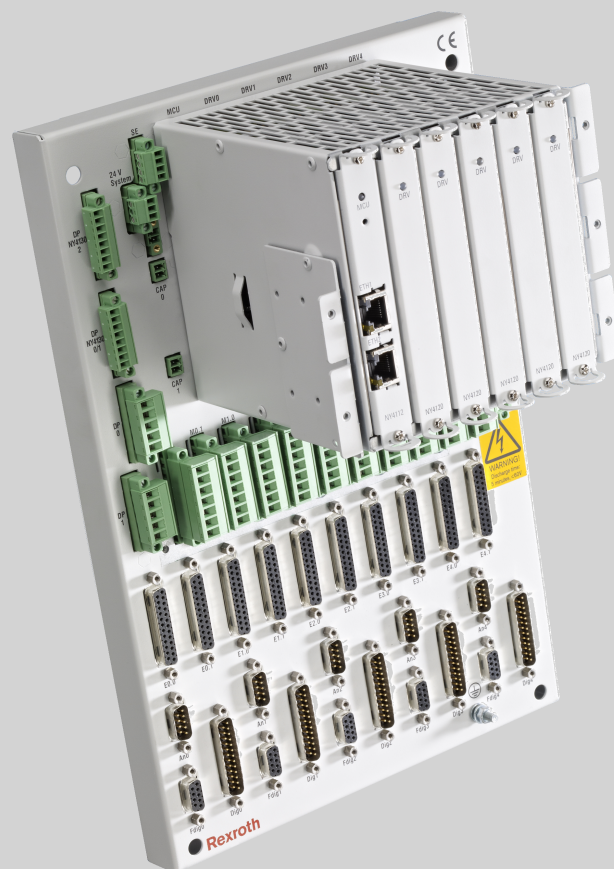


NYCe 4000

Multi-axis motion control system
Tools Manual

Commissioning
R911389806

Edition 01



Title NYCe 4000
Multi-axis motion control system
Tools Manual

Type of Documentation Commissioning

Document Typecode DOK-NY4000-TOOLS***V48-CO01-EN-E

Internal File Reference RS-9f9cf278618540b9c0a802861f6148f6-1-en-US-10

Record of Revision

Edition	Release Date	Notes
DOK-NY4000-TOOLS***V48-CO01-EN-E	08/2018	1st edition 48V04

List of trademarks

Microsoft®, Windows 7®, and Windows 10® are either registered trademarks or trademarks of Microsoft® Corporation in the United States and/or other countries.

Products in this publication are referred to by their general trade names. In most, if not all cases, these designations are claimed as trademarks or registered trademarks of their respective companies.

Copyright

© Bosch Rexroth AG 2018

All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

Liability

The specified data is intended for product description purposes only and shall not be deemed to be a guaranteed characteristic unless expressly stipulated in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.

Published by

Bosch Rexroth AG

P.O. Box 7170 ■ 5605 JD Eindhoven ■ The Netherlands

<http://www.boschrexroth.com>

Table of Contents

	Page
1 NYCe 4000 Tools.....	5
1.1 Introduction.....	5
1.2 General remark on XML formatted files.....	5
1.3 NYCe 4000 product and file version information.....	5
1.4 Used abbreviations in this manual.....	8
2 Getting started.....	11
2.1 Introduction.....	11
2.2 First time start-up of the NYCe 4000 System.....	11
2.2.1 Introduction.....	11
2.2.2 Startup of the MCU.....	15
2.2.3 Initialization with NYCeConfigurator.....	15
2.2.4 Initialization with NYCeCommand.....	24
2.2.5 Initialization with application program code.....	27
2.3 Behavioral differences between NY4110 and NY4112.....	27
2.3.1 Introduction.....	27
2.3.2 Network configuration.....	27
2.3.3 Network synchronization.....	28
2.3.4 Event response.....	29
2.4 Configuring versus tuning: parameter handling.....	30
2.5 Define an axis.....	30
2.5.1 Introduction.....	30
2.5.2 Create an axis.....	31
2.5.3 Configure the axis for a DC, BLAC, BLDC motor, or external drive amplifier.....	32
2.5.4 Configure the axis for a stepper motor or sensing stepper motor.....	36
2.5.5 Configure the axis for an external stepper drive motor.....	37
2.5.6 Configure the axis for an IndraDrive controlled motor.....	39
2.6 Quick start.....	43
2.6.1 Introduction.....	43
2.6.2 Signal Verification (not for stepper motors).....	48
2.6.3 Current Loop.....	49
2.6.4 Alignment (for BLAC motors only).....	50
2.6.5 Commutation (for BLDC motors only).....	51
2.6.6 Stepper Parameters (for stepper motors only).....	52
2.6.7 Polarity.....	53
2.6.8 Position Loop.....	54
2.6.9 Feedforwards.....	55
2.6.10 Settling Criteria.....	56
2.6.11 Get an IndraDrive-based motor operational (NYCeTuner "Expert" mode).....	57
3 NYCeConfigurator.....	59
3.1 Introduction.....	59
3.2 Parameter handling.....	60

Table of Contents

	Page
3.3	Save and restore settings..... 60
3.4	Configuration at the node level..... 62
3.5	Configuration at the MCU level..... 66
3.6	Configuration at the drive module level..... 67
3.6.1	Introduction..... 67
3.6.2	NY4120, NY4125, NY4130, NY4140 drive module..... 67
3.6.3	NY4150 SERCOS III Master module..... 71
3.6.4	IndraDrive..... 72
3.7	Configuration at the axis level..... 82
3.7.1	Introduction..... 82
3.7.2	"General" tab..... 83
3.7.3	Position sensor interface – "Encoders" tab (not for IndraDrive)..... 89
3.7.4	"Frequencies" tab..... 97
3.7.5	"Commutation" tab (not for IndraDrive)..... 99
3.7.6	"Motor" tab..... 102
3.7.7	"External Stepper" tab..... 104
3.7.8	"Digital I/O" tab..... 105
3.7.9	"Sensor supply" tab..... 106
3.7.10	"Home" tab..... 107
3.7.11	"Error Bounds" tab..... 108
3.7.12	"Error Handlers" tab..... 110
3.7.13	"Save / Download" tab..... 111
3.8	Miscellaneous..... 112
4	NYCeTuner..... 115
4.1	Introduction..... 115
4.2	Overview..... 115
4.3	Select and initialize an axis..... 116
4.4	Saving and shutting down..... 116
4.5	Expert mode..... 117
4.6	Standard controller parameters..... 121
4.7	Enhanced controller parameters..... 126
4.8	Control filter parameters..... 133
4.9	Current control parameters..... 138
4.10	Stepper control parameters..... 139
4.11	Alignment parameters (BLAC and IndraBLAC only)..... 141
4.11.1	Introduction..... 141
4.11.2	Mode 1: Kick & Swing..... 143
4.11.3	Mode 2: Wake & Shake..... 144
4.11.4	Mode 3: Set Commutation Offset..... 145
4.11.5	Mode 4: Align on Encoder Position..... 145
4.11.6	Mode 5: Catch & Move..... 146
4.11.7	Mode 6: Indra saturation..... 151
4.11.8	Mode 7: Indra sine wave..... 151
4.11.9	Mode 8: Indra locked..... 152
4.11.10	Mode 9: Push & Settle..... 152

Table of Contents

	Page
4.12	Miscellaneous NYCeTuner parameters..... 153
4.13	Controller settling criteria..... 155
4.14	AutoTweak tool..... 158
4.14.1	Introduction..... 158
4.14.2	Create the tweak data table..... 159
4.15	NYCeTuner motion options..... 160
4.15.1	Introduction..... 160
4.15.2	Test signal generator..... 161
4.15.3	Point To Point motion generator..... 162
4.15.4	Jogging motion generator..... 163
4.15.5	Gantry tuning..... 163
4.15.6	Drive Monitor Function..... 168
4.16	Axis tuning in "expert" mode..... 175
4.16.1	Introduction..... 175
4.16.2	Current loop tuning..... 175
4.16.3	Setting the alignment parameters (BLAC only)..... 179
4.16.4	Commutation order (BLDC only)..... 181
4.16.5	Measurement direction..... 181
4.16.6	The position loop tuning..... 182
4.16.7	Stepper motor tuning..... 184
4.17	Miscellaneous..... 186
5	NYCeScope..... 187
5.1	Introduction..... 187
5.2	Trace channel selection..... 187
5.3	Trace setup..... 189
5.3.1	Introduction..... 189
5.3.2	Trace size..... 189
5.3.3	Trigger source, slope, and trigger level..... 190
5.3.4	Frequency Analyzer..... 193
5.3.5	Start the measurement..... 195
5.4	Trace display manipulation..... 195
5.4.1	Time domain analysis..... 195
5.4.2	Control options in the display window..... 196
5.4.3	Frequency domain analysis..... 198
5.5	Trace states..... 201
5.6	Export and import trace data..... 202
5.6.1	Introduction..... 202
5.6.2	Export trace data..... 203
5.6.3	Import trace data..... 205
5.7	Sessions..... 205
5.8	Miscellaneous..... 206
6	NYCeInspector..... 209
6.1	Introduction..... 209

Table of Contents

	Page
6.2	Overview..... 209
6.3	Inspection at node level..... 210
6.4	Inspection at drive level..... 211
6.5	Inspection at axis level..... 212
6.6	Miscellaneous..... 215
7	NYCeCommand..... 217
7.1	Introduction..... 217
7.2	Commandline window..... 217
7.3	Script editor window..... 218
7.4	Help window..... 219
7.5	Using NYCeCommand from the "CMD" prompt..... 219
7.6	Script example..... 219
7.7	Miscellaneous..... 220
8	NYCeLogger..... 223
8.1	Introduction..... 223
8.2	Logging options..... 223
8.3	Miscellaneous..... 225
9	NYCeWatch..... 227
9.1	Introduction..... 227
9.2	Watching variables and parameters..... 227
9.3	Miscellaneous..... 228
10	NYCeNetMonitor..... 229
11	NYCeUpgradeTool..... 231
11.1	Introduction..... 231
11.2	Preconditions for NYCeUpgradeTool..... 231
11.3	Usage of NYCeUpgradeTool..... 232
11.4	Parameters of NYCeUpgradeTool..... 232
11.5	Return codes of NYCeUpgradeTool..... 234

1 NYCe 4000 Tools

1.1 Introduction

The NYCe 4000 Tools Manual provides information about the possibilities and functions of the software tools included with the NYCe 4000 software. The following tools are described in this manual.

- NYCeConfigurator (chapter 3 "NYCeConfigurator" on page 59)
- NYCeTuner (chapter 4 "NYCeTuner" on page 115)
- NYCeScope (chapter 5 "NYCeScope" on page 187)
- NYCeInspector (chapter 6 "NYCeInspector" on page 209)
- NYCeCommand (chapter 7 "NYCeCommand" on page 217)
- NYCeLogger (chapter 8 "NYCeLogger" on page 223)
- NYCeWatch (chapter 9 "NYCeWatch" on page 227)
- NYCeNetMonitor (chapter 10 "NYCeNetMonitor" on page 229)
- NYCeUpgradeTool (chapter 11 "NYCeUpgradeTool" on page 231)

The Sequencer and Simulation tools are described in separate manuals, NYCe 4000 Sequencer User Manual and NYCe 4000 Simulation User Manual.

A Getting Started chapter has been included for first time users of the NYCe 4000 system besides the functional descriptions of the tools.

The dialog screens shown in this manual may slightly differ from the actual displayed windows on the screen.

1.2 General remark on XML formatted files

The NYCe 4000 tools and software API functions read and write XML formatted files on the host. You can also modify these XML (configuration) files with a suitable (XML) editor.

If you do not use the NYCe 4000 tools or software API functions, but use an external editor to modify an XML formatted file, make sure that the file is saved ANSI encoded.

If the file is saved with a byte order mask (for example a UTF-8 byte order mask), the NYCe 4000 tools and API functions report the error `SAC_ERR_OPEN_FILE_ERROR` when such a file is opened. Further, make sure that the information that you add manually in an XML file is correct. Invalid information in an XML file may generate error messages or cause unexpected behavior.

1.3 NYCe 4000 product and file version information

Several files of the NYCe 4000 software contain version information of the NYCe 4000 product or the specific file. This information can be viewed in several ways, depending on the type of file.

- The NYCe 4000 tools "About" dialog window contains NYCe 4000 product and file specific information of the tool from which the "About" entry is selected.
- The file "Properties" dialog window contains version information of the file (.exe or .dll) from which the "Properties" entry is selected of the pop-up that appears after right-clicking the file entry.

NYCe 4000 Tools

- The file entry tool tip that appears when the mouse pointer hovers over a file entry.

Version information in the "About" dialog window

You can read the version information of the NYCe 4000 product and the version information of the tool in the "About" dialog window. The product version information is shown in the second line and the tool version information is shown in the third line, see [fig. 1-1 "Example: the "About NYCeConfigurator" dialog window" on page 6.](#)

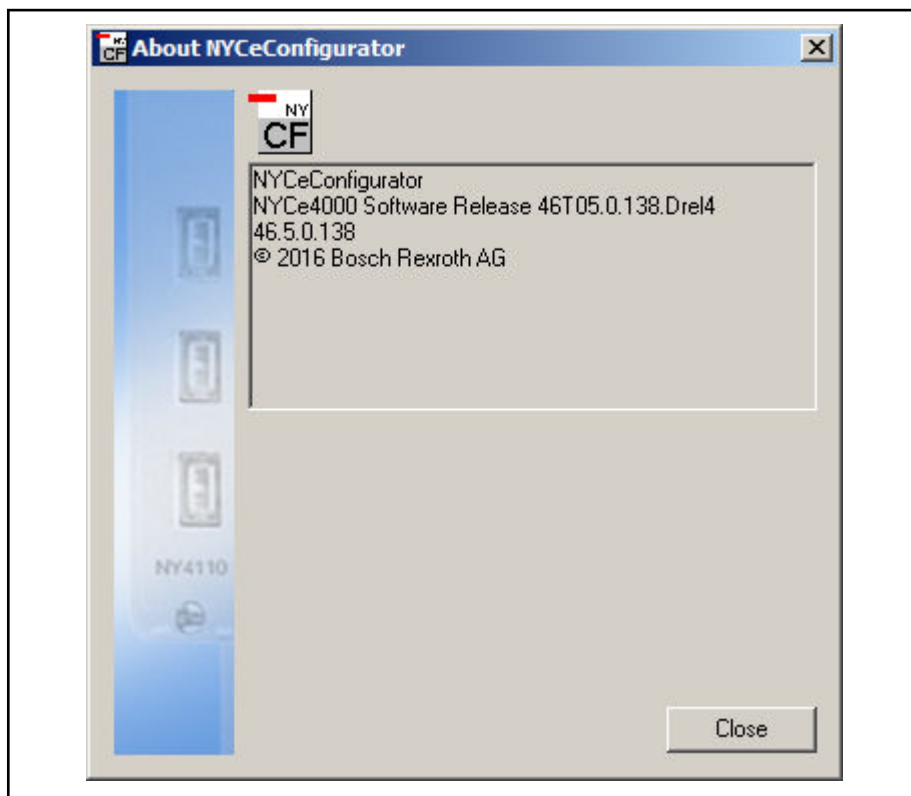


Fig. 1-1: Example: the "About NYCeConfigurator" dialog window

The product version can consist of 3 parts separated by a dot. The first part consists of a two digit product version number followed by the release type letter followed by a two digit release number. The release type can be either "V" for an official release, or "T" for a test version. The second part and third part are only present for release type "T". The second part consists of 4 digits and represents the NYCe 4000 software build number. The last part is the development release description which can be any identification text.

The tool version information consists of 4 numbers separated by a dot and represents the following data.

1. NYCe 4000 major release number.

The major release number is identical to the product version number and changes when new functionality is added or functionality is changed in the NYCe 4000 software.

2. NYCe 4000 minor release number.

The minor release number is identical to the product version number and changes when new functionality is added or functionality is changed in the NYCe 4000 software.

3. NYCe 4000 micro release number.

The micro release number changes when based on a Change Request an urgently required correction is implemented in a Service Pack.

4. NYCe 4000 software build number.

Version information in the file "Properties" dialog window

You can read the version information of the NYCe 4000 product and the version information of the tool in the "Properties" dialog window, see [fig. 1-2 "Example: file Properties dialog window of nyceconfigurator.exe" on page 7](#). The product version information is shown in the "Other version information" section when you select the Item name "Product Version". Likewise, the Item name "File Version" shows the file version information. The file version is also shown in the top section of the "Version" tab. If the file is a .Net module, the "Product Version" information is identical to the "File Version" information. The product version and file version information is explained in the paragraph "Version information in the "About" dialog window".

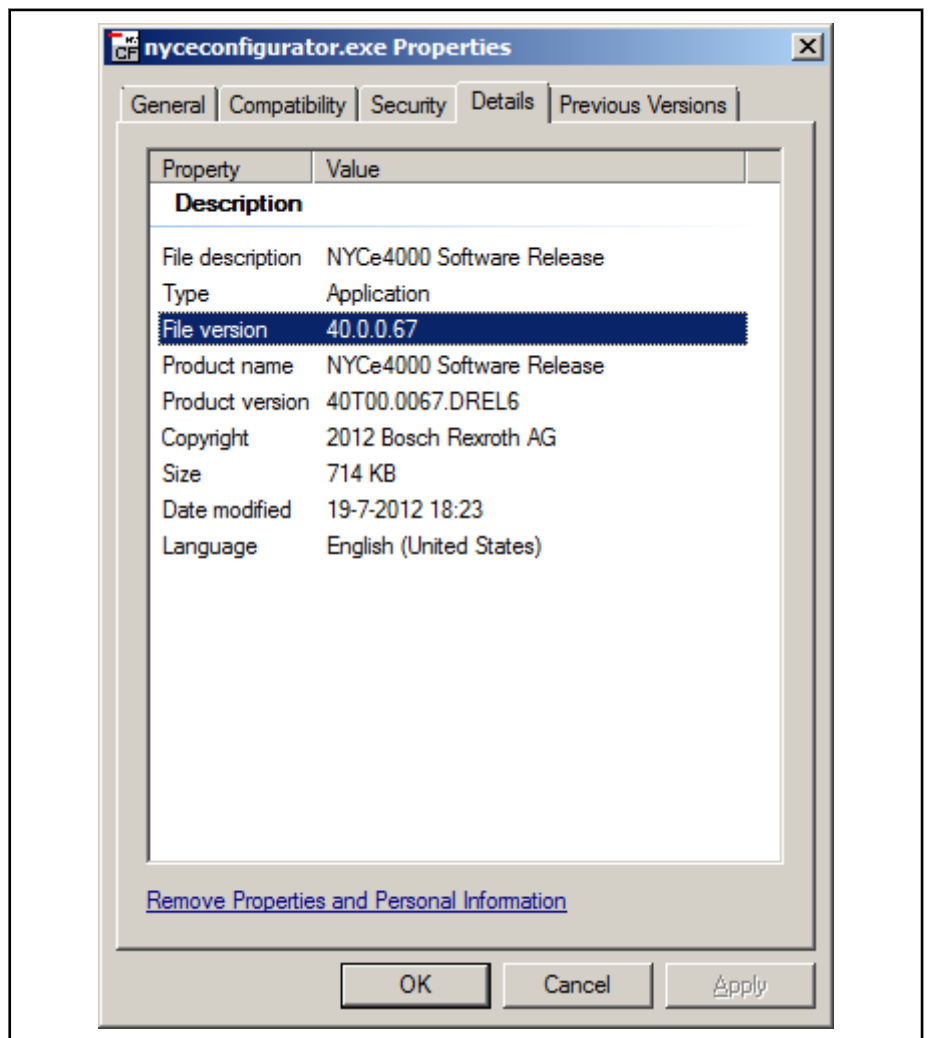


Fig. 1-2: Example: file Properties dialog window of nyceconfigurator.exe

However, if the file is a ".sys" file, the file version in the top section of the "Version" tab does not show the build number of the NYCe 4000 software in

NYCe 4000 Tools

the last part but the Windows version number of the header files used to build the .sys system file.

Version information in the tool tip of a file

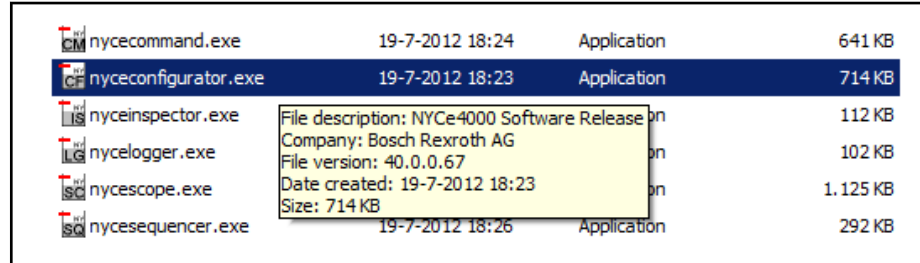


Fig. 1-3: Example: tool tip information of nyceconfigurator.exe

When the mouse pointer hovers over a file entry a tool tip appears. The tool tip shows the file version information of that file, see [fig. 1-3 "Example: tool tip information of nyceconfigurator.exe" on page 8](#). If the file is a system file, the last part of the version information does not show the build number of the NYCe 4000 product but the Windows version number of the header files used to build the .sys system file, see [fig. 1-4 "Example: tool tip information of n4kdriver.sys" on page 8](#).

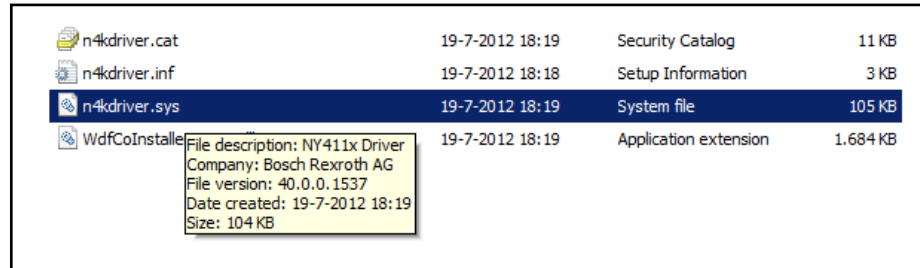


Fig. 1-4: Example: tool tip information of n4kdriver.sys

1.4 Used abbreviations in this manual

The following table gives a summary of abbreviations used in this manual.

Abbreviation	Description
BLAC	Brushless AC motor
BLDC	Brushless DC motor
CCL	Current Control Loop
cout	Controller out unit (user-defined dimension of controller out)
DMF	Drive Monitor Function
FFT	Fast Fourier Transformation
FRF	Frequency Response Function
IEEE	Institute of Electrical and Electronics Engineers
MCU	Motion Control Unit
NHI	Node Hardware Interface (subsystem)

Abbreviation	Description
NY4110	Motion Control Unit (MCU) with IEEE 1394b connections
NY4112	Motion Control Unit (MCU) with Ethernet connections
NY4120 NY4120/10	PWM Drive Module
NY4125	5-axis low power PWM Drive Module
NY4130	DC Drive Module
NY4140	High Voltage Drive Module
NY4150 NY4150/10	SERCOS III Master Module
NY4199	SinCos Option module
pu	Position unit (user-defined dimension of position)
PVL	Position Velocity Loop
PWM	Pulse Width Modulation
SAC	Single Axis Control (subsystem)
SERCOS III	Third generation SERCOS (Serial Real time Communication System) drive communication protocol based on the Ethernet physical hardware architecture
UDC	User Definable Control

Tab. 1-1: Summary of used abbreviations

2 Getting started

2.1 Introduction

This chapter describes how to get started with the NYCe 4000 system, using a basic way to get an axis operational with the NYCe 4000 tools. You can use this chapter as an introduction to the NYCe 4000 tools. The axis will be configured with revolutions as the dimension unit for the position and amperes as the dimension unit for the controller out. The axis must be driven by the appropriate drive module in this "getting started" example.

When you use this walkthrough, you must have a NYCe 4000 system ready for use with a DC motor, a brushless DC (BLDC) or a brushless AC (BLAC) rotary motor, a stepper motor, sensing stepper motor (stepper motor with a position encoder), or an IndraDrive-based motor connected to the first axis of the first drive. The motor must be equipped with an S0S90 encoder, if applicable. The S0S90 encoder is also known as A/B encoder, quadrature encoder and 3-channel digital incremental encoder. Consult the NYCe 4000 Hardware System Manual and NYCe 4000 Standard Housings and Accessories Manual for more information about the hardware setup of the NYCe 4000 system. The NYCe 4000 software must be installed on the computer.

⚠ WARNING

Unexpected movements may occur.

For safety reasons, disconnect the motor from the system until you use the NYCeTuner in this walkthrough. The encoder can always be connected.

2.2 First time start-up of the NYCe 4000 System

2.2.1 Introduction

When you start the NYCe 4000 node for the first time using a new NYCe 4000 software release, you must initialize the Motion Control Unit. Check the Release Bulletin that is included on the CD-ROM before you start. The initialization of the MCU consists of the download of the bootloader, the gateway files for the installed drive modules, and the firmware for the MCU. If the node contains an NY4150 or NY4150/10, you must also download the microware and the SERCON gateway. You can download the bootloader, gateway files, firmware, and microware and SERCON gateway, if applicable, with the NYCeConfigurator or NYCeCommand tool of the NYCe 4000 software package. NYCeConfigurator has a graphical user interface and is best suited for interactive use. NYCeCommand offers command-line style entry of NYCe 4000 commands and is best suited for command scripts.

The configuration of the ethernet interface of the NY4112 MCU cannot be done with the NYCeConfigurator or NYCeCommand tool. You must use the web browser interface of the NY4112 to do these actions.

Getting started



When you install a new NYCe 4000 software release and already have an upgrade-compatible NYCe 4000 software release installed, you can choose the "upgrade installation" method in the NYCe 4000 Software Release installation process. If "upgrade installation" is chosen, all nodes powered and connected to the PC are automatically upgraded with the new bootloader, gateway files for the installed drive modules, the firmware for the MCU, and if the node contains an NY4150 or NY4150/10, the microware and SERCON gateway.

This chapter describes the following tasks.

- [chapter 2.2.3 "Initialization with NYCeConfigurator" on page 15](#)
- [chapter 2.2.4 "Initialization with NYCeCommand" on page 24](#)
- [chapter 2.2.5 "Initialization with application program code" on page 27](#)

Points of attention

- Every factory-new NY4110 MCU has the same name (NY4110_node). If you build a multi-node network with new NY4110 MCU modules, the duplicate node name error is reported. You can solve this problem with the NYCeConfigurator tool, or via the `SysSetNodeName` function after you downloaded the bootloader, gateway file(s) and firmware to all MCUs and the MCUs.
- The MCU reverts to the default name NY411x_node when the flash memory is downgraded.
- Every factory-new NY4112 MCU has the same default IP address (192.168.41.12) and node name (NY411x_node). If you build a multi-node network with new NY4112 MCU modules, the duplicate IP address error and duplicate node name error is reported. You can solve both errors with the web server of the NY4112 MCU module. The web server can be accessed by the assigned IP address with any modern web browser. You must assign a unique IP address to every NY4112 MCU before the MCU is added to the NYCe 4000 network. Note that the IP address of the PC must also be unique in the network. Do the following steps to change the IP address of the MCU.
 1. Connect the NY4112 MCU to the PC ethernet port.



You must use ethernet port ETH1.

2. Enter in the address field of the web browser the IP address of the NY4112 MCU, `http://192.168.41.12`. The home page of the web server appears.



If you changed the IP address of the MCU in the past, you must use that IP address to access the MCU. Make sure to write down the changed IP address. Tools are available to find out the IP address if you forgot the IP address, for example a network sniffer.

Getting started

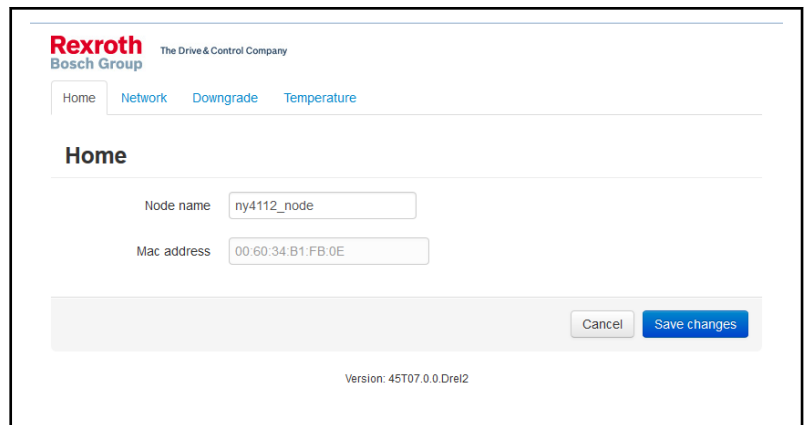


Fig. 2-1: The "Home" tab of the NY4112 web server

On the "Home" tab you can see and change the node name of the NY4112. Further, you can read the Media Access Code (MAC) address of the NY4112. You cannot change the MAC address, but you might need it for configuration of a switch or DHCP server in the ethernet network.

3. Select the "Network" tab.

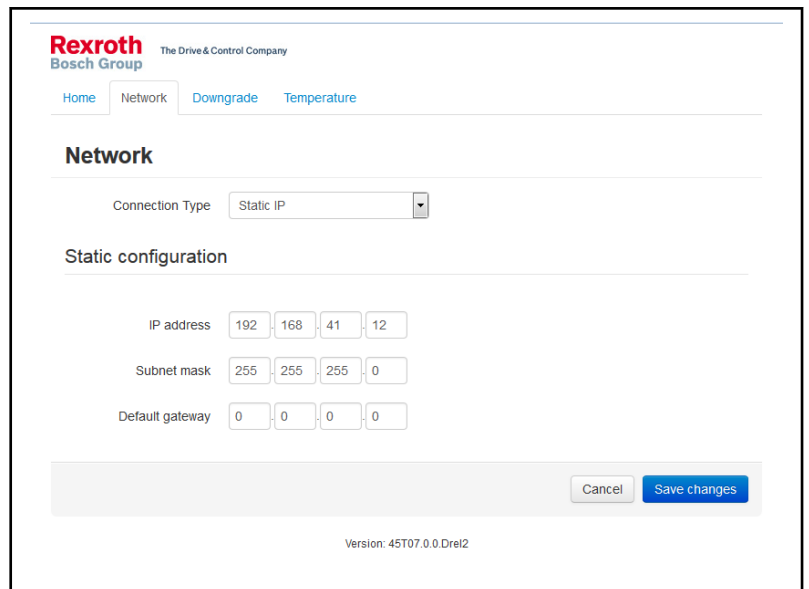


Fig. 2-2: The "Network" tab of the NY4112 web server

On the "Network" tab you can select the network IP address assignment method. You can choose between a static IP address, or a DHCP assigned address. If you select "Static IP" from the "Connection Type" drop-down box, you must specify the IP address, Subnet mask and Default gateway in the fields in the "Static configuration" section. If you select "Automatic - DHCP" from the "Connection Type" drop-down box, you must have a correctly configured DHCP server in the network.

If you have made changes on the "Network" tab you must click the "Save changes" button. The actual change is performed after the NY4112 is reset (power cycled).

Getting started

You can solve the duplicate node name on the "Home" tab of the web browser, with the NYCeConfigurator tool, or via the `SysSetNodeName` function after you downloaded the bootloader, gateway file(s) and firmware to all MCUs and the MCUs.

The web server of the MCU has two additional tabs, "Downgrade" and "Temperature".

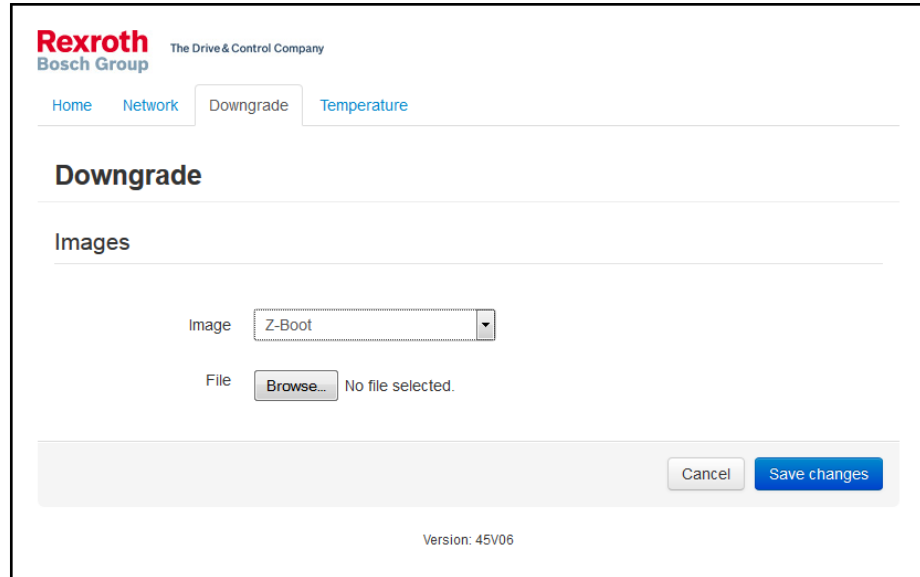


Fig. 2-3: The "Downgrade" tab of the NY4112 web server

The "Downgrade" tab offers the possibility to downgrade to a previous software release. See for more information the NYCe 4000 Release Bulletin.

The "Temperature" tab shows the minimum, measured maximum, and actual temperature of main components in the MCU.

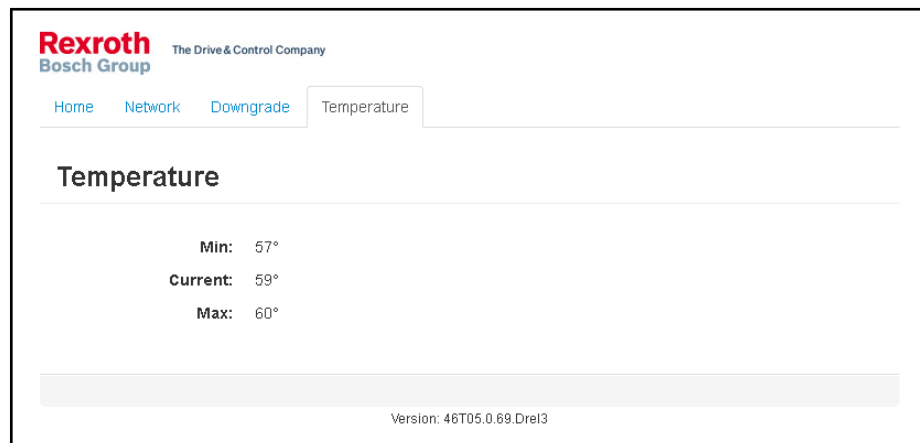


Fig. 2-4: The "Temperature" tab of the NY4112 web server

The maximum temperature is 125° Celsius.

2.2.2 Startup of the MCU

The MCU executes the start-up sequence after the 24V System power supply is switched on.

If the MCU is NY4112, the start-up sequence differs for Software Releases before 45V06 and Software Releases 45V06 or higher. If the MCU is NY4110, the start-up sequence is always as shown for Software Releases before 45V06.

- Software Releases before 45V06

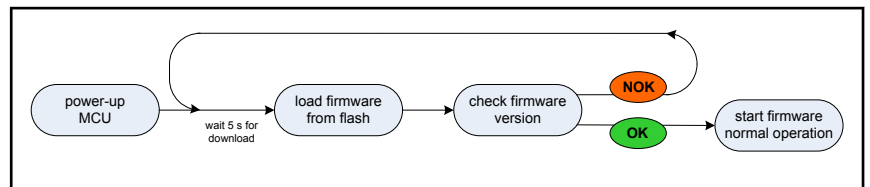


Fig. 2-5: NY4112 MCU power-up sequence (before 45V06)

The MCU loads the bootloader from flash memory and then starts the bootloader. The bootloader waits 5s for a possible download of a new bootloader or firmware to flash memory. If a new bootloader is downloaded the MCU must be reset to start the new bootloader. After the delay the bootloader checks whether firmware is stored in flash memory and, if available, loads and starts the firmware. If the firmware is not in the flash memory or is a wrong version, the bootloader starts the cycle again.

- Software Release 45V06 or higher

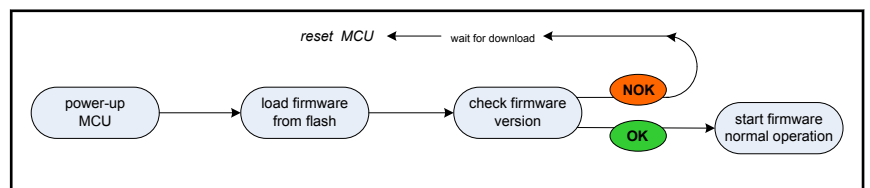


Fig. 2-6: NY4112 MCU power-up sequence (45V06 and higher)

The MCU loads the bootloader from flash memory and then starts the bootloader. The bootloader checks whether firmware is stored in flash memory and, if available, loads and starts the firmware. If the firmware is not in the flash memory or is a wrong version, the bootloader waits for the download of new software. The download can be a new bootloader, firmware or gateway file(s). If a new bootloader is downloaded the MCU can be reset to start the new bootloader before you continue. You can download the firmware and gateway file(s) in any order. After the download of the firmware and gateway file(s) the MCU must be reset. Note that you can also download gateway files after the firmware started the MCU in normal operation mode.

2.2.3 Initialization with NYCeConfigurator

NYCeConfigurator offers the possibility to download files to a single node, or download files to several nodes simultaneously. This chapter first describes the steps to download the required files to a single node. At the end of this chapter the steps are described to do a multi-node simultaneous download.

Getting started

Single-node initialization

Do the following steps in the described order to load a new bootloader, firmware and gateway.

1. Switch on the 24V System power supply to the NYCe 4000 system.
2. Start the NYCeConfigurator tool.
3. Select a specific node icon in the tree shown in the pane at the left side to download the required files to that selected node only.
4. Click the "Save / Download" tab in the pane at the right side, see [fig. 2-7 "Download the required files to a specific node" on page 16](#).

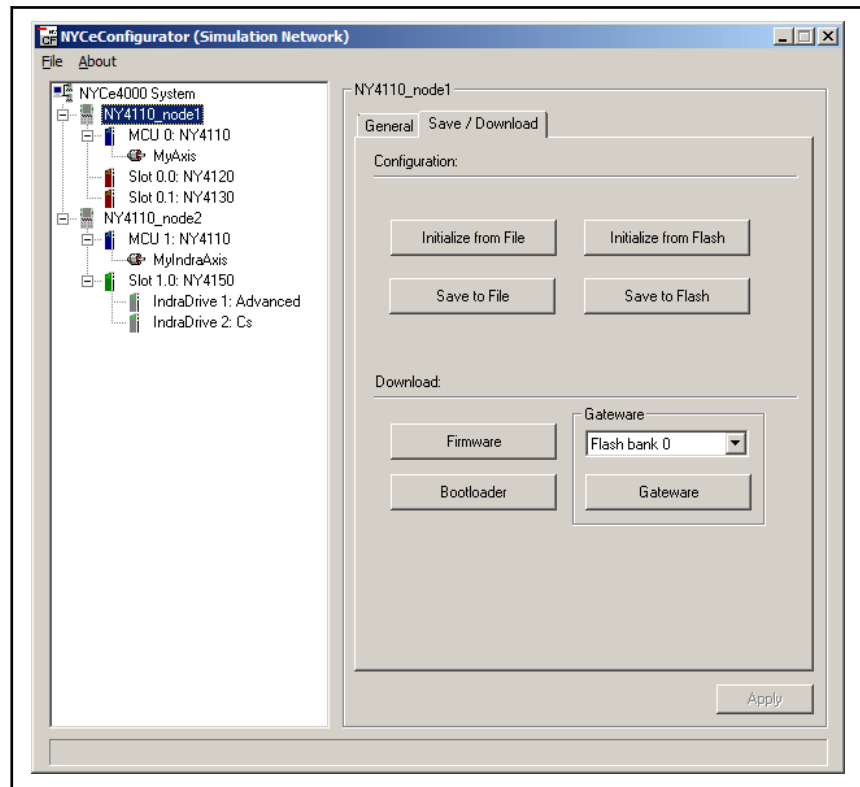


Fig. 2-7: Download the required files to a specific node

5. Click the "Bootloader" button on the "Save / Download" tab in the "Download" section. The "Download Bootloader to Node" dialog box appears.
6. Browse to the "upgrade" folder located in the folder that you specified during the installation of the NYCe 4000 software.

The default path used during the NYCe 4000 software installation is "C:\Program Files\Rexroth\NYCe4000".

If the MCU is an NY4110, select the bootloader file "boot4110.bin". If the MCU is an NY4112, select the bootloader file "boot4112.bin". Click the "Open" button, see [fig. 2-8 "Browse to the location of the bootloader file and select the bootloader file" on page 17](#). A message box is displayed during the download action.

Getting started

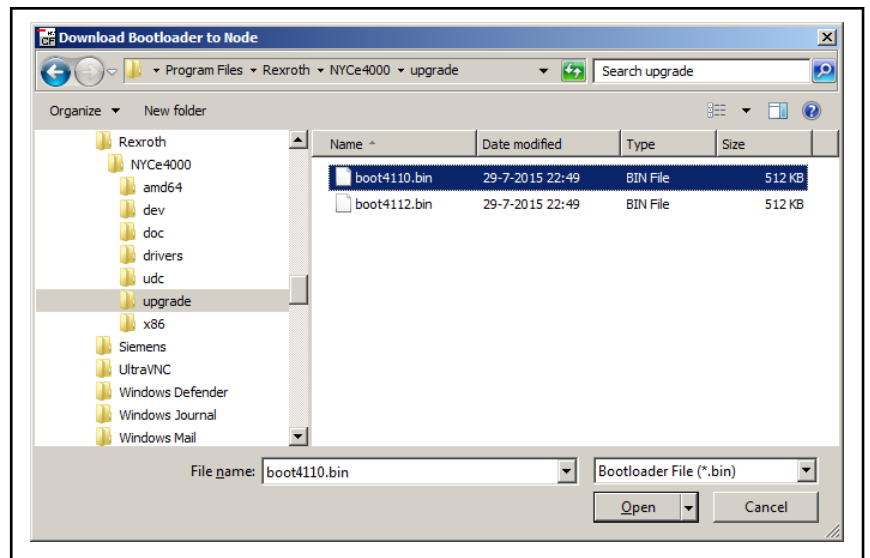


Fig. 2-8: Browse to the location of the bootloader file and select the bootloader file



- Use the "Bootloader" action with care: when the download is interrupted or aborted, the MCU will become unusable.
- Make sure that you use the correct bootloader file. The file "boot4110.bin" is only for the NY4110, and "boot4112.bin" is only for the NY4112. If you download the wrong bootloader, the error `DWN_ERR_FILE_CONTENTS_ERROR` is reported.
- You can download the bootloader only when all axes are in the IDLE state. An axis not in IDLE state can occur if you have a sequence stored in flash memory which uses an axis and has the bootstrap checkmark set. See the NYCe 4000 Sequencer User Manual for more information about sequences.

7. Reset the node.

After the bootloader is downloaded, you must reset the node to start the new bootloader.

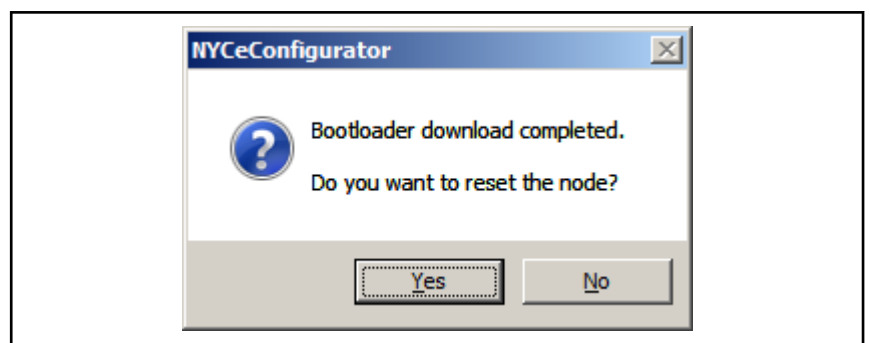


Fig. 2-9: Node reset after the bootloader is downloaded

8. Select "Flash bank 0" in the "Gateway" section.

The node has 3 banks in which gateway files can be downloaded. Best practice is to start downloading into bank 0. Note that the gateway file

Getting started

of the NY4125 requires 2 consecutive banks, thus can only be loaded into bank 0 or bank 1. The names of the gateway files that must be downloaded if the specified drive module is installed and the size of the gateway file (number of required banks) are listed in [tab. 2-1 "Module, gateway file name and size" on page 26](#). Download the gateway files, as described in the next two steps.

9. Click the "Gateway" button in the "Gateway" section.

The "Download Gateway to Node" dialog box appears, see [fig. 2-10 "Select the appropriate gateway file\(s\)" on page 18](#).

10. Select from the "upgrade" folder the appropriate gateway ".bit" file. You must download the gateway file of each drive module installed in the node. The names of the gateway files are self-explaining, that is, they include the drive module identification number (see [tab. 2-1 "Module, gateway file name and size" on page 26](#)). Click the "Open" button.

A message box is displayed during the download action.

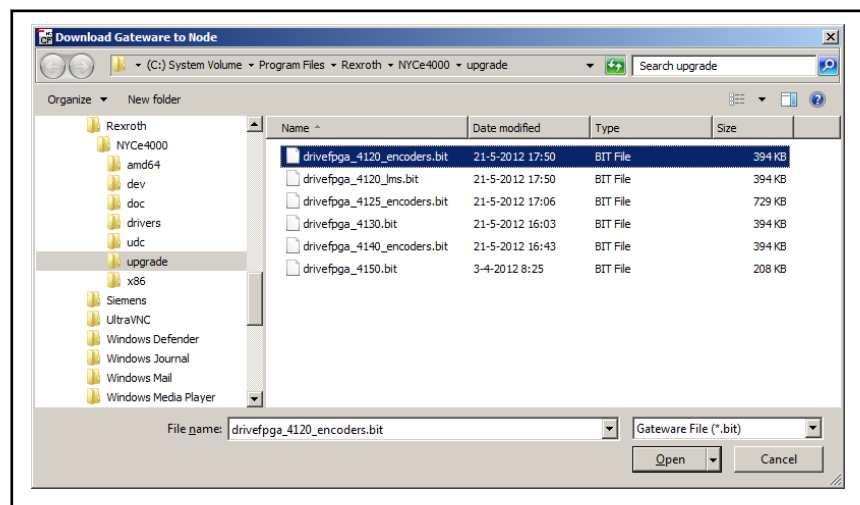


Fig. 2-10: Select the appropriate gateway file(s)

11. Repeat the previous 3 steps for all drive module types you want to install in the node. Make sure to select the consecutive next flash bank in the "Gateway" section.
12. Click the "Firmware" button in the "Download" section.

The "Download Firmware to Node" dialog box appears, see [fig. 2-11 "Select the firmware file" on page 19](#).

Getting started

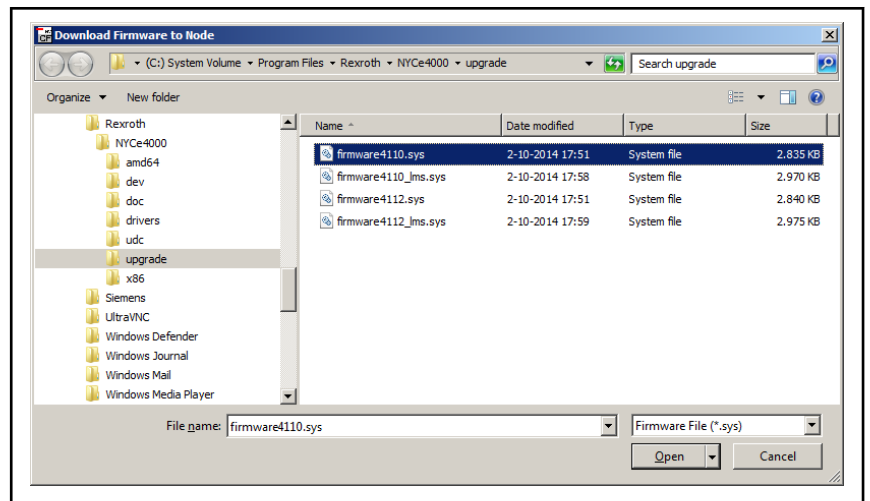


Fig. 2-11: Select the firmware file

13. Select the correct firmware file from the "upgrade" folder and click the "Open" button. The files "firmware4110.sys" and "firmware4110_lms.sys" are only for the NY4110, and "firmware4112.sys" and "firmware4112_lms.sys" are only for the NY4112. NYCeConfigurator displays an error message ("DWN_ERR_FILE_CONTENTS_ERROR") if you download NY4110 firmware to the NY4112 MCU, or download NY4112 firmware to the NY4110 MCU.

A message box is displayed during the download action.



- Do not interrupt or abort the download of the firmware, the MCU will become unusable.

When the download of the firmware is finished, you may be asked to restart the node, if necessary. You must restart the node to start the (new) downloaded firmware.

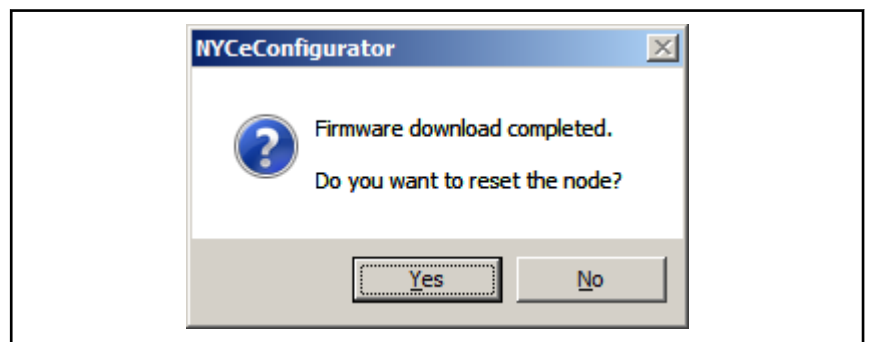


Fig. 2-12: Required node reset after a firmware download

Getting started



If you first download the firmware, you must reset the node, and you will have to reset the node again after you download the gateway. Download the gateway files and then the firmware in the described order to minimize the number of times that a node must be reset.

After the system is restarted you may get another pop-up message. This message reports that the data in the flash memory is not consistent with the current configuration. You are asked whether you want to save the current configuration to flash memory and reset the node to get a consistent configuration.

If the NYCe 4000 system includes the NY4150 module, you must download the microware and the SERCON gateway, after the MCU is restarted. Note that you must have downloaded the gateway of the NY4150 in step 7, 8 and 9 else the "Download" tab will not be available.

Do the following steps to configure the NY4150 SERCOS III Master module.

1. Select the NY4150 module icon of the specific node in the tree shown in the pane at the left side, and click the "Download" tab in the pane at the right side. See [fig. 2-13 "Download tab specific for the NY4150 SERCOS III Master module" on page 20.](#)

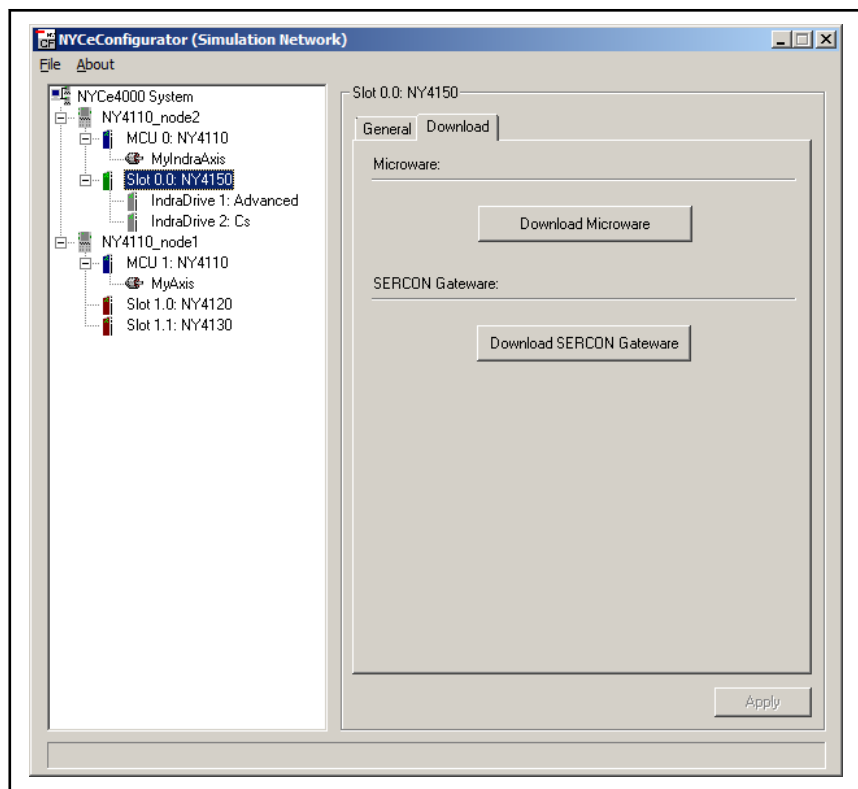


Fig. 2-13: Download tab specific for the NY4150 SERCOS III Master module

2. Click the "Download SERCON Gateway" button in the "SERCON Gateway" section. The "Download SERCON Gateway to Slot 0.0: NY4150" dialog box appears.

Getting started

3. Select the SERCON gateway file "sercon100m.sgw" from the "upgrade" folder and click the "Open" button. A message box is displayed during the download action.
4. Click the "Download Microware" button in the "Microware" section. The "Download Microware to Slot 0.0: NY4150" dialog box appears.
The numbers in the title caption "Slot 0.0" depend on the node number and the I/O slot number (always 0) in which the NY4150 module is installed.
5. Select a microware file, for example "microware4khz.mcr", from the "upgrade" folder and click the "Open" button. A message box is displayed during the download action.

When the download of the microware is finished, you are asked to restart the node. You must restart the node to start the (new) downloaded microware.

Multi-node initialization

All involved nodes must be connected and powered before you can do a multi-node initialization. The way how the NYCe 4000 network connections are realized depends on the type of MCU installed in the system housings.

NY4110-based multi-node network

- The MCUs are connected either in a tree topology (preferred), or in a daisy-chain setup, with IEEE1394b cables. This configuration setup can already be installed in the machine.

NY4112-based multi-node network

- If the NY4112 MCU has previously been used in a machine setup, any already installed network configuration can be used to do a multi-node initialization.

Factory-new NY4112 MCUs

However, if you use factory-new NY4112 MCUs, you can only establish a connection with ethernet port ETH1.

Two setups are possible to configure the factory-new MCUs.

1. Connect one MCU at a time to the PC and do the configuration steps. See [chapter 2.2.1 "Introduction" on page 11](#) "Points of attention".
2. Use a switch between the PC and all MCUs and do the configuration steps. A daisy-chain configuration is not possible, because ETH2 is non-functional on a factory-new NY4112 MCU. The switch is needed to connect all nodes. Every node is connected to a port on the switch with a standard ethernet cable, see [fig. 2-14 "Connection between PC and nodes with a switch \(for configuration\)" on page 22](#). If this configuration does not match the configuration in the machine, you must setup this configuration separately. The switch does not need to support IEEE1588 for configuration in this (separate) configuration setup.

Getting started

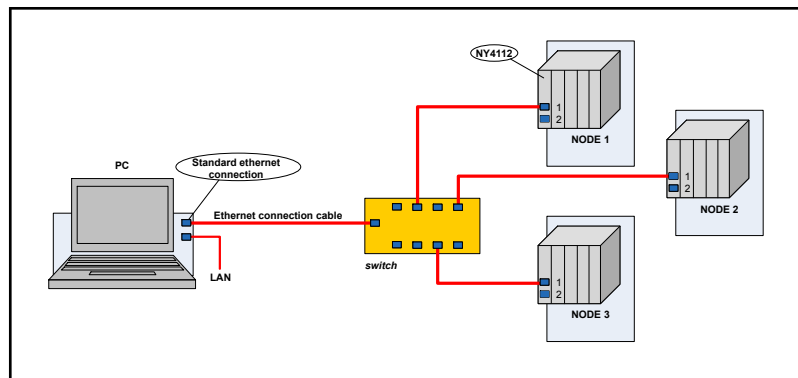


Fig. 2-14: Connection between PC and nodes with a switch (for configuration)



Every MCU must have a unique IP address.

Do the following steps to download the files to all nodes simultaneously.

1. Switch on the 24V System power supply to all nodes the NYCe 4000 system.
2. Start the NYCeConfigurator tool.
3. Right-click the "NYCe4000 System" icon in the pane at the left side.

A popup menu appears.

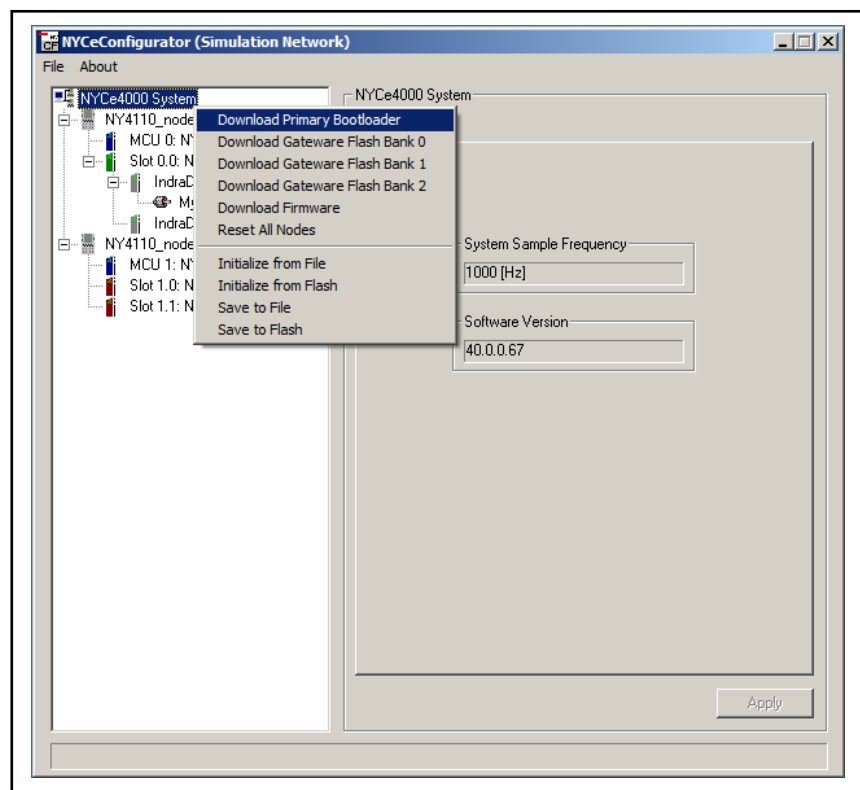


Fig. 2-15: The menu options at the NYCe4000 System level

Getting started

After the selection of an entry in the popup menu, the specific dialog box appears as described before, and you can select the file that must be downloaded.

4. Select the entry "Download Primary Bootloader" from the popup menu.
The "Download Bootloader to Node" dialog box appears.
5. Browse to the "upgrade" folder located in the folder that you specified during the installation of the NYCe 4000 software.

The default path used during the NYCe 4000 software installation is "C:\Program Files\Rexroth\NYCe4000".

If the MCU is an NY4110, select the bootloader file "boot4110.bin". If the MCU is an NY4112, select the bootloader file "boot4112.bin".

Click the "Open" button. A message box is displayed during the download action.



- Use the "Bootloader" action with care: when the download is interrupted or aborted, the MCU will become unusable.
- Make sure that you use the correct bootloader file. The file "boot4110.bin" is only for the NY4110, and "boot4112.bin" is only for the NY4112. If you download the wrong bootloader, the error `DWN_ERR_FILE_CONTENTS_ERROR` is reported.
- You can download the bootloader only when all axes are in the `IDLE` state. An axis not in `IDLE` state can occur if you have a sequence stored in flash memory which uses an axis and has the bootstrap checkmark set. See the NYCe 4000 Sequencer User Manual for more information about sequences.

-
6. Select the entry "Download Firmware" from the popup menu.
The "Download Firmware to Node" dialog box appears.
 7. Select the correct firmware file from the "upgrade" folder and click the "Open" button. The files "firmware4110.sys" and "firmware4110_lms.sys" are only for the NY4110, and "firmware4112.sys" and "firmware4112_lms.sys" are only for the NY4112. NYCeConfigurator displays an error message ("`DWN_ERR_FILE_CONTENTS_ERROR`") if you download NY4110 firmware to the NY4112 MCU, or download NY4112 firmware to the NY4110 MCU.

A message box is displayed during the download action.



- Do not interrupt or abort the download of the firmware, the MCU will become unusable.

-
8. Power cycle off/on the 24V System to reset the nodes.



You cannot have a mix of firmware types in a single download. If your application uses a mix of LMS nodes and STD nodes, you must do the multi-node download in two setups. The first setup has only all LMS nodes and you download the "firmware4110_lms.sys" or "firmware4112_lms.sys" file. The second setup has only all STD nodes and you download the "firmware4110.sys" or "firmware4112.sys" file.

Getting started

After these steps you can continue the initialization of all nodes. If the MCUs are not in the machine configuration, you can install the MCUs in the machine before you do the final steps of the initialization.

1. Download the required gateway files.
2. Rename the node, if needed. Node names must be unique in the network.
3. Power cycle off/on the 24V System to reset the nodes.
4. After the system is restarted you may get another pop-up message. This message reports that the data in the flash memory is not consistent with the current configuration. You are asked whether you want to save the current configuration to flash memory and reset the node to get a consistent configuration.
Save the configuration data via the pop-up message "Inconsistent Flash Configuration".
5. If the system contains an NY4150 SERCOS III Master module, you must use the described steps 1 to 5 of the previous paragraph to download the SERCON gateway and microware.

The nodes are now ready for use.

2.2.4 Initialization with NYCeCommand

Switch on the 24V System power supply to the NYCe 4000 system (only the MCU needs to be installed) and start the NYCeCommand tool.

Enter the following commands in this order:

1. Download the new primary bootloader. The command `DwnDownloadPrimaryBoot` can be abbreviated to `DDLPB`.
`DDLPB [NodeNr] [PATH]\boot4110.bin`



- `DwnDownloadPrimaryBoot` must be used with care: when the download is interrupted or aborted, the MCU will become unusable.
- Make sure that you use the correct bootloader file. The file "boot4110.bin" is only for the NY4110, and "boot4112.bin" is only for the NY4112. If you download the wrong bootloader, the error `DWN_ERR_INVALID_FILE_TYPE` is reported.
- You can download the bootloader only when all axes are in the `IDLE` state. An axis not in `IDLE` state can occur if you have a sequence stored in flash memory which uses an axis and has the bootstrap checkmark set. In that case you can downgrade the flash memory and download the bootloader only successfully when the node status is `NODE_STATUS_BL_WAITING_FOR_DOWNLOAD_COMMAND`.

2. After the bootloader is downloaded, you must power cycle off/on the 24V System to reset the node to start the new bootloader.
3. Erase old gateway files. The command `DwnEraseGateway` can be abbreviated to `DERG`.
`DERG [NodeNr]`
4. After the gateway banks are erased, you can download the new gateway files. The node has 3 banks in which gateway files can be downloaded. Best practice is to start downloading into bank 0. Which

Getting started

gateway files you download to the MCU depends on the installed drive modules in the node. Note that the gateway file of the NY4125 requires 2 consecutive banks, thus can only be loaded into bank 0 or bank 1. The names of the gateway files that must be downloaded if the specified drive module is installed and the size of the gateway file (number of required banks) are listed in [tab. 2-1 "Module, gateway file name and size" on page 26](#).

The command `DwnDownloadGateway` can be abbreviated to `DDLG`.

Examples of gateway download commands.

- NYCe 4000 system with NY4120, NY4130 and NY4150 modules.
 - `DDLG [NodeNr] 0 [PATH]\drivefpga_4120_encoders.bit`
 - `DDLG [NodeNr] 1 [PATH]\drivefpga_4130.bit`
 - `DDLG [NodeNr] 2 [PATH]\drivefpga_4150.bit`
- NYCe 4000 system with NY4125 and NY4140 drive modules.
 - `DDLG [NodeNr] 0 [PATH]\drivefpga_4125_encoders.bit`
 - `DDLG [NodeNr] 2 [PATH]\drivefpga_4140_encoders.bit`
- NYCe 4000 system LMS with NY4120 LMS drive module(s).
 - `DDLG [NodeNr] 0 [PATH]\drivefpga_4120_lms.bit`

5. Download the firmware. The command `DwnDownloadFirmware` can be abbreviated to `DDLDF`.

```
DDLDF [NodeNr] [PATH]\firmware4110.sys
```

The files "firmware4110.sys" and "firmware4110_lms.sys" are only for the NY4110, and "firmware4112.sys" and "firmware4112_lms.sys" are only for the NY4112.



- Do not interrupt or abort the download of the firmware, the MCU will become unusable.
-

- If you download firmware for the NY4112 to an NY4110, the MCU and NET LED on the front of the NY4110 will be continuous orange after the MCU is reset. Download the correct firmware to correct this condition.
 - If you download firmware for the NY4110 to an NY4112, the MCU LED on the front of the NY4112 will be continuous orange, then for 1 second green and then for 1 second red, and then this cycle repeats. Download the correct firmware to correct this condition.
6. You must reset the MCU after you downloaded the firmware to start the new firmware.

```
SysResetNode [NodeNr]
```

or power cycle off/on the 24V System power supply.

Getting started



If you first download the firmware, you must reset the node, and you will have to reset the node again after you download the gateway. Download the gateway files and then the firmware in the described order to minimize the number of times that a node must be reset.

After the system is restarted it is possible that the data in the flash memory is not consistent with the current configuration. You can use NYCeConfigurator to save the current configuration to flash memory. You must reset the node to get a consistent configuration.

Drive module	Gateway file name	Size
NY4120	drivefpga_4120_encoders.bit	1
	drivefpga_4120_aspi.bit	1
NY4120 LMS	drivefpga_4120_lms.bit	1
NY4125	drivefpga_4125_encoders.bit	2
NY4130	drivefpga_4130.bit	1
NY4140	drivefpga_4140_encoders.bit	1
NY4150 NY4150/10	drivefpga_4150.bit	1

Tab. 2-1: Module, gateway file name and size



- All commands are case sensitive, and entered on one line.
- [NodeNr] is the number of the node in the node network. The first node connected to the computer usually has node number 0.
- [PATH] is the location of the .bit, .bin and .sys files. If you installed the NYCe 4000 software in the default installation folder, [PATH] is "C:\Program Files\Rexroth\NYCe4000\upgrade"

If the NYCe 4000 system includes the NY4150 (or NY4150/10) module, you must download the microware and the SERCON gateway, after the NYCe 4000 is restarted. Note that you must have downloaded the gateway of the NY4150 in step 4 else you cannot download the microware and the SERCON gateway.

Do the following steps to configure the NY4150 SERCOS III Master module.

1. Download the NY4150 SERCON gateway. The command `DwnDownloadSerconGateway` can be abbreviated to `DDLS`.
`DDLS 0 [PATH]\sercon100m.sgw`
2. Download the NY4150 microware. The command `DwnDownloadMicroware` can be abbreviated to `DDLm`. To download the microware4khz.mcr file (for example), the command is
`DDLm 0 [PATH]\microware4khz.mcr`
3. You must reset the node after you downloaded the SERCON gateway and microware to start this new software.

```
SysResetNode [NodeNr]
```

or power cycle off/on the 24V System power supply.

The node is now ready for use.

2.2.5 Initialization with application program code

As described in the previous two chapters, you can initialize a node with NYCeConfigurator or NYCeCommand. The initialization of a system can also be realized with the functions of the DWN and SYS subsystems. Care must be taken of the synchronization after the bootloader, gateway, and firmware is downloaded and the node is reset.

If the application consists of several nodes, the initialization will take some time. In a multi-node network you can decrease the initialization time by executing commands in parallel (in different threads). Parallel execution of commands is possible if the following rules are followed.

- All DWN download commands can be executed in parallel as long as the commands are not sent to the same node.
- Every MCU in an NY4112-based multi-node network stops after the download operation of the files is finished. See [chapter 2.2.2 "Startup of the MCU" on page 15](#). You must reset all nodes **after** the download operation for the last node is finished.
- In an NY4110-based network, if a node executes the bootloader and the firmware is stored in the node, the node will start the initialization if for a time period of 5 seconds no command is received. If a node starts the initialization a bus reset is generated by the node. All download operations that have not yet finished to other nodes in the network will return with an error. To prevent this condition to occur, send a command to every node every few seconds. A command that can be used for this purpose is, for example, `SysBlinkNode (nodeNr, 0)` and ignore the returned error code.

2.3 Behavioral differences between NY4110 and NY4112

2.3.1 Introduction

The major difference that has impact on the application is the different network interface. The NY4110 MCU uses the IEEE1394b interface and the NY4112 MCU uses the ethernet interface. These different interfaces have several consequences with regard to configuration and response to events that can occur in the network.

2.3.2 Network configuration

The address allocation of nodes in an NY4110-based network is handled by the IEEE1394b protocol. The address allocation of nodes in an NY4112-based network is realized by the assigned IP address of every MCU in the network. A factory-new NY4112 MCU has always the same default IP address (192.168.41.12). If you build a multi-node network with new NY4112 MCU modules, the duplicate IP address error is reported. You must assign a unique IP address to every NY4112 MCU before the MCU is added to the NYCe 4000 network.

The node number assigned to each node in an IEEE1394b network is based on the physical location of the node in the network (tree). The node number assigned to each node in an ethernet network depends on the name of every node. The assigned node number is based on the alphabetical list order of the node names in the network.

Getting started

Factory-new NY4112 MCUs in a multi-node network

If you have a multi-node NY4110-based network and want to upgrade the network with (factory-new) NY4112 MCUs, or create a new multi-node network with (factory-new) NY4112 MCUs, you must do the following steps.

1. Each (factory-new) NY4112 MCU must be configured separately (in a single-node network).
 - 1.1 Connect the NY4112 MCU to the PC ethernet port.



You must use ethernet port ETH1.

- 1.2 Switch on the 24V System power.
- 1.3 Use the web browser to assign a unique IP address to the MCU.
- 1.4 Use NYCeConfigurator to download files and set a unique node name.
 - 1.4.1 Download the bootloader.
 - 1.4.2 Power cycle off/on the 24V System to reset the MCU.
 - 1.4.3 Download the appropriate gateway files (optional).
 - 1.4.4 Download the firmware.
 - 1.4.5 Power cycle off/on the 24V System to reset the MCU.
 - 1.4.6 Change the node name.
- 1.5 Switch off the 24V System power.

2. Install the configured NY4112 MCUs in the machine.
3. Power up all nodes.
4. Wait until the network is stable. The network is stable when the "view" of the network no longer changes. When the network is stable, you can see all nodes in the network tree view of NYCeConfigurator.
5. Download the appropriate gateway files (if needed). If you downloaded gateway files, you must reset the nodes.

A NYCe 4000 multi-node ethernet network can be in the following state.

1. not stable

The nodes do not have the same "view" of the network. The "not stable" state typically exists

- shortly after power-up,
- when one (or more) nodes are reset,
- when ethernet cables are connected or disconnected,
- when a node name is changed.

2. stable

All nodes have the same "view" of the network.

3. synchronized

All nodes have the same "view" of the network, and all nodes run sample-synchronous.

2.3.3 Network synchronization

In an NY4110-based network all nodes execute commands "sample synchronous", meaning that you can start the movement of several axes on different nodes at exactly the same moment. You can say that all nodes are

synchronized. This "all nodes are synchronized" is part of the handling of the IEEE1394b communication. In an ethernet network a separate protocol is required (IEEE1588, Precision Time Protocol) to ensure that "all nodes are synchronized". The time needed to synchronize all nodes in a network takes longer in an ethernet network compared to an IEEE1394b network. The application can check that the nodes in the network are synchronized with the API function `SysSynchronize`. Note that all nodes in the NYCe 4000 network must be synchronized to guarantee "sample synchronous" command execution. If a function is called that requires a synchronized network and the network is not synchronized, an error is returned.

All NYCe 4000 tools that display the network will display each node as announced and added to the network after start-up. You can see the network "grow". The entire network is displayed as soon as the network is stable. Stable means that all nodes on the network are recognized. When the network tree is displayed you can execute all kind of functionality offered by the tools. However, certain functionality requires a synchronized network, for example the trace functionality (and NYCeScope), MAC and error groups and IndraDrives. The tree view of the NYCe 4000 network shows in the root whether the network is synchronized or not. A synchronized network is indicated with the text "NYCe4000 System". A not-synchronized network is indicated with the text "NYCe4000 System (synchronizing)". The time between a stable network and a synchronized network is instantaneous for an IEEE1394b-based network and approximately 90s for an ethernet-based network.



It is advised that a NYCe 4000 application always starts with the execution of the function `SysSynchronize` with the synchronization request `SYS_REQ_NETWORK_SYNCHRONIZED`. Although not required for an IEEE1394b-based network, this is good programming practice and prepares the application for an easy upgrade to an ethernet-based motion solution.

The following actions cause the system to drop out of the synchronized state.

- Reset of a node,
- connection or disconnection of ethernet cables,
- rename a node name.



Do no actions that cause the system to drop out of the synchronized state if one axis or more axes in the system is not in the idle state.

2.3.4 Event response

Several asynchronous events can occur in a network. The handling of an event may be different, depending on the type of network, IEEE1394b or ethernet.

A node name change has no effects in an IEEE1394b-based network, because the node order is not based on the node name, but on the location of the node in the IEEE1394b network. A node name change in an ethernet-based network will cause a "network configuration change" event, because the node order is based on the node name. As a result of the node name change in an ethernet-based network, the network is temporarily not stable and all nodes in the network are no longer synchronized. After the network is stable again it takes up to 90s before all nodes in the network are synchronized.

Getting started

2.4 Configuring versus tuning: parameter handling

You can change parameters in the NYCe 4000 system with the NYCeConfigurator and the NYCeTuner tool. The NYCeConfigurator is used for setting parameters that are fixed for a certain configuration. These parameters, for example motor and sensor type, only need to be set once. The NYCeTuner is used for setting parameters during optimizing the axis behavior, for example the gain of the position loop. These parameters are changed by the user during the process optimization.

Parameter handling is different between the NYCeConfigurator and the NYCeTuner for this reason. Compare the two figures below. If you change a parameter in the NYCeConfigurator, the change is only applied in the NYCeConfigurator itself and not in the node. Only after you click the "Apply" or "OK" button, the new parameter value(s) are sent to the node. When you enter a value in the NYCeTuner, that value is directly sent to the node.

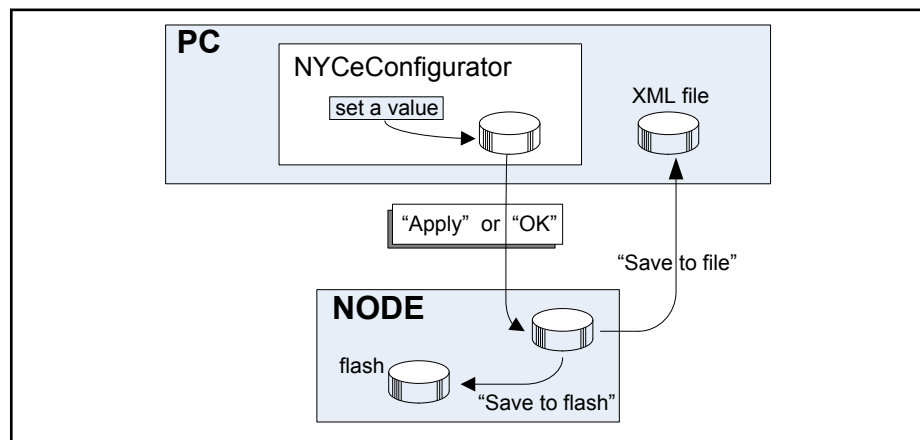


Fig. 2-16: Parameter handling using NYCeConfigurator

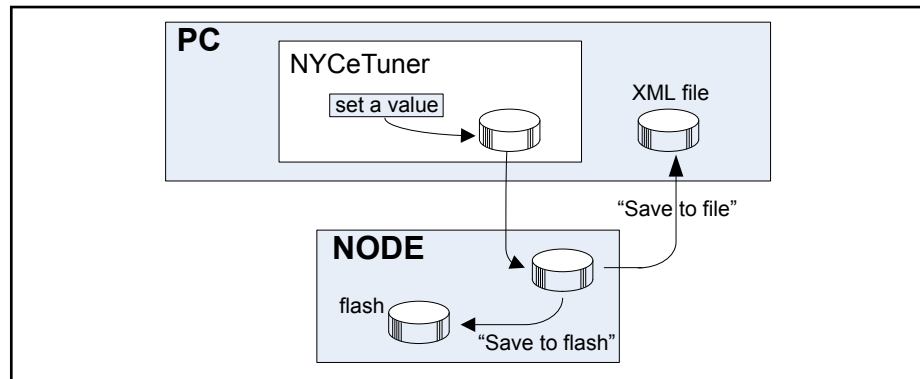


Fig. 2-17: Parameter handling using NYCeTuner

To apply certain configuration settings, the NYCeConfigurator requests for an axis shutdown. To be able to use the applied settings after a (re-)initialize, you must first save the settings to a file or flash memory after the shutdown of the axis is done.

2.5 Define an axis

2.5.1 Introduction

The procedure to get a motor operational with the NYCe 4000 system basically consists of 3 steps.

1. Create an axis with the NYCeConfigurator tool.
2. Configure the axis with the NYCeConfigurator tool.

The steps to configure a stepper motor axis, sensing stepper motor axis, external stepper drive axis, or an IndraDrive controlled motor are different from the steps to configure a DC, BLAC, or BLDC motor. For this reason the steps for the configuration of a stepper motor, sensing stepper motor or external stepper motor and the steps for the configuration of an IndraDrive controlled motor are separately described.

3. Tune the axis with the NYCeTuner tool.

In this chapter an axis is created and configured. The [chapter 2.6 "Quick start" on page 43](#) helps you to get an axis initially operational. For optimal performance of an axis, the NYCe 4000 software provides additional tools, like the NYCeTuner and NYCeScope tool, which offer much more functionality to tune an axis.



- The presented information in this "Getting started" is based on general rules and values, but may not be correct in specific cases. Always consult the motor specification datasheet and take external factors in consideration when specifying values for certain settings. For example, the acceleration of a motor depends on the load on the axis.
- Some of the steps to configure an axis specify a setting. These setting are only indicative values. See [chapter 3 "NYCeConfigurator" on page 59](#) for a detailed explanation of the fields.
- When you use the NYCeConfigurator, always click the "Apply" button before you select another tree item (node, drive, axis, IndraDrive) otherwise any change that you made is not applied.

2.5.2 Create an axis

Start the NYCeConfigurator to create a new axis.

Do the following steps to create a new axis for the NY4120 in NY4110_node1.

1. Click on the "Slot 1.0: NY4120" element in the NYCe 4000 system tree, displayed in the pane at the left side of the NYCeConfigurator, see [fig. 2-18 "NYCeConfigurator window" on page 32](#).
2. Click the "New" button in the pane at the right side. A new axis named DEF_AXIS_1 is created and displayed in the NYCe 4000 system tree as a child of "Slot 1.0".

Getting started

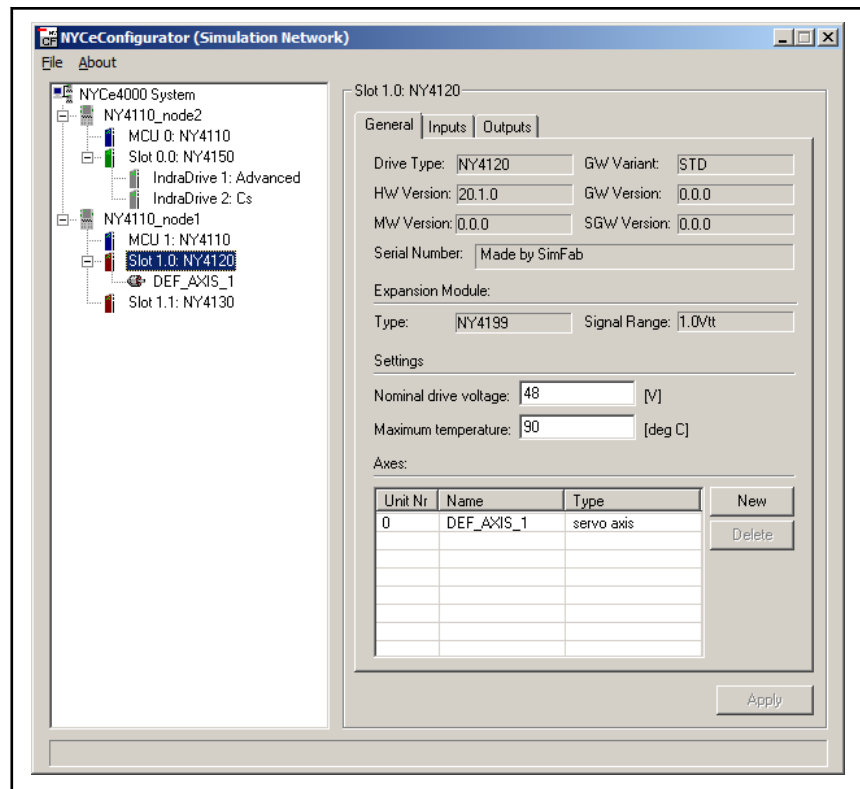


Fig. 2-18: NYCeConfigurator window



If you create a new axis or rename an existing axis name, you must save the node configuration to make the change persistent. To save the node configuration, select the node, click the "Save / Download" tab, and click the button "Save to File" or "Save to Flash".

2.5.3 Configure the axis for a DC, BLAC, BLDC motor, or external drive amplifier

Click on the "DEF_AXIS_1" axis in the pane at the left side of the NYCeConfigurator. You can change the name of the axis in the "General" tab, see [fig. 2-19 "New axis selected in the NYCeConfigurator" on page 33](#). Enter the new name for the axis in the "Axis Name" entry field and click the "Apply" button, if no more changes are needed on this tab.

In the following walkthrough the axis was renamed to "MyAxis".

