

#### **Measurement & Automation Explorer Help for Motion**

#### February 2007, 370176J-01

You can perform the following motion-related tasks with Measurement & Automation Explorer (MAX):

- View the resources and check the status of motion controllers in your system
- Change the configuration of motion controllers in your system
- Check the firmware version of your motion controllers and update them as necessary
- Initialize your motion controllers
- Test your controllers by interactively making moves.

For more information about this help file, refer to the following topics:

#### <u>Using Help</u>

Related Documentation

Important Information

Technical Support and Professional Services

To comment on National Instruments documentation, refer to the <u>National</u> <u>Instruments Web site</u>.

© 2000–2007 National Instruments Corporation. All rights reserved.

#### **Related Documentation**

Most NI-Motion manuals also are available as PDFs. You must have Adobe Acrobat Reader with Search and Accessibility 5.0.5 or later installed to view the PDFs. Refer to the <u>Adobe Systems Incorporated</u> <u>Web site</u> to download Acrobat Reader. Refer to the <u>National Instruments</u> <u>Product Manuals Library</u> for updated documentation resources.

The following documentation is included with your NI motion controller and the NI-Motion software. Use this information to determine which piece of documentation will be useful for your task.

- *Getting Started with NI-Motion for NI 73xx Motion Controllers* Provides installation instructions and general information about the NI-Motion product.
- NI 73xx User Manuals—Provide information about optional equipment, cabling, and user connectors, including pin assignments.
- NI-Motion User Manual—Contains information about selecting a motor, how to design a basic move, how to design a multi-axis move, incorporating Vision and DAQ into Motion applications, and how to work with common Motion concepts, such as Bode analysis, contouring, and breakpoints. The software user manual is task-based and takes you through each phase of designing and executing a Motion application.
- NI-Motion Readme—Contains system requirements, installation instructions, descriptions of any changes made to the software, information about new features in the release, and information about late-breaking known issues that are not documented in other NI-Motion documents.
- *NI-Motion Function Help*—Contains function reference files for C and Visual Basic and provides details about each function, including a description of the function, a list of the function parameters, illustrations, and error codes.
- *NI-Motion VI Help*—Contains LabVIEW VI reference files and provides details about each VI, including VI descriptions, lists of control and input terminals, usage, illustrations, and error codes.

# **Using Help**

<u>Conventions</u> <u>Navigating Help</u> <u>Searching Help</u> <u>Printing Help File Topics</u>

#### Conventions

# This help file uses the following formatting and typographical conventions:

<>	Angle brackets that contain numbers separated by an ellipsis represent a range of values associated with a bit or signal name—for example, AO <03>.	
»	The » symbol leads you through nested menu items and dialog box options to a final action. The sequence <b>File</b> » <b>Page Setup</b> » <b>Options</b> directs you to pull down the <b>File</b> menu, select the <b>Page Setup</b> item, and select <b>Options</b> from the last dialog box.	
P	This icon denotes a tip, which alerts you to advisory information.	
$\overline{\mathbb{N}}$	This icon denotes a note, which alerts you to important information.	
	This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.	
bold	Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.	
dark red	Text in this color denotes a caution.	
green	Underlined text in this color denotes a link to a help topic, help file, or Web address.	
italic	Italic text denotes variables, emphasis, cross-references, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply.	
monospace	Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.	

# **Navigating Help (Windows Only)**

To navigate this help file, use the **Contents**, **Index**, and **Search** tabs to the left of this window or use the following toolbar buttons located above the tabs:

- Hide—Hides the navigation pane from view.
- Locate—Locates the currently displayed topic in the Contents tab, allowing you to view related topics.
- **Back**—Displays the previously viewed topic.
- **Forward**—Displays the topic you viewed before clicking the **Back** button.
- **Options**—Displays a list of commands and viewing options for the help file.

# Searching Help (Windows Only)

Use the **Search** tab to the left of this window to locate content in this help file. If you want to search for words in a certain order, such as "related documentation," add quotation marks around the search words as shown in the example. Searching for terms on the **Search** tab allows you to quickly locate specific information and information in topics that are not included on the **Contents** tab.

### Wildcards

You also can search using asterisk (\*) or question mark (?) wildcards. Use the asterisk wildcard to return topics that contain a certain string. For example, a search for "prog\*" lists topics that contain the words "program," "programmatically," "progress," and so on.

Use the question mark wildcard as a substitute for a single character in a search term. For example, "?ext" lists topics that contain the words "next," "text," and so on.



**Note** Wildcard searching will not work on Simplified Chinese, Traditional Chinese, Japanese, and Korean systems.

#### **Nested Expressions**

Use nested expressions to combine searches to further refine a search. You can use Boolean expressions and wildcards in a nested expression. For example, "example AND (program OR VI)" lists topics that contain "example program" or "example VI." You cannot nest expressions more than five levels.

# **Boolean Expressions**

Click the **•** button to add Boolean expressions to a search. The following Boolean operators are available:

- AND (default)—Returns topics that contain both search terms. You do not need to specify this operator unless you are using nested expressions.
- **OR**—Returns topics that contain either the first or second term.
- **NOT**—Returns topics that contain the first term without the second term.
- **NEAR**—Returns topics that contain both terms within eight words of each other.

