of the window allows switching back and forth between individual input pages. The back button in the input pages header allows switching to the previous page.

The shown parameters correspond to the configurable objects in the control. If the mouse pointer is briefly hovered over an input element, a tooltip is displayed, with information on the object.

Changes can be made directly at the graphical user interface and loaded to the drive unit by pressing the ***Send*** button. The new setting is active immediately and remains in force until the power supply to the control is switched off.

If the new setting should be saved permanently, the ***Save*** button must then be pressed. This loads the current parameters to the non-volatile memory in the drive unit.

The meaning and functions of the individual objects are described in the Functional Manual.

7.2.4 Changing the controller parameters

The **Controller parameters** window allows controller parameters, filter settings and profile parameters to be changed whilst the drive is in operation. In contrast to the other configuration dialogues, which remain in the foreground until they are closed, the controller parameters dialogue can be used in parallel with other windows.

1. Open the **Controller parameters** window using the ***Controller parameters*** button in the quick access toolbar or via the **Configuration** menu.
2. Change the settings of the desired parameters via the arrows in the input fields or by inputting the values at the keyboard.

   ↳ A value that is changed is automatically sent to the drive. When making entries via the keyboard, the respective input field remains grey until the input field is exited or the ***Enter*** key is pressed.

Buttons in the toolbar

Button	Function
	Permanently saves the current settings in the control
	Opens the configuration dialogue for the drive functions, to allow further parameters to be set
	Shows an overview of current controller configuration

7.2.5 Connection parameters

As well as a dialogue for configuring the connection parameters of the Motion Manager, a dialogue for the connection parameters of the control is also available. It is accessed via the **Configuration** menu.

This dialogue allows configuration of the node number and if necessary the transfer rate of the control. After confirmation of the changes, the connection parameters of the Motion Manager are updated.

For controls with CAN interface, additional inputs are necessary (see chap. 7.3.4, p. 58).

When operating in a network, make sure that no node addresses have been assigned more than once and that all nodes are operating at the same transfer rates.

7.2.6 Controller tuning

The **Controller tuning** tool can be opened via the ***Controller tuning*** button in the quick access toolbar or via the **Tools** menu.

It offers functions to accept and evaluate step responses using graphical analysis and permits manual optimisation of the controller parameters.

The buttons in the toolbar permit activation and deactivation of the drive (enable/disable the output stage).

NOTICE!

When performing step sequences, the drive moves according to the values that were input. Obstacles within the movement range can lead to collisions and material damage.

- Make sure that when performing step sequences the drive is free to move within the values that were input.

The following steps must be performed for optimising the controller parameters. For positioning tasks it is recommended that the speed controller is optimised first, followed by the position controller.

1. Record step response :
   a) Setting the desired set values for speed and position.
   b) Press the appropriate button to select step response or continuous sequence:

Button	Function
One step response	Recording an individual step response.
Continuous	Execution of a continuous sequence, to enable adjustment of the controller parameters in the active operation by continuous repetition of the step.

2. Evaluate step response.

   After at least one complete step response has been loaded, this can be evaluated on the **Analysis** page. If multiple steps have been performed (continuously) within a sequence, only the last complete step response is displayed.

   - **Step response**:

     An overview of the key data on the recording shown are displayed in this area. The target corridor shown corresponds to the corridor that is set in the control. This is used for the evaluation of the step response. The following variables are analysed in the evaluation area:

Tab. 2: Values of the analysis

Value	Meaning
Control rise time	Period between changing the set value and the first entry of the actual value into the target corridor.
Control settling time	Period between changing the set value and the last entry of the actual value into the target corridor.
Overshoot	Maximum deviation of the actual value from the set value.

   - **Comparison**:

     This area permits toggling back and forth between the recorded step responses, to allow comparison between the results. The ***Recycle bin*** button deletes the displayed recording from the list of step responses. The ***Apply parameters*** button loads the controller parameters shown in the lower area to the control. The display of the parameters corresponds to the configuration that was effective at the time of recording the displayed step response.

Use the controller configuration wizard for the basic setting of the controller.

For positioning tasks, set the speed controller to the most dynamic setting possible first, i.e. with the least overshooting and short rise time and transient time, and then optimise the position controller.

Not all controller parameters can be set using the controller tuning tool. For extended settings, use the **Controller parameters** form in the quick access toolbar or in the **Configuration** menu. This form can also be displayed together with the trace window for graphical analysis, in order to monitor the effects of changes to the controller behaviour (see chap. 7.5, p. 60).

7.2.6.1 Adjust controller parameters to application

In order to get a powerful positioning or speed-regulated system with regard to dynamics and disturbance rejection, the feedback control parameters must be adapted to the application.

Further details (including troubleshooting) are described in application note 151 (EN) which is available via your FAULHABER sales partner.

✓ The complete drive system, including power supply unit, is present.

1. Use the motor selection wizard to adapt the control to the connected motor (see chap. 7.2.1, p. 42).
2. Use the controller configuration wizard to specify the inertia of the load (see chap. 7.2.2, p. 43).
3. Start the **Controller tuning** tool.

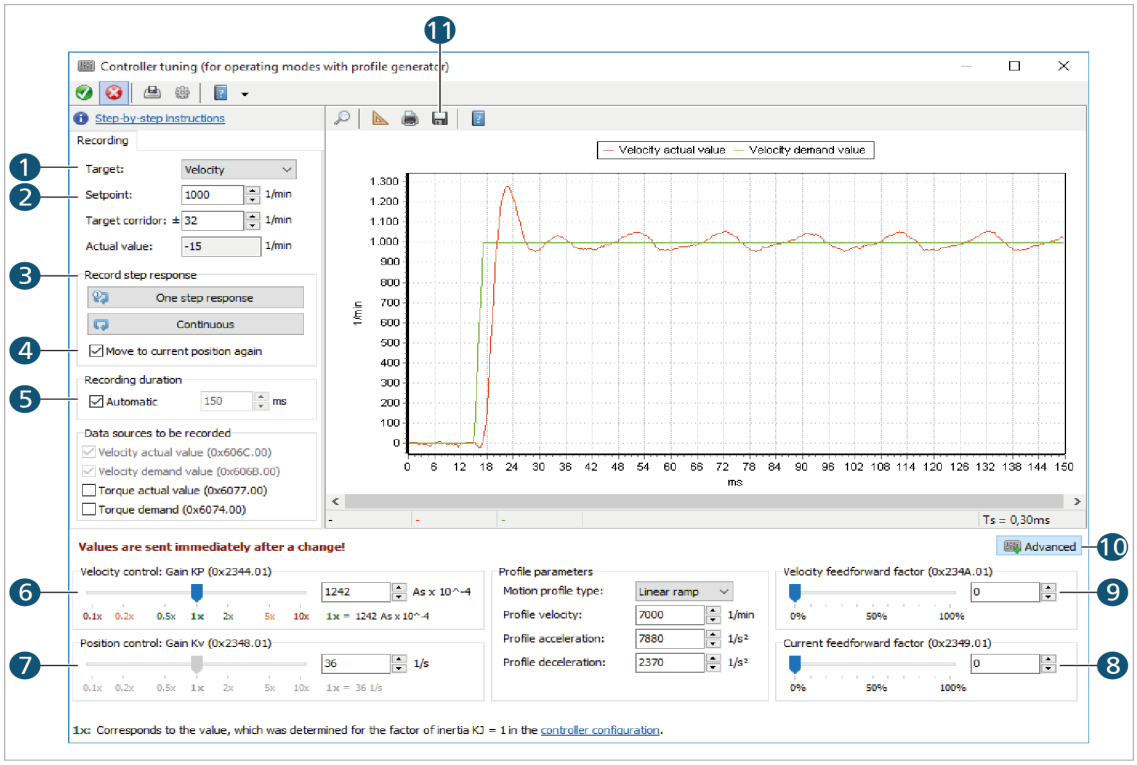


Fig. 3: Controller tuning tool

4. Set the control gain K_P (speed control loop) based on the step response:

   a) Select **Velocity** as target (1).

   b) Set Setpoint (2) to 1000 rpm.

   c) Ensure that **Move to current position again** (4) is activated.

   d) Ensure that the **Recording duration** is set to **Automatic** (5).

   e) Record step responses (3) and increase control gain K_P with the velocity control slider (6) until the system behaviour is well-attenuated.

   Depending on the sensor type, the following values for the control gain K_P are plausible:

	Analogue Hall sensors	Optical incremental encoder	Magnetic incremental encoder
K_P	1.5x…3x	3x…5x	1.5x…5x

↳ The following tuning results are characteristic of well-attenuated system behaviour (see Fig. 4):

- The first overshooting is within the range of 110…135% of the set value
- In comparison to a control gain K_P = 1x, the first overshooting has a reduction of 50…250 rpm.
- The second overshooting is low in comparison to the remaining ripple.
- The system then remains in the target corridor.