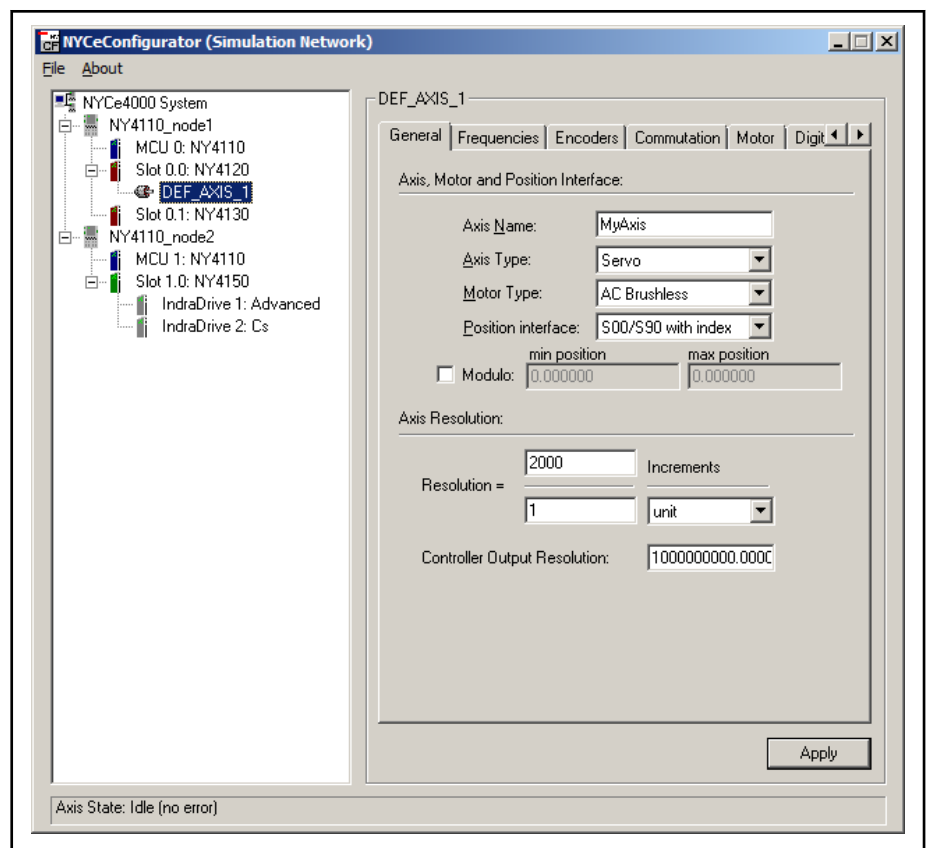


Fig. 2-19: New axis selected in the NYCeConfigurator

Do the following steps in the "General" tab to setup a rotational axis with revolution as axis position unit.

1. Set the "Axis Type" to Servo, if this field shows another selection.
2. Select the "Motor Type". The listed motor types depend on the chosen drive module. The types available are DC Brushed, AC Brushless, DC Brushless, and External (on AnOut0). The motor type IndraBLAC is only available with the NY4150 SERCOS III Master module.
3. Select the "Position interface". Possible choices are Analog in, S0S90 with index, S0S90 without index, MSM absolute, MSM incremental, and EnDat2.2. The S0S90 encoder is also known as A/B encoder, quadrature encoder and 3-channel digital incremental encoder.

If the drive module has the optional NY4199 SinCos Option module installed, additional encoder types SinCos, EnDat2.1 and Hiperface can be selected.

4. Set the "Resolution" equal to the encoder resolution.

Example: If the encoder connected to the motor shaft has a quadrature 512 cycles per revolution (CPR) output, then the resolution of the encoder must be set to 2048 increments per revolution.

This is also known as the Pulses Per Revolution of an encoder (PPR).

5. Set axis unit "Increments" to "revolution".
6. Set the "Controller Output Resolution" equal to the maximum drive current or voltage that the drive module can generate. For the NY4120 and NY4140 the value is 20.0, for the NY4125 the value depends on the drive power supply voltage level, and for the NY4130 the value is 24.0.

Getting started

The default setting is a high value to prevent damage to the motor caused by a high current or voltage. For more information see [chapter 3.7 "Configuration at the axis level"](#) on page 82.

7. Click the "Frequencies" tab.
8. Set the frequencies as shown in [fig. 2-20 "Axis frequency settings"](#) on page 34. Note that if you have a small motor, the Pulse Width Modulation Frequency possibly should be chosen higher to prevent instability.

For the NY4130 drive module only the "Setpoint (Node) Frequency" and "Position Velocity Loop Frequency" fields are available.

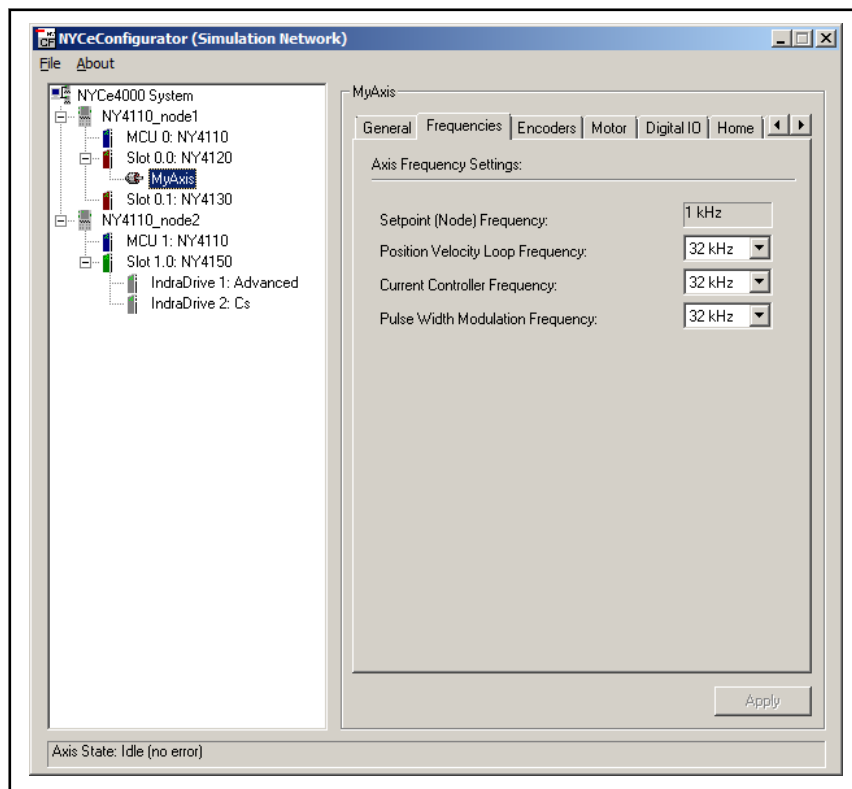


Fig. 2-20: Axis frequency settings

9. Depending on the selected position interface, you have to specify information about the encoder. In this example, the selected position interface is an S0S90 encoder. On the "Encoders" tab is for this encoder one option that must be set according to the hardware configuration of the encoder. Normally, the S0S90 will be connected hardware-wise using differential signals, but if you connect the encoder hardware-wise single ended, you must put a check mark in the "Single Ended" check box, see [fig. 2-21 "Position interface settings"](#) on page 35. See [chapter 3.7 "Configuration at the axis level"](#) on page 82 for more information about the different encoder settings.

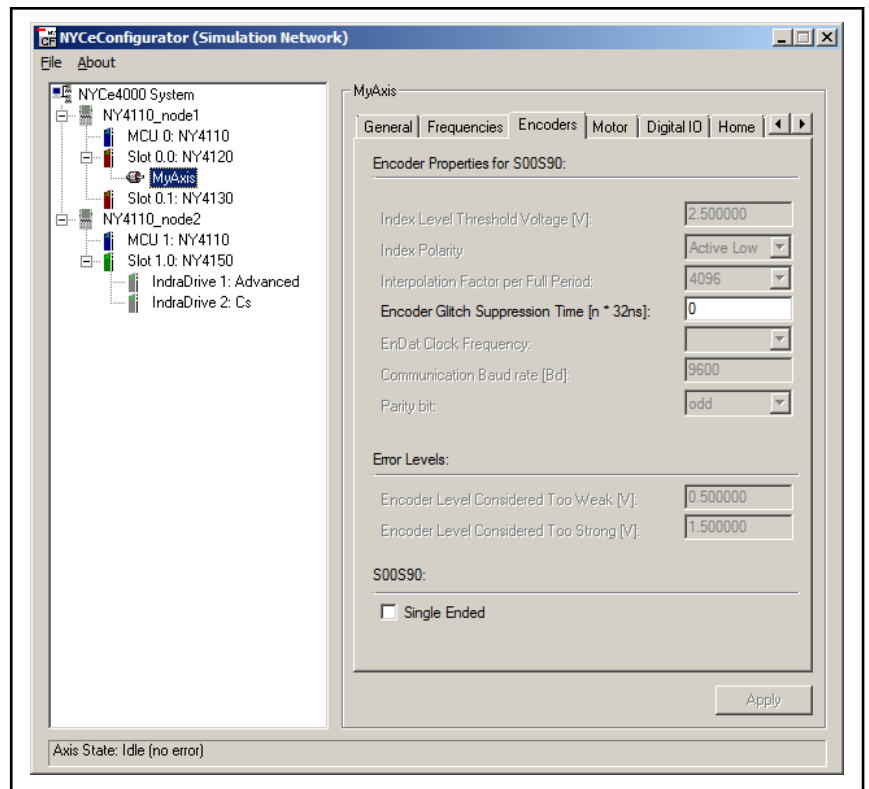


Fig. 2-21: Position interface settings

10. If you selected a BLAC or BLDC motor in step 2, there is also a tab called "Commutation". Click the "Commutation" tab.
 - For a BLAC motor, get the number of pole pairs from the motor manufacturer datasheet and set the "Pos units / Total nr. of Pole Pairs" to 1, as the position unit has been set to revolutions.
 - For a BLDC motor, select the Hall sensor input type (DIGIN or RS422), and get the commutation table from the manufacturer datasheet. If the commutation table is unknown, you can use the initial table shown by the NYCeConfigurator.
11. Click the "Motor" tab.
12. Set Saturation level of the axis equal to the peak current of the motor if you use an NY4120 or NY4140 drive module. If you use an NY4125 or NY4130 drive module, the saturation level is the maximum voltage for the controller output of the drive module. Note that the saturation level depends on the scale factor chosen for Controller Output Resolution.
13. Click the "Error Bounds" tab, set the following fields.
 - Set servo bus voltage "High pos" to 120% of the nominal voltage.
 - Set servo bus voltage "Low pos" to 80% of the nominal voltage.



These settings are based on general rules and values, but may not be correct in specific cases. Always consult the motor specification datasheet and take external factors in consideration when specifying values for certain settings. See [chapter 3 "NYCeConfigurator" on page 59](#) for a detailed explanation of the fields.

Getting started

14. Click the "Error Handlers" tab, set the following fields.
 - Set smooth stop acceleration equal to 10x the nominal speed.
 - Set smooth stop jerk equal to 100x the nominal speed.
 - Set quick stop acceleration equal to 10x the nominal speed.
15. Click the "Apply" button.
16. Click the "Save / Download" tab.
17. Save the axis settings to a file, for example "MyAxis.xml".



If you create a new axis or rename an existing axis name, you must save the node configuration to make the change persistent. To save the node settings, select the node, click the "Save / Download" tab, and click the button "Save to File" or "Save to Flash". If you do not save the node configuration, the created (or changed) axis name is lost after a reset or power down.

2.5.4 Configure the axis for a stepper motor or sensing stepper motor

Click on the "DEF_AXIS_1" axis in the pane at the left side of the NYCeConfigurator. You can change the name of the axis in the "General" tab. Enter the new name for the axis in the "Axis Name" entry field and click the "Apply" button, if no more changes are needed on this tab.

In the following walkthrough the axis was renamed to "MyAxis".

Note that the position encoder of a sensing stepper motor is not used by the control loop. It can only be used to read the actual axis position.

Do the following steps in the "General" tab to setup a rotational axis with revolution as axis position unit.

1. Set the "Axis Type" to "Stepper" or "Sensing Stepper". If you select "Sensing Stepper", you must also define the used position interface.
2. Set the "Resolution" equal to the number of (full) steps per revolution of the stepper motor (consult the motor manual).
3. Set axis unit "Increments" to "revolution".
4. Set the "Controller Output Resolution" equal to the maximum drive current that the NY4120 drive module can generate. The maximum drive current is 20A. The default setting is a high value to prevent damage to the motor caused by a high current. For more information see [chapter 3.7 "Configuration at the axis level" on page 82](#).
5. Click the "Commutation" tab.
6. Set the number of micro-steps per step equal to 1.
7. Click the "Motor" tab.
8. Set the saturation level equal to the rated current of the stepper motor (consult the motor manual).
9. Click the "Error Bounds" tab, set the following fields.
 - Set servo bus voltage "High pos" to 120% of the nominal voltage.
 - Set servo bus voltage "Low pos" to 80% of the nominal voltage.



These settings are based on general rules and values, but may not be correct in specific cases. Always consult the motor specification datasheet and take external factors in consideration when specifying values for certain settings. See [chapter 3 "NYCeConfigurator" on page 59](#) for a detailed explanation of the fields.

10. Click the "Error Handlers" tab, set the following fields.
 - Smooth stop acceleration equal to 20.
 - Smooth stop jerk equal to -1 (disables the jerk for the smooth stop).
 - Quick stop acceleration equal to 20.
 11. Click the "Apply" button.
 12. Click the "Save / Download" tab.
 13. Save the axis settings to a file, for example "MyAxis.xml".
-



If you create a new axis or rename an existing axis name, you must save the node configuration to make the change persistent. To save the node settings, select the node, click the "Save / Download" tab, and click the button "Save to File" or "Save to Flash". If you do not save the node configuration, the created (or changed) axis name is lost after a reset or power down.

2.5.5 Configure the axis for an external stepper drive motor

Click on the "DEF_AXIS_1" axis in the pane at the left side of the NYCeConfigurator. You can change the name of the axis in the "General" tab. Enter the new name for the axis in the "Axis Name" entry field and click the "Apply" button, if no more changes are needed on this tab.

In the following walkthrough the axis was renamed to "MyExtStepper".



Note that the "pulse" signal of the external stepper drive must be connected to fast output 0, and the "direction" signal of the external stepper drive must be connected to fast output 1 of the module. Make sure that these outputs are not used by some other functionality.

Do the following steps in the "General" tab to setup a rotational axis with revolution as axis position unit.

1. Set the "Axis Type" to "Stepper".
2. Set the "Motor Type" to "External Stepper".
3. Set the "Resolution" equal to the number of (full) steps per revolution of the stepper motor (consult the motor manual).
4. Set axis unit "Increments" to "revolution".
5. Click the "External Stepper" tab.

Getting started

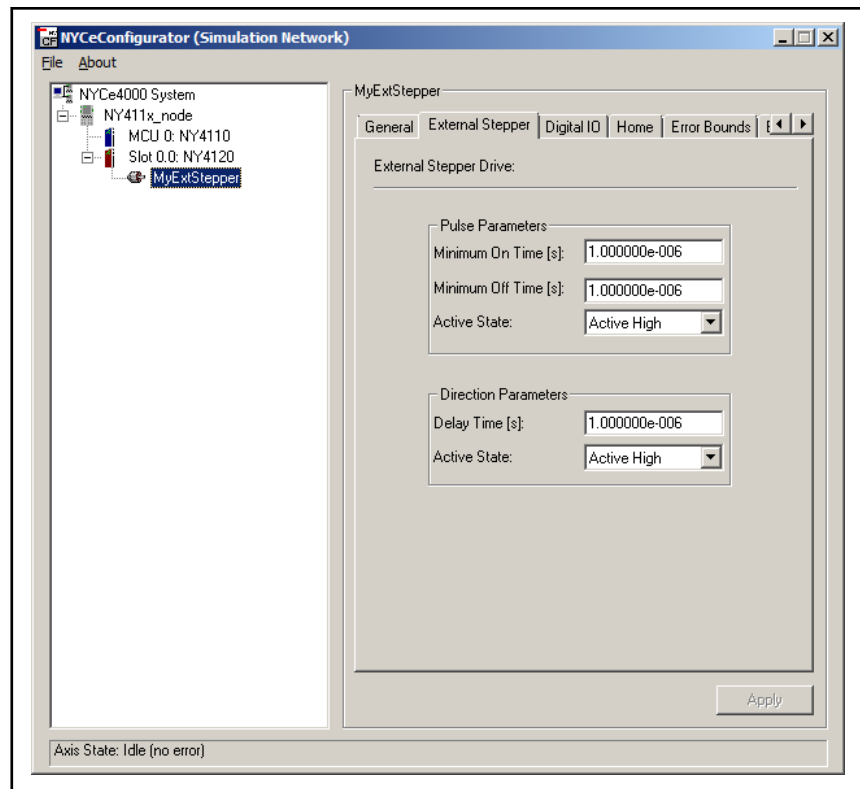


Fig. 2-22: External stepper signals

Enter appropriate values for the pulse signal parameters (minimum on and off time and active state), and the direction signal parameters (delay time and active state).

6. Click the Digital I/O tab.

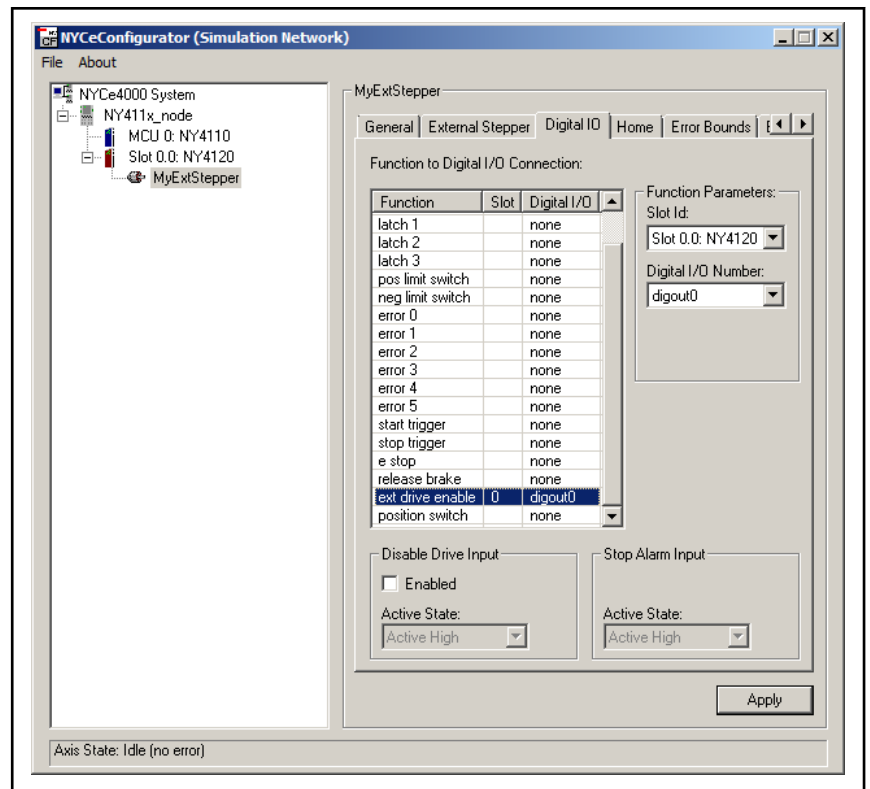


Fig. 2-23: External drive enable signal

If applicable, define the digital output assigned to switch on and off the power supply of the external stepper drive.

7. Click the "Error Handlers" tab, set the following fields.
 - Smooth stop acceleration equal to 20.
 - Smooth stop jerk equal to -1 (disables the jerk for the smooth stop).
 - Quick stop acceleration equal to 20.
8. Click the "Apply" button.
9. Click the "Save / Download" tab.
10. Save the axis settings to a file, for example "MyExtStepper.xml".



If you create a new axis or rename an existing axis name, you must save the node configuration to make the change persistent. To save the node settings, select the node, click the "Save / Download" tab, and click the button "Save to File" or "Save to Flash". If you do not save the node configuration, the created (or changed) axis name is lost after a reset or power down.

2.5.6 Configure the axis for an IndraDrive controlled motor

The NYCeConfigurator displays the configuration with the NY4150 in the pane at the left hand side. Click on the "IndraDrive 1" axis in the pane at the left side of the NYCeConfigurator. The number "1" is the unit number which is the same as the SERCOS address of the IndraDrive. The SERCOS address

Getting started

can be any number from 1 to 99 (inclusive). You can change the SERCOS address through the panel on the IndraDrive C or IndraDrive Cs. Changing the SERCOS address with the NYCe 4000 software or NYCe 4000 tools is not supported.



1. The SERCOS address must be unique in the SERCOS network.
2. If you change the SERCOS address on an IndraDrive, you must initialize the SERCOS network again to force an update of the administration in the node and the host.
3. If the SERCOS network was broken (for example by disconnecting and reconnecting a SERCOS network cable or resetting an IndraDrive in the SERCOS network), the SERCOS network will not start if not all axes on all IndraDrives in this SERCOS network are shut down first, to prevent potentially dangerous situations.

Click on the "DEF_AXIS_1" axis in the pane at the left side of the NYCeConfigurator. You can change the name of the axis in the "General" tab, see [fig. 2-24 "New IndraDrive "MyIndraAxis" selected in the NYCeConfigurator" on page 40](#). Enter the new name for the axis in the "Axis Name" entry field and click the "Apply" button, if no more changes are needed on this tab.

In the following walkthrough the axis was renamed to "MyIndraAxis".

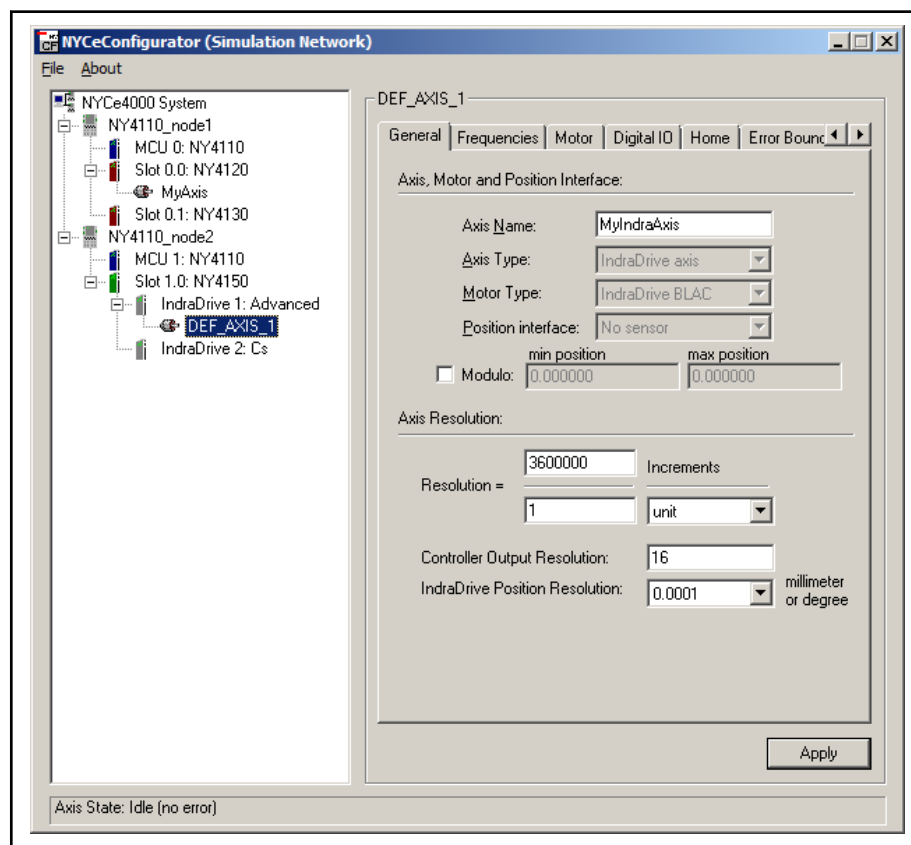


Fig. 2-24: New IndraDrive "MyIndraAxis" selected in the NYCeConfigurator

The fields "Axis Type", "Motor Type" and "Position interface" are predefined with an IndraDrive, and cannot be changed.

Getting started

Do the following steps in the "General" tab to setup a rotational axis with revolution as axis position unit. See [fig. 2-24 "New IndraDrive "MyIndraAxis" selected in the NYCeConfigurator"](#) on page 40 for typical values.

1. Set the "IndraDrive Position Resolution" and "Resolution". You can choose 0.001, 0.0001, 0.00001, or 0.000001 degree for rotational encoders or millimeter for linear encoders from the drop-down selection box. For example, if you select 0.0001 as "IndraDrive Position Resolution" and the position unit is one revolution, you enter $360 \text{ degrees} / 0.0001 = 3600000$ in the field "Resolution".
2. Set axis unit "Increments" to "revolution".
3. Set the "Controller Output Resolution" to the specified peak current of the type of IndraDrive connected to the NY4150. Note that IndraDrives specify the nominal current, not the peak current. The "Controller Output Resolution" is a scale factor of the controller output. Internally, the controller output uses a range of -1.0 ... +1.0. With the "Controller Output Resolution" you can scale this range to any other range.
4. Click the "Motor" tab.
5. Set "Saturation Level [Cout]" of the axis lower than or equal to the minimum of peak current of the motor or peak current of the IndraDrive. This is the same value that you entered in the "General" tab in "Controller Output Resolution" field.
6. Click the "Error Handlers" tab, set the following values.
 - Set the smooth stop acceleration equal to 10x the nominal speed.
 - Set the smooth stop jerk equal to 100x the nominal speed.
 - Set the quick stop acceleration equal to 10x the nominal speed.



These settings are based on general rules and values, but may not be correct in specific cases. Always consult the motor specification datasheet and take external factors in consideration when specifying values for certain settings. See [chapter 3 "NYCeConfigurator" on page 59](#) for a detailed explanation of the fields.

Getting started

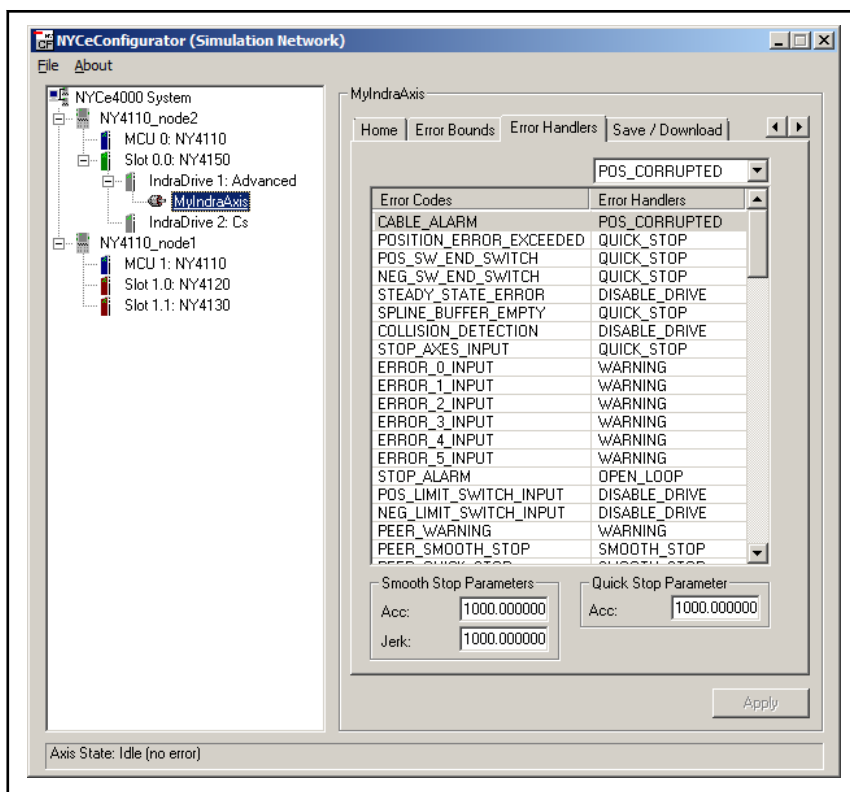


Fig. 2-25: Axis Error Handlers settings

7. Click the "Apply" button.

- The entered values may change. This is caused by the arithmetic used to convert the value to a parameter value sent to the IndraDrive.
- NYCeConfigurator may display a warning message, because several parameters are set automatically to correct default values.

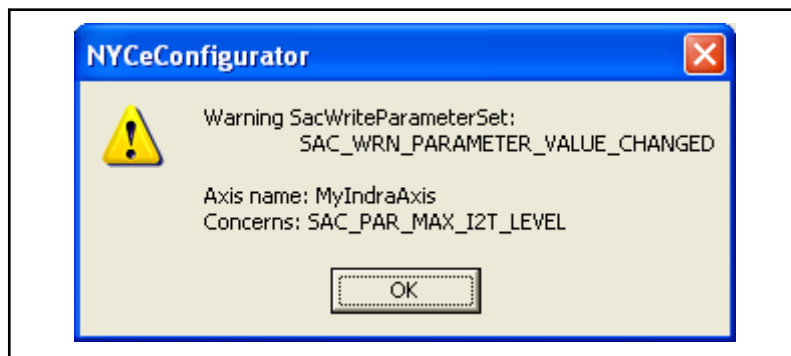


Fig. 2-26: NYCeConfigurator warning message

Click the "OK" button.

8. Click the "Save / Download" tab.

9. Save the axis settings to a file, for example "MyIndraAxis.xml".



If you create a new axis or rename an existing axis name, you must save the node configuration to make the change persistent. To save the node settings, select the node, click the "Save / Download" tab, and click the button "Save to File" or "Save to Flash". If you do not save the node configuration, the created (or changed) axis name is lost after a reset or power down.

2.6 Quick start

2.6.1 Introduction

This chapter describes the steps to get an operational axis with a motor using the NYCeTuner. The NYCeTuner has a so-called "wizard" mode that guides you through the configuration of a servo based axis. The wizard shows, graphically supported, the procedure step by step by going through the tabs of the NYCeTuner from left to right. It is important that you follow the described sequence.

The flow diagram in [fig. 2-27 "Sequence of steps to get an operational axis using NYCeTuner" on page 44](#) shows the steps you must do to get an axis of each type of motor operational. The names of the steps in the flow diagram are identical to the tabs of the NYCeTuner.

The "wizard" mode is not supported for external stepper drive motors and the IndraDrive. See the end of this chapter for a short description to get an IndraDrive-based axis operational.

Getting started

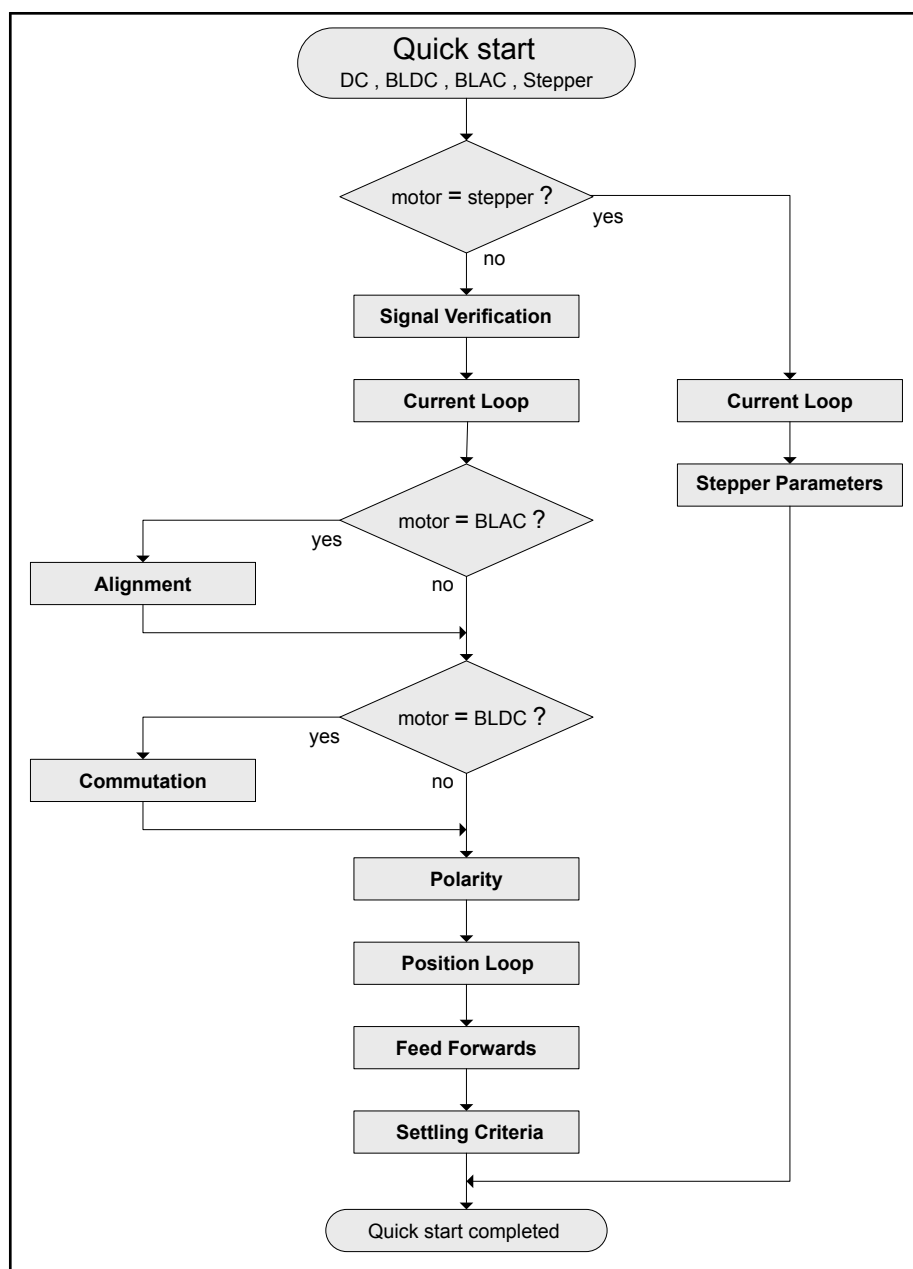


Fig. 2-27: Sequence of steps to get an operational axis using NYCeTuner

When you start the NYCeTuner the first time after the installation of the NYCe 4000 software, the NYCeTuner always starts in the "wizard" mode. When the NYCeTuner is closed the mode of operation is remembered and is started automatically the next time the NYCeTuner is started. The NYCeTuner also has a so-called "expert" mode which allows you to fine tune many parameters, see for detailed information [chapter 4 "NYCeTuner" on page 115](#).

After you start the NYCeTuner in "wizard" mode, the "Welcome" tab is opened. The information displayed in the right hand pane depends on the type of motor selected for the axis. When you select an axis in the left hand pane, the "Initialize Axis" dialog box may appear to initialize the selected axis. You can load the axis parameters from a previously saved axis XML file or from the flash memory.

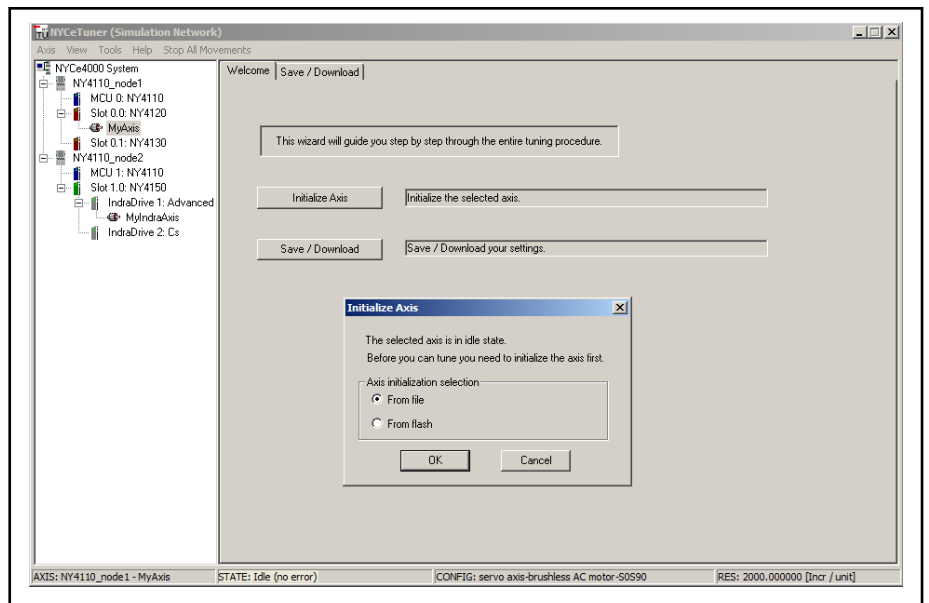


Fig. 2-28: NYCeTuner "Initialize Axis" dialog box

The "Initialize Axis" dialog box also appears when you click the "Initialize Axis" button.

After you initialized the selected axis, the "Welcome" tab shows the buttons which represent the configuration steps needed for the motor type of the selected axis.

In fig. 2-29 "NYCeTuner "Welcome" tab for a BLAC motor" on page 45 the motor of the selected axis is a BLAC motor, and the 8 buttons, shown from top to bottom, represent the configuration steps for a BLAC motor. The tabs next to the "Welcome" tab access the same tabs that you can open with the buttons on the "Welcome" tab.

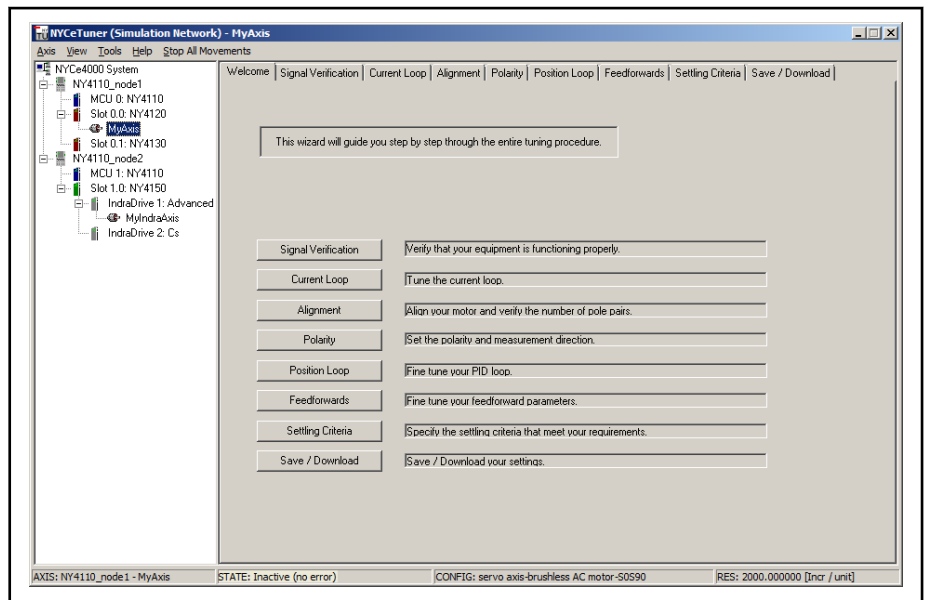


Fig. 2-29: NYCeTuner "Welcome" tab for a BLAC motor

The available tabs depend on the motor type configured for the selected axis. See fig. 2-30 "Examples of NYCeTuner tabs for the supported motor types"

Getting started

on page 46 for the tabs of the NYCeTuner for these motor types. ① are the tabs for a BLAC motor, ② are the tabs for a BLDC motor, ③ are the tabs for a DC motor, ④ are the tabs for a stepper motor, and ⑤ are the tabs for an external stepper motor drive.

Welcome	Signal Verification	Current Loop	Alignment	Polarity	Position Loop	Feedforwards	Settling Criteria	Save / Download	①
Welcome	Signal Verification	Current Loop	Commutation	Polarity	Position Loop	Feedforwards	Settling Criteria	Save / Download	②
Welcome	Signal Verification	Current Loop	Polarity	Position Loop	Feedforwards	Settling Criteria	Save / Download		③
Welcome	Current Loop	Stepper Parameters	Save / Download						④
Welcome	Stepper Parameters	Save / Download							⑤

Fig. 2-30: Examples of NYCeTuner tabs for the supported motor types

The tabs are discussed in the following chapters in the sequence of the quick start procedure. Only the "Save / Download" tab is not discussed in this chapter, see chapter 4.4 "Saving and shutting down" on page 116.

Select and initialize an axis

The NYCeTuner displays in the pane at the left side a tree with the nodes and the axes in the network. Click on an axis in the pane at the left side to do the configuration of that axis. You can also use the menu bar, click "Axis → Select".

If needed, you can load the parameters from an axis parameter file on the host, click "Axis → Initialize" in the menu bar. You have 2 possible locations from where you can load an axis parameter file.

1. "initialize from file"

load parameters from an axis parameter file on the host.

2. "initialize from flash"

load parameters from the flash memory of the node in real mode, or from the .FLS and .EEP files in simulation mode.

Sliders

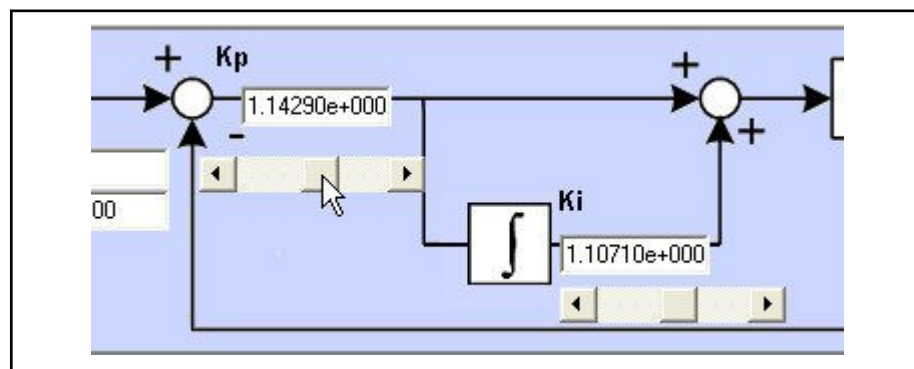


Fig. 2-31: Example of two sliders

Some values in the tabs can be edited through an edit box and a slider. The center value of the slider is determined by the value of the corresponding edit box. To change the center value of the slider, enter a new value in the edit box and press "Enter" on the keyboard.

Motion settings

In various tabs, you will set a motion profile. This is done in the "Motion settings" pane. In this tab, you initiate movements of the axis to check whether the entered values are good enough or need further adjustment.

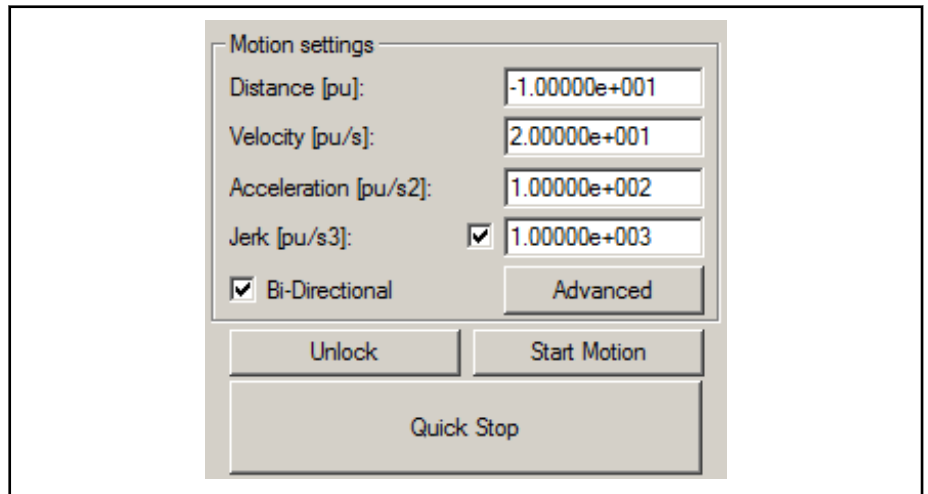


Fig. 2-32: "Motion settings" pane

Explanation of the objects in the "Motion settings" pane

- Distance
This field specifies the distance the motor will move. This is especially important for a linear motor, because you need to make sure it stays within its bounds.
- Velocity
This field specifies the velocity the motor will try to reach during the movement. If the acceleration is not high enough, the speed may not be reached before the motor must decelerate again.
- Acceleration
This field specifies the acceleration the motor will try to reach during the movement. If the acceleration is too low, the specified speed may not be reached. Also, if the specified jerk value is too low, the acceleration specified may not be reached.
- Jerk
This field specifies the jerk the motor will have during the movement. Jerk is the acceleration of the acceleration, or the derivative of the acceleration. The jerk is always instantly the specified value. It is not uncommon to set the jerk to its maximum value to simulate infinite jerk. You can do this by putting or removing the check mark in the checkbox.
- Bi-Directional
If the "Bi-Directional" checkbox has a check mark, the distance will be inverted at the end of a movement. Then you can move up and down between the boundaries. This option is useful if your motor cannot move beyond certain boundaries.
- Advanced
If you click the "Advanced" button, a Point-to-point dialog is displayed with more extensive options for your motion (for example auto move and typed profile calculation). This dialog is the same as the dialog in the

Getting started

expert mode of the NYCeTuner using the "Edit" button. See [chapter 4.15 "NYCeTuner motion options" on page 160](#) for a description of the Point to Point motion editor. When you click the "OK", the settings from that window are copied onto the text fields in the "Motion settings" tab.

- Lock / Unlock

This button puts the axis in the READY state, which is the correct state to start a motion. It is also used to take the motor out of the READY state again, back to the INACTIVE state where there is no power on your motor. The text in the button changes to reflect the action.

- Start Motion / Finish Motion

This button starts or stops a motion. The motor must be in the READY state before this button is enabled. Once a motion has been started, the description text in the button changes to "Finish Motion". The motion is stopped when you click the button, but the motion is finished first. For example, when you are auto-moving and you click this button, the current movement is finished first.

- Quick Stop

This button stops the motor immediately. You can use this button if for some reason the motor must be stopped in a controlled, but as fast as possible manner.

2.6.2 Signal Verification (not for stepper motors)

In the "Signal Verification" tab you can move the axis a number of position units and check how many positions the axis has actually moved. This does not apply for stepper motors, but it does apply for sensing stepper motors.

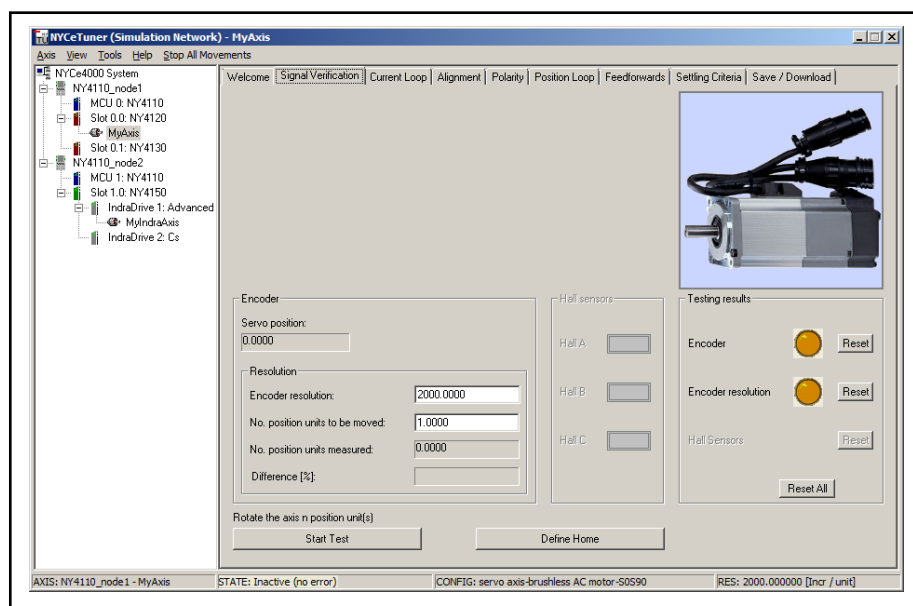


Fig. 2-33: NYCeTuner "Signal Verification" tab

Do the following steps.

1. Enter the resolution of the encoder in the "Encoder resolution" field. See the data sheet of the manufacturer of the encoder.
2. Enter the number of units to move in the "No. position units to be moved" field. The value depends on the ratio (the number of increments per unit, micrometer, ..., radian, revolution) of the data entered in the "Axis Resolution" section in the General tab of the NYCeConfigurator,

see chapter 2.5.3 "Configure the axis for a DC, BLAC, BLDC motor, or external drive amplifier" on page 32.

3. Click the "Start Test" button.

The text of the button changes to "Stop Test".

4. Rotate the axis manually for approximately the number of position units that you specified in the "No. position units to be moved" field.
5. If you have collected enough measurement data, click the "Stop Test" button.

The actual number of positions that the axis has moved is displayed in the "No. position units measured" field.

If the difference between the positions to be moved and the positions measured is larger than 10%, the encoder resolution is probably set to a wrong value. The percentage of difference is displayed and you can use this information to adjust the encoder resolution to the correct value.

The indicators in the "Testing results" pane at the right hand side show the status of the encoder and the encoder resolution. You can clear the indicators with the "Reset" button.

The "Define Home" button enables you to set the current actual position of the axis as its home position.

2.6.3 Current Loop

In the "Current Loop" tab, you can tune the current loop. A test signal is injected in the loop. You can observe the response and make adjustments to improve the loop behavior.

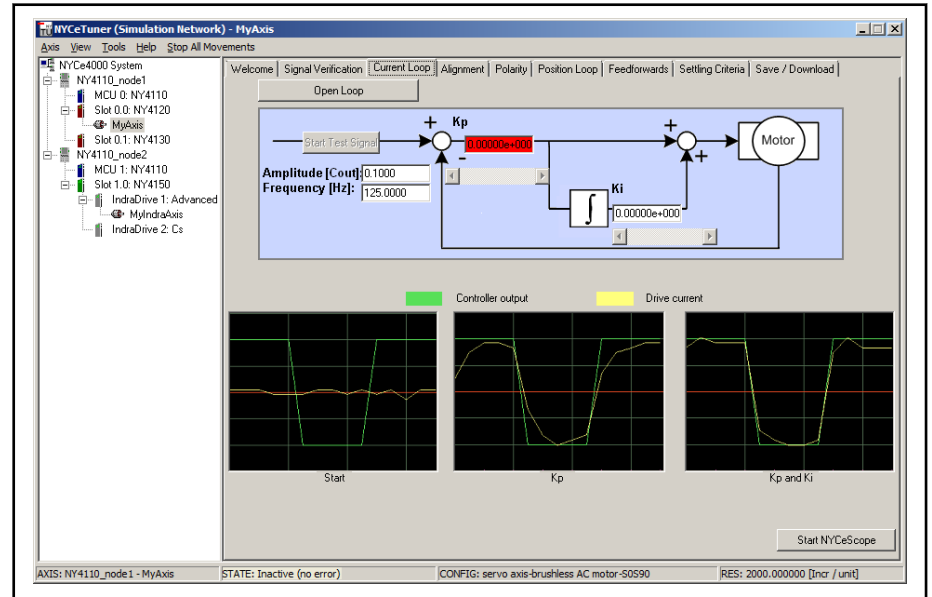


Fig. 2-34: NYCeTuner "Current Loop" tab

Do the following steps.

1. Enter an amplitude value for the test signal in the "Amplitude" edit box.
2. Enter a frequency for the test signal in the "Frequency" edit box.
3. Click the "Open Loop" button.
4. Click the "Start Test Signal" button.

Getting started

The text of the button changes to "Stop Test Signal".

5. Tune Kp (proportional gain) by entering a value or using the slider.
6. Tune Ki (integral gain) by entering a value or using the slider.
7. When the drive current (Iq for a stepper motor) equals the test signal (see the scope pictures included in the "Current Loop" tab), click the "Stop Test Signal" button.
8. To view the current characteristics of the axis, you can start the NYCeScope tool by clicking the "Start NYCeScope" button. See [chapter 5 "NYCeScope" on page 187](#) for more information about the NYCeScope tool.

2.6.4 Alignment (for BLAC motors only)

In the "Alignment" tab you can perform the alignment procedure for the motor. This tab is only available when you selected an axis with a BLAC motor.

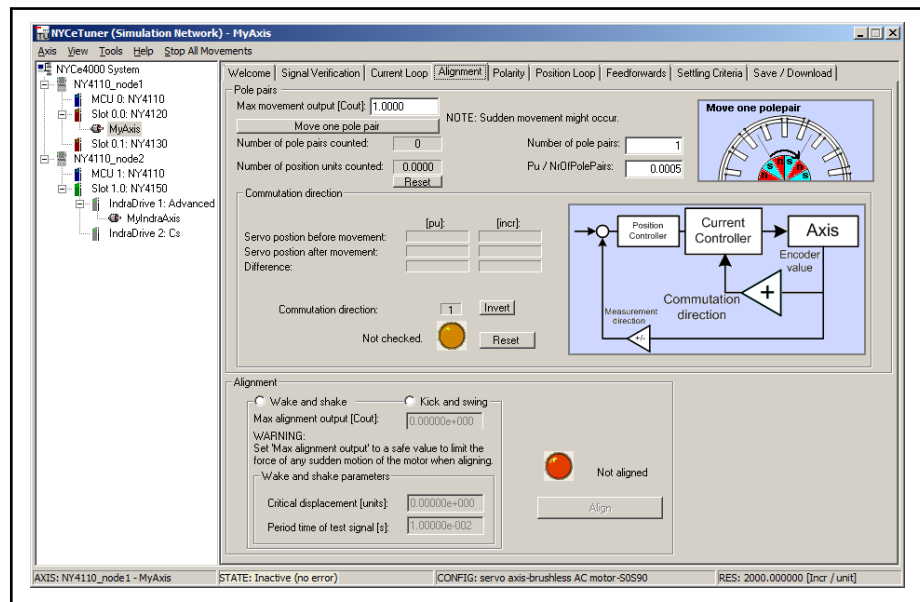


Fig. 2-35: NYCeTuner "Alignment" tab

Do the following steps.

1. Enter a value in the "Max movement output" field. The value determines the maximum value of the current which will be applied to move one pole pair.
2. If desired, enter values in the "Number of pole pairs" field and the "Pu / NrOfPolePairs" field to start with. These values should match the settings in the Commutation tab in NYCeConfigurator (see [chapter 3 "NYCeConfigurator" on page 59](#)).
3. Click the "Move one pole pair" button to move the motor one pole pair.
4. Based on the data displayed in the "Number of pole pair counted" and the "Number of position units counted" fields, you can verify the number of pole pairs of the motor.
5. You can also manually set the "Number of pole pairs" with the edit box.
6. After the button "Move one pole pair" is clicked, the change in servo position is measured and displayed as the difference in percentage. The difference should have the same sign (+/-) as the commutation

direction. If this is not the case, click the "Invert" button in the "Commutation direction" section to invert the commutation direction.

The test of commutation direction is reset every time the "Invert" or "Reset" button is clicked. The commutation direction must be correct before you continue to the alignment section in this tab. The indicator must be green and the text next to the indicator says "Correct".

7. Select the alignment mode that must be used to align the motor with the radio buttons in the "Alignment" section.
8. Enter the parameters specified for the selected alignment mode.
9. Click the "Align" button to align the motor in the selected mode.

The indicator and the text next to the indicator indicate whether the alignment was successful.

2.6.5 Commutation (for BLDC motors only)

In the "Commutation" tab you can measure and configure the settings of Hall sensors for the BLDC motor. This tab is only available when you selected an axis with a BLDC motor.

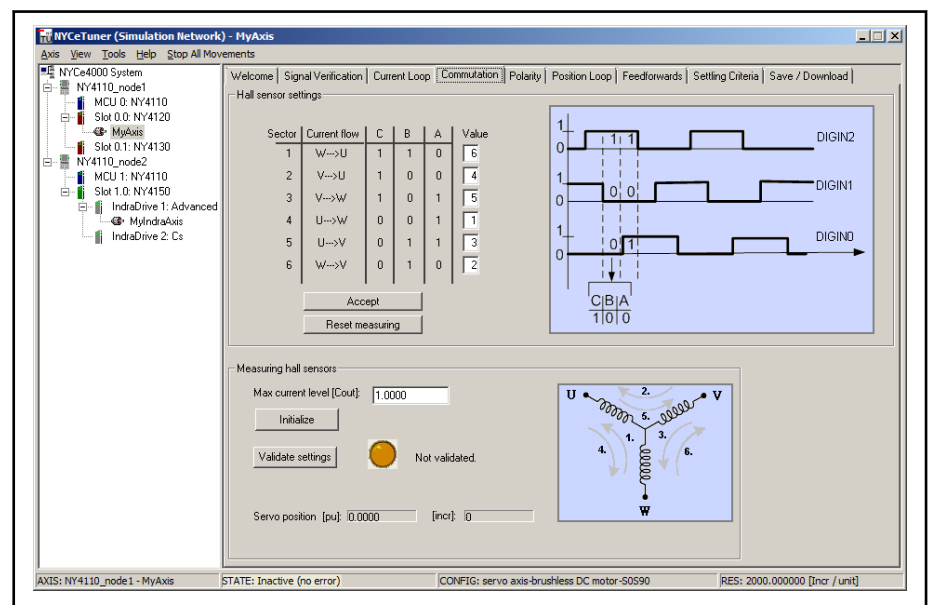


Fig. 2-36: NYCeTuner "Commutation" tab

Do the following steps.

1. Enter a value in the "Max current level" field. The value determines the maximum value of the current that will be applied to move one pole pair.
2. Click the "Align to: Step 1" button to measure the Hall sensors at Step 1.
 The measurement is represented by a rectangle box around the values at step 1. The Hall values of the step 1 are displayed bold to indicate that these values were measured and have not yet been saved.
3. Repeat the previous step for the other Hall sensors.
4. After the last measurement, click the "Finish" button.
5. Click the "Validate settings" button to collect all current Hall values in the table for a validation of the Hall sensors.

The indicator next to the button indicates whether the measured Hall settings are valid or not. If the settings are valid, the table should

Getting started

resemble the table in the "Commutation" tab of the NYCeConfigurator tool (see to [chapter 3 "NYCeConfigurator" on page 59](#)).

- Click the "Accept" button to save the Hall settings.

You can click the "Reset measuring" button to start a new measurement. The current Hall table settings from the configuration will be loaded into the table which is displayed in the "Hall sensor settings" section.

2.6.6 Stepper Parameters (for stepper motors only)

In the "Stepper Parameters" tab you can set the stepper motor settings. This tab is only available when you selected an axis with a stepper motor, sensing stepper motor or external stepper drive. Note that the position encoder of a sensing stepper motor is not used by the control loop. It can only be used to read the actual axis position.

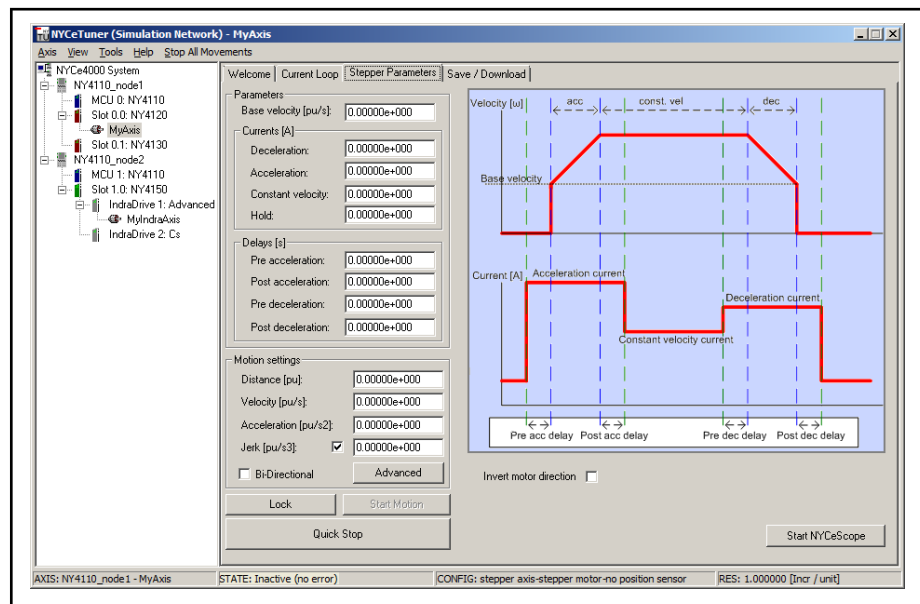


Fig. 2-37: NYCeTuner "Stepper Parameters" tab - stepper motor

The current and delay settings are not available for an external stepper drive. If the axis is an external stepper drive, those settings are disabled on the "Stepper Parameters" tab.

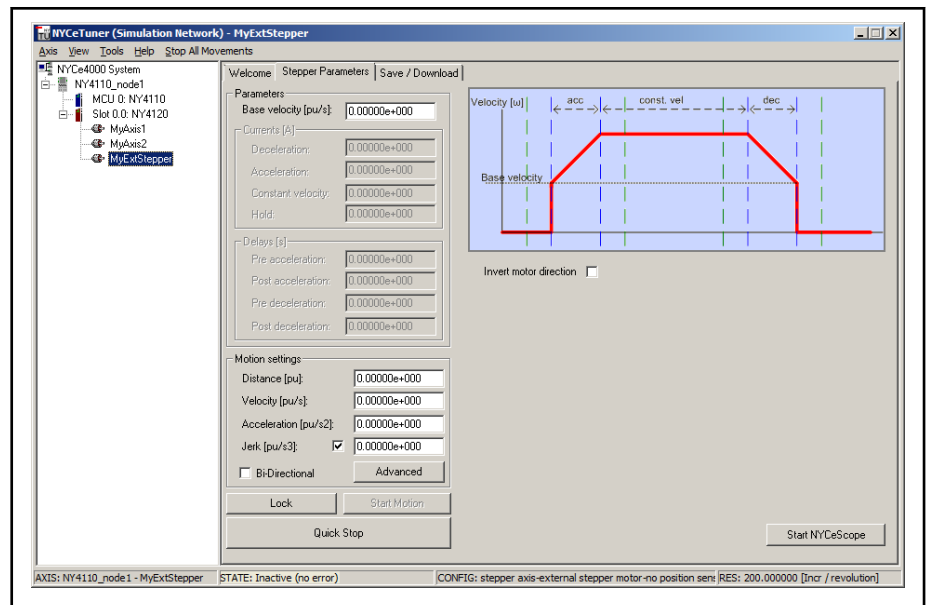


Fig. 2-38: NYCeTuner "Stepper Parameters" tab - external stepper drive

Do the following steps.

1. Enter or change stepper motor values.
2. If there is no motion profile, create a motion profile in the "Motion settings" section. Click the "Advanced" button to adjust additional settings of the profile.
3. If the axis is not running, click the "Lock" button and then the "Start" button.
4. Check whether the movement is correct. Do the following steps, if the movement is not correct.
 - Stop the motor.
 - If the rotation direction is not correct, put a check mark in the "Invert motor direction" checkbox.
 - If the movement is not correct, stop the motor, adjust the values.
 - Restart the movement.
5. Repeat the previous steps until the movement is correct.
6. To view the movements of the axis, you can start the NYCeScope tool by clicking on the "Start NYCeScope" button. See [chapter 5 "NYCeScope" on page 187](#) for more information about the NYCeScope tool.

2.6.7 Polarity

The "Polarity" tab enables you to set the polarity and the measurement direction.

Getting started

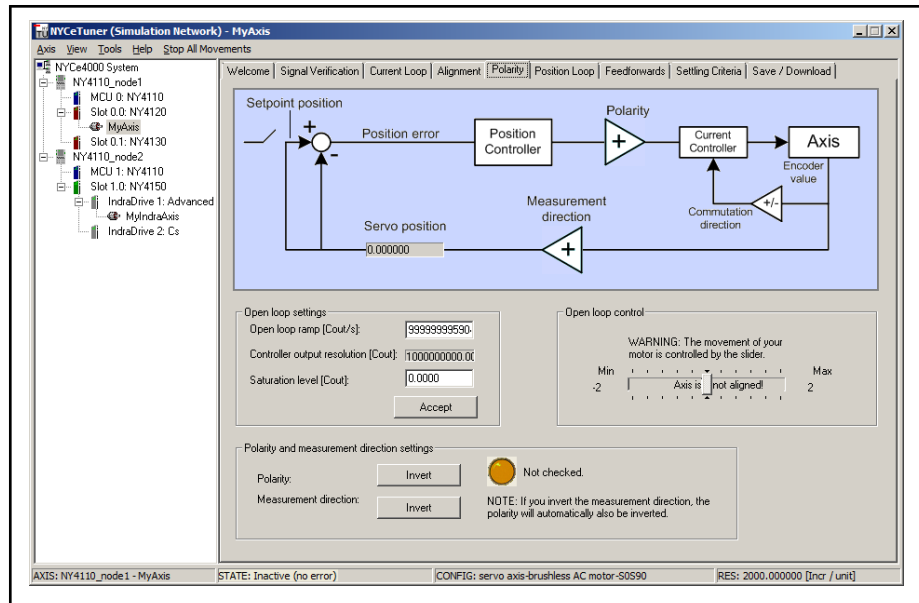


Fig. 2-39: NYCeTuner "Polarity" tab

Do the following steps.

1. Set the safe limit settings in the "Open loop settings" section and click the "Accept" button.
2. Use the slider in the "Open loop control" section to set the open loop control.
3. If the axis moves in an unexpected direction while you move the slider, click the "Invert" buttons to invert polarity and measurement direction.

If the polarity and measurement direction are correctly set, the "Not checked" indicator changes to green, and the text changes to "Correct".

2.6.8 Position Loop

In the "Position Loop" tab you compare the position at which the axis currently is and the position at which the axis should be.

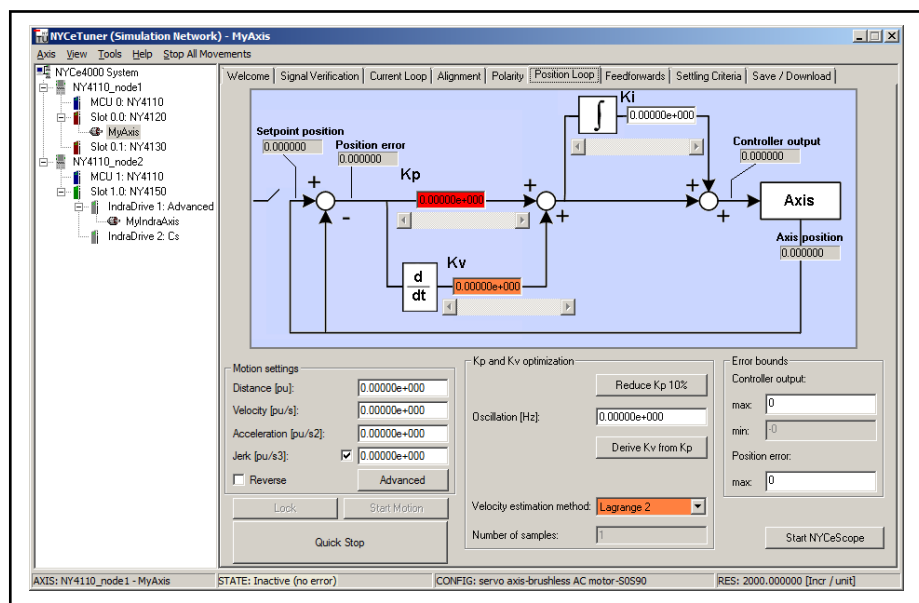


Fig. 2-40: NYCeTuner "Position Loop" tab

Do the following steps.

1. If there is no motion profile, create a motion profile in the "Motion settings" section. Click the "Advanced" button to adjust additional settings of the profile.
2. If the axis is not running, click the "Lock" button and then the "Start Motion" button.
3. Set the PID values (Kp, Ki and Kv) by entering a value or using the slider.
4. You can optimize Kp and Kv with the following steps.
 - Click the "Reduce Kp 10%" button to reduce the value of Kp by 10%.
 - Enter an oscillation frequency and click the "Derive Kv from Kp" button. This action will disable the Kv and Oscillation text field. Whenever you change Kp, Kv will automatically change accordingly.
 - Choose the velocity estimation method from the drop down box.
 - Enter a value for the number of samples.
5. Click the "Finish Motion" button to stop the axis when the adjustments are ready.
6. To view the movements of the axis, you can start the NYCeScope tool by clicking the "Start NYCeScope" button. See [chapter 5 "NYCeScope" on page 187](#) for more information about the NYCeScope tool.

2.6.9 Feedforwards

The feedforwards can compensate static forces (like gravity), coulomb frictions, viscous frictions and the acceleration forces (needed to accelerate the mass / inertia driven by the axis).

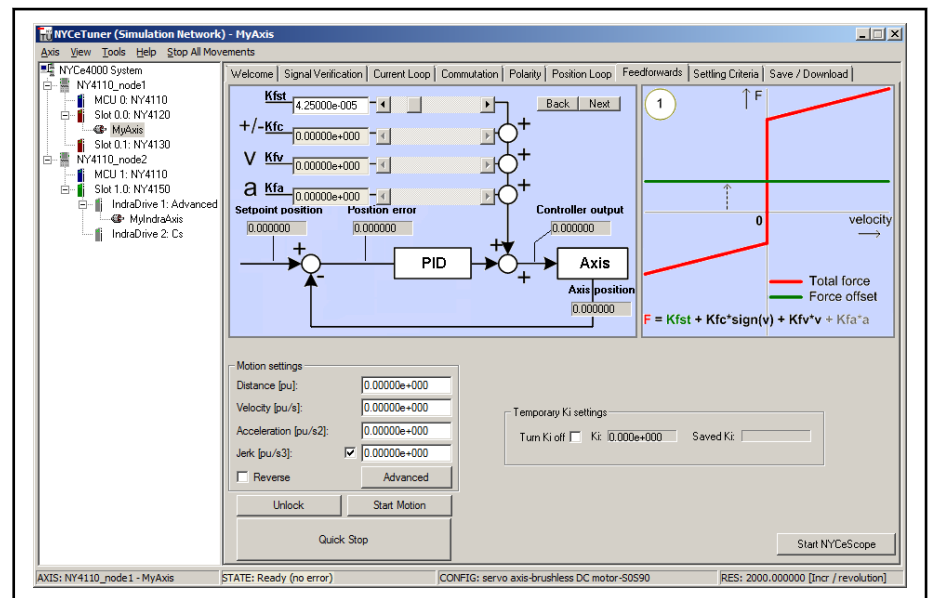


Fig. 2-41: NYCeTuner "Feedforwards" tab

Do the following steps.

Getting started

1. If there is no motion profile, create a motion profile in the "Motion settings" section. Click the "Advanced" button to adjust additional settings of the profile.
2. If the axis is not running, click the "Lock" button and then the "Start Motion" button.
3. Adjust the various feedforward parameters by entering a value or by using the slider. Use the "Back" and "Next" buttons to step through the parameters.
4. If necessary, you can turn off Ki temporarily by putting a check mark in the "Turn off Ki" checkbox.
5. Click the "Finish Motion" button to stop the axis when the adjustments are ready.
6. To view the movements of the axis, you can start the NYCeScope tool by clicking the "Start NYCeScope" button. See [chapter 5 "NYCeScope"](#) on page 187 for more information about the NYCeScope tool.

2.6.10 Settling Criteria

In the "Settings" tab you can set the settling criteria.

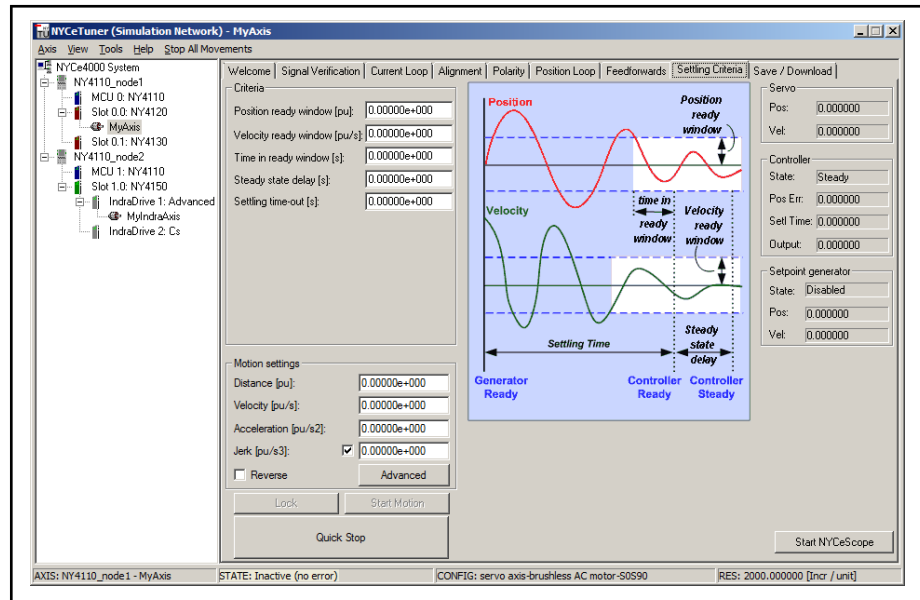


Fig. 2-42: NYCeTuner "Settling Criteria" tab

Do the following steps.

1. Make sure that the motor is not running.
2. If there is no motion profile, create a motion profile in the "Motion settings" section. Click the "Advanced" button to adjust additional settings of the profile.
3. Click the "Lock" button and then the "Start Motion" button to start the motor.
4. Enter the settling criteria in the "Criteria" section.
5. Click the "Finish Motion" button to stop the axis when the adjustments are ready.
6. To view the movements of the axis, you can start the NYCeScope tool by clicking the "Start NYCeScope" button. See [chapter 5 "NYCeScope"](#) on page 187 for more information about the NYCeScope tool.

2.6.11 Get an IndraDrive-based motor operational (NYCeTuner "Expert" mode)

The NYCe 4000 system supports IndraDrive-based motors with the NY4150 and NY4150/10 SERCOS III Master module. However, the current release of the NYCeTuner does not support the IndraDrive in the "Wizard" mode. This chapter describes the steps you must do after you configured the IndraDrive. See [chapter 2.5.6 "Configure the axis for an IndraDrive controlled motor" on page 39](#).

Do the following steps to get an IndraDrive-based motor operational.

1. Start the NYCeTuner tool.

NYCeTuner initializes the axis and will display an error message, because some parameters are initialized for the IndraDrive. Click "OK", you can ignore this error message.

2. Click "View → Expert", if the NYCeTuner tool starts in the "wizard" mode.

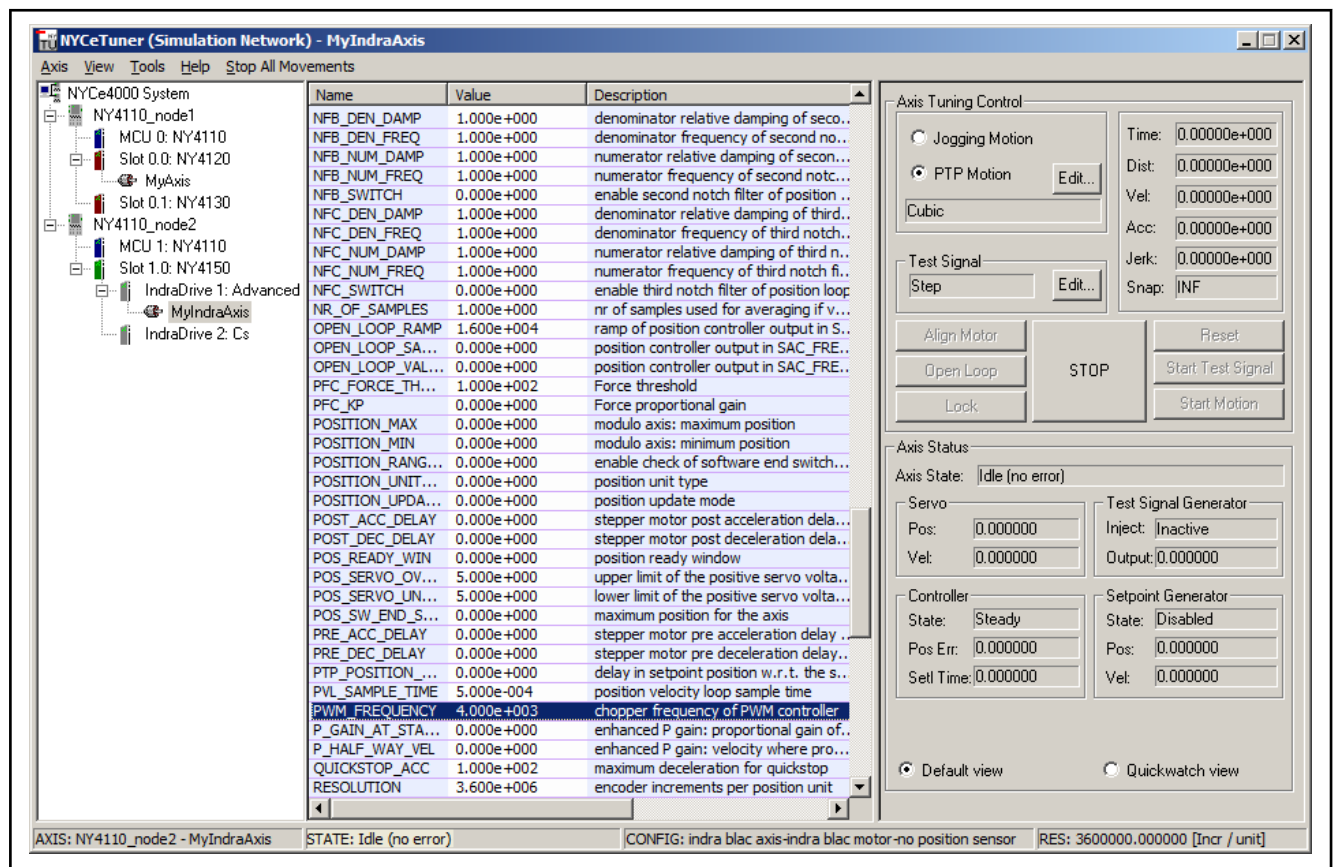


Fig. 2-43: NYCeTuner "Expert" mode, setting parameters for the IndraDrive-based motor

If the motor has motor encoder flash memory (Bosch Rexroth MSK motors), right-click on the axis. Select "Load default parameters from encoder" from the pop-up menu. The current control and Position Velocity Loop parameters will be loaded.

Getting started

If the motor does not have motor encoder flash memory, the parameters must be determined by the customer or taken from the data sheets of the manufacturer of the motor and set using the NYCeConfigurator.

3. Set the PWM frequency to 4000 Hz.
4. Click "Axis → Save to File..." to save the settings to a file.
5. Click the "Lock" button.
6. You can activate the motor, for example, select "Jogging Motion", click the "Edit..." button to set the jog parameters and click the "Start Motion" button.

The following SAC parameters are scaled to IndraDrive parameters at start-up of a NYCe 4000 node, if one or more IndraBLAC axes are present.

Parameter	NYCe 4000 firmware	IndraDrive rotary motor	IndraDrive linear motor
Torque / force	cout [-1 ... 1]	0.01 Nm	N
Position	increment	0.0001 °	0,0001 mm
Velocity	increment/s	0.0001 rev/min	0.001 mm/min
Acceleration	increment/s ²	0.001 rad/s ²	0.001 mm/s ²
Jerk	increment/s ³	0.001 rad/s ³	0.001 mm/s ³
Temperature	°C	0.1 °C	0.1 °C

Tab. 2-2: NYCe 4000 node start-up, default scaled parameters

3 NYCeConfigurator

3.1 Introduction

The NYCeConfigurator is designed to perform the following configuration options.

- Saving and restoring the settings of the axis, the nodes and the complete NYCe 4000 system
- Sample frequency, Service and Stop_Axes input settings (Node level)
- The configuration of virtual axes (MCU level)
- The configuration of the drives (Drive level)
 - Connected axes
 - Digital and analog I/O settings
- The configuration of real axes (Axis level)
 - Motor
 - Encoder
 - Frequencies
 - Digital I/O assignment
 - Homing
 - Error Bounds
 - Error handling

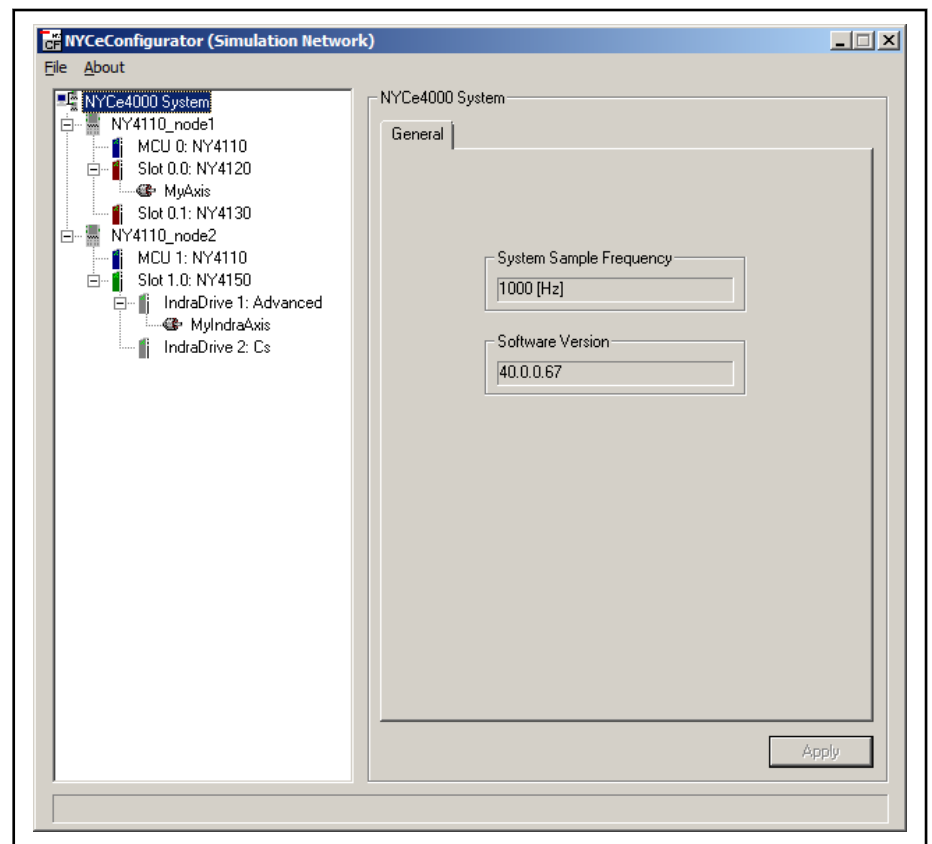


Fig. 3-1: The NYCeConfigurator window

NYCeConfigurator

fig. 3-1 "The NYCeConfigurator window" on page 59 shows the NYCeConfigurator window. In the pane at the left side of the NYCeConfigurator you select the component, and in the pane at the right side you can make adjustments for the selected item. In the bottom part of the NYCeConfigurator window is a line which displays status information if the selection is an axis of a node.