# **Search Options**

Use the following checkboxes on the **Search** tab to customize a search:

- Search previous results—Narrows the results from a search that returned too many topics. You must remove the checkmark from this checkbox to search all topics.
- Match similar words—Broadens a search to return topics that contain words similar to the search terms. For example, a search for "program" lists topics that include the words "programs," "programming," and so on.
- Search titles only—Searches only in the titles of topics.

# **Printing Help File Topics (Windows Only)**

Complete the following steps to print an entire book from the **Contents** tab:

- 1. Right-click the book.
- 2. Select **Print** from the shortcut menu to display the **Print Topics** dialog box.
- 3. Select the **Print the selected heading and all subtopics** option.
  - Note Select Print the selected topic if you want to print the single topic you have selected in the **Contents** tab.
- 4. Click the **OK** button.

#### **Printing PDF Documents**

This help file may contain links to PDF documents. To print PDF documents, click the print button located on the Adobe Acrobat Viewer toolbar.

#### NI 7330 Controllers

The NI 7330 family of controllers are stepper motor controllers for PXI and PCI.

The NI 7330 provides fully programmable motion control for up to four independent or coordinated axes of motion with dedicated motion I/O for limit and home switches and additional I/O for general-purpose functions.

You can use the NI 7330 motion controller for straight-line vector moves and arc moves for stepper motor applications. The NI 7330 controller adds the ability to perform arbitrary and complex motion trajectories using stepper motors.

#### NI 7340 Controllers

The NI 7340 family of controllers are combination servo and stepper motor controllers for PXI, CompactPCI, and PCI bus computers.

The NI 7340 provides fully programmable motion control for up to four independent or coordinated axes of motion with dedicated motion I/O for limit and home switches and additional I/O for general-purpose functions.

You can use the NI 7340 to perform arbitrary and complex motion trajectories using stepper motors or servo devices.

You also can use the NI 7340 motion controller to perform the following advanced operations:

- Circular interpolation
- Contouring
- Electronic gearing
- Onboard programming

#### NI 7350 Controllers

The NI 7350 family of controllers are combination servo and stepper motor controllers for PXI, CompactPCI, and PCI bus computers.

The NI 7350 provides fully programmable motion control for up to eight independent or coordinated axes of motion with dedicated motion I/O for limit and home switches and additional I/O for general-purpose functions.

You can use the NI 7350 to perform arbitrary and complex motion trajectories using stepper motors or servo devices.

You also can use the NI 7350 motion controller to perform the following advanced operations:

- Circular interpolation
- Contouring
- Electronic gearing
- Onboard programming
- Sinusoidal commutation
- Buffered breakpoints
- Periodic breakpoints
- Buffered high-speed capture
- Notch and lowpass filtering

#### NI PCI-7390 Controller

The NI PCI-7390 is a position-command motion controller with isolation designed for use with any drives that support p-command mode or step (pulse) command input.

The PCI-7390 has accessories specifically designed for direct connectivity to Yaskawa Sigma II, Mitsubishi MR-J2S and Panasonic Minas A drives. The PCI-7390 provides fully programmable motion control for up to four independent or coordinated axes of motion, with dedicated motion I/O for limit and home switches and additional I/O for general-purpose functions.

You can use the PCI-7390 controller for point-to-point and straight-line vector moves. The PCI-7390 also performs arbitrary and complex motion trajectories through circular, spherical, or helical interpolation.

#### **NI SoftMotion Controller**

The NI SoftMotion Controller uses standard PC-based platforms and open standards to connect intelligent drives to a real-time host. In this architecture, the software components of the motion controller run on a real-time host and all I/O is implemented on the drives.

This separation of I/O from the motion controller software components helps to lower system cost and improve reliability by improving connectivity. Open standards, such as CANopen, are used to connect these components.

The NI SoftMotion Controller supports CANopen intelligent drives from Copley Controls Corporation.

#### Using MAX to Work with Your NI Motion Controller

Complete the following steps to begin using your NI motion controller:

- 1. Activate your device license.
- 2. If you are using the NI SoftMotion Controller, add an axis for the drive you are using the controller with.
- 3. <u>If you do not have a 73xx controller, but you want to explore the features of Measurement & Automation Explorer (MAX), add a virtual 73xx controller.</u>
- 4. If you are using a 73xx controller, download firmware to it.

#### **Adding Axes to Vector Spaces**

Complete the following steps to add an axis to a vector space:

- 1. Launch Measurement & Automation Explorer (MAX).
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion controller.
- 4. Select **Default** *xxxx* **Settings**.
- 5. Click the **Vector Spaces** tab. MAX displays a list of available vector spaces.
- 6. Right-click the appropriate vector space, select **Add Axis**, and select the axis you want to add. Repeat this step until you have added all axes you want to include in the vector space.