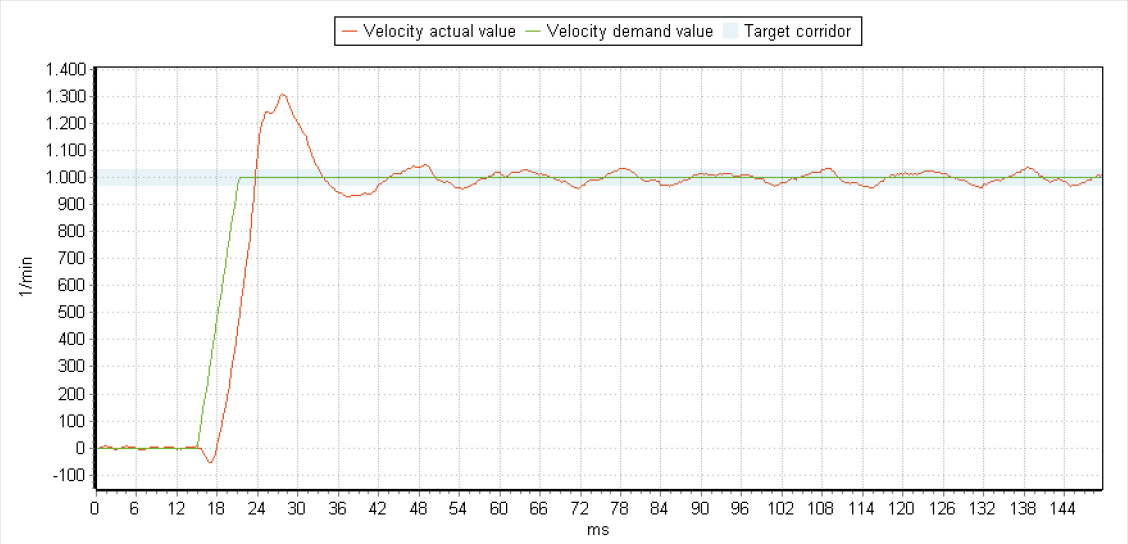


Fig. 4: Example of well-attenuated system behaviour

5. Set the control gain K_V of the position control based on the step response:

   The performance of the position control mainly depends on having a well-tuned speed control loop. The speed control loop must therefore always be set before the position control loop. (see step 4.).

   a) Select **Position** (relative) as target (1).

   b) Set Setpoint (2) to 1/4 motor revolutions.

      If analogue Hall sensors are being used, this corresponds to 1024 increments.

   c) Record step responses and increase control gain K_V using the position control slider (7) until the desired dynamic is reached.

      Depending on the sensor type, the following values for the control gain K_V are plausible:

	Analogue Hall sensors	Optical incremental encoder	Magnetic incremental encoder
K_V	2x…5x	5x…10x	2x…10x

   ↳ The following tuning results are characteristic of well-tuned dynamics (see Fig. 5):

   - The step response does not show any overshooting.
   - The step response has a following error of 5…20 ms.

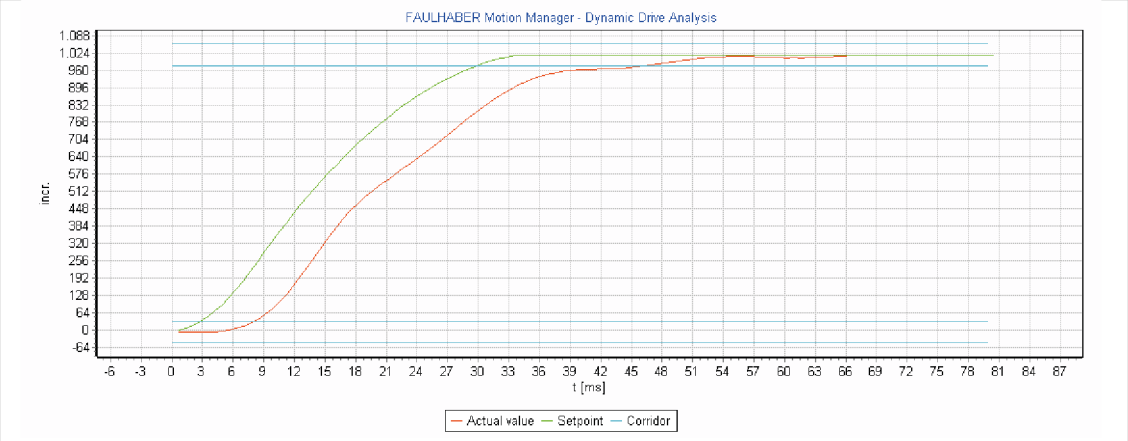


Fig. 5: Example of well-tuned dynamics

6. Save the determined values (11).
7. Option: carry out feedforward control:

   Feedforward control must only be used if the system is already well-tuned (see steps 1. … 6.). Otherwise, feedforward control will conceal controller performance that is otherwise poor, which will lead to poor disturbance rejection.

   a) Pressing the ***Advanced*** (10) button displays the advanced settings.

   b) Set the velocity feedforward control slider (9) to 100%.

      ↳ The following tuning results are expected (see Fig. 6):

      - The following error becomes smaller.
      - The control settling time can be reduced to 1…10 ms depending on how well tuned the system was beforehand.
      - The velocity feedforward control can cause a slight amount of spurious oscillation. In this event, additional current feedforward control may be useful (see substep c)).

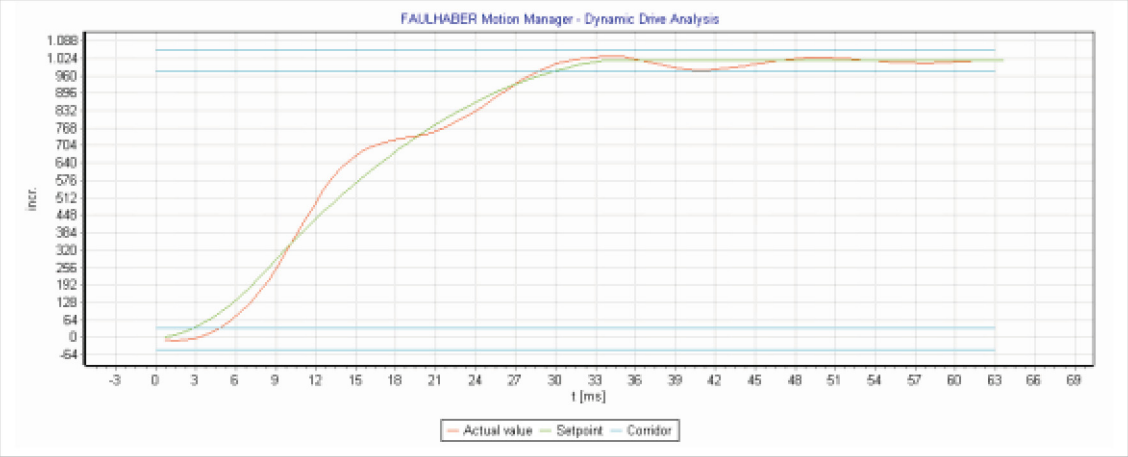


Fig. 6: Expected tuning results with velocity feedforward control

c) Set the current feedforward control slider (8) to 80…100%.

↳ The following tuning results are expected (see Fig. 6):

- The following error becomes minimal.
- The spurious oscillation tendency should be reduced.

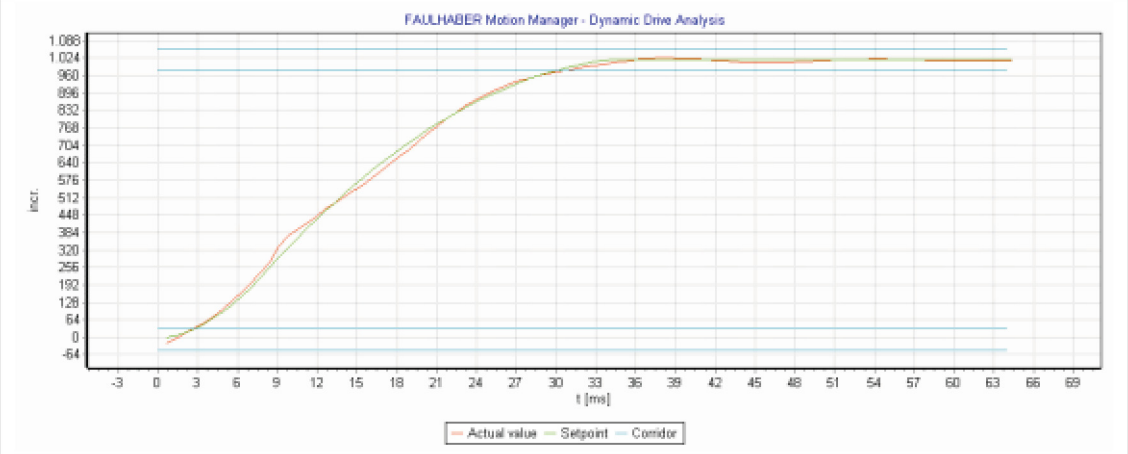


Fig. 7: Expected tuning results with current feedforward control

7.2.7 Trace function

The trace function allows recording of up to 4 parameters of the control.

Types of recording
Two different types of recording are available:

Type of recording	Description
Logger	The parameter values are continuously requested and read. Thus, continuous data recording over a longer period of time is possible. For drives with CAN interface, the PDO communication service is used. The resolution remains limited because each data request must pass through the available communications paths.
Recorder	The parameter values are written into an internal device buffer and then read. The trace buffer can store data at the highest possible resolution, however, it is limited by the size of the memory, so that the performance of continuous recording is not possible. The recorder has a trigger function.

Options:

- Source:

  One of the available configurations can be selected as the source. They contain the source selection and the appropriate configuration for the logger and recorder. The configuration can be viewed and changed via the **Edit settings** link.

- Mode:

  Selection of the type of recording: logger or recorder. If a trigger source has been defined for the recorder, the trigger mode can be selected during operation.

Motion Controllers with CAN interface must be in the Operational NMT state to allow running in logger mode.

- Data sources:

  Every assigned data source in the current source selection list contains its own area for deactivating and activating the scaling, and for setting it

- Switch automatic scaling on and off. For manual scaling the Y-axis has the values listed below for the axis minimum and maximum.

- Activate or deactivate synchronisation with other Y-axes. A change made to an axis setting is applied to the other axes.