3.2 Parameter handling

As explained in chapter 2 "Getting started" on page 11, the parameter handling in the NYCeConfigurator is different compared to the parameter handling in the NYCeTuner. If a parameter setting has been changed in the NYCeConfigurator, this is only done in the NYCeConfigurator itself and not in the node. Only after you click the "Apply" or "OK" button, the new setting(s) will be sent to the node.

If you made changes to settings, but want to revert to the current settings, press the "Esc" key.

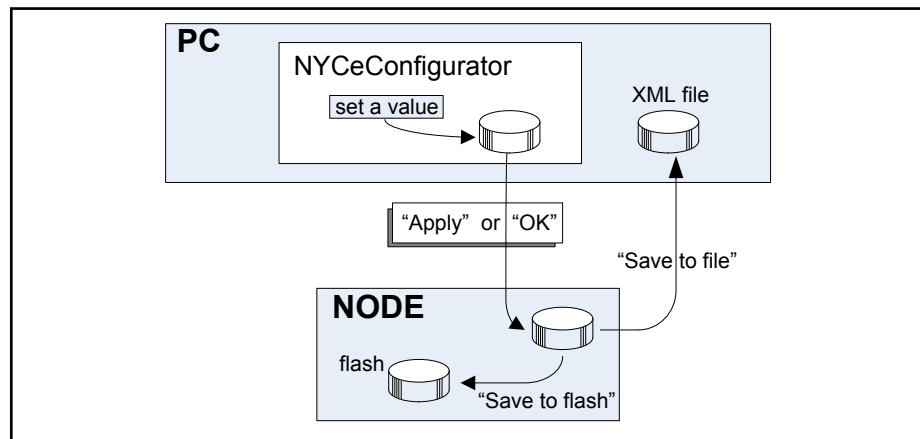


Fig. 3-2: Parameter handling using NYCeConfigurator

To apply certain configuration settings, the NYCeConfigurator requests for an axis shutdown. To be able to use the applied settings after a (re-)initialize, you must first save the settings to a file or flash memory after the shutdown of the axis is done.

3.3 Save and restore settings

When you work with the NYCeConfigurator, the changes that you make to the settings of the nodes and / or the axes can be saved at different levels. A message box appears to show the progress of the requested action(s), see fig. 3-3 "NYCeConfigurator requested action progress message box" on page 61.

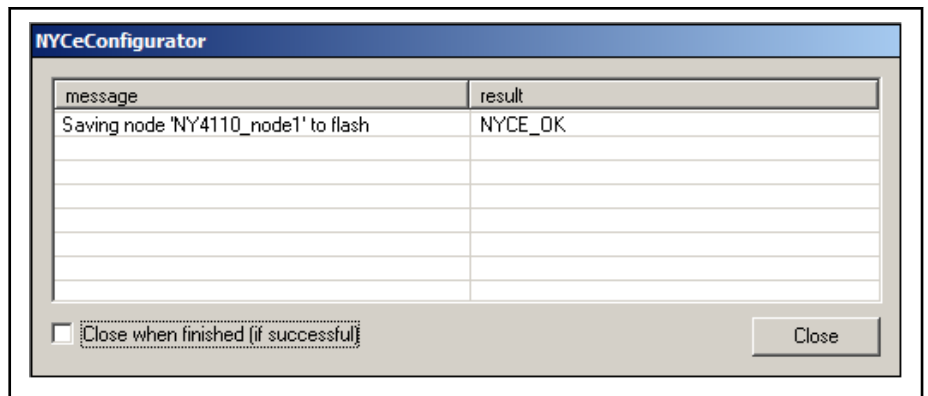


Fig. 3-3: NYCeConfigurator requested action progress message box

Save and restore actions are possible at the following levels.

- At the system level
- At the node level
- At the axis level

At the system level

Right-click on the "NYCe4000 System" in the system tree.

To keep the settings of all nodes and axes that constitute the NYCe 4000 system, these settings can be saved to an XML file (on the host), and / or to the flash memory of all the nodes, see [fig. 3-4 "The menu options at the System level" on page 61](#).

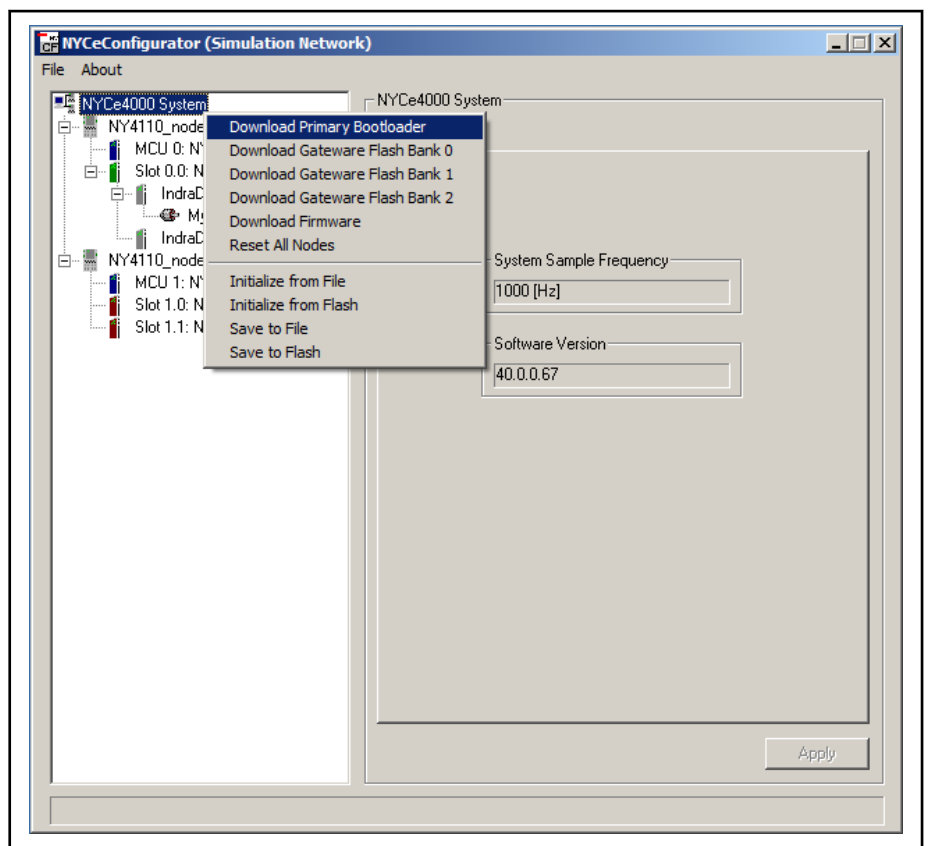


Fig. 3-4: The menu options at the System level

At the node level

Click the node in the system tree, and then click the "Save / Download" tab.

To keep the settings of the selected node (see [chapter 3.4 "Configuration at the node level" on page 62](#)), these settings can be saved to an XML file (on the host), and / or to the flash memory of the node.

At the axis level

Click the axis in the system tree, and then click the "Save / Download" tab.

To keep the settings of the selected axis (see [chapter 3.7 "Configuration at the axis level" on page 82](#)), these settings can be saved to an XML file (on the host), and / or to the flash memory of the node.

The configuration that has been saved during previous sessions can be restored by the same method.

- At the system level.
Right-click on the "NYCe4000 System" in the system tree.
- At the node level.
Click the node in the system tree, and then click the "Save / Download" tab.
- At the axis level.
Click the axis in the system tree, and the click the "Save / Download" tab.

If you load a configuration from file and, for some reason, that configuration misses a parameter or contains a parameter with an invalid value, NYCeConfigurator displays a message box with a warning or error for every missing parameter or parameter of which the value is out of the defined valid range. If you save the configuration to file or flash memory, missing parameters are added with their default value and parameters with an invalid value are saved with their default value.

3.4 Configuration at the node level

When you select a node in the "NYCe4000 System" tree in the pane at the left hand side of the NYCeConfigurator, the pane at the right hand side shows 2 tabs, "General" and "Save / Download".

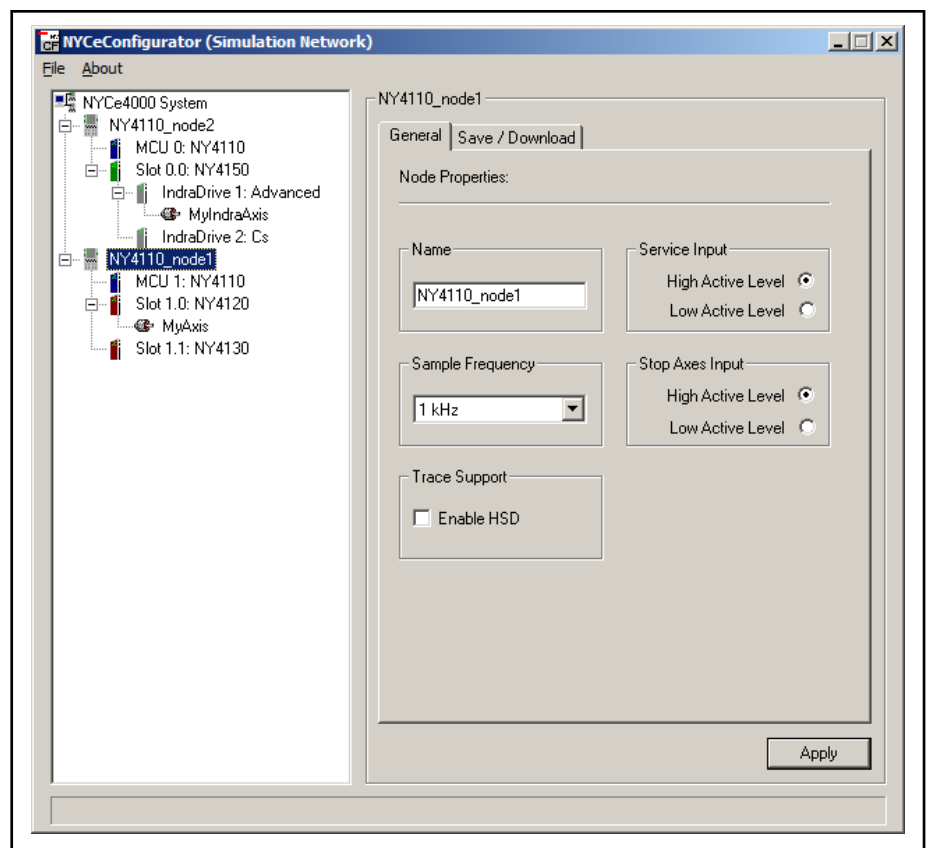


Fig. 3-5: NYCeConfigurator, node level, "General" tab

"General" tab

The "General" tab enables the user to change the name and sample frequency of the node, define the input levels of the Service input and Stop_Axes input, enable the HSD trace support, and read the IP address of the node if the node is NY4112-based. If the node is not in "normal operation" mode, the information that can be changed is greyed.

The node can have any name as long as the length is limited to a maximum of 15 characters, and no special characters (< ? * | >, etc.) are used.

You can set the node frequency with the "Sample Frequency" dropdown list. For several frequency settings the preferred maximum number of axes is indicated. If the node contains an NY4150 SERCOS III Master module, the sample frequency can only be set to 1 or 2 kHz. The other sample frequencies (4 kHz, and 8 kHz) are not supported if the node contains an NY4150.

If the node sample frequency is changed, all axes connected to the node are shut down. If the node contains an NY4150, the IndraDrives connected to the NY4150 are (re)initialized.

Note that NYCe 4000 has an error handler defined for the Service input. You must have a function defined for the Service input in your application software and it cannot be configured with any NYCe 4000 tool.

A high active level means that when the Service or Stop_Axes input is connected to 24V, the Service and Stop_Axes states are activated (and thus when not connected or connected to a voltage of 0V, the Service and Stop_Axes inputs are inactive). With a low active level setting, the Service and Stop_Axes inputs are activated when their inputs are not connected (or

NYCeConfigurator

connected to a voltage potential of 0V). The Service and Stop_Axes inputs are now inactive when their inputs are connected to 24V.

The low active level is important when working with a situation in which cable break between the Service or Stop_Axes switches must always result into an Service or Stop_Axes response of the NYCe 4000 system. The actual response of each axis, when the Service input is activated, is described in the axis error handling tab in [chapter 3.7.12 ""Error Handlers" tab" on page 110](#).

The Stop_Axes input will always result in a complete stop of all axes of the NYCe 4000 system.

The node can be set in the "High-Speed Data enabled" state with a checkmark in the "Enable HSD" box in the "Trace Support" section. See the NYCe 4000 Software User manual, chapter NYCE, for more information about the trace functionality and High-Speed Data (HSD) variables. When the node is set in the "HSD enabled" state, you can trace or view (using NYCeScope) the HSD variables. Default the node is in the "HSD disabled" state. A node should only be set in the "HSD enabled" state for measurements. The amount of data generated when HSD variables are selected in the "HSD enabled" state is substantial, which causes additional delays in the data transfer from the drive module to the MCU. The HSD state can only be changed (to "enabled" or "disabled") when all axes in the node are in the SAC_IDLE state, thus you must initialize all axes again. If you want to initialize the node (from file or flash memory) you must save the node parameters first, otherwise the node is again in the "HSD disabled" state.



If you use node initialization, make sure that the saved file or flash memory is updated (again) with node initialization in the desired HSD state after you completed the measurements.

"Save / Download" tab

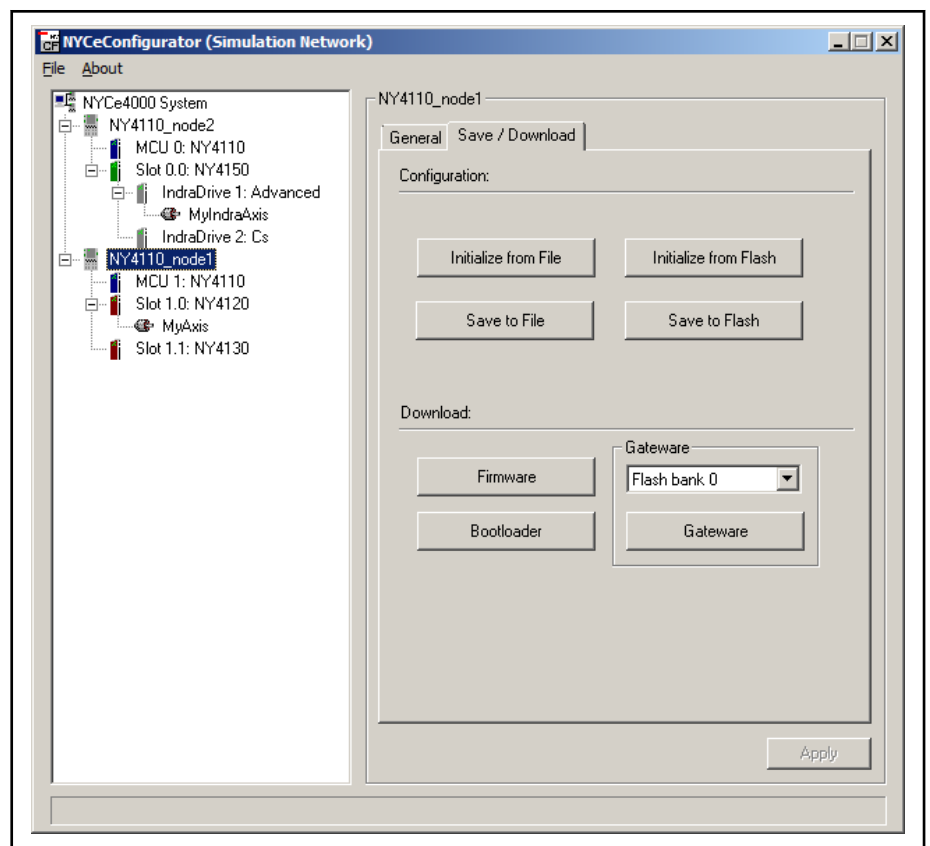


Fig. 3-6: NYCeConfigurator, node level, "Save / Download" tab

The node settings can be saved or initialized from a file or from the flash memory of the node itself on the "Save / Download" tab in the "Configuration" section.

In the "Download" section, buttons are available to download a new bootloader, firmware or gateway files into the node. The download operations are only possible as long as no axes are initialized in the node.

Click the "Firmware" button to download a new firmware file to the node. The "Download Firmware to Node" dialog window appears. Select a Firmware File (*.sys) file, and click the "Open" button. A dialog box appears. NYCeConfigurator is not responsive during the download.

Click the "Bootloader" button to download a new bootloader file to the node. The "Download Bootloader to Node" dialog window appears. Select a Bootloader File (*.bin) file, and click the "Open" button. A dialog box appears.

In the "Gateway" section you can select a gateway file to download to the node. First select the bank into which the new gateway file must be downloaded. The "Download Gateway to Node" dialog window appears. Select a Gateway File (*.bit) file, and click the "Open" button. A dialog box appears.

3.6 Configuration at the drive module level

3.6.1 Introduction

When you select in the "NYCe4000 System" tree a drive module (NY4120, NY4125, NY4130, or NY4140) or an IndraDrive to which a motor with motor encoder flash memory is connected via the NY4150, the pane at the right hand side of the NYCeConfigurator window displays 3 tabs.

1. "General" tab.
2. "Inputs" tab.
3. "Outputs" tab.

If the motor connected to the IndraDrive does not have motor encoder flash memory, a fourth tab called "IndraDrive Motor" appears.

If no axes are defined on the IndraDrive, the "Inputs" and "Outputs" tab is not displayed.

Note that the NY4150 is not a drive module, it is called "SERCOS III Master module". If you select the NY4150 module, the pane at the right hand side displays 2 tabs.

1. "General" tab.
2. "Download" tab.

3.6.2 NY4120, NY4125, NY4130, NY4140 drive module

"General" tab

On the "General" tab, in the "Settings" section, you can set the nominal drive voltage and the maximum temperature of the drive module.

If the drive power supply voltage decreases below $0.5 \cdot$ set nominal drive voltage or increases over $1.5 \cdot$ set nominal drive voltage an error is generated.

The maximum drive temperature is a safeguard for the drive itself, preventing the drive module from overheating. Setting this parameter to 0 will turn off the overheat protection.

In the "Axes" section you can create and delete an axis with the "New" and "Delete" buttons respectively.

NYCeConfigurator

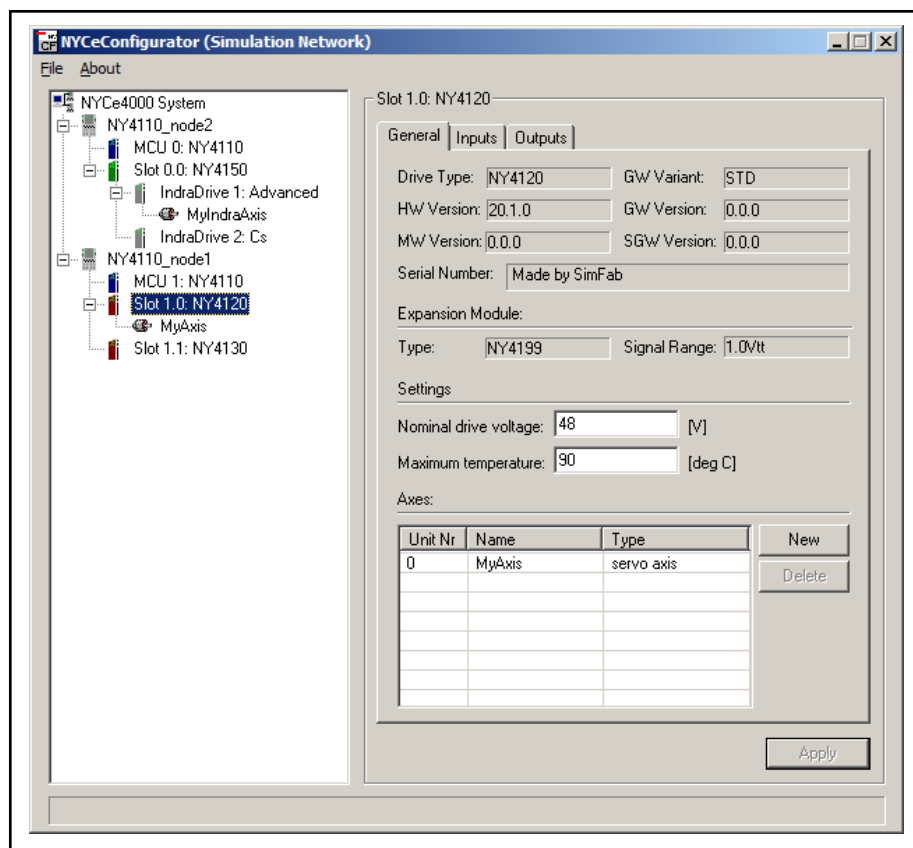


Fig. 3-8: NYCeConfigurator, drive level, "General" tab

Select the drive module in the pane at the left hand side and click the "New" button in the "Axes" section to add a new axis to the drive. If you have created the maximum number of axes the selected drive module supports, the "New" button is disabled.

To delete an axis, select the axis in the "Axes" section at the right hand side and click the "Delete" button. You can delete an axis only when the axis state is "Idle", in all other axis states the "Delete" button is disabled. Note that you are not asked for confirmation of the delete operation.

"Inputs" tab

The "Inputs" tab shows the available digital and analog inputs on the drive.

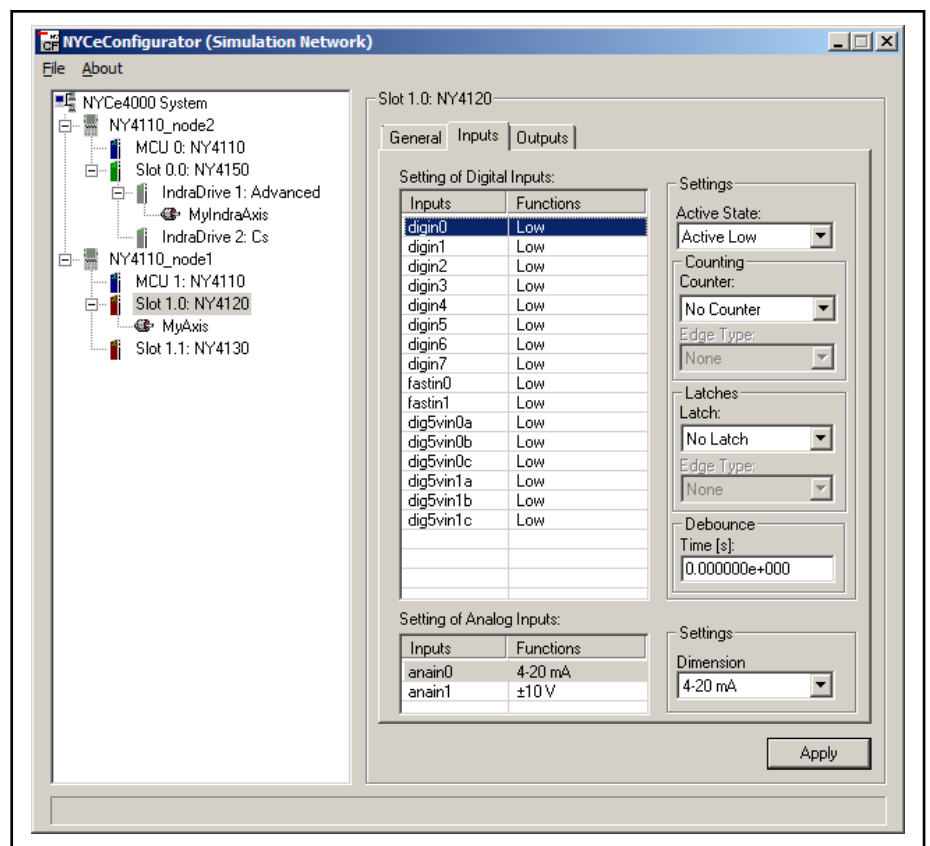


Fig. 3-9: NYCeConfigurator, drive level, "Inputs" tab

The displayed options in the "Settings" section in the pane at the right hand side depends on the selected digital input in the "Settings of Digital Inputs" section.

The digital inputs can be configured as follows (if supported on the selected drive module).

- The active state level can be set to high or low.

A high active state means that the input state is activated, when the input is connected to a high voltage (5V or 24V depending on the input). The input state is inactive when the input is not connected, or connected to 0V.

A low active state means that the input state is activated, when the input is not connected, or connected to 0V. The input state is inactive when the input is connected to a high voltage (5V or 24V depending on the input).
- An input can be configured as a counter input. The input counts the number of edges (falling edge, rising edge or any edge).
- An input can be configured as a latch input. If an edge (falling edge, rising edge or any edge) is detected, the axis position is stored. Latch functionality is supported on an IndraDrive, but must be configured via the axis.
- The debounce time defines the time that an input is insensitive for changes after the input is activated or deactivated. This option is only available for the digital inputs DIGINx.

If PWM is chosen, the "On Delay", "Frequency" and "Duty Cycle" entry boxes appear where you can specify the properties of the PWM signal. The PWM choice sets a pulse width modulated signal on the selected output. See the NYCe 4000 Hardware System Manual for resolution constraints on PWM frequency and duty cycle selection.

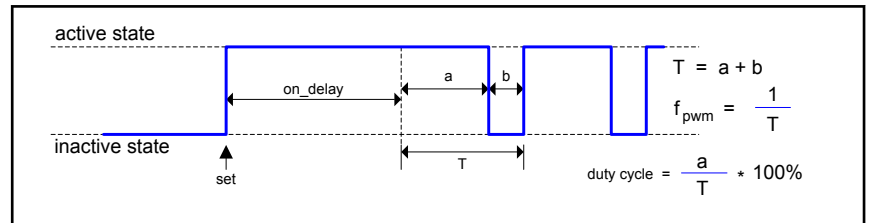


Fig. 3-11: Parameters of the PWM output

3.6.3 NY4150 SERCOS III Master module

In the "General" tab you can see the microware version in the "MW Version" field, and the SERCON gateway version in the "SGW Version" field. See the description of the "General" tab of the drive modules for the other fields.

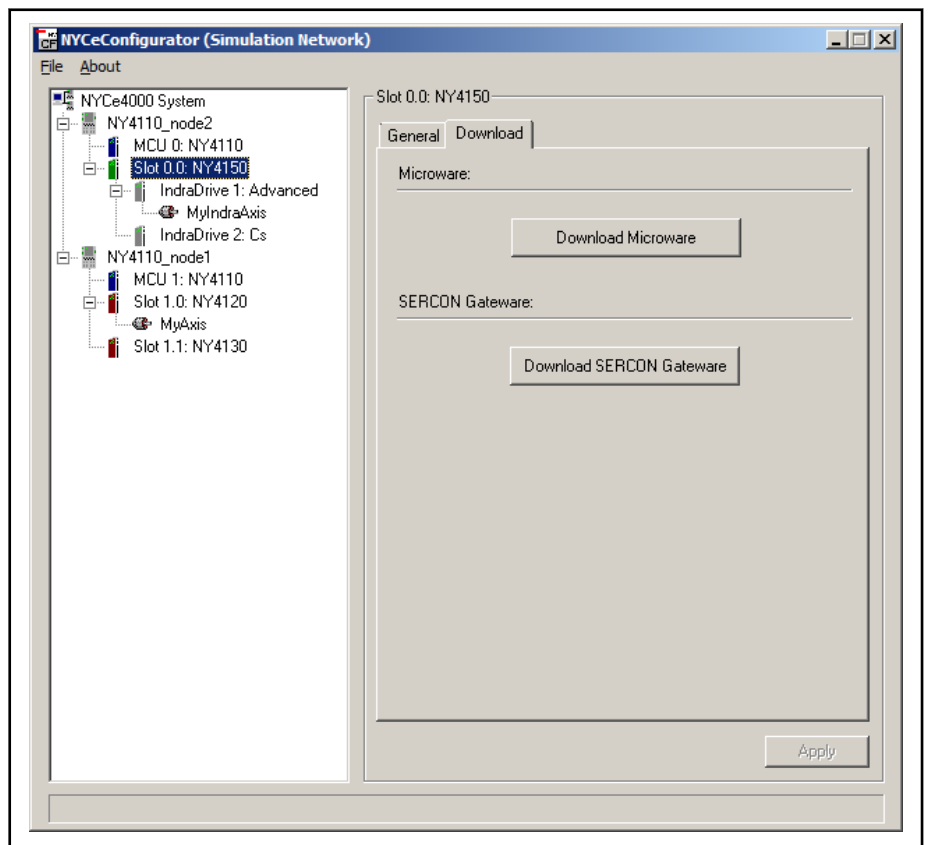


Fig. 3-12: NYCeConfigurator, module level, „Download“ tab (NY4150 only)

With the 2 buttons on the "Download" tab of the NY4150 module you can download an update of the microware or the SERCON gateway to the NY4150 module. The download operations are only possible as long as no axes are initialized of the node (all axes of the node must be in the idle state) and no axes are configured on an IndraDrive.

NYCeConfigurator

If the microware does not start on the NY4150 (because the microware is not loaded into the flash of the NY4150, or another problem occurred), the status line will display "drivelink error". If the microware starts, but it is the wrong version, the status line will display the message "fw wrong microware error". If the microware is started successfully, but the SERCON gateway could not be loaded or is the wrong version, the status line will display "fw wrong sercon gw error".

3.6.4 IndraDrive

If you select an IndraDrive in the pane at the left side, the information shown in the pane at the right side depends on the motor connected to the IndraDrive. If the motor has motor encoder flash memory, the pane at the right side shows the tabs "General", "Inputs" and "Outputs". If the motor does not have motor encoder flash memory, an additional tab "IndraDrive Motor" appears where you can enter the motor parameters manually.



The configuration of an IndraDrive can be accomplished with NYCe 4000. However, you need IndraWorks if you must upgrade the firmware of the IndraDrive. IndraWorks may also prove useful for advanced configuration settings and diagnostics.

The pane at the left side shows the status of the SERCOS network. The following SERCOS network statuses are reported.

SERCOS network status	Explanation
Not responding	The status of the network cannot be retrieved. This condition can occur when the host cannot communicate with the node, the microware on the NY4150 is not running, or there is no communication with the IndraDrive(s).
Detecting network	Initial state, waiting for more information to become available.
Enter CPx failed	An unexpected error occurred while entering SERCOS communication phase "x" (where x is 0, 1, 2, 3, or 4). The error can be caused by invalid IndraDrive parameters. A restore of the parameters to the factory settings might solve the problem. IndraWorks is required for this action.
Double address	Two or more devices with the same SERCOS address are detected.
Ring not closed	The network topology is not a ring. The cause can be a disconnected SERCOS cable or a switched off device.
Too many devices	More than <code>NYCE_MAX_NR_OF_INDRADRIVES</code> devices are detected in one SERCOS network.
Initializing network	The network topology is correct. The network will remain in the initializing state as long as no errors are detected and the devices in the network are not detected.
Initializing devices	The devices in the network are identified and will now be initialized.
Unknown device detected	The network contains one or more SERCOS devices that are not supported.

SERCOS network status	Explanation
Ready ⁽¹⁾	Network is configured correctly and ready for use.
Waiting for axes idle	The network is not operational any more. The network cannot be initialized before all axes using the network are shutdown and unconfigured from the network.

① all statuses except the "ready" status are actually indicated with a text.

Tab. 3-1: SERCOS network statuses

The IndraDrive in the pane at the left side shows the IndraDrive type, C or Cs, and its SERCOS address. The SERCOS address must be unique in the SERCOS network. You can change the SERCOS address of an IndraDrive via the display. While the display shows the standard information, SERCOS address and status, press the keys "Enter" and "Esc" at the same time, and keep the keys pressed for 8 seconds. The display shows "1. Er. Anz". Press the up key to enter the commands/settings menu, the display shows "2. Comand". Select the element "2.1 drive access" with the up or down key, then confirm with the "Enter" key. Now you can set the SERCOS address using the up and down keys. First you set the tens digit, the display shows "2.1.1 drive address". Press the "Enter" key when the tens digit is correct. Then you set the units digit, the display shows "2.1.2 drive address". Press the "Enter" key when the units digit is correct.

Press the "Esc" key until the display shows the standard information, SERCOS address and status. After you changed the SERCOS address on an IndraDrive, you must initialize the SERCOS network again to force an update of the administration in the node and the host by power cycling the IndraDrive and node.

If the IndraDrive is not shown in the pane at the left side, it is possible that the IndraDrive communication protocol is set to a wrong type. If this is the case the error `SERC_ERR_UNKNOWN_DEVICE` is not reported. To check the protocol setting, press the "Enter" key while the display shows the standard information. The IndraDrive displays the "Easy menu". Use the up or down key to select the menu entry "Comm. Protocol" if that entry is not shown on the display. Press the "Enter" key. The display shows the selected communication protocol, for example "SERCOS 3", "PROFINET", or "EtherCAT". If the shown protocol is not "SERCOS 3", press the "Enter" key. Then press the up or down key to select the communication protocol, "SERCOS 3". When the display shows "SERCOS 3" as selected protocol, press the "Esc" key to return to the previous menu.

Press the "Esc" key until the display shows the standard information, SERCOS address and status. After you changed the communication protocol setting on an IndraDrive, you must initialize the SERCOS network again to force an update of the administration in the node and the host by power cycling the IndraDrive and node.

If a SERCOS device is recognized and supported by NYCe 4000, it can have a device state. If the SERCOS device is not recognized or supported, the device cannot have a device state. The following device states are available.

NYCeConfigurator

SERCOS device state	Explanation
Normal operation ⁽¹⁾	The device is ready for use. Note that the device state is not related to the axis state of the axis configured for the device. If the axis is in an error state, the SERCOS device state will be in the "normal operation state".
Initializing	The device is initializing.
Wrong firmware version	The IndraDrive firmware is identified by a version number and a release number. The version number must be equal to the version for which the NYCe 4000 software version is released, and the release number must be equal to or higher than the number mentioned in the NYCe 4000 Release Bulletin. You need IndraWorks to install an other firmware version in the IndraDrive.
Close loop package not loaded	IndraDrives cannot be used with NYCe 4000 without the closed loop package.
Initialization failed	General initialization error. The cause can only be determined by debugging the SERCOS network start up, see tab. 3-1 "SERCOS network statuses" on page 72 .

① all states except the "normal operation" state are actually indicated with a text.

Tab. 3-2: SERCOS device states

If the microware detects that the closed loop package is not active on the IndraDrive, the error `SERC_ERR_CLP_NOT_LOADED` is generated and the closed loop package is selected. The closed loop package is activated after the IndraDrive is power cycled.

The relation between SERCOS network state and SERCOS device state is shown in [fig. 3-13 "State diagram for a SERCOS network in NYCe 4000" on page 75](#). The state diagram does not show the outgoing transitions from the error states for clarity reasons. The 4 highlighted (green) states show the transitions for a correct setup of the SERCOS network. The following abbreviations are used in the transition state boxes.

- NS – Network Status (see [tab. 3-1 "SERCOS network statuses" on page 72](#))
- DT – Device Type (IndraDrive C Advanced or IndraDrive Cs)
- DS – Device Status (see [tab. 3-2 "SERCOS device states" on page 74](#))

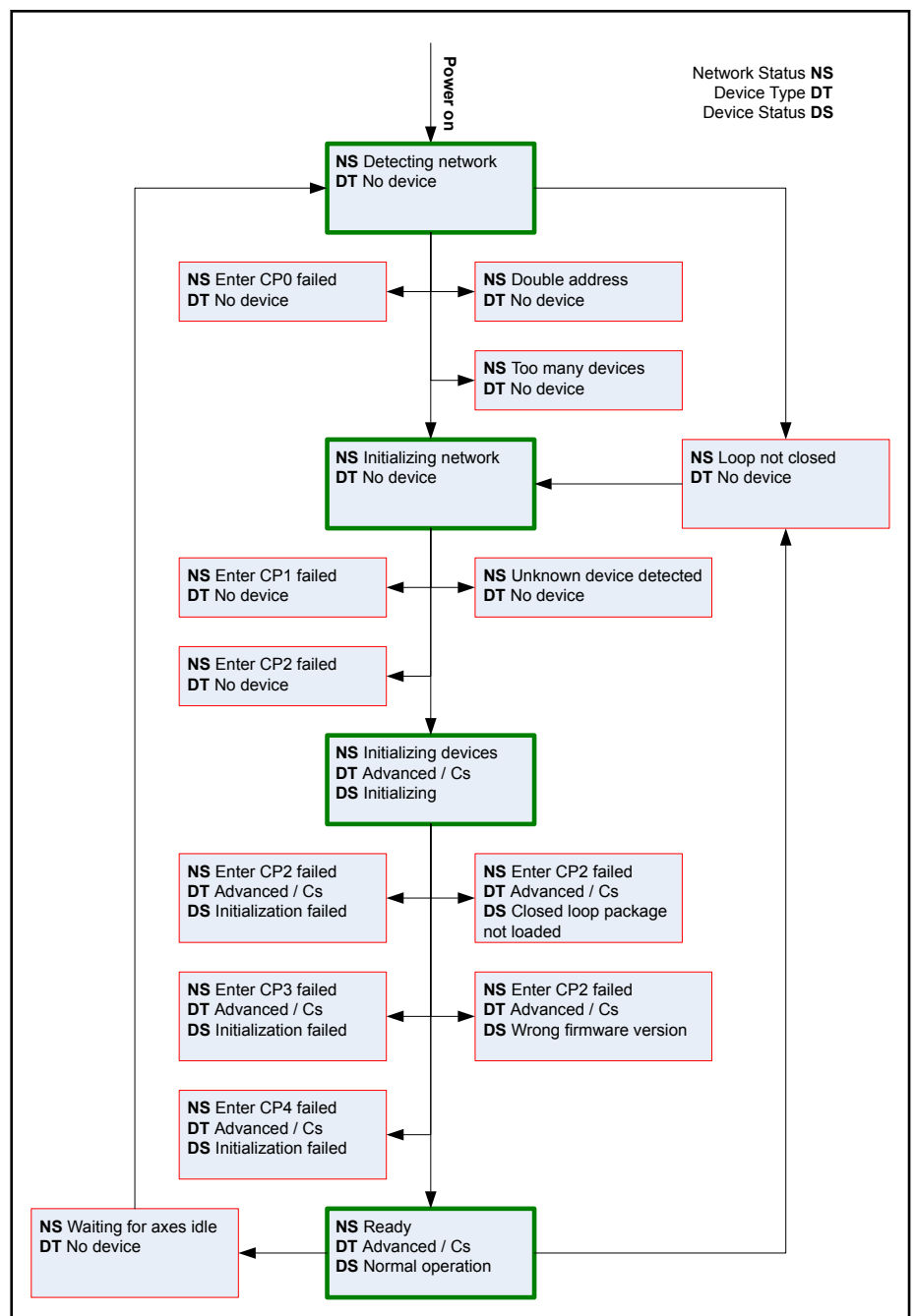


Fig. 3-13: State diagram for a SERCOS network in NYCe 4000

IndraDrive (motor with motor encoder flash memory)

On the "General" tab you can see information of the control section of the IndraDrive (firmware, type and serial number) and the power section (type and serial number) if an axis is assigned to the IndraDrive. If no axis is assigned to the IndraDrive the fields do not show any information. There is no "Settings" section available as for the NY4120, NY4125, NY4130, and NY4140 drive modules.

In the "Axes" section you can create and delete an axis with the "New" and "Delete" buttons respectively.

NYCeConfigurator

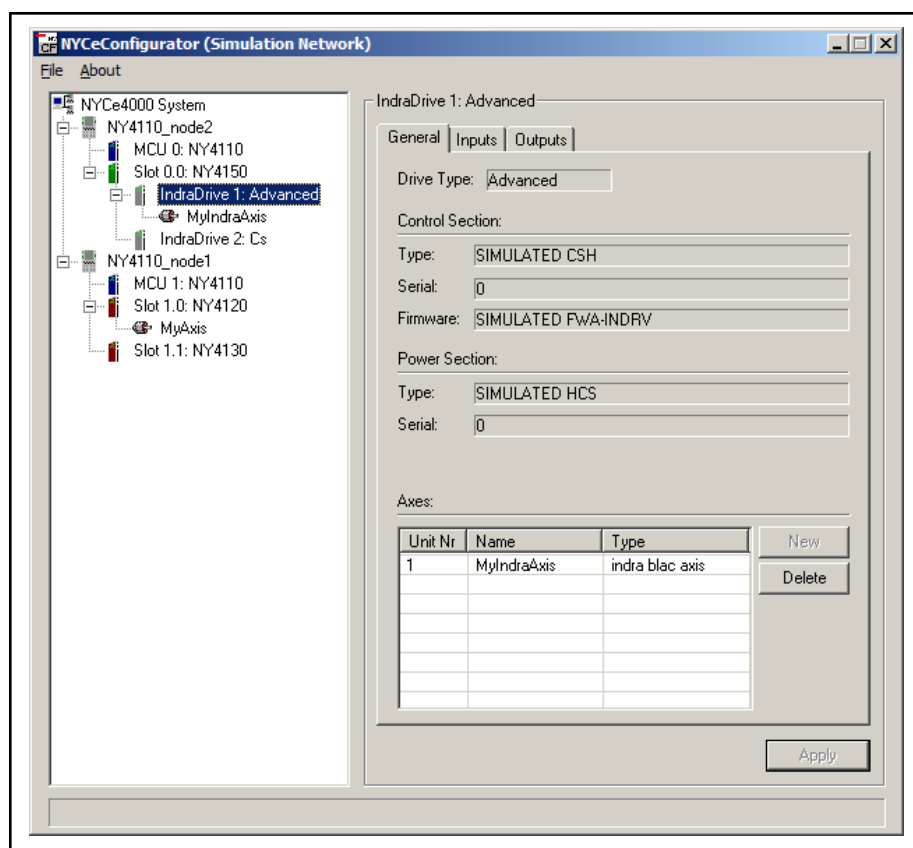


Fig. 3-14: IndraDrive with (rotary) motor with motor encoder flash memory

Select the IndraDrive in the pane at the left hand side and click the "New" button in the "Axes" section to add a new axis to the IndraDrive. If you have created the maximum number of axes the selected drive module supports, the "New" button is disabled.

To delete an axis, select the axis in the "Axes" section at the right hand side and click the "Delete" button. Note that you are not asked for confirmation of the delete operation.

IndraDrive (motor without motor encoder flash memory)

If a motor without motor encoder flash memory is connected to the IndraDrive, a fourth tab called "IndraDrive Motor" appears at the drive module level. On this tab are 5 tabs to define the parameters for the motor and its encoder, see [fig. 3-15 "IndraDrive with \(rotary\) motor without motor encoder flash memory" on page 77](#).

For detailed information of the mentioned P parameters and S parameters, see Bosch Rexroth documentation "Rexroth IndraDrive for Drive Controllers MPH-05, MPB-05, MPD-05", order number R911320182.

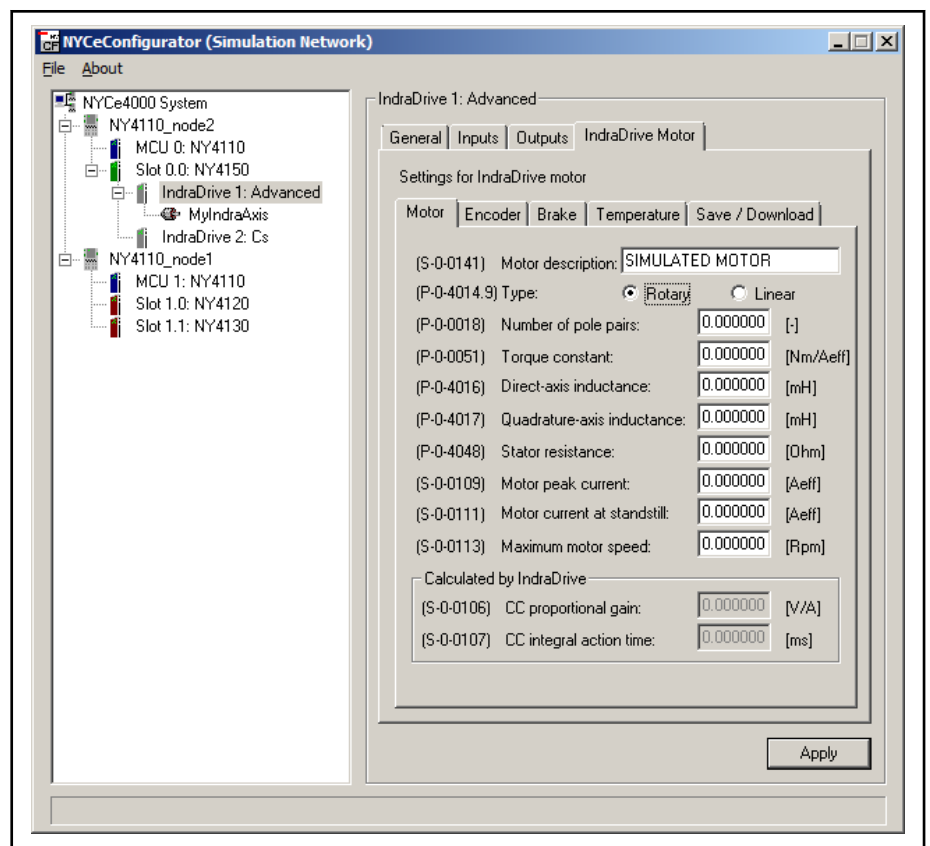


Fig. 3-15: IndraDrive with (rotary) motor without motor encoder flash memory

"Motor" tab

On the "Motor" tab you can enter a name in the "Motor description" field, to describe the connected motor. The name is stored in the S parameter S-0-0141. The radio button "Type" (parameter P-0-4014.9) defines the type of motor: rotary or linear. Other P parameters and S parameters must be assigned correct values on the "Motor" tab, see [fig. 3-15 "IndraDrive with \(rotary\) motor without motor encoder flash memory" on page 77](#). Most parameters are self-explanatory and are not mentioned here.

Parameters P-0-4016 "Direct-axis inductance" and P-0-4017 "Quadrature-axis inductance" need an explanation. For motors with reluctance property, different values are observed when the inductance is measured, depending on the position of the primary part of the motor with regard to the secondary part of the motor. The values fluctuate between a minimum and a maximum value. The minimum value is relevant for the direct-axis inductance and the maximum value is relevant for the quadrature-axis inductance.

The parameter value for P-0-4016 is $L_{(U-V)min} * 0,5$ where $L_{(U-V)min}$ is the minimum value of one coil, taken from the motor manufacturer data sheet.

The parameter value for P-0-4017 is $L_{(U-V)max} * 0,5$ where $L_{(U-V)max}$ is the maximum value of one coil, taken from the motor manufacturer data sheet.

[fig. 3-15 "IndraDrive with \(rotary\) motor without motor encoder flash memory" on page 77](#) lists the parameters with their dimensions when the motor type is set to rotary. If the motor type is set to linear, the parameters and their dimensions reflect that type of motor, see [fig. 3-16 ""Motor" tab for a \(linear\) motor without motor encoder flash memory" on page 78](#).

NYCeConfigurator

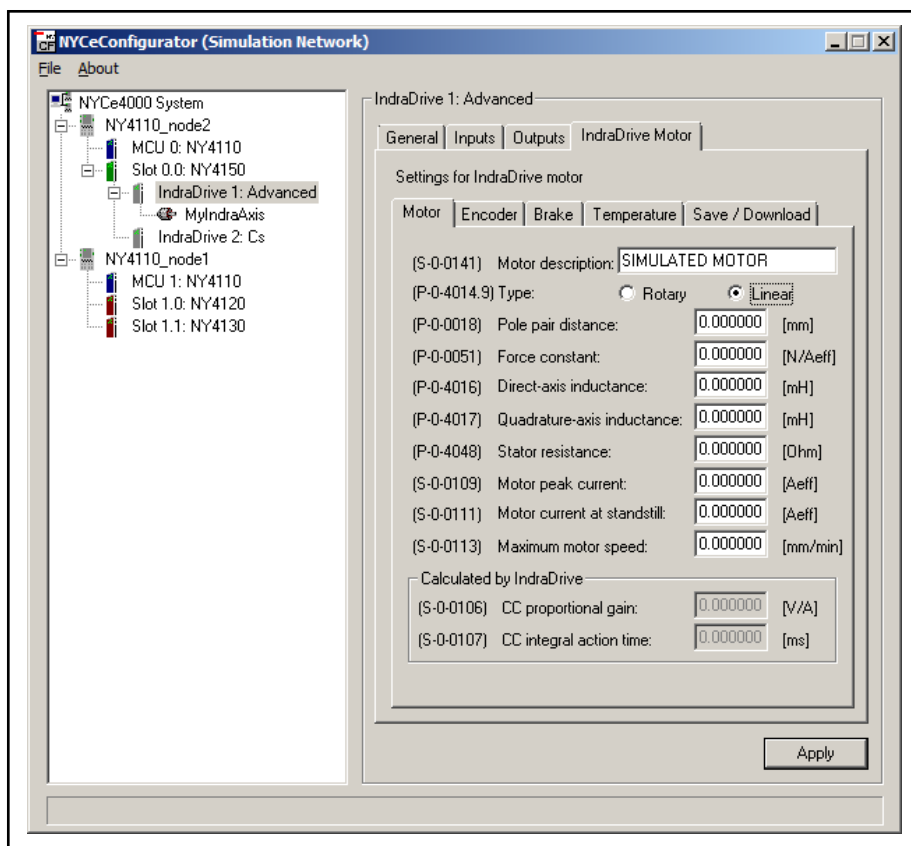


Fig. 3-16: "Motor" tab for a (linear) motor without motor encoder flash memory

Click the "Apply" button after you entered the correct values. The "CC proportional gain" (parameter S-0-0106) and "CC integral action time" (parameter S-0-0107) are calculated and displayed in the section "Calculated by IndraDrive".

The calculated values can be used as initial values for tuning the current control loop. This can only be done in NYCeTuner and for that reason the values cannot be changed in NYCeConfigurator.

"Encoder" tab

On the "Encoder" tab you define the type of encoder connected to the motor, the connector on the IndraDrive used to connect the encoder, and the resolution of the encoder, see [fig. 3-17 "Encoder" tab for a motor without motor encoder flash memory](#) on page 79.

From the drop-down box "Encoder type" (parameter P-0-0074) you can select the following encoders. The S0S90 encoder is also known as A/B encoder, quadrature encoder and 3-channel digital incremental encoder.

- 2: 5V SinCos
- 4: Hiperface
- 5: 5V S0S90
- 8: EnDat2.1
- 10: Resolver
- 13: 5V S0S90 + 12V Hall
- 15: 5V SinCos + 12V Hall

- 16: 12V SinCos
- 18: 12V S0S90
- 19: NYCe encoder
- 23: 5V SinCos + 12V dig. Hall
- 26: Digital

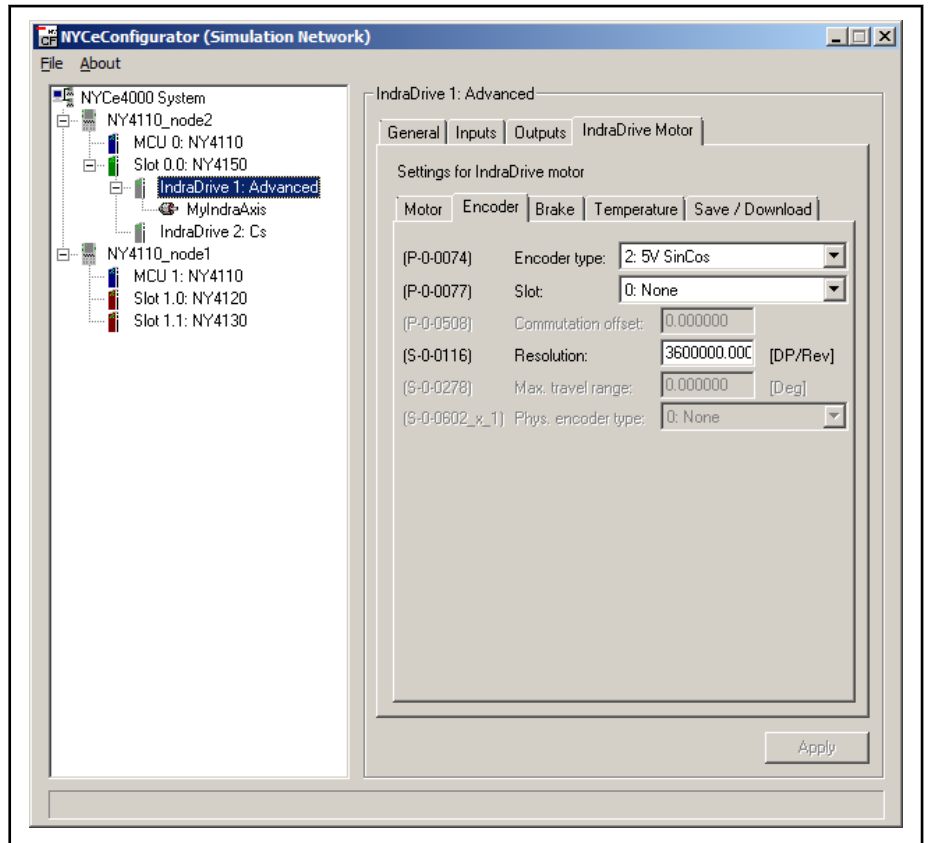


Fig. 3-17: "Encoder" tab for a motor without motor encoder flash memory

The drop-down box "Slot" (parameter P-0-0077) defines the connector indicated on the IndraDrive which is used to connect the encoder. The following connector selections are available.

- 0: None
- 1: X4 (option1)
- 2: X8 (option 2)
- 3: X10 (option 3)
- 10: SERCOS

If the encoder type "13: 5V S0S90 + 12V Hall", "15: 5V SinCos + 12V Hall", or "23: 5V SinCos + 12V dig. Hall" is selected, you can set the commutation offset parameter P-0-0508 with the field "Commutation offset". This field is disabled for all other encoders.

Specify the encoder resolution in the field "Resolution" (parameter S-0-0116). If the selected motor type (on the "Motor" tab) is a rotary motor, the dimension is DP/Rev. "DP" stands for "Division Period" and equals one cycle of the signals of (for example) a SinCos encoder, see [fig. 3-18 "Division Period" on page 80](#).

If the selected motor type is a linear motor, the dimension is mm.

NYCeConfigurator

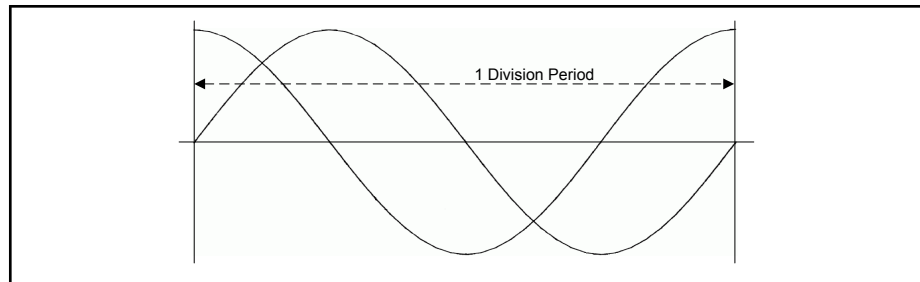


Fig. 3-18: Division Period

The "Max. travel range" setting is only available if the motor type is "Linear".

The "Phys. encoder type" can only be set to "0: None" or "106: EnDat2.2" if the selected Encoder type is "26: Digital".

"Brake" tab

On the "Brake" tab you can specify if the motor has a brake (with a check mark in the field "Motor holding brake available", parameter P-0-0525.2), see [fig. 3-19 "Brake" tab for a motor without motor encoder flash memory](#) on page 80. The radio button "Holding" (parameter P-0-0525.0) specifies whether the brake is self-holding when no power is applied or electrically holding when power is applied.

The entry fields "Drive on delay" (parameter S-0-0206) and "Drive off delay" (parameter S-0-0207) specify the reaction times of the brake.

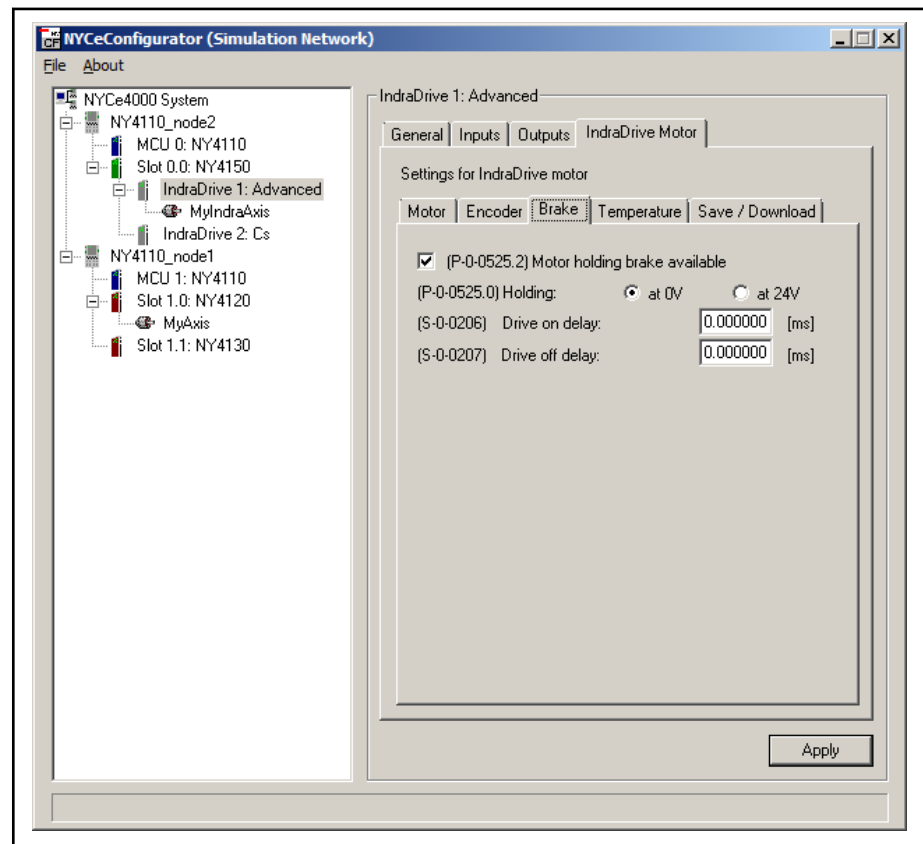


Fig. 3-19: "Brake" tab for a motor without motor encoder flash memory

"Temperature" tab

On the "Temperature" tab you can specify whether a sensor is available to monitor the temperature of the motor. The following sensors are available from the "Sensor type" drop-down list (parameter P-0-0512).

- No sensor
- Thermik SNM150-DK for other motors (PTC)
- Siemens K227 (NTC)
- Siemens KTY84 (PTC)
- Thermik SNM150-DK for MKE motors (PTC)
- Sensor for Bosch SF motors (NTC)
- PTC LMS 1k (PTC)

If a sensor is defined, you can set the temperature when a warning is generated (parameter S-0-0201) and a shut down is performed (parameter S-0-0204). See [fig. 3-20 "Temperature" tab for a motor without motor encoder flash memory](#) on page 81. For the PTC LMS 1k sensor, used in LMS on IndraDrive Cs, you must set the warning temperature value to 110 degrees Celsius and the error temperature value to 120 degrees Celsius.

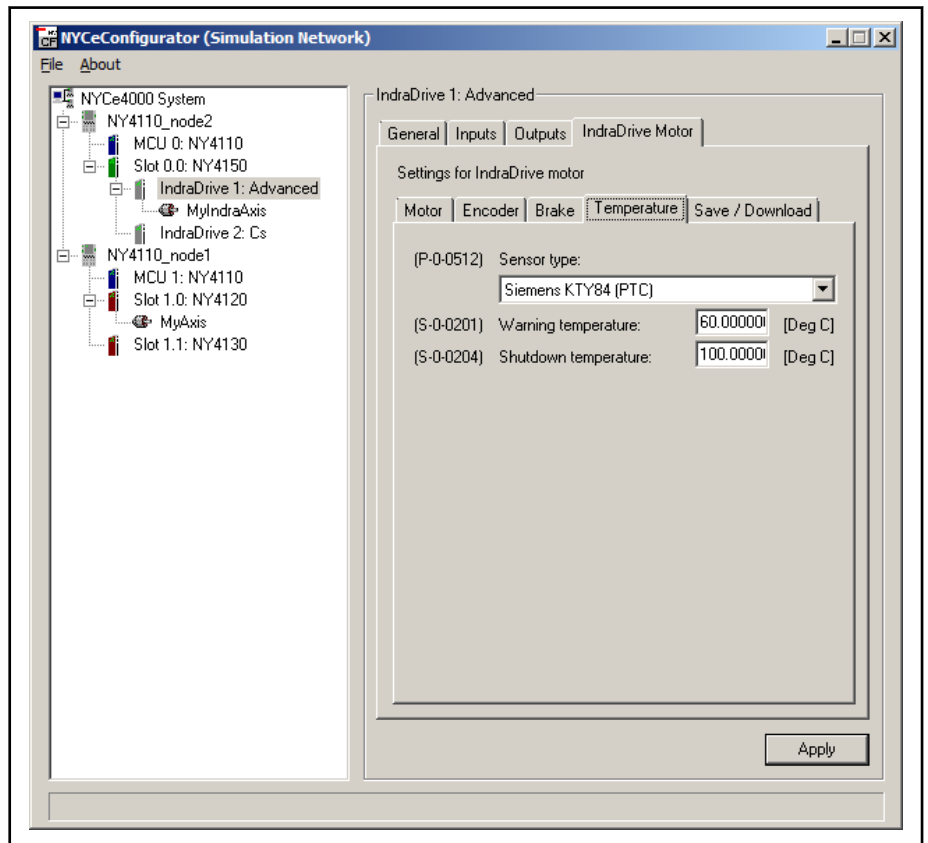


Fig. 3-20: "Temperature" tab for a motor without motor encoder flash memory



The warning and shutdown temperatures are specified in degrees Celsius. If you change the parameter S-0-0208 to "1", the IndraDrive scales the setting to degrees Fahrenheit, but NYCeConfigurator does not display that change. The default for parameter S-0-0208 is "0", the IndraDrive scales the setting to degrees Celsius.

NYCeConfigurator

"Save / Download" tab

On the "Save / Download" tab are 3 buttons available to save and download the motor parameters, see [fig. 3-21 "Save / Download" tab for a motor without motor encoder flash memory](#) on page 82.

- Load from File

If you have a motor XML file with the motor parameters, you can use the "Load from File" button to download the motor parameters to the IndraDrive.

- Save to File

With the button "Save to File" you can save the motor parameters from the IndraDrive into a motor XML file.

- Save to Flash

With the button "Save to Flash" you can save the motor parameters from the IndraDrive into the flash memory of the IndraDrive.

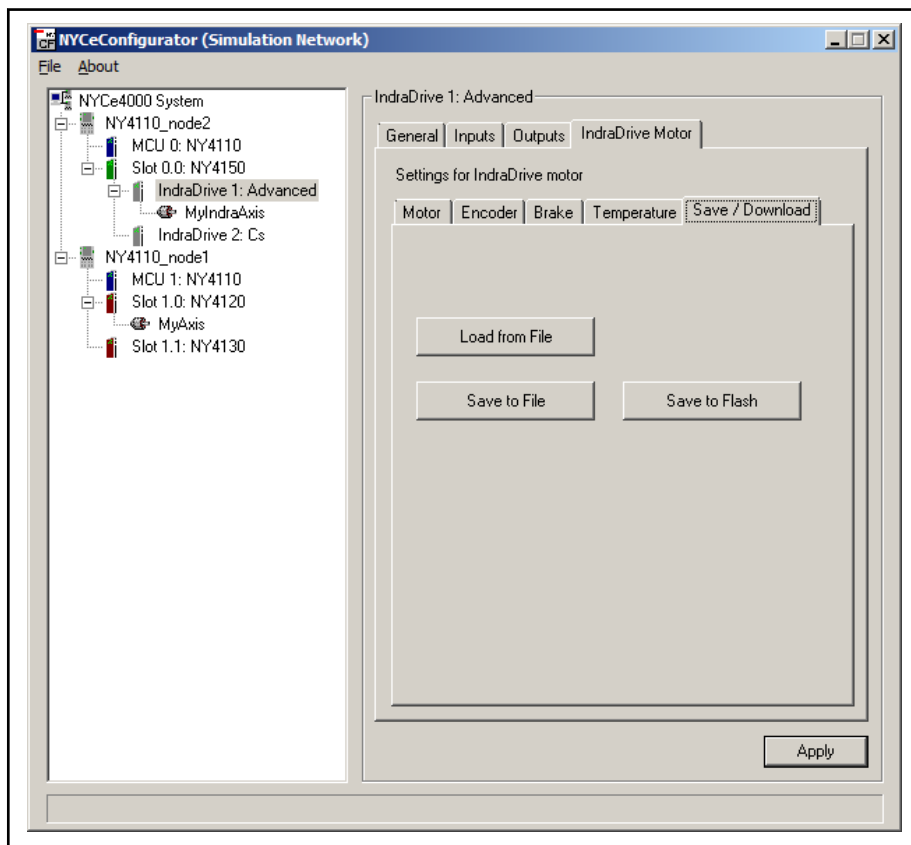


Fig. 3-21: "Save / Download" tab for a motor without motor encoder flash memory

3.7 Configuration at the axis level

3.7.1 Introduction

When you select an axis in the "NYCe4000 System" tree, the pane at the right hand side of the NYCeConfigurator window shows several tabs.

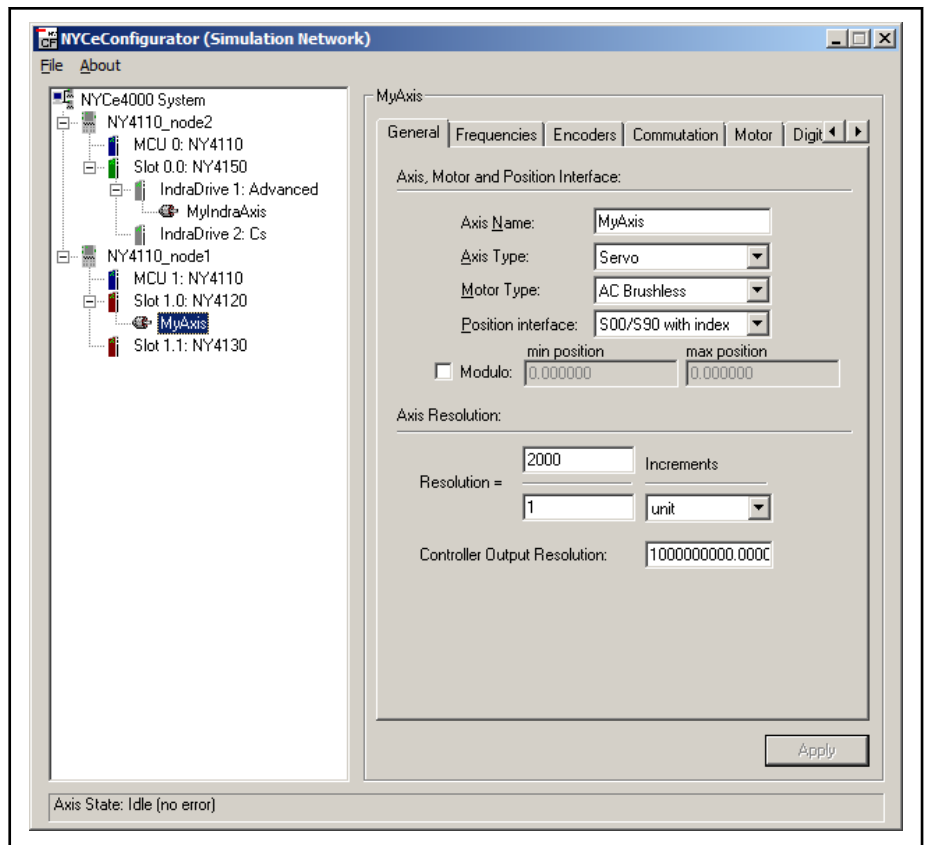


Fig. 3-22: NYCeConfigurator, axis level, "General" tab

The displayed tabs depend on properties of the axis. For example, if the "Motor Type" selection is "AC Brushless" or "DC Brushless" an additional tab "Commutation" is added.

3.7.2 "General" tab

On the "General" tab, you can change the name of the axis. The axis can have any name, as long as the length is limited to a maximum of 15 characters and no special characters (< ? * | >, etc.) are used.

[tab. 3-3 "Axis Type setup" on page 84](#) shows the axis type setup options. If an axis is selected which is defined on an IndraDrive, the axis type is set to "IndraDrive axis", the motor type is set to "IndraDrive BLAC" and the position interface is set to "No sensor", and these settings cannot be changed.

NYCeConfigurator

Axis type	Motors	Feedback	Resolution definition ⁽¹⁾ ⁽²⁾ ⁽⁷⁾
Sensor	None	Analog IN ⁽⁵⁾	$(2^{ADC_{resolution}})/pu$
		Rexroth MSM incremental encoder	encinc/pu
		Rexroth MSM absolute encoder	encinc/pu
		S00/S90 with index ⁽⁴⁾	encinc/pu
		S00/S90 without index ⁽⁴⁾	encinc/pu
		SinCos	encinc/pu
		EnDat2.1	encinc/pu
		EnDat2.2	encinc/pu
		Hiperface encoder	encinc/pu
Sanyo Denki absolute	encinc/pu		
Open loop	DC External ⁽⁶⁾	None	none
Stepper	Stepper External stepper ⁽⁸⁾	None	steps / unit
Sensing Stepper ⁽³⁾	Stepper	Analog IN ⁽⁵⁾	$(2^{ADC_{resolution}})/pu$
		Rexroth MSM incremental encoder	encinc/pu
		Rexroth MSM absolute encoder	encinc/pu
		S00/S90 with index ⁽⁴⁾	encinc/pu
		S00/S90 without index ⁽⁴⁾	encinc/pu
		SinCos	encinc/pu
		EnDat2.1	encinc/pu
		EnDat2.2	encinc/pu
		Hiperface encoder	encinc/pu
Sanyo Denki absolute	encinc/pu		

Axis type	Motors	Feedback	Resolution definition ⁽¹⁾ ⁽²⁾ ⁽⁷⁾	
Servo	DC	Analog IN ⁽⁵⁾	$(2^{ADC_{Resolution}})/pu$	
	BLDC	Rexroth MSM incremental encoder	encinc/pu	
	BLAC	Rexroth MSM absolute encoder	encinc/pu	
	External ⁽⁶⁾	S00/S90 with index ⁽⁴⁾		encinc/pu
		S00/S90 without index ⁽⁴⁾		encinc/pu
		SinCos		encinc/pu
		EnDat2.1		encinc/pu
		EnDat2.2		encinc/pu
		Hiperface encoder		encinc/pu
		Sanyo Denki absolute		encinc/pu
IndraDrive axis	IndraDrive BLAC	No sensor	iprinc/pu	
Coil (for LMS only)	Coil Drive Coil IndraDrive	Dual SinCos Triple SinCos		

- ① pu = position unit.
- ② encinc = encoder increment.
- ③ The feedback on a sensing stepper is not used in the control loop.
- ④ The S0S90 encoder is also known as A/B encoder, quadrature encoder and 3-channel digital incremental encoder.
- ⑤ Analog input AnIn0 is fixed allocated to axis0, and AnIn1 is fixed allocated to axis1.
- ⑥ Analog output AnOut0 is fixed allocated to axis0, and AnOut1 is fixed allocated to axis1.
- ⑦ iprinc = IndraDrive Position Resolution increment.
- ⑧ Fast output FastOut0 is fixed allocated for the "pulse" signal, and fast output FastOut1 is fixed allocated for the "direction" signal.

Tab. 3-3: Axis Type setup



The Rexroth MSM absolute encoder and Sanyo Denki absolute encoder count downward (position increases) when the shaft of the encoder is rotated clockwise. This behavior is opposite to the behavior of the other encoders. When the parameter SAC_PAR_MEAS_SYST_DIRECTION is set to -1, the Rexroth MSM absolute and Sanyo Denki absolute encoder behavior is identical to the other encoders.

NYCe 4000 defines the following axis types.

- Sensor

NYCeConfigurator

The sensor axis consists of only a sensor, no motor. The sensor axis can be used to follow the position of an object, and based on that position, control the position of another object.

- Open loop

An open loop axis does not have an encoder, only a motor.

- Stepper

A stepper axis does not have an encoder, only a stepper motor. The stepper motor can be connected to the drive outputs of the module, or via an external stepper drive unit using the fast digital outputs.

- Sensing stepper

A sensing stepper axis consists of an encoder and a stepper motor. Note that the encoder is not used in the control loop.

- Servo

A servo axis consists of an encoder and a motor (not stepper motor or IndraDrive) or an analog output to control an external drive. The encoder is used in the control loop.

- IndraDrive axis

The IndraDrive axis is an IndraDrive connected to the NYCe 4000 via the NY4150 SERCOS III Master module. The IndraDrive has its own encoder.

- Coil

This axis type is only possible in an LMS system, and the Axis Type field is greyed, that is, the entry cannot be changed. Further, the Motor Type field is set to either "Coil Drive" or "Coil IndraDrive", and cannot be changed. The fields are set by the LMS software.

See the NYCe 4000 Hardware System Manual for the ADC resolution of the NY4120, NY4130, and NY4140 drive modules.

You must also define the controller output resolution. This is the scaling of the controller output. Internally, the controller output uses a range of -1.0 ... +1.0. With the "Controller Output Resolution" you can scale this range to any other range for data presentation, for example in NYCeScope.



All parameters shown in NYCeTuner that have "cout" as part of their dimension are also scaled by "Controller Output Resolution" and are automatically rescaled when you change the "Controller Output Resolution".

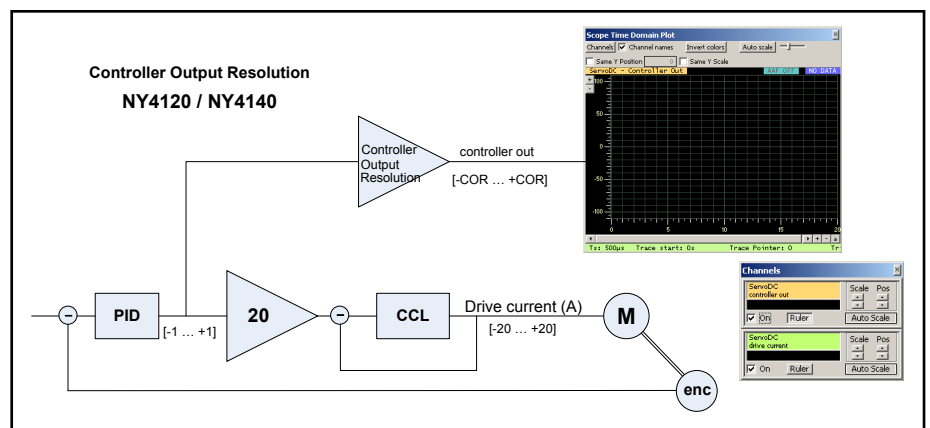


Fig. 3-23: Controller Output Resolution for NY4120 and NY4140

For NY4120 and NY4140 drive modules, when the "Controller Output Resolution" value is set to 20.0 (the maximum drive current), the controller output value can be seen as a value between -20.0 ... +20.0 which matches the output current, thus the controller output unit is Amperes, see [fig. 3-23 "Controller Output Resolution for NY4120 and NY4140" on page 87](#).

Using the drive current multiplied by the motor constant makes the controller out equal to the torque produced by the motor (thus the controller output unit is Newton meter). However, the range of the motor drive current is physically always -20.0 ... +20.0 A.

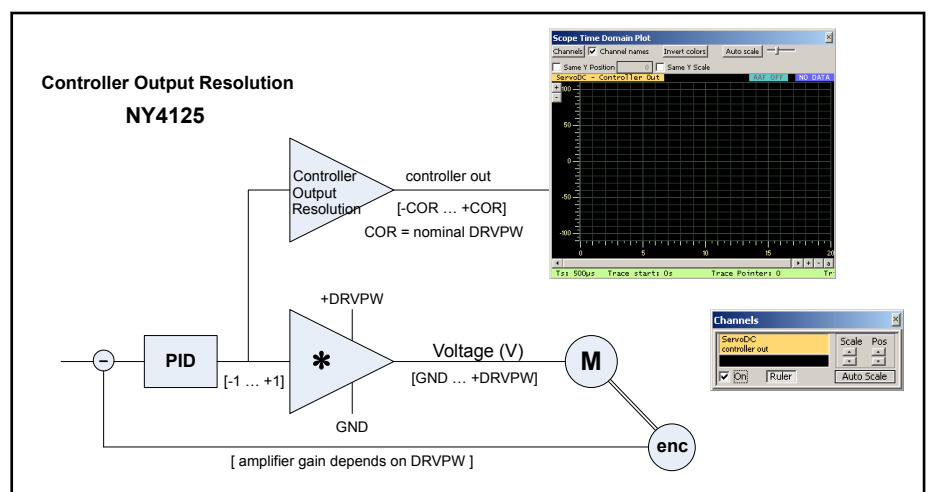


Fig. 3-24: Controller Output Resolution for NY4125

For NY4125 drive modules, the correlation between the "Controller Output Resolution" and the actual motor voltage differs from the NY4120, NY4130, and NY4140 drive modules. For the NY4125 drive module the "Controller Output Resolution" must be set equal to the nominal drive voltage to get a 1:1 correlation between the controller out and the motor voltage, because the gain of the amplifier depends on the drive power voltage. The controller output value can be seen as a value which matches the output voltage, thus the controller output unit is Volts, see [fig. 3-24 "Controller Output Resolution for NY4125" on page 87](#). The controller out value is used for data presentation, for example in NYCeScope. However, the range of the motor drive voltage is physically always the supplied drive power voltage.



"Controller Output Resolution" is independent of the drive power voltage for the NY4120, NY4130, and NY4140 drive module, but depends on the drive power voltage for the NY4125 drive module.

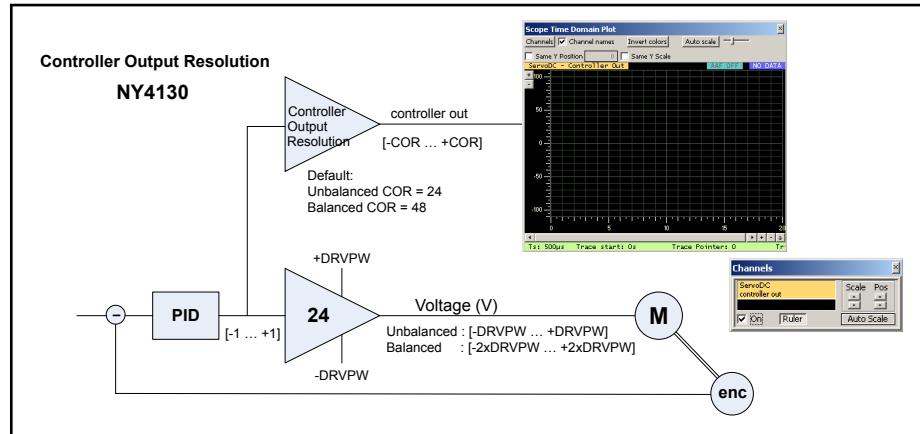


Fig. 3-25: Controller Output Resolution for NY4130

The NY4130 does not support the current controller, so the controller output resolution is the voltage scaling of the drive. When the "Controller Output Resolution" value is set to 24.0 (the gain of the output amplifier, unbalanced, meaning that the motor is connected to an amplifier output and ground), or 48 (the gain of the output amplifier, balanced, meaning that the motor is connected between two amplifier outputs), the controller output value can be seen as a value which matches the output voltage, thus the controller output unit is Volts, see [fig. 3-25 "Controller Output Resolution for NY4130" on page 88](#). The controller out value is used for data presentation, for example in NYCeScope. However, the range of the motor drive voltage is physically always the supplied drive power voltage.

For external drive use, "Controller Output Resolution" is normally set to 10 which will make the controller output unit equal to Volts. Note that the output is normally specified in Volt, whereas the analog input resolution (used for positioning) is defined in the number of bits.

For an IndraDrive, set the "Controller Output Resolution" to the specified peak current of the type of IndraDrive connected to the NY4150. Note that IndraDrives specify the nominal current, not the peak current.

If the selected axis is on an IndraDrive, the "Axis Resolution" section on the "General" tab includes an additional setting, the "IndraDrive Position Resolution" drop-down selection box. The position read from the IndraDrive has a granularity of 0.001, 0.0001, 0.00001, or 0.000001 degree for rotational encoders or millimeter for linear encoders, defined by the "IndraDrive Position Resolution" drop-down selection box. See [fig. 3-26 "IndraDrive Position Resolution drop-down box for IndraDrive-based axes" on page 89](#). For example, if you select 0.0001 as "IndraDrive Position Resolution" and the position unit is one revolution, you enter $360 \text{ degrees} / 0.0001 = 3600000$ in the field "Resolution".