- Centring curve. If automatic scaling is active, the current axis setting is entered in the input fields. Otherwise the axis minimum and axis maximum are shifted and the spacing remains unchanged.

Recording
After recording start, the activated data sources are shown in the **Graphical analysis** window as curves in different colours (see chap. 7.5, p. 60)

Recording is terminated in the following cases:

- Recording stopped by pressing the button.
- The control fails to respond.
- The PC cannot display the data quickly enough.
- The recording is ended in the "Recorder" recording type (single shot).
- A linear buffer is used and it is full (see chap. 7.2.7.1, p. 52).

7.2.7.1 Trace settings

The recording parameters and the data sources are configured in the **Settings** window. Several configurations are available and can be freely customised. The settings made are saved in the project file.

General

- Total buffer: The maximum number of parameter values to be recorded can be limited so as not to fill the working memory with too much trace data. A maximum of 60,000 parameter values is set by default. At a resolution of 1 ms this corresponds to a recording duration of 60 s.
- TxPDOs released for logger: Used to determine which TxPDOs can be used by the Motion Manager for the logger mode via the CAN interface.

Data sources

All mappable objects from the object dictionary in the control are available as data sources. The most important parameters are listed under the **Proposed sources** tab.

The ***Arrow*** button allows highlighted parameters from the list displayed on the right to be assigned to a source selection or the trigger source. The parameters can also be assigned via Drag & Drop. The ***Recycle bin*** button or the ***Delete*** key are used to remove highlighted data sources.

Logger

The sampling interval can be specified in the case of continuous data recording.

- Fastest possible sampling: New data is requested as soon as the previous query has been completed.
- Fixed sampling interval: Data is requested at the specified interval.

Depending on the used communication interface, it may not be possible to keep to the specified interval. In this case, fastest possible sampling is used.

Recorder

With the "Recorder" recording type, a specific amount of data is recorded with a fixed sampling interval. The configuration is performed to match the trace objects that are defined for the control (see the Communication Manual).

Further information on the trace functions can be found in the Communications Manual for the control.

7.2.8 Status display

The status display shows the status of various control parameters, which are interrogated cyclically. The error history of the controller is also shown here; the error history can be reset if required.

The status display can be opened by pressing the ***Status display*** button in the quick access toolbar or via the **Tools** menu.

7.2.9 Sequence programmes

Starting and stopping the sequence programme
A sequence programme saved in the controller can be started in the following ways:

In the toolbar by pressing the ***Load sequence programme from the EEPROM and start it*** button

Via the respective menu item under **Commands – sequence programme**

When a sequence programme is executed via the start button in the editor, it is first loaded into the control RAM, and is then executed there. The programme is not permanently saved on the EEPROM of the Motion Controller until the ***Save to EEPROM*** button is pressed. For execution of a sequence programme via the main menu or the main toolbar, the selected programme is loaded into the RAM from the control EEPROM first, and is then executed.

Stop sequence programme
A sequence programme that is running can be stopped in the following ways:

In the toolbar with the ***Stop running sequence programme*** button

Via the respective menu item under **Commands – sequence programme**

Edit sequence programme
An integrated development environment is available in the editor area for editing sequence programmes. Here sequence programmes can not only be created and edited but also downloaded, executed or debugged .

More information on developing sequence programmes using the Motion Manager and on the programming language can be found in the programming manual.

Read sequence programmes
Sequence programmes saved in the control can be read in the following ways and displayed in the editor area:

- Via the **Commands - Sequence programme – Read out and show all** menu
- By double-clicking on the **Upload** item in the Node Explorer under **Sequence programmes** of the active node

Under **Sequence programmes** the Node Explorer also manages links to programme files which were edited and downloaded within the loaded project. By double clicking on them, these files can be opened directly in the editor.

Save sequence programmes
The sequence programme is permanently saved on the EEPROM of the Motion Controller by pressing the ***Save to EEPROM*** button.

When a sequence programmes is saved to the EEPROM of the Motion Controller, the file name and time of downloading are saved with it. This information is also inserted into the project file. This allows the correctness of this information to be checked at upload; after successful check, the file suitable to the device programme is loaded and displayed.

Before a sequence program is transferred to the control, a pre-processing step is performed in the Motion Manager to prepare the program for the machine. Among other things, comments and formatting are not transferred.

Examples of sequence programmes can be found in the installation directory under **Motion Manager 6\Examples**.

7.2.10 Frequency response measurement

The frequency response of the drive can be measured as follows and presented in a Bode diagram:

✓ Motor selection is completed (see chap. 7.2.1, p. 42).

▸ Frequency response measurement using the **Tools – Frequency response measurement** menu.

↳ The motor is excited with a noise signal. This measures the frequency response of the controlled system at the open control loop. The level of the signal is matched to the properties of the motor. This ensures no damage can be done to the motor. Nevertheless excitation of the motor can give rise to increased noise generation and to heavy oscillation of the drive shaft.

If during the excitation process the motor is restrained by mechanical limitations, the measurement is invalid. In this case the start position of the drive must be corrected using the function provided, and the procedure repeated.

7.2.11 Saving and restoring user configuration

The configuration currently stored in the controller can be stored in an XDC file using the object browser (see chap. 7.3.3.1, p. 56).

The configuration stored here can be reloaded and transferred to the controller at any time. Only the following objects are transferred during this process:

- Writeable objects
- Objects that are not excluded from download

Not transferred are, for example:

- Node no. and baud rate
- Hall adaptation values for the integrated MCS products

7.2.11.1 Replacing the controller while the motor remains the same

▸ Transfer XDC file to the controller.

↳ The drive is fully configured.

The node number and the baud rate remain unchanged.

7.2.11.2 Transferring the configuration of a controller to another external controller

1. Transfer XDC file to external controller.
2. Perform a Hall sensor adaptation to adapt to the new motor.

↳ All motor and controller parameters are also reset. The existing configuration is thereby overwritten.

The node number and the baud rate remain unchanged and must be set separately if necessary.

7.2.11.3 Transferring the configuration of a controller to another integrated MCS controller

- Transfer XDC file to the integrated MCS controller.
  - The drive is fully configured.

    The node number and the baud rate remain unchanged and must be set separately if necessary.

7.3 CANopen standard functionality

The Motion Manager contains the standard functionality of CANopen acc. to CiA 301 and CiA 402. In principle all CANopen-capable devices can thus be displayed in the Node Explorer and operated in their basic functionality.

7.3.1 NMT

The CANopen network management functions are available in the **Commands** menu under **Network management**. In accordance with the transitions of the NMT state machine, here CANopen nodes can be started, stopped, switched to the *Pre-Operational* NMT state, or reset.

Commands from chap. 10.1, p. 72 can also be used in addition to the menu items.

7.3.2 Device Control

The CANopen device control functions for drive controls are available under **Device Control** in the **Commands** menu. In accordance with the transitions of the Device Control state machine in accordance with CiA 402, here the CANopen drives can be operated using the Controlword.

Commands from chap. 10.1, p. 72 can also be used in addition to the menu items.

7.3.3 SDO/PDO

The commands *SOBJ* and *GOBJ* in the **Commands – Object dictionary** menu are available for data exchange using a SDO (Service Data Object). These can be used for describing or reading any object in the object dictionary.

The use of this command is described in chap. 10.1.4, p. 76 and chap. 10.1.5, p. 76. Additional Motion Manager commands for the easy access to specific objects in the object dictionary are described in chap. 10.1.2, p. 74 and chap. 10.1.3, p. 75. The object browser described in chap. 7.3.3.1, p. 56 offers another convenient facility for accessing all the objects in the object dictionary.

Any PDOs (Process Data Objects) can be sent using the ***Send telegram directly*** dialogue, which is available in the **Commands** menu under **Send telegram directly**. In addition to this dialogue, the ***TRANSMIT*** command from chap. 10.1.6, p. 77 can also be used.

The mapping of PDOs can be viewed and changed via the dialogue described in chap. 7.3.3.2, p. 57.

7.3.3.1 Object browser

The object browser allows all entries of the object dictionary to be viewed and changed.

The object browser can be opened using the ***Object browser*** button in the quick access toolbar or via the **Configuration** menu.

For the object browser to be displayed, an EDS file matching the firmware version of the connected control is contained in XML format (XDD file) in the Motion Manager installation. If this is not the case, an appropriate file can be loaded and imported when the browser is started.

Register

- Communication: Communication objects acc. to CiA 301
- Manufacturer: Manufacturer-specific objects
- Device: Objects of the drive profile acc. to CiA402

Read-only parameters which cannot be changed, are identified with "–" in the **new value** column. All other values can be changed by double clicking on the respective line and inputting a new value.

Buttons

Button	Meaning	Description
	Update	The object dictionary is updated and all objects are read out again from the device.
	Transfer changes	The changed settings are loaded to the device.
	Save object dictionary	The current values of the object dictionary are permanently saved in the device.
	Load configuration	Loads an XDC configuration file that was previously saved, or a new XDD device description file.
	Save the current configuration as a file	The current parameter configuration is saved as an XDC file.
	Open Help	The context-sensitive Help content is called up.
	Filter objects	The output displays only those objects which match the search text that was input. For this purpose, the entries in columns "Actual value" and "New value" are disregarded. The following rules apply to the input of the search text: ▪ "+" can be used to combine several search expressions. ▪ Inserting "*" after the search expression selects all entries which start with the search expression. ▪ There is generally no differentiation made between upper and lower case characters.

EDS files of the standard firmware variants, for incorporation into PLCs and other CANopen tools, can be found in the installation directory under \Motion Manager 6\EDS.

Read device configuration

When the object browser is started or the ***Update*** button is clicked, the entire object dictionary together with its current values is read out from the control. The available objects are taken from the XDD file.

Change device configuration
Double-clicking an adjustable parameter line opens an input window. When confirmed, the value input here is applied in the **New value** line.

Clicking on the ***Load changes*** button loads all changed values to the control. To save the changes permanently on the control, press the ***Save object dictionary*** button.

Save device configuration
Clicking on the ***Save current configuration as a file*** button saves the last settings read out from the control to an XDC file (XML Device Configuration File).

Load device configuration
If a previously saved device configuration is available in the form of an XDC file, it can be loaded to the object browser using the ***Load configuration*** button.

Clicking on the ***Load changes*** button loads all changed values to the control. To save the changes permanently on the control, press the ***Save object dictionary*** button.

7.3.3.2 PDO mapping

The "PDO mapping" dialogue offers a convenient facility for configuring the mapping of the PDOs of a CANopen control. The dialogue can be opened via the **Configuration** menu.

PDOs are Process Data Objects in the CANopen protocol, with a maximum length of 8 bytes. In the dialogue there is a tab for mapping each of the receipt and transmit PDOs supported by the control.

- Receipt PDOs are received by the control (RxPDOs)
- Transmit PDOs are transmitted by the control (TxPDOs).

The data content of a PDOs can be composed of any of the parameters of the CANopen object dictionary which support the respective direction of mapping (right side of the window). The number of parameters in a PDO is dependent on their length, subject to the maximum PDO length of 8 bytes.

Buttons

Button	Function
	Assign parameters from the list (right window) of the highlighted PDOs in the structure tree (left window). The parameters can also be assigned via Drag & Drop.
	Remove highlighted PDOs from the structure tree (left window). Optionally, parameters can also be removed using the ***Del*** key.

Clicking on the ***Send*** button loads the displayed PDO mapping to the control. To save the settings permanently in the control, click the ***Save*** button as well.

7.3.4 LSS

Using the functions of the LSS (Layer Setting Services) protocol to CiA 305, changes to the node number and transfer rate of a FAULHABER CANopen node can be performed via the **Configuration – connection parameters** menu item.

The LSS protocol offers two options for the configuration:

Type of configuration	Description
Switch Mode Global	All connected LSS slaves are switched into configuration mode. The baud rate and node ID (node address) can be set by this method only if precisely one LSS slave is connected.
Switch Mode Selective	Precisely one LSS slave, whose vendor-ID, product code and serial number is known, is switched into configuration mode. This mode can be used to configure individual drives in the network selectively by their serial number.

There is a dedicated dialogue available for LSS mode. This enables either an individual node to be configured globally or a node whose LSS data (vendor-ID, product code and serial number) are then input to be configured selectively.

- If a previously configured node selected via Node Explorer is to be reconfigured (e.g. assigned another node number), its LSS data is displayed directly, so that the data must be only confirmed by clicking on ***Next***.
- If an unconfigured node (node ID = 0xFF) is to be configured in the network, the vendor ID and the product code are already pre-assigned and must be selected. The serial number of the desired node must still be input. If only one drive is connected, Switch Mode Global can be selected, and no further data have to be specified.

For Motion Controllers of the MC V3.x family, the following data are required:

- Vendor ID: 327
- Product code: 48
- Revision number: 1.0

For Motion Controllers of the MC V2.x family, the following data are required:

- Vendor ID: 327
- Product code: 3150

The dialogue provides the possibility of adapting all PDO COB-IDs and the EMCY COB-ID to the set node number. If the option is active, the COB-IDs are set according to the "Pre-defined connection set" as per CiA 301.

7.4 Motion Cockpit

The motion cockpit contains input and display elements which permit easy movement control. This is started via ***Motion Cockpit*** button in the quick access toolbar or via the **Tools** menu.

NOTICE!

When the movement commands are executed, the drive moves corresponding to the entered values. Obstacles within the movement range can lead to collisions and material damage.

- Make sure that when performing movement commands the drive is free to move within the values that were input.

Prepare drive:

1. Select the desired operating mode from the selection list.
2. Press the ***Activate*** button to activate the operating mode.
3. Check that the output stage is switched on and, if necessary, switch it on by pressing the button that is displayed.
   - The output stage must be switched on so that the drive can execute the movement commands that were issued.

Enter set values:

Once the control is in the selected operating mode, set values can be input via the input area.

Send set values alternately:

In this area of the motion cockpit, the set values previously input can be sent alternately. The waiting time between movement commands can be set via the respective selection list.

Buttons in the toolbar

Button	Function
	Switches off the output stage (emergency stop), alternatively with ***F5*** button.
	Stops the motor.
	Opens the **Graphical analysis** window for recording device parameters.
	Opens the **Status display** window for monitoring device sates.
	Opens the **Controller parameters** window for setting the controller parameters and profile parameters.

7.5 Graphical analysis

The graphical analysis of the Motion Manager (Trace) offers comprehensive possibilities for monitoring and evaluating the behaviour of the drive. One potential application is the assessment of the dynamic behaviour of the motor and control, or the optimisation of the controller parameters (recording step responses).

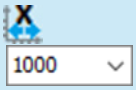
The graphical analysis is started by pressing the ***Graphical analysis*** button in the quick access toolbar or via the **Tools** menu

The settings options are controller-specific and are described in the relevant chapter of the respective controller family.

- MC V2.x: see chap. 7.1.5, p. 37
- MC V3.x: see chap. 7.2.7, p. 51

7.5.1 The main toolbar

The toolbar for the graphical analysis permits direct access to the selected functions:

Symbol	Function
	Start and stop the recording.
	Scaling of the X-axis. ▪ Activated: Automatic scaling of the X-axis, so that the entire period of the recording is displayed. ▪ Not activated (default): The scaling of the X-axis corresponds to the value of the input field in ms for the overall display range. ▪ Displayed time window in milliseconds by manual scaling.

	Activates / deactivates zoom mode. When zoom mode is activated, the following mouse functions are available:
	▪ The left mouse button as well as the mouse wheel can be used for enlarging/reducing the selected window section:
	  ▪ To enlarge a window section, place the cursor cross at the top of the recording window, and with the left mouse button pressed down drag the cross downwards.
	  ▪ To reduce a window section, place the cursor cross at the bottom of the recording window, and with the left mouse button pressed down drag the cross upwards.
	▪ With the right mouse button pressed down, the window section can be moved.

	Activates the calculation mode which permits the distances between two data points to be calculated in the X and Y directions. ▪ If more than one parameter was recorded, select a curve, since the calculation of distances must be performed for a specific Y-axis. Right click to undo the selection again. ▪ Use the left mouse button to select a point on the curve or any point in the area. ▪ Select a further point. The selected points and the distances in the X and Y directions will be displayed. ▪ To recalculate, click on any point.
	Opens the trace settings window.
	Opens a dialogue for printing the displayed graphics.
	The currently displayed recording can optionally be saved as a bitmap file or CSV file in text form. A CSV file can, for instance, be read into a mathematical or table calculation programme for further processing.

7.6 Macro function

The macro function of the Motion Manager offers the facility to specify a freely definable sequence of commands supported by the control, which can be sent to the control with a single mouse click.

The function can be called up via **Tools – Macros** or the quick access toolbar.

There is a set of macros available for each Motion Controller family which can be extended and changed as required. The macros thus created can be assigned to buttons; pressing the button then executes the macro.

 Add button: Adds an unassigned button

 Delete button: Deletes the selected button

 Editing macros: Open the editor for editing macros

The pull-down menu or the context menu for a button can be used to select a macro and assign it to the button.

7.6.1 Edit macros

The macro editor allows the pre-defined macros to be extended and changed.

The definition of macros must comply with the following rules:

- Each macro consists of a command block, headed by the macro name in square brackets.
- The available commands can be found in the Motion Manager command reference or the command reference (see chap. 10.1, p. 72) for the control that is used.
- With the key word WAIT a delay in ms can be inserted between two commands (e.g. WAIT(1000) creates a delay of 1000 ms).
- Comments are introduced with ";".

With the ***Apply*** button the changes are applied and saved into the macro file.

During execution of the macro the individual command are loaded line by line.

The pre-defined macro file can be regenerated by fully deleting the content introduced by the macro editor, and then applying the changes.

8 Additional functions

8.1 VB Script programmes

The FAULHABER Motion Manager offers the facility to create simple automation scripts and to allow them to run on the PC within the Motion Manager. The VB Script code that is input is then executed via the Scripting Host incorporated in Microsoft Windows.

The time behaviour of the Microsoft Scripting Host is not deterministic. Therefore script programmes are intended only for commissioning tests.

For productive operation, suitable programming tools are to be used.

8.1.1 Create script programme

Script programmes can be created in the Editor window and loaded from there. They have the ending "*.vbs".

Script programmes always begin with „SUB MAIN“ and end with „END SUB“. The VB Script code can be written within these delimiters. As an aid, the Motion Manager offers a selection of code examples which can be used to create Script programmes.

An explanation of the VB Script command set can be found in the Microsoft Help for VB Script (Help - Help for VB Script).

Examples of VB Script programmes can be found in the installation directory under \Motion Manager 6\Examples

8.1.2 Starting and stopping script programme

When a VB Script file is opened in the editor, the following control elements are displayed in the toolbar of the file tab:

Symbol	Function
	Execute script programme: The script programme is executed. Whilst the script programme is running, the background of the editor area is displayed in grey. During this period the programme code cannot be changed. If an error occurs, the script programme will stop and an error message is displayed. Furthermore, the line at which the script programme stopped will be highlighted in red. After the execution has been completed, edit mode is reactivated in the editor area.
	Exit programme: Ends a script programme that is running and reactivates edit mode in the editor area.
	Extras: Displays and hides a list of code templates. The code templates contain typical programme constructs, which can be incorporated into the script programme by means of drag & drop. Further code templates can be defined by highlighting an area of code within the script programme, and incorporating it into the list of code templates via drag & drop.