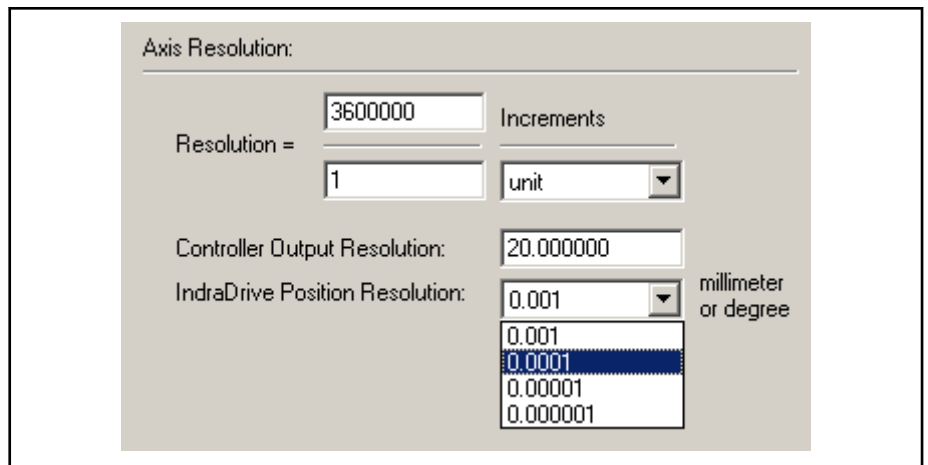


Fig. 3-26: IndraDrive Position Resolution drop-down box for IndraDrive-based axes

3.7.3 Position sensor interface – "Encoders" tab (not for IndraDrive)

On the "General" tab you can select a position interface, if applicable. Depending on the selected position interface a new tab called "Encoders" appears.

Rexroth MSM incremental and absolute encoder

The Rexroth MSM encoder (incremental and absolute type) counts downward (position increases) when the shaft of the encoder rotates clockwise. This behavior is opposite to the behavior of most other encoders. When the parameter `SAC_PAR_MEAS_SYST_DIRECTION` is set to -1, the Rexroth MSM encoder behavior is identical to the other encoders.

NYCe 4000 supports MSM incremental encoders with 10^4 increment/revolution and 2^{20} increment/revolution, and MSM absolute encoders with 2^{17} increment/revolution and 2^{20} increment/revolution.

If you select the MSM encoder (absolute or incremental), only the 4 kHz and 8 kHz PVL frequency are allowed, see [chapter 3.7.4 "Frequencies" tab](#) on page 97. If the selected PVL frequency is set higher and you select the MSM encoder, the PVL frequency is adjusted and a notification message is displayed. The higher PVL frequencies are not shown in the drop-down list on the "Frequencies" tab as long as the MSM encoder is selected.

If you select an S0/S90, SinCos, EnDat2.1, EnDat2.2, or Hiperface encoder, the NYCeConfigurator displays the tab "Encoders" in the pane at the right hand side. Click the tab "Encoders" to set the relevant encoder parameters.

- **S00/S90 with index, S00/S90 without index encoder**

The S0S90 encoder is also known as A/B encoder, quadrature encoder and 3-channel digital incremental encoder. All drive modules except the NY4125 support cable break detection for the S0/S90 encoder on all 3 signals. The NY4125 supports cable break detection on the S0 and S90 signal only. If an S0/S90 encoder without index is used, you must connect the Index- and Index+ inputs of that axis of the drive module to EncGND to prevent a cable break detection error. See the NYCe 4000 Hardware System Manual for detailed information.

NYCeConfigurator

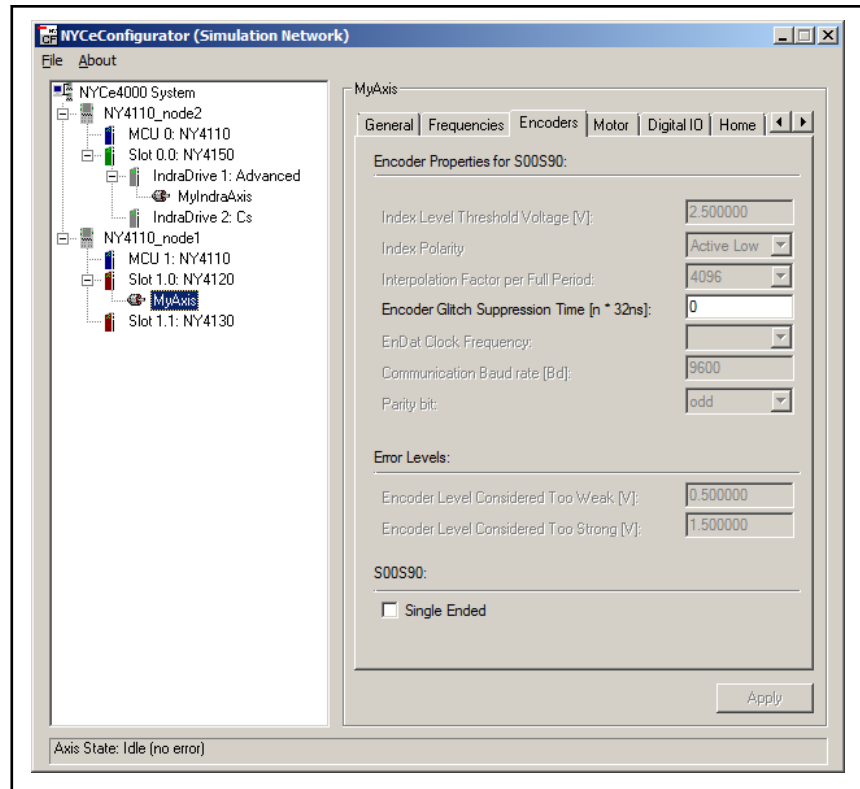


Fig. 3-27: NYCeConfigurator, axis level, "Encoders" tab for S0S90 encoder

Encoder Glitch Suppression

If long cables or cables with improper shielding are used to connect the encoder, position errors may occur due to encoder signal disturbances. In motor control systems these disturbances are typically large short duration spikes. The spikes can be suppressed with a digital filter and down-sampling the S0, S90, and Index encoder signals. The filtered signals only change if the input signal has the same value for 3 consecutive samples. The time between two samples is $n * 16 \text{ ns}$, where n is an integer number between 1 and 31. The number n is specified in the entry field "Encoder Glitch Suppression Time". The digital filter suppresses spikes and glitches up to $[n * 32\text{ns}]$. If n is set to 0, which is the default setting, the filter is disabled.

You can increase the glitch suppression time to improve the filtering, but this limits the maximum encoder (axis) velocity. At least 13 sample moments are required per half period time (which allows for two consecutive missed samples caused by a disturbance). Given the maximum velocity of an axis, you can determine the maximum allowed suppression time with the following formulas.

- For SinCos encoders: $1 / (2 * 13 * n * 16 * 10^{-9}) \text{ Hz}$
- For S0S90 encoders: $4 / (2 * 13 * n * 16 * 10^{-9}) \text{ inc/sec}$

See also the following graphs.

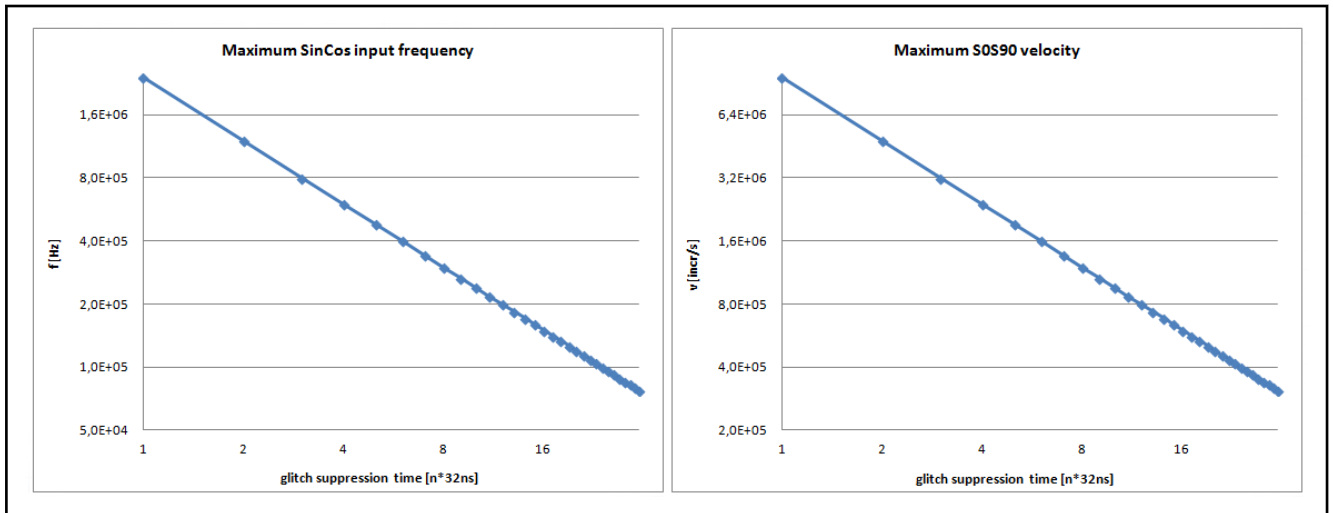


Fig. 3-28: Glitch suppression time selection vs. maximum encoder velocity

You can indicate whether the S0/S90 encoder is connected hardware-wise differential or single ended with the check box "Single Ended" in the "S0S90" section. Put a check mark in the box if the encoder is connected single ended. See the NYCe 4000 Hardware System manual for connection information. If a single ended S0/S90 encoder is selected, the cable break detection is disabled in software, and the cable alarm will never occur.

Secondary S0S90 encoder support

You can connect a secondary S0S90 encoder to the S0S90 counter inputs of the NY4120, NY4130, and NY4140 drive modules. This secondary counter can be used to keep track of the position measured by the secondary encoder. The secondary counter can be used as position source for commutation (BLAC and DC motors only) and position feedback control (PVL).

You can switch between primary encoder and secondary encoder by selecting the standard S0S90 encoder interface or the S0S90 counter as position source using the parameter SAC_PAR_ENCODER_SELECTION. The parameter has 2 possible values, defined by the enumeration SAC_ENCODER_SELECTION, SAC_PRIMARY_ENCODER and SAC_SECONDARY_ENCODER. The initial and default value of SAC_PAR_ENCODER_SELECTION is SAC_PRIMARY_ENCODER. The primary encoder is connected to the S0S90 inputs of the drive module, the secondary encoder is connected to the Dig5VIn inputs of the drive module. Only one of the 2 encoders can be used for feedback control and commutation. The selection of the other encoder is allowed in all axis states, except SAC_IDLE. Note that the parameter SAC_PAR_ENCODER_SELECTION is not saved in an XML file or flash memory.

With both encoders connected and using the same count direction configuration, you can switch between the primary and secondary encoder without discontinuity in the axis position. If needed, you can configure the count direction of the secondary encoder with the parameter SAC_PAR_SECONDARY_COUNTER_DIRECTION, so that both

NYCeConfigurator

primary and secondary encoder have the same count direction. Alignment and homing is possible with each encoder, although for homing some restrictions apply.

You can trace both encoders using the variables `SAC_VAR_AXIS_POS` and `SAC_VAR_SECONDARY_COUNTER_VALUE`. Keep in mind that the dimension of `SAC_VAR_AXIS_POS` is position units and the dimension of `SAC_VAR_SECONDARY_COUNTER_VALUE` is increments.

Restrictions

- As the secondary S0S90 encoder is connected to the Dig5VIn inputs, you cannot use a BLDC motor with 5V Hall sensors on axis0. See the NYCe 4000 Hardware System manual, chapter "Encoders".
- As the index mark is only supported on the primary encoder, all index-related home modes (`SAC_HOME_MODE_INDEX_XXX`) are not supported when the secondary encoder is selected.
- Marker functionality, both standard and S0S90, is not supported when the secondary encoder is used.
- Encoder glitch suppression is not supported for the secondary encoder.
- The primary and secondary encoder must have the same resolution.



Due to noise and thermal influences it is possible that position deviations of some increments may occur when changing the selected encoder. However, the axis is still said to be homed after a change of position encoder, because the accuracy is still sufficient for error checks such as software end switch checks. Advice is to implement a home strategy whenever you revert your position interface from the secondary encoder to the primary encoder.

- SinCos encoder

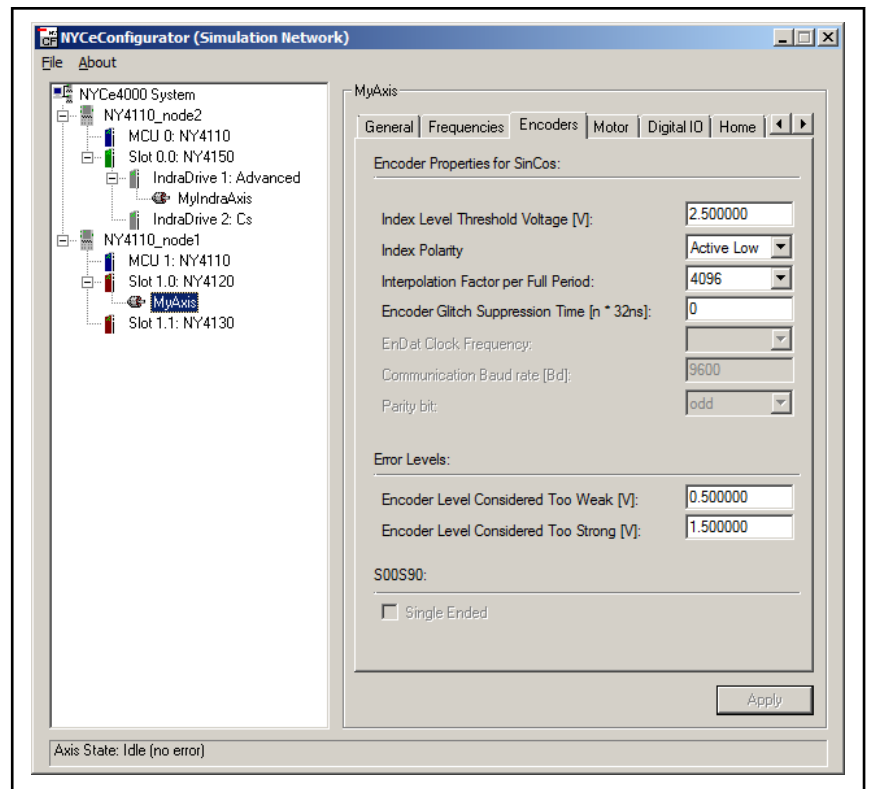


Fig. 3-29: NYCeConfigurator, axis level, "Encoders" tab for SinCos encoder

The "Index Level Threshold Voltage" is used for SinCos encoders only to perform the home mode "homing on index". The threshold level (in Volts) represents the level at which the index is latched from the SinCos reference signal. The drive module amplifies the signal with a factor 4, thus the index level threshold voltage can be set between 0 and +4 V.

The "Index Polarity" field selects the active level of the index signal.

The "Interpolation Factor per Full Period" parameter is used for the SinCos interpolation algorithm. You can set the "Interpolation factor" to the integer value 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 or 4096.

The "Encoder Glitch Suppression Time [n * 32ns]" field is explained in the description for the S0/S90 encoder. For a SinCos encoder the sine, cosine and index signal are de-glitched, if the setting is not 0.

Use the "Encoder Level Considered Too Weak" field to set the level (in Volts) at which the encoder input is considered too weak.

Use the "Encoder Level Considered Too Strong" field to set the level (in Volts) at which the encoder input is considered too strong.

- EnDat2.1 encoder

NYCeConfigurator

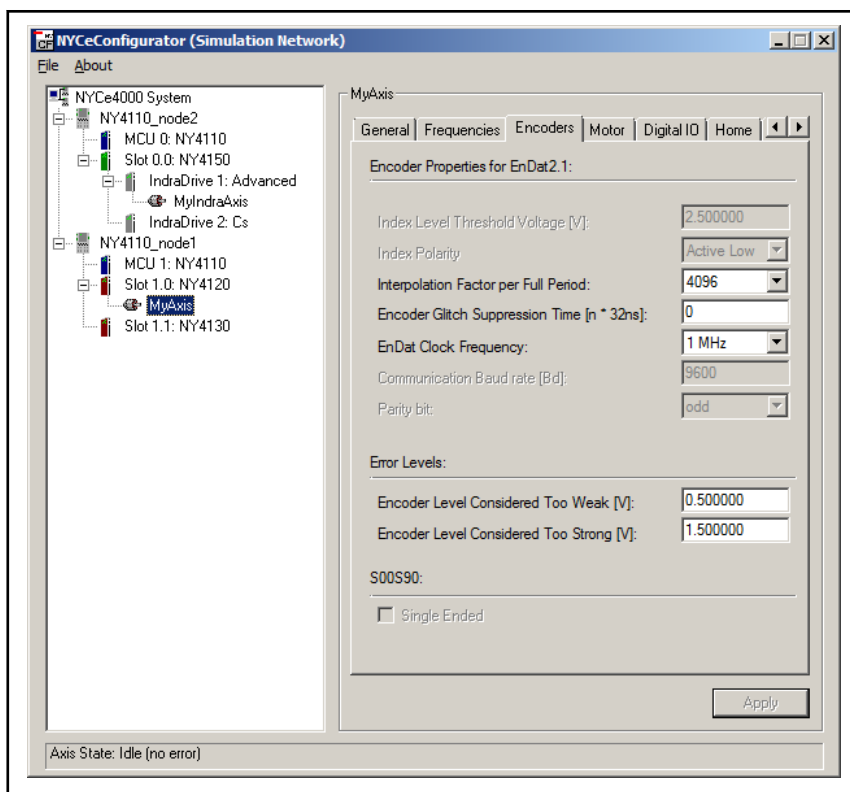


Fig. 3-30: NYCeConfigurator, axis level, "Encoders" tab for EnDat2.1 encoder

The "Interpolation Factor per Full Period" field is explained in the description of the SinCos encoder.

The "Encoder Glitch Suppression Time [n * 32ns]" field is explained in the description for the S0/S90 encoder. For an EnDat2.1 encoder the sine and cosine signal are de-glitched, if the setting is not 0.

For EnDat2.1 encoders the EnDat Clock Frequency (communication speed) can be set to 100 kHz, 500 kHz or 1 MHz.

The "Encoder Level Considered Too Weak" and "Encoder Level Considered Too Strong" fields are explained in the description of the SinCos encoder.

- **EnDat2.2 encoder**

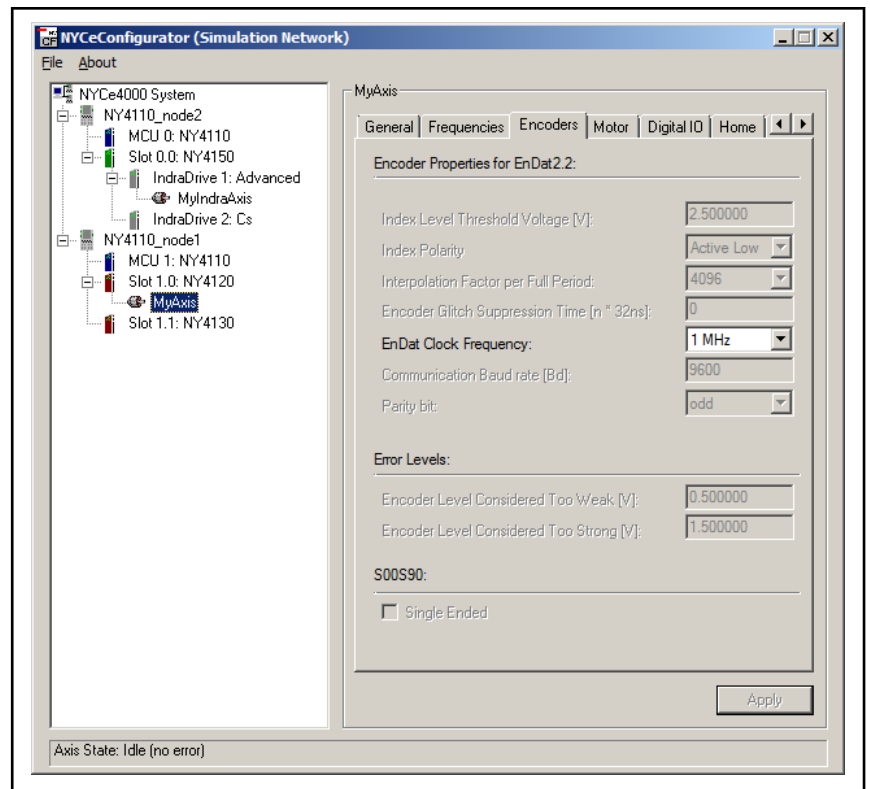


Fig. 3-31: NYCeConfigurator, axis level, "Encoders" tab for EnDat2.2 encoder

For EnDat2.2 encoders the EnDat Clock Frequency (communication speed) can be set to 1 MHz, 2 MHz or 4 MHz.

If you select the EnDat2.2 encoder, only certain PVL frequencies are allowed depending on the EnDat Clock Frequency, see [chapter 3.7.4 "Frequencies" tab](#) on page 97. If the selected PVL frequency is set higher and you select the EnDat2.2 encoder, the PVL frequency is adjusted and a notification message is displayed. The higher PVL frequencies are not shown in the drop-down list on the "Frequencies" tab as long as the EnDat2.2 encoder is selected.

- Hiperface encoder

NYCeConfigurator

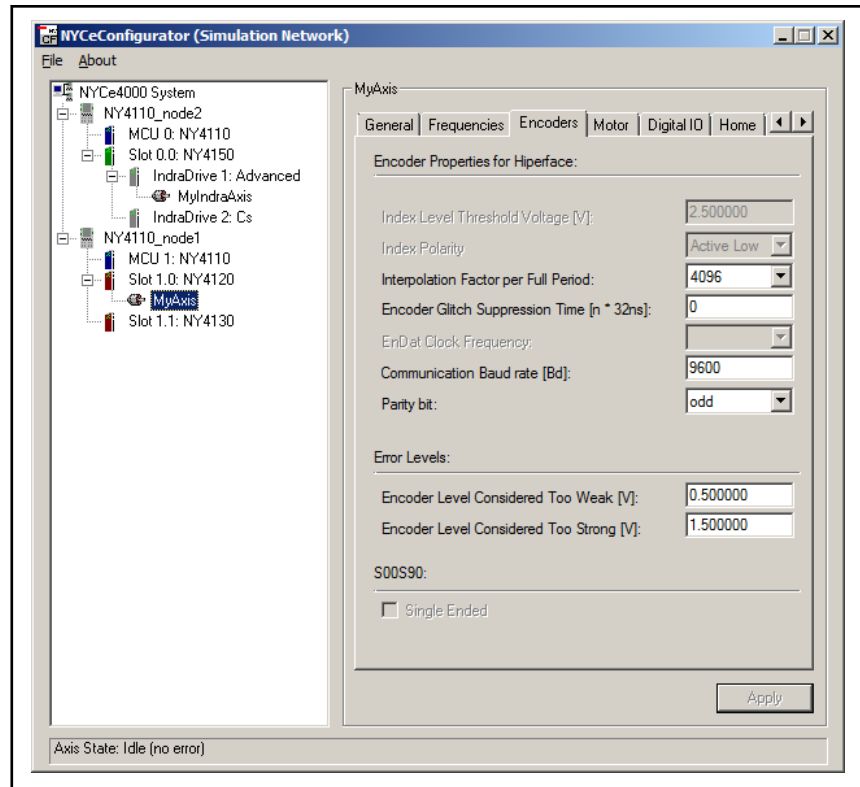


Fig. 3-32: NYCeConfigurator, axis level, "Encoders" tab for Hiperface encoder

The "Interpolation Factor per Full Period" field is explained in the description of the SinCos encoder.

The "Encoder Glitch Suppression Time [n * 32ns]" field is explained in the description for the S0/S90 encoder. For a Hiperface encoder the sine and cosine signal are de-glitched, if the setting is not 0.

The Communication Baud rate is the communication speed of the Hiperface encoder. The Communication Baud rate can be set to a value between 600 Baud and 38400 Baud. This depends on the type of Hiperface encoder. Default value for Rexroth Hiperface encoders is 9600 Baud.

The Parity bit is used for error detection in the Hiperface communication protocol. The Parity bit can be set to none, odd or even. Default setting for Rexroth Hiperface encoders is odd.

The "Encoder Level Considered Too Weak" and "Encoder Level Considered Too Strong" fields are explained in the description of the SinCos encoder.

Sanyo Denki absolute encoder

The Sanyo Denki encoder is supported on the NY4120, NY4125, and NY4140 drive module. The Sanyo Denki absolute encoder counts downward (position increases) when the shaft of the encoder rotates clockwise. This behavior is opposite to the behavior of most other encoders. When the parameter `SAC_PAR_MEAS_SYST_DIRECTION` is set to -1, the Sanyo Denki absolute encoder behavior is identical to the other encoders.

If you select the Sanyo Denki absolute encoder, only the 4 kHz and 8 kHz PVL frequency are allowed, see [chapter 3.7.4 "Frequencies" tab](#) on page

97. If the selected PVL frequency is set higher and you select the Sanyo Denki absolute encoder, the PVL frequency is adjusted and a notification message is displayed. The higher PVL frequencies are not shown in the drop-down list on the "Frequencies" tab as long as the Sanyo Denki absolute encoder is selected.

Determining increments per revolution for a SinCos based encoder

This is also known as the Pulses Per Revolution of an encoder (PPR) of a SinCos based encoder. The analog signal of the SinCos based encoder is converted to a digital value. The resolution of the ADC on the drive module is 12 bits, see the NYCe 4000 Hardware System manual. However, the lowest 2 bits of the ADC are discarded in the gateway. The used ADC resolution is 10 bits.

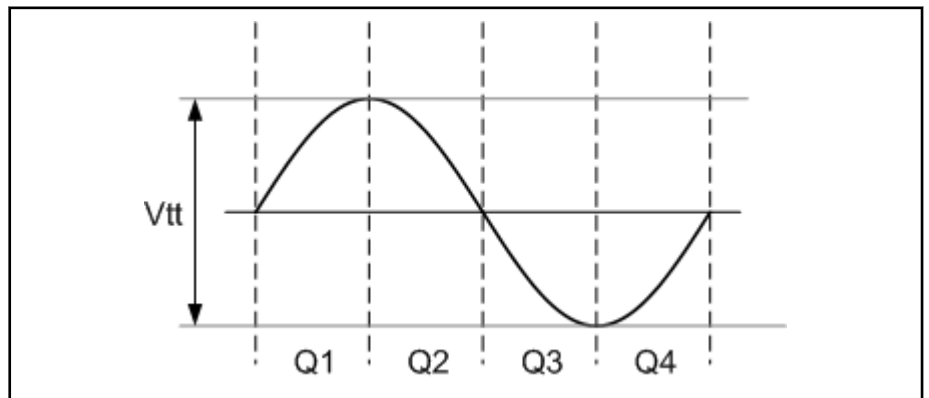


Fig. 3-33: Sine signal divided into 4 quadrants

The SinCos signal is divided in 4 quadrants. This means that the SinCos algorithm interpolates with the number of resolution bits for every quadrant of the Sine and Cosine signal. Thus:

$$\text{PPR} = 2^{\text{number of resolution bits}} * 4 * \text{number of SinCos line counts}$$

The term " $2^{\text{number of resolution bits}} * 4$ " is the Interpolation Factor per Full Period that you set in the "encoders" tab. Example: If the encoder connected to the motor shaft has 1000 line counts (1000 SinCos signals per revolution) and the Interpolation factor is 32 is used for the SinCos interpolation algorithm, the resolution of the encoder is 32000 PPR.

3.7.4 "Frequencies" tab

On the "Frequencies" tab you can see and choose, depending on the drive module type, the update frequency for the Position Velocity Loop (PVL), Current Control Loop (CCL) and Pulse Width Modulation (PWM) processes. The Setpoint update frequency, which is the sample frequency, is set at the node level.



The default PWM (pulse width modulation) frequency for a new created axis is 32 kHz. However, if the axis is configured on an IndraDrive, the PWM frequency is set default to 4, 8 or 16 kHz depending on the downloaded microware version into the NY4150 (microware4khz.mcr, microware8khz.mcr or microware16khz.mcr).

NYCeConfigurator

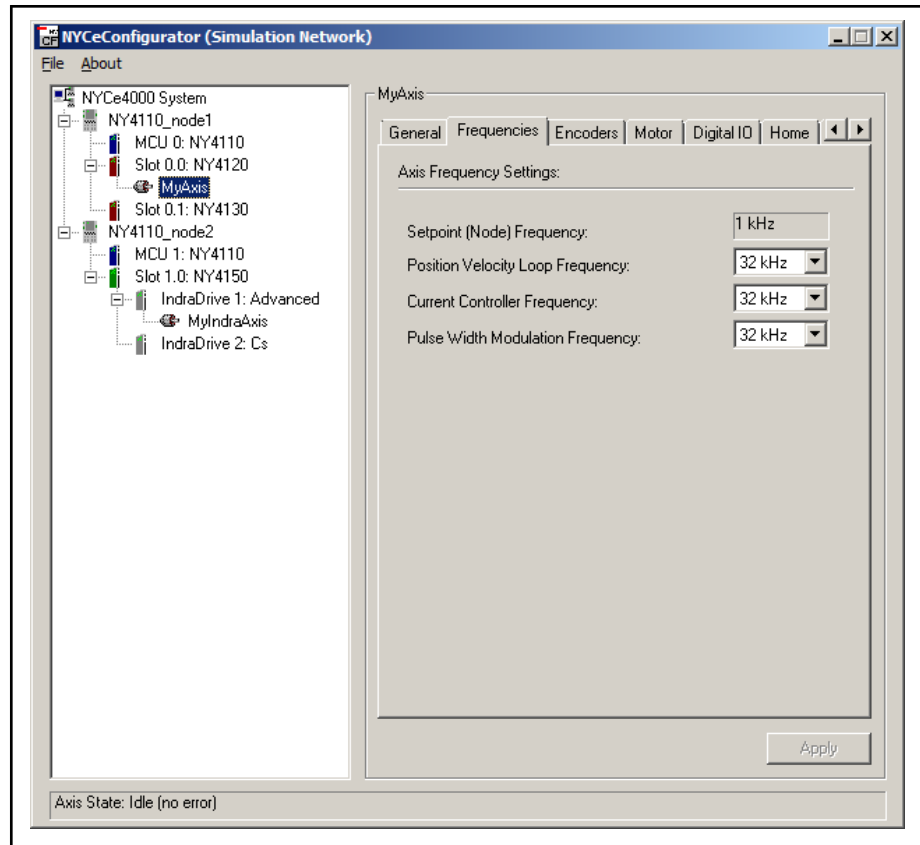


Fig. 3-34: NYCeConfigurator, axis level, "Frequencies" tab

The frequencies cannot be randomly chosen. The following rule applies.

$$f_{\text{PWM}} \geq f_{\text{CCL}} \geq f_{\text{PVL}} \geq f_{\text{node}}$$

tab. 3-4 "Frequencies that can be chosen" on page 98 indicates the frequencies that are available. They may not be the optimal frequencies.

	NY4120	NY4125	NY4130	NY4140	NY4150
Setpoint/Node frequency	1, 2, 4, 8 kHz	1, 2, 4, 8 kHz	1, 2, 4, 8 kHz	1, 2, 4, 8 kHz	1, 2 kHz
Position – velocity loop (PVL) frequency	4, 8, 16, 32 kHz	4, 8, 16, 32 kHz	4, 8, 16, 32 kHz	4, 8, 16, 32 kHz	4 kHz
Current control loop (CCL) frequency	4, 8, 16, 32 kHz	not available	not available	4, 8, 16, 32 kHz	8, 16 kHz
Pulse width modulation (PWM) frequency	16, 32, 64, 96 kHz	16, 32, 64, 96 kHz	not available	16, 32 kHz	4, 8, 16 kHz

Tab. 3-4: Frequencies that can be chosen

When setting frequencies, keep the following in mind.

- The Setpoint (Node) Frequency setting can only be changed at the node level (see [chapter 3.4 "Configuration at the node level" on page 62](#)) and the setting depends on the number of axes connected.
- The NY4150 supports 8 IndraDrives at a Setpoint (Node) Frequency of 1 kHz, but supports only 5 IndraDrives at a Setpoint (Node) Frequency of 2 kHz.

- Allowed PVL frequencies for Rexroth MSM encoders are 4 and 8 kHz.
- Allowed PVL frequencies for Sanyo Denki absolute encoders are 4 and 8 kHz.
- Maximum PVL frequency for EnDat2.2 encoders depends on the selected EnDat Clock frequency and are listed in [tab. 3-5 "Maximum PVL frequencies EnDat2.2 encoders"](#) on page 99. EnDat2.1 encoders do not have this limitation.

EnDat clock frequency	Maximum PVL frequency
1 MHz	4 kHz
2 MHz	8 kHz
4 MHz	16 kHz

Tab. 3-5: Maximum PVL frequencies EnDat2.2 encoders

- If you use a sensor axis in combination with a Rexroth MSM incremental, Rexroth MSM absolute, EnDat2.2, or Sanyo Denki absolute encoder, make sure the PVL frequency is valid.
- Pulse width modulation frequency (PWM) is used to generate the motor currents with the NY4120, NY4125 and NY4140. The PWM cycle time $1/f_{PWM}$ should at least be 3 times smaller than the motor time constant ($T_e = L / R$, consult motor manufacturer datasheet). Setting the PWM frequency too high will cause excessive power dissipation in the drive module.
- The rule that the PWM frequency must be higher than or equal to the CCL frequency does not apply for an IndraDrive-based axis. The following combinations of PWM, CCL and PVL frequencies are valid for IndraDrive C and IndraDrive Cs. These frequencies can be set for an IndraDrive C with the NYCe 4000 software and tools. The PWM frequency for an axis configured on an IndraDrive Cs is fixed to 4 kHz, 8 kHz or 16 kHz defined by the version of the microware downloaded to the NY4150, and cannot be changed with the NYCe 4000 software or tools.

PWM frequency	CCL frequency	PVL frequency
4 kHz	8 kHz	4 kHz
8 kHz	16 kHz	4 kHz
16 kHz	16 kHz	4 kHz

Tab. 3-6: Valid PWM, CCL and PVL frequency combinations for IndraDrive C and IndraDrive Cs

3.7.5 "Commutation" tab (not for IndraDrive)

When you select a BLAC, BLDC or a Stepper motor, the NYCeConfigurator will display a new tab called "Commutation" in the pane at the right hand side.

BLAC Commutation

NYCeConfigurator

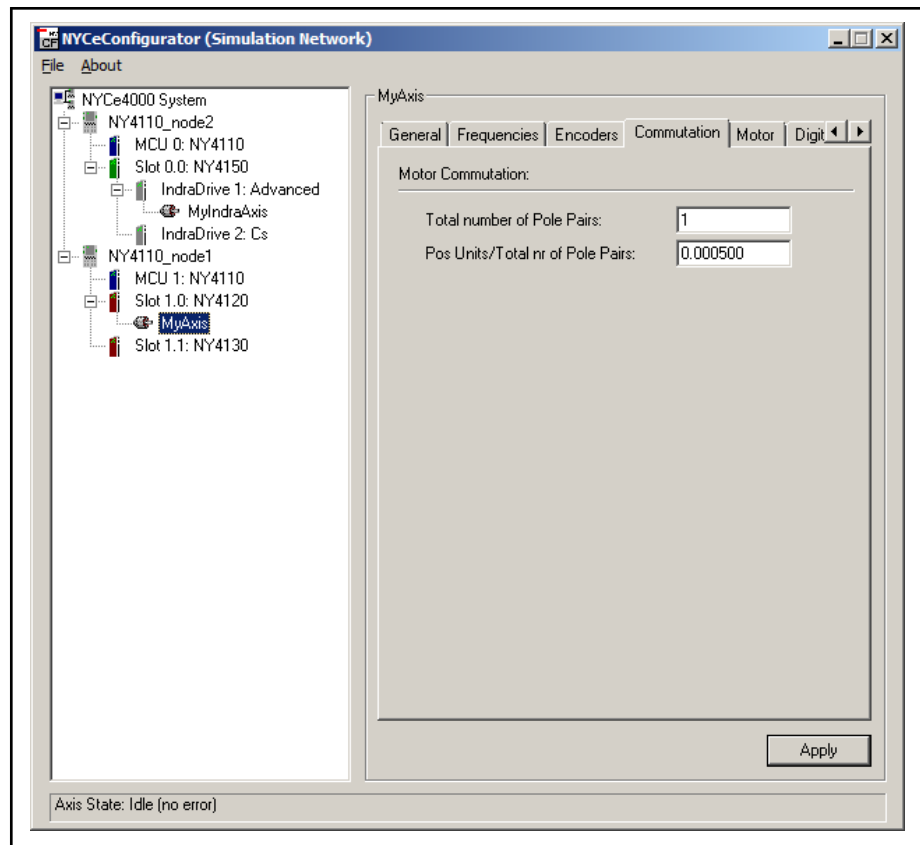


Fig. 3-35: NYCeConfigurator, axis level, "Commutation" tab for BLAC motor

- The total number of pole pairs can be acquired from the motor manufacturer datasheet. When using linear motors, this parameter should be set to 1.
- The number of position units per total number of pole pairs depends on the situation. When using a rotary BLAC motor with revolution as unit, the parameter should be set to 1. In general, when using a rotary BLAC motor, with "x" encoder increments for one position unit, and "y" encoder increments for a full revolution, results in a position units per total number of pole pairs equal to "y" divided by "x". For a linear BLAC motor this parameter should be equal to the number of increments between 2 adjacent north poles of the motor magnets divided by the number of position units between the 2 north poles of the motor magnets (the distance between adjacent North pole pairs should be always the same).

BLDC Commutation

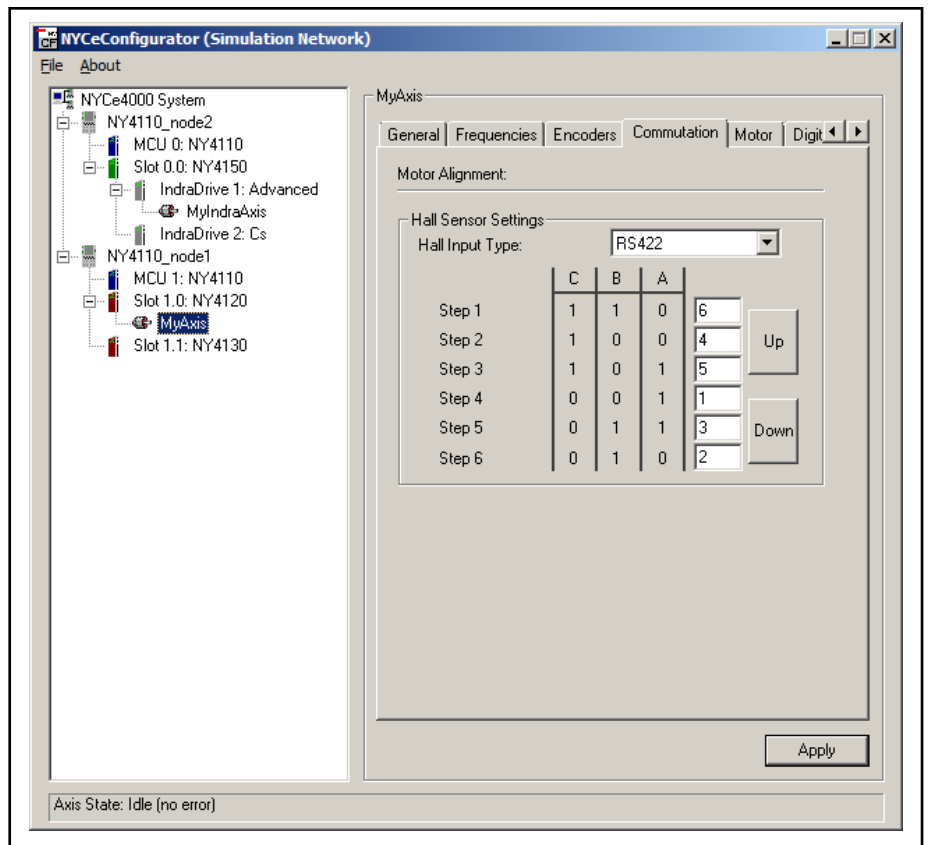


Fig. 3-36: NYCeConfigurator, axis level, "Commutation" tab for BLDC motor

- You can choose for the Hall input type between the DigIn inputs and the RS422 inputs. The DigIn Hall input uses the 24V digital inputs to read the Hall sensors in the motor. The inputs are DigIn0, DigIn1, and DigIn2 for the first axis connected to the drive and DigIn4, DigIn5, and DigIn6 for the second axis connected to the drive. The RS422 inputs are the 5V Hall inputs. These inputs are Dig5VIn0a, Dig5VIn0b and Dig5VIn0c for the first axis connected to the drive, and Dig5VIn1a, Dig5VIn1b and Dig5VIn1c for the second axis connected to the drive.



If an encoder is configured that uses one or more of the Dig5VIn inputs of a specific drive module, it is not possible to use a BLDC motor on that same drive module.

- The commutation table is initialized with the data shown in [tab. 3-7 "Hall sensor table" on page 101](#). This table represents the physical orientation of the Hall sensors relatively to the stator poles of the BLDC motor needed for six step commutation. The table shows a common Hall sensor lay-out for BLDC motors. Most motor manufacturers provide tables like this with the motor specifications.

	C	B	A	
Step 1	1	1	0	6
Step 2	1	0	0	4
Step 3	1	0	1	5
Step 4	0	0	1	1

NYCeConfigurator

	C	B	A	
Step 5	0	1	1	3
Step 6	0	1	0	2

Tab. 3-7: Hall sensor table

Stepper Commutation

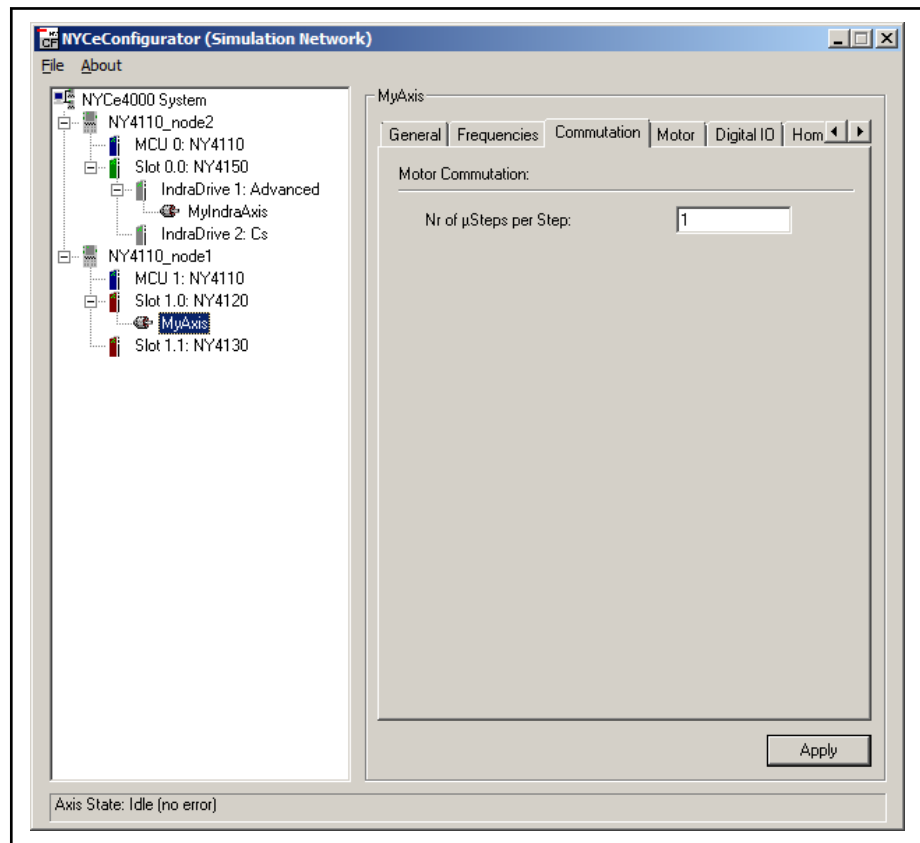


Fig. 3-37: NYCeConfigurator, axis level, "Commutation" tab for stepper motor

With the Nr of μ Steps per full step, a user defined number of electrical equilibrium positions are created between each 2 successive physical step positions. This increases the stepping resolution of the stepper motor (in practice, increasing the number of μ Steps above 4, will not result into a mechanical increase of the step resolution). The number of μ Steps can be any number.

3.7.6 "Motor" tab

The "Motor" tab only enables the user to supply the saturation level of the controller. The unit of this level depends on the unit defined via the controller output resolution (see [chapter 3.7.2 "General" tab](#) on page 83). Supplying a saturation level is important to protect the motor that is being used from damaging current levels (or voltage levels for the NY4130). Consult the motor manufacturer datasheet before entering a value for this parameter. (For example, the nominal and peak currents of the motor when the controller output is set to the unit Amperes or the nominal and peak torques of the motor when the controller output is set to the unit Newton meters).

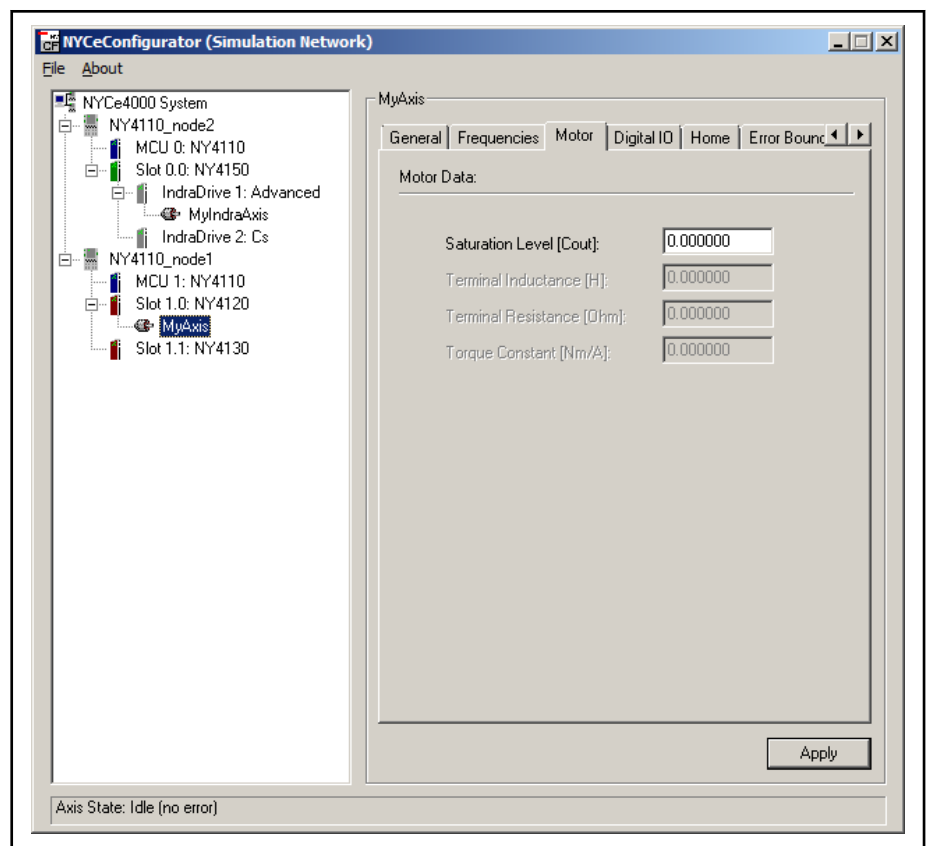


Fig. 3-38: NYCeConfigurator, axis level, "Motor" tab

If the controller output value exceeds the defined saturation level, the axis on a drive module (NY4120, NY4125, NY4130, or NY4140) is limited to the specified saturation level. No warning or error is reported if the controller output value is limited. However, if the axis is on an IndraDrive, and the output value exceeds the defined saturation level, the IndraDrive behavior is as follows.

- If the PVL is on the IndraDrive, the saturation limit value is **not** used, thus the controller output value is **not** limited. No warning or error is reported.
- If the PVL is on the MCU, the saturation limit value is used, thus the controller output value is limited to the specified saturation level. No warning or error is reported.



If a BLAC motor is connected to a PWM drive module (NY4120, NY4125, or NY4140) a torque ripple will occur if the required servo voltage level is not sufficient. To avoid this torque ripple you must set the saturation level at 87.6% ($\frac{1}{2}\sqrt{3}$) of the servo voltage for a BLAC motor connected to an NY4125 drive module. For a BLAC motor connected to the NY4120 or NY4140 drive module you must set the saturation level of the current controller output (see [chapter 4.9 "Current control parameters" on page 138](#)), because "saturation level" on the NY4120 and NY4140 is not a voltage level, but a current level.

NYCeConfigurator

3.7.7 "External Stepper" tab

On the "External Stepper" tab you can set

- the minimum on and off time of the "pulse" signal,
- the delay time after the change of the "direction" signal before the next step pulse can be generated,
- and the active state for the "pulse" and "direction" signals

that control the external stepper drive hardware. The smallest time that can be set for the "pulse" on time, "pulse" off time, and "direction" delay time is 0.1 μ s.

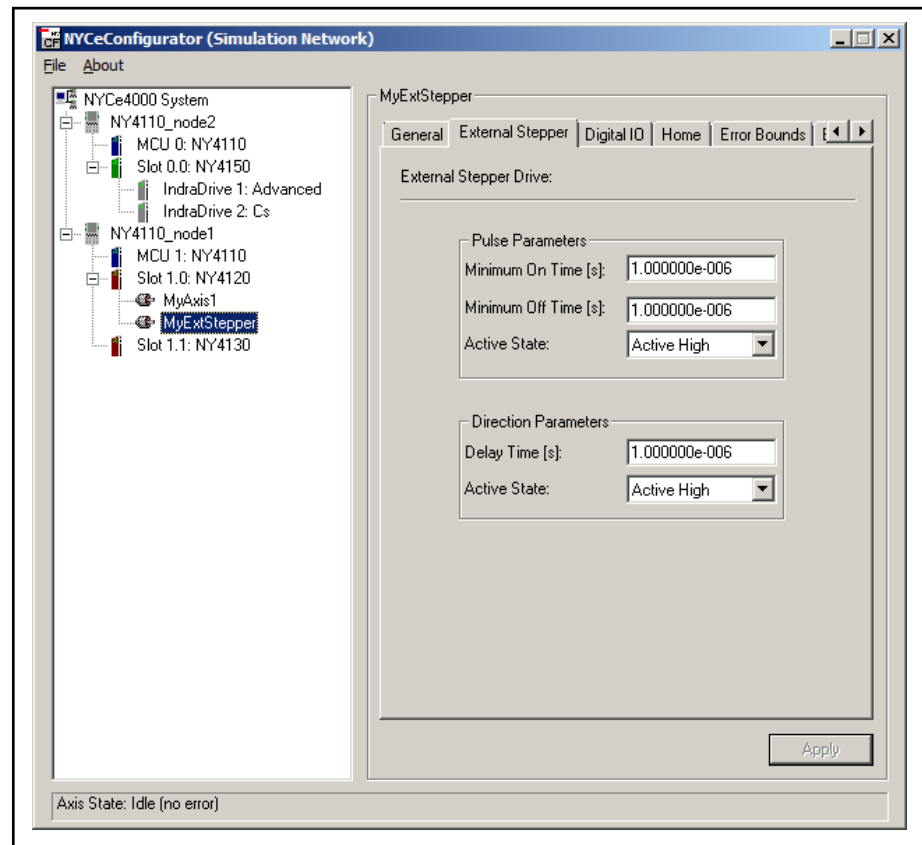


Fig. 3-39: NYCeConfigurator, axis level, "External Stepper" tab

Note that the "pulse" signal is always allocated to fast digital output FastOut0, and the "direction" signal is always allocated to fast digital output FastOut1.

The maximum number of steps per second of the stepper motor depends on the following timing parameters.

- Maximum switch time of the fast digital output ($T_{sw} = 2 \mu$ s)
- Required pulse "on" time ($T_{on} \mu$ s)
- Required pulse "off" time ($T_{off} \mu$ s)
- Required direction delay time ($T_{delay} \mu$ s)

The minimum pulse period is $P_{pulse} = T_{sw} + T_{on} + T_{sw} + T_{off} \mu$ s.

The time that must be waited before the first step pulse can be generated, if the "direction" signal is changed, is $T_{firstpulse} = T_{sw} + T_{delay} \mu$ s.

3.7.8 "Digital I/O" tab

On the "Digital I/O" tab you can classify the different I/O ports with their functions. For example, the digital input 0 on a drive inserted in slot 1 is connected to a positive limit switch for one of the axes connected to the drive. Selecting the function "pos limit switch", with the function parameters: drive 1 and digital I/O number digin0, will define this input as the positive limit switch for the axis.

If a function is specific for the inputs, the digital I/O list will only display the input ports, and for output function the digital I/O list will only display the output ports.

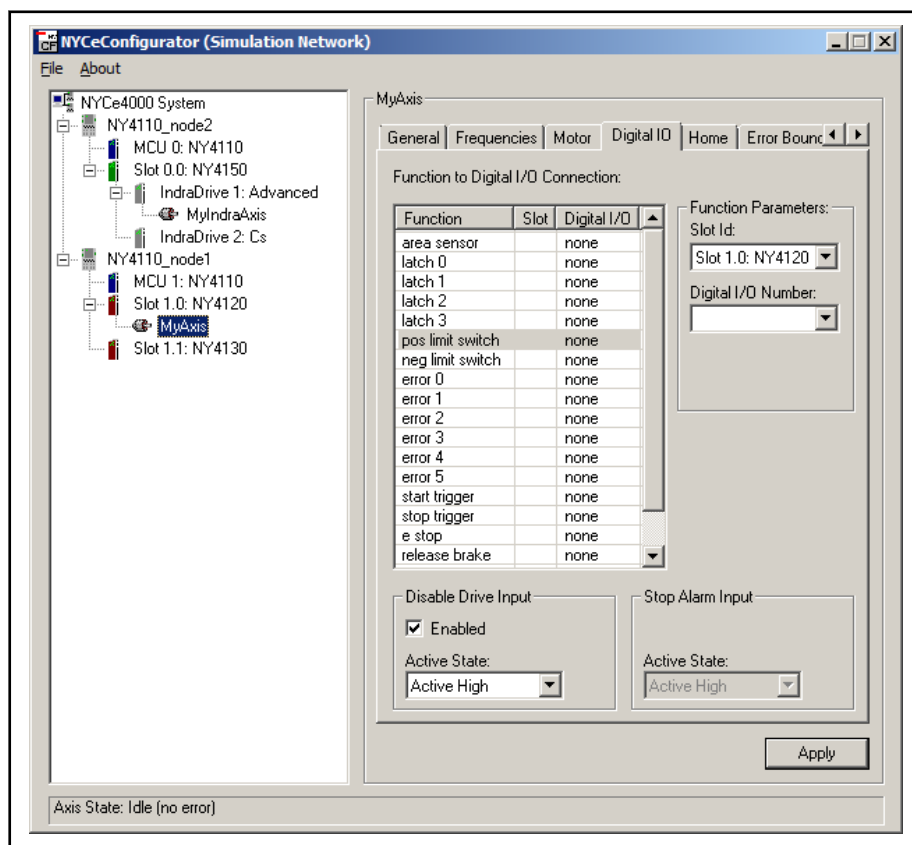


Fig. 3-40: NYCeConfigurator, axis level, "Digital I/O" tab

On the "Digital I/O" tab you can also configure the Disable Drive and Stop Alarm inputs, if available for the selected axis, see [fig. 3-40 "NYCeConfigurator, axis level, "Digital I/O" tab" on page 105](#). For example, an axis on an NY4120 drive module only supports the Disable Drive input, whereas an axis on an NY4130 drive module only supports the Stop Alarm input.

The Disable Drive input is default disabled and is enabled when a checkmark is put in the "Enabled" checkbox. When the Disable Drive input is enabled you can define the active signal level with the options of the "Active State" drop-down box.

The Stop Alarm input cannot be disabled. The only option is the setting of the active signal level with the options of the "Active State" drop-down box.

When configuring digital I/O, keep the following in mind.

NYCeConfigurator

- Remember that the active levels of the inputs and outputs are defined at the drive level. Before you assign a function to an I/O port, make sure that the active level is set correctly. This especially applies to the Disable Drive input, else the axis will immediately enter the error state.
- Latches can only be set if the input used for latching has been configured as a latch at the drive level.
- If an encoder is configured which uses digital inputs (differs per encoder type), these inputs are no longer available for configuration, even though NYCeConfigurator still displays these inputs. You cannot define the digital inputs used by the encoder for any other functionality.

To configure function I/O on the I/O of an IndraDrive, for example the Emergency Stop input (E_STOP), you must select the IndraDrive from the "Slot Id" drop-down box, see [fig. 3-41 "Configuration of IndraDrive function I/O" on page 106](#). If a latch is configured on an IndraDrive, you must also select the edge type.

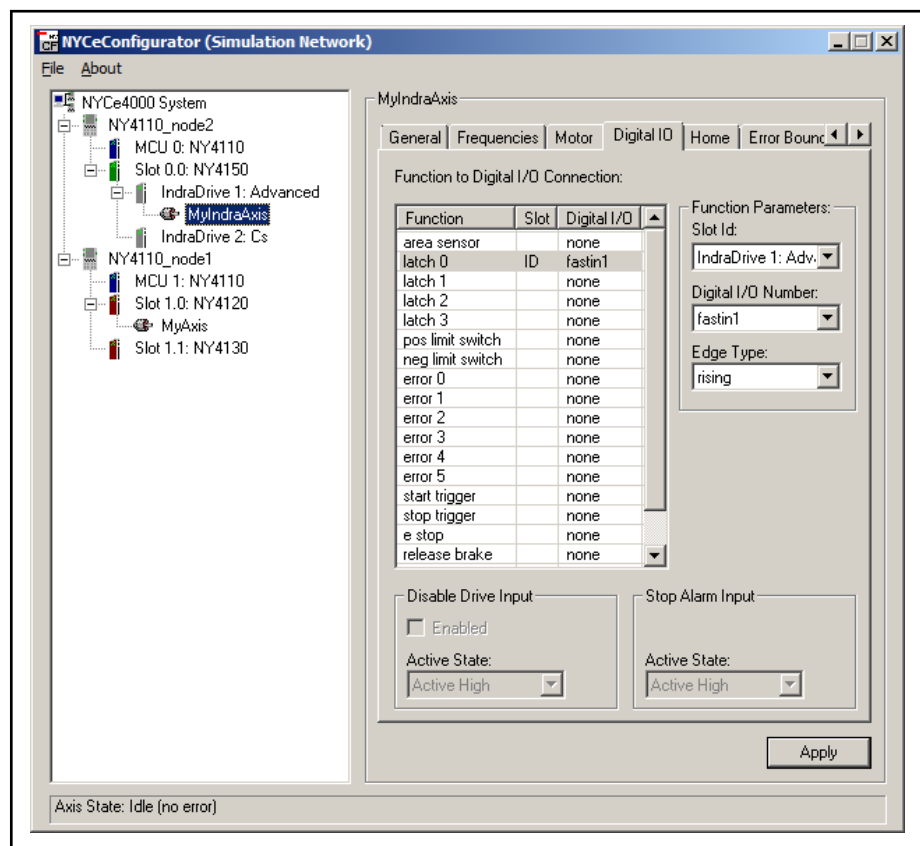


Fig. 3-41: Configuration of IndraDrive function I/O

3.7.9 "Sensor supply" tab

If NYCeConfigurator is used in an LMS with IndraDrive Cs, an additional tab "Sensor Supply" is available. On this tab you specify the slot identifier and the analog input port identifier to which the sensor is connected.

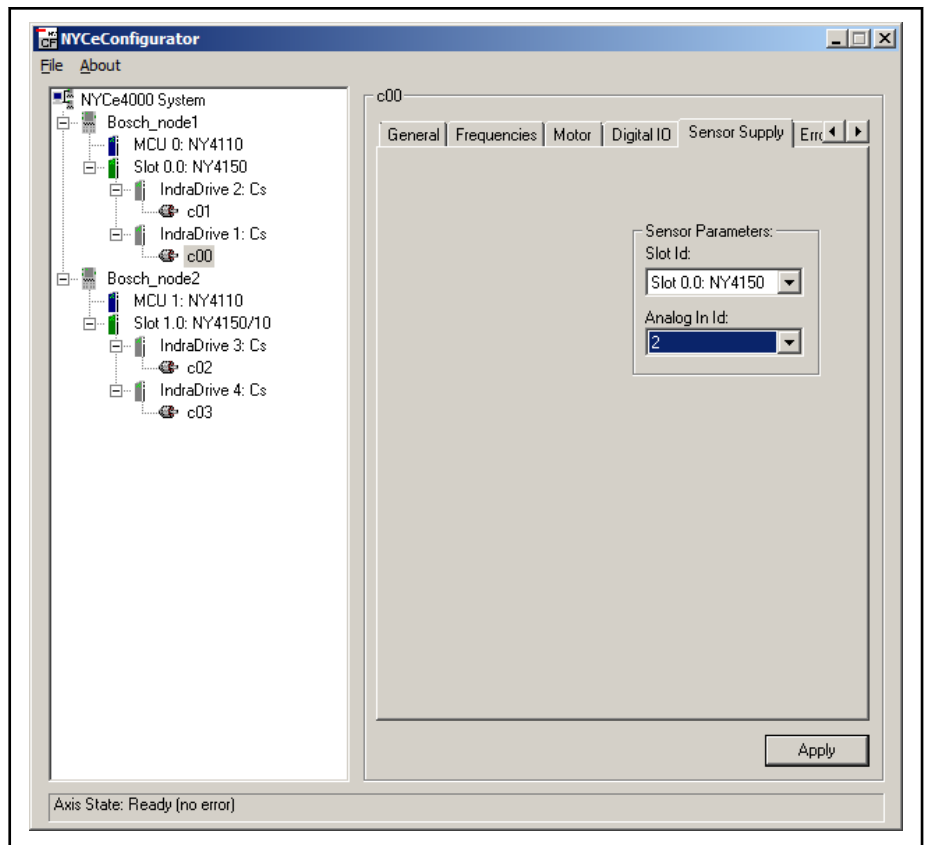


Fig. 3-42: NYCeConfigurator, axis level, LMS IndraDrive Cs "Sensor Supply" tab

3.7.10 "Home" tab

The homing settings of an axis can be set with the parameters of the "Home" tab. The home mode used for the axis is set by the "Modus" drop-down box. The entry fields below the "Modus" drop down box that you can set depend on the selected home mode. Details about the homing options and the parameters are described in the NYCe 4000 Software User Manual.

NYCeConfigurator

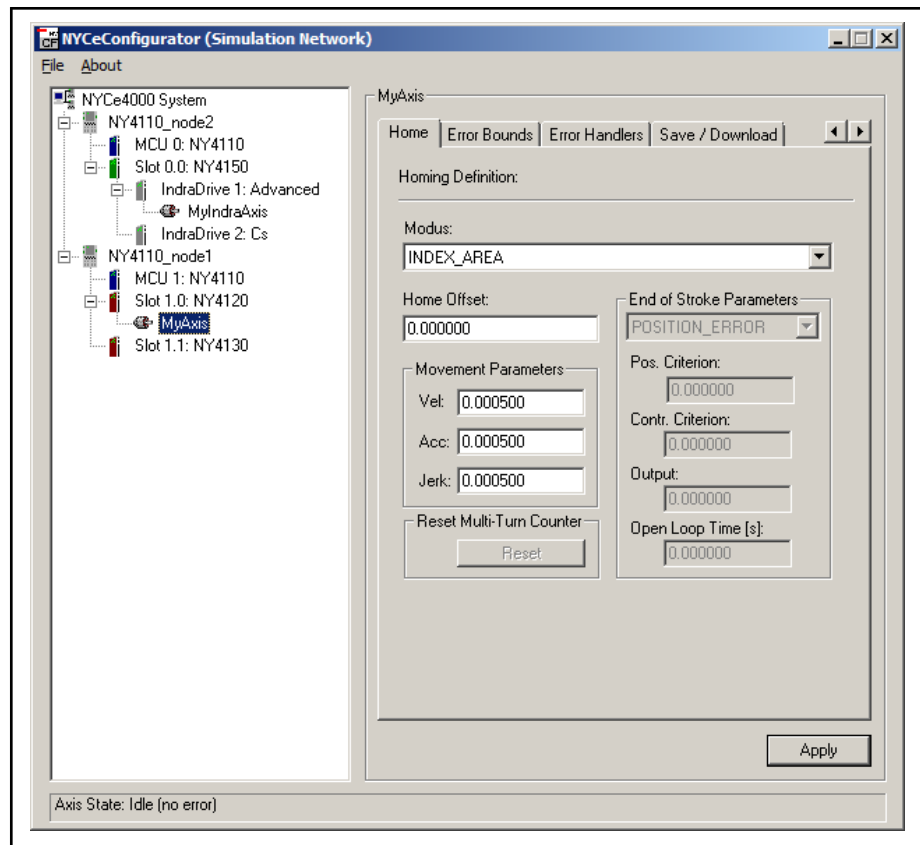


Fig. 3-43: NYCeConfigurator, axis level, "Home" tab

If a Rexroth MSM absolute encoder or Sanyo Denki absolute encoder is configured, the multi-turn position information of the encoder can be cleared with the "Reset" button in the "Home" tab. Resetting the multi-turn position data will not affect the absolute single-turn position data.

3.7.11 "Error Bounds" tab

In the "Error Bounds" tab a series of parameters can be specified which define the safe limits and actions for the axis behavior.

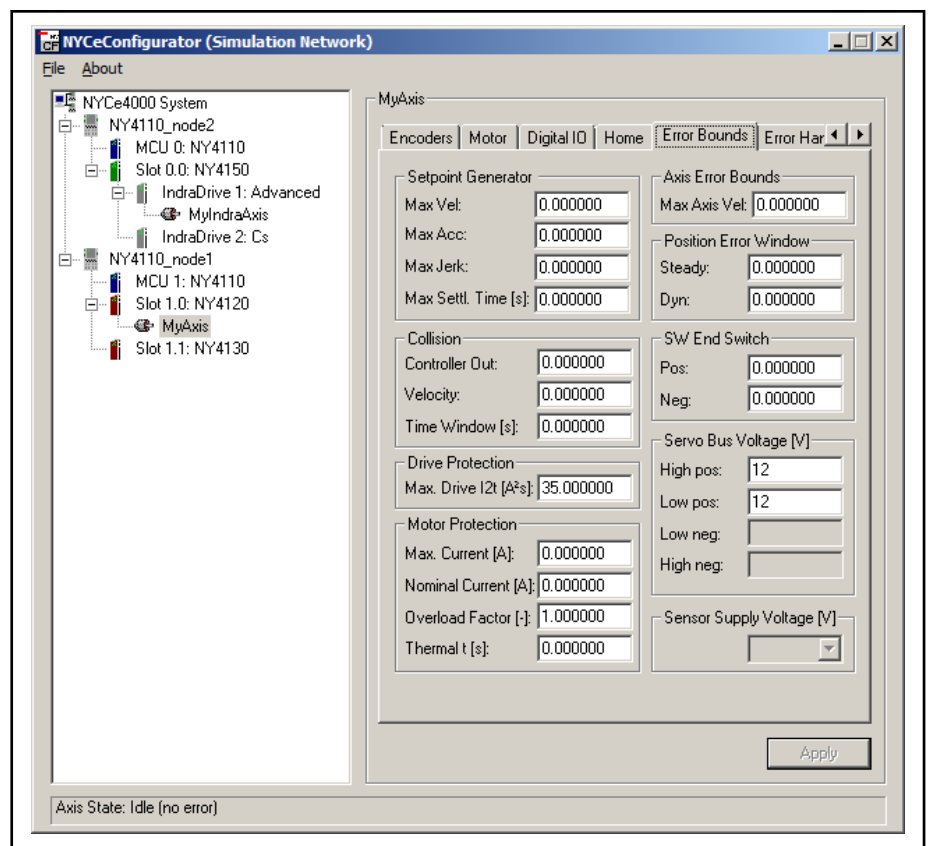


Fig. 3-44: NYCeConfigurator, axis level, "Error Bounds" tab

The parameters that can be defined are:

- The speed, acceleration, jerk and settling time limits for the setpoint generator. Setting these parameters to 0 will turn these limits off.
- The collision controller out, velocity and time window parameters define the collision detection criterion. A collision is detected if the actual axis velocity is below the velocity parameter, and the actual controller out is higher than the controller output parameter for a time span defined by the time window parameter.
- The "Drive Protection" box sets the maximum power dissipation allowed on the drive module.
- With the parameters in the "Motor Protection" box you can set the maximum current, nominal current, overload factor and thermal time constant for the motor. The overload factor defines the temporary allowed current through the motor higher than the specified nominal current. If the overload factor is set to 1.1, then a motor current of 1.1 times the nominal current is allowed. Typical value for the overload factor is 1.0. The overload factor is most likely not higher than 1.1, or perhaps 1.2. Related to the overload factor is the thermal time constant, which is a measure for the warming up of the windings and the housing of the motor. Note that the thermal time constant of a winding is lower than the thermal time constant of a housing. Make sure that you specify the correct thermal time constant from the data sheet of the motor manufacturer.

Note that this protection mechanism does **not** guarantee any safety and can **not** replace any fuses in drive power connections.

NYCeConfigurator

This protection is only available for motors connected to the NY4120 or NY4140 drive module, or the NY4150 SERCOS III Master module. The protection is not supported for virtual axes, sensor axes, external drives, and motors connected to the NY4125 or NY4130 drive module. If the nominal motor current or motor thermal time constant is set to 0, the protection is disabled.

- The maximum velocity of the axis.
- In the position error window, both the maximum steady state error (the maximum allowed position error when the axis is not moving) and the maximum dynamic position error (the maximum allowed position error when the axis is moving) can be defined. Setting these parameters to 0 will disable the position error check.
- The software (SW) end switches, to define virtual position limits for the axis.
- The servo bus voltage, to supply the node with information about the voltage on the drive module. It also expects the user to supply the maximum and minimum bus voltages, to protect the drive from damaging voltages during operation.

The "High pos" and "Low pos" fields define the maximum and minimum bus voltage for the positive power supply. The "High neg" and "Low neg" fields define the maximum and minimum bus voltage for the negative power supply.
- With the "Sensor Supply Voltage [V]" field you can set the sensor supply voltage for the LMS-MUX. For the LMS-MUX NY4960/10 the sensor supply voltage is 5 V, and for the NY4960/20 the sensor supply voltage is 15 V.

3.7.12 "Error Handlers" tab

The "Error Handlers" tab shows a list of all possible errors, and enables the user to choose a handler for the error. The execution of the error handler is the automatic response from the NYCe 4000 system when one of the errors occurs. The NYCe 4000 Software User Manual has a complete list of all possible errors and the default and minimum severity response of the system, when one of these errors occurs. You can change the error handler of an error by selecting the error code displayed in the list, and then setting the error handler in the selector box above the error handlers list.

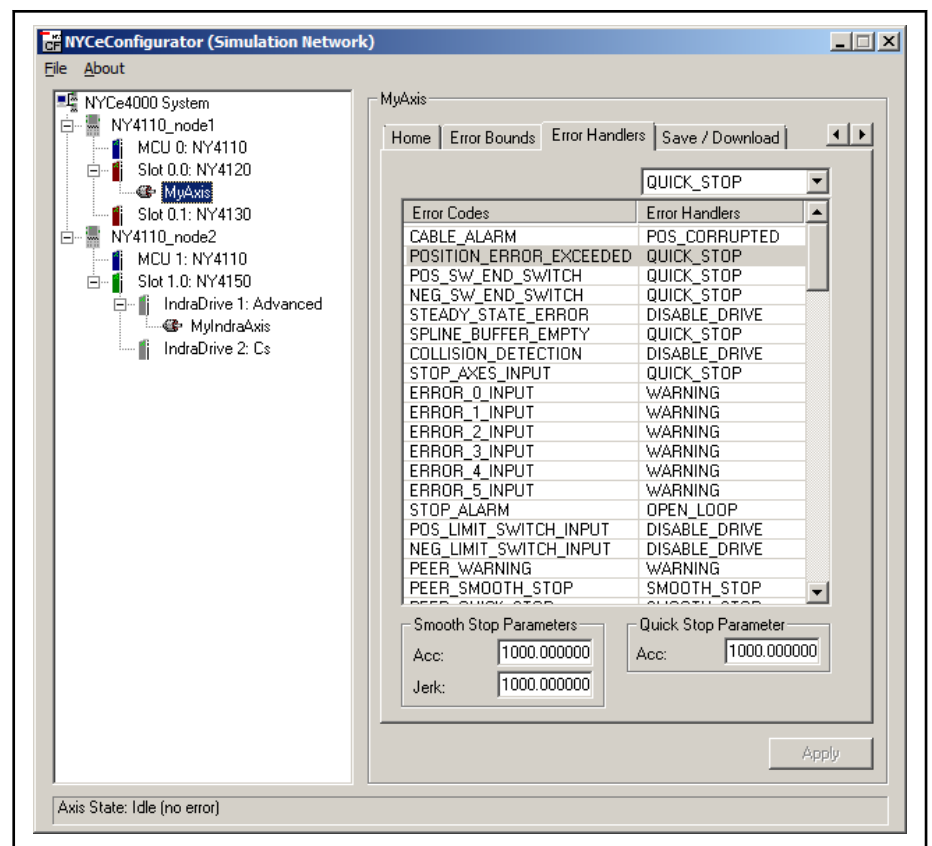


Fig. 3-45: NYCeConfigurator, axis level, "Error Handlers" tab

The Service input error handler must be defined by the application. When the Service input is activated other constraints for velocity, acceleration, jerk and controller output are set, but also other error handlers are activated if a constraint is violated. See also the chapter "Error and Service Mode Handling" in the NYCe 4000 Software User Manual.

Each error handler has a defined minimum response. Choosing a lower severity response than the defined minimum severity response for an error handler will result in an error message.

Further, you define the parameters for the smooth stop and quick stop movement on this tab.

- The smooth stop acceleration and jerk value. The smooth stop is a 3rd order stop. The smooth stop is a more subtle way of setting the axis to a halt compared with the quick stop.
- The acceleration value for a quick stop. The quick stop is one of the predefined actions the node can make when an axis detects an error or a setting exceeds a defined maximum or minimum value. As the name suggests, the quick stop performs a fast (2nd order) profile stop when called via software calls or via error handlers.

3.7.13 "Save / Download" tab

The last tab of the axis configuration, "Save / Download", can be used to save the created or changed axis to a file or the flash memory, or to initialize an axis from a file or the flash memory.

NYCeConfigurator

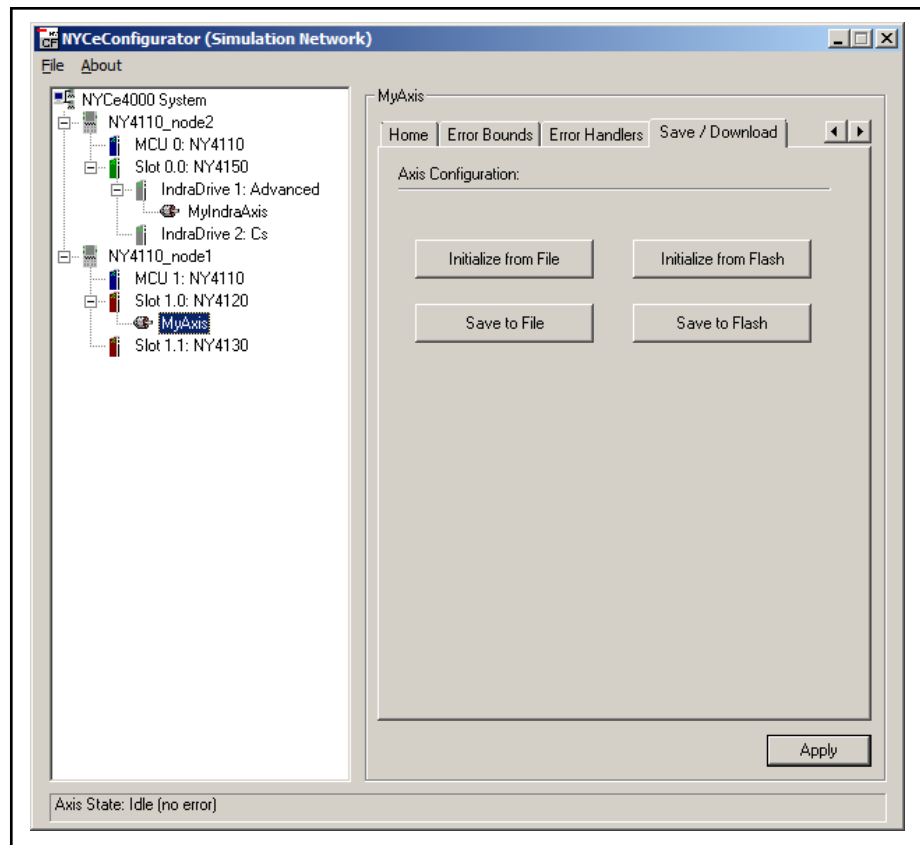


Fig. 3-46: NYCeConfigurator, axis level, "Save / Download" tab

If you load a configuration from file and, for some reason, that configuration misses a parameter or contains a parameter with an invalid value, NYCeConfigurator displays a message box with a warning or error for every missing parameter or parameter of which the value is out of the defined valid range. If you save the configuration to file or flash memory, missing parameters are added with their default value and parameters with an invalid value are saved with their default value.

3.8 Miscellaneous

When you click the icon at the left in the title bar, the popup with the standard Windows functions (for example "Move") appears. The last entry of the popup ("About NYCeConfigurator") displays the "About NYCeConfigurator" dialog window. This dialog window is also displayed when you click the "About" button in the menu bar.

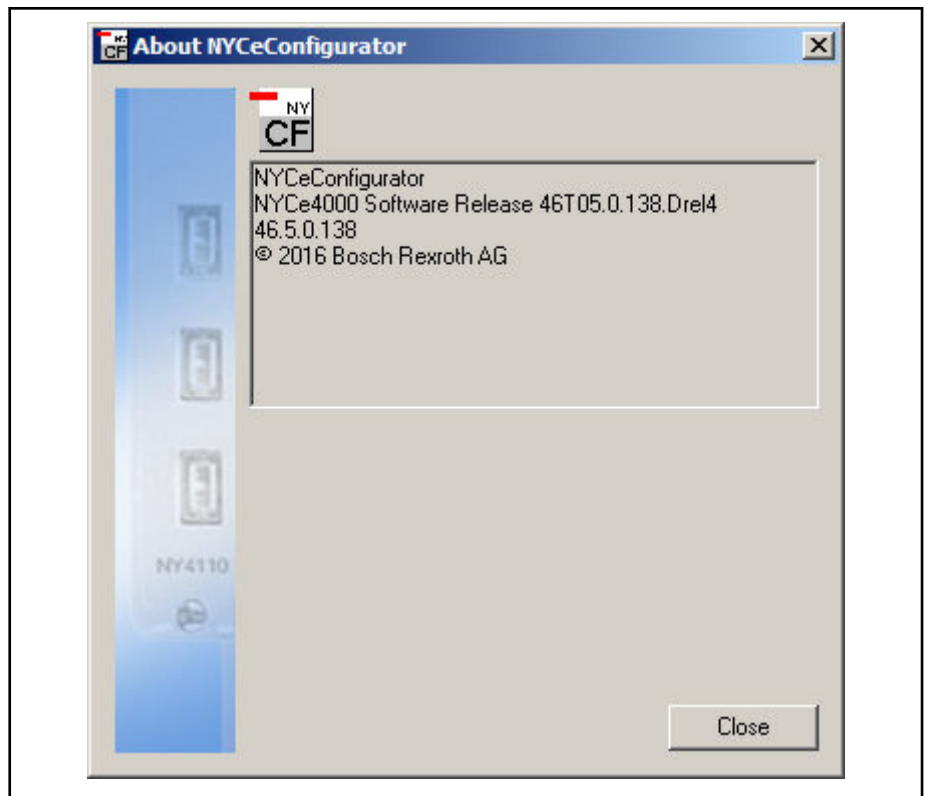


Fig. 3-47: The "About NYCeConfigurator" dialog window

See [chapter 1.3 "NYCe 4000 product and file version information"](#) on page 5 for an explanation of the product and file version numbers.

4 NYCeTuner

4.1 Introduction

You can use the NYCeTuner tool to setup and tune the axes of a system.

The NYCeTuner has 2 working modes, called the "wizard" mode and the "expert" mode. When you start the NYCeTuner the first time after the installation of the NYCe 4000 software, the NYCeTuner starts in the "wizard" mode. When the NYCeTuner is closed, the mode of operation is remembered and is started automatically the next time the NYCeTuner is started. The "wizard" mode is described in [chapter 2.6 "Quick start" on page 43](#), and the "expert" mode is described in [chapter 4.5 "Expert mode" on page 117](#). To toggle between the "wizard" mode and the "expert" mode, click "View" in the menu bar of the NYCeTuner and select "Wizard" or "Expert", respectively.

Before you can tune an axis you must select and initialize the axis, see [chapter 4.3 "Select and initialize an axis" on page 116](#).

Saving parameters is discussed in [chapter 4.4 "Saving and shutting down" on page 116](#).

4.2 Overview

When you start the NYCeTuner, the tool starts in the "wizard" or "expert" mode. The started mode is equal to the active mode when the tool was closed the previous time it was used. Only the first time, after the installation of the NYCe 4000 software, the operating mode is the "wizard" mode.