A script programme runs on the PC within the Motion Manager and cannot be loaded to the control. Only commands and data are exchanged between a running script and the control.

8.1.3 General Motion Manager functions

The VB script command set has been extended by adding special Motion Manager functions, in order to exchange data with the Motion Controllers. These functions must always be called up using the object identifier "MC.".

Motion Manager functions can quickly be accessed using automatic integration (key combination ***Ctrl*+*space***).

8.1.3.1 SendCommand

Syntax:

```
MC.SendCommand(command)
```

Description:

Sends an ASCII command to the Motion Controller. The Motion Manager commands listed in chap. 10.1, p. 72 are available.

If Motion Manager commands that read or write objects in the object dictionary are used, subsequently MC.WaitAnswer() must be called up to read the answer.

For Motion Controllers of the family MC V2.x with an RS interface, the specified commands are sent directly to the drive control. These commands can also be used for Motion Controllers of the family MC V2.x with a CF interface, in which case they are sent via a CAN telegram.

Parameter:

`command(string)`: Command to be sent to the Motion Controller.

Example:

```
MC.SendCommand("V100")
```

8.1.3.2 WaitAnswer

Syntax:

```
answer = MC.WaitAnswer(timeout, answMode)
```

Description:

Waits for the specified time (milliseconds) for an answer from the Motion Controller.

Parameter:

```
answer (string): Antwort, die vom Motion Controller eingelesen wurde
  answer = "": Keine Antwort nach timeout
timeout (int): Ganzzahliger Wert in ms, bis Antwort eintreffen muss
answMode (int): Ganzzahliger Wert für die Interpretation der eingelesenen Daten
  answMode = 0: Antwort-Daten werden als Integer interpretiert
  answMode = 1: Antwort-Daten werden als String interpretiert
  answMode > 255: Es wird auf eine Nachricht mit der hier angegebenen COB-ID
  bzw. Knotennummer (LowWord) mit Befehls-ID (HighWord) gewartet
```

Additional answMode values are available for Motion Controllers of the family MC V2.x with an RS interface:

```
  answMode = 10: Asynchrone Nachrichten und Quittierungen werden ignoriert
  answMode = 11: Quittierungen werden ignoriert
  answMode = 12: Asynchrone Nachrichten werden ignoriert
  answMode = 13: Asynchrone Nachrichten bis auf v, e und h werden ignoriert
```

Example:

```
a = MC.WaitAnswer(1000,0)
```

8.1.3.3 WriteToHistory

Syntax:

```
MC.WriteToHistory(text)
```

Description:

Writes the specified text line into the log window of the Motion Manager.

Parameter:

`text (String)`: Text to be output in the log window.

Example:

```
MC.WriteToHistory("Position 1 reached")
```

8.1.3.4 CloseCom

Syntax:

```
MC.CloseCom
```

Description:

Closes the currently active communication interface.

8.1.3.5 OpenCom

Syntax:

```
ret = MC.OpenCom
```

Description:

Opens the selected communications interface.

Parameter:

`ret (int)`: Integer return value.

`ret = 1`: Interface successfully opened.

`ret = 0`: Error when opening the interface.

Example:

```
ret = MC.OpenCom
IF ret = 1 THEN
    MsgBox(„Interface successful opened!“)
ELSEIF ret = 0 THEN
    MsgBox(„Error opening Interface!“)
END IF
```

8.1.3.6 CmdExecute

Syntax:

```
MC.CmdExecute(command)
```

Description:

Executes the specified command at the system level. This function can be used for example to start an external programme.

Parameter:

Command (string): string which specifies the command or the file name of the application to be executed, with file path if required.

Example:

```
MC.CmdExecute("C:\tools\mytool.exe")
```

8.1.3.7 UpdateWindows

Syntax:

```
MC.UpdateWindows
```

Description:

If no other MC functions are available, this function can be called up within a loop in order to allow other parts of the application time for updating.

8.1.4 Functions for Motion Controllers with object dictionary

8.1.4.1 GetObj

Syntax:

```
answer = MC.GetObj(nodeNr, index, subIndex, dataType)
```

Description:

Reads a parameter from the object dictionary.

Parameter:

`answer (string)`: Parameter value read
- `answer = ""`: Parameter value could not be read

`nodeNr (int)`: Node number

`index (int)`: Index of the object entry

`subIndex (int)`: Subindex of the object entry

`dataType (int)`: Data type of the parameter to be read
- `dataType = 0`: Integer
- `dataType = 1`: String

Example:

```
value = MC.GetObj(1, &h6067, &h00, 0)
```

8.1.4.2 SetObj

Syntax:

```
ret = MC.SetObj(nodeNr, index, subIndex, value, len)
```

Description:

Writes a new numeric value to an integer parameter in the object dictionary.

Parameter:

`ret (int)`: Integer return value.
- `ret = 1`: Function successfully executed.
- `ret = 0`: Error when executing the function.

`ret (int)`: Integer return value

`nodeNr (int)`: Node number

`index (int)`: Index of the object entry

`subIndex (int)`: Subindex of the object entry

`value (int)`: New value of the parameter

`len (int)`: Data length of the parameter in bytes

Example:

```
ret = MC.SetObj(1, &h6067, &h00, 30, 4)
IF ret = 0 THEN
    MsgBox("Error writing Parameter")
END IF
```

8.1.4.3 SetStrObj

Syntax:

```
ret = MC.SetStrObj(nodeNr, index, subIndex, value)
```

Description:

Writes a new string value to a string parameter in the object dictionary.

Parameter:

`ret (int)`: Integer return value.

`ret = 1`: Function successfully executed.

`ret = 0`: Error when executing the function.

`nodeNr (int)`: Node number

`index (int)`: Index of the object entry

`subIndex (int)`: Subindex of the object entry

`value (int)`: New value of the parameter

8.1.5 Functions for Motion Controllers of the MC V2.x family with RS interface

8.1.5.1 ComXonXoff

Syntax:

```
MC.ComXonXoff(on)
```

Description:

Activates or deactivates the Xon/Xoff protocol for Motion Controllers with a serial RS232 interface and ASCII protocol (MC V2.x). The activation is necessary when large amounts of data are to be sent in succession without waiting for an answer.

Parameter:

`on (int)`: Integer input value which activates or deactivates the Xon/Xoff protocol.

`on = 1`: Activate the Xon/Xoff protocol.

`on = 0`: Deactivate the Xon/Xoff protocol.

Example:

```
MC.ComXonXoff(1)
```

8.1.5.2 SendBin

Syntax:

```
MC.SendBin(value)
```

Description:

Sends a binary value to the control.

Parameter:

`value (int)`: 8-bit value to be sent

Binary values are used only for setting the system parameters and are generally not accessible to the user.

8.1.5.3 SetBinMode

Syntax:

```
MC.SetBinMode(mode1, mode2)
```

Description:

Sets the binary mode for parameters 1 and 2 via

- the binary interface on Motion Controllers of the MC V2.x family with RS interface,
- the trace channel on Motion Controllers of the MC V2.x family with CF interface.

Then up to two variables can be read in simultaneously via the BinRequest() function.

Parameter:

`mode1 (int)`: Integer value for the first parameter to be read.

`mode2 (int)`: Integer value for the second parameter to be read.

Available values for mode1 and mode2 can be found in the documentation for the respective control.

8.1.5.4 BinRequest

Syntax:

```
ret = MC.BinRequest(timeout)
```

Description:

Data request. Reads the parameters set by SetBinMode() at the current point of time. The results can then be read via the following properties:

- `MC.BinVal1`
- `MC.BinVal2`
- `MC.BinTimecode`

Parameter:

`ret (int)`: Integer return value.

`ret = 1`: Data received.

`ret = 0`: No data received within the timeout time.

`timeout (int)`: Integer value in ms, until the answer must be received.

Example:

```
CALL MC.SetBinMode(0,1) 'Ist- und Solldrehzahl einlesen
IF MC.BinRequest(500) THEN 'Datenanforderung
    Istdrehzahl = MC.BinVal1
    Solldrehzahl = MC.BinVal2
    Timecode = MC.BinTimecode
END IF
```

8.2 Firmware update

The integrated firmware update functions permit checking and updating the firmware on the connected FAULHABER control. After start-up via **Extras – Firmware Update** there are 2 functions available for selection:

Function	Description
Check for update	A check is made whether there is an update available for the current device firmware. If there is an update available it can be used to update the device firmware.
Load firmware file	A firmware file supplied separately by FAULHABER can be loaded and transferred to the control.

A firmware update can be performed only via the interface that is supported by the connected control as the update interface.

8.3 Virtual devices

Virtual devices can be managed and activated for offline mode. This allows the controller-specific functions of the Motion Manager to be performed, even if no matching hardware is available.

Communication with a virtual device is depicted realistically in the terminal (see chap. 5.5, p. 22). Support of all services of a communication protocol is, however, not guaranteed for all virtual devices.

Virtual devices have a limited range of functions and are primarily intended for depicting a set configuration. The configuration of a virtual device must NOT be transferred to real devices.

8.3.1 Establishing connection to virtual controller

1. Open the list of available virtual controllers via the **Extras – Virtual devices...** menu.
2. Activate the desired controller using the checkbox.
   - A connection can be established to active devices only.
3. Confirm the selection using the ***OK*** button.
4. Establish connection with the controller (see chap. 6.1, p. 28).

Virtual devices always use the port with number 0 (COM0, USB0, CAN0) as communication interface.

Activated virtual devices can be selected in the wizard to establish a connection. For Speed Controllers, a selection of devices is also available under **Drive functions** for offline mode.

8.3.2 Managing virtual Motion Controllers

The following functions in the **Extras – Virtual devices...** menu can be used to manage virtual Motion Controllers:

Button	Function
New	Opens a window for selecting the device family, the device name and the node number for a new virtual Motion Controller. ▪ FAULHABER Motion Controllers which can be depicted as virtual devices are available for selection. ▪ The node number must be unique so that communication is not disrupted if multiple devices are connected.
Import	Opens a window for selecting a configuration file (*.xdc) used to create a new virtual Motion Controller.
Edit	Opens a window for editing the selected virtual Motion Controller. Predefined devices cannot be edited.
Delete	Deletes the selected virtual Motion Controller. Predefined devices cannot be deleted.

9 Warranty

Products of the company Dr. Fritz Faulhaber GmbH & Co. KG are produced using the most modern production methods and are subject to strict quality inspections. All sales and deliveries are performed exclusively on the basis of our General Conditions of Sale and Delivery which can be viewed on the FAULHABER home page www.faulhaber.com/gtc and downloaded from it.

10 Appendix

10.1 Motion Manager command reference

Commands for the following tasks can be used in the Motion Manager:

- Direct input into the command input field of the terminal window.
- Use within macros
- Use within VB Script programmes by means of the `MC.SendCommand()` function

The commands listed in this chapter should only be used within the Motion Manager environment. The commands are interpreted by the Motion Manager and are converted into the protocol of the respective interface. The commands cannot be used within sequence programmes, or when the drive is being actuated by other host systems.

Any node in the network can be addressed, using the pre-assigned decimal node number. If no node number is specified, the node that is active in the Motion Manager is addressed. Upper and lower case are not distinguished.

10.1.1 Control the CANopen state machine

Tab. 3: Commands to the control of the CANopen NMT state machine

Command	Function	CAN telegram
START	Start Remote Node (Motion Controller is switched into the *Operational* state)	Id 0x000: 0x01 Node ID
STOP	Stop Remote Node (Motion Controller is switched into the *Stopped* state)	Id 0x000: 0x02 Node ID
PREOP	Enter Pre-Operational State (Motion Controller is switched into the *Pre-Operational* state)	Id 0x000: 0x80 Node ID
RESET	Reset Node	Id 0x000: 0x81 Node ID
RESETCOM	Reset Communication	Id 0x000: 0x82 Node ID
STARTALL	Start all Remote Nodes (Motion Controllers are switched into the *Operational* state)	Id 0x000: 0x02 0x00

Tab. 4: Commands for controlling the CiA 402 Controlword

Command	Function	CiA 402 Controlword (0x6040)
SHUTDOWN	Shutdown (Motion Controller is switched into the *Ready to Switch On* state)	0x0006
SWITCHON	Switch On (Motion Controller is switched into the *Switched On* state)	0x0007
DISABLE	Disable Voltage (Motion Controller is switched into the *Switch On Disabled* state)	0x0000
QUICKSTOP	Quick Stop (Motion Controller is switched into the *Quick Stop Active* state)	0x0002
DIOP	Disable Operation (Motion Controller is switched into the *Switched On* state)	0x0007
ENOP	Enable Operation (Motion Controller is switched into the *Operation Enabled* state)	0x000F
FAULTRESET	Fault Reset (Motion Controller is switched into the *Switch On Disabled* state)	0x0080
MA	Move Absolute (PP)	0x003F
MR	Move Relative (PP)	0x007F
HS	Homing Start (HM)	0x001F

Example: Set node 10 to the *Operational* state:

- 10 START
  - ↳ Node 10 is set to the *Operational* state.

Example: Switch-on sequence, to switch the active drive in the Motion Manager into the *Operation Enabled* state:

- SHUTDOWN
- SWITCHON
- ENOP
  - ↳ The drive is now in the *Operation Enabled* state

In PP operating mode a new position value is applied only on the rising flank of bit 4 in the Controlword. For this reason, before performing the MA or MR commands, it must be ensured that this bit is reset again e.g. by means of the ENOP command.

10.1.2 Describing objects in the object dictionary

Commands with an argument expect a decimal numeric value subsequent to a command. After successful execution, the value **OK** is returned in the terminal window or in the script function MC.WaitAnswer(). If execution is unsuccessful, an SDO error message is returned in plain text, or a timeout error (empty string) is returned.

Tab. 5: Commands for writing objects in the object dictionary

Command	Function	Object
Commands with argument:		
OPMOD	Configure operating mode	0x6060.00 (Modes of Operation)
SPOS	Specify actual position (PP)	0x607A.00 (Target Position)
V	Specify target speed (PV)	0x60FF.00 (Target Velocity)
HM	Set Homing Mode (HM)	0x6098.00 (Homing Mode)
U	Specify output voltage (Voltage Mode)	Device-dependent
SP	Set maximum speed	0x6081.00 (Profile Velocity)
AC	Configure the acceleration ramp	0x6083.00 (Profile Acceleration)
DEC	Configure brake ramp	0x6084.00 (Profile Deceleration)
Commands without argument:		
SAVE	Save all parameters	0x1010.01 (Store all Parameters)
SAVE_ALL	Save all parameters	0x1010.01 (Store all Parameters)
SAVE_COM	Save the communication parameters	0x1010.02 (Store all Communication Parameters)
SAVE_APP	Save the application parameters	0x1010.03 (Store all Application Parameters)
RESTORE	Load all factory parameters	0x1011.01 (Restore all Parameters)
RESTORE_ALL	Load all factory parameters	0x1011.01 (Restore all Parameters)
RESTORE_COM	Load factory communication parameters	0x1011.02 (Restore Communication Parameters)
RESTORE_APP	Load factory application parameters	0x1011.03 (Restore Application Parameters)

Example: Set Profile Position Mode (PP) and move the drive relatively by 10,000 increments (or the set positioning unit).

- Send the following command sequence:
  a) **OPMOD 1**
  b) **SPOS 10000**
  c) **MR**

↳ The drive has been moved in Profile Position Mode by 10,000 increments (or the set positioning unit).

10.1.3 Reading objects in the object dictionary

After successful execution, the answer is returned as a string in the terminal window or in the script function MC.WaitAnswer() (numeric values in decimal form). If execution is unsuccessful, an SDO error message is returned in plain text, or a timeout error (empty string) is output.

Tab. 6: Commands for reading objects in the object dictionary

Command	Function	Object
GTYP	Read device name	0x1008.00 (Manufacturer Device Name)
VER	Read software version	0x100A.00 (Manufacturer Software Version)
GSER	Read serial number	0x1018.04 (Serial Number)
GSW	Read Statusword	0x6041.00 (Statusword)
GOPMOD	Read set operating mode	0x6061.00 (Modes of Operation Display)
POS	Read actual position	0x6064.00 (Position Actual Value)
POSI	Read actual position (internal units)	0x6063.00 (Position Actual Value)
TPOS	Read target position	0x6062.00 (Position Demand Value)
GV	Read target speed	0x606B.00 (Velocity Demand Value)
GN	Read actual speed	0x606C.00 (Velocity Actual Value)
GRC	Read actual current consumption	0x6078.00 (Current Actual Value)
GSP	Read maximum speed	0x6081.00 (Profile Velocity)
GAC	Read acceleration ramp	0x6083.00 (Profile Acceleration)
GDEC	Read brake ramp	0x6084.00 (Profile Deceleration)

Example: Read actual position.

- Send the following command:

  a) **POS**

↳ The actual value of the actual position was read.

10.1.4 Write any object in the object dictionary

Any objects in the object dictionary can be addressed by specifying the index (xxxx) and subindex (yy) in hexadecimal form. The value to be entered must then be specified as a hexadecimal argument. The length of the hexadecimal argument (number of data bytes) must correspond to that required for the type of object to be written. The lowest-value byte is placed at the far right.

- Int8 = 1 byte
- Int16 = 2 bytes
- Int32 = 4 bytes

A byte in the argument is represented with two hexadecimal characters (00...FF).

After successful execution, the value **OK** is returned in the terminal window or in the script function MC.WaitAnswer(). If execution is unsuccessful, an SDO error message is returned in plain text, or a timeout error (empty string) is output.

Tab. 7: Command for writing any object in the object dictionary

Command	Function
SOBJ xxxx.yy	Writing an object (Set Object)

Example: Change the acceleration value of node 10 to the value 500 (Profile Acceleration 0x6083.00, Int32).

▶ Send the following command sequence:

a) **10 SOBJ 6083.00 000001F4**

↳ The acceleration value was changed to 500.

The command SLOBJ is available for writing string objects. For this purpose the ASCII characters of the string must be set as arguments, reading left to right.

10.1.5 Reading any object in the object dictionary

Any objects in the object dictionary can be addressed by specifying the index (xxxx) and subindex (yy) in hexadecimal form. After successful execution, the answer is returned as a string in the terminal window or in the script function MC.WaitAnswer(). If execution is unsuccessful, an SDO error message is returned in plain text, or a timeout error (empty string) is output.

Tab. 8: Command for reading any object in the object dictionary

Command	Function
GOBJ xxxx.yy	Read object (Get Object)

Example: Read the acceleration value of node 10.

▶ Send the following command:

a) **10 GOBJ 6083.00**

↳ The acceleration value of node 10 was output.