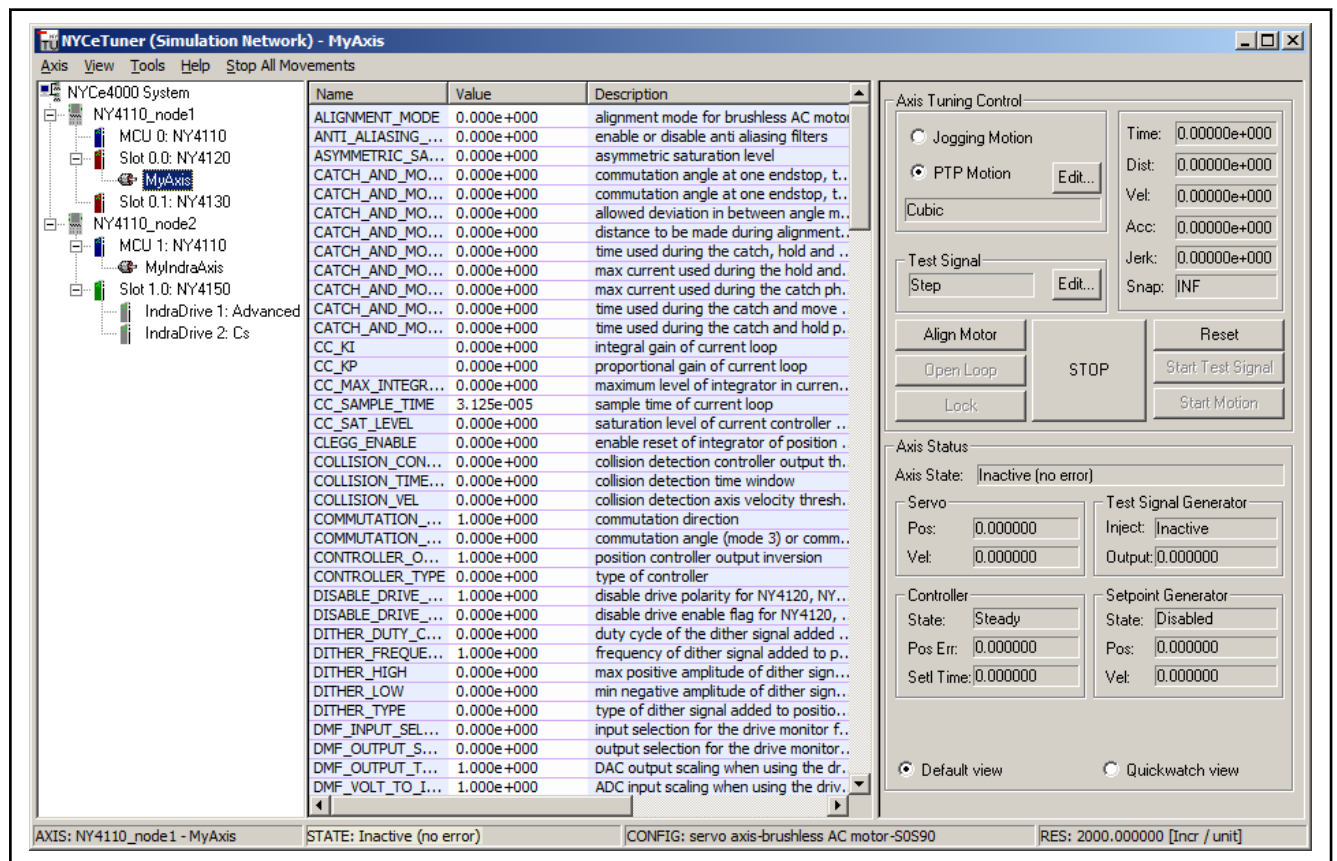


Fig. 4-1: NYCeTuner window in "expert" mode

NYCeTuner

In the pane at the left hand side a tree structure is displayed of the nodes and axes in the network. You can hide the tree structure. Click "View" in the menu bar and remove the checkmark for the "Axis tree" entry. Likewise, if the tree structure is hidden, click "View" in the menu bar and put a checkmark for the "Axis tree" entry.

When the NYCeTuner operates in the "wizard" mode, the window consists of 2 panes, and when the NYCeTuner operates in the "expert" mode, the window consists of 3 panes. [fig. 2-29 "NYCeTuner "Welcome" tab for a BLAC motor" on page 45](#) shows the NYCeTuner window in the "wizard" mode, and [fig. 4-1 "NYCeTuner window in "expert" mode" on page 115](#) shows the NYCeTuner window in the "expert" mode.

4.3 Select and initialize an axis

The NYCeTuner displays in the pane at the left side a tree with the nodes and the axes in the network. Click on an axis in the pane at the left side to set up and tune the configuration of that axis. You can also use the menu bar, click "Axis → Select".

If needed, you can load the parameters from an axis parameter file on the host, click "Axis → Initialize" in the menu bar. You have 2 possible locations from where you can load an axis parameter file.

1. "initialize from file"
load parameters from an axis parameter file on the host.
2. "initialize from flash"
load parameters from the flash memory.

4.4 Saving and shutting down

Two methods are available to save the changed parameters after you have completed your tests.

1. Click "Axis" in the menu bar of the NYCeTuner and select "Save to File..." or "Save to Flash", depending on where you want to store the parameters.
 - "Save to File..."
save parameters to an axis parameter file on the host.
 - "Save to Flash"
save parameters to the flash memory.
2. Click the "Save / Download" tab if the tool operates in the "wizard" mode.

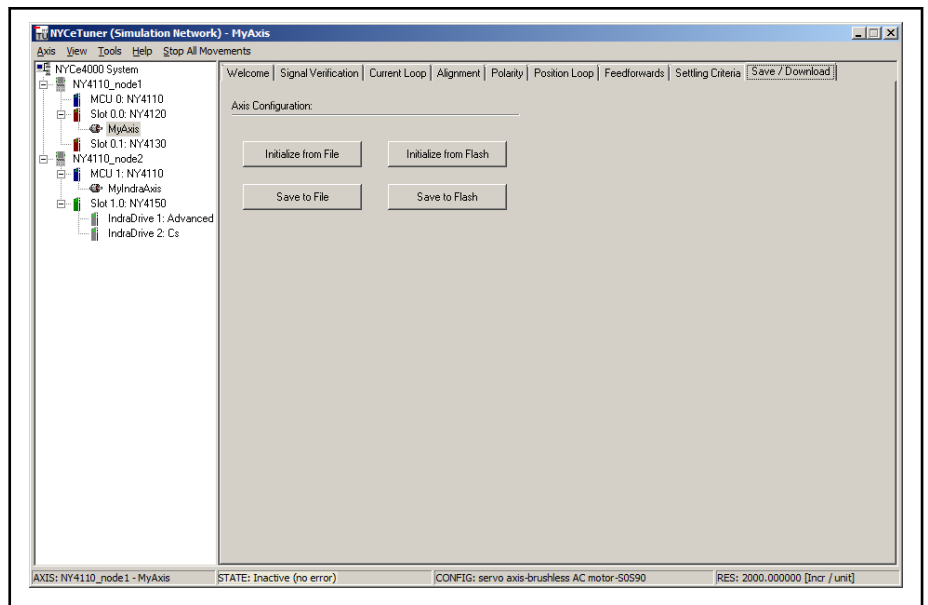


Fig. 4-2: NYCeTuner "wizard" mode, "Save / Download" tab

You can shutdown the axis from the menu bar. Click "Axis → Shutdown".

4.5 Expert mode

You can set all parameters of the axis individually in the "expert" mode.

To enter the "expert" mode, choose "View → Expert" from the menu bar of the NYCeTuner window.

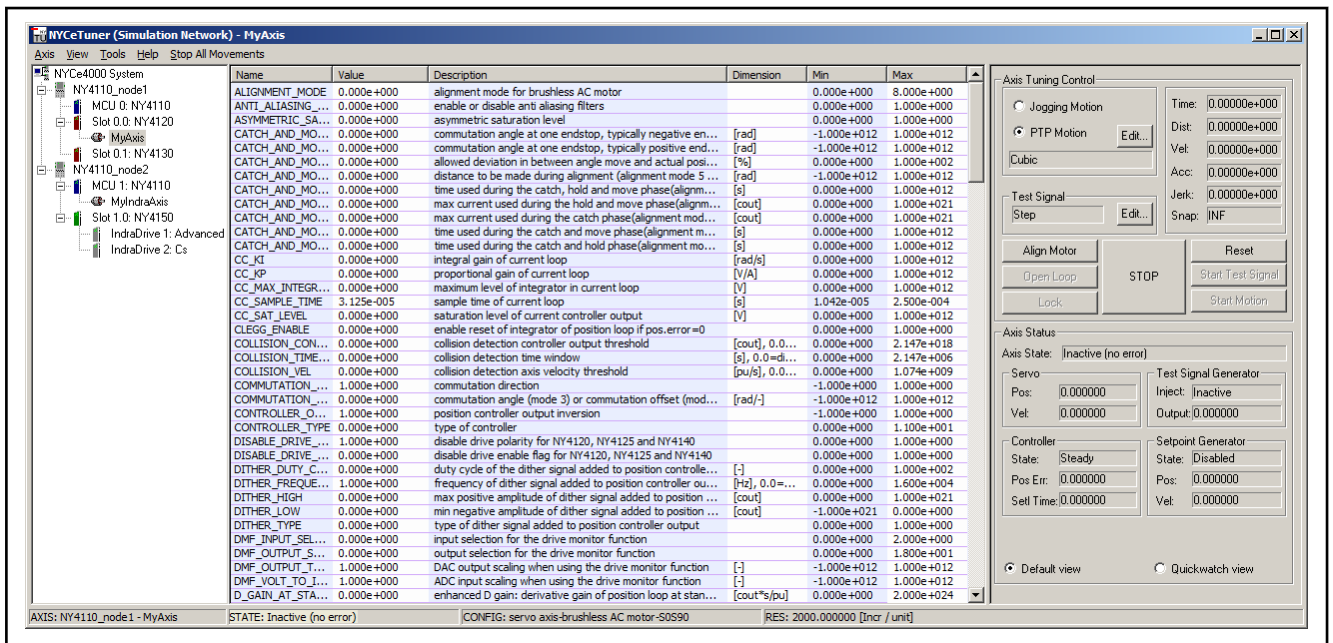


Fig. 4-3: The NYCeTuner window ("expert" mode)

Tree

NYCeTuner

The pane at the left hand side of the NYCeTuner window displays the nodes and axes connected to the system. You can hide the tree view and display it again by selecting and deselecting "View → Axis tree" in the menu bar.

Parameters

The pane in the middle displays the NYCe 4000 parameters that you can change to setup or (fine) tune the axes. You can enlarge the horizontal and vertical size of this pane. Position the mouse pointer over the edge of the NYCeTuner window and drag the size of the window to the desired size. The width of the columns in the pane in the middle can be set by changing the header field at the top, see [fig. 4-3 "The NYCeTuner window \("expert" mode\)" on page 117](#) for an example of a resized NYCeTuner window in Expert mode.

To change the value of a parameter, click the mouse pointer in the "Value" edit field or press F2 when the parameter line is selected. The latter method is practical if you must change several parameters, because you can use the up and down cursor keys to navigate to the parameters that you want to change.

By default, all the parameters are displayed in the pane in the middle, and the parameters are listed in alphabetical order. You can display the individual groups (except the miscellaneous group) by selecting the group in the "View" menu. The "Control Filter Parameters" is selected from the "View" menu in [fig. 4-4 "Select a specific group of parameters" on page 118](#).

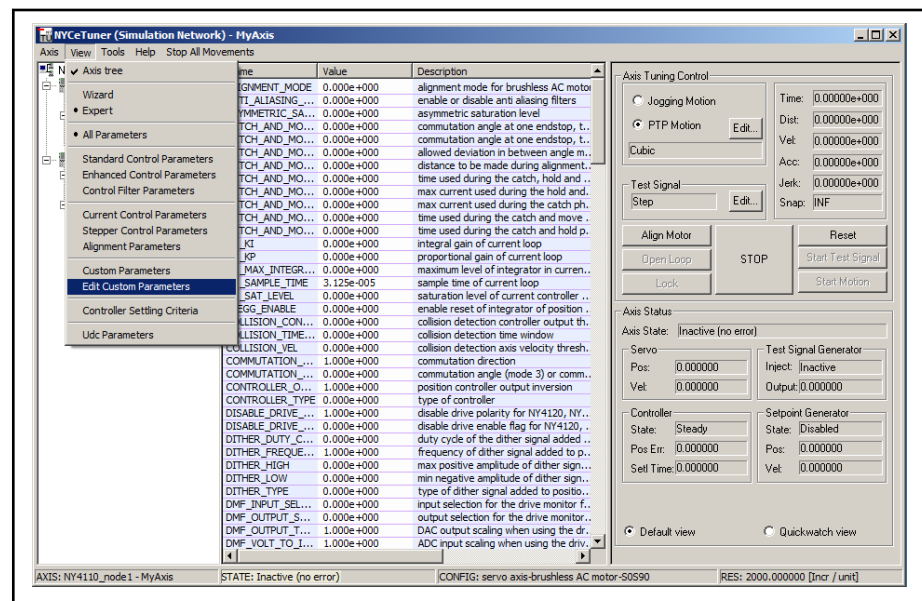


Fig. 4-4: Select a specific group of parameters

The parameters are divided into the following groups.

- The Standard control parameters, to setup and tune the PID controller together with the feedforward gains, see [chapter 4.6 "Standard controller parameters" on page 121](#).
- The Enhanced control parameters, for custom PID controllers (for example speed dependent proportional gain), see [chapter 4.7 "Enhanced controller parameters" on page 126](#).

- The Control filter parameters, to further improve axis behavior together with the PID controller, see [chapter 4.8 "Control filter parameters" on page 133](#).
- The Current control parameters, to setup the current loop of an axis, see [chapter 4.9 "Current control parameters" on page 138](#).
- The Stepper control parameters, to setup stepper motor axes, see [chapter 4.10 "Stepper control parameters" on page 139](#).
- The Alignment parameters, to get BLAC motors up and running, see [chapter 4.11 "Alignment parameters \(BLAC and IndraBLAC only\)" on page 141](#).
- The Controller settling criteria, to define the criteria for an axis to get in steady state, see [chapter 4.13 "Controller settling criteria" on page 155](#).

You can revert to the full parameter list by choosing "View → All parameters". In the full parameter list are also a series of miscellaneous parameters, see [chapter 4.12 "Miscellaneous NYCeTuner parameters" on page 153](#).

Custom parameters

You can create a custom defined list of parameters displayed in the pane in the middle. Do the following steps to create a custom defined list of parameters.

1. Choose "View → Edit custom parameters".

The "Selection Dialog" window appears, see [fig. 4-5 "Selecting custom parameters" on page 119](#).

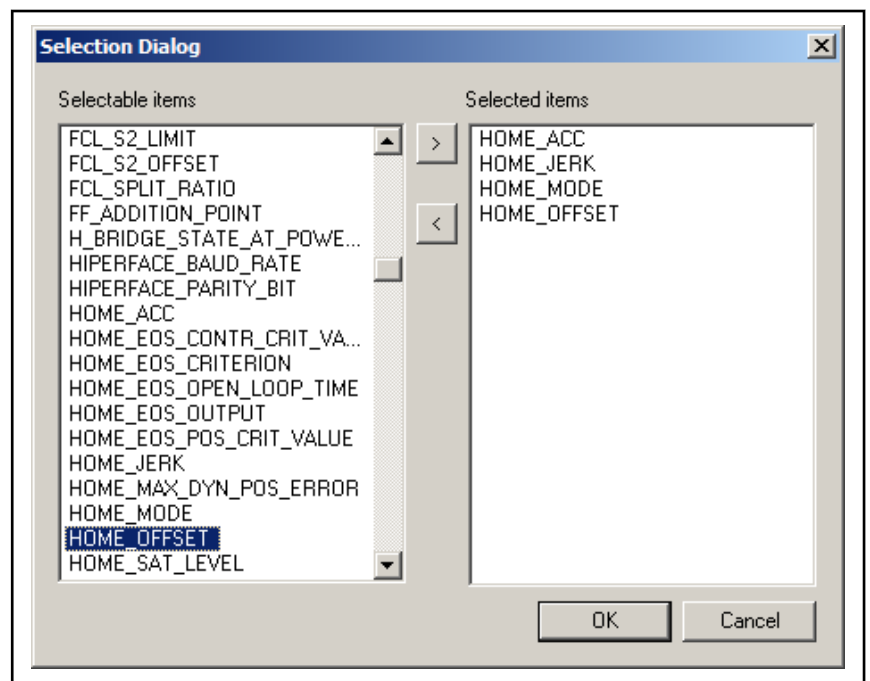


Fig. 4-5: Selecting custom parameters

2. Click on a parameter in the "Selectable items" pane of the "Selection Dialog" window and click the ">" button to move the parameter to the "Selected items" pane. Likewise, click on a selected parameter in the "Selected items" pane and click the "<" button to move a selected parameter to the "Selectable items" pane. You can also double-click on

NYCeTuner

a parameter in the "Selectable items" pane to move the parameter to the "Selected items" pane, or double-click on a selected parameter in the "Selected items" pane to move the selected parameter to the "Selectable items" pane.

Click the "OK" button when you have completed your custom defined list. The pane in the middle now only shows the selected parameters.

- To see all parameters, click "View → All Parameters". The custom defined parameters can be displayed again by clicking "View → Custom parameters".

Parameters for IndraDrive

Most SAC parameters are also applicable for an axis defined on an IndraDrive. However, the following SAC parameters are not applicable if the axis is defined on an IndraDrive.

- Some current control parameters, see [chapter 4.9 "Current control parameters" on page 138](#).
- All stepper control parameters, see [chapter 4.10 "Stepper control parameters" on page 139](#).
- All BLAC alignment parameters, see [chapter 4.11 "Alignment parameters \(BLAC and IndraBLAC only\)" on page 141](#).
- Parameters listed in [tab. 4-1 "SAC parameters not applicable for IndraDrive axes" on page 120](#).

SAC parameters not applicable for IndraDrive axes		
BLAC_NR_OF_POLE_PAIRS	BLAC_PU_PER_NR_OF_POLE_PAIRS	BLDC_HALL_SENSOR_SET
BLDC_SIX_STEP_COMM_x (x=0...5)	BRAKE_DIG_IO_NR	BRAKE_SLOT_ID
ENDAT_CLOCK_FREQUENCY	EXT_DRIVE_EN_DIG_IO_NR	EXT_DRIVE_EN_SLOT_ID
FF_ADDITION_POINT	HIPERFACE_BAUD_RATE	HIPERFACE_PARITY_BIT
S0S90_SINGLE_ENDED	SECONDARY_COUNTER_DIRECTION	SINCOS_INDEX_LEVEL_THRESHOLD
SINCOS_INTERPOLATION_FACTOR	SINCOS_MAX_ERROR_LEVEL	SINCOS_MIN_ERROR_LEVEL
STEP_NR_OF_MICRO_STEPS	STOP_ALARM_ACTIVE_LEVEL	STOP_ALARM_ENABLE

Tab. 4-1: SAC parameters not applicable for IndraDrive axes

AutoTweak tool

You can use the AutoTweak tool to automatically measure the position dependent feedforward, see [chapter 4.14 "AutoTweak tool" on page 158](#).

Motion stimuli

The NYCeTuner also has a series of motion stimuli to enable the user to perform and monitor the motion related actions during the tuning procedures. These motion options are explained in [chapter 4.15 "NYCeTuner motion options" on page 160](#).

Editors

The NYCeTuner contains 3 editors, a point-to-point motion editor, a jogging motion editor, and the test signal editor.

Axis Status

Default, the bottom section of the pane at the right hand side displays the axis status information.

The radio button "Quickwatch view" changes this section to a window in which you can select parameters that subsequently are displayed in this section. When you right-click on an entry in the "Name" column a drop-down list appears from which you can select a parameter you want to see in the quickwatch view. When you left-click on an entry in the "Name" or "Value" column a pop-up appears with two selections, "Edit List" and "Reset List". If you select "Edit List", the "Selection Dialog" window appears. See "Custom parameters" for a description. If you select "Reset List", all selected parameters are removed from the list.

The button "Default view" reverts the view to the "Axis Status" view.

4.6 Standard controller parameters

NYCeTuner can selectively display the standard controller parameters via the "View" menu, see [tab. 4-2 "Standard controller parameter list" on page 121](#). The lay-out of the standard NYCe 4000 position controller is shown in [fig. 4-6 "The standard controller lay-out" on page 123](#).

Name	Description (displayed by NYCeTuner)	Unit
ANTI_WINDUP_GAIN	Gain for integrator windup	-
ASYMMETRIC_INTEGRATOR_SATURATION	Asymmetric integrator saturation	-
ASYMMETRIC_SATURATION	Asymmetric saturation level	-
CONTROLLER_OUTPUT_INVERT	Position controller output inversion	-
CONTROLLER_TYPE	Type of controller (0=PVL on DRV, 1=PVL on MCU, 2=UDC_0, 3=UDC_1, 4=UDC_2 ... 11=UDC_9)	-
FF_ADDITION_POINT	Feedforward addition point	-
INDRA_LOAD_INERTIA	IndraDrive load inertia	kg or kgm ²
INTEGRATOR_RESET_VALUE	Reset value of the integrator of the position loop	cout
INTEGRATOR_SWITCH_OFF_MODE	Integrator behavior above threshold velocity(0 = reset and disable, 1 = hold)	-
KFA	Acceleration feedforward gain of position loop	cout * s ² /pu
KFC	Coulomb friction feedforward gain of position loop	cout
KFJ	Jerk feedforward gain of position loop	cout * s ³ /pu
KFSNAP	Snap feedforward gain of position loop	cout * s ⁴ /pu
KFST	Static feedforward gain of position loop	cout
KFV	Velocity feedforward gain of position loop	cout * s/pu
KI	Integral gain of position loop	rad/s

NYCeTuner

Name	Description (displayed by NYCeTuner)	Unit
KP	Proportional gain of position loop	cout/pu
KV	Derivative gain of position loop	cout * s/pu
MAX_INTEGRATOR_LEVEL	Maximum level of the integrator of the position loop	cout
MEAS_SYST_DIRECTION	Measuring system direction(-1=neg, 1=pos)	-
MIN_INTEGRATOR_LEVEL	Minimum level of the integrator of the position loop	cout
NR_OF_SAMPLES	Number of samples used for averaging if velocity estimation method = derivative sample	-
PFC_KP	Position force proportional gain	pu/N
PVL_SAMPLE_TIME	Position velocity loop sampling time	s
SAT_LEVEL	Maximum saturation level of the position controller output	cout
SAT_MIN_LEVEL	Minimum saturation level of the position controller output, only used when asymmetric saturation level is enabled	cout
THRESHOLD_VELOCITY	Below threshold velocity, integrator of the position loop is enabled	pu/s
VELOCITY_ESTIMATION_METHOD	The velocity estimation method in the position loop (0 = Lagrange2, 1 = derivative sample)	-

Tab. 4-2: Standard controller parameter list

The position controller can run from both the motion control unit (MCU) as the drive module (DRV). The parameter `CONTROLLER_TYPE` defines this as follows:

- `CONTROLLER_TYPE = 0`, position controller runs on the drive
- `CONTROLLER_TYPE = 1`, position controller runs on the MCU

Besides the standard position controller, a user defined controller (UDC) can be applied. The defined UDCs can be selected by setting the parameter `CONTROLLER_TYPE` to any value between 2 and 11.

- `CONTROLLER_TYPE = 2 ...11`, user defined controller runs on the MCU

If the parameter `CONTROLLER_TYPE` is set to a value higher than 0 (controller runs on the MCU), the sample frequency of the position velocity loop is equal to the "Setpoint (Node) Frequency", as specified in the "Frequencies" tab in the NYCeConfigurator. If the parameter `CONTROLLER_TYPE` is set to 0, the sample frequency of the position velocity loop is equal to the "Position Velocity Loop Frequency" as defined in NYCeConfigurator.



For smallest delays, leave the "Position Velocity Loop Frequency" on 32 kHz even if the control loop is not running on the drive, but runs with the "Setpoint (Node) frequency" value on the MCU.

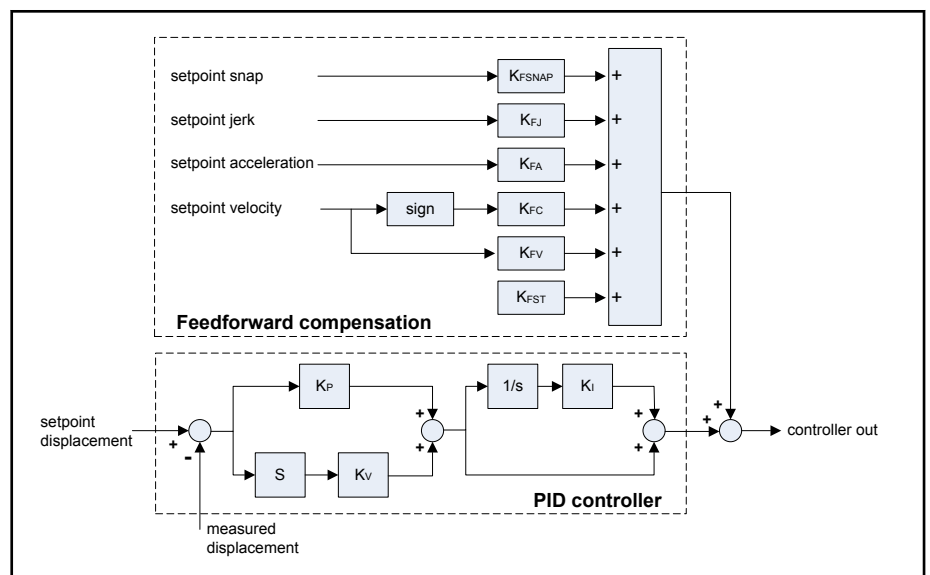


Fig. 4-6: The standard controller lay-out

The parameters K_P (proportional gain), K_V (derivative gain) and K_I (integral gain) are the PID parameters. The PID controller is limited by the saturation level of the controller out (SAT_LEVEL) and the position loop integrator level ($MAX_INTEGRATOR_LEVEL$, $MIN_INTEGRATOR_LEVEL$, and $ASYMMETRIC_INTEGRATOR_SATURATION$). The integrator is limited by the $THRESHOLD_VELOCITY$ as well. This parameter defines the velocity below which the integrator of the position loop is active. The integrator will always be active when the velocity parameter is set to -1.

The differentiator of the PID loop can differentiate with two algorithms.

- $VELOCITY_ESTIMATION_METHOD = 0$ for Lagrange differentiation.
- $VELOCITY_ESTIMATION_METHOD = 1$ for derivative sample differentiation.

This method is typically used when the sample frequency of the position velocity loop is high (so, especially when running this loop on the drive module, $CONTROLLER_TYPE = 0$). When using this method, be sure to specify the number of samples used for averaging with the $NR_OF_SAMPLES$ parameter. Typical values are 4 samples at 8 kHz, 8 samples at 16 kHz, 16 samples at 32 kHz.

The maximum number of samples used for averaging is 16.

If the selected axis is on the IndraDrive the parameter PVL_SAMPLE_TIME is fixed, and must be set to 0.5 ms.

The saturation level of the controller out can also be set asymmetrical, by enabling the parameter $ASYMMETRIC_SATURATION$. Therefore, the integrator saturation can also be set asymmetrical with the parameter $ASYMMETRIC_INTEGRATOR_SATURATION$. The parameter $MAX_INTEGRATOR_LEVEL$ sets the upper limit and $MIN_INTEGRATOR_LEVEL$ sets the lower limit. The value of $MIN_INTEGRATOR_LEVEL$ must be mathematically lower than the value of $MAX_INTEGRATOR_LEVEL$. However, the absolute value of $MIN_INTEGRATOR_LEVEL$ can be higher than the absolute value of $MAX_INTEGRATOR_LEVEL$.

When the $ASYMMETRIC_INTEGRATOR_SATURATION$ is enabled, the value 0 might not be a correct value to reset the integrator buffer. The value 0 can be

NYCeTuner

outside the `MIN_INTEGRATOR_LEVEL` to `MAX_INTEGRATOR_LEVEL` range. With the parameter `INTEGRATOR_RESET_VALUE` you can set the integrator buffer reset value.

The integrator reset value is, besides initial integrator buffer value, also used as a clamp value when the integrator anti-windup is applied. The anti-windup is expressed as

$$U_{\text{anti_windup}} = U_{\text{out}} - \text{Saturate}(U_{\text{out}}) \quad \textcircled{1}$$

$$\text{Integrator} = \text{Integrator} - (\text{Gain}_{\text{anti_windup}} * U_{\text{anti_windup}}) \quad \textcircled{2}$$

The following clamping rules apply if saturation occurs, even if the anti-windup gain is 0.

- If the controller output saturates at its maximum level: `integrator >= integrator reset value`
- If the controller output saturates at its minimum level: `integrator <= integrator reset value`

Thus, if the integrator value goes "to the wrong side" of the integrator reset value when saturation occurs, the integrator value is clamped at the integrator reset value. For example, if the controller output saturates at its maximum, the integrator decreases. However, if the controller output goes below its reset value, it is clamped at the integrator reset value. Also, if the integrator is already below its reset value, it is also forced to the reset value.

The anti-windup is even more flexible by applying an anti-windup gain with the parameter `ANTI_WINDUP_GAIN`. The default value of this parameter is 1.0. When the parameter `ANTI_WINDUP_GAIN` is set to 0.0 the anti-windup is effectively disabled. More specific, the integrator buffer is not changed, although the integrator reset value crossing check is still executed.

Besides the PID controller, the NYCe 4000 system also supports feedforward compensation. The feedforwards can compensate static forces (like gravity), coulomb frictions, viscous frictions and the acceleration forces (needed to accelerate the mass / inertia driven by the axis). These feedforward parameters must be acquired by measurements or from a model of the mechanical dynamics.

The feedforward compensation can be added to the PVL before or after the notch filters with the parameter `FF_ADDITION_POINT`. If the parameter is set to 0 (`SAC_FF_BEFORE_NOTCH`) the feedforward compensation is added before the notch filters, the value 1 (`SAC_FF_AFTER_NOTCH`) adds the feedforward compensation after the notch filters. The `FF_ADDITION_POINT` parameter is not supported for the IndraDrive.

fig. 4-7 "Direction parameters in the control loops" on page 125 shows a block diagram of the control loops and shows in which loop the parameters `MEAS_SYST_DIRECTION`, `CONTROLLER_OUTPUT_INVERT` and `COMMUTATION_DIRECTION` act. The `COMMUTATION_DIRECTION` parameter is used in alignment procedures for BLAC motors, see chapter 4.11 "Alignment parameters (BLAC and IndraBLAC only)" on page 141.

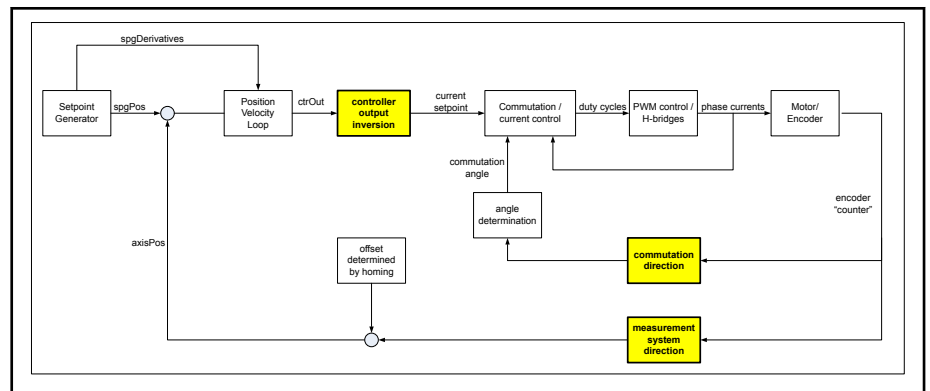


Fig. 4-7: Direction parameters in the control loops



For BLAC motors the parameter COMMUTATION_DIRECTION must be set first.

When you set up your motion control loop, 2 conditions are possible. Either the motion system is known or it is not known.

- Motion system is known
 - If the motor type is a BLAC, set the parameter COMMUTATION_DIRECTION to +1
 If the axis angle of the BLAC motor equals 0° when $U = 0^\circ$, $V = 120^\circ$, and $W = 240^\circ$ else set the parameter COMMUTATION_DIRECTION to -1.
 - If the motor type is a DC motor, the parameter COMMUTATION_DIRECTION is not relevant.
 - Set the parameter CONTROLLER_OUTPUT_INVERT to +1 if the axis of the motor rotates clockwise when the motor current is positive else set the parameter CONTROLLER_OUTPUT_INVERT to -1.
 - Set the parameter MEAS_SYST_DIRECTION to +1 if the counter value of the encoder increments when the encoder axis rotates clockwise else set the parameter MEAS_SYST_DIRECTION to -1.
- Motion system is unknown
 - If the motor type is BLAC, set the parameters MEAS_SYST_DIRECTION and CONTROLLER_OUTPUT_INVERT to +1, and execute a SacMoveCurrentAngle function with an angle displacement of 2π . If the axis position changes in positive direction, set the parameter COMMUTATION_DIRECTION to +1, else set the parameter COMMUTATION_DIRECTION to -1.
 - Set the parameters MEAS_SYST_DIRECTION and CONTROLLER_OUTPUT_INVERT to +1. Try if the position control loop is stable. This is always the case for a BLAC motor after the alignment procedure is executed. For all other motor types you must check whether the control loop is stable. If the control loop is not stable you must invert the parameter CONTROLLER_OUTPUT_INVERT.

If the direction, in which the axis position changes, matches with the definition of the machine coordinate frame in your application, the measurement

NYCeTuner

system direction is correct, else the parameter `MEAS_SYST_DIRECTION` must be inverted.

If the motor is a DC motor and you inverted the parameter `MEAS_SYST_DIRECTION` you must also invert the parameter `CONTROLLER_OUTPUT_INVERT`.

If the motor is a BLAC and you inverted the parameter `MEAS_SYST_DIRECTION` you must execute the alignment procedure again. If only `MEAS_SYST_DIRECTION` or `CONTROLLER_OUTPUT_INVERT` is inverted and no re-alignment is performed, the position velocity loop becomes unstable. This issue is corrected when the alignment procedure is executed again, or both parameters are inverted.



- If the commutation offset is determined and only `MEAS_SYST_DIRECTION` or `CONTROLLER_OUTPUT_INVERT` is inverted, alignment results is a commutation offset which theoretically differs π radians (180°) from the original commutation offset.
- If only `MEAS_SYST_DIRECTION` or `CONTROLLER_OUTPUT_INVERT` is inverted, alignment with the Catch & Move procedure requires that the parameters `CATCH_AND_MOVE_ANGLE_ALPHA` and `CATCH_AND_MOVE_ANGLE_BETA` must be changed by π radians.
- If the axis is on an IndraDrive, the controller output inverted parameter and the measurement direction parameter must always have the same direction.

4.7 Enhanced controller parameters

NYCeTuner can selectively display the enhanced controller parameters via the "View" menu, see [tab. 4-3 "Enhanced controller parameter list" on page 126](#). The complete control lay-out, including the enhanced controller options is shown in [fig. 4-8 "The control layout with the enhanced controller blocks" on page 128](#).



If the PVL is on the IndraDrive, all parameters listed in [tab. 4-3 "Enhanced controller parameter list" on page 126](#) have a fixed value (which cannot be changed). See the NYCe 4000 Software User Manual, chapter SAC Parameters, for the fixed assigned values.

Name	Description (displayed by NYCeTuner)	Unit
<code>CLEGG_ENABLE</code>	Enable reset of integrator of position loop if <code>pos.error=0</code> (0=disable, 1=enable)	-
<code>DITHER_DUTY_CYCLE</code>	Duty cycle of the dither signal added to the position controller output	-
<code>DITHER_FREQUENCY</code>	Frequency of the dither signal added to the position controller output (0=disabled)	Hz
<code>DITHER_HIGH</code>	Max positive amplitude of dither signal added to the position controller output	cout
<code>DITHER_LOW</code>	Min negative amplitude of dither signal added to the position controller output	cout

Name	Description (displayed by NYCeTuner)	Unit
DITHER_TYPE	Type of dither signal added to position controller output (0=dither block, 1=dither pulse height modulation)	-
D_GAIN_AT_STAND_STILL	Enhanced D gain: derivative gain of position loop at standstill	cout s/pu
D_HALF_WAY_VEL	Enhanced D gain: velocity where der. gain is average of Kv and gain at standstill	pu/s
IFC_DEAD_ZONE_CONTROL_OUT_NEG	Intrinsic friction compensation: contr. output at negative bound of dead zone	cout
IFC_DEAD_ZONE_CONTROL_OUT_POS	Intrinsic friction compensation: contr. output at positive bound of dead zone	cout
IFC_DEAD_ZONE_GAIN	Intrinsic friction compensation: gain in dead zone	-
I_GAIN_AT_STAND_STILL	Enhanced I gain: integral gain of position loop at standstill	rad/s
I_HALF_WAY_VEL	Enhanced I gain: velocity where int. gain is the average of Ki and gain at standstill	pu/s
MAX_INTEGRATOR_LEVEL_AT_STAND_STILL	Enhanced max integrator level: max integrator level of position loop at standstill	cout
MAX_INTEGRATOR_LEVEL_HALF_WAY_VEL	Enhanced max integrator level: velocity where max integrator level is avg of max integrator level and max integrator level at standstill	pu/s
P_GAIN_AT_STAND_STILL	Enhanced P gain: proportional gain of position loop at standstill	cout/pu
P_HALF_WAY_VEL	Enhanced P gain: velocity where prop. gain is the avg of Kp and gain at standstill	pu/s
SCM_ENABLE	Switched control mechanism: enable (0=disable, 1=enable)	-
SCM_RESET_NEG	Switched control mechanism: negative range of reset window	pu
SCM_RESET_POS	Switched control mechanism: positive range of reset window	pu
SCM_RESTART_INT_NEG	Switched control mechanism: negative restart value of the integrators	cout
SCM_RESTART_INT_POS	Switched control mechanism: positive restart value of the integrators	cout

Tab. 4-3: Enhanced controller parameter list

NYCeTuner

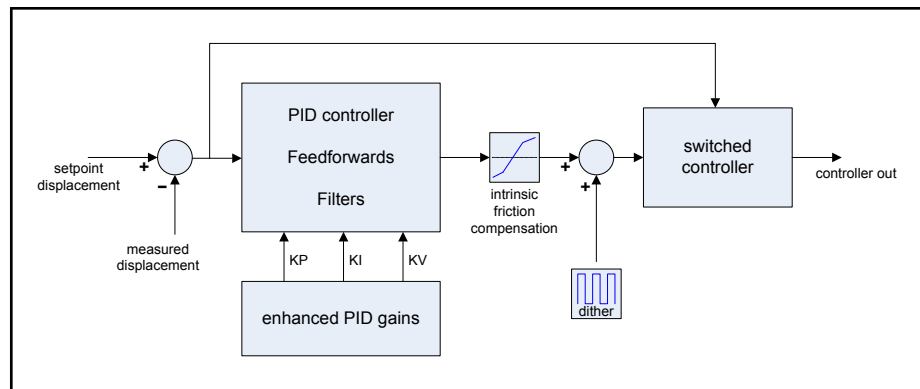


Fig. 4-8: The control layout with the enhanced controller blocks

The different functions are explained in the following paragraphs.

Enhanced PID gains

The enhanced gains are an option to make the gains of the PID controller speed dependent. The choice which gains are made speed dependent is completely free, but the speed dependency of each gain has the same profile. [fig. 4-9 "Gain – velocity" diagram of the enhanced controller parameters](#) on page 129 shows the "gain – velocity" profile used for each enhanced gain parameter.

For each enhanced gain two parameters must be defined.

- The gain at stand-still
 - P_GAIN_AT_STAND_STILL
 - I_GAIN_AT_STAND_STILL
 - D_GAIN_AT_STAND_STILL
- The velocity at which the enhanced gain is the average of the stand-still gain and the gain supplied at the standard control parameters
 - P_HALF_WAY_VEL
 - I_HALF_WAY_VEL
 - D_HALF_WAY_VEL

The calculations of the gain versus velocity enhanced controller parameters are as follows.

Program:

```

kp = parKp + ( ( pGainAtStandStill - parKp ) /
              ( 1 + ( 1 / pHalfWayVel ) * fabsf( spgVelocity ) ) )
kv = parKv + ( ( dGainAtStandStill - parKv ) /
              ( 1 + ( 1 / dHalfWayVel ) * fabsf( spgVelocity ) ) )
ki = parKi + ( ( iGainAtStandStill - parKi ) /
              ( 1 + ( 1 / iHalfWayVel ) * fabsf( spgVelocity ) ) )

```

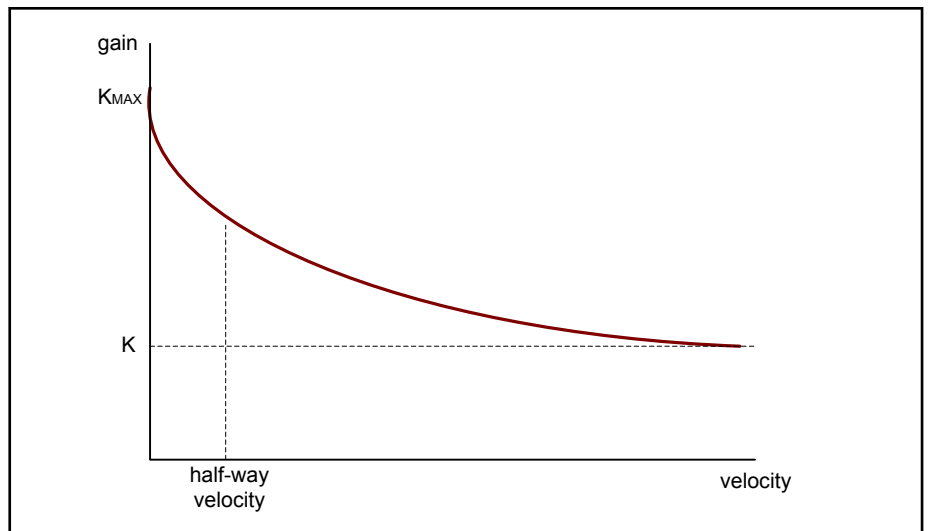


Fig. 4-9: "Gain – velocity" diagram of the enhanced controller parameters



The gain at stand-still may also be chosen smaller than the K parameter from [fig. 4-9 "Gain – velocity" diagram of the enhanced controller parameters](#) on page 129.

Intrinsic friction compensation

With the intrinsic friction compensation the loop gain of the controller can be enhanced or reduced for controller output near zero (the dead-zone). [fig. 4-10 "Intrinsic friction compensation diagram"](#) on page 130 shows a diagram of the intrinsic friction compensation function. The gain of the dead-zone can be defined by the `IFC_DEAD_ZONE_GAIN` parameter. The bounds of the dead-zone are defined by the parameters `IFC_DEAD_ZONE_CONTROL_OUT_NEG` and `IFC_DEAD_ZONE_CONTROL_OUT_POS`.

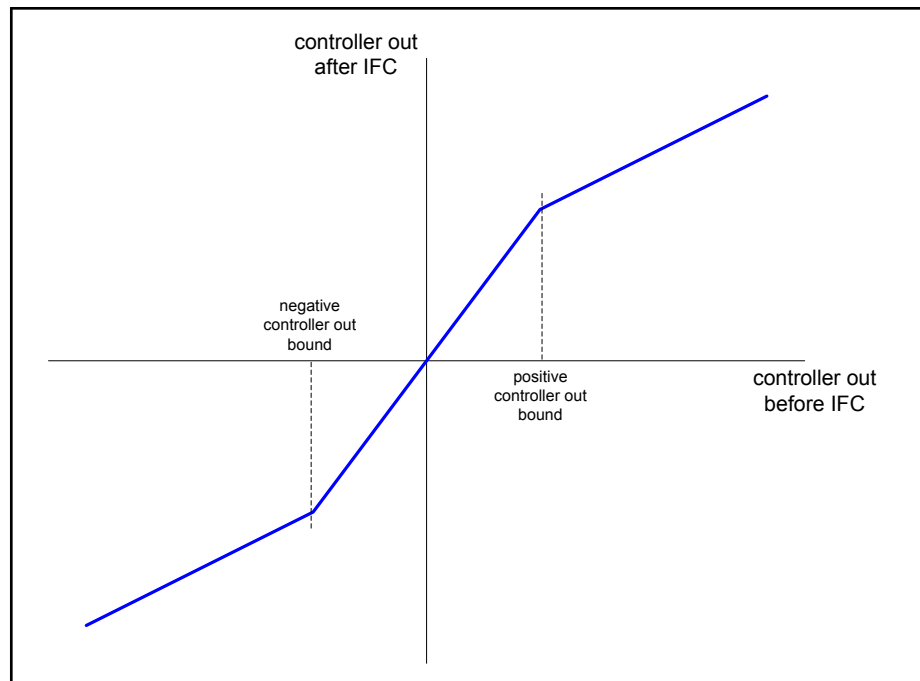


Fig. 4-10: Intrinsic friction compensation diagram

Dither signal

If an axis suffers from static friction problems, or the axis has to move at ultra low speed it may benefit from the addition of a (high frequency, up to 16 kHz) dither signal to the controller output. The dither signal is available in two different modes.

- `DITHER_TYPE = 0`: BLOCK
- `DITHER_TYPE = 1`: PHM (Pulse Height Modulation)

BLOCK consists of a block wave signal with adjustable amplitude, duty cycle and frequency. The amplitude is specified by the `DITHER_HIGH` and `DITHER_LOW` parameters. The duty cycle sets the percentage within one period that the BLOCK signal is equal to `DITHER_HIGH`. The BLOCK signal is added to the controller output. See [fig. 4-11 "Controller output with dither BLOCK" on page 131](#) for an example. In this example the duty cycle is set to 33%.

This mode is typically used for static friction problems.

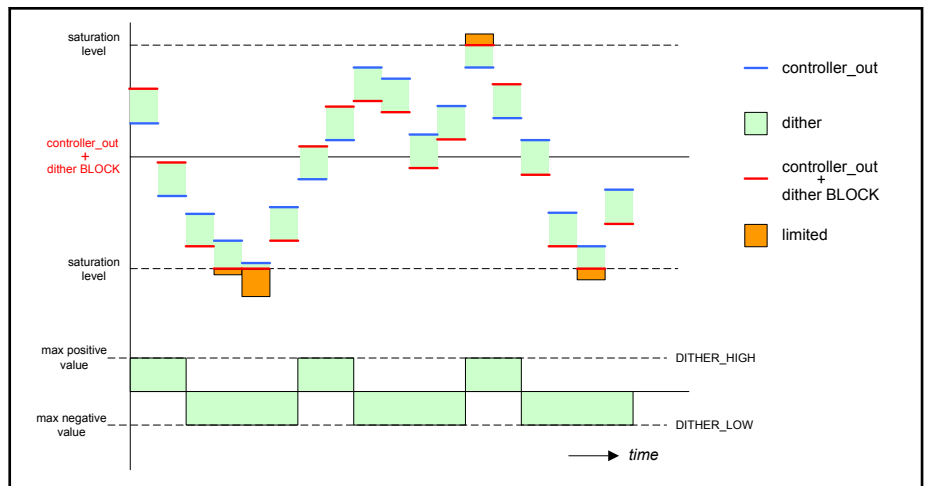


Fig. 4-11: Controller output with dither BLOCK

PHM (Pulse Height Modulation) also consists of a block signal. The output of PHM is related to the controller output. The duty cycle gives the "on time" within one period of the PHM signal. During this "on time" the output of the PHM is equal to the controller output (so the resulting output is equal to two times the controller output).

However, if the controller output is positive and larger than DITHER_HIGH, the PHM output is restricted to DITHER_HIGH. If the controller output is negative and (absolute value) larger than DITHER_LOW, the PHM output is restricted to DITHER_LOW. See fig. 4-12 "Controller output with dither PULSE" on page 132 for an example. In this example the duty cycle is 33%.

This mode is used to move the axis at ultra low speed.

As explained, the position velocity loop (PVL) can run on the MCU as well as on the drive module. However, the dither signal is always generated by the drive. The highest possible frequency of the dither signal is equal to half the PVL frequency on the drive. So even if the PVL runs on the MCU, the PVL frequency on the drive module has to be run at the highest possible frequency (32 kHz) to achieving the highest dither frequency (16 kHz).

Note that the controller output variable (to be read or traced) does not contain this dither signal.

NYCeTuner

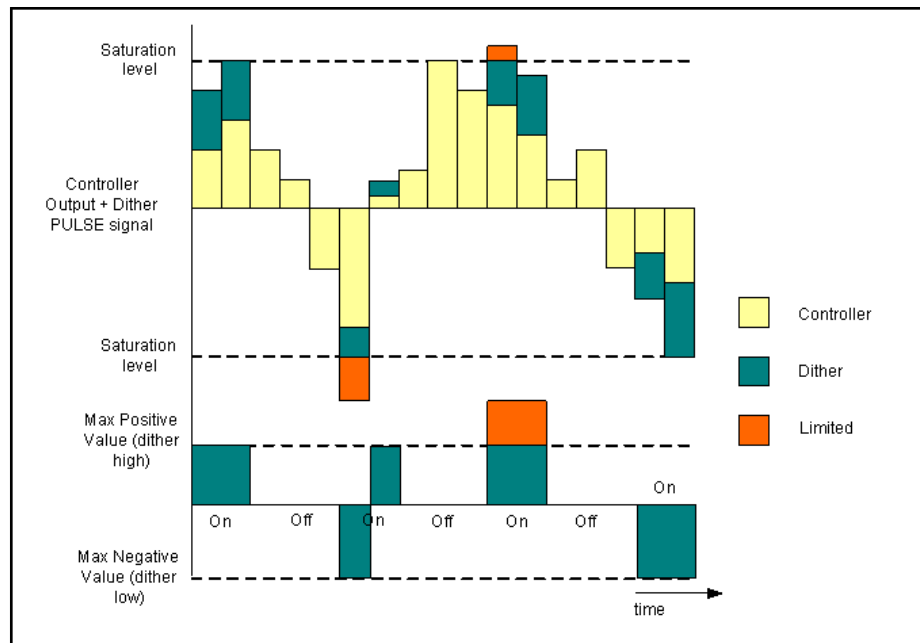


Fig. 4-12: Controller output with dither PULSE

Integrator level

With the parameters `MAX_INTEGRATOR_LEVEL_AT_STAND_STILL` and `MAX_INTEGRATOR_LEVEL_HALF_WAY_VEL` the integrator level can be controlled. The maximum integrator level is scaled with these maximum integrator level parameters. The actual maximum integrator level depends on the setpoint velocity and is calculated every sample. Note that the maximum integrator level scaling is only executed if the PVL runs on the MCU. You can read the actual maximum integrator level with the SAC variable `SAC_VAR_ACTUAL_MAX_INTEGRATOR_LEVEL`.

<code>parMaxIntLevel =</code>	=	<code>MAX_INTEGRATOR_LEVEL</code>
<code>intLevelAtStandStill =</code>	=	<code>MAX_INTEGRATOR_LEVEL_AT_STAND_STILL</code>
<code>intLevelHalfWayVel =</code>	=	<code>MAX_INTEGRATOR_LEVEL_HALF_WAY_VEL</code>
<code>spgVelocity =</code>	=	setpoint velocity actual
<code>MaxIntLevel =</code>	=	<code>parMaxIntLevel +</code> $\left(\frac{\text{intLevelAtStandStill} - \text{parMaxIntLevel}}{1 + \left(\frac{1}{\text{intLevelHalfWayVel}} \right) * \text{fabsf}(\text{spgVelocity})} \right)$

Tab. 4-4: Actual maximum integrator level

Switched Control Mechanism

The switched controller mechanism is an option to switch off the controller when the axis position is in between user defined limits, see [fig. 4-13 "The switched controller principle" on page 133](#).

- The controller output is set to 0 as soon as the target position is nearly reached (`SCM_RESET_POS` or `SCM_RESET_NEG`).
- The internal friction then serves as a brake and stops the axis.
- The controller out is switched on as soon as the position drifts away (out of position criterion window, see controller settling criteria).

- If controller is switched on again, parameter `SCM_RESTART_INT_NEG` or `SCM_RESTART_INT_POS` gives the start value of the integrator. The parameters `SCM_RESTART_INT_NEG` and `SCM_RESTART_INT_POS` are specified relative to `INTEGRATOR_RESET_VALUE`.

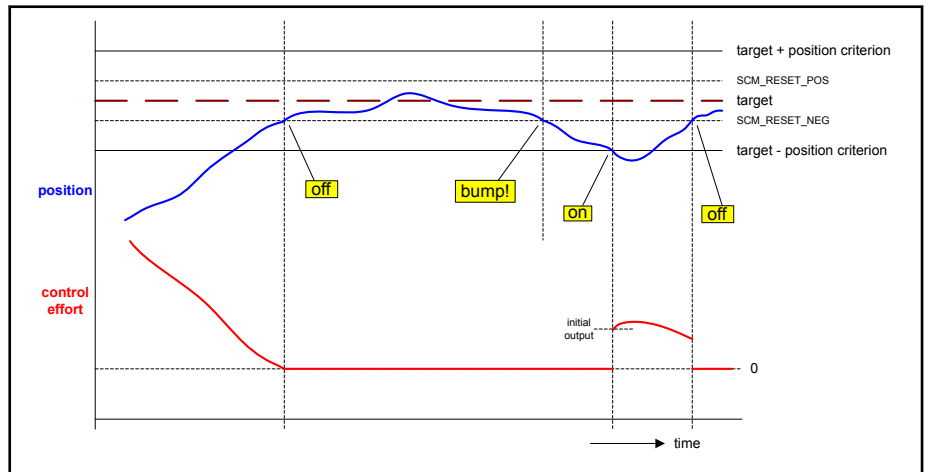


Fig. 4-13: The switched controller principle

Clegg Integrator

If the parameter `CLEGG_ENABLED` is set to 1, the integrator of the position velocity loop will reset every time the position error crosses the zero.

4.8 Control filter parameters

The position of the PID controller, the feedforward compensation and the available control filters is shown in fig. 4-14 "The available control filters" on page 133. Three types of control filters can be chosen.

- Low pass filter (LPF)
- Lead/lag filter (LLF)
- Notch filters (NF)

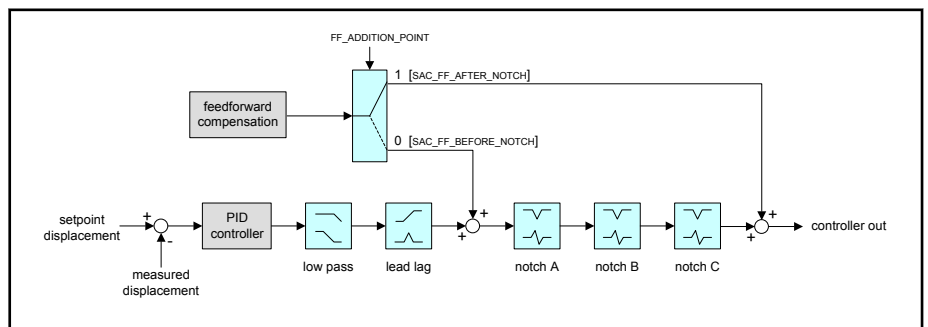


Fig. 4-14: The available control filters

tab. 4-5 "Control filter parameter list" on page 134 shows the control filter parameters as displayed by the NYCeTuner when selecting the parameters via the "View" menu.

The filters on the drive modules (NY4120, NY4125, NY4130, and NY4140) are explained in the following paragraphs, and the transfer functions are given in the Laplace s-domain.

NYCeTuner



The behavior of the low-pass filters implemented on the IndraDrive is not identical to the behavior of the low-pass filters on the drive modules (NY4120, NY4125, NY4130, and NY4140). The low-pass filter on the IndraDrive is a first order Butterworth filter with a relative damping ratio of (fixed) $\frac{1}{2}\sqrt{2}$.

Name	Description (displayed by NYCeTuner)	Unit
LLF_LAG_FREQ	Lag frequency of the lead-lag filter of the position loop	Hz
LLF_LEAD_FREQ	Lead frequency of the lead-lag filter of the position loop	Hz
LLF_SWITCH	Enable lead-lag filter of the position loop (0=disable, 1=enable)	-
LPF_DAMPING	Damping ratio of low pass filter of position loop	-
LPF_FREQ	Break frequency of low pass filter of the position loop	Hz
LPF_SWITCH	Low pass filter of the position loop: 0 = disable, 1 = enable	-
NFA_DEN_DAMP	Denominator relative damping of first notch filter of position loop	-
NFA_DEN_FREQ	Denominator frequency of first notch filter of position loop	Hz
NFA_NUM_DAMP	Numerator relative damping of first notch filter of position loop	-
NFA_NUM_FREQ	Numerator frequency of first notch filter of position loop	Hz
NFA_SWITCH	Enable first notch filter of position loop (0=disable, 1=enable)	-
NFB_DEN_DAMP	Denominator relative damping of second notch filter of position loop	-
NFB_DEN_FREQ	Denominator frequency of second notch filter of position loop	Hz
NFB_NUM_DAMP	Numerator relative damping of second notch filter of position loop	-
NFB_NUM_FREQ	Numerator frequency of second notch filter of position loop	Hz
NFB_SWITCH	Enable second notch filter of position loop (0=disable, 1=enable)	-
NFC_DEN_DAMP	Denominator relative damping of third notch filter of position loop	-
NFC_DEN_FREQ	Denominator frequency of third notch filter of position loop	Hz
NFC_NUM_DAMP	Numerator relative damping of third notch filter of position loop	-

Name	Description (displayed by NYCeTuner)	Unit
NFC_NUM_FREQ	Numerator frequency of third notch filter of position loop	Hz
NFC_SWITCH	Enable third notch filter of position loop (0=disable, 1=enable)	-

Tab. 4-5: Control filter parameter list

Low pass filter

The low pass filter is a second order filter. [fig. 4-16 "Frequency response of the low pass filter" on page 135](#) shows the frequency response of a low pass filter. Two parameters can be supplied.

- LPF_FREQUENCY, the break away frequency of the low pass filter.
- LPF_DAMPING, the damping of the low pass filter.

The output of the LPF is given by

$$U_{LPF}(s) = \frac{\omega_{LP}^2}{s^2 + 2\beta_{LP}\omega_{LP}s + \omega_{LP}^2} U_{PID}(s),$$

Fig. 4-15: Transfer function of the low pass filter

where $\omega_{LP} = 2\pi f_{LP}$. The LPF may be applied to reduce high frequency disturbances typically above the break frequency f_{LP} .

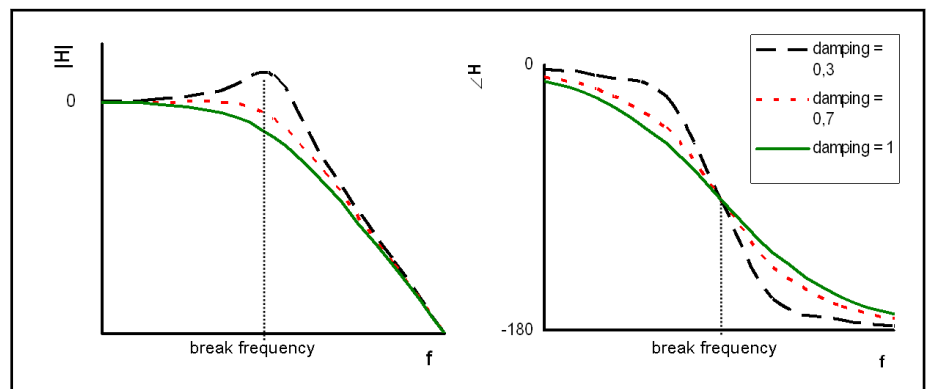


Fig. 4-16: Frequency response of the low pass filter

The low pass filter can be switched on and off with its switch parameter (LPF_SWITCH).

Lead lag filter

The lead lag filter can be used as an alternative for the differentiator of the standard PID controller. [fig. 4-17 "Differentiator response versus lead lag frequency response" on page 136](#) shows the frequency response of a differentiator versus the lead lag response. For low frequencies the lead lag response is identical to the differentiator. At frequencies higher than the lag frequency, the lead lag filter has a lower frequency response than the

NYCeTuner

differentiator, making a controller with a lead lag filter less sensitive to high frequency noise as a similar controller with a differentiator.

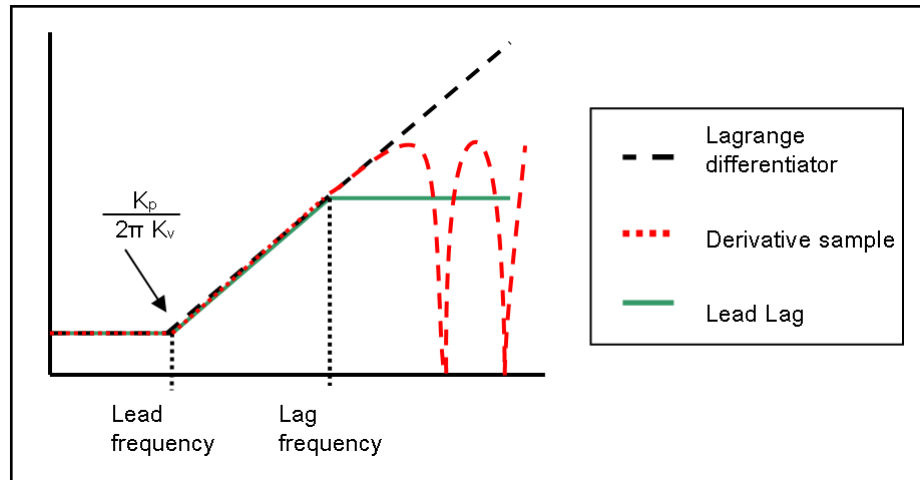


Fig. 4-17: Differentiator response versus lead lag frequency response

The lead-lag filter is characterized by its break frequencies (f_{LEAD} and f_{LAG} in Hz). It has a DC-gain of 1 and it is switched on or off by the parameter `LLF_SWITCH`. The lead-lag filter output $U_{LL}(s)$, is given by

$$U_{LL}(s) = \frac{1 + (s / \omega_{LEAD})}{1 + (s / \omega_{LAG})} U_{LPF}(s)$$

Fig. 4-18: Transfer function of the lead lag filter

where $\omega_{LEAD} = 2\pi f_{LEAD}$ and $\omega_{LAG} = 2\pi f_{LAG}$.

The lead lag filter can be switched on and off with its switch parameter (`LLF_SWITCH`).



NYCe 4000 has separate parameters for the lead lag filter and the K_v . That means that you can set the lead lag filter parameters and the K_v independently if the PVL is on the NY4120, NY4125, NY4130, or NY4140 drive module.

However, if the PVL is on the IndraDrive, it is not possible to use the lead lag filter and set a K_v at the same time, because the lead lag filter parameters and the K_v are both converted to the same parameters mapped on the IndraDrive. The lead lag filter takes precedence if enabled, and K_v is ignored. If the lead lag filter is disabled, the previously set K_v is converted to the same parameters mapped on the IndraDrive.

Notch filter

The notch filter is a band stop filter. [fig. 4-20 "Notch filter frequency response \(\$f_{numerator} = f_{denominator}\$ \)"](#) on page 137 and [fig. 4-21 "Notch filter frequency](#)

response ($f_{\text{numerator}} \neq f_{\text{denominator}}$)" on page 137 show a series of typical frequency responses of a notch filter. You can set the numerator and denominator frequencies with the parameters NFA_NUM_FREQ and NFA_DEN_FREQ when using notch filter A, NFB_NUM_FREQ and NFB_DEN_FREQ parameters when using filter B, and NFC_NUM_FREQ and NFC_DEN_FREQ when using notch filter C. The output of a notch filter is given by

$$U_{NF}(s) = \frac{\omega_{ND}^2}{\omega_{NN}^2} \cdot \frac{s^2 + 2\beta_{NN}\omega_{NN}s + \omega_{NN}^2}{s^2 + 2\beta_{ND}\omega_{ND}s + \omega_{ND}^2} U_{FF}(s),$$

Fig. 4-19: Transfer function of the notch filter

where $\omega_{NN} = 2\pi f_{NN}$ and $\omega_{ND} = 2\pi f_{ND}$. The notch filter may be used as a pure notch filter ($f_{NN} = f_{ND}$) to suppress a certain frequency, or as a second order lead-lag filter (if $f_{ND} > f_{NN}$) or lag-lead filter (if $f_{NN} > f_{ND}$). Besides the frequencies, the damping (β) of the two filters can be set as well

- NFA_NUM_DAMP , NFB_NUM_DAMP , NFC_NUM_DAMP , for the high pass filter
- NFA_DEN_DAMP , NFB_DEN_DAMP , NFC_DEN_DAMP , for the low pass filter

where $0 \leq \beta_{NN} < 1$ and $0 \leq \beta_{DN}, \beta_{DN} < 1$. The filter has DC-gain equal to 1.

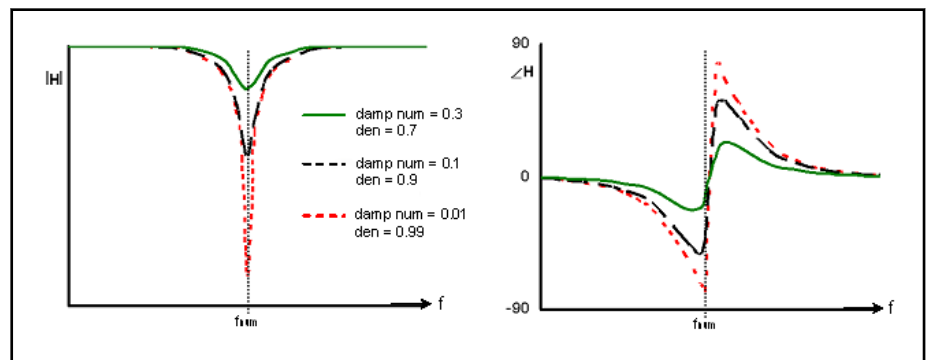


Fig. 4-20: Notch filter frequency response ($f_{\text{numerator}} = f_{\text{denominator}}$)

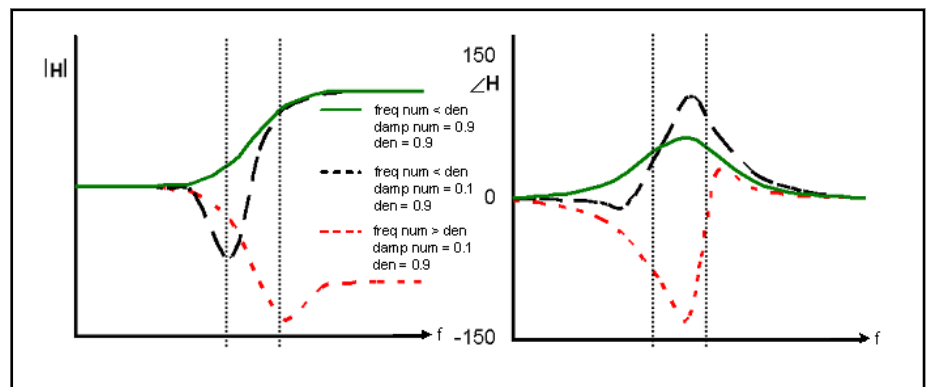


Fig. 4-21: Notch filter frequency response ($f_{\text{numerator}} \neq f_{\text{denominator}}$)

NYCeTuner

Each notch filter can be switched on and off with its switch parameter (NFA_SWITCH, NFB_SWITCH, NFC_SWITCH).

4.9 Current control parameters

NYCeTuner can selectively display the current control parameters via the "View" menu, see [tab. 4-6 "Current control parameter list" on page 138](#). The control lay-out of the current controller is shown in [fig. 4-22 "The current controller lay-out" on page 138](#). These parameters do not apply for the NY4125 and NY4130 as these drive modules do not have a CCL.

The parameters `CC_KI` and `CC_KP` are the PI controller parameters. The `CC_KI` and `CC_KP` value is checked for overflow at the moment the drive power is enabled or, if the drive power is already enabled, when the `CC_KI` or `CC_KP` value is changed. Drive power is enabled via the functions `SacEnablePower` or `SacLock`, see the Axis State Transition diagram in the NYCe 4000 Software User Manual. Overflow occurs when the value is higher than 32767.

If `CC_KI` is invalid the error `CML_ERR_CC_KI_TOO_HIGH` is returned, if `CC_KP` is invalid the error `CML_ERR_CC_KP_TOO_HIGH` is returned.

The maximum servo voltage to be used for current control can be set with the parameter `CC_SAT_LEVEL`, and the maximum integrator level can be set with the parameter `CC_MAX_INTEGRATOR_LEVEL`.

Name	Description (displayed by NYCeTuner)	Unit
<code>CC_KI</code>	Integral gain of current loop	rad/s
<code>CC_KP</code>	Proportional gain of current loop	V/A
<code>CC_MAX_INTEGRATOR_LEVEL</code>	Maximum level of integrator in current loop	V
<code>CC_SAMPLE_TIME</code>	Sample time of current loop	s
<code>CC_SAT_LEVEL</code>	Saturation level of current controller output	V

Tab. 4-6: Current control parameter list

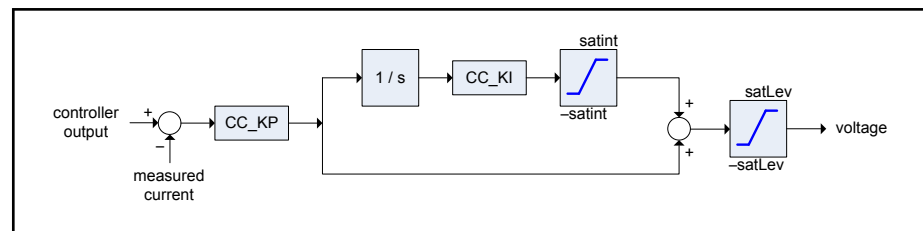


Fig. 4-22: The current controller lay-out



If a BLAC motor is connected to a PWM drive module (NY4120, NY4125, or NY4140) a torque ripple will occur if the required servo voltage level is not sufficient. To avoid this torque ripple you must set the saturation level at 87.6% ($\frac{1}{2}\sqrt{3}$) of the servo voltage. For a BLAC motor connected to the NY4120 or NY4140 drive module you must set the saturation level of the current controller output. For a BLAC motor connected to an NY4125 drive module, use the "saturation level", see [chapter 3.7.6 "Motor" tab](#) on [page 102](#).

IndraDrive

If the selected axis is on the IndraDrive, the parameter `CC_SAMPLE_TIME` is fixed and determined by the IndraDrive, based on the PWM frequency. The parameter `CC_MAX_INTEGRATOR_LEVEL` and `CC_SAT_LEVEL` are not used in the IndraDrive and are set to zero for an axis on an IndraDrive.

4.10 Stepper control parameters

NYCeTuner can display the Stepper control parameters via the "View" menu, see [tab. 4-7 "Stepper control parameter list" on page 139](#). The meaning of each parameter is shown in [fig. 4-23 "Stepper motor velocity / current profile" on page 140](#).

Name	Description (displayed by NYCeTuner)	Unit
POST_ACC_DELAY	Stepper motor post acceleration delay time	s
POST_DEC_DELAY	Stepper motor post deceleration delay time	s
PRE_ACC_DELAY	Stepper motor pre acceleration delay time	s
PRE_DEC_DELAY	Stepper motor pre deceleration delay time	s
STEPPER_ACCELERATION_CURRENT	Stepper motor acceleration current	A
STEPPER_ACCELERATION_THRESHOLD	Stepper motor acceleration threshold	pu/s ²
STEPPER_BASE_VELOCITY	Stepper motor base velocity	pu/s
STEPPER_CONSTANT_VEL_CURRENT	Stepper motor constant velocity current	A
STEPPER_DECELERATION_CURRENT	Stepper motor deceleration current	A
STEPPER_HOLD_CURRENT	Stepper motor hold current	A
STEP_DELAY_COMPENSATION_TIME	Time to compensate for calculation delay in the current controller for a stepper motor	s

Tab. 4-7: Stepper control parameter list

NYCeTuner

Because the stepper motor is used without feedback, this type of axis has less parameters than, for example, a servo axis. However, current loop parameters also apply for the stepper motor.



The stepper motor can have a position sensor. This is called a "sensing stepper". However, the position sensor data is not used in the control algorithm.

The position sensor is only used to read the actual stepper position and can be used for homing of a stepper axis. The resolution of the sensor must be higher than or equal to half of a micro step.

IndraDrive

The stepper motor is not supported on the IndraDrive, thus all mentioned stepper control parameters are not used in the IndraDrive.

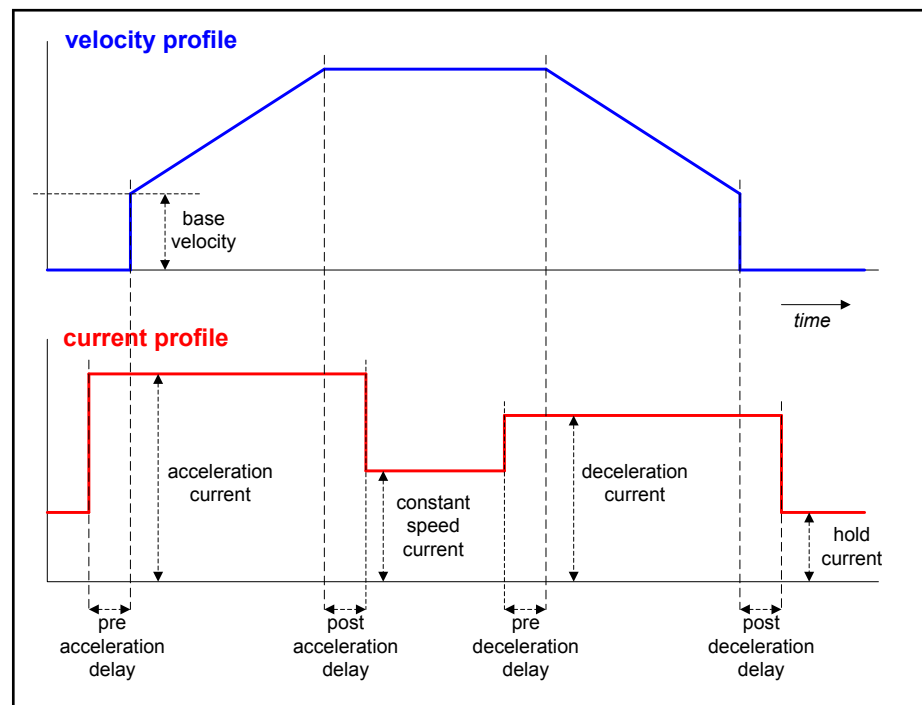


Fig. 4-23: Stepper motor velocity / current profile

The `STEPPER_BASE_VELOCITY` parameter sets the start speed of stepper motor (to prevent the motor from losing steps while going through the mechanical instability region).

Four current levels can be defined.

- During acceleration and deceleration the stepper motor must produce the highest torques (for accelerating / decelerating the load), and thus generally needs the highest current levels.
- When performing a constant velocity movement, the required motor torque will be lower than the motor torques during acceleration and deceleration. In most cases the motor current at constant speed can be set to a lower level than the currents needed during acceleration and deceleration.

- At stand still the required motor torque will be at its lowest, making it common to even further lowering the motor current.

The stepper motor currents cannot be increased or decreased instantaneously (when changing from one state to another state). The delay parameters make it possible to change the winding currents before actually accelerating or decelerating (`PRE_ACC_DELAY` and `PRE_DEC_DELAY`), and leave the currents at the acceleration or deceleration current levels for a short time after the change acceleration or deceleration (`POST_ACC_DELAY` and `POST_DEC_DELAY`).

`STEPPER_ACCELERATION_THRESHOLD` is a parameter to reduce the motor current during low accelerations or decelerations. If acceleration or deceleration of the stepper motor is below the threshold, the motor currents are equal to the constant velocity motor currents (`STEPPER_CONSTANT_VEL_CURRENT`). If above the threshold, motor currents are equal to the acceleration or deceleration motor currents (`STEPPER_ACCELERATION_CURRENT` or `STEPPER_DECELERATION_CURRENT`). This parameter is only supported for splines.

During the calculation of the stepper motor winding currents (by the current controller), an estimate of the stepper axis position is used twice. Because of the delay between both calculations, it may be necessary to provide a compensation for the change of the axis position during this delay. With the parameter `STEP_DELAY_COMPENSATION` the calculation delay can be supplied, improving the stepper axis behavior in some situations.

4.11 Alignment parameters (BLAC and IndraBLAC only)

4.11.1 Introduction

Initially, when power is applied, the position of the motor windings (denoted R, S and T and sometimes U, V, and W) relative to the magnets is unknown. The current through the windings R, S, and T of the motor sums up to 0. Therefore, the current values can be written as $I_R = I * \cos(\Phi)$, $I_S = I * \cos(\Phi+120^\circ)$, and $I_T = I * \cos(\Phi+240^\circ)$, where I is the signed current amplitude and Φ is the commutation angle in rad. The commutation angle rotates with a period corresponding to the distance between two successive pole pairs of the motor. If the length of the magnetic period, p , expressed in raw position units (inc) of the position encoder is known, the angle Φ can therefore be updated on the basis of the measured raw axis position with the formula $\Phi = \Phi_{\text{offset}} + 2\pi * d * x/p$. In this formula d is the commutation direction and x is the measured raw position in inc. As the initial raw position is 0, Φ_{offset} is the initial (unknown) angle between the position of the motor windings and a pole pair of the magnets. Φ_{offset} is called the commutation offset and is specified in rad.

The force in the direction of the movement, generated by the magnetic fields of the permanent magnets and the motor windings is proportional to the amplitude of the current I and $\cos(\Phi_{\text{offset}} + 2\pi * d * x/p - \Phi)$. To have the maximum force in the direction corresponding to the sign of I , the commutation offset must be determined in such a way that $\Phi_{\text{offset}} + 2\pi * d * x/p - \Phi$ equals 0 (mod 2π). The method to determine the commutation offset is called the alignment procedure, as it determines how the magnetic fields of the permanent magnets and the motor windings are initially aligned.

Commutation direction

NYCeTuner

Alignment can be performed once the commutation direction is set with the parameter `COMMUTATION_DIRECTION`. The alignment methods Kick & Swing, Wake & Shake, Catch & Move and Push & Settle autonomously determine the commutation offset, see the following chapters.

Do the following steps to set the commutation direction of an IndraDrive without encoder flash memory.

1. Click the "Edit" button in the "Test Signal" section in NYCeTuner. See [chapter 4.15.2 "Test signal generator" on page 161](#) for detailed information.
2. Configure a block wave signal with a small current (set "Amplitude" to 0.2 .. 0.5 A) and a frequency of 100 Hz.
3. Set the injection point of the signal to "B: at controller out, open loop".
4. Click the "OK" button in the "Test Signal Editor" dialog window.
5. Click the "Align Motor" button in the "Axis Tuning Control" section in NYCeTuner.

After the alignment has finished, click the "Open Loop" button and then the "Start Test Signal" button. You must move the axis of the motor manually to verify that the commutation direction is set correctly. The axis generates a humming sound. If the humming sound has a constant volume over the full stroke, the commutation direction is correct. If the volume of the humming sound depends on the axis position, the commutation direction is probably wrong. Change the commutation direction with the parameter `COMMUTATION_DIRECTION`. Save the axis configuration and initialize the axis again.

Alignment parameters

[tab. 4-8 "Alignment parameter list" on page 142](#) shows the alignment parameters as displayed by the NYCeTuner when selecting the parameters via the "View" menu.

Name	Description (displayed by NYCeTuner)	Unit
<code>ALIGNMENT_MODE</code>	Alignment mode for brushless AC motor 0 - none 1 - Kick & Swing ① 2 - Wake & Shake ① 3 - Set commutation offset ① 4 - Encoder position ① 5 - Catch & Move ① 6 - Indra saturation ② 7 - Indra sine wave ② 8 - Indra locked ② 9 - Push & Settle ①	-
<code>CATCH_AND_MOVE_XXX</code>	Catch & Move alignment parameters See tab. 4-9 "Catch & Move alignment parameters" on page 150	
<code>COMMUTATION_DIRECTION</code>	Commutation direction (-1=neg, 1=pos)	-
<code>COMMUTATION_OFFSET</code>	Commutation angle (mode 3) or commutation offset (mode 4)	rad

Name	Description (displayed by NYCeTuner)	Unit
INDRA_TEST_SIGNAL_START_AMPLITUDE	Amplitude for angle acquisition on IndraDrive (alignment mode 6 and 7 only)	V or %
INDRA_TEST_SIGNAL_START_FREQUENCY	Test frequency for angle acquisition on IndraDrive (alignment mode 6 and 7 only)	Hz
MAX_ALIGNMENT_OUTPUT	Maximum controller output during alignment procedure	cout
PUSH_AND_SETTLE_RAMP_TIME	time used during the push phase (alignment mode 9 only)	s
WAKE_AND_SHAKE_CRIT_DISPL	Critical displacement: threshold (alignment mode 2 only)	pu
WAKE_AND_SHAKE_PERIOD_TIME	Period time of vibration signal (alignment mode 2 only)	s

- ① Alignment method not available for IndraBLAC.
- ② Alignment method only for IndraBLAC.

Tab. 4-8: Alignment parameter list

The 9 modes supported for the alignment of a brushless AC motor are described in the following chapters.

4.11.2 Mode 1: Kick & Swing

⚠ WARNING

The Kick & Swing alignment has no reliable result and may cause unstable axis alignment which may cause personal injury or machine damage.

Do not use the Kick & Swing alignment procedure in a machine in a production environment. For an alternative alignment method, see [chapter 4.11.10 "Mode 9: Push & Settle" on page 152](#).

In the Kick & Swing procedure, the motor coils receive currents defined by magnetic angle 0° with specified maximum amplitude MAX_ALIGNMENT_OUTPUT. The axis jumps ("kick") to magnetic angle position 0° and settles ("swing") on this position.

The software expects that the axis moved to the position where the magnetic angle is 0°.

MAX_ALIGNMENT_OUTPUT is the maximum value to which the algorithm increases the current applied to the motor, or the voltage if the motor is connected to an NY4125 drive module. If the controller output is configured in Amperes, a typical value is 50% to 80% of the maximum continuous motor current; likewise for the voltage if the motor is connected to an NY4125 drive module. MAX_ALIGNMENT_OUTPUT is saturated with the parameter MAX_CURRENT.

If the parameter is set too small, the alignment may be unsuccessful. If it is set too high, the movement may be too rough compared to the user limits.



- Kick & Swing causes a movement of the motor of (maximum) ½ pole pair. If the movement of the axis is undesirable, see mode 2 "Wake & Shake". Mode 2 applies a minimal current (which is stepwise incremented) until the axis moves. This movement is minimal. Where Kick & Swing uses a current to force a defined commutation angle, Wake & Shake uses an angle offset to find the correct commutation angle with a minimal applied current.
- If the alignment is not successful, there will be no error message.

4.11.3 Mode 2: Wake & Shake

The Wake & Shake procedure applies a vibration signal to the motor to find the correct commutation angle with a minimal displacement of the axis. The alignment procedure seeks the angle of the axis where the force is 0 and starts with a vibration signal of low amplitude. If the axis does not move (the movement is smaller than the specified `WAKE_AND_SHAKE_CRIT_DISPL` value) the amplitude is increased in small steps. If the axis moves ("wake") more than the specified maximum allowed displacement value ("shake"), the current is immediately switched off. The angle is adapted and the vibration signal is switched on again. This cycle is repeated until the amplitude of the vibration signal equals the specified maximum amplitude `MAX_ALIGNMENT_OUTPUT` and the axis displacement is lower than the specified `WAKE_AND_SHAKE_CRIT_DISPL` value. In this position the motor torque is zero while the current equals the `MAX_ALIGNMENT_OUTPUT` value. The correct commutation angle is calculated from this position.