In the return data, object entries with a data length of up to 4 bytes are generally interpreted as integer values.

The command **GLOBJ** is available for reading string objects.

10.1.6 Sending any telegrams

By specifying the COB-ID for CAN communication or the node number for RS232/USB communication (xxx) in three-digit hexadecimal form, any telegrams of the basic underlying protocol can be sent. The telegram data bytes to be sent must also be specified as hexadecimal values, located in the telegram position (lowest-value byte on the left).

Tab. 9: Commands for reading any object in the object dictionary

Command	Function
TRANSMIT xxx	Transfer telegram

Example: Send the Controlword with the value 0x007F via RxPDO1 from node 10 (COB-ID = 0x20A).

- Send the following command:

  a) **TRANSMIT 20A 7F00**

With RS232/USB communication, the data to be sent must be specified without the SOF, EOF, length and CRC bytes.

10.2 Problem solution

10.2.1 Port not present

The Motion Manager supports the interfaces and interface plug-ins listed in chap. 5.4.2, p. 19. Instructions for the individual interfaces are listed there as well.

10.2.2 Port cannot be opened

If the port cannot be opened, it is being used by another application.

- Check the open application and close it, if access to the desired port is available.
  - In some cases the port is not released by the application, even when it has been closed.
- If the port was not released, reboot the PC.

10.2.3 No connection to the connected device

The easiest way to establish a connection is to use the "Wizard for establishing a connection".

- If no connection can be established using the wizard, check the following causes:

Possible cause	Remedy
The device is not connected to the interface selected in the Motion Manager.	▪ Select the correct interface or the correct port.
The device interface is not correctly connected to the PC.	▪ Check the installation of the electrical connections as specified in the device manual, or make those connections.
The device is not being supplied with sufficient voltage.	▪ Check the voltage supply.
The device is located in a network that is not correctly configured.	▪ Check the network configuration (see chap. 5.4.3, p. 21).
It is not possible to establish a connection to a single device if this device does not react to broadcast commands.	▪ Select the connection type "Network" and specify the appropriate transfer rate and the scan range.

10.2.4 Motor does not start

1. Using the wizard for motor selection, configure the control to the connected motor.
2. If the motor does not start afterwards, check the, following possible causes:

Possible cause	Remedy
The Motor is not correctly connected.	▪ Check the connection of the motor according to the product description of the motors and the Device manual.
The motor is not being supplied with power.	▪ Check whether the control power supply is separate from that of the motor.
The supply voltage is too low.	▪ Check the voltage range of the control.
The output stage is not switched on.	▪ Switch on output stage (chap. 5.2, p. 17).
The control is in fault state.	▪ Execute the menu command ***Commands – Device Control – Fault Reset***.
The control is in the wrong operating mode.	▪ Set the operating mode via the **Configuration – Drive functions** menu.

11 Licences

11.1 FAULHABER licence agreement

End User Licence Agreement for Software of
Dr. Fritz Faulhaber GmbH & Co. KG

between

(1) **Dr. Fritz Faulhaber GmbH & Co. KG**, Faulhaberstraße 1, 71101 Schönaich
- hereinafter "FAULHABER" -

and

(2) You as the user
- hereinafter "Licensee" -

The parties (1) and (2) hereinafter referred to individually as "**Party**" and together as "**Parties**".

PRELIMINARY REMARKS

(A) Faulhaber designs drive systems and produces them. In addition, FAULHABER has developed various software products. For example, the "FAULHABER Motion Manager" (hereinafter "**Motion Manager**") supports the commissioning and configuration of FAULHABER drive systems. Details - if available - can be found in the respective software manual. Unless otherwise expressly agreed, the software product is made available to the Licensee free of charge as a supplement to other hardware and software products offered by FAULHABER.

(B) The Licensee intends to use one or more software product(s) in its business. FAULHABER is willing to grant the Licensee rights of use to the software product(s) of Dr. Fritz Faulhaber GmbH & Co. KG under the terms of this End User License Agreement (hereinafter "**Contract**"). The details of this result from **§ 2**.

This having been explained, the Parties agree the following:

§ 1
Subject of the Contract

(1) The subject of this Contract is the transfer of one or more of the software products listed in subsection (2) (hereinafter "**License Subject**") and the granting of the rights of use described in **§ 2** by FAULHABER to the Licensee.

(2) The provisions of this Agreement apply to the following categories of License Subjects, including related manuals, if available:

a) Motion Manager with associated user documentation;

b) b) Programming Libraries;

c) Firmware;

d) Sequential Programs.

(3) The following services in particular are not the subject of this Contract:

a) Installation or other setup of the License Subject at the Licensee;

b) any individual settings of variable parameters of the License Subject according to the requirements of the Licensee (customizing);

c) individual program extensions for the Licensee (individual modifications);

d) Adaptations of interfaces to the needs of the Licensee;

e) Instruction and training of the Licensee's program users;

f) Maintenance of the License Subject, in particular delivery of new, future versions.

§ 2
Content and Scope of the Rights of Use

(1) FAULHABER or its licensors are exclusively entitled to all rights to the License Subject. The Licensee is only entitled to the rights to the License Subject as agreed in this Contract.

(2) The Parties agree that the License Subject and the associated documents, including future versions, are protected by copyright and constitute confidential information and trade secrets of FAULHABER in accordance with **§ 6**.

(3) Unless otherwise provided in subsection (4), FAULHABER grants the Licensee the following rights of use to the License Subject:

a) FAULHABER grants the Licensee the non-exclusive right, unlimited in time and territory, to use the License Subject for its own purposes and for the purposes of Licensee's customer in accordance with the following subsections.

b) This right includes the installation of the License Subject as well as the loading, displaying and running of the installed License Subject as well as the saving of the License Subject in the memory of the hardware on which the License Subject is installed. In particular, the Licensee is not entitled to edit or otherwise modify the License Subject, unless this is expressly permitted in this Contract.

c) Duplications of the License Subject shall only be permitted to the extent necessary for the contractual use. The Licensee may make backup copies of the License Subject in accordance with the rules of technology to the extent necessary and in unchanged form, and in particular also as part of its normal backup of the system environment.

d) The Licensee shall be entitled to transfer the License Subject if (i) the Licensee transfers the License Subject together with original hardware components from FAULHABER, (ii) the transfer of the License Subject is free of charge for the third party, (iii) the Licensee ensures that the third party is not granted any further rights to the License Subject than the Licensee is entitled to under this Contract and (iv) the third party is imposed at least the obligations of this Contract with regard to the License Subject. Third parties shall also include companies which are affiliated with the Licensee within the meaning of § 15 of the German Stock Corporation Act (AktG).

e) FAULHABER is entitled to update the License Subject without prior notice, e.g. in order to correct errors or to improve or extend functions. If the updated version replaces the previous License Subject, it shall also be subject to the provisions of this Contract.

f) The Licensee may only use the License Subject within the scope of the intended use and only for productive operation if the License Subject is qualified for the specific application. "**Productive operation**" means the control of the respective drive system manufactured by FAULHABER by the License Subject in the ongoing operation of the application in the specific area of use of the Licensee, alone or in combination with other components of an overall system. Qualification for the specific application presupposes in particular that appropriate tests in the productive environment have been carried out successfully and that existing legal requirements and requirements for the specific application are met in full by the Licensee during use (e.g. international standards and norms). This applies in particular to use for medical and military purposes and in safety-critical areas (e.g. in the aerospace sector and for controlling nuclear facilities).

g) The Licensee shall have no claim against FAULHABER for release of the source code or the source code documentation. Notwithstanding the foregoing, the source code of the License Subject shall be part of the grant of use, provided that this is expressly stipulated in this Contract (in particular in subsection (4) below).

h) Insofar as the License Subject provided to the Licensee by FAULHABER contains open source software or software for which FAULHABER only has a derived right of use (hereinafter "**Third Party Software**"), the usage regulations to which this Third Party Software is subject shall apply additionally and with priority. The Third Party Software used within a License Subject, the license condition(s) applicable to the Third Party Software as well as any existing copyright notices are each named in the associated manual or are made available to the Licensee for downloading in a separate file with the delivery of the License Subject. The Licensee is obliged to comply with the respective license conditions. In the event of a breach of these license conditions by the Licensee, the licensors as well as FAULHABER shall be entitled to assert the resulting claims and rights in their own name.