The commutation offset value can be read in NYCeWatch, from the variable with identifier `DEH_CML_AXC_COMMUTATION_OFFSET` located in the `{axisname}\CML\AXC\ tree`.

The Wake & Shake alignment mode has 3 parameters.

- `MAX_ALIGNMENT_OUTPUT` is the maximum value to which the algorithm increases the current applied to the motor, or the voltage if the motor is connected to an NY4125 drive module. If the controller output is configured in Amperes, a typical value is 50% to 80% of the maximum continuous motor current; likewise for the voltage if the motor is connected to an NY4125 drive module. `MAX_ALIGNMENT_OUTPUT` is saturated with the parameter `MAX_CURRENT`.
- `WAKE_AND_SHAKE_CRIT_DISPL` is the maximum allowed displacement. One pole pair represents 360° magnetical. A typical value for the allowed displacement is in the range of 0.5° to 5° magnetical. The parameter must be expressed in position units, pu. For the conversion of x magnetical degrees to position units the following formula can be used:

$$\text{WAKE_AND_SHAKE_CRIT_DISPL} = \frac{x}{360^\circ} * \frac{\text{BLAC_PU_PER_NR_OF_POLE_PAIRS}}{\text{BLAC_NR_OF_POLE_PAIRS}}$$

Fig. 4-24: Displacement formula

- `WAKE_AND_SHAKE_PERIOD_TIME` is the period time of the vibration signal used by the Wake & Shake procedure. The axis should vibrate within the bandwidth of the mechanical system. A typical value is 30 ... 50 ms up to 100 ms, depending on the inertia mass.



If one of these three parameters is not well chosen, the alignment may be unsuccessful. If the alignment is detected to be unsuccessful, an error message will appear.

Even if you choose good parameter values, and the alignment seems to be successful, an error may occur later on. A motor can be "aligned" with correct parameters, but there are other parameters that are also important for correct alignment, such as commutation direction and the number of pole pairs. As soon as the motor is locked or moved, the axis will report an error.

4.11.4 Mode 3: Set Commutation Offset

In this mode, the axis is not moving. Using this mode, it is assumed that the axis is at a certain position (for example at an end stop), where the commutation angle is known. The parameter `COMMUTATION_OFFSET` must contain this angle, and must be set by the user.

4.11.5 Mode 4: Align on Encoder Position

This mode is only available for a motor with a Rexroth MSM absolute encoder, an EnDat2.2 encoder, or a Sanyo Denki absolute encoder, because these encoders supply an absolute position.

In this mode, the motor is not moving. The commutation offset is specified via the parameter `COMMUTATION_OFFSET`.

For MSM absolute and Sanyo Denki absolute encoders, the commutation offset equals 0 if the encoder and motor are connected as prescribed.

Prerequisites for the alignment on MSM absolute and Sanyo Denki absolute encoder position are

- the `COMMUTATION_DIRECTION` parameter is set to -1 for Rexroth MSM absolute encoder,
- the `COMMUTATION_DIRECTION` parameter is set to $+1$ for Sanyo Denki absolute encoder,
- the parameters `MEAS_SYST_DIRECTION` and `CONTROLLER_OUTPUT_INVERT` have identical signs (that is both set to $+1$ or both set to -1),
- the motor connections (U-V-W) are correctly connected to the drive module U-V-W connections.



For Rexroth MSM incremental encoders, alignment mode 1, 2, 3, 5 or 9 must be used.

For EnDat2.2 encoders, the commutation offset must be determined once using one of the other alignment methods, with the restriction that the measurement system direction and controller output inversion have identical sign. You can read the commutation offset in NYCeWatch with the data identifier `DEH_CML_AXC_COMMUTATION_OFFSET` located in the `{axisname}\CML\AXC\` tree. You must copy the read value manually to `COMMUTATION_OFFSET`.

The prerequisites for the alignment on EnDat2.2 encoder position are

- the `COMMUTATION_DIRECTION` parameter is set to $+1$ or -1 , see [chapter 4.6 "Standard controller parameters" on page 121](#),

NYCeTuner

- the parameters `MEAS_SYST_DIRECTION` and `CONTROLLER_OUTPUT_INVERT` have identical signs (that is both set to +1 or both set to -1).

After you have determined the commutation offset for the EnDat2.2 encoder, this commutation offset can be used for future alignment on encoder position, even after power down of the system.

4.11.6 Mode 5: Catch & Move



Make sure that you have set the parameters `CONTROLLER_OUTPUT_INVERT`, `MEAS_SYST_DIRECTION` and `COMMUTATION_DIRECTION` correctly before you use the Catch & Move alignment procedure. See [chapter 4.6 "Standard controller parameters" on page 121](#) for more information.

You can use the "Catch & Move" alignment method for axes that have a preferred position at one side against or near an endstop (for example caused by the pull of attached cables) or for non-horizontal axes where the preferred position is caused by gravitation. An obvious example is an axis that moves in the vertical direction; the preferred position is always at the bottom. Catch & Move can also be used when Wake & Shake does not lead to an aligned axis, for example because the axis moves against a mechanical endstop during the movement used in the alignment procedure.

In the normal situation the axis will be in its preferred position. The Catch & Move algorithm assumes that the specified "`ANGLE_ALPHA`" parameter is at the preferred position unless this is too close to the endstop, then a safe position way from the endstop is set. The "`DISTANCE`" parameter specifies the distance moved during the alignment. The sign of the "`DISTANCE`" parameter defines the movement direction. If the preferred position of the axis is near the negative endstop, the sign of the "`DISTANCE`" parameter must be positive so that the axis moves away from the negative endstop. If the preferred position of the axis is near the positive endstop, the sign of the "`DISTANCE`" parameter must be negative so that the axis moves away from the positive endstop. Angles and distances are defined in [rad] and one pole pitch equals 2π .

The Catch & Move alignment method has 4 distinctive phases.

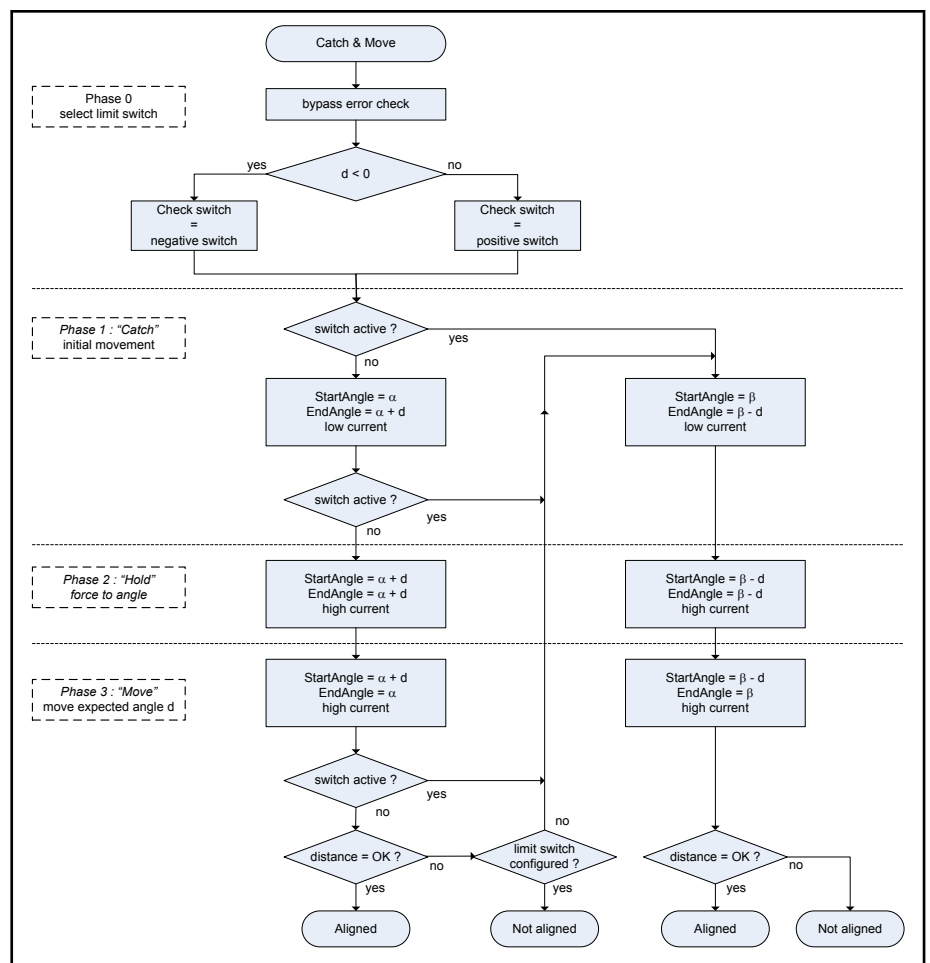


Fig. 4-25: Flow chart of "Catch & Move" procedure

- Phase 0, "prepare"
 Based on the specified movement direction, the algorithm defines which end switch will be checked during the alignment procedure. If the movement direction is positive, the positive end switch is selected. If the movement direction is negative, the negative end switch is selected.
- Phase 1, "catch"
 The state of the selected end switch is checked. The switch can be activated or not activated. If the end switch is not activated, the algorithm assumes that the axis is in its preferred position and uses "ANGLE_ALPHA" + "DISTANCE" to set the end position of the movement. If the end switch is activated, the axis is at the opposite side of the preferred position, and the algorithm uses "ANGLE_BETA" – "DISTANCE" to set the end position of the movement. The movement is executed with a low current defined by the "MIN_CURRENT_LEVEL" parameter to prevent a jump to the start angle during the initial movement. The parameters "RAMP_TIME" and "MOVE_TIME" set the allowed time for the movement. Note that this first movement is an uncontrolled movement. During this movement the rotating magnet field of the coils will "catch" the rotor or translator (permanent magnets) of the motor. The axis is moved the distance (in terms of the pole pitch) specified by the "DISTANCE" parameter.

NYCeTuner

Because the initial commutation angle is not yet valid, it is unknown into which direction the first movement will be. Even if limit switches or an area switch are used, this is still the case. A collision against a hard endstop cannot be prevented without pre-knowledge. Parameters need to be set to minimize collision risks.

Further, it is possible that the axis is not in its preferred position, but neither at the opposite side. If the selected end switch is activated during the movement using "ANGLE_ALPHA" + "DISTANCE", the algorithm starts the "catch" phase again, but uses the "ANGLE_BETA" - "DISTANCE" to set the end position of the movement.

- Phase 2, "hold"

After the movement executed in the "catch" phase, a high current defined by the parameter "MAX_CURRENT_LEVEL" is applied during the time defined by the "HOLD_TIME" parameter. This current forces the rotor or translator to the correct position and to stay at that position.

- Phase 3, "move"

The actual position is read at the start of phase 3. Now a movement is executed with the defined "DISTANCE" d , but opposite to the direction used in the movement in phase 1. If the selected end switch is activated during this movement or is activated after the movement, the initial position of the axis was not its preferred position but at the opposite side; therefore the algorithm restarts the "catch" phase of the alignment procedure with the "ANGLE_BETA" parameter.

If the selected end switch is not activated, the end position of the axis is checked. If the difference between the actual distance and the expected distance based on "DISTANCE" is smaller than the deviation specified by the parameter "DEVIATION", the commutation angle is correct and the motor is aligned.

From the flow chart it is clear that in the case that no limit switches are configured, and the position change in phase 3 is not as expected, the alignment procedure is repeated in the other direction and angle β is used.

The commutation offset value can be read in NYCeWatch, from the variable with identifier `DEH_CML_AXC_COMMUTATION_OFFSET` located in the `{axisname}\CML\AXC\ tree`.

Parameter settings

For a correct functioning of this alignment procedure and minimization of collision risks, some parameter values need to be estimated.

A proper start angle needs to be found for each endstop (if present). If the commutation angle is once known (by using another alignment procedure), these start angles can be determined. This can be done by moving the axis (for example manually) towards each endstop and reading the commutation angle, for example in NYCeScope. Then, a proper start angle can be defined at a certain distance from each endstop, see [fig. 4-26 "How to set commutation angles at endstops" on page 149](#). The distance from the endstop must be smaller than π [rad], but not too small, otherwise the uncontrolled movement in phase 1 can cause a collision against the endstop. Typically, a distance of $\frac{1}{2}\pi$ is chosen.

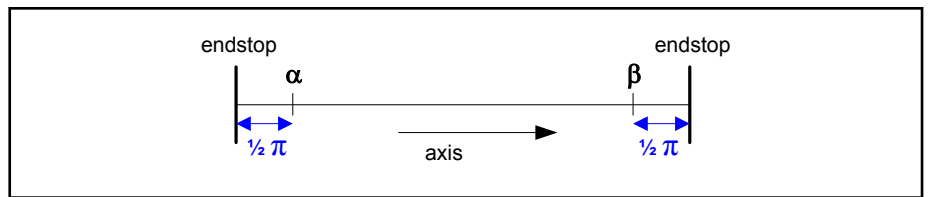


Fig. 4-26: How to set commutation angles at endstops

The sign of the distance parameter "d" defines the initial movement direction, see fig. 4-27 "Initial movement direction "d" without limit switches" on page 149.

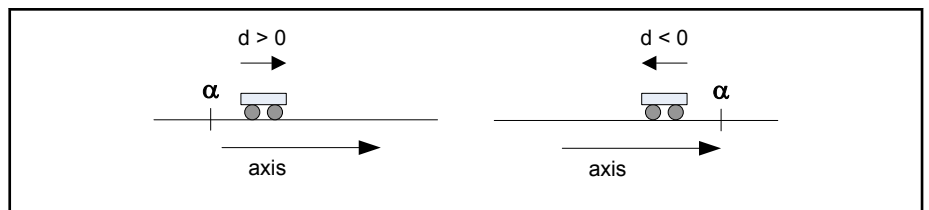


Fig. 4-27: Initial movement direction "d" without limit switches

When limit switches are configured, the algorithm knows which start angle to use if the axis is initially in an endstop. If the axis is initially not in an endstop, it is attempted to move to the limit switch as the wrong start angle (from the other endstop) may be used. If the limit switch is found, the alignment procedure will be started again using the correct start angle.

When limit switches are configured, use the settings as shown in fig. 4-28 "Movement direction "d" with limit switches" on page 149.

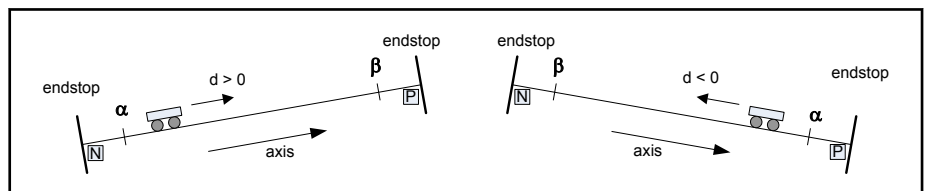


Fig. 4-28: Movement direction "d" with limit switches

If the axis has a preferred initial position (for example caused by gravitation force), the settings as shown in fig. 4-28 "Movement direction "d" with limit switches" on page 149 will give a minimum number of movements and collisions.

In an application that only contains a home area switch, this switch can be configured as a positive or negative limit switch. The area switch must be active at one endstop. To minimize collision risks, the settings should be as shown in fig. 4-29 "Use of area switch as limit switch" on page 149.

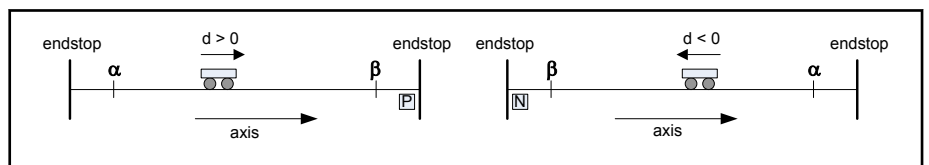


Fig. 4-29: Use of area switch as limit switch

tab. 4-9 "Catch & Move alignment parameters" on page 150 shows the parameters with typical values. Specified time and current values depend on mechanical characteristics like mass, stiffness and friction.

NYCeTuner

Name	Unit	Description	Typical value
CATCH_AND_MOVE_ANGLE_ALPHA	rad	The commutation angle at a certain distance $< \pi$ from the endstop near the preferred position	Commutation angle at $\frac{1}{2}\pi$ from endstop
CATCH_AND_MOVE_ANGLE_BETA	rad	The commutation angle at a certain distance $< \pi$ from the endstop at the opposite side of the preferred position	Commutation angle at $\frac{1}{2}\pi$ from endstop
CATCH_AND_MOVE_DISTANCE	rad	The distance to move during the alignment, in terms of the pole pitch	$\frac{1}{2}\pi - \pi$, depending on stroke length
CATCH_AND_MOVE_DEVIATION	%	The allowed deviation between angle movement and actual position change, as a percentage of pole pitch (2π)	5%
CATCH_AND_MOVE_RAMP_TIME	s	The time used to ramp up the current in phase 1 (catch) and phase 2 (hold)	1 – 2 s.
CATCH_AND_MOVE_MOVE_TIME	s	The time used to move the distance d in phase 1 (catch) and phase 3 (move)	1 – 2 s.
CATCH_AND_MOVE_HOLD_TIME	s	The time used to settle in phase 1 (catch), 2 (hold) and 3 (move)	1 – 2 s.
CATCH_AND_MOVE_MIN_CURRENT_LEVEL ⁽¹⁾	A (V)	The maximum current used in phase 1 (catch). For NY4125: the maximum voltage	10 – 30% of the maximum continuous value
CATCH_AND_MOVE_MAX_CURRENT_LEVEL ⁽¹⁾	A (V)	The maximum current used during in phase 2 (hold) and phase 3 (move). For NY4125: the maximum voltage	50 – 80% of the maximum continuous value

①

The NY4125 drive module does not have a Current Control Loop. You should read the parameter names CATCH_AND_MOVE_MIN_CURRENT_LEVEL and CATCH_AND_MOVE_MAX_CURRENT_LEVEL as the minimum and maximum level of the voltage applied to the motor.

Tab. 4-9: Catch & Move alignment parameters

fig. 4-30 "Current, commutation angle and axis position during the alignment procedure" on page 151 shows the current, commutation angle and the actual axis position during the phases. As shown, the movement in phase 1 depends on the initial position. There, the start angle (equal to α or β) may even not always be reached.

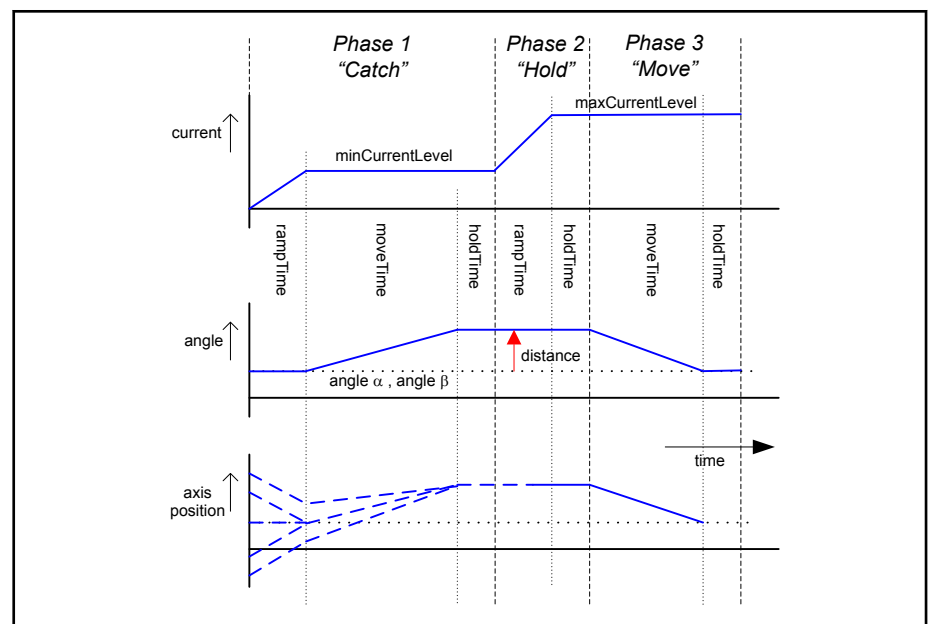


Fig. 4-30: Current, commutation angle and axis position during the alignment procedure

4.11.7 Mode 6: Indra saturation

The saturation method for determining the commutation offset can be used for synchronous motors. The commutation offset is determined with the supplied current. During the alignment, motor motion is not required although some motion may occur. If the motor has a brake, the brake must remain applied. As a prerequisite for successful use of the saturation method, the iron material of the motor must be magnetically saturated when current is applied. The saturation method is not suited for motors which are not magnetically saturated at the maximum allowed (peak) current. Ironless motors are therefore an example of motors for which the commutation offset cannot be determined with the saturation method. Use the sine wave method in these cases. If the alignment is started with the parameter `INDRA_TEST_SIGNAL_START_AMPLITUDE` set to 0, the variables `SAC_VAR_INDRA_TEST_SIGNAL_AMPLITUDE` and `SAC_VAR_INDRA_TEST_SIGNAL_FREQUENCY` hold a value specific for the connected BLAC motor after the first alignment is finished. These variables can be read and, setting the parameters `INDRA_TEST_SIGNAL_START_AMPLITUDE` and `INDRA_TEST_SIGNAL_START_FREQUENCY` to these values, the alignment procedure needs less time the next time the alignment is executed. Note that for Bosch Rexroth motors with an integrated motor encoder it is not necessary to determine the commutation offset.

4.11.8 Mode 7: Indra sine wave

You can use the sine wave method to determine the commutation offset, but this method should only be used if you cannot use the saturation method. To use the sine wave method, unrestricted movement of the axis is required. The axis must be able to move easily and freely, otherwise the quality of the offset determination is reduced or even cause the offset determination to fail. If the motor has a brake, you cannot use the sine wave method. Check if the saturation method is possible, because unrestricted movement of the axis

NYCeTuner

generally causes problems. A sinusoidal test signal is applied and the controller in the IndraDrive determines the commutation offset.

The movement of the axis can be rather rough. The IndraDrive determines optimal alignment movement parameters after the initial alignment movement. The movement is an alternating clockwise and counter-clockwise rotation with a large rotation angle which gradually becomes smaller. Because the movement starts with a large rotation angle, the alignment procedure needs time to complete. If the alignment is started with the parameter `INDRA_TEST_SIGNAL_START_AMPLITUDE` set to 0, the variables `SAC_VAR_INDRA_TEST_SIGNAL_AMPLITUDE` and `SAC_VAR_INDRA_TEST_SIGNAL_FREQUENCY` hold a value specific for the connected IndraBLAC motor after the first alignment is finished. These variables can be read and, setting the parameters `INDRA_TEST_SIGNAL_START_AMPLITUDE` and `INDRA_TEST_SIGNAL_START_FREQUENCY` to these values, the motion can be made smoother the next time the alignment is executed. In this case the alignment procedure needs less time than the rough movement, because the clockwise and counter-clockwise rotation starts with a smaller rotation angle. The smooth movement can be more rough at low movement speeds depending on the type of motor. The default value for the start amplitude is 0, the range is 0 ... 1000. The default value of the start frequency is 500 Hz, the range is 0 ... 1000 Hz. The movement is rough when the start amplitude is set to 0.

Note that for Bosch Rexroth motors with an integrated motor encoder it is not necessary to determine the commutation offset.



The variables `SAC_VAR_INDRA_TEST_SIGNAL_AMPLITUDE` and `SAC_VAR_INDRA_TEST_SIGNAL_FREQUENCY` are **not** updated with the Indra sine wave method if the motor has an integrated motor encoder.

4.11.9 Mode 8: Indra locked

The Indra locked mode is an alignment procedure for an IndraDrive after the initial commissioning of the IndraDrive. This alignment is a motionless procedure. It only sets the flag in the NYCe 4000 software indicating that the alignment procedure is performed.

4.11.10 Mode 9: Push & Settle

The "Kick & Swing" alignment mode may result in an incorrect commutation offset, because the motor may not move when the axis is in an unstable zero position, and further, the stabilization time may not be sufficient. The "Kick & Swing" alignment may produce a large and rough axis movement. The alignment mode "Push & Settle" overcomes these issues, and can be used for the motor type `SAC_BRUSHLESS_AC_MOTOR`, and for `SAC_COIL_DRIVE` and `SAC_COIL_INDRADRIIVE` in the Linear Motion System. The "Push & Settle" alignment may produce a large, but smooth, movement. The "Push & Settle" alignment consists of 3 phases.

1. Set the comutation angle initially to $-\pi/2$ and increment the current from 0 to a maximum value for a number of seconds. The axis of the motor always moves away from an unstable zero position. Parameters to set these values are `MAX_ALIGNMENT_OUTPUT` and `PUSH_AND_SETTLE_RAMP_TIME`, respectively.
2. Increment the commutation angle from $-\pi/2$ to 0 in 1 second.

3. Wait until the axis of the motor is settled. The axis is settled, if during 0.25 seconds the axis positions differ less than 0.1 radian of the pole pitch.

4.12 Miscellaneous NYCeTuner parameters

A selected list of parameters that are not filtered out by using the "View" menu of the NYCeTuner are shown in [tab. 4-10 "Miscellaneous parameters of the NYCeTuner" on page 153.](#)

Name	Description (displayed by NYCeTuner)	Unit
ANTI_ALIASING_FILTERS_ENABLE	Enable or disable anti aliasing filter	-
ECG_PREDICTION_FILTER_FREQ	ECG prediction filter frequency, the maximum allowed value is sample frequency * 0.49	Hz
ECG_PREDICTION_HORIZON	ECG prediction filter, number of samples to be looked ahead	-
ENCODER_GLITCH_SUPPRESSION_TIME	Glitch suppression time for quadrature and sine/cosine based encoders	-
IGNORE_ENCODER_ERRORS	Enable or disable (ignore) digital encoder axis errors	-
OPEN_LOOP_RAMP	Ramp of position controller output in SAC_FREE state	cout/s
OPEN_LOOP_SAFE_VALUE	Position controller output in SAC_FREE_STOPPED state	cout
OPEN_LOOP_VALUE	Position controller output in SAC_FREE state	cout
PFC_FORCE_INPUT	Position force input	N
PFC_FORCE_THRESHOLD	Position force threshold	N
PFC_SOURCE_SELECTION	Position force source selection	-
S0S90_SINGLE_ENDED	Measuring system single ended (TRUE) or differential (FALSE) mode	-
STOP_ALARM_ACTIVE_LEVEL	Stop alarm polarity for NY4125 and NY4130	-
TRIGGERED_STOP_ACC	Maximum deceleration used with stop on event function	pu/s ²
TRIGGERED_STOP_JERK	Maximum jerk used with stop on event function	pu/s ³

Tab. 4-10: Miscellaneous parameters of the NYCeTuner

The electronic camming and gearing prediction filter frequency (ECG_PREDICTION_FILTER_FREQ) is a parameter for axes that are driven via a cammed or geared relationship. The maximum value for this parameter equals the sample frequency * 0.49. If the prediction filter frequency is set while the sample frequency is 2 kHz (on the node where the slave axis is), and the sample frequency is changed to 1 kHz, the axes go to the SAC_IDLE state. When the axes are initialized, two conditions are possible. If the prediction filter frequency is \leq new sample frequency * 0.49, the prediction filter frequency is not changed. If the prediction filter frequency is $>$ new sample frequency * 0.49, the error ECG_ERR_INVALID_PARAMETER is issued. The NYCe 4000 Software User Manual contains more information about camming and gearing.

The open loop parameters set the controller out value, when the axis is set to the state SAC_FREE or SAC_FREE_STOPPED. The parameter OPEN_LOOP_VALUE sets the output value of the controller out, while the parameter OPEN_LOOP_RAMP defines the ramp to the open loop controller out. [fig. 4-31 "The open loop controller out parameters" on page 154](#)

NYCeTuner

situation "normal operation" shows the open loop response during normal operation of the NYCe 4000 system.

The parameter `IGNORE_ENCODER_ERRORS` enables the application to (temporarily) ignore certain errors generated by digital position encoders. The default value of this parameter is `FALSE` when the axis is initialized. The parameter can be written with `SacWriteParameter` and read with `SacReadParameter` in all axis states. The parameter is not saved in the XML configuration file to avoid possible errors during axis definition and configuration. When the parameter value is changed from `FALSE` to `TRUE`, the home flag is reset and encoder error detection is disabled. When the parameter value is changed from `TRUE` to `FALSE`, any existing error of an encoder is cleared and encoder error detection is enabled. If the parameter is written for an IndraDrive-based axis, the error `CFG_ERR_PARAMETER_NOT_SUPPORTED` is returned. The following errors can be disabled.

- Errors from digital position encoders (MSM absolute, MSM incremental, EnDat2.2, Sanyo Denki).
- Phase alarm of S0S90 and SinCos encoders.

Errors such as cable alarm and communication lost cannot be ignored.

When the NYCe 4000 system is set to the service mode, the `OPEN_LOOP_VALUE` and `OPEN_LOOP_RAMP` parameters still apply if the axis state changes to `SAC_FREE`. However, if the axis state changes to `SAC_FREE_STOPPED`, only one parameter for the controller out can be set during the service mode, the `OPEN_LOOP_SAFE_VALUE`. [fig. 4-31 "The open loop controller out parameters" on page 154](#), situation b) "service operation" shows the service mode open loop response of the controller out.

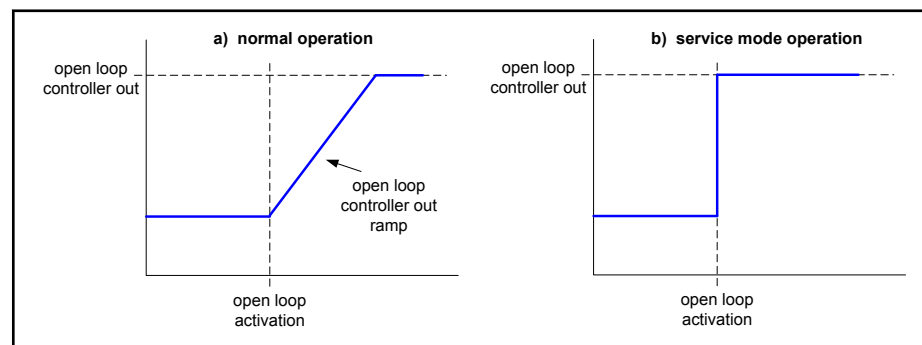


Fig. 4-31: The open loop controller out parameters

With the parameter `PFC_FORCE_THRESHOLD` the force sensor threshold (in Newton) can be set. When the actual force sensor output crosses the threshold, an event is generated. The NYCe 4000 Software User Manual contains more information about the position force control functionality.

You can indicate whether the S0/S90 encoder is connected hardware-wise differential or single ended with parameter `S0S90_SINGLE_ENDED`. If the parameter is set to 0 the S0/S90 encoder is connected differential, if the parameter is set to 1 the encoder is connected single ended. See the NYCe 4000 Hardware System manual for connection information. If a single ended S0/S90 encoder is selected, the cable break detection is disabled in software, and the cable alarm will never occur.

4.13 Controller settling criteria

You can select the controller settling criteria via the "View" menu of the NYCeTuner. [fig. 4-32 "Controller State Transition Diagram" on page 155](#) shows the State Transition Diagram of the controller, [fig. 4-33 "States for the controller with respect to setpoint position and axis position" on page 155](#) shows the controller states with respect to the setpoint position and the axis position, and [fig. 4-34 "The NYCeTuner "Settling Criteria Selection" dialog" on page 156](#) shows the Settling Criteria Selection dialog. The parameters that can be edited via this dialog can also be changed in the NYCeTuner with the parameters shown in [tab. 4-11 "The controller settling criteria parameter list" on page 155](#).

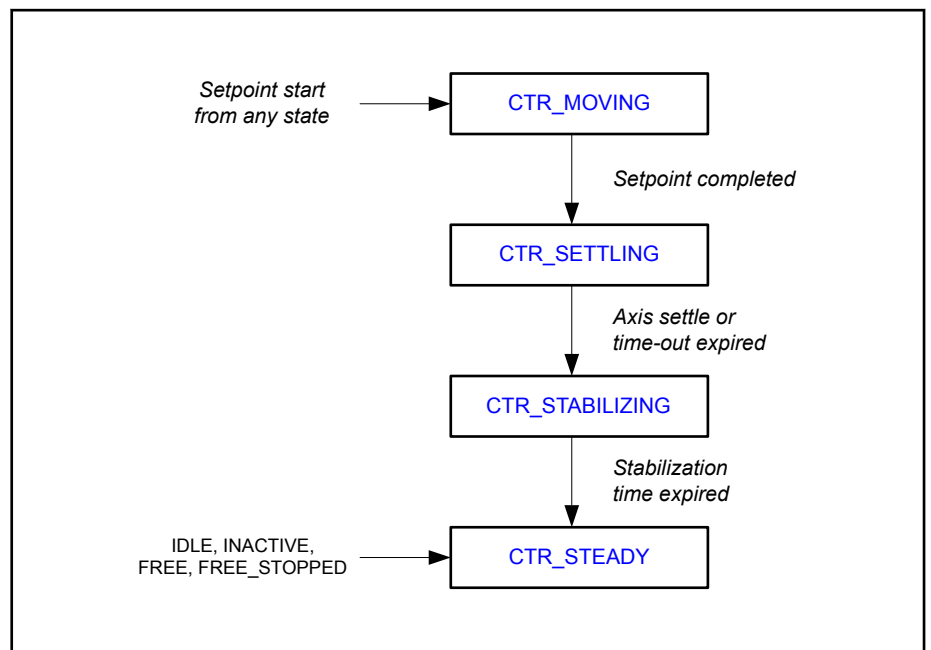


Fig. 4-32: Controller State Transition Diagram

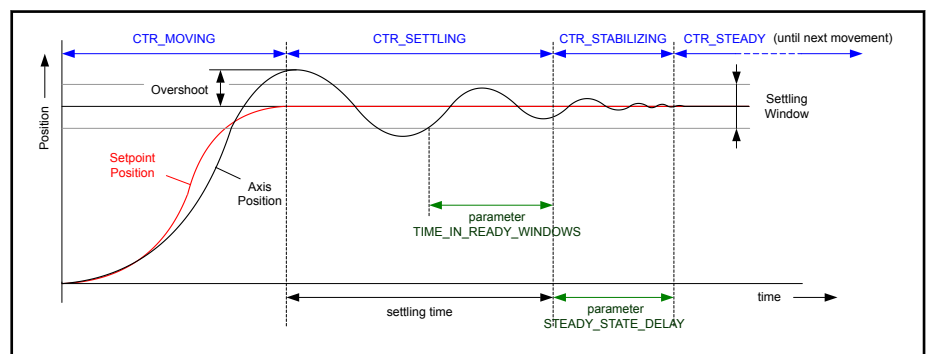


Fig. 4-33: States for the controller with respect to setpoint position and axis position

Name	Description (displayed by NYCeTuner)	Unit
POS_READY_WIN	Position ready window	pu
VEL_READY_WIN	Velocity ready window	pu/s

NYCeTuner

Name	Description (displayed by NYCeTuner)	Unit
TIME_IN_READY_WINDOWS	Time that position and velocity should be in their ready windows	s
STEADY_STATE_DELAY	After this time period the controller is set to steady	s
SETTLING_TIMEOUT	Maximum allowed time between generator ready and controller ready	s

Tab. 4-11: The controller settling criteria parameter list

The "Position" and "Velocity" plot in the Settling Criteria Selection dialog show the meaning of the parameters. The "Time in ready window" specifies the minimum time that the "Position ready window" and the "Velocity ready window" must be in their "ready window" before the controller makes the transition from the state `CTR_SETTLING` to the state `CTR_STABILIZING`.

The "Steady state delay" specifies the delay time before the controller state changes from `CTR_STABILIZING` to `CTR_STEADY`.

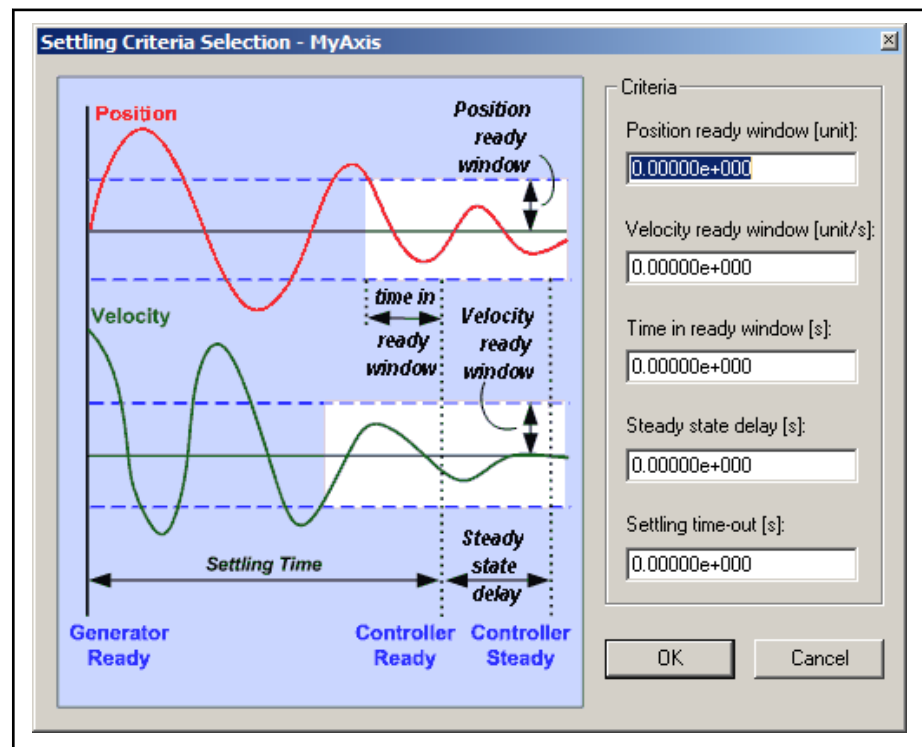


Fig. 4-34: The NYCeTuner "Settling Criteria Selection" dialog

The controller can go to the state `CTR_MOVING` at any time. After the position and velocity criteria are met, the `STEADY_STATE_DELAY` parameter defines the time the controller waits before the controller goes to the `CTR_STEADY` state (the position and velocity of the axis must stay in between their criteria). See the NYCe 4000 Software User Manual.

The parameter `SETTLING_TIMEOUT` sets the maximum time that the controller is allowed or can be in the state `CTR_SETTLING`. After that time the controller always makes the transition to the state `CTR_STABILIZING`. Even if an axis is not settled within the timeout period, the controller will go to

the next state, CTR_STABILIZING. No error handler is defined if an axis is not settled within the specified SETTLING_TIMEOUT time.

If "Settling time-out" is 0, the settling criteria are disabled and the state goes to CTR_STEADY. If a "Settling time-out" is set, you must enter valid data in the "Position ready window", "Velocity ready window", and "Time in ready window".

In the NYCeConfigurator you can set the maximum allowed settling time with the parameter MAX_SETTLING_TIME on the "Error Bounds" tab. You can also assign an error handler on the "Error Handlers" tab of the NYCeConfigurator if the MAX_SETTLING_TIME settling time is not met, see [fig. 4-35 ""Error Bounds" tab, setting the "Maximum Settling Time on page 157](#) and [fig. 4-36 ""Error Handlers" tab, setting an error handler for "Max settling Time on page 158](#).

The condition

$$\text{MAX_SETTLING_TIME} < \text{SETTLING_TIMEOUT}$$

must be met, else the error will never occur.

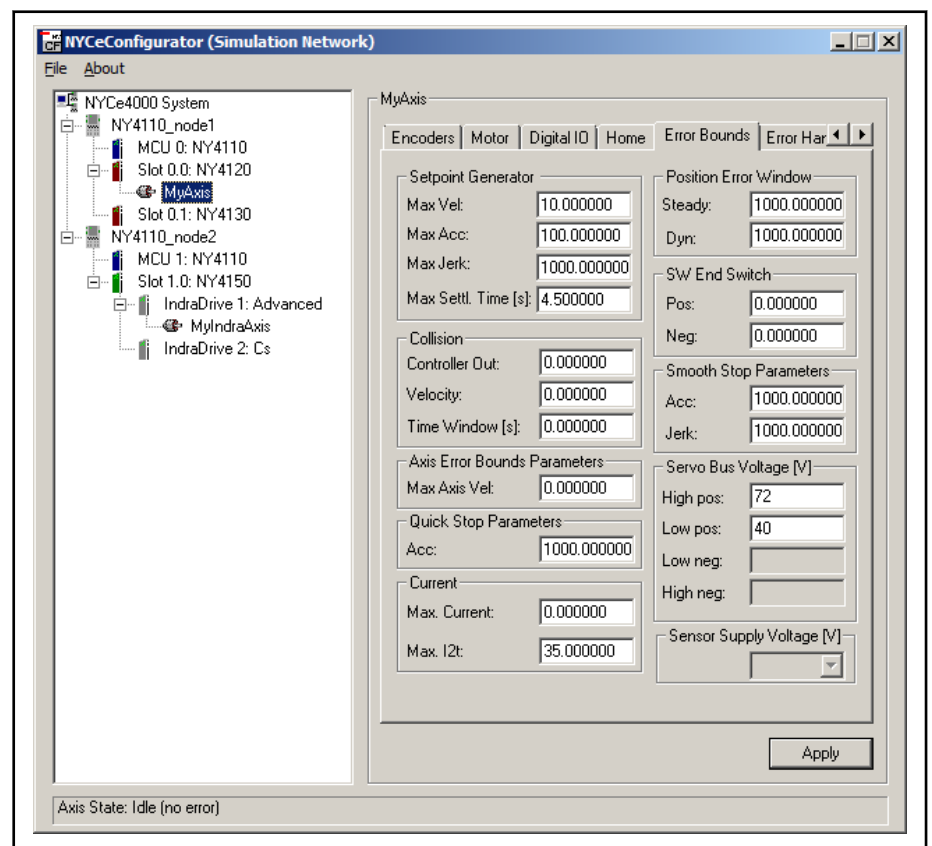


Fig. 4-35: "Error Bounds" tab, setting the "Maximum Settling Time"

NYCeTuner

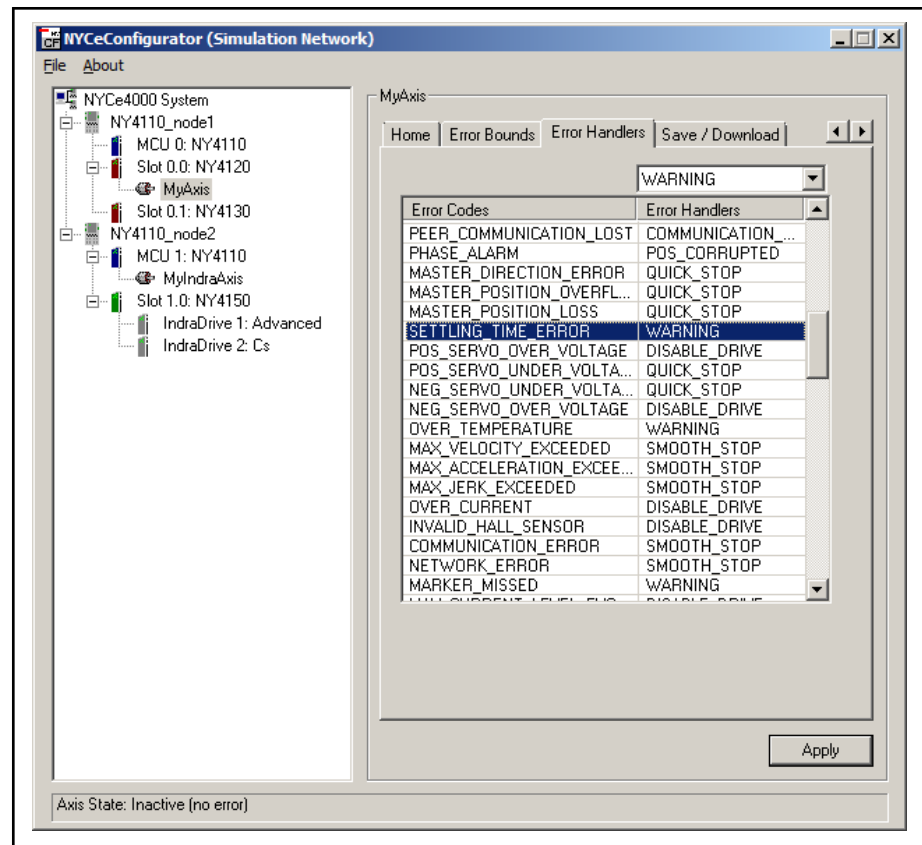


Fig. 4-36: "Error Handlers" tab, setting an error handler for "Max settling Time"

4.14 AutoTweak tool

4.14.1 Introduction

The AutoTweak tool is designed to provide an estimate of the position dependent feedforward. These position dependent feedforward values are identical every time a movement is executed. If the determined feedforward values are included in the controller, the resulting movement can be more smooth.

The AutoTweak function estimates the position dependent feedforward by performing a series of point to point movements from specified minimum and maximum position and back. After each point to point movement, the AutoTweak tool determines the controller out before the feedforwards (the output of the lead lag filter, see [fig. 4-14 "The available control filters" on page 133](#)). These values are used to create a tweak table with the position dependent feedforwards. The values are the average of the positive direction movement(s) measurement data and negative direction movement(s) measurement data.

Every point to point movement has a prior and next point to point movement, except the first movement in the positive direction and the first movement in the negative direction (the start of the Autotweak movement in both directions). Because there is no prior movement for the first point to point movement of the Autotweak, the controller out contains a movement direction dependent deviation (caused by friction and/or hysteresis). To eliminate this error the first measurement is discarded for both movement directions. The two discarded controller out values are replaced by an extrapolated value.

fig. 4-37 "The AutoTweak tool" on page 159 shows the user interface of the AutoTweak tool. The AutoTweak tool can be started via the NYCeTuner "Tools" menu. Before starting an AutoTweak the axis must be homed and in the READY state (click the "Lock" button in the "Axis Tuning Control" section of NYCeTuner).

The AutoTweak tool is only supported for the axis type "servo".



- An axis must be able to move in closed loop before performing an AutoTweak.
- You cannot change the field "Position update mode" in the AutoTweak tool, but you can change that definition of the axis with NYCeConfigurator or in your application software. It is not allowed to change the "Position update mode" (linear/modulo) while the AutoTweak tool is active.
- When you change the controller parameters of an axis after an AutoTweak, it is recommended to repeat the AutoTweak procedure.
- While executing the AutoTweak measurement procedure, it is the responsibility of the user to avoid that parameters are changed or motion commands are issued by other processes.
- The results of the AutoTweak will improve when the integrator of the standard controller is activated.

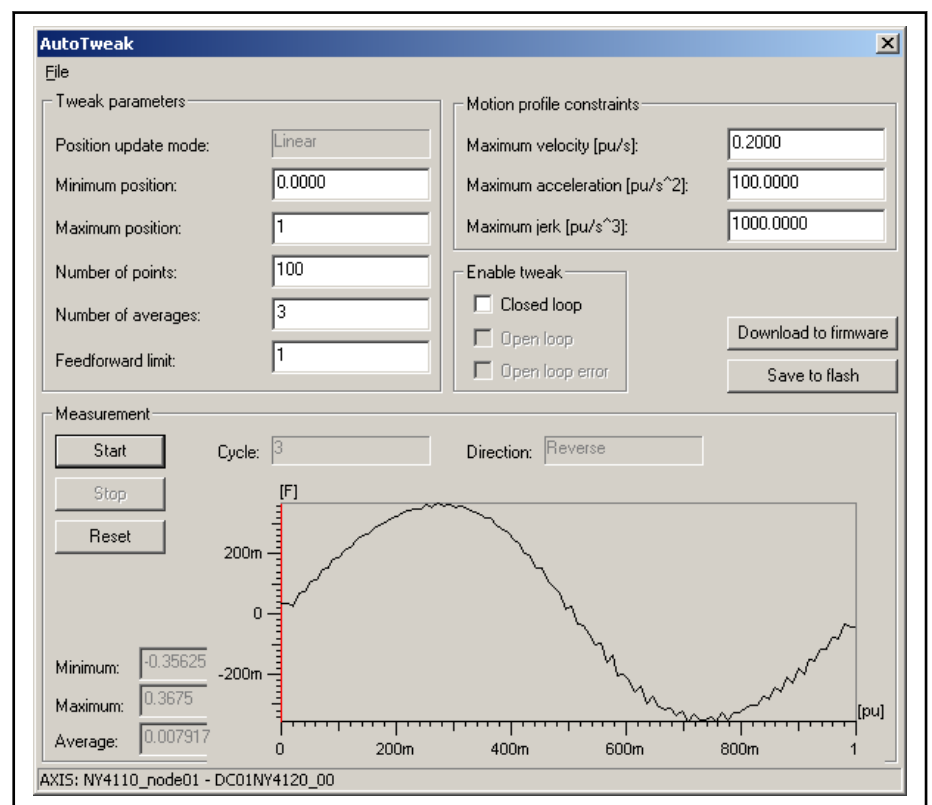


Fig. 4-37: The AutoTweak tool

4.14.2 Create the tweak data table

Do the following steps to create a tweak data table with the AutoTweak tool.

NYCeTuner

1. Enter the begin and end position limits in the fields "Minimum position" and "Maximum position", respectively.
2. Specify the number of tweak data measurement points in the "Number of points" field.
3. Specify the number of point to point movements in the "Number of averages" field. More point to point movements improves the quality of the data in the tweak table.
4. Optionally, you can limit the position dependent feedforward, if necessary. Enter the limit value in the "Feedforward limit" field.
5. Enter the motion profile for the point to point movements in the "Motion profile constraints" section.

After the "Start" button is clicked, a small plot appears in the "Measurement" section, showing the measured feedforward estimates. A vertical red line shows the progress of the measurement process. The "Cycle" and "Direction" boxes indicate the current movement. The AutoTweak procedure can always be aborted with the "Stop" button. When the AutoTweak procedure is finished and the results are satisfactory, the tweak table can be stored. The button "Download to firmware" stores the table in the firmware. The button "Save to flash" stores the table in the flash memory, but only if the table is already stored in the firmware and the tweak function is enabled. Note that you cannot download the tweak table into the firmware, if the tweak functionality is enabled.

From the firmware the tweak table is available for the controller. The check boxes in the "Enable tweak" section activate the position dependent feedforward for different axis states (closed loop, or closed loop and open loop, or closed loop and open loop and open loop error). The activation of the position dependent feedforward can also be accomplished with the parameter `TWEAK_MODE`. The parameter `TWEAK_MODE` can be set to 0 (= off), 1 (= close loop), 2 (= closed loop and open loop), or 3 (= closed loop and open loop and open loop error).

Click the "Save to flash" button to save the tweak table to flash memory. You can also save the tweak table to disk via the "File" menu. Previously saved tweak data can be retrieved from disk via the "File" menu.

4.15 NYCeTuner motion options

4.15.1 Introduction

Besides the list with parameters to setup the axis, the NYCeTuner also provides a number of stimuli to tune the different controllers of an axis.

- The Test Signal generator
- The Point To Point (PTP) motion generator
- The Jogging motion generator

The motion related data of an axis can be monitored with the bottom pane at the right hand side of the NYCeTuner window.

- The servo position and velocity.
- The Test signal generator status and output.
- The Controller status, measured position error and measured settling time.
- The Setpoint generator status, and reference position and velocity.

Besides the stimuli generated by the NYCeTuner, the DMF functionality (Drive Monitor Function) allows the measurement of the frequency response, and more, of the NYCe 4000 controller with an external measurement device like an HP frequency analyzer or a SigLab Toolbox, see [chapter 4.15.6 "Drive Monitor Function" on page 168](#).

Note that this DMF functionality using external equipment is not the same as FRF (Frequency Response Function) available in NYCeScope.

4.15.2 Test signal generator

The Test Signal generator can be configured by clicking "Edit..." button in the "Test Signal" section of the NYCeTuner. [fig. 4-38 "The Test Signal generator" on page 161](#) shows the Test Signal dialog.

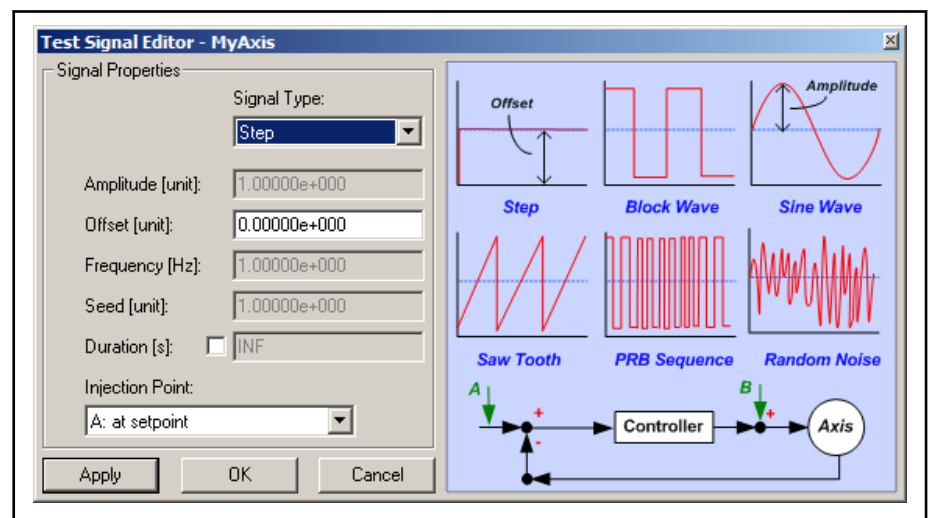


Fig. 4-38: The Test Signal generator

The test signal generator can produce the following signal types that can be chosen from the "Signal Type" drop-down box.

- step
- block wave
- sine wave
- saw tooth
- pseudo random binary sequence (PRBS)
- random noise

If you select a block wave, sine wave or saw tooth test signal, you can also set the frequency of the signal. The frequency setting depends on the electrical time constant of the motor. A good value for the duration of one pulse is 3 times the electrical time constant of the motor (rise time for 95% of the step), thus the frequency equals $1 / (2 * 3 * \text{time_constant})$. Time constants are typically in the range of 0.1 to 10 ms. If you choose the pulse time a little longer, it is not relevant for the tuning of the motor, but the motor may make small movements.

The generation of the test signal is based on the node sample time. This means that a complete period of the test signal must "fit" in an even number of the duration of one sample, thus the frequency of the test signal should be equal to $\text{sample_frequency} / (2 * n)$, where $n = 1, 2, 3 \dots$

The injection point of the test signal generator can be set using the "Injection Point" drop-down box, and can be set to

NYCeTuner

- A: at setpoint
- B: at controller out, closed loop
- B: at controller out, open loop

Before you start the test signal generator, the axis must be set to the open loop or closed loop state (depending on the chosen injection point).

By clicking the button "Start Test Signal" in the NYCeTuner overview, the test signal is started.

The test signal properties cannot be changed real time while the test signal is active. The test signal must be stopped, before the settings can be changed.

4.15.3 Point To Point motion generator

The Point To Point (PTP) motion generator can be used as a setpoint generator to move the axis a predefined distance in closed loop. The PTP generator can be configured by selecting "PTP motion" in the NYCeTuner, and clicking the "Edit..." button.

fig. 4-39 "The Point to Point Motion Editor" on page 162 shows the PTP motion editor. The PTP motion editor can be used to create a PTP motion profile, which can be displayed in the right hand part of the window by clicking the "Check Profile" button. You can select the profile movement parameters with the radio buttons below the graphic representation.

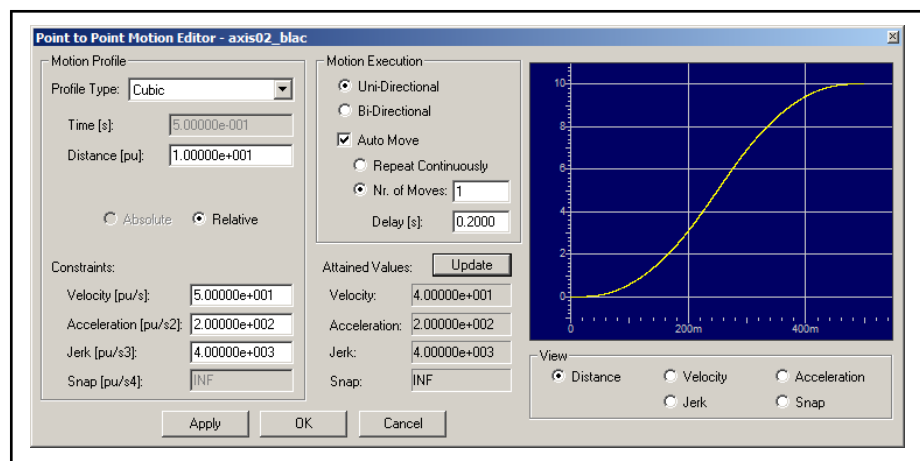


Fig. 4-39: The Point to Point Motion Editor

From the "Profile Type" dropdown box, the following profile types can be chosen.

- Parabolic
- Cubic
- Quartic
- Quartic (filtered cubic)
- Minimal jerk (cubic)
- Minimal snap (quartic)
- Energy optimized (cubic)
- Energy optimized (quartic)

The motion can be relative (move a distance from the actual position) or absolute (move a distance between begin and end position). You must enter a begin position and end position if the movement is absolute. An absolute movement can be executed from NYCeTuner only after the axis is homed. If

you select the Quartic (filtered cubic) profile, the "Snap [pu/s4]" constraint changes to "Jerk Ramp Time [s]".

You can repeat the motion profile when a check mark is put in the "Auto Move" check box. The profile can be repeated by just moving to one direction (uni-directional motion execution, with the direction defined by the distance of the PTP), or the movement can be changing direction after each PTP (the bi-directional motion execution, resulting in a "ping pong" style movement).

4.15.4 Jogging motion generator

The jogging motion generator can be used to ramp an axis to a constant velocity (jog). fig. 4-40 "The Jogging motion editor" on page 163 shows the Jogging motion editor.

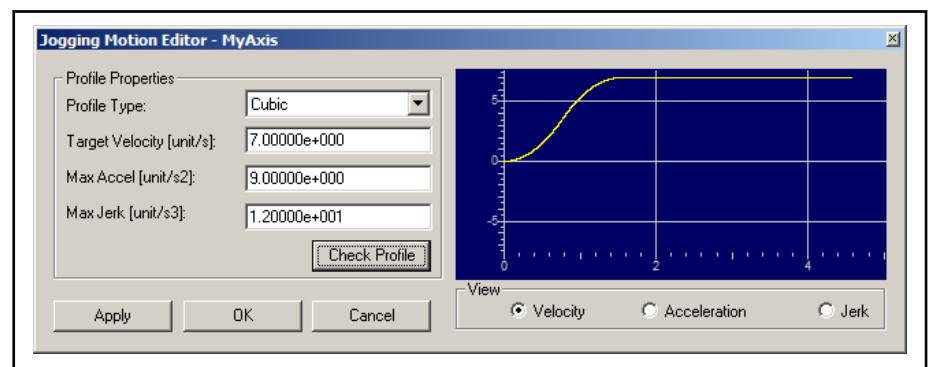


Fig. 4-40: The Jogging motion editor

By selecting "Jogging motion" in the NYCeTuner and clicking the "Edit..." button, the Jogging Motion Editor is opened. The chosen jogging motion profile can be displayed in the right hand side plot by clicking the "Check Profile" button.



Changes in the PTP and Jogging editors cannot be applied while the axis is moving.

4.15.5 Gantry tuning



The "Gantry" menu entry in the menu bar of NYCeTuner is only available when you start NYCeTuner from the command line with the "/gantrytoy" option, or the shortcut "NYCeTuner(Gantry Mode)" in the amd64/toys (for 64-bit systems) or x86/toys (for 32-bit systems) folder.

Gantry tuning is added to NYCeTuner to avoid a separate tool, but does not have the maturity level of the NYCe 4000 tools.

Prerequisites for gantry tuning

Before you can tune the gantry, you must do the following actions.

- Each gantry axis must separately be configured.
- Error handlers and peer error handlers are defined for both axes.
- The current loop of each gantry axis must be tuned independent of the other axis.
- ECG parameters must have been set correctly for both gantry axes.

NYCeTuner

Create a gantry After both axes have been configured and tuned, do the following steps to create a gantry.

1. Select the "expert mode" of NYCeTuner.
2. Right-click in the left pane of NYCeTuner while the mouse pointer is over the axis that will become the 2nd axis of the gantry.
3. Choose "Select as 2nd gantry axis" from the popup menu that appears.

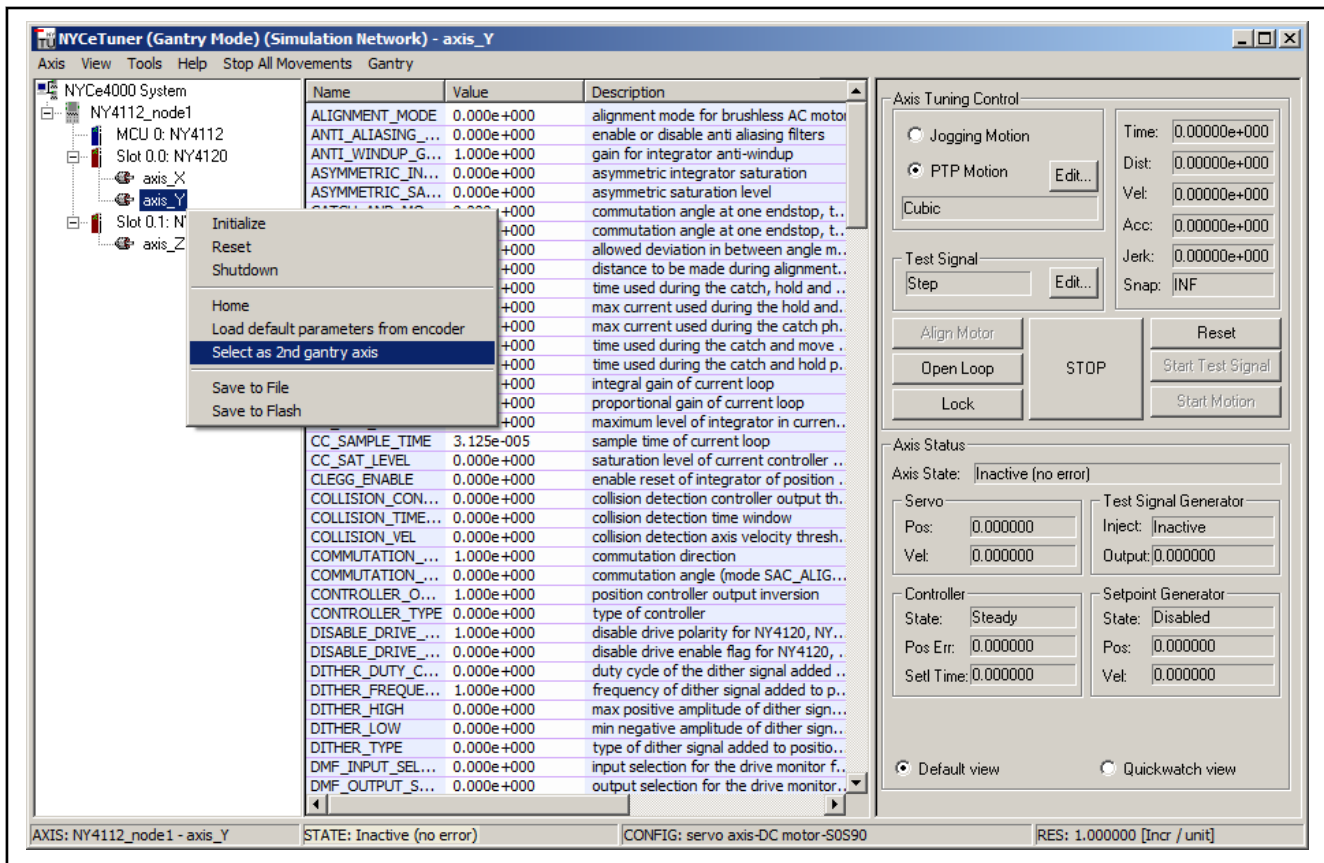


Fig. 4-41: Select the second gantry axis

The identification of the selected axis is now shown as the axis name followed by "[1st axis][2nd axis]" to indicate that this axis will become the second gantry axis (but this axis is also the first axis, because this axis is also selected).

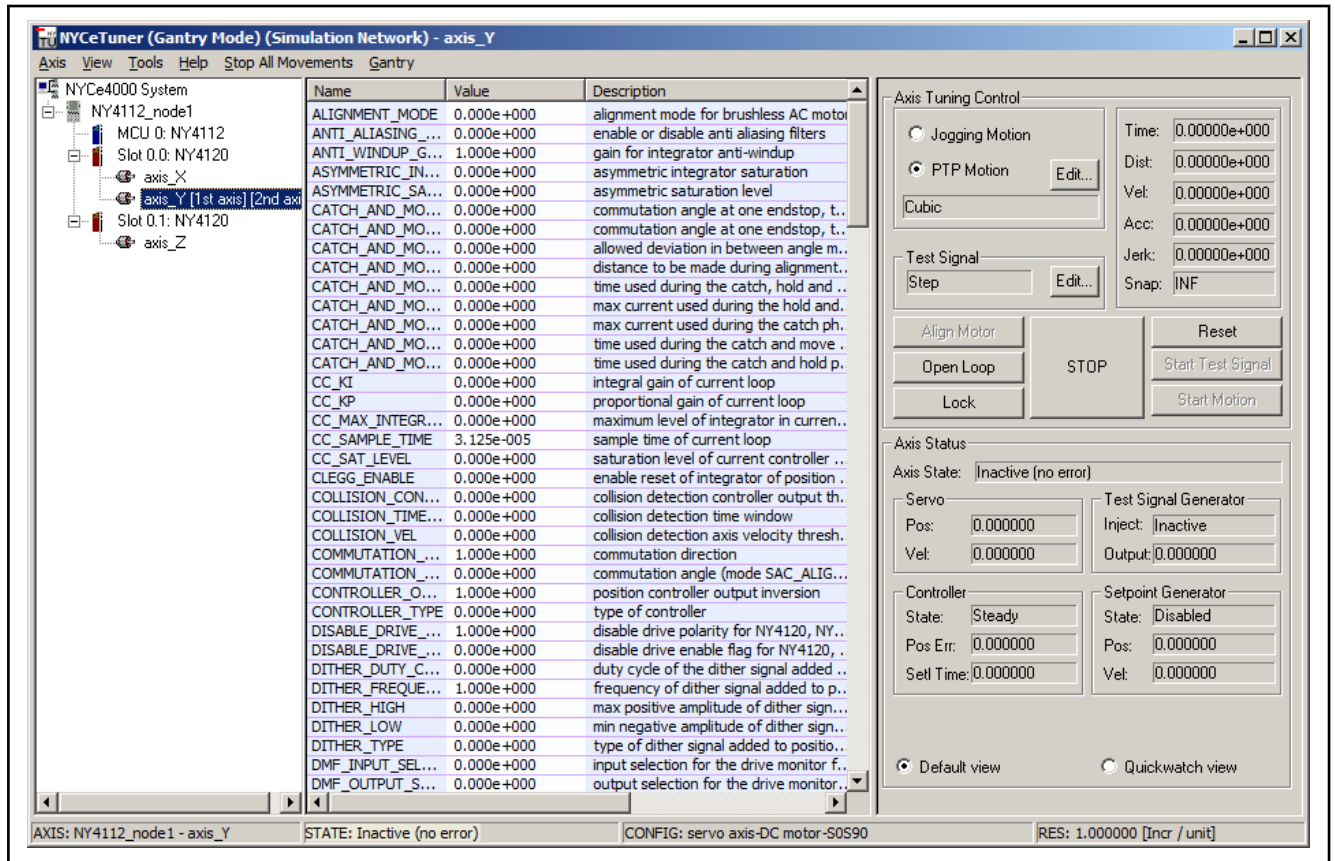


Fig. 4-42: Second gantry axis selected

4. If you select any other axis, that axis becomes the first axis of the gantry.

NYCeTuner

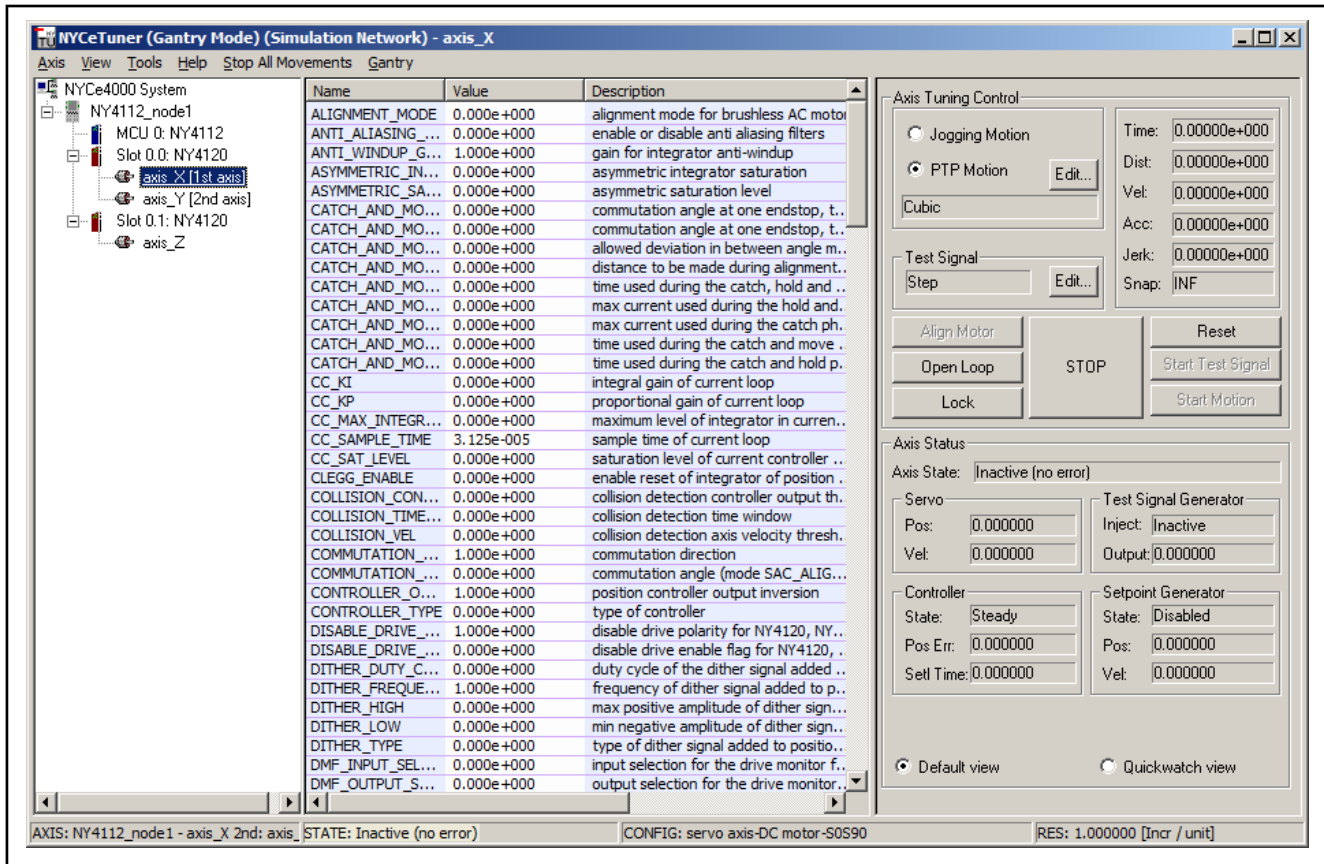


Fig. 4-43: Both axes of the gantry defined

At the left side of the status bar the gantry configuration is shown after both gantry axes are selected.

If you select any other axis, except the axis which is defined as the 2nd gantry axis, that axis becomes the 1st gantry axis. If you select the 2nd gantry axis, that axis becomes the 1st gantry axis and the axis that was the 1st axis becomes the 2nd gantry axis. Thus, the axis that is selected is always the 1st gantry axis.

Initialize, alignment and lock

You can initialize, align and lock each axis of the gantry separately.

Home procedure

After you created a gantry and start a home procedure for one of the two axes, the other axis is also homed. The home procedure homes the first axis while the second axis is geared to the first axis. Subsequently, the second axis is homed while the first axis is geared to the second axis.

Gantry movement

When a point to point or jog movement is started, the current selected axis is defined as the "master" axis and the second axis as "slave" axis with a gear ratio of 1. The master and slave axis are put in an error group.

Gantry menu options

If you change a parameter of the current selected axis, the parameter for the second axis is changed to the same modified value. You can set a parameter specifically for one axis to a different value with the option "Separate gains" in the "Gantry" menu bar entry.

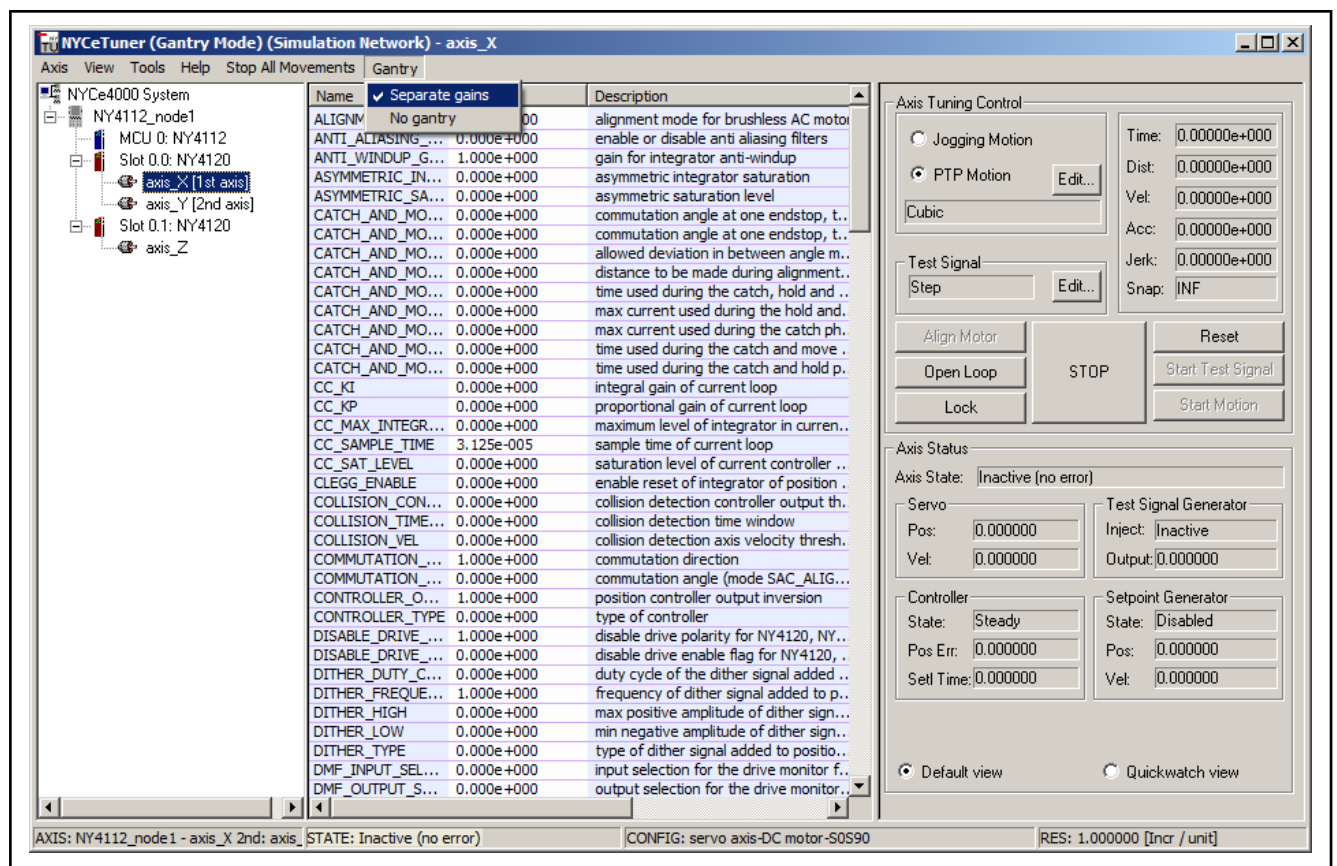


Fig. 4-44: Separate axis parameter setting



It is assumed that the positive direction is identical for both gantry axes. If the encoders count in different directions, you can correct the direction of one of the encoders with the measurement direction parameter.

Gantry termination

Select "No gantry" in the "Gantry" menu bar entry to terminate the gantry mode of NYCeTuner.

The gantry of the two axes is terminated in the following conditions.

- If an axis with a second gantry axis is active, and another axis than the second gantry axis is selected, the gantry link is broken and a popup messages appears to notify this event.
- The gantry link between the two axes is broken when NYCeTuner is terminated.
- The gantry link between the two axes is broken when the "No gantry" option in the "Gantry" menu bar is selected.

Restrictions and notes

- The gantry option is only available if the first and the second axis are configured on the same node.
- When movement commands are issued to one of the gantry axes by any other application than the NYCeTuner gantry option, the other gantry axis will not move.
- There is no axis state transition linking between the first and second gantry axis.

NYCeTuner

4.15.6 Drive Monitor Function

With the NYCeScope FRF (Frequency Response Function) you can measure the frequency response of the NYCe 4000 controller. However, the measurement with the NYCeScope has some drawbacks.

- The data collection (measurement results) is implemented with the trace functionality. The trace functionality is executed at the sample frequency of the MCU. The sample frequency is typically a factor 8 or 16 lower than the sample frequency of the PVL on the drive module. This limits the possible frequency range of the measurements, and causes aliasing in the FRF output.
- External signal generators offer more possibilities than the NYCe 4000 test signal generator. In particular, this applies to signal generators included in measurement equipment like an HP frequency analyzer.

The Drive Monitor Function uses SAC parameters, available in the "expert mode" of the NYCeTuner, to inject an external test signal to the NYCe 4000 controller via an analog input, and monitor the output of the controller (typically running at 32 kHz) via an analog output.



The Anti-Aliasing Function (AAF) and the Drive Monitor Function (DMF) can only be active at the same time if the PVL frequency is set to 16 kHz or lower. If AAF or DMF is active and the PVL frequency is set to 32 kHz the error `CTR_ERR_SIMULTANEOUS_DMF_AND_AAF_NOT_SUPPORTED` is generated when the other functionality is also activated.

Drive Monitor Function for the PVL

fig. 4-45 "Simplified DMF measurement setup using external equipment" on page 168 shows a simplified diagram of the measurement setup for the PVL that can be defined with the SAC parameters.

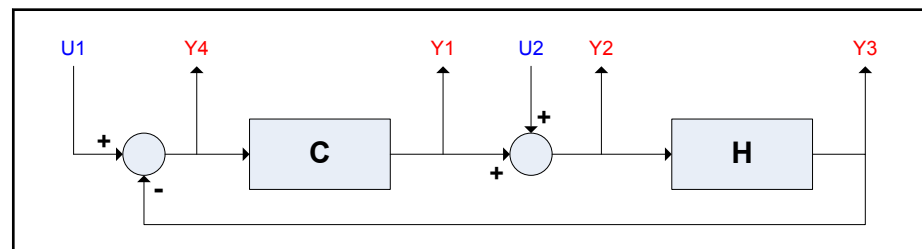


Fig. 4-45: Simplified DMF measurement setup using external equipment

The controller is indicated with the box "C", and the plant to be controlled is represented with the box "H". fig. 4-46 "DMF measurement setup of the PVL" on page 169 shows the measurement setup in more detail.

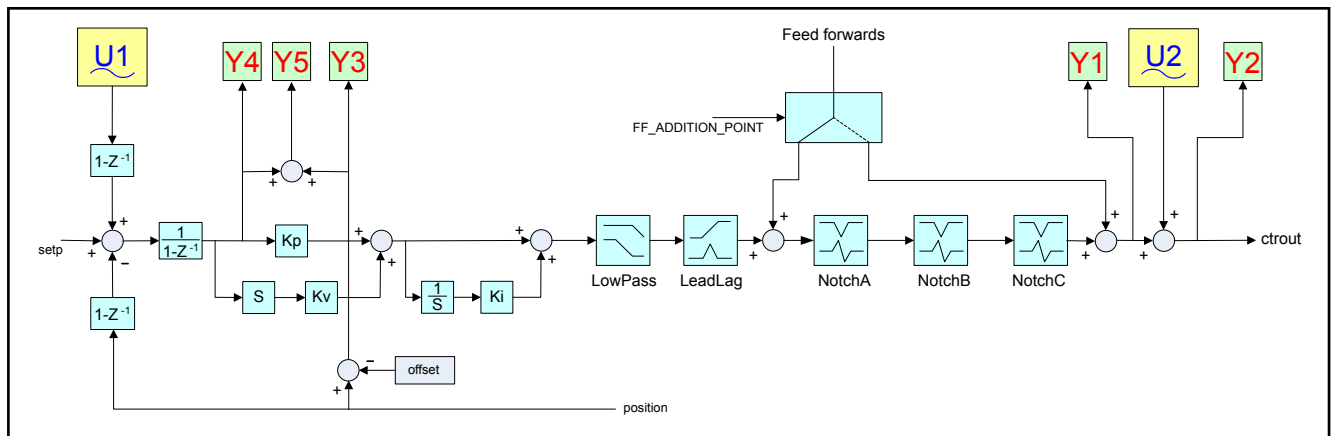


Fig. 4-46: DMF measurement setup of the PVL

The test signal injection points for the (digitized) analog input signals are indicated by U1 and U2. The output signals Y1, Y2, Y3, Y4, and Y5 are the signals (either or not with a test signal superposed) that can be monitored via the analog output. The inputs are sampled and the outputs are updated with the update frequency of the PVL (up to 32 kHz).

With the described setup the measurement of the frequency response of the transfer functions listed in [tab. 4-12 "Frequency response transfer functions" on page 169](#) is possible.