(4) The rights of use pursuant to subsection (3) shall be supplemented or modified as follows for the License Subjects listed below:

a) Motion Manager

aa) The intended use of the Motion Manager results - if available - from the respective current version of the associated manual, which is made available on the FAULHABER website.

bb) The Motion Manager may only be used if the Licensee ensures that no injury or damage to health and no risk of material damage to property (e.g. equipment) is possible when it is used.

cc) The Licensee may not use the Motion Manager in productive operation. By way of clarification, the Parties state that such use does not constitute an intended use of the Motion Manager. The same applies to the use for controlling drive systems which were not manufactured by FAULHABER, as well as the use for controlling drive systems which were manufactured by FAULHABER but which are not listed in the program description. Deviating from this, Sequential Programs contained in the Motion Manager may be adapted for and used in productive operation, provided that they have been qualified in the application in accordance with subsection (3) f).

dd) Decompiling and other types of reverse engineering are generally not permitted. This does not apply to the Licensee's right to observe, examine or test the functioning of the Motion Manager in order to determine the ideas and principles underlying a program element if this is done by actions to load, display, run, transmit or store the program, which the Licensee is entitled to do in accordance with this Contract (§ 69d (3) of the German Copyright Act (UrhG)). In addition, notwithstanding sentence 1, the Licensee shall be entitled to decompile for the purpose of producing an interoperable program exclusively under the conditions of § 69e (1) and within the limits of § 69e (2) UrhG. The foregoing rights shall only exist if the Licensee has requested the information it requires from FAULHABER prior to any such action and has not received the required information within a reasonable period of time. As part of its request, the Licensee shall provide FAULHABER with all information necessary to evaluate the request.

ee) Any further use of the Motion Manager, in particular the granting of sublicenses, requires the prior express written consent of FAULHABER. This shall not apply in the event of a sale of the drive systems, insofar as their proper use requires the use of the Motion Manager.

ff) The Motion Manager may only be used in connection with original hardware components from FAULHABER. Use for third-party hardware is prohibited.

b) Programming Libraries

aa) The Licensee is granted the right to edit source code files of the Programming Libraries and to transfer these edits to third parties. However, the editing of object code files of the Programming Libraries is prohibited.

bb) The use of the Programming Libraries is only permitted in connection with original hardware components from FAULHABER. Use for third-party hardware is prohibited.

cc) FAULHABER shall make manuals for the Programming Libraries available to the Licensee for downloading on FAULHABER's website as required and at its own discretion. The Licensee shall have no claim to the provision of a manual. Insofar as an intended use of the Programming Libraries is specified, this shall result from the respective current version of the associated manual - if available.

dd) The Programming Libraries may only be used if the Licensee ensures that no injury or damage to health and no risk of material damage to property (e.g. equipment) is possible when they are used.

c) Firmware

aa) The intended use of the Firmware results from the respective current version of the associated manual, which is made available on FAULHABER's website.

bb) The Firmware may only be used if the Licensee ensures that no injury or damage to health and no risk of material damage to property (e.g. equipment) is possible when it is used.

cc) The use of the Firmware is only permitted in connection with original hardware components from FAULHABER. The right to use the Firmware for hardware of a third party shall only exist after prior written consent by FAULHABER.

dd) Decompiling and other types of reverse engineering are generally not permitted. This does not apply to the Licensee's right to observe, examine or test the functioning of the Firmware in order to determine the ideas and principles underlying a program element if this is done by actions to load, display, run, transmit or store the program, which the Licensee is entitled to do in accordance with this Contract (§ 69d (3) of the German Copyright Act (UrhG)). In addition, notwithstanding sentence 1, the Licensee shall be entitled to decompile for the purpose of producing an interoperable program exclusively under the conditions of § 69e (1) and within the limits of § 69e (2) UrhG. The foregoing rights shall only exist if the Licensee has requested the information it requires from FAULHABER prior to any such action and has not received the required information within a reasonable period of time. As part of its request, the Licensee shall provide FAULHABER with all information necessary to evaluate the request.

d) Sequential Programs

aa) Sequential Programs are programs that can be executed on specific FAULHABER controller hardware.

bb) The Licensee is granted the right to edit Sequential Programs and to transfer these edits to third parties, provided that they have been delivered in source code form.

cc) The use of Sequential Programs is only permitted in connection with original hardware components from FAULHABER. The right to use Sequential Programs for hardware of a third party shall only exist after prior written consent by FAULHABER.

§ 3
Delivery

(1) The License Subject shall be delivered in the form existing at the time of delivery ("as is").

(2) The delivery of the License Subject shall be made in digital form by providing it for download on FAULHABER's website or individually by e-mail. FAULHABER is not obliged to provide the License Subject on physical data carriers.

(3) Prior to provision, FAULHABER shall check the License Subject for any malware using a virus scanner that is up-to-date at the time of the respective provision. FAULHABER shall have no further obligations with regard to freedom from malware.

§ 4
Obligations of the Licensee

(1) The Licensee is obliged to ensure a sufficient technical operating and system environment and the proper operation of the License Subject. The establishment of the operating and system environment for the License Subject shall be the sole responsibility of the Licensee.

(2) If productive operation of the License Subject is permitted under this Agreement, the Licensee shall ensure that the requirements under A.I. § 2 (3) f) are fully met prior to productive operation of the License Subject. § 377 of the German Commercial Code (HGB) remains unaffected.

(3) The Licensee shall be solely responsible for the installation and implementation of the License Subject on Licensee's systems.

(4) The Licensee shall take all necessary and reasonable measures to prevent or limit damage caused by the License Subject. In particular, the Licensee shall use the current protective mechanisms to defend against malware.

(5) The Licensee is aware that, under certain circumstances, separate drivers from the adapter manufacturers may be required for any required communication interfaces (in particular for the Motion Manager and Programming Libraries), which are not provided by FAULHABER. Communication interfaces are interfaces for data exchange between PC and controller, e.g. via CAN, RS232, USB or EtherCAT. The Licensee is obliged to independently procure, license and install any drivers required to use these communication interfaces.

(6) The Licensee is prohibited from removing or modifying any copyright information from the License Subject.

§ 5
Liability of FAULHABER

(1) Insofar as the License Subject is provided free of charge, the following liability regulations shall apply:

a) FAULHABER shall only be liable for material and legal defects if FAULHABER fraudulently conceals such a defect.

b) FAULHABER shall be liable in accordance with the statutory provisions in the event of intent, gross negligence and claims under the Product Liability Act, fraudulent concealment of a defect, guarantee claims and in the event of injury to life, limb or health. Otherwise, FAULHABER's liability for claims for damages and reimbursement of expenses - irrespective of the reason - shall be excluded.

c) Insofar as a loss or destruction of data at the Licensee was caused by grossly negligent or intentional breach of contractual or statutory obligations, FAULHABER shall only be liable up to the amount of the typical recovery expense that arises despite regular, state-of-the-art data backup.

(2) In the event that Firmware is provided for installation on hardware components of FAULHABER, the liability regulations applicable to the respective hardware component shall apply in deviation from subsection (1).

§ 6
Confidentiality

(1) The Parties undertake to treat as confidential for an unlimited period of time all information of the other Party obtained in the course of the initiation and execution of the Contract which is marked as confidential or is confidential by its nature ("**Confidential Information**") for an unlimited period of time and to use it only for the purpose of executing this Contract. FAULHABER's Confidential Information shall also include the License Subject. Notwithstanding the foregoing, unless otherwise agreed, License Subjects made publicly available for download by FAULHABER on FAULHABER's website shall not be deemed Confidential Information.

(2) The Licensee shall only make the License Subject accessible to employees and other third parties insofar as this is necessary to exercise the granted rights of use. The Licensee shall inform all persons to whom it grants access to the License Subject of FAULHABER's existing rights thereto and of the confidentiality obligations and shall oblige these persons in writing to maintain confidentiality to the same extent as in this § 6, insofar as the persons concerned are not obliged to maintain secrecy at least to the aforementioned extent for other legal reasons.

(3) The obligations of confidentiality under the preceding subsections shall not apply to Confidential Information that (i) was already in the public domain or known to the other Party at the time it was transmitted by the Party; (ii) became apparent after being transmitted by the Party through no fault of the other Party; (iii) have been made available to the other Party by a third party after their transmission by the Party in a non-illegal manner and without restriction as to confidentiality or exploitation; and/or (iv) have been developed by one Party independently without using the Confidential Information or trade secrets of the other Party. Furthermore, the obligations shall not apply if the Confidential Information has to be published according to law, in particular due to an administrative order or a court decision; in this respect, the publishing Party shall inform the other Party thereof without undue delay and support it in defending such orders or decisions.

§ 7
Final Provisions

(1) Changes or additions to this Contract shall be made in writing. If they do not satisfy this requirement, they are invalid. The same also applies to changes to this written form clause.
(2) This Contract is subject to and shall be interpreted according to the laws of the Federal Republic of Germany. The application of the UN sales law (CISG United Nations Convention on Contracts for International Sale of Goods of April 11, 1980) is excluded.
(3) The sole place of jurisdiction is Stuttgart, if the Licensee is a businessperson in the meaning of the Commercial Code, a legal person under public law or a special fund under public law or on bringing the action the Licensee does not have any registered offices or usual place of residence (permanent address) in the Federal Republic of Germany.
(4) Should a provision of this Contract be or become invalid, all other provisions shall remain unaffected. Statutory law shall take the place of provisions that are not included or are invalid (§ 306 (2) of the German Civil Code (BGB)). Otherwise, the Parties shall replace the void or invalid provision with a valid provision that comes as close as possible to it in economic terms, unless a supplementary interpretation of the contract takes precedence or is possible.

11.2 Components contained in the Motion Manager

11.2.1 SOEM (Simple Open EtherCAT Master Library)

Source code files: https://github.com/OpenEtherCATsociety/SOEM

11.3 Drivers supplied with the Motion Manager

The license texts of the drivers delivered with the Motion Manager are located in the respective installation folder under "\Motion Manager 6\Drivers" or are displayed when the setup is executed.

FTDI
USB-to-serial driver from Future Technology Devices International Limited for USB adapter boards on drives with RS232 interface.

MC3_WinUSB
USB driver for the Motion Controller Familie MC V3.0 family. Uses LibusbK from Travis Lee Robinson.

PCAP
Ethernet driver for connecting the Motion Manager to an EtherCAT network.

USBX
USB-to-serial driver from SILICON LABORATORIES, INC. for the programming adapters of the Speed Controller.

FAULHABER

DR. FRITZ FAULHABER GMBH & CO. KG
Antriebssysteme

Faulhaberstraße 1
71101 Schönaich • Germany
Tel. +49(0)7031/638-0
Fax +49(0)7031/638-100
info@faulhaber.de
www.faulhaber.com

7000.05057, 6th edition, 8.10.2022