	Function name and description	Transfer function
1	"Controller sensitivity" Transfer U1 → Y1 (or U1 → Y2, as U2 is 0 when U1 is selected)	$R = \frac{C}{1 + CH}$
2	"Sensitivity" Transfer U2 → Y2 and U1 → Y4	$S = \frac{1}{1 + CH}$
3	"Closed loop" or "complementary sensitivity" Transfer U2 → (-Y1) and U1 → Y3	$T = \frac{CH}{1 + CH}$
4	"Plant sensitivity" or "process sensitivity" Transfer U2 → Y3 and U2 → (-Y4)	$Q = \frac{H}{1 + CH}$

Tab. 4-12: Frequency response transfer functions

A special case of measurement 4 is obtained by choosing the controller gains for the P-action and D-action equal to 0. This measurement gives the frequency response of the plant transfer function.

All frequency responses of the mentioned transfer functions are relevant for the evaluation of the system performance. You can derive frequency responses of other interesting transfer functions from the above mentioned

NYCeTuner

measurements. Deriving one of the frequency responses mentioned above from other measurements can be advantageous, as sometimes the measurement quality (coherence) of for example the sensitivity frequency response function can be considerably better than for example the quality of the measurement of the closed loop frequency response function.

- "Open loop" transfer function

$$CH = \frac{1 - S}{S} = \frac{T}{1 - T} \quad \text{or} \quad CH = \frac{T}{S}$$

- "Sensitivity" transfer function (similar to the result of measurement 2, but obtained by the closed loop measurement 3)

$$\frac{1}{1 + CH} = 1 - T \quad (= S)$$

- "Closed loop" transfer function (similar to the result of measurement 3, but obtained by the sensitivity measurement 2)

$$\frac{CH}{1 + CH} = 1 - S \quad (= T)$$

- "Plant sensitivity" transfer function (similar to the result of measurement 4)

$$\frac{H}{1 + CH} = \frac{ST}{R} \quad (= Q)$$

- "Plant transfer" function (similar to the result of measurement 4 with the controller gains set to 0).

$$H = \frac{T}{R} = \frac{Q}{S}$$

- "Controller transfer" function (mainly of interest for validation of the controller)

$$C = \frac{R}{S} = \frac{T}{Q}$$



- The signal Y1, Y2, Y3, Y4, or Y5 can be output via the analog output, but only possible if the analog output of that axis is not in use for other purposes ("external drive" or monitor).
- Axes with motor type "external drive" always have Y2 as output to the analog output. This output cannot be scaled, but measurements using Y2 as output are possible to some extent.

Constraints

- The analog inputs and analog outputs are all filtered with a second order low pass filter with a break frequency at approximately 4 kHz. The setup described involves these filters in the measurements 1, 2 and 3. The derived frequency responses T/R and Q/S (of the plant transfer function, H), T/S (of the open loop transfer function, CH) and R/S (of the controller transfer function C) do not contain the filters.
- The analog inputs are converted to digital values with limited resolution.
- Filtering of the analog input suppresses aliasing frequencies, but still some aliasing effect may affect the measurement, due to 32 kHz sampling.

- If the (scaled) output exceeds the range of the DAC, the output signal is clipped to the maximum value (+10 Volt) and minimum value (-10 Volt).
- The measurement circuit may contain sample & hold circuits, delays and different sampling frequencies, which may affect the measurements. In particular, the analog inputs are sampled with the PVL frequency, and the analog outputs are updated with that same frequency. Between the input and output is a delay of at least one PVL sample and the DAC conversion time.
- Analog output 0 is physically connected to axis unit 0, and analog output 1 is physically connected to axis unit 1. This configuration is fixed and cannot be changed.

Most of the mentioned effects can be measured by performing a sensitivity measurement (measurement 2) with feedback P-gain (K_p) and D-gain (K_v) in the controller set to 0. The resulting frequency response can be subtracted from other measurements 1, 2, 3 and 4 as a compensation for the effects mentioned.

SAC parameters that define the measurement setup of the PVL

The SAC parameters `DMF_INPUT_SELECTION` and `DMF_OUTPUT_SELECTION` define the measurement setup, see [fig. 4-46 "DMF measurement setup of the PVL" on page 169](#).

The SAC parameter `DMF_INPUT_SELECTION` selects the input signal point, U1 or U2. If this parameter is set to the value

- `SAC_DMF_INPUT_NONE`, the injection of test signals in the control loop is disabled (default setting).
- `SAC_DMF_INPUT_POSITION`, the analog input is added as U1 in the control loop.
- `SAC_DMF_INPUT_CONTROLLER`, the analog input is added as U2 in the control loop.

See [tab. 4-13 "DMF input selection parameters and input signal injection point" on page 171](#) for an overview of the input signal injection possibilities.

The digitized input signal is scaled after the ADC before the signal is applied to the loop with the SAC parameter `DMF_VOLT_TO_INPUT`.

With the scaling set to 1,

- the position input U1 is converted to position units, where 1 V input represents 1 position unit (pu). The scaling unit is pu/V.
- the controller input U2 is converted to controller output, where 1 V input represents 1 A. The scaling unit is A/V.

DMF_INPUT_SELECTION	input
DMF_INPUT_NONE	-
DMF_INPUT_POSITION	U1
DMF_INPUT_CONTROLLER	U2

Tab. 4-13: DMF input selection parameters and input signal injection point

The SAC parameter `DMF_OUTPUT_SELECTION` selects the output signal point, Y1 or Y2 or Y3 or Y4 or Y5 for PVL measurements. If this parameter is set to the value

NYCeTuner

- SAC_DMF_OUTPUT_NONE, writing to the DAC output is disabled (default setting).
- SAC_DMF_OUTPUT_CONTROLLER, the value of the controller output Y1 is put on the DAC output.
- SAC_DMF_OUTPUT_CONTR_PLUS_TEST, the value of the controller output, augmented with the test signal value U2, that is Y2, is put on the DAC output.
- SAC_DMF_OUTPUT_POSITION, the position value, valid at that moment, is retrieved to be subtracted as an offset from the (future) Y3 output. The difference, Y3 minus offset, is put on the DAC output.
- SAC_DMF_OUTPUT_POSITION_ERROR, the value of the position error, that is Y4, is put on the DAC output.

See [tab. 4-14 "DMF output selection parameters and output connection points for the PVL"](#) on page 172 for an overview of the output signal possibilities.

The digital output signal is scaled before the signal is converted to an analog signal with the parameter DMF_OUTPUT_TO_VOLT.

With the scaling set to 1,

- the controller outputs Y1 and Y2 are converted to values between -10 V and +10 V. On NY4120 and NY4140 drive modules, a 1 V output represents 1 A. The scaling unit is V/A.
- the position outputs Y3, Y4, and Y5 are converted to values between -10 V and +10 V, where 1 position unit (pu) represents 1 V. The scaling unit is V/pu.

DMF_OUTPUT_SELECTION	Output
DMF_OUTPUT_NONE	-
DMF_OUTPUT_CONTROLLER	Y1
DMF_OUTPUT_CONTR_PLUS_TEST	Y2
DMF_OUTPUT_POSITION	Y3
DMF_OUTPUT_POSITION_ERROR	Y4
DMF_OUTPUT_SPG_POSITION	Y5

Tab. 4-14: DMF output selection parameters and output connection points for the PVL

Drive Monitor Function for the CCL

Output connection points Y6 thru Y18 are available in the CCL for monitoring, especially for stepper motor usage, see [fig. 4-47 "DMF measurement setup of the CCL"](#) on page 173.

These output connection points can only be selected on the NY4120 and NY4140 drive modules.

NYCeTuner

DMF_OUTPUT_SELECTION	Output	description
DMF_OUTPUT_UQ	Y17	output of CCL Uq before inverse park transformation
DMF_OUTPUT_ANGLE	Y18	angle

Tab. 4-15: DMF output selection parameters and output connection points for the CCL

The digital output signal is scaled before the signal is converted to an analog signal with the parameter `DMF_OUTPUT_TO_VOLT`.

With the scaling set to 1,

- the motor currents Y6 thru Y9 and the current errors Y10 and Y11 are converted to values between -10 V and +10 V. On NY4120 and NY4140 drive modules, a 1 V output represents 1 A. The scaling unit is V/A.
- the integrator values Y12 and Y13 and CCL outputs Y14 thru Y17 are converted to values between -10 V and +10 V, where 1 V represents 1 V servo voltage. The scaling unit is V/V_{drive} .
- the angle value Y18 is converted to values between -10 V and +10 V, where 3,14 V represents 1 radian. The scaling unit is 3,14V/rad.

PVL and CCL DMF timing

The PVL on the drive module controls the DAC of the analog output. For that reason the timing of the DMF depends on the PVL sample frequency.

See [fig. 4-48 "PVL and CCL DMF timing" on page 174](#) for the timing sequence from the ADC latch moment to the DAC output moment.

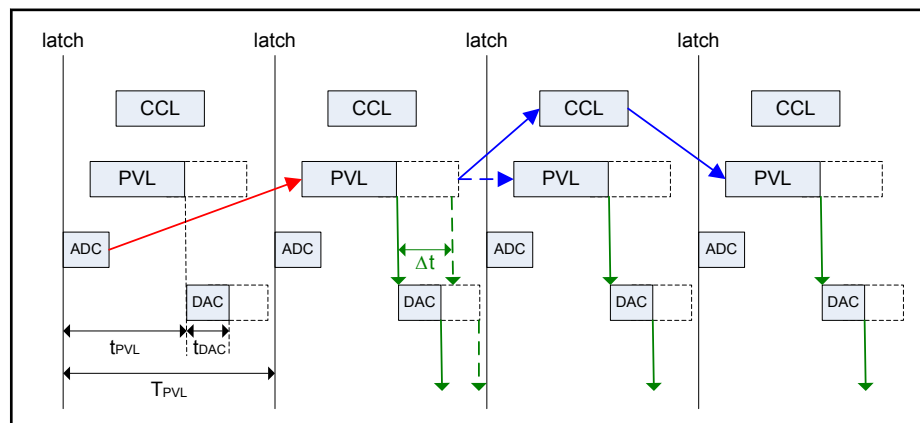


Fig. 4-48: PVL and CCL DMF timing

For PVL DMF measurements, the timing between the ADC latch moment and DAC output moment is $T_{PVL} + t_{PVL} + t_{DAC}$. This timing applies for the measurements from $U1 \rightarrow Y1$, $U1 \rightarrow Y2$, or $U2 \rightarrow Y2$.

For the measurement $U2 \rightarrow Y1$ the maximum timing delay is $2 * T_{PVL} + t_{PVL} + t_{DAC}$. The Δt shown in [fig. 4-48 "PVL and CCL DMF timing" on page 174](#) (in green) represents the variation in t_{PVL} which depends on the operation mode of the PVL controller. If the PVL is open loop, the overall delay from ADC \rightarrow DAC is shorter than when the PVL is closed loop.

For CCL DMF measurements, the timing delay is fixed $3 * T_{PVL} + t_{PVL} + t_{DAC}$.

4.16 Axis tuning in "expert" mode

4.16.1 Introduction

In the chapter "chapter 2.6 "Quick start" on page 43" the steps are described how you can get an axis operational with the NYCeTuner in "wizard" mode. You can also get an axis operational in the "expert" mode, and fine-tune the axis for better performance. This chapter describes the following procedures in the "expert" mode.

- Current loop tuning
- Setting the alignment parameters for a BLAC motor
- Get the commutation order for a BLDC motor
- Determining the measurement direction
- Position loop tuning
- Stepper motor tuning

4.16.2 Current loop tuning

Start the NYCeTuner, select "View → Expert" to enter the "expert" mode if the NYCeTuner started in the "wizard" mode, and select the "MyAxis" axis via the "Axis" menu. Choose to initialize the axis from the file MyAxis.xml. In the "View" menu, select "Current Control Parameters". The parameter list in the pane in the middle now only displays the current loop parameters, see fig. 4-49 "NYCeTuner in "expert" mode, "Current Control Parameters" view" on page 175.

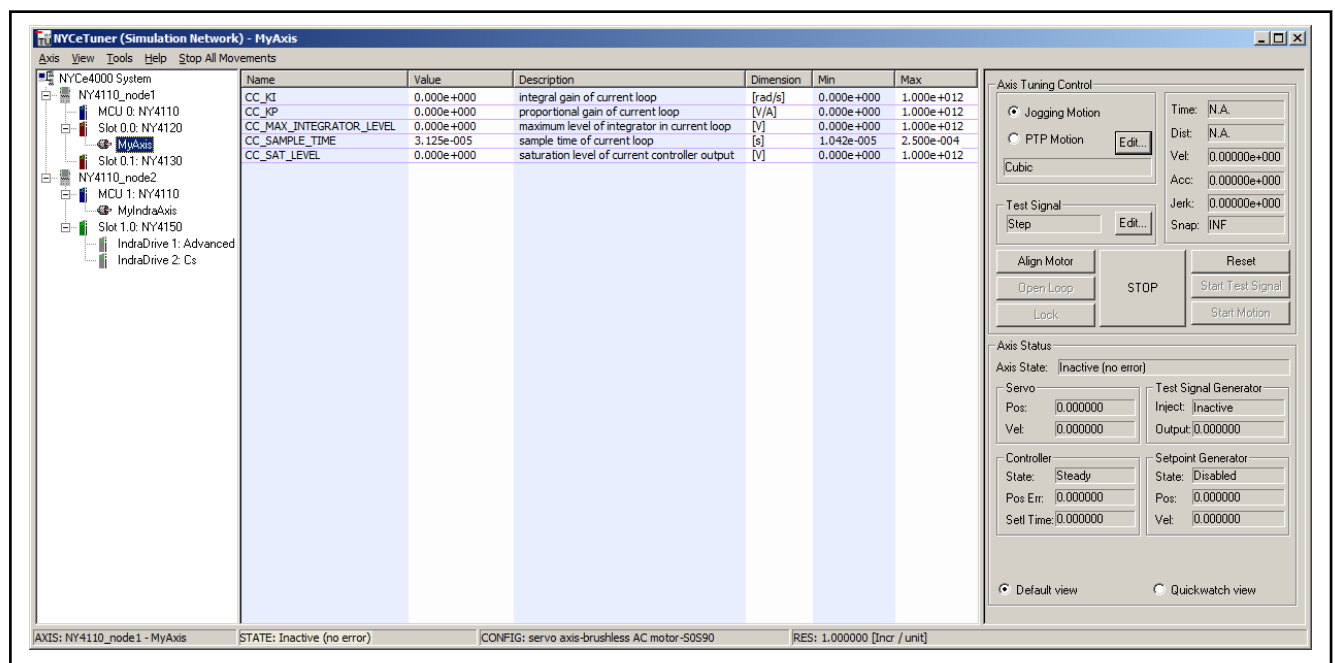


Fig. 4-49: NYCeTuner in "expert" mode, "Current Control Parameters" view

Consult the motor manufacturer datasheet to get the induction and resistance (phase to phase, expressed in H and Ω respectively). The electrical time constant T_e equals L / R .

Apply these initial settings to the following parameters.

1. Set the saturation level (CC_SAT_LEVEL) and the maximum integrator level of the current controller equal to the drive voltage.

NYCeTuner

2. Set the proportional gain (CC_KP) of the current loop equal to $3 * R$, with R the winding resistance of the motor.
3. Set the integral gain (CC_KI) of the current loop equal to $1.5 / T_e = 1.5 * R / L$.
4. Set the PWM frequency $> 10 * 1 / T_e$. This setting prevents noisy behavior and needless energy dissipation in the drive module.

When you use a BLAC motor, the motor must be aligned before any power can be applied to the motor. Although the axis does not need to be aligned for the current loop tuning, the NYCe 4000 system has a built-in check for alignment, before allowing any power on the motor. The current loop fine-tuning procedure described is failsafe, so the BLAC motor does not have to be aligned properly to get the tuning correct.

The alignment procedure below does not correctly align the BLAC motor, but is good enough to fine-tune the current controller. The correct alignment procedure is described in the paragraph "Setting the alignment parameters (BLAC only)".

1. Select the Alignment parameters via the "View" menu.
2. Set the alignment mode to 3 (set commutation offset).
3. Click the "Align Motor" button.
4. Return to the standard control parameters (via the "View" menu).

Fine-tune procedure

1. Click the "Edit..." button in the "Test Signal" section in NYCeTuner. Click the "Prop" button (properties) of the test signal in the window that appears.
 - Select the block wave signal type.
 - Set the amplitude equal to 10% of the maximum continuous motor current.
 - Leave the offset 0.
 - Set the frequency to $\text{sample_frequency} / (2 * n)$, where $n = 1, 2, 3...$
 In general, 200 Hz is a good choice, except when the sample frequency is 1 kHz. If the sample frequency of the node is 1 kHz, 125 Hz is a good choice.
 - Set the duration equal to 1000 s.
 - Choose the injection point at B, controller out open loop.
2. Start the NYCeScope tool, click the "Channels" button (see [fig. 4-50 "NYCeScope - selecting channels" on page 177](#)) and select the channels "Controller Out" and "Drive Current" of the MyAxis axis (see [fig. 4-51 "NYCeScope - selecting variables" on page 177](#)). Then click the "Close" button.

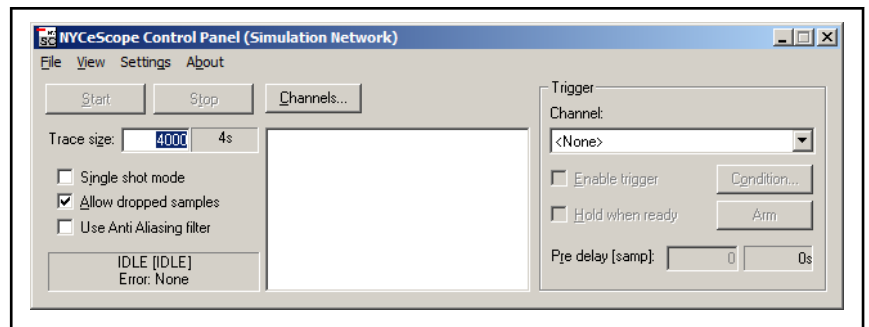


Fig. 4-50: NYCeScope - selecting channels

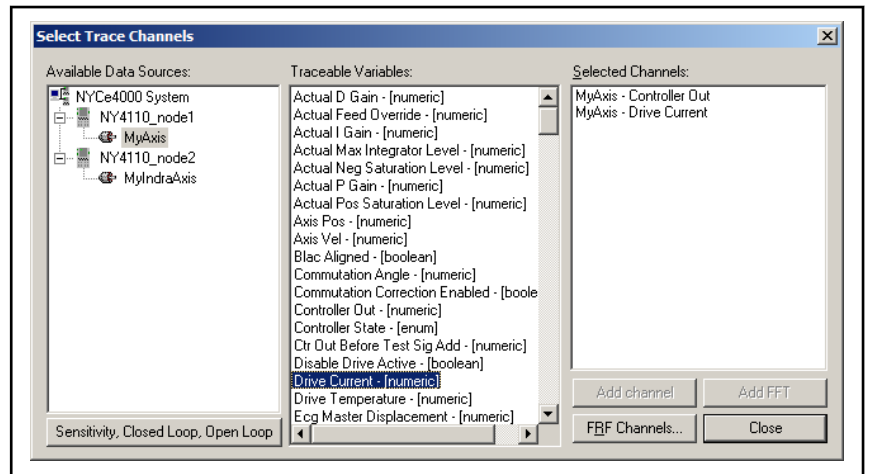


Fig. 4-51: NYCeScope - selecting variables

3. Set the Trace size to a small value, for example 40 samples in the Control panel of NYCeScope.
4. Set the trigger of the NYCeScope to controller output crossing 0 and enable the trigger.

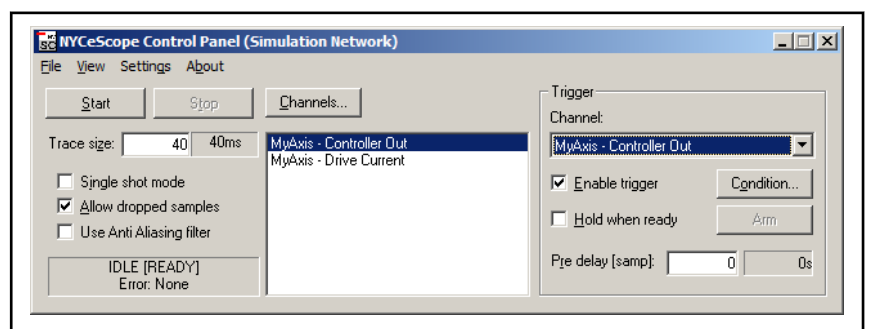


Fig. 4-52: Set the trace trigger condition and size in NYCeScope Control panel

5. Click the "Start" button of the NYCeScope to start the measurement and adjust the axis to the amplitude level of the block signal.
6. Click the "Start Test Signal" button of the NYCeTuner.
7. The NYCeScope shows a plot like [fig. 4-53 "Current loop tuning procedure - initial view" on page 178](#) after both channels are scaled within viewing range, by using the "pos" and "scale" buttons of each channel and the "Full time" button on the plot window. The yellow curve represents controller out, the green curve represents drive current.

NYCeTuner

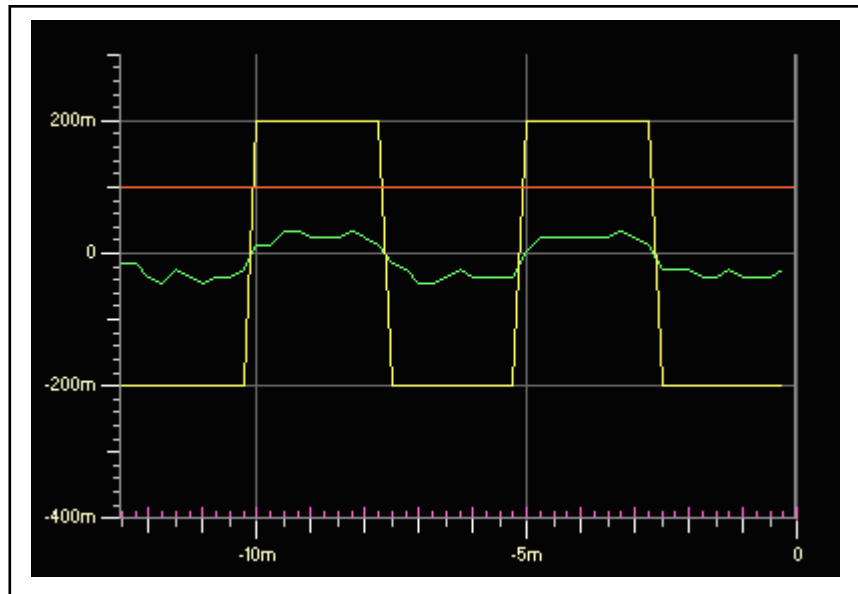


Fig. 4-53: Current loop tuning procedure - initial view



A delay can be present between the Controller Out plot and the Drive Current plot if the PVL runs on the MCU. This delay depends on the node frequency and the CCL frequency; (there is no delay if the PVL runs on the drive module).

If the CCL frequency is equal to the node frequency, the delay is 2 MCU samples.

If the CCL frequency is twice the node frequency, the delay is 1 MCU sample.

If the CCL frequency is at least four times the node frequency, there is no delay.

8. Increase the `CC_KP` parameter in the NYCeTuner until either the drive current response is satisfactory or the drive current response does not improve any more (see [fig. 4-54 "Current loop tuning procedure - adjusting KP" on page 179](#) and [fig. 4-55 "Current loop tuning procedure - adjusting KP more" on page 179](#)). The yellow curve represents controller out, the green curve represents drive current.

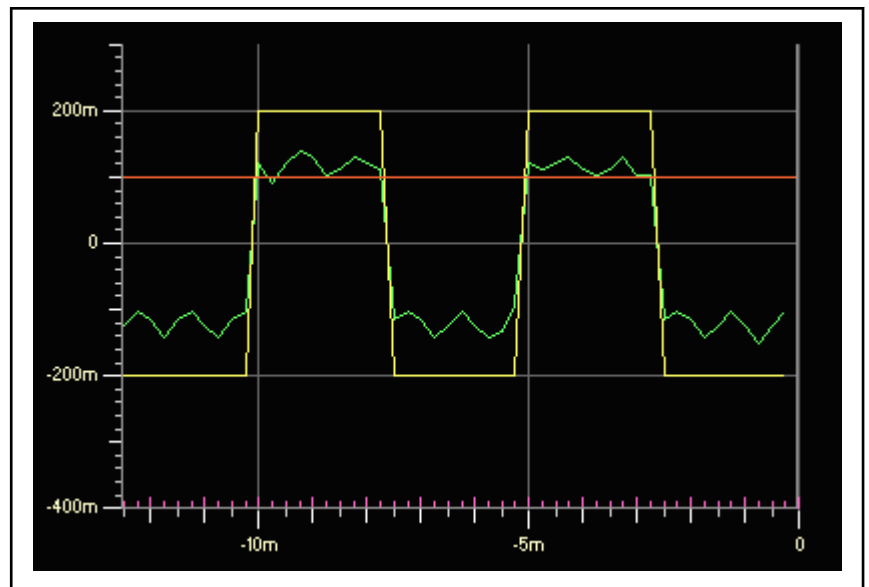


Fig. 4-54: Current loop tuning procedure - adjusting K_P

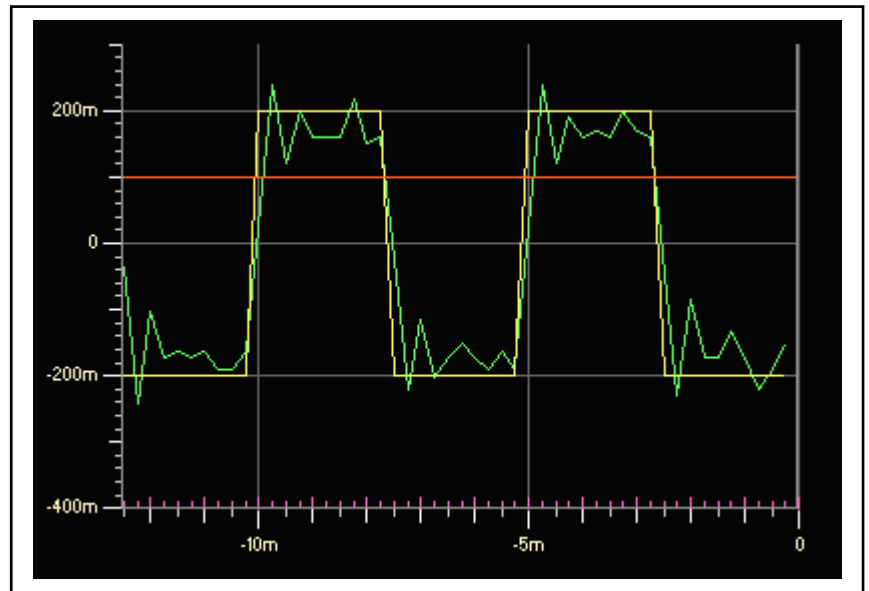


Fig. 4-55: Current loop tuning procedure - adjusting K_P more

9. Increase the CC_KI parameter until the drive current response is satisfactory.



Be careful while scaling the axes of the NYCeScope. Each channel must be scaled independently. The ruler of the y-axis of a figure only applies to one of the channels that are displayed by the NYCeScope time plot.

4.16.3 Setting the alignment parameters (BLAC only)

1. Select the Alignment parameters of the "View" menu of the NYCeTuner.
2. Check if the alignment mode is set to 1 (kick & swing method).

NYCeTuner

3. Set maximum controller output during alignment to 50% (up to 80%) of the maximum continuous current of the motor.
4. Click the "Align Motor" button in the pane at the right hand side. Note that the motor will move up to ½ pole pair.

The following steps check that the alignment parameters are set correctly. A block wave test signal is added to the motor current. Then it is manually checked that the force on the motor stays constant during a movement.

1. Select the test signal properties and choose the following settings.
 - Select the block wave test signal type.
 - Set the amplitude equal to 10% of the maximum continuous motor current.
 - Leave the offset 0.
 - Set the frequency to $\text{sample_frequency} / (2 * n)$, where $n = 1, 2, 3...$
In general, 200 Hz is a good choice, except when the sample frequency is 1 kHz. If the sample frequency of the node is 1 kHz, 125 Hz is a good choice.
 - Set the duration equal to 1000 s.
 - Choose the injection point at B, controller out open loop.

2. Click the "Open Loop" button, and start the test signal. Check that you feel some resistance from the motor, when you try to rotate the axis manually.

If you do not feel any resistance, you can increase the amplitude of the block wave test signal to 20% of the maximum continuous motor current. Do not increase the amplitude more than 20% of the maximum continuous motor current. Return the amplitude of the block wave test signal to 10% and check the wiring for possible hardware problems.

3. Slowly move the motor shaft manually at least one rotation (or some pole pairs), and check if the motor has preference for certain axis positions.
4. Click the "Disable Power" button after you rotated the motor shaft one turn.

If you can feel that the motor shaft has preference for certain axis positions, you must change the commutation direction.

- Set the parameter `COMMUTATION_DIRECTION` to -1.
Align the motor again and check again if the motor has preference for certain axis positions while you rotate the motor shaft manually. If the preference for certain axis positions is still present, then the number of pole pairs is set wrong during the configuration of the axis. Make sure that the motor specifications and NYCeConfigurator settings (on the "Commutation" tab) are the same. If the number of pole pairs was wrong and the problem still appears, check the `COMMUTATION_DIRECTION` again and finally check the motor cabling.



The alignment functionality is also available and valid for axes with the motor type "Coil Drive" (Linear Motion Systems).

4.16.4 Commutation order (BLDC only)

To check if the BLDC motor is correctly connected, go to the NYCeTuner window.

1. Select the test signal properties and choose the following settings.
 - Select the block wave signal type.
 - Set the amplitude equal to 10% of the maximum continuous motor current.
 - Leave the offset 0.
 - Set the frequency to $\text{sample_frequency} / (2 * n)$, where $n = 1, 2, 3...$

In general, 200 Hz is a good choice, except when the sample frequency is 1 kHz. If the sample frequency of the node is 1 kHz, 125 Hz is a good choice.

- Set the duration equal to 1000 s.
 - Choose the injection point at B, controller out open loop.
2. Click the "Open Loop" button, and start the test signal. Check that you feel some resistance from the motor, when you try to rotate the axis manually.

If you do not feel any resistance, you can increase the amplitude of the block wave test signal to 20% of the maximum continuous motor current. Do not increase the amplitude more than 20% of the maximum continuous motor current. Return the amplitude of the block wave test signal to 10% and check the wiring for possible hardware problems.

3. Slowly move the motor shaft manually at least one rotation (or some pole pairs), and check if the motor has preference for certain axis positions.
4. Click the "Disable Power" button after you rotated the motor shaft one turn.

If you can feel that the motor shaft has preference for certain axis positions, you must change the commutation table.

1. Go to the NYCeConfigurator, and select the "Commutation" tab. Click the "Up" button, then the "Apply" button and save the new commutation table via the "Save / Download" tab. Return to the NYCeTuner and repeat the process until the BLDC motor has no preference positions anymore.
2. If the described procedure was unsuccessfully for six times, verify that the motor specifications and NYCeConfigurator settings are the same.

4.16.5 Measurement direction

CAUTION

During this procedure the motor shaft will rotate for a short period of 0.5 seconds.

Keep clear from the motor shaft.

Start the NYCeTuner. Click the test signal properties button and select a step signal specifying

- an offset of 10% of the maximum continuous current of the motor,
- a duration of 0.5 seconds,

NYCeTuner

- set the injection point at B, open loop.

Click the "Open Loop" button (in the pane at the right hand side), then click the "Start Test Signal" button.

- If the servo position increases while the motor is powered, the measurement direction of the motor is OK.
- If the motor position decreases, select the standard control parameters in the "View" menu and invert the measuring system direction (if the measurement direction is equal to 1, change to -1 or vice versa).
- If the motor remains at the same position, increase the test signal offset to 20% of the maximum continuous current of the motor, and repeat process current. Abort this procedure if the step test signal has an offset of 20% of the continuous motor current, and the motor does not rotate. Set the offset of the step signal back to 10% and check all wiring for possible hardware problems.

4.16.6 The position loop tuning

Before you start tuning the position loop, set the following error bounds parameters.

- Set the `MAX_STEADY_STATE_ERROR` to a safe value.
 - Set the `MAX_DYN_POS_ERROR` to a safe value.
 - Set the `SAT_LEVEL` to a safe value, for example 1.
1. Go to the NYCeTuner window, and select the Standard Control parameters in the "View" menu.
 2. Set the following parameters.
 - `CONTROLLER_TYPE = 0`
 - `VELOCITY_ESTIMATION_METHOD = 1`
 - `NR_OF_SAMPLES = 16`
 3. Start the position controller by setting the proportional gain (K_P) to 1.



When setting the proportional gain to 1, the controller out will be 1A if the position error is equal to 1 revolution. If this value for the proportional gains seems to be unfit for the axis, choose a more appropriate value for the axis.

4. Select the PTP motion in the pane at the right hand side of the NYCeTuner, and click the properties button.
5. Choose the energy optimized motion profile
 - Set the time equal to 2 seconds
 - Set the distance equal to 1 unit
 - Set the motion direction to bi-directional
 - Select auto move with continuous repeat
6. Open the NYCeScope and select the MyAxis channels
 - Axis position
 - Setpoint position
 - Controller Out
 - Position error

- Eventually, the FFT of the position error (in the NYCeScope, select the position error in the selected channels list and click the "Add FFT" button).
7. At the time plot, disable the position error plot and click on the "Ruler" button of the set-point position. Click the "Start" button in the NYCeScope and the "Start Motion" button in the NYCeTuner. [fig. 4-56 "Tuning the proportional gain of the position loop" on page 183](#) shows an example of the axis position and setpoint position plots of the motor.

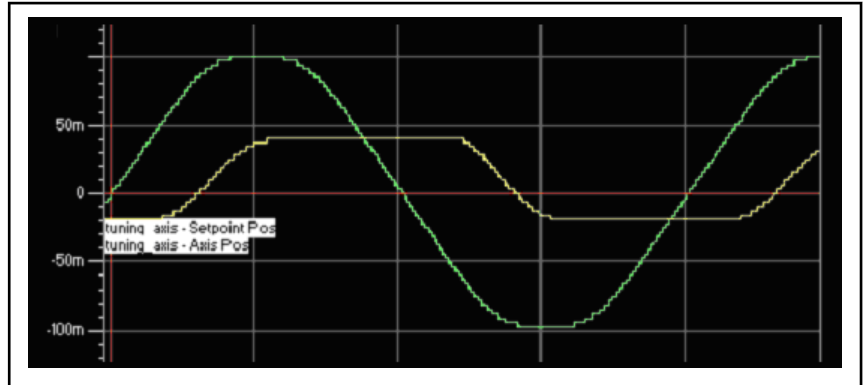


Fig. 4-56: *Tuning the proportional gain of the position loop*

8. Increase the proportional gain of the position loop K_P until the axis position plot shows an oscillation (see [fig. 4-57 "Tuning the proportional gain of the position loop - increase \$K_P\$ " on page 183](#) and [fig. 4-58 "Tuning the proportional gain of the position loop - increase \$K_P\$ more" on page 184](#)). Count the number of oscillations (per second) in the position error plot (or use the FFT plot of the position error to determine the frequency of this oscillation).

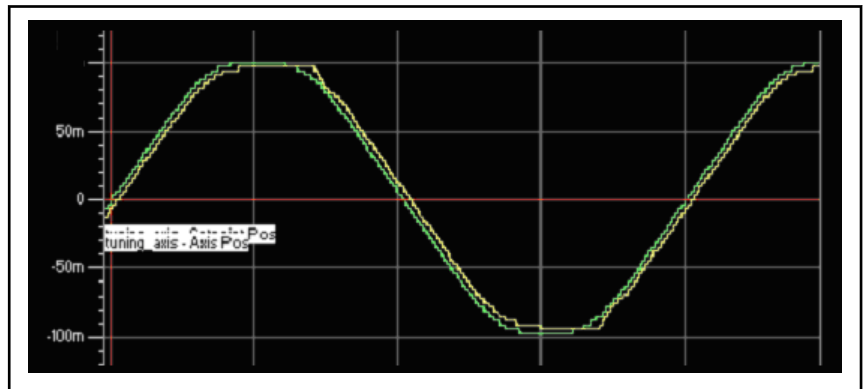


Fig. 4-57: *Tuning the proportional gain of the position loop - increase K_P*

NYCeTuner

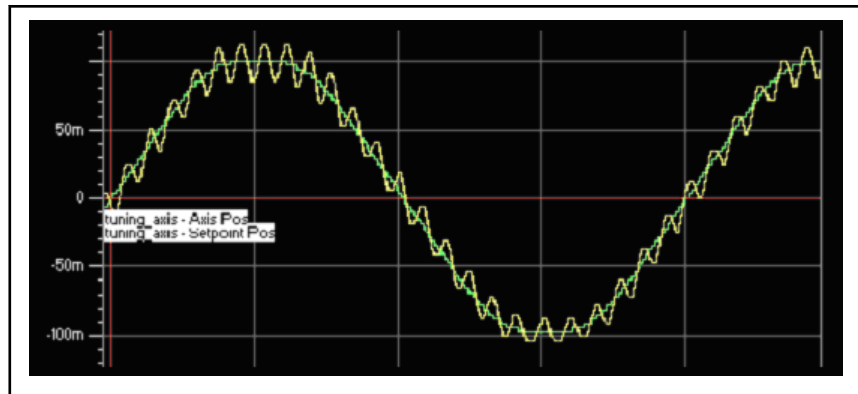


Fig. 4-58: *Tuning the proportional gain of the position loop - increase K_P more*

9. Now reduce the proportional gain of the position loop K_P with 10%. Then calculate K_V from the found oscillation frequency and the K_P according to the following formula

$$K_V = K_P / (2 * \pi * \text{oscillation frequency}).$$
10. Perform a movement to see the result. If the oscillation is still present, increase K_V a small amount, for example 10%, and look at the result.
11. Then, to decrease the position error, increase K_P and K_V at the same time with the same percentage, for example 10% (to keep the same relation between K_p and K_v) until an acceptable position error without oscillations is achieved.



This tuning procedure does not deliver an optimal controller. It is just a procedure to get an idea about the way the NYCe 4000 tools operate.

4.16.7 Stepper motor tuning

Open the NYCeTuner, select "View → Expert" to enter the "expert" mode if the NYCeTuner started in the "wizard" mode, and select the stepper motor via the axis menu. Initialize the motor from the MyAxis.xml file created by the NYCeConfigurator. In the "View" menu, select "Current Control Parameters."

Consult the motor manufacturer datasheet to get the inductance and resistance (phase to phase, expressed in H and Ω respectively). The electrical time constant T_e equals L / R .

Apply these initial settings to the following parameters.

1. Set the saturation level (CC_SAT_LEVEL) and the maximum integrator level of the current controller equal to the drive voltage.
2. Set the proportional gain (CC_K_P) of the current loop equal to $3 * R$, with R the winding resistance of the motor.
3. Set the integral gain (CC_K_I) of the current loop equal to $1.5 / T_e = 1.5 * R / L$.
4. Set the PWM frequency $> 10 * 1 / T_e$.

This setting prevents noisy behavior and needless energy dissipation in the drive module.

Fine-tune procedure

1. Click the "Edit..." button in the "Test Signal" section in NYCeTuner. Click the "Prop" button (properties) of the test signal in the window that appears.
 - Select the block wave signal type.
 - Set the amplitude equal to 10% of the rated current of the stepper motor.
 - Leave the offset 0.
 - Set the frequency to $\text{sample_frequency} / (2 * n)$, where $n = 1, 2, 3...$

In general, 200 Hz is a good choice, except when the sample frequency is 1 kHz. If the sample frequency of the node is 1 kHz, 125 Hz is a good choice.
 - Set the duration equal to 1000 s.
 - Choose the injection point at B, controller out open loop.
2. Open the NYCeScope, click the button "Channels" and select the channels Controller out and quiescent current of the MyAxis axis, see [fig. 4-50 "NYCeScope - selecting channels" on page 177](#) and [fig. 4-51 "NYCeScope - selecting variables" on page 177](#).



In the current NYCe 4000 implementation, the terms "drive current" (Id) and "quiescent current" (Iq) are used. These confusing mnemonics will be corrected in a future release of NYCe 4000. Id will be renamed to I torque and Iq will be renamed to I flux.

3. Set the Trace size to a small value, for example 40 samples in the Control panel of NYCeScope.
4. Set the trigger of the NYCeScope to controller output crossing 0 (enable the trigger while Controller out is selected).
5. Click the "Start" button of the NYCeScope to start the measurement and adjust the axis to the amplitude level of the block signal.
6. Click the "Start Test Signal" button of the NYCeTuner.
7. The NYCeScope shows a plot like [fig. 4-53 "Current loop tuning procedure - initial view" on page 178](#) (after both channels are scaled within viewing range, by using the "pos" and "scale" buttons of each channel and the "Full time" button on the plot window).
8. Increase the `CC_KP` parameter in the NYCeTuner until either the drive current response is satisfactory or the drive current response does not improve any more, see [fig. 4-54 "Current loop tuning procedure - adjusting KP" on page 179](#) and [fig. 4-55 "Current loop tuning procedure - adjusting KP more" on page 179](#). The yellow curve represents controller out, the green curve represents drive current.
9. Increase the `CC_KI` parameter until the drive current response is satisfactory.

Select the stepper control parameters in the NYCeTuner. The parameter list is reduced to 9 parameters.

The stepper motor can be driven with the following parameters (as good starting values).

- Set the acceleration and deceleration current equal to the rated current to obtain the best response.

NYCeTuner

- Set the constant velocity and hold current equal to 50% of the rated current to minimize heat generation.
- Set all delay times (Pre Acceleration / Post acceleration / Pre deceleration / Post deceleration) to 10 ms.
- Set the delay compensation time to 0.
- Set the base velocity set higher than the velocity corresponding to the mechanical resonance frequency of the stepper motor.
- Use only parabolic profiles with stepper motors.

After defining the movement (second order profile, so no jerk) and clicking the "Lock" button, the stepper motor can be moved.

4.17 Miscellaneous

When you click the icon at the left in the title bar, the popup with the standard Windows functions (for example "Move") appears. The last entry of the popup ("About NYCeTuner") displays the "About NYCeTuner" dialog window. This dialog window is also displayed when you select "Help → About NYCeTuner" from the menu bar.

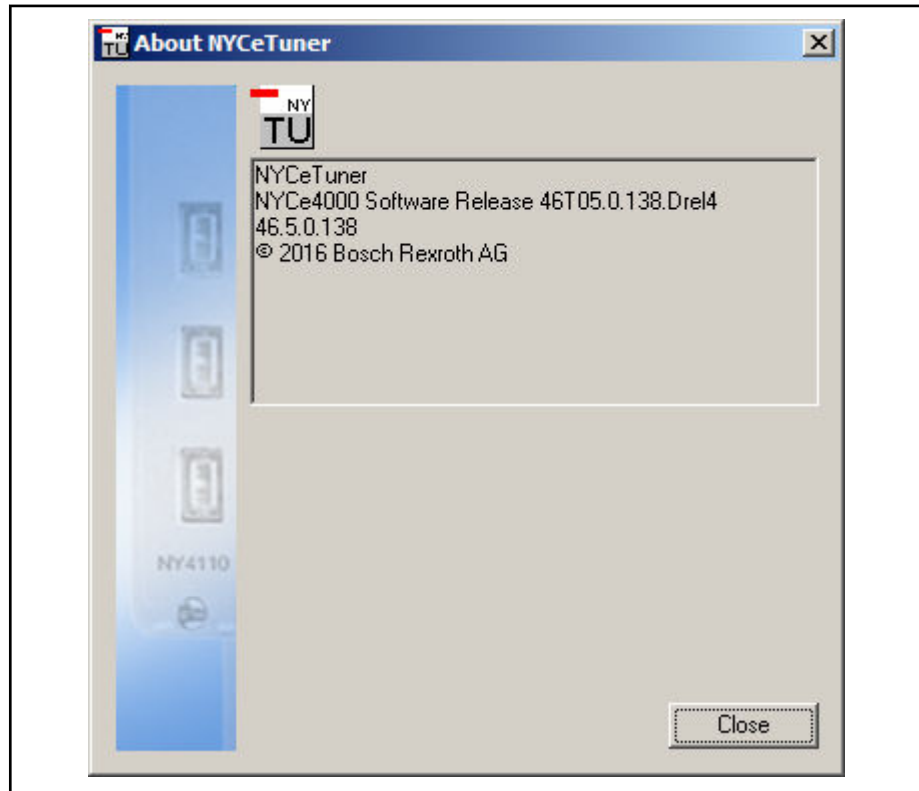


Fig. 4-59: The "About NYCeTuner" dialog window

See [chapter 1.3 "NYCe 4000 product and file version information"](#) on page 5 for an explanation of the product and file version numbers.

5 NYCeScope

5.1 Introduction

You can trace almost any variable of a NYCe 4000 node, drive and axis, with the NYCeScope tool, which makes NYCeScope a useful tool for

- Current loop tuning
- Position loop tuning
- Performance monitoring
- I/O monitoring
- Error monitoring

NYCeScope has been designed to resemble a real oscilloscope. First, you select the data you want to view with the "Channels..." button, and set the trace settings in the NYCeScope Control Panel. Then NYCeScope is ready for tracing. [fig. 5-1 "NYCeScope Control Panel" on page 187](#) shows the Control Panel of the NYCeScope.

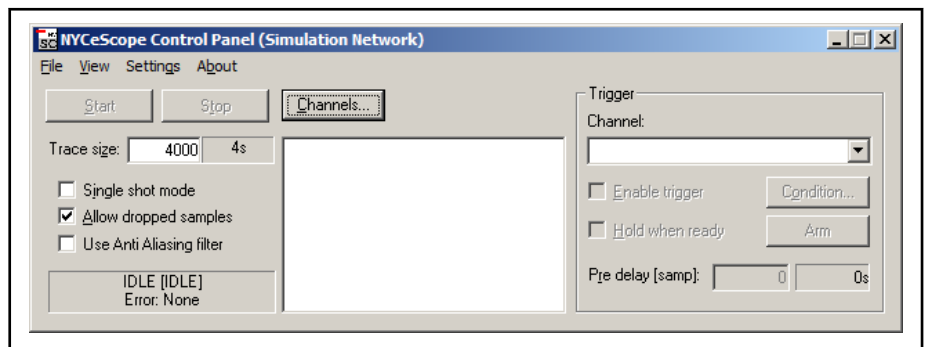


Fig. 5-1: NYCeScope Control Panel

NYCeScope also includes the possibility to save data traces to files, and NYCeScope can also import these saved files and display the trace data again in the appropriate plot window, see [chapter 5.6 "Export and import trace data" on page 202](#). Furthermore, the complete state of NYCeScope, called a "NYCeScope session", can be saved and restored; see [chapter 5.7 "Sessions" on page 205](#).



If NYCeScope is active, the exclusive write access functionality is not available. See NYCe 4000 Software User Manual for detailed information about the exclusive write access functionality.



You can start more instances of NYCeScope.

5.2 Trace channel selection

The traceable variables can be assigned to a trace channel via the "Channels..." button in the NYCeScope Control Panel. Do the following steps to add variables to a trace or remove selected variables from a trace.

1. Click the "Channels..." button in the NYCeScope Control Panel.

The "Select Trace Channels" dialog window appears, see [fig. 5-2 "Select Trace Channels dialog" on page 188](#).

NYCeScope

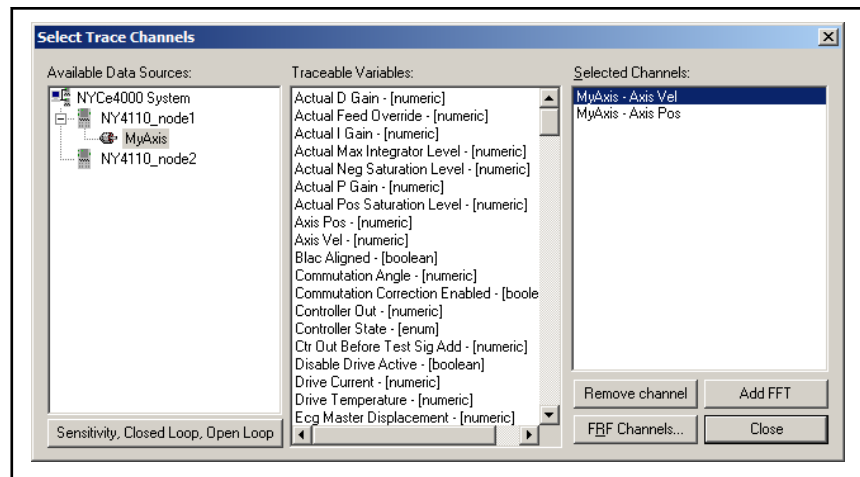


Fig. 5-2: *Select Trace Channels dialog*

2. Select the node in the "Available Data Sources" section to select the node from which you want to trace node variables, or click the axis to select the axis variables you want to trace.
3. Select the variable in the "Traceable Variables" section to select a variable, and click the "Add channel" button. The selected variable is added in the "Selected Channels" section. Double-clicking a variable in the "Traceable Variables" section gives the same result.
4. To remove a selected variable from the "Selected Channels" section, select that variable. The button "Add channel" changes to "Remove channel". Click the "Remove channel" button to remove the variable from the "Selected Channels" section. Double-clicking a variable in the "Selected Channels" section gives the same result.



NYCeScope uses the Variable Trace functionality of the NYCE subsystem to collect data. For some variables, data is not from the same MCU sample as the rest. Take this into consideration when you compare data for analysis. See NYCe 4000 Software User Manual for detailed information about the variable trace functionality and data aging.



Note that the variables contained in the trace are limited to 4 nodes on an NY4110-based system and 24 nodes on an NY4112-based system.



You can also trace the so-called High-Speed Data (HSD) variables with NYCeScope. See NYCeConfigurator, [chapter 3.4 "Configuration at the node level" on page 62](#) and the NYCe 4000 Software User Manual, chapter NYCE, "High-speed data trace". In NYCeScope you can always define channels for 2 HSD variables, even if the node sample frequency is 1 kHz. When the axis is initialized, a warning is displayed in the NYCeScope Control Panel (lower left corner). The value of the second added HSD variable channel is 0.

You can trace node variables and axis variables with NYCeScope.

- At the node level, you can trace I/O and MCU related variables, for example
 - Service / Stop_Axes inputs
 - Analog inputs / outputs
 - (Fast) digital inputs and outputs
 - Sample duration
- At the axis level, you can trace motion related variables, for example
 - Setpoint generator variables
 - Position loop variables
 - Drive variables
 - Error variables

The type of a variable can be numeric, boolean, enumeration or bitfield. All types of variables can be analyzed in the time domain. Numeric typed variables can also be analyzed in the frequency domain.

If a channel is created while no time window is open, a time window is opened. If one or more channels are already created, the new channel is added to the existing time window.

When you click a selected variable in the "Selected Channels" section, the "Add FFT" button is enabled. When you click the "Add FFT" button, a power spectrum plot of the trace will be either added to a new Scope Signal Spectrum Plot window or to an existing Scope Signal Spectrum Plot window.

Besides the power spectrum of variables, you can also select the Frequency Response Function between 2 variables. When you click the "FRF Channels..." button, the "Create Frequency Response Function Channel" dialog window appears which displays the created Frequency Response Function (FRF) channels, and enables the user to create new FRF plots from the selected channels.

The following three FRF channels can be created automatically with the "Sensitivity, Closed Loop, Open Loop" button in the "Select Trace Channels" dialog window.

- The sensitivity transfer function
- The closed loop transfer function
- The open loop transfer function

Only the sensitivity is actually measured. The closed and open loop are calculated from the sensitivity.

5.3 Trace setup

5.3.1 Introduction

Just like a real oscilloscope you must configure NYCeScope to correctly capture traces.

5.3.2 Trace size

Enter an appropriate trace size specified in samples in the field "Trace size". Consider the expected trace dynamics (microseconds? / milliseconds?). Traces are collected at the highest sample rate of the selected channels. For example, selecting a trace size of 1 Msamples for a 1 kHz variable captures data for a longer duration than a 32 kHz HSD variable.

NYCeScope

5.3.3 Trigger source, slope, and trigger level

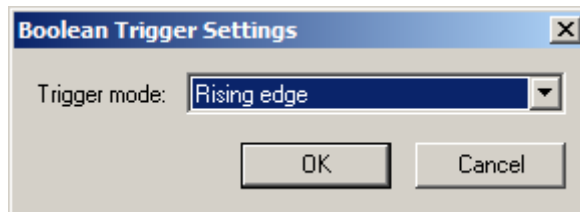
Choose the correct trigger source, slope and trigger level (if applicable).

Do the following steps to select a trigger channel and set the trigger condition.

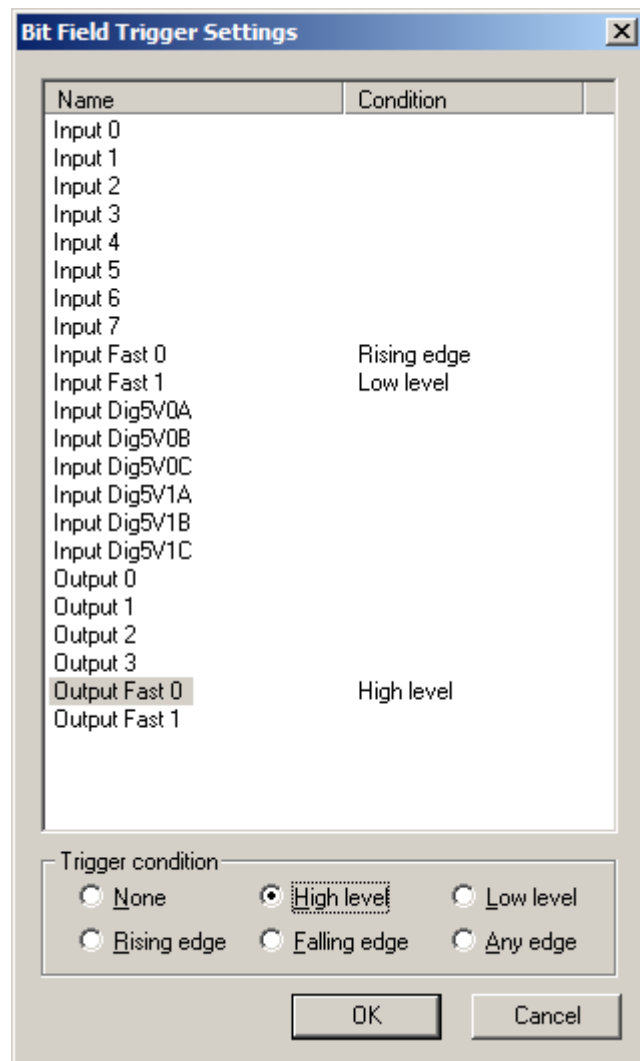
1. Select a channel from the "Channel" drop-down list in the "Trigger" section of the NYCeScope Control Panel.

After a channel is selected the appropriate trigger condition dialog box appears.

- If you select a boolean typed variable for trigger condition, the only selection available in the dialog window is the "Trigger mode" drop-down box where you specify whether the trigger is edge sensitive or level sensitive.



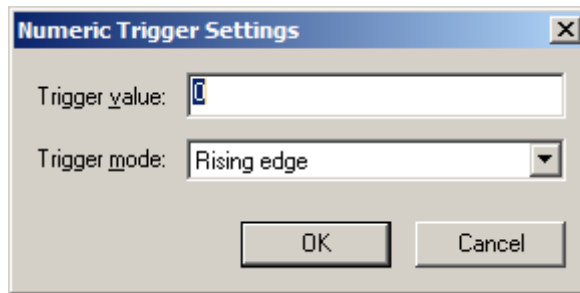
- If you select a bit field channel for trigger condition, you specify in the dialog window the trigger condition for the lines.



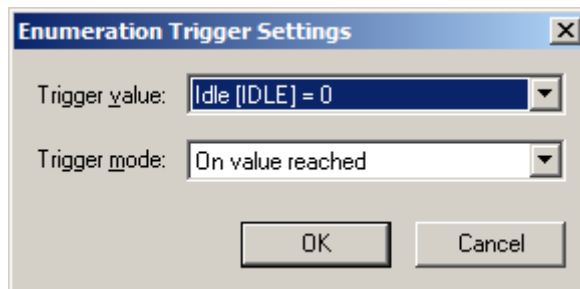
To create a trigger condition, click on the signal name, for example "Input Fast 0" and then click the radio button in the "Trigger condition" section to select the trigger condition. You can select more signal names, each with its own trigger condition. If you select more than one signal, the trigger condition is satisfied if the condition for all specified signals is satisfied, thus the trigger condition is an AND operation of all selected signal conditions. Note that the trigger condition is evaluated every sample. If you select two signals with a rising or falling edge, the edge transition for both signals must occur during the same sample, else the trigger condition is not satisfied. It is best to specify only one edge transition channel for a trigger condition.

- If you select a numeric typed variable for trigger condition, you specify in the dialog window the "Trigger mode" and a trigger value.

NYCeScope



- If you select an enumerated typed variable for trigger condition, you specify in the dialog window the "Trigger mode" and a trigger value.



2. Put a check mark in the checkbox "Enable trigger" to enable the trigger selection.

You must accept a trigger condition, else you cannot enable the trigger. If you accept the trigger condition immediately after a channel is created, the trigger is automatically enabled.

You can only enable a trigger of a channel that is actually available.

3. With the checkbox "Enable trigger" you can always enable and disable the trigger condition.
4. Put a check mark in the checkbox "Hold when ready" if you do not want the trace to start again when the trigger condition is (again) satisfied. If the checkbox "Hold when ready" has a check mark, you must click the "Arm" button to arm the trigger specification again.

Click the "Condition..." button in the "Trigger" section to modify the trigger condition of a selected channel. If you select another trigger channel, all specified trigger conditions are reset.

With the "Pre delay [samp]" field you can define the amount of data that is traced before the sample at which the trigger condition is satisfied.

The trigger state is displayed in the upper righthand corner of the plot windows. The following trigger state indications are possible.

- WAIT FOR TRIGGER...
This trigger state indicates that a trigger has been defined and is enabled. The trigger condition is not yet satisfied, or, if "Pre delay [samp]" is not zero, data gathering is started but the pre-delay time is not expired.
- TRACING

The trigger condition is satisfied; NYCeScope is gathering trace data.

- **READY**

NYCeScope has gathered trace data which is displayed in the plot window; no new trace is active.

- **WAIT FOR ARM TRIGGER**

This trigger state is entered after the **READY** state, if the checkbox "Hold when ready" has a check mark. You must click the "Arm" button to change the trace state to **WAIT FOR TRIGGER...**

- **NO DATA**

The trace is stopped by the "Stop" button in the NYCeScope Control Panel before the trigger condition was satisfied.

Put a check mark in the "Single shot mode" checkbox, if you want to stop tracing completely if a full trace is available. To start a new trace you must use the "Start" button. All previously collected data is immediately discarded.

Put a check mark in the "Use Anti Aliasing filter" checkbox to enable the anti-aliasing filter.



Anti-aliasing only operates on servo axes configured on an NY4120, NY4125, NY4130, or NY4140 drive module (with the PVL on the drive module), Further, anti-aliasing only operates on the following PVL variables: "position error", "controller out", "controller-out-before-testsignal-addition", and the test signal if injection point is at controller out.

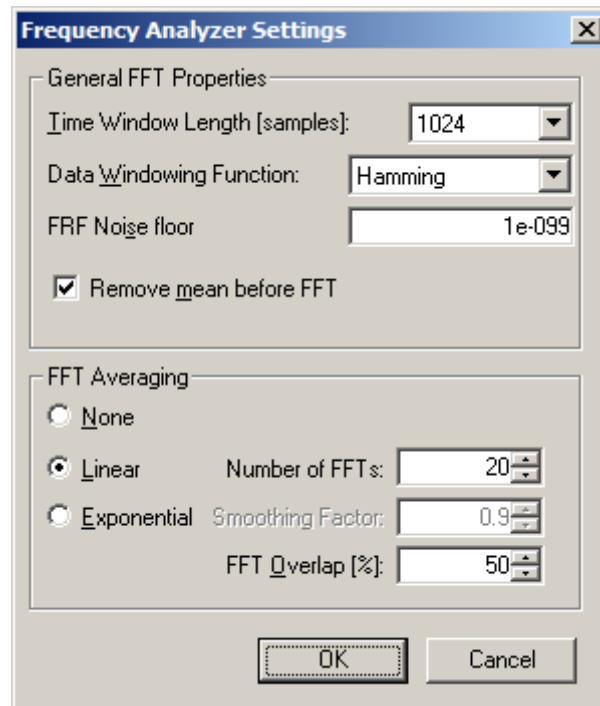
When you trace variables that have a higher update rate than the generation of trace data, for example the variable "position error", you may need to activate the anti-aliasing filter. The anti-aliasing filter has a cut-off at the node sample frequency. Therefore, the amplitude of the measured signal is not exactly identical to the actual signal. Note that HSD variables do not need anti-aliasing filtering, because they are traces at the same high frequency. Also, the ratio between the PVL frequency and the trace rate must be 4 or higher, else the anti-aliasing filter is not activated (for example, when the PVL runs at 16 kHz, and the MCU runs at 8 kHz).

5.3.4 Frequency Analyzer

Select the "Settings" entry in the menu bar and select the entry "FFT Settings..." to change the frequency analyzer settings.

The "Frequency Analyzer Settings" dialog box appears.

NYCeScope



- Set the time window length for the FFT calculations. A longer FFT window results in a higher frequency resolution / lower time resolution, but also increases the delay between the measurement and the display of the FFT. Likewise, a shorter FFT window results in lower frequency resolution, but higher time resolution.
- You can choose a windowing function with the "Data Windowing Function" drop-down box. A window function is a trade-off between leakage reduction, frequency resolution, amplitude measurement error and other aspects. The available windowing functions are Blackmann-Harris, Blackmann, Hanning, Hamming, Triangular and Boxcarr (none). The following observations are given.
 - If only leakage reduction is important, for example while looking for a small signal in the spectral vicinity of a larger signal, use the Blackmann-Harris windowing function.
 - For measurement, the trade-off between leakage reduction and interpolation errors is best when the Hanning windowing function is used.
 - The triangular windowing function has the lowest worst-case processing loss. If a small signal must be detected in broadband noise you can use the triangular windowing function.
 - The Boxcarr windowing function is best used when you must separate 2 equally strong frequencies, but has worst leakage reduction.
- Put a check mark in the checkbox "Remove mean before FFT" to remove the average value (DC / 0 Hz offset) of a measurement before the FFT is applied. This removes 'leakage' of the DC component in the FFT result.
- Set the averaging method for FFT measurement. For signals with fast changing frequency plots this option should be disabled. Averaging is

used to improve plots of noisy measurements and dynamic range of the measurement.

5.3.5 Start the measurement

- Use the "Hold when ready" checkbox, if you do not want NYCeScope to automatically arm the trigger after each trace is completed.
- Click the "Start" button to start the measurement.

5.4 Trace display manipulation

5.4.1 Time domain analysis

You can configure the Time Domain Plot window, Signal Spectrum Plot window and FRF plot window while NYCeScope started tracing, that is, after you clicked the "Start" button. The Time Domain Plot window can display all the traces of its type at once. [fig. 5-3 "NYCeScope time plot"](#) on page 195 shows an example of a Time Domain Plot.



Fig. 5-3: NYCeScope time plot

Use the "Auto-scale" button for quick lay-out and vertical scaling. It auto-scales data in the horizontal visible range. The bitfield typed variables are always displayed at the bottom of the plot. Above the bitfield typed variables plots are the boolean typed variables displayed, and above the bitfield typed variables are the enumerated typed variables displayed. Finally, in the top part of the plot window, the numeric typed variables are displayed.

Numeric typed variables are plotted using linear interpolation between the measured data points. Not-numeric typed variables are plotted in a "sample & hold" style.

When zooming in on the time axis far enough, the individual trace sample moments are plotted using purple-colored ticks on the time axis. Note that the trace sample time is the sample time of the selected trace variable with the highest update rate.

NYCeScope draws lines to connect the measured data samples. For numeric data a line is drawn to the next data sample. For non-numeric data a

NYCeScope

horizontal line is drawn starting from a sample point, and the drawn level is changed at the next sample. This drawing behavior explains why non-numeric plotted lines run to the end of the plot and numeric plotted lines "stop" at the last data sample. The effect is visible in [fig. 5-4 "Numeric typed and not-numeric types variables in a plot"](#) on page 196. It is not a trace data inconsistency, but a result of how the numeric and not-numeric data is represented.

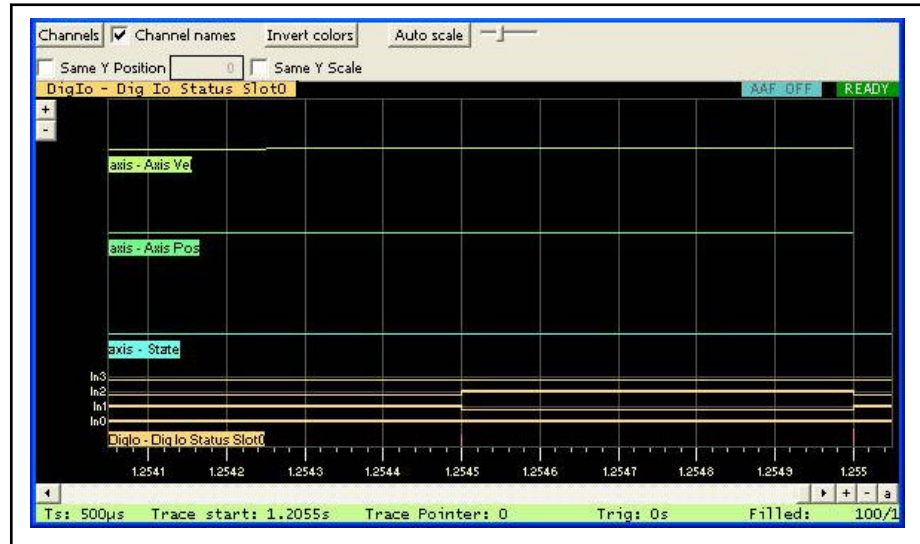


Fig. 5-4: Numeric typed and not-numeric types variables in a plot

5.4.2 Control options in the display window

The signals displayed in the Time Domain Plot can be scaled simultaneously with the "+" and "-" buttons in the time window itself (Y axis zoom function). The time (X) axis can be scaled with the "+" and "-" buttons in the lower right hand corner. The "a" button displays all data at full time range.

You can use the mouse to select a zoom region in the time window. The zoom region is shown as a rubber band box and is defined by selecting the zoom region from any corner to the diagonally opposite corner (click the left mouse button, and while holding down the mouse button, draw a box).

If you click the right mouse button in the plot, a pop-up menu with zoom options appears. The first 2 entries in the pop-up menu set the operation mode of the mouse while dragging the mouse pointer. "Zoom mode" performs the zooming of a plot as described above. If "Pan mode" is selected, you can use the mouse to perform panning in horizontal and vertical direction of the plot window. The other 3 entries in the pop-up menu ("Zoom X in", "Zoom X out", and "Zoom X full") have the same functionality as the "+", "-", and "a" buttons in the lower right hand corner.

With the checkbox "Channel names" you can hide or display the channel names of the displayed signals in the time plot. The "Channels" button hides or displays the "Channels" display control window, see [fig. 5-5 "Channels panel"](#) on page 197.

The "Invert colors" button inverts the color of the plot. You can choose between a black background and a white background.

With the checkbox "Same Y Position" you can ensure that all numeric typed channels are positioned such that the specified Y value coincides. You enter the Y value in the field next to the "Same Y Position" checkbox.

If you put a check mark in the checkbox "Same Y Scale" you ensure that the zoom factor is identical for all numeric typed channels.

When both checkboxes are check marked, the ruler for numeric typed channels will be identical for all numeric typed channels.

If you click the "Auto scale" button at the top of the Time Domain Plot, all displayed channel plots are auto scaled vertically, also using the "Same Y Position" and "Same Y Scale" settings. The slider next to the "Auto scale" button allows you to set the amount of "vertical separation" between the displayed plots of the numeric typed variables. Set the slider before you click the "Auto scale" button. If the slider is set to the left-most position, the Y scale is set to maximum and the displayed plots (of the numeric typed variables) have total overlap. If the slider is set to the right-most position, the Y scale is set to minimum and the displayed plots have no overlap.

Channels panel

You can control the displayed plot of each channel with the "Channels" panel. Each channel is identified by a color. If a variable is not available (for example, because the associated node was removed from the network), the trace in the plot is still shown, but the channel display of that variable is greyed out.

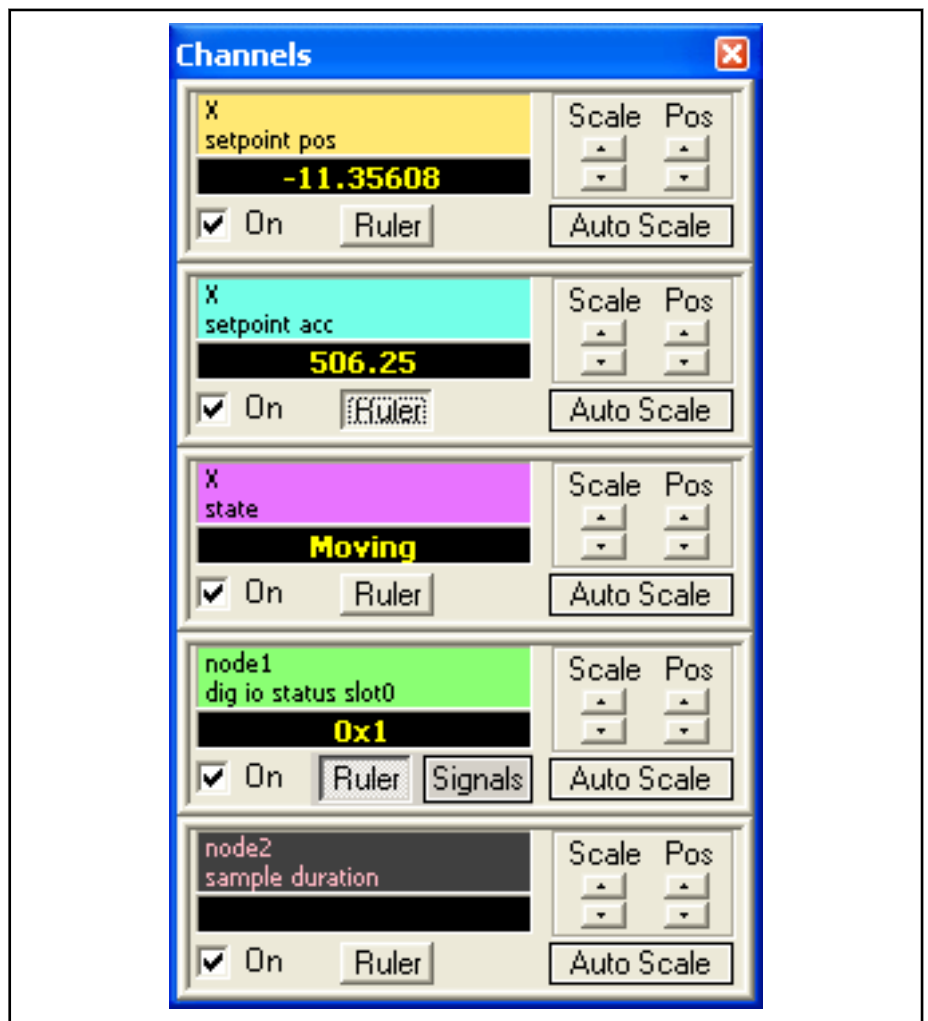


Fig. 5-5: Channels panel

NYCeScope

With the "On" checkbox, you select whether or not a channel is plotted in the time plot window.

You can use the "Scale" selectors, "Pos" selectors, and the "Auto Scale" button to improve the view of each trace separately.

In the Time Domain Plot window, the amplitude of a trace can be measured by clicking the ruler button. Only one ruler button can be active.

If the channel is for a bit field type variable, an additional button "Signals" is shown. When you click the "Signals" button the "Bit Field View Settings" dialog window appears.

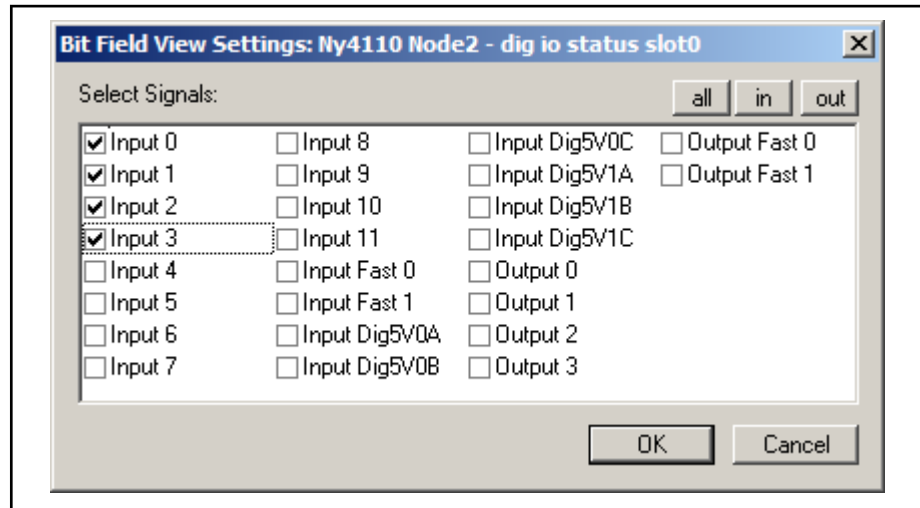


Fig. 5-6: Bit Field View Settings

In the "Bit Field View Settings" dialog window you can select which signal is included in the time plot window. You can select and deselect every signal individually with its checkbox. For Digital I/O, the buttons "all", "in", and "out" select or deselect either all inputs and outputs, all inputs, or all outputs respectively.

5.4.3 Frequency domain analysis

The Signal Spectrum Plot window can only display one FFT channel, see [fig. 5-7 "NYCeScope FFT plot"](#) on page 199.

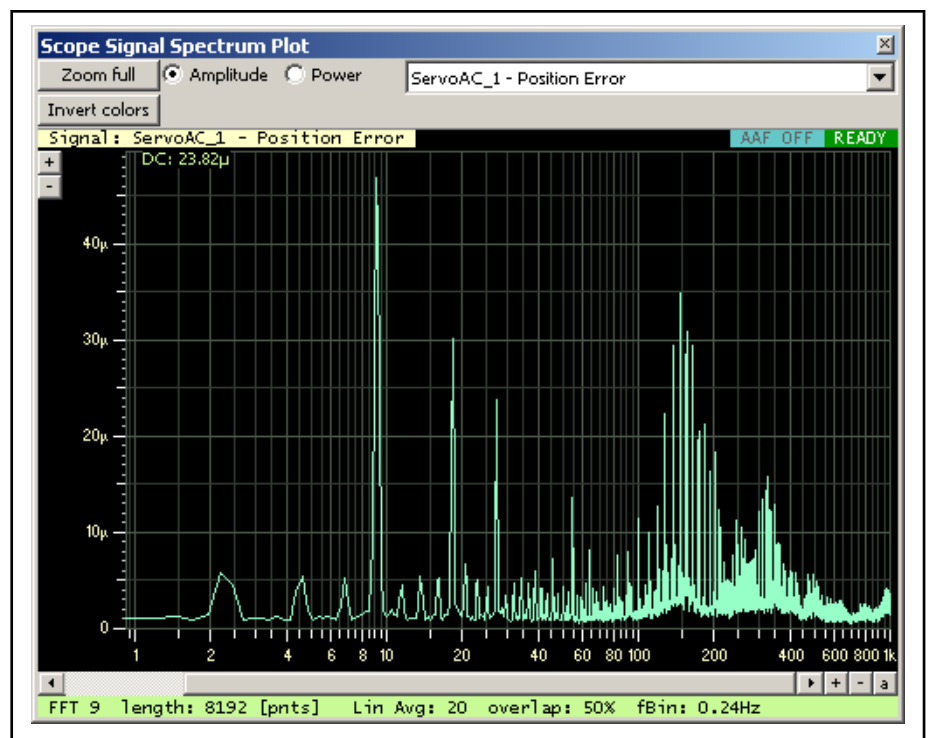


Fig. 5-7: NYCeScope FFT plot

Checking different FFT plots simultaneously can only be done by viewing them in separate FFT views (click the entry "View → New FFT plot" in the menu bar). At the upper right part of the Signal Spectrum Plot window is a selector, with which the different FFT traces can be selected.

The type of FFT display can be set to Amplitude spectrum or Power (A)² spectrum with the two radio buttons.

Use the "Zoom full", "+" and "-" buttons to manipulate the view of the FFT plot.

In the top left corner of the plot you can see the DC component value (0 Hz offset).

The Direct Transfer Function Estimate Plot (FRF) window has a similar "look and feel" as the Signal Spectrum Plot window (see [fig. 5-8 "NYCeScope FRF Bode plot" on page 200](#)). The difference is the fact that a frequency response is a measurement between 2 channels.

NYCeScope

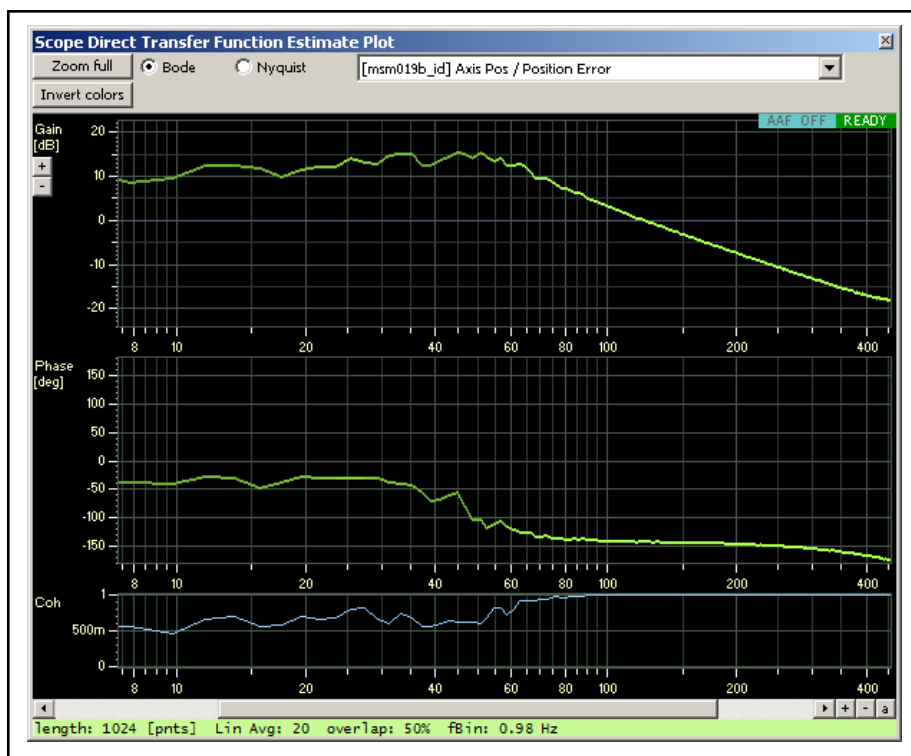


Fig. 5-8: NYCeScope FRF Bode plot

The plot shows the gain (upper plot) and phase (middle plot) of the FRF. The lower plot shows the coherence for the estimated FRF. The coherence is an estimate of the causality between input and output. The FRF plot uses the coherence to display the signal more or less bright. A higher coherence value gives a more brightly displayed signal and a lower coherence gives a dimmed displayed signal.

The FRF can also be displayed as a Nyquist diagram with the "Nyquist" radio button, see [fig. 5-9 "NYCeScope FRF Nyquist diagram" on page 201](#).

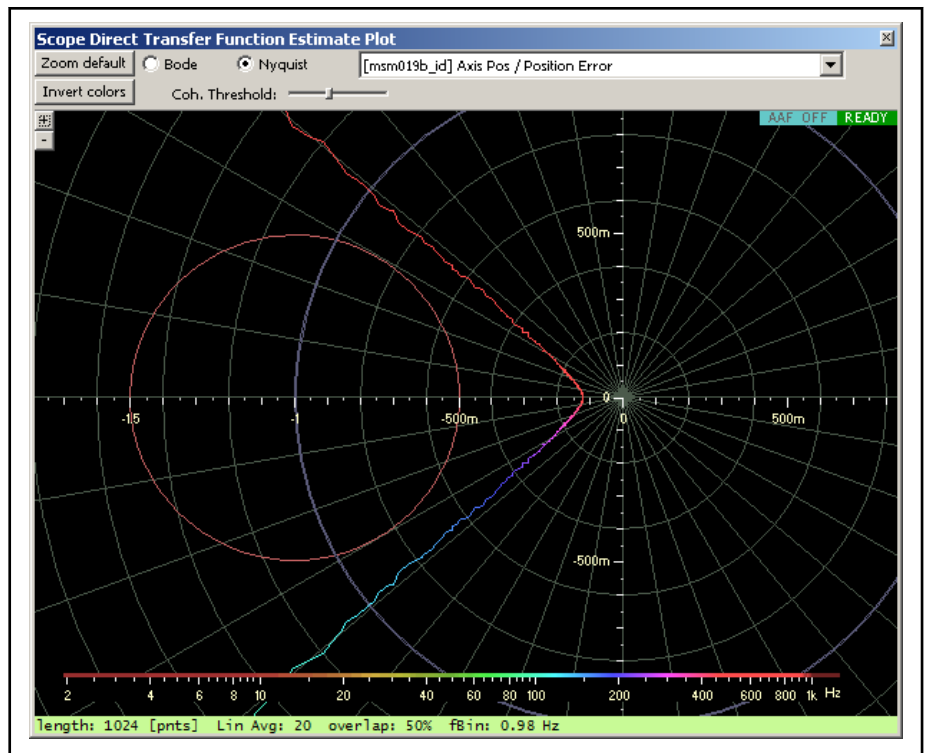


Fig. 5-9: NYCeScope FRF Nyquist diagram

You can set the coherence threshold for plotting with the "Coh. Threshold" slider. If the coherence is lower than the set threshold, the frequency information is not included in the plot. This enables the removal of possible incorrect data.

The frequency scale at the bottom of the Nyquist diagram is colored, where the color represents the frequency along the Nyquist plot.



Frequency domain analysis (FFT and FRF) can only be done correctly if the nodes, from which variables are traced, run at the same sample frequency.

If frequency measurements are done on more than one node, and the node sample frequencies are different, keep in mind that the trace sample time is determined by the node with the highest sample frequency in the system. In such a system, traced signals of a node that runs at a lower sample frequency may show strange results, due to aliasing effects. This applies to FFT, Bode and Nyquist plots.

5.5 Trace states

NYCeScope displays the trace state in the lower left corner to inform the user about the status of the trace activity, see [fig. 5-10 "The trace state display of NYCeScope" on page 202](#).

NYCeScope

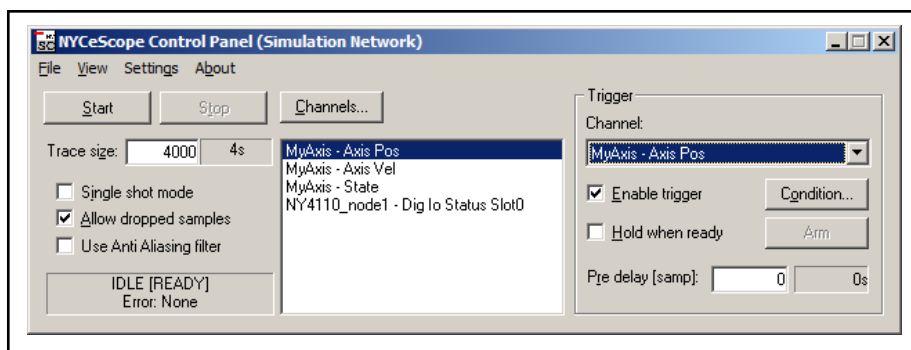


Fig. 5-10: The trace state display of NYCeScope

Two states are displayed, the global state of NYCeScope and the state of the NYCe 4000 trace. The global state of NYCeScope is either "IDLE" (NYCeScope is not collecting data) or "TRACING" (NYCeScope is collecting data). The global state is printed first, followed by the NYCe 4000 trace state between square brackets.

- IDLE [IDLE]
In this state no trace channels have been selected.
- IDLE [READY]
This is the state after variable(s) are selected, and the trace is not running. The trace channel data is still available for analysis in the time or frequency windows, or can be exported to a text file.
- IDLE [ERROR]
Tracing is not running, caused by an error while the trace buffer was being filled. The buffer may contain valid data of the traced variables before the error condition occurred. You still can analyze this data with NYCeScope or export the data to a file.
- TRACING [RUNNING]
In this state, the trace data is being collected.
A stop trace command has not yet been given. In this state it is possible to retrieve blocks of trace data. It is also possible to restart the trace with a new start trace command.
- TRACING [STOPPING]
Tracing is stopping, not all trace data is processed yet.

5.6 Export and import trace data

5.6.1 Introduction

With NYCeScope you can trace several variables of a motion system. The trace results can be stored ("exported") in a file. The data in the file can be used for analysis with external programs.

Stored trace results can also be loaded ("imported") in NYCeScope. NYCeScope displays the data in the appropriate window. Imported trace files can, for example, be used for reference purposes.

The following trace data can be exported; each type of trace data creates its own data file.

- Time channels
This file contains the time-sampled (traced) data.

- FFT channels
This file contains the frequency domain transformed data (amplitude per frequency band).
- FRF channels
This file contains data of the available frequency response functions (gain, phase shift and coherence per frequency band).

5.6.2 Export trace data

If the trace is not active, thus the global state is "IDLE", you can export the traced data.

Do the following steps to save a trace to a file.

1. Click the "File" entry in the menu bar and select the entry "Export Data...".

The "NYCeScope Export Channels" dialog box appears.

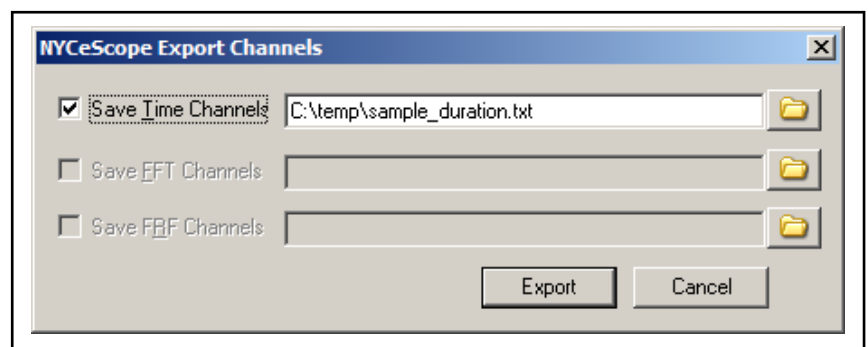


Fig. 5-11: NYCeScope Export Channels

2. To save data you specify the type of data with the checkbox and a file name for each selected measurement type. You can select more than one measurement type. You cannot save the different measurement types to the same file.
 - 2.1 Put a check mark in the checkbox before the channel type identification (for example "Save Time Channels").
 - 2.2 Click the folder icon after the channel type identification entry field (for example "Save Time Channels").

The "Select file to save time signals" dialog box appears (in this example).

You can select an existing file or enter a new file name and click the "Save" button. If you select "Save FFT Channels", the "Select file to save FFT signals" dialog box appears, and if you select "Save FRF Channels", the "Select file to save FRF signals" dialog box appears. Note that, when you click the "Save" button, the trace data is not immediately saved to the specified file.

Repeat this step if you also want to save trace data of the other channels.

3. Click the "Export" button in the "NYCeScope Export Channels" dialog box.

Note that only the data of the Channels with a check mark are actually saved to the specified files.

The data is stored in a space-delimited text file.

NYCeScope

The file consists of two parts, the header information and the trace data. All lines in the header start with the percentage character. After the header information is the data. Note that the data will not contain any other lines starting with a "%" character.

Header field description

- The number shown in the "Trigger channel" line indicates the used trigger channel. If this number is 0, it indicates that no trigger was specified.
- The number shown in the "Trigger time" line indicates the trace sample clock number at which the trigger condition was satisfied. This number is 0 if no trigger was specified.
- If the number shown in the "Number of value tables" line is not 0, a list follows for every enumerated type variable (for example state variables and I/O bits). The number shown in the "Number of values" line represents the number of lines that follow, containing a translation of the traced "ENUM" number to a value and description text.

Program:

```
% NYCESCOPE EXPORT FILE   TIMECHANNELS   VERSION 1
%
% NYCe4000 version 45.6.1.3
%
% Sample time[seconds] : 0.001
% Number of samples    : 6
% Number of channels   : 5
% UTC time of sample 0 : 2016-06-30 11:03:49.673
%
% Channel 1 : AXIS , X , SAC_VAR_STATE , ENUMERATED , VALUE_TABLE_1
% Channel 2 : AXIS , X , SAC_VAR_SPG_STATE , ENUMERATED , VALUE_TABLE_2
% Channel 3 : NODE , node1 , NHI_VAR_DIG_IO_STATUS_SLOT0 , BITFIELD , VALUE_TABLE_3
% Channel 4 : AXIS , X , SAC_VAR_SETPOINT_ACC , NUMERIC
% Channel 5 : AXIS , X , SAC_VAR_SETPOINT_POS , NUMERIC
%
% Trigger channel      : 4
% Trigger time        : 1885
%
% Number of value tables : 3
% VALUE_TABLE_1
% Number of values     : 2
% 0 : 0 , "SAC_MOVING" , "Moving" , "MOV"
% 1 : 1 , "SAC_READY" , "Ready" , "RDY"
%
% VALUE_TABLE_2
% Number of values     : 2
% 1 : 1 , "SAC_SPG_IDLE" , "Idle" , "IDLE"
% 2 : 2 , "SAC_SPG_PTPING" , "PointToPoint-ing" , "PTPNG"
%
% VALUE_TABLE_3
% Number of values     : 22
% 1 : 0 , "NYCE_DIGIN0" , "Input 0" , "In0"
% 2 : 1 , "NYCE_DIGIN1" , "Input 1" , "In1"
% 4 : 2 , "NYCE_DIGIN2" , "Input 2" , "In2"
% 8 : 3 , "NYCE_DIGIN3" , "Input 3" , "In3"
% 16 : 4 , "NYCE_DIGIN4" , "Input 4" , "In4"
% 32 : 5 , "NYCE_DIGIN5" , "Input 5" , "In5"
% 64 : 6 , "NYCE_DIGIN6" , "Input 6" , "In6"
% 128 : 7 , "NYCE_DIGIN7" , "Input 7" , "In7"
% 4096 : 8 , "NYCE_FASTIN0" , "Input Fast 0" , "InF0"
% 8192 : 9 , "NYCE_FASTIN1" , "Input Fast 1" , "InF1"
% 16384 : 10 , "NYCE_DIG5VIN0A" , "Input Dig5V0A" , "In5V0A"
% 32768 : 11 , "NYCE_DIG5VIN0B" , "Input Dig5V0B" , "In5V0B"
% 65536 : 12 , "NYCE_DIG5VIN0C" , "Input Dig5V0C" , "In5V0C"
% 131072 : 13 , "NYCE_DIG5VIN1A" , "Input Dig5V1A" , "In5V1A"
% 262144 : 14 , "NYCE_DIG5VIN1B" , "Input Dig5V1B" , "In5V1B"
% 524288 : 15 , "NYCE_DIG5VIN1C" , "Input Dig5V1C" , "In5V1C"
% 16777216 : 16 , "NYCE_DIGOUT0" , "Output 0" , "Out0"
% 33554432 : 17 , "NYCE_DIGOUT1" , "Output 1" , "Out1"
% 67108864 : 18 , "NYCE_DIGOUT2" , "Output 2" , "Out2"
% 134217728 : 19 , "NYCE_DIGOUT3" , "Output 3" , "Out3"
% 268435456 : 20 , "NYCE_FASTOUT0" , "Output Fast 0" , "OutF0"
% 536870912 : 21 , "NYCE_FASTOUT1" , "Output Fast 1" , "OutF1"
%
```

NYCeScope

1882	0	2	0	-11999.9949	0.982
1883	0	2	0	-7999.99552	0.99467
1884	0	2	0	-3999.99488	0.99933
1885	1	1	0	0	1
1886	1	1	0	0	1
1887	1	1	0	0	1

Data field description

- The first column of the saved data represents the trace sample clock. The trace sample clock starts at 0, where 0 represents the first sample after the trace was started. The sample clock is specified by the "Sample time" in the header field.
- The following columns show the trace data of the specified channels.

5.6.3 Import trace data

When you saved trace data with the option "Export Data...", you can import that data. The data is automatically presented in a plot window in NYCeScope.

Do the following steps to import (and display) data.

1. Click the "File" entry in the menu bar and select the entry "Import Data...".

The "Import NYCeScope data file" dialog box appears.

2. Enter the file name, or select the file, and click the "Open" button.

The selected file is imported, and when the import operation succeeds a plot window is displayed with the imported data. The file name of the selected file is shown in the title bar of the plot window.

The data of every imported file is displayed in its own plot window, thus, if you import multiple files, each file has its own plot window.

The displayed plot window depends on the data contained in the selected file. Time channel data is displayed in a Time Plot window, FFT channel data is displayed in a Spectrum Plot window, and FRF channel data is displayed in an FRF window (Bode or Nyquist plot).

If the selected file does not have the correct format or contains invalid data, the import operation is cancelled and an error message is displayed.



The "Import Data" option is introduced in NYCe 4000 Software Release 38VRS and required a change of the file format. For this reason the exported data from previous versions of NYCeScope cannot be imported.

The import operation and display of the plot window does not interfere with other activities of NYCeScope; an active trace or other plot window handling continues.

5.7 Sessions

When you start NYCeScope you get all the windows and settings as they were when you closed NYCeScope the previous time. NYCeScope stores information in a specific file which is read when NYCeScope is started again. This file contains, among others, the following information.

- All channel information.
- Trigger setting information.

NYCeScope

- All open windows (time, FFT, FRF) with their settings, selections and positions.

This complete set of selected channels, configured settings and windows layout is called a "NYCeScope session", or short "session".

Note that the actual traced data or imported trace data files are not part of the session. However, just like the trace data, the session can be saved, and loaded from a NYCeScope session file.

The NYCeScope session file has the extension ".nyscope". This extension is associated with NYCeScope. If you double-click on a file with the extension ".nyscope", NYCeScope starts and loads the settings stored in the session file.

When a session is open, NYCeScope uses that session, and the session file name is shown in the title bar of NYCeScope.

The "File" entry in the menu bar of NYCeScope offers the following session file related selections.

- **New Session**
Create a new session; the settings are loaded from the default session file. Before the new session is started, a popup message appears if changes are made to ask whether the current session must be saved.
- **Open Session...**
Open and load the settings from the specified session file. Before the "Open Session File" dialog window is shown, a popup message appears if changes are made to ask whether the current session must be saved.
If the opened session contains the definition of channels that do not exist in the currently connected motion system, because a component (node or axis) is no longer part of the actual configuration, the channel and component(s) are displayed "greyed-out" to indicate that they are not available.
- **Save Session**
Save the current settings to the opened (active) session file. The "Save Session" entry is only available if a session is opened.
- **Save Session As...**
Save the current settings to a new session file. The "Save Session File" dialog window appears.
- **Most Recently Used**
Open a session from a history list.

When NYCeScope is closed, its current state is stored per user. When NYCeScope is started again, the previous state is restored per user.

5.8 Miscellaneous

When you click the icon at the left in the title bar, the popup with the standard Windows functions (for example "Move") appears. The last entry of the popup ("About NYCeScope") displays the "About NYCeScope" dialog window. You can also open the "About NYCeScope" dialog window if you click the "About" entry in the menu bar.

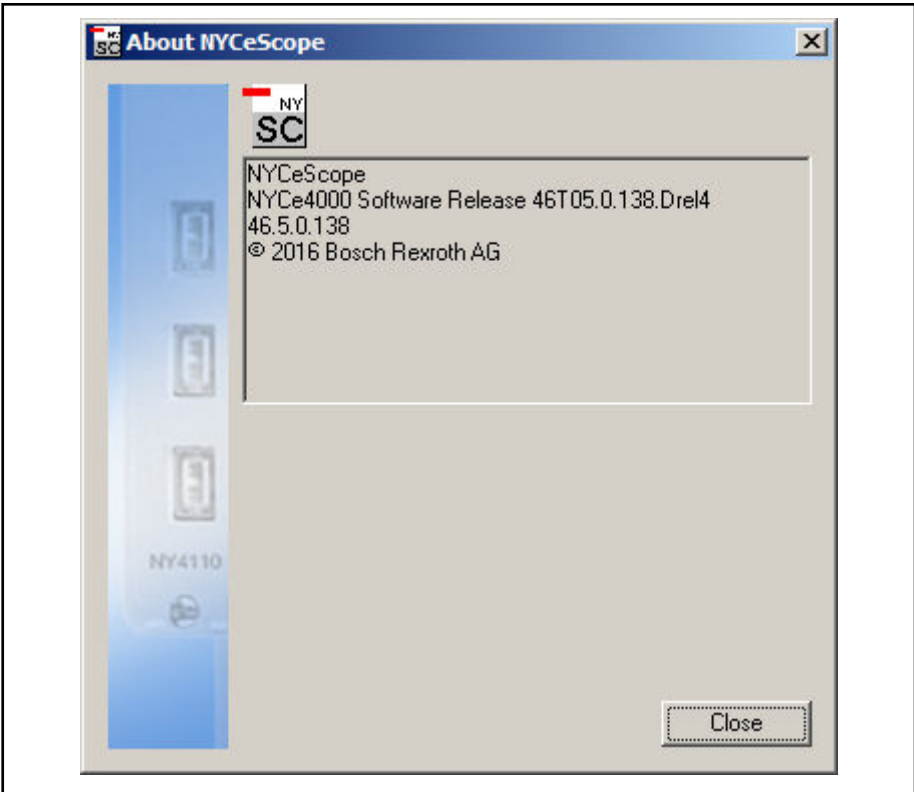


Fig. 5-12: The "About NYCeScope" dialog window

See [chapter 1.3 "NYCe 4000 product and file version information"](#) on page 5 for an explanation of the product and file version numbers.

6 NYCeInspector

6.1 Introduction

The NYCeInspector is designed for diagnostic testing and can be used to inspect the status of the nodes, drives and axes in the NYCe 4000 system. It also provides an intuitive way for setting the values of digital and analog outputs. In addition, several controls for changing the state, position and feed override of an axis are available.

6.2 Overview

The NYCeInspector can be used to perform the following tasks.

- At node level
 - View the status of the Service and Stop_Axes inputs
 - View the number of connected clients
 - View the sample time, sample duration and the maximum sample duration
- At drive level
 - View the status of digital I/O
 - View the status of analog I/O
 - View the I/O functions of digital I/O
 - Set the values of digital and analog outputs
 - View the status of digital input counters
 - View the drive temperature and the servo voltage
- At axis level
 - View the status of the axis
 - Move the axis using jog or point-to-point
 - Stop the axis using quick stop or smooth stop
 - Align the axis
 - Home the axis
 - View the associated digital I/O of I/O functions
 - View the status of the feed override
 - Apply a feed override
 - View the latched positions
 - View the number of connected clients
 - View the axis position and other important position variables like the axis velocity, controller output, position error and integrator output

[fig. 6-1 "NYCeInspector window" on page 210](#) shows the NYCeInspector window. In the pane at the left side you can select a component from the "NYCe4000 System" tree. The selected component is indicated in the bottom part of the NYCeInspector window. Using the middle pane, you can make adjustments for the selected component. The pane at the right side displays information about the position of an axis.

Both the left and the right pane can be hidden with the ">>" , respectively the "<<" button to save space on the desktop.

NYCeInspector

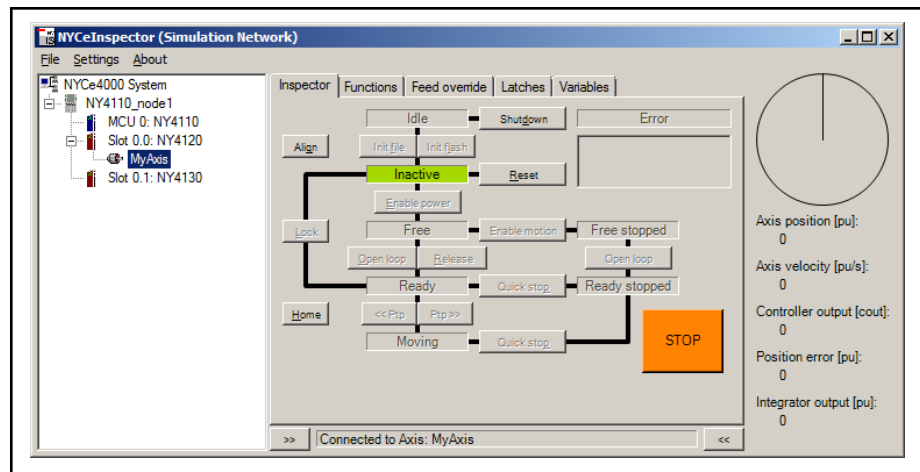


Fig. 6-1: NYCeInspector window

6.3 Inspection at node level

If a node is selected in the "NYCe4000 System" tree, the middle pane shows one tab named "General".

"General" tab

On the "General" tab, the status of the Service and Stop_Axes inputs of the selected node is displayed. For both inputs the "Logical value" and the "Active state" are shown. The logical value is visualized using a check box, but cannot be changed from within the NYCeInspector. See [fig. 6-2 "NYCeInspector "General" tab" on page 210](#).

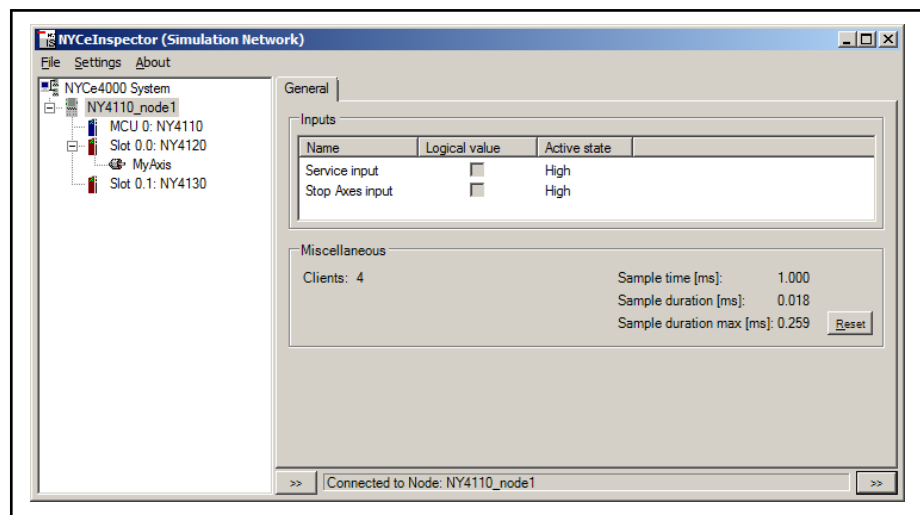


Fig. 6-2: NYCeInspector "General" tab

Furthermore, the values of the parameter "Sample time" and the variables "Sample duration" and "Sample duration max" are displayed. The variable "Sample duration max" holds the maximum sample duration and can be reset using the "Reset" button displayed next to its value.

The displayed number of connected clients can be used to determine if the node is properly disconnected after usage by other applications.

6.4 Inspection at drive level

If a drive is selected in the system tree, the middle pane shows three tabs named "I/O", "Counters" and "Variables".

"I/O" tab

On the "I/O" tab, the status of the digital and analog I/O is displayed. For digital inputs the "Logical value", "Active state" and defined "Function" are displayed. For digital outputs the "Logical value", "Active state", "Output type" and "Function" are displayed. For analog I/O only the "Value" is displayed. See [fig. 6-3 "NYCeInspector "I/O" tab" on page 211](#).

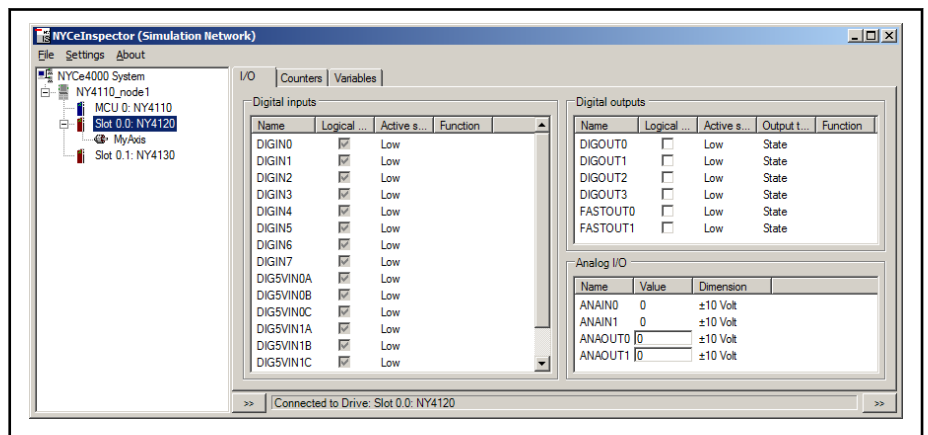


Fig. 6-3: NYCeInspector "I/O" tab

The defined I/O functions are displayed in the format `axis.io_function`. Here, `axis` and `io_function` denote the name of the axis and the I/O function respectively, for example `MyAxis.AreaSensor`.

The logical values of digital I/O are visualized using check boxes. Only the logical values of digital outputs can be changed from within the NYCeInspector.

The values of analog outputs are visualized using text boxes that can be used to change them. When the data of a text box is changed, the text color is changed to red. This indicates that the displayed value is not the actual but the edited value, see [fig. 6-4 "Editing the value of an analog output" on page 211](#). The text color remains red until the edited value is committed by pressing 'Enter' or discarded by pressing 'Esc'. While the text box has the input focus, its value will not be updated to prevent interruption of the editing process.

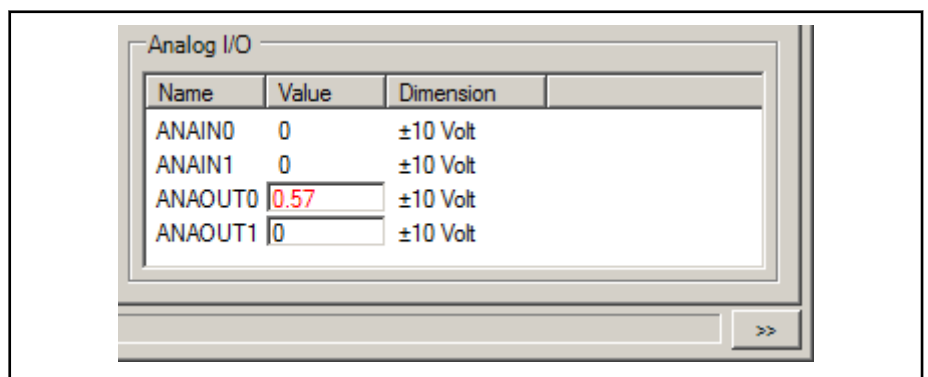


Fig. 6-4: Editing the value of an analog output

"Counters" tab

On the "Counters" tab, the status of defined digital input counters is displayed. If a digital input counter is defined, its "Value", associated "Digital input" and "Edge type" are shown.

"Variables" tab

On the "Variables" tab, the values of the variables "Drive temperature" and "Servo voltage" are displayed.

6.5 Inspection at axis level

If an axis is selected in the system tree, the middle pane shows five tabs named "Inspector", "Functions", "Feed override", "Latches" and "Variables". Additionally, if the right pane is not hidden, it displays information about the position of an axis.

"Inspector" tab

On the "Inspector" tab, a state diagram for the selected axis is shown, see [fig. 6-1 "NYCeInspector window" on page 210](#). The current state is indicated in color while other states are colored gray. Buttons are provided to make state transitions, and to execute motion commands and stop commands. A complete list of supported commands is shown in [tab. 6-1 "Supported commands on "Inspector" tab" on page 212](#).

Transitions	Motion	Stop	Other
Shutdown	Point-to-point	Quick stop	Align
Reset	Jog	Smooth stop	Home
Initialize			
Enable power			
Enable motion			
Lock			
Hold			
Open loop			
Release			

Tab. 6-1: Supported commands on "Inspector" tab

The availability of the buttons for executing motion commands and stop commands depends on the selected motion type and stop type. By default, point-to-point and quick stop are selected. A different motion type or stop type can be selected in the "Settings" menu. The parameters used for motion commands and stop commands can also be changed via this menu.

The buttons for executing motion commands and stop commands are disabled if the parameters are incorrect.

The orange button named "STOP" can be used to stop the motor using quick stop if for some reason the motor must be stopped in a controlled, but as fast as possible manner.

Motion settings

The parameters used for point-to-point or jog movements can be changed in the "Motion settings" dialog. This dialog is available from the "Settings" menu or by pressing Ctrl+1. The motion settings are remembered per axis until the NYCeInspector is shutdown.

Stop settings

The parameters used for quick stop or smooth stop can be changed in the "Stop settings" dialog. This dialog is available from the "Settings" menu or by pressing Ctrl+2. The stop settings are retrieved from and stored in the SAC subsystem.

The new stop settings are saved only in the memory of the node. Therefore, if the axis is shutdown and initialized from flash or file, the settings will be overwritten.

"Functions" tab

On the "Functions" tab, status information of I/O functions is displayed. If an I/O function is assigned, the associated "digital I/O", "slot number" and "value" are shown.

"Feed override" tab

On the "Feed override" tab, status information of the feed override functionality is displayed, see [fig. 6-5 "NYCeInspector "Feed override" tab" on page 213](#). The status information consists of the "State", which can be either active or inactive, and the current "Feed override" factor applied to the motion profile.

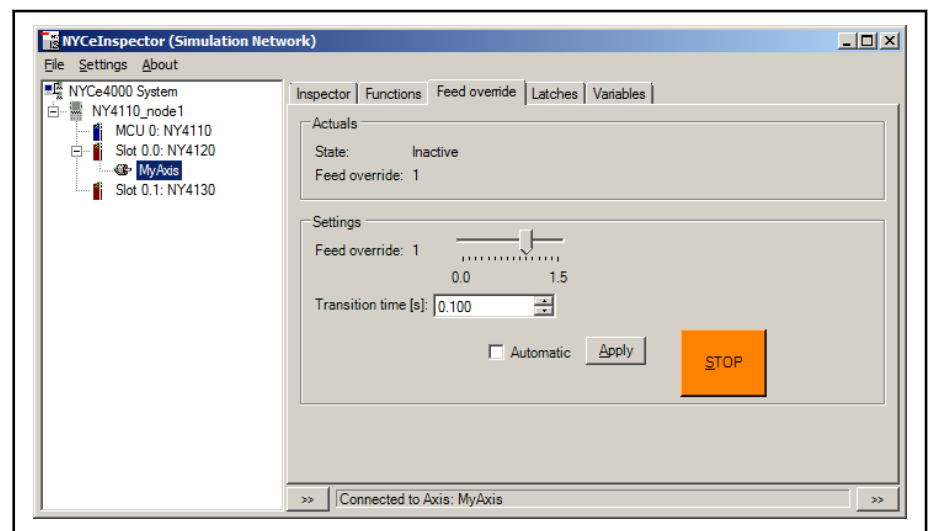


Fig. 6-5: NYCeInspector "Feed override" tab

From this tab it is also possible to apply a new feed override. To apply a new feed override, specify the "Feed override" factor and "Transition time" and click the "Apply" button. If the "Automatic" check box is checked, a new feed override is automatically applied when the feed override factor is changed using the slider.

The "Apply" button will be disabled if the "Automatic" check box is checked.

NYCeInspector

The "STOP" button can be used to stop the axis using quick stop if for some reason the axis must be stopped in a controlled, but as fast as possible manner.

For more information about the feed override functionality, please refer to the NYCe 4000 Software User Manual, chapter "Feed override".

"Latches" tab

On the "Latches" tab, the latched positions of the selected axis are displayed. Initially, only the single latched positions for each latch are shown. The multiple latched positions can be viewed by expanding the single latched positions. See [fig. 6-6 "NYCeInspector "Latches" tab" on page 214](#).

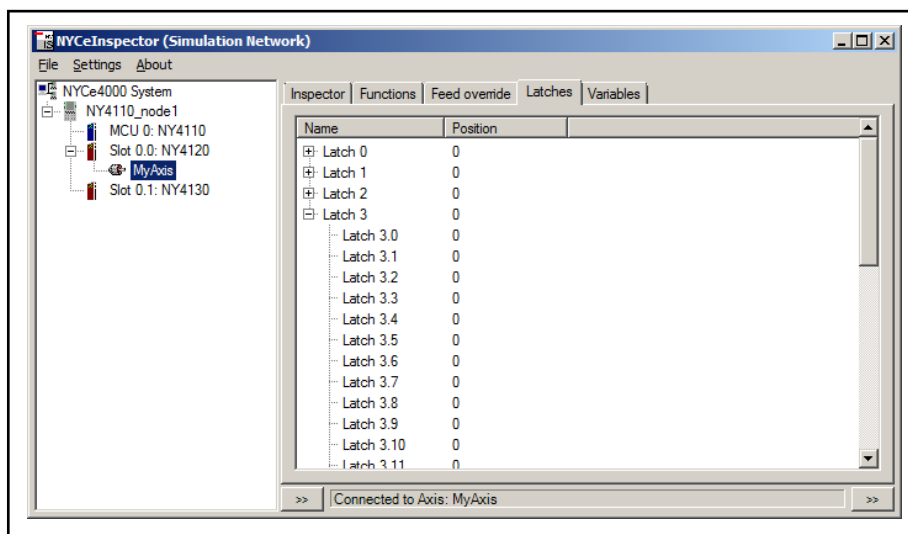


Fig. 6-6: NYCeInspector "Latches" tab

"Variables" tab

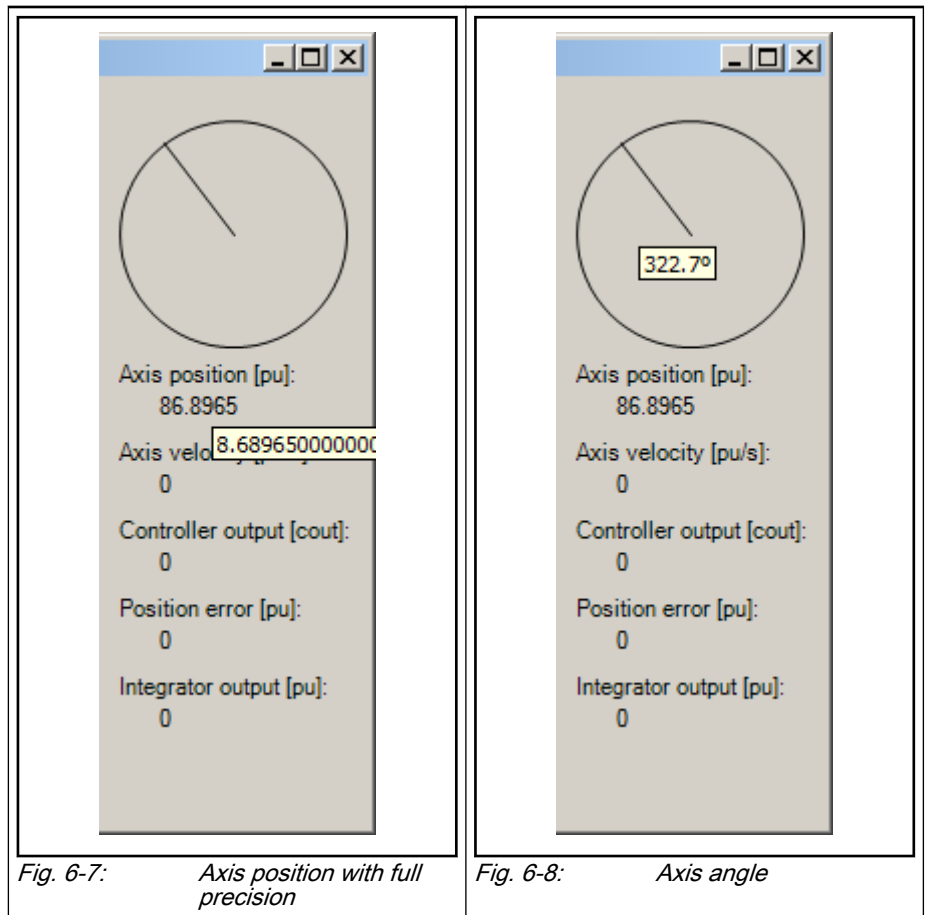
On the "Variables" tab, you can see the number of connected clients. This value can be used to determine if the axis is properly disconnected after usage by other applications.

"Axis" pane

The "Axis" pane at the right side of the NYCeInspector window displays the "Axis position" and other important position variables like the "Axis velocity", "Controller output", "Position error" and "Integrator output". The pane can be hidden by clicking the "<<" button, or shown by clicking the ">>" button. The hidden or visible state of the pane is stored on shutdown of the tool and restored the next time the tool is started.

Most numerical values in the tool are displayed using as much digits as needed, but no more than 7. To view more digits of the axis position, a tool tip is provided. The tool tip displays the axis position in scientific notation using a fixed number of 17 digits. See [fig. 6-7 "Axis position with full precision" on page 215](#).

The axis position is also displayed graphically, showing only the angle of the axis. A tool tip is available to view the numerical value of the angle. See [fig. 6-8 "Axis angle" on page 215](#).



6.6 Miscellaneous

When you click the icon at the left in the title bar, the popup with the standard Windows functions (for example "Move") appears. The last entry of the popup ("About NYCeInspector") displays the "About NYCeInspector" dialog window. This dialog window is also displayed when you select "About" from the menu bar.

NYCeInspector

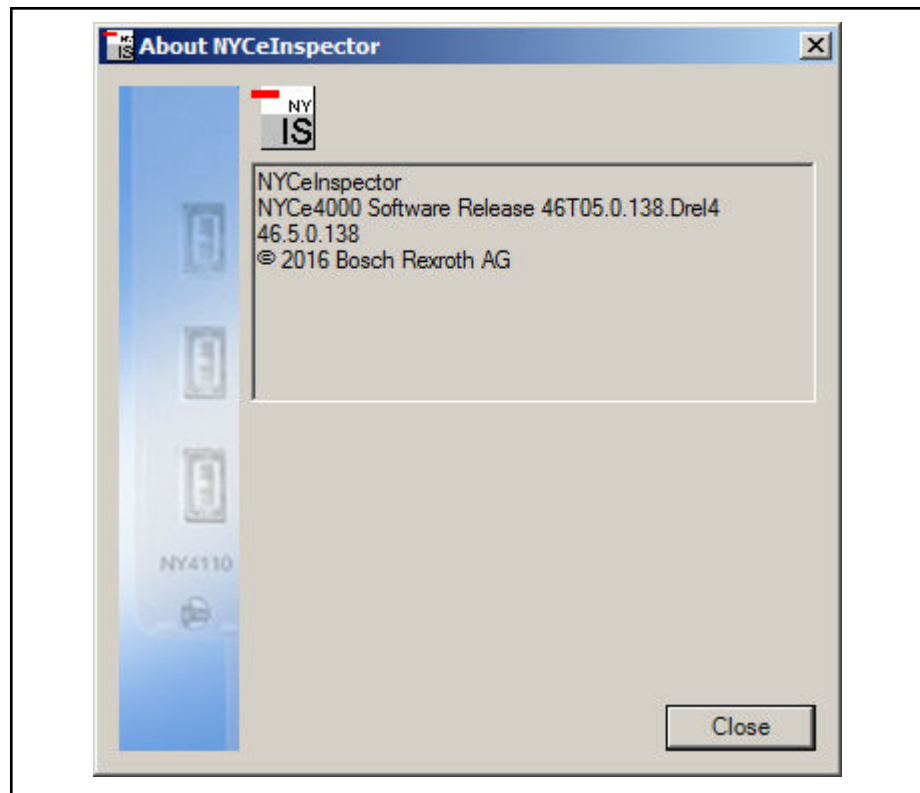


Fig. 6-9: The "About NYCeInspector" dialog window

See [chapter 1.3 "NYCe 4000 product and file version information"](#) on page 5 for an explanation of the product and file version numbers.

7 NYCeCommand

7.1 Introduction

Once the configuration and tuning of an axis (or axes) is completed, you can test the movements of one or several axes using a test script. In addition, you can use analog and digital I/O in the same script.

NYCeCommand is a tool that enables the user to perform motion and issue I/O commands in a short time, without the use of a C programming environment. The commands have the same names and input arguments as in the C programming environment and almost all commands that are supported in C are also supported in this tool. The number of arguments that you can enter for a command given on the command line is dynamic.

7.2 Commandline window

When you start NYCeCommand, the tool always opens in the "Commandline" window. In the "Commandline" window you can enter single commands which are executed immediately. The entered command and the result are displayed in the command output pane, see [fig. 7-1 "NYCeCommand Commandline window" on page 217](#). Commands that are executed successfully, return NYCE_OK. If the result of an entered command generates information, you can see that information with NYCeLogger. Only the output of the string conversion functions is displayed in the command output pane.

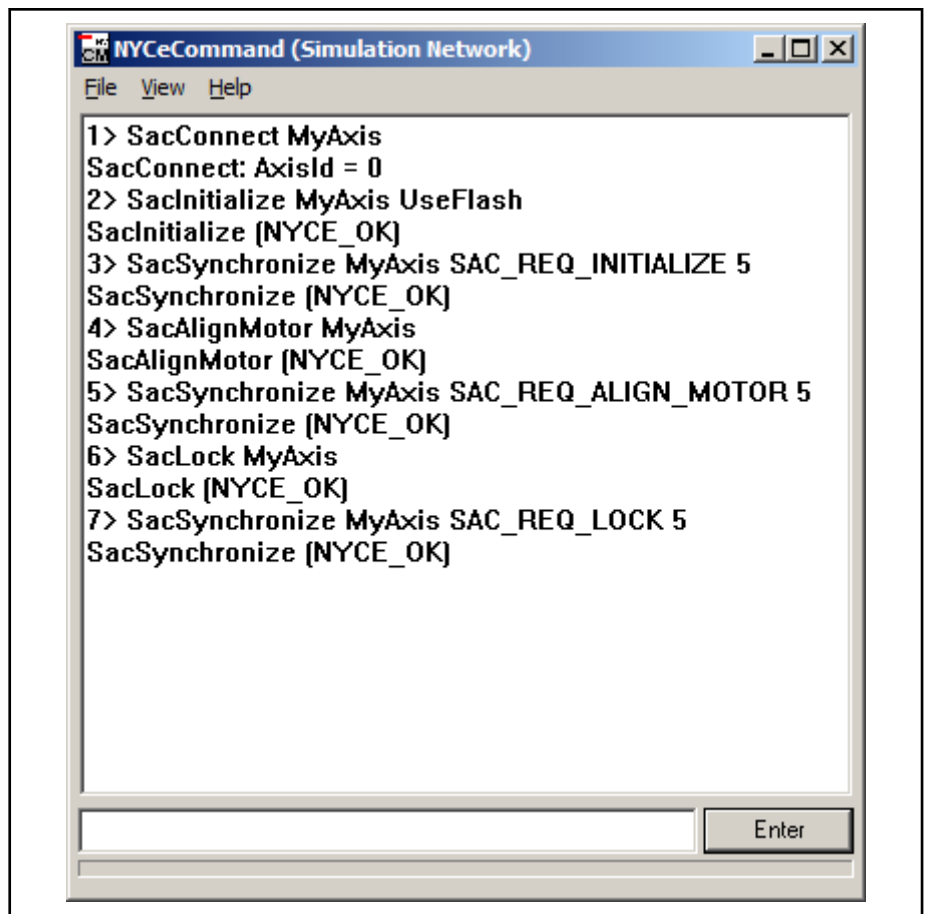


Fig. 7-1: NYCeCommand Commandline window

NYCeCommand

7.3 Script editor window

NYCeCommand also supports the execution of a script. A script is a file that contains one or more commands that are executed consecutively.

Click "File → Run commands from script..." to execute the commands in the script file after you created the script file. The commands and results are displayed in the command output pane.

NYCeCommand includes a script editor. Click "View → Script editor" in the menu bar to switch from the "Commandline" window to the "Script editor" window.

Click "File → Load and display script..." to load a previously written script in the "Script editor" window but you can also enter commands and save the entry in a script file. The entered commands are executed when you click "Run script" in the menu bar, see [fig. 7-2 "NYCeCommand script editor window"](#) on page 218.

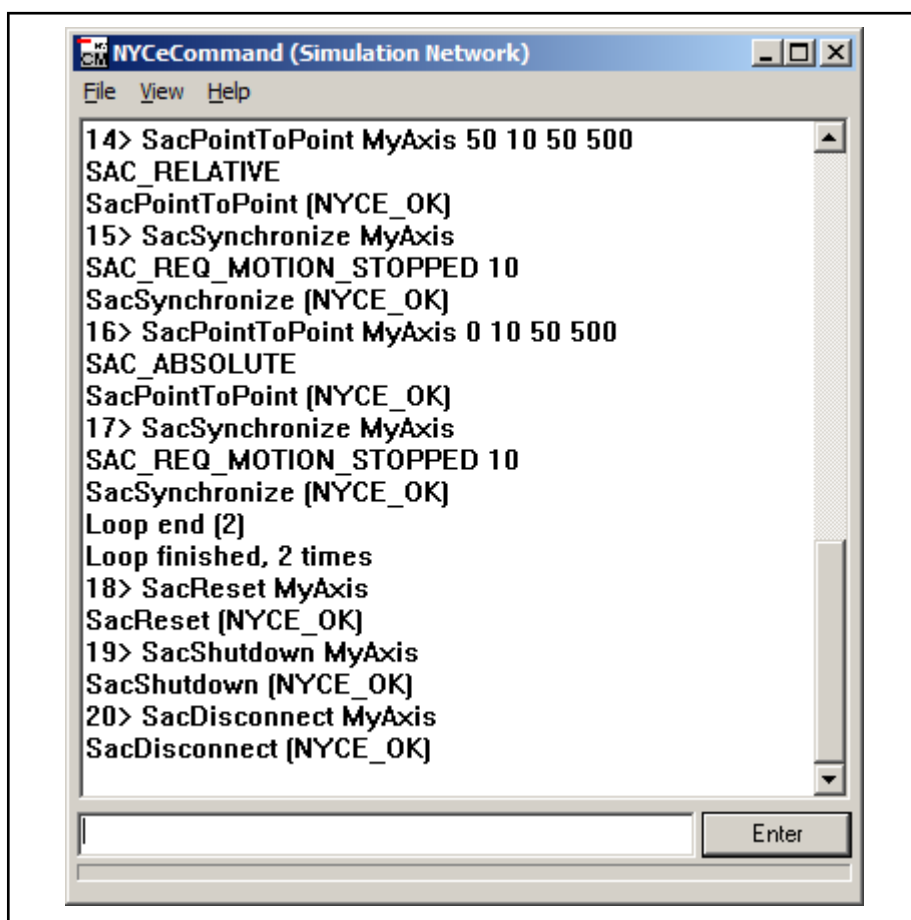


Fig. 7-2: NYCeCommand script editor window

Note that when you click "Run script" from the script editor menu bar, the script is run, but the output is not displayed. To see the output, you must return to the Commandline window and then run the script.

You can edit scripts with the NYCeCommand tool but you can also write scripts with a simple text editor like Notepad.

7.4 Help window

The NYCeCommand tool includes an extensive help function. Press the "F1" key or click "Help → Index" to open the Help information, see [fig. 7-3 "The NYCeCommand help" on page 219.](#)

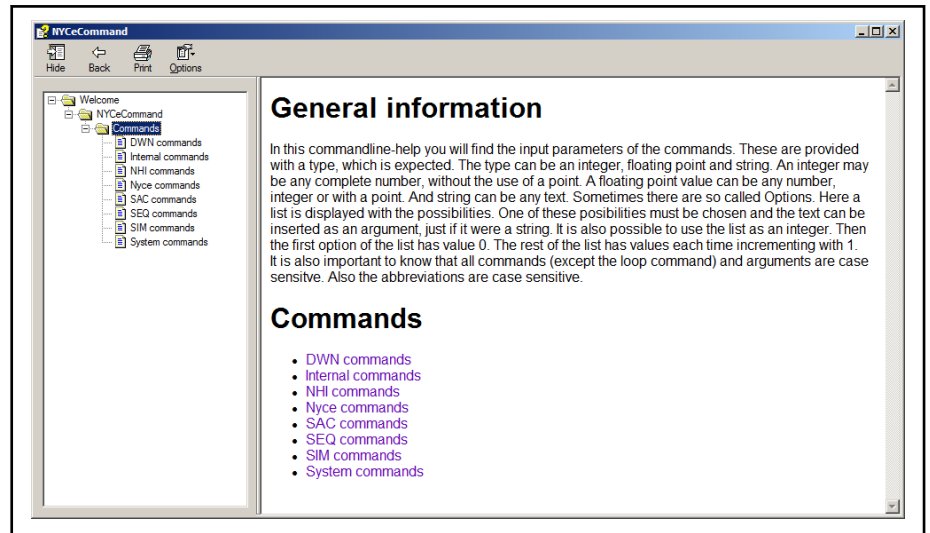


Fig. 7-3: The NYCeCommand help

7.5 Using NYCeCommand from the "CMD" prompt

The NYCeCommand tool can also be used via the MS Windows command prompt (for example, when you make a batch file which must run a sequence of several NYCeCommand script files). The command line syntax is

```
NYCeCommand [/I <scriptname> [/Q]]
```

with:

Parameter	Description
/I	Run a script at startup
<scriptname>	Name of the script
/Q	Quit after script has completed
/?	Display the command line options

7.6 Script example

The following script is an example of a NYCeCommand script. To use this script, either configure an axis with the name MyAxis, or change the MyAxis axis name in the script into the name of the axis to be driven by the script.

Program:

```
# -----
# Connecting / initializing MyAxis
SacConnect MyAxis
SacInitialize MyAxis UseFlash
# Waiting for initialization to be completed
SacSynchronize MyAxis SAC_REQ_INITIALIZE 5

# -----
# motion part
# Align MyAxis (Enable if motor is BLAC)
SacAlignMotor MyAxis
```

NYCeCommand

```
# Waiting for alignment to be completed
SacSynchronize MyAxis SAC_REQ_ALIGN_MOTOR 5

# lock MyAxis
SacLock MyAxis

# Waiting for lock to be completed
SacSynchronize MyAxis SAC_REQ_LOCK 5

# Home MyAxis
SacHome MyAxis

# Waiting for homing to be completed
SacSynchronize MyAxis SAC_REQ_HOMING_COMPLETED 5

# start loop, with 2 repeats
Loop 2

# One simple motion
SacPointToPoint MyAxis 50 10 50 500 SAC_RELATIVE

# Synchronize motion with 10s timeout
SacSynchronize MyAxis SAC_REQ_MOTION_STOPPED 10

# Back to home
SacPointToPoint MyAxis 0 10 50 500 SAC_ABSOLUTE

# Synchronize motion with 10s timeout
SacSynchronize MyAxis SAC_REQ_MOTION_STOPPED 10

# End loop
Loop end

# Resetting motors
SacReset MyAxis

# -----
# Closing part

# Disconnect MyAxis
SacShutdown MyAxis
SacDisconnect MyAxis
```

7.7 Miscellaneous

When you click the icon at the left in the title bar, the popup with the standard Windows functions (for example "Move") appears. The last entry of the popup ("About NYCeCommand") displays the "About NYCeCommand" dialog window. This dialog window is also displayed when you select "Help → About" from the menu bar.

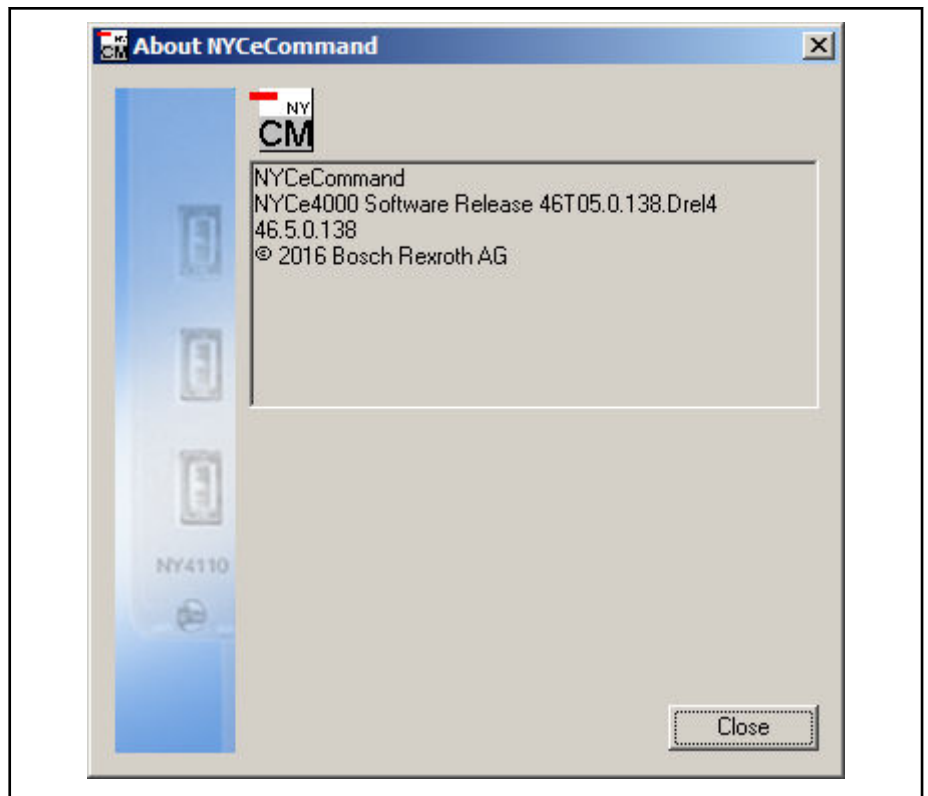


Fig. 7-4: The "About NYCeCommand" dialog window

See [chapter 1.3 "NYCe 4000 product and file version information"](#) on [page 5](#) for an explanation of the product and file version numbers.

8 NYCeLogger

8.1 Introduction

The NYCeLogger is a useful tool for debugging. If an application program is written and run, all executed NYCe 4000 commands and their results can be logged and written to a file. This is useful for debugging purposes and useful for Bosch Rexroth customer support. [fig. 8-1 "NYCeLogger - Output window" on page 223](#) shows an example of the NYCeLogger window.

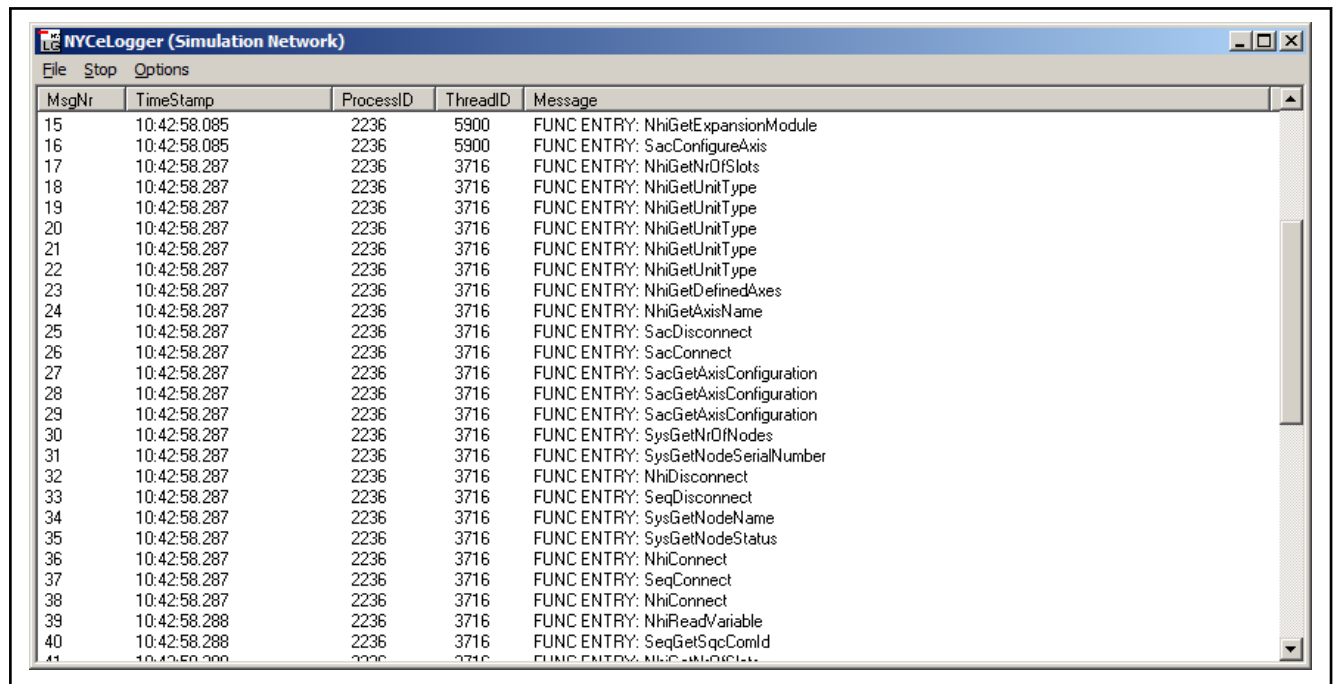


Fig. 8-1: NYCeLogger - Output window

The NYCeLogger output window shows the time stamp as "wall-clock" time from the PC. The notation is HH:MM:SS.ms, where ms is milliseconds in 3 digits. The output window also shows the process identifier and thread identifier along with the message.

8.2 Logging options

Click "Options" in the menu bar of the NYCeLogger to open the "Logging options" window, see [fig. 8-2 "NYCeLogger - Logging options window" on page 224](#).

NYCeLogger

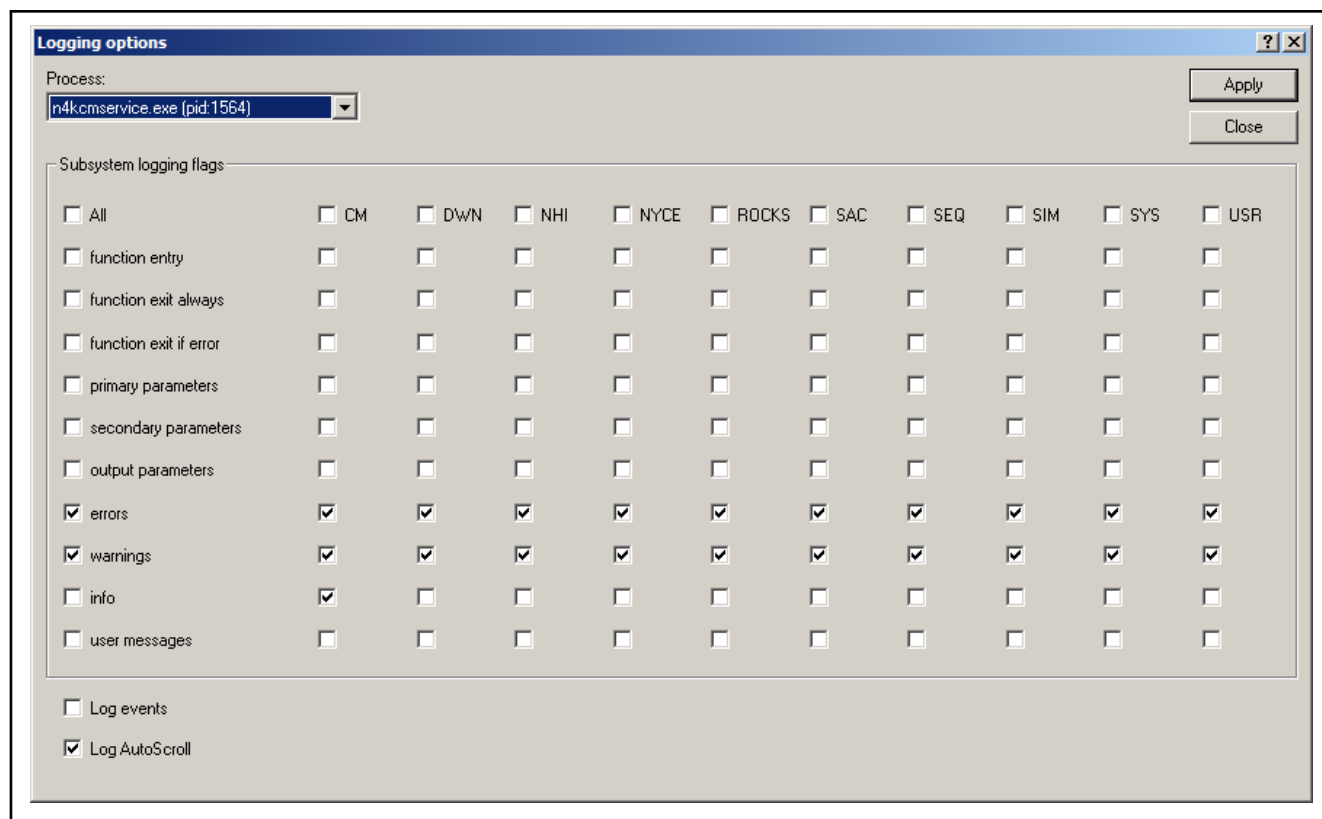


Fig. 8-2: NYCeLogger - Logging options window

In the "Process" section you can select the process from which you want to log data. The process identifier is included to identify the correct program if more than one instance is running. Using the Windows Task Manager you can see which instance has which process identifier.

The "Subsystem logging flags" section has several areas.

- "All" checkbox. If a check mark is put in this checkbox, all subsystems and all message types are selected.
- "function entry", "function exit always", ... , "user messages" checkboxes. If a check mark is put in this checkbox, the selected message type is selected for all subsystems.

When you select the checkbox "function exit always", the checkbox "function exit if error" is also selected and greyed out. You can select the checkbox "function exit if error" separately, to limit the number of "function exit" log entries.

- "CM", "DWN", ... "USR" checkboxes. If a check mark is put in this checkbox, all message types for the selected subsystem are selected. The "CM" checkbox is only available in the NYCe 4000 LMS product.
- Checkbox field. If a check mark is put in one (or more) of these checkboxes, the specific message type for the specific subsystem is selected.

If a check mark is put in the "Log events" checkbox at the bottom of the "Logging options" window, all events are logged.

If the "Log AutoScroll" checkbox has a check mark, the message display window of NYCeLogger scrolls "up" one line every time a new data item is logged. The new data item is displayed at the bottom of the message display

window. If the "Log AutoScroll" checkbox does not have a check mark, the message display window is static, that is, new data items are not automatically displayed in the window.



You can use the checkbox "All" to clear all active selections. Click the checkbox twice to select and then deselect all selections.



- NYCeLogger stores the messages in a buffer. The buffer can store 100000 messages. Thus, when message 100000+1 is stored in the buffer, the first message is lost, and so on.
 - If many log data items are written to the buffer and NYCeLogger is used at the same time to display the logged data, log data may be missed because NYCeLogger has performance limitations to retrieve the log data from the buffer and display the data in the window.
-



In Windows 7, all NYCe 4000 tools are executed with "local user" permissions. The "local user" can access the local user's processes and change the log settings via NYCeLogger. However, CM service (of LMS) is executed with "system user" permissions. As a consequence, a "local user" has no permission to change the log settings via NYCeLogger, thus the CM service log settings cannot be changed.

8.3 Miscellaneous

When you click the icon at the left in the title bar, the popup with the standard Windows functions (for example "Move") appears. The last entry of the popup ("About NYCeLogger") displays the "About NYCeLogger" dialog window.

NYCeLogger

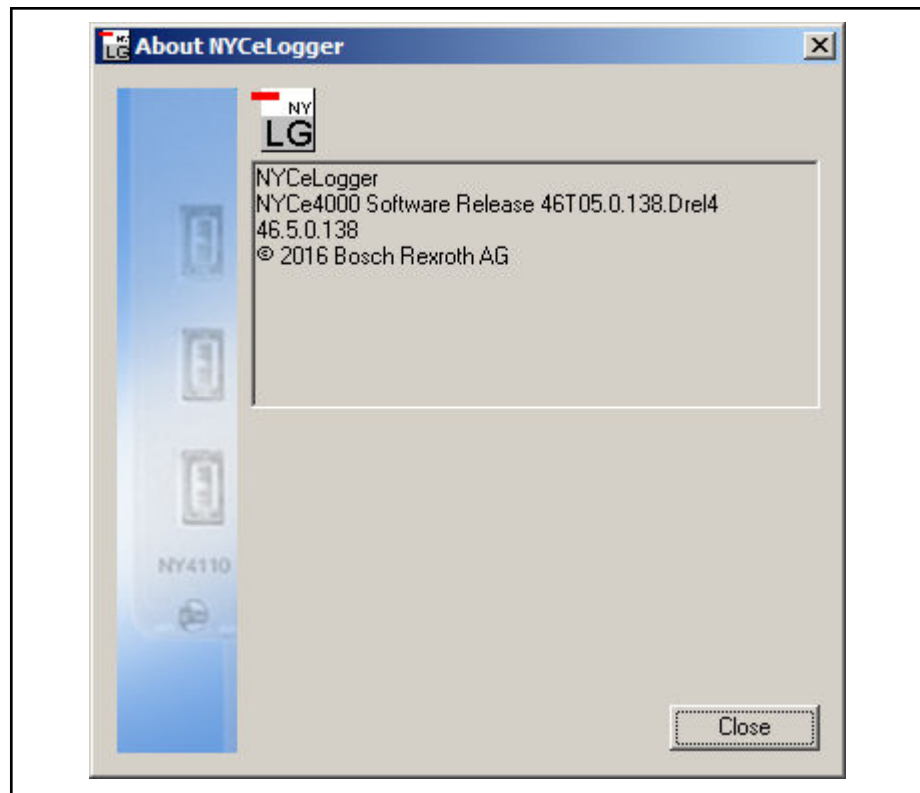


Fig. 8-3: The "About NYCeLogger" dialog window

See [chapter 1.3 "NYCe 4000 product and file version information"](#) on page 5 for an explanation of the product and file version numbers.

9 NYCeWatch

9.1 Introduction

With the NYCeWatch tool you can monitor variables and parameters of the entire NYCe 4000 system. The variables and parameters are grouped in the defined NYCe 4000 subsystems. As far as the variables and parameters are relevant for the user, they are explained in the NYCe 4000 Software User Manual.

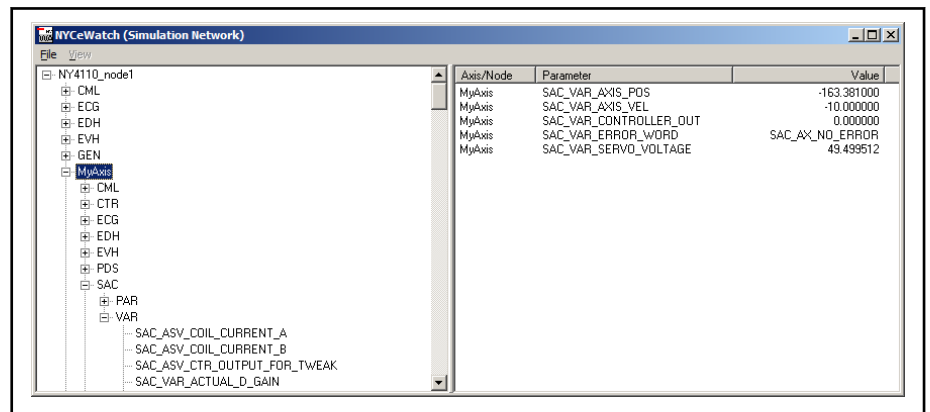


Fig. 9-1: The NYCeWatch window

9.2 Watching variables and parameters

The NYCe 4000 system has many variables and parameters. However, most of these variables and parameters are not relevant during normal operation of the NYCe 4000 system and are for internal use by engineers at Bosch Rexroth.

For the general user, the following subsystems are of interest. See [fig. 9-1 "The NYCeWatch window" on page 227](#).

- On the node level, parameters and variables of the NHI subsystem
- On the axis level, parameters and variables of the SAC subsystem

Do the following steps to get the information of, for example, a SAC variable.

- Select the axis, the SAC subgroup, and the VAR sub-subgroup in the "NYCe4000 System" in the pane at the left hand side of the NYCeWatch window.

- Double-click the variables that you want to watch.

The selected variables are now displayed in the pane at the right hand side.

- A selected parameter can be removed from the pane at the right hand side of the NYCeWatch window by double-clicking the selected parameter with the right mouse button.

- In the View menu entry in the menu bar the format of the variable can be chosen. Select the parameter in the pane at the right hand side, and click "View" in the menu bar. The default is "Dec", but, for example, for the variable SAC_VAR_ERROR_WORD you can select the format "Error Code" to see the value of that variable in a human-readable format, see [fig. 9-2 "Set the displayed data format" on page 228](#).

NYCeWatch

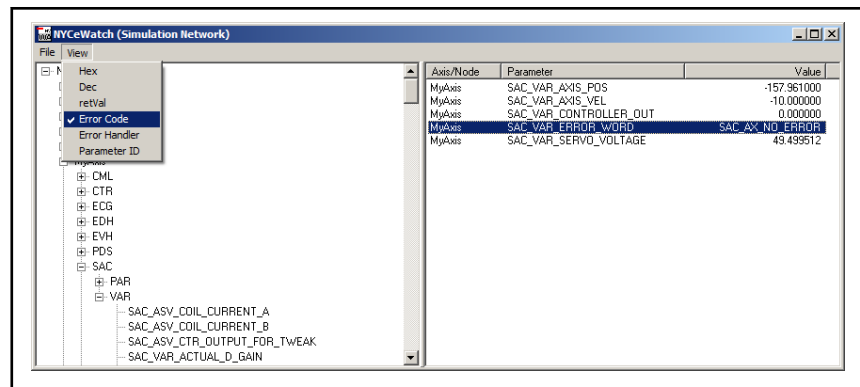


Fig. 9-2: Set the displayed data format

The File menu entry in the menu bar offers the options to save and open the selected configuration (definition of which information displayed by NYCeWatch).

The extension of the NYCeWatch configuration files is ".NYW".

9.3 Miscellaneous

When you click the icon at the left in the title bar, the popup with the standard Windows functions (for example "Move") appears. The last entry of the popup ("About NYCeWatch") displays the "About NYCeWatch" dialog window.

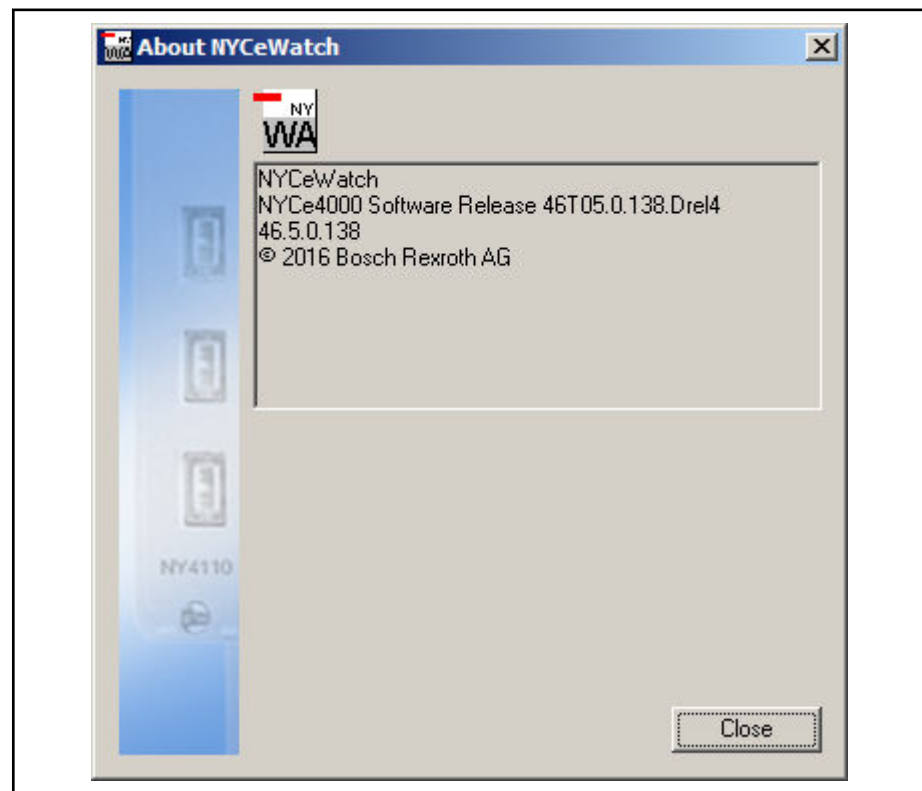


Fig. 9-3: The "About NYCeWatch" dialog window

See [chapter 1.3 "NYCe 4000 product and file version information"](#) on [page 5](#) for an explanation of the product and file version numbers.

10 NYCeNetMonitor

The NYCeNetMonitor tool is used to check the presence and status of all NYCe 4000 nodes in the NYCe 4000 network (real or simulated). Typically, this can also be done by the NYCeConfigurator, but the NYCeNetMonitor tool has some advantages.

- NYCeNetMonitor has a small window which only displays the status of the nodes in the selected network, nothing else. For example, if you have multiple nodes you can directly see and compare the firmware versions of all the nodes, see [fig. 10-2 "The NYCeNetMonitor window" on page 229](#).
- When NYCeNetMonitor is started it only runs in the background. An icon appears in the Windows system tray. If a system event occurs, a node is added or removed, or the network configuration has been changed, a message appears from the icon. For example, if a network cable is disconnected or a node is switched off, you are immediately informed about this event.

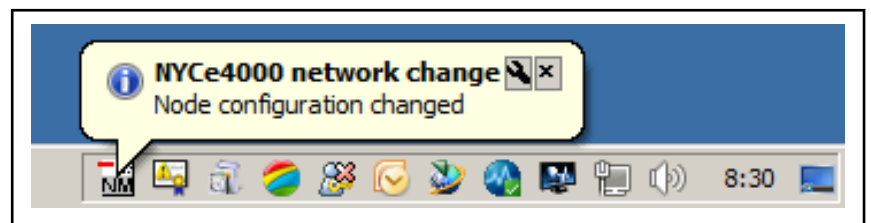


Fig. 10-1: Example of a message from NYCeNetMonitor in the System Tray

After the tool is started, you can display the window of this tool by double-clicking the left mouse button on the icon in the System Tray, or click the right mouse button on the icon and select "Open" in the popup that appears.

When you click the Close button in the title bar, only the window of the tool is removed from the Desktop, the tool remains active in the System Tray.

Click the right mouse button on the icon in the System Tray and select "Exit" in the popup that appears to terminate the tool.

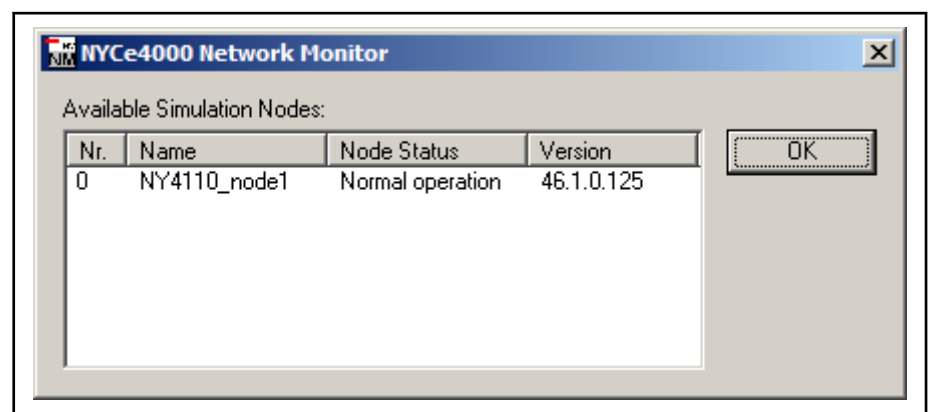


Fig. 10-2: The NYCeNetMonitor window

NYCeNetMonitor



Note that when NYCeNetMonitor reports "Normal Operation" it does not imply that the host software can control the node. After you upgraded a multi-node network, it is possible that the status reported by NYCeConfigurator and NYCeNetMonitor are not identical. NYCeNetMonitor reports "Normal Operation", while NYCeConfigurator may report that a node has the "wrong firmware version". Make sure that the firmware version of all nodes in the network are identical.

11 NYCeUpgradeTool

11.1 Introduction

This chapter gives you information about the NYCeUpgradeTool.

After a new NYCe 4000 software release is installed on the PC, you must also download the new node software to all connected nodes. The node software is the bootloader, gateway files, firmware and microware. The procedure is described in [chapter 2 "Getting started" on page 11](#). However, if the NYCe 4000 "upgrade installation" method for the software installation was available, but you selected to use the "clean installation" method for the software installation on the PC, you can use the NYCeUpgradeTool to download the new node software to all connected nodes. The advantage of the upgrade, if possible, are less actions required to install the new node software. Further, the installation process itself is easier and faster.

The NYCeUpgradeTool is a command-line tool with several parameters to control the actions done by the tool. NYCeUpgradeTool determines whether the "upgrade installation" method is possible. If you select an action that conflicts with the specified parameters, an exit code is returned. The problem description can be found in the log file. Only if the "upgrade installation" is possible, NYCeUpgradeTool continues, else the tool terminates with an exit code, see [chapter 11.5 "Return codes of NYCeUpgradeTool" on page 234](#).

11.2 Preconditions for NYCeUpgradeTool

The following preconditions must be fulfilled for successful operation of the NYCeUpgradeTool.

- The NYCe 4000 Software Release that you installed on the PC must support the "upgrade installation" method. Check the NYCe 4000 Software Release Bulletin whether upgrade is supported for the software version that you want to use with the NYCeUpgradeTool.
- All nodes in the network must be connected to the PC, and must continuously be connected to the 24V System power supply during the upgrade process.
- The firmware in the node(s) must not include a UDC.
- You must guarantee a safe NYCe 4000 machine upgrade condition. Among others, the following conditions must be met before the upgrade is started.
 - All axes of all connected nodes must be in the idle or inactive state.
 - No sequences and no UDSX programs are running on any node.
 - No NYCe 4000 host applications (including the tools) are running.
 - No applications that interact with the NYCe 4000 system are running.
 - All other applications that might interfere are terminated.

NYCeUpgradeTool



- NYCeUpgradeTool is not supported in simulation.
- Downgrade to a previous NYCe 4000 software release is not supported.
- When the NYCeUpgradeTool version matches the installed host software version, all functionality is available. When the versions differ, only the backup functionality is available.
- NYCeUpgradeTool cannot distinguish which version of the microware is installed in the node (4 kHz, 8 kHz or 16 kHz version). NYCeUpgradeTool always installs microware4khz.mcr.

11.3 Usage of NYCeUpgradeTool

The NYCeUpgradeTool can be used to

- backup and restore NYCe 4000 data, such as configuration and node information,
- check whether an upgrade of the node software is possible,
- download the node software (bootloader, gateway files, firmware and microware).

11.4 Parameters of NYCeUpgradeTool

The NYCeUpgradeTool has 3 or 4 mandatory parameters and 1, 2 or 3 optional parameters, depending on the specified action. If one or more parameters is not correct, the tool displays an appropriate help text. The full help text is displayed with the command `NYCeUpgradeTool /?`. The first parameter, "action", specifies the operation that the tool must execute.

NYCeUpgradeTool	/upgradecheck /upgrade	path nrOfNodes [/I] [logPath] logFile
NYCeUpgradeTool	/backup /restore	path nrOfNodes [/I] [/R] [logPath] logFile
NYCeUpgradeTool	/enablesequencebootstrap /issequencebootstrapactive	nrOfNodes [logPath] logFile

Fig. 11-1: NYCeUpgradeTool commandline

Command-line parameter	Parameter value and description
<p>action (mandatory)</p>	<p>Specifies the requested operation of NYCeUpgradeTool.</p> <ul style="list-style-type: none"> • <code>/upgradecheck</code> Check whether nodes can be upgraded. The following checks are done. <ul style="list-style-type: none"> – NYCe 4000 software is installed on the PC. – NYCe 4000 software version can be upgraded. – Check that no sequences are active on the connected nodes. – Check that all axes on the connected nodes are idle or inactive. – Check that the number of detected nodes is equal to the <code>nrOfNodes</code> parameter. – Check that all files, required for the upgrade, are available. • <code>/upgrade</code> Complete upgrade of all connected nodes. The operation upgrades all connected nodes, including a reset of all nodes, and waits until all connected nodes are operational. The parameter <code>path</code> must specify the location of the bootloader, gateway, and firmware files. • <code>/backup</code> Backup the user data. The parameter <code>path</code> must be a valid backup location to store the files. If the optional parameter <code>/R</code> is specified, the current run-time node and axis configuration are saved to the specified file, else the nodes are reset and the flash memory node and axis configuration are saved. • <code>/restore</code> Restore the user data. The parameter <code>path</code> must be a valid backup file location of the files. If the optional parameter <code>/R</code> is specified, only the run-time node and axis configuration are updated, else the nodes are reset and run-time and flash memory are updated. • <code>/enablesequencebootstrap</code> The sequence bootstrap is always disabled when NYCeUpgradeTool starts. With this action the bootstrap sequence can be enabled again. • <code>/issequencebootstrapactive</code> Check whether a sequence bootstrap is active.

NYCeUpgradeTool

Command-line parameter	Parameter value and description
path (mandatory)	Specifies the location of the configuration data if the action parameter is <code>/backup</code> or <code>/restore</code> . Specifies the location of the bootloader, gateway, and firmware files if the parameter is <code>/upgradecheck</code> or <code>/upgrade</code> .
nrOfNodes (mandatory)	The expected number of NYCe 4000 nodes in the system. If the actual number of nodes is not equal to the specified number of nodes, NYCeUpgradeTool terminates with an exit code.
/I (optional)	Optional parameter that indicates whether a check must be executed that sequences are running on the node(s). Default, the check is executed, and if sequences are running, NYCeUpgradeTool terminates. If the parameter <code>/I</code> is specified, running sequences are ignored.
/R (optional)	If the action is <code>/backup</code> , this parameter specifies that the run-time configuration must be saved. If the parameter is not specified, the nodes are reset and the flash memory configuration is saved. If the action is <code>/restore</code> , this parameter specifies that only the run-time configuration data is updated. If the parameter is not specified, the nodes are reset and the run-time configuration data and the flash memory configuration data are updated.
logPath (optional)	Specifies the optional file path for the log file. If <code>logPath</code> is not specified, the log file is stored in the current path.
logFile (mandatory)	Specifies the name of the log file.

Tab. 11-1: NYCeUpgradeTool command-line parameters

11.5 Return codes of NYCeUpgradeTool

When NYCeUpgradeTool terminates, the tool returns an exit code. The following exit codes are possible.

exit code	value	Explanation
EXIT_CODE_OK	0	Normal termination. The requested actions completed successfully.
EXIT_CODE_INVALID_ACTION	1	Invalid action specified on the command-line.
EXIT_CODE_INVALID_PARAMETER	2	Invalid action parameter specified.
EXIT_CODE_WRONG_NR_OF_ARGUMENTS	3	Number of specified arguments not correct.
EXIT_CODE_ACTION_FAILED	4	The requested actions have not completed successfully.
EXIT_CODE_NYCE_INIT_FAILED	5	NyceInit initialization failed.
EXIT_CODE_NYCE_TERM_FAILED	6	NyceTerm termination failed.

NYCeUpgradeTool

exit code	value	Explanation
EXIT_CODE_NOT_UPGRADEABLE	7	Upgrade not possible.
EXIT_CODE_WARNING_COMPATIBILITY	8	Normal termination. Check documentation for application compatibility.
EXIT_CODE_WARNING_NO_SEQUENCE_BOOTSTRAP_ACTIVE	9	Normal termination. No sequence bootstrap active.

Tab. 11-2: NYCeUpgradeTool exit codes

Notes

Bosch Rexroth AG

Electric Drives and Controls

P.O. Box 13 57

97803 Lohr, Germany

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr, Germany

Phone +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911389806