#### **Removing Axes from Vector Spaces**

Complete the following steps to remove an axis from a vector space:

- 1. Launch Measurement & Automation Explorer (MAX).
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion controller.
- 4. Select **Default** *xxxx* **Settings**.
- 5. Click the **Vector Spaces** tab.
- 6. In the appropriate vector space, right-click the axis you want to remove and select **Remove Axis**.

# Adding Axes for the NI SoftMotion Controller

Complete the following steps to add an axis for the NI SoftMotion Controller:

- 1. Launch Measurement & Automation Explorer (MAX).
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**.
- 4. Right-click the **NI SoftMotion Controller** item, and select **Add Axis**.
- 5. Select the device you want to add as an axis, and then click **Next**.
- 6. If desired, double-click in the **Name** column to change the name of the axis. You also can rename the axis in the configuration tree after you add it.
- 7. Click Finish.



**Note** The ID number you assign to the drive you are controlling with the NI SoftMotion Controller must be less than or equal to the number of axes activated.

# **Activating Your Device License**

P

Complete the following steps to activate the license for your motion controller:

- 1. Launch Measurement & Automation Explorer (MAX).
- 2. Expand **Devices and Interfaces**.
- 3. Expand the **NI–Motion Devices** item in the configuration tree.
- 4. Right-click the item for the appropriate motion device and select **Activate Device** to launch the NI Activation Wizard.
- 5. Follow the onscreen prompts to work through the licensing wizard.

**Tip** You also can launch the NI Activation Wizard by selecting the appropriate NI Motion device in MAX and clicking the **Activate Device** button at the top of the MAX user interface.

#### Adding Virtual 73xx Motion Controllers

**Note** This information is not applicable if you are using the NI SoftMotion Controller.

If you do not have an NI 73*xx* motion controller, but you want to explore the features of Measurement & Automation Explorer (MAX), complete the following steps to add a virtual motion controller to your list of devices:

- 1. Install NI-Motion 6.0 or later if you have not already done so.
- 2. Launch MAX.
- 3. Right-click NI Motion Devices and choose Add Virtual Motion Device.
- 4. Select the type of virtual device you want and click **Finish**.

#### Downloading Firmware to Your NI 73xx Motion Controller

National Instruments 73*xx* motion controllers have onboard Flash memory, which is where the NI-Motion firmware is downloaded. Use MAX to update the <u>firmware</u> installed on your motion controller.



**Tip** This information and the procedures listed below do not apply to the NI SoftMotion Controller.

Complete the following steps to update firmware on your motion controller:

- 1. Launch Measurement & Automation Explorer (MAX).
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then select the appropriate device.
- 4. Select the **Firmware** tab.
- 5. Firmware <u>sectors</u> that show the icon are out of date. Click the **Update Firmware** button from the toolbar to update the firmware on your controller.

If the item displays the  $\square$  icon, the sector has the latest version of the firmware installed.

The firmware configuration view displays the latest firmware files installed on your hard drive when you installed the NI-Motion software.



**Tip** For optimal performance, ensure that the firmware on your controller matches the driver version you install. To download all firmware files to an earlier version of the NI-Motion firmware, install the version of NI-Motion that you need and right-click one of the firmware files in the **Firmware Name** column and select **Download All Files**.

#### **Firmware Files**

The following table lists the firmware files and their extensions.

Firmware File	Extension
68331	.cpu
DSP	.rbf
FPGA1	.fp1
FPGA2	.fp2

The firmware files for the motion controllers are different, even though they have the same extensions. All 7344 files have a **7344** prefix, all 7334 files have a **7334** prefix, and so on.

Firmware Sectors Downloading Firmware to Your Motion Controller

#### **Firmware Sectors**

The NI motion controllers have the following firmware sectors:

- **68331 firmware sector**—contains the software, including the realtime OS, required by the 68331 microprocessor
- **DSP firmware sector**—contains the software needed by the DSP chip on the motion controller
- **FPGA1 firmware sector**—contains the code needed to boot the FPGA that implements the encoder functionality on the motion controller
- **FPGA2 firmware sector**—boots the FPGA that implements the digital I/O and RTSI functionality

Firmware Files Downloading Firmware to Your Motion Controller

#### Adding Objects to the Onboard Memory Manager

Objects are user-created items that are stored in the onboard memory of a motion controller. An object can be an onboard program or a buffer.

The configuration view shows only objects that are stored in the onboard memory of a motion controller. MAX automatically detects these objects and displays information about them. To store an object in onboard memory, you must create and execute a program that uses the memory storage functionality of NI-Motion. Refer to the <u>Onboard Programs</u> section of the *NI-Motion Help* and the *NI-Motion VI Help* or *NI-Motion Function Help* for information about the memory management VIs and functions included in NI-Motion.

# **Changing the Board ID for Your Controller**

Complete the following steps to change the board ID for your motion controller:

- 1. Launch Measurement & Automation Explorer (MAX).
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NIMotion Devices**, and then right-click the appropriate device.
- 4. Click **Change Board ID**, and follow the prompts.
- 5. Reboot the motion control system.

If the motion controller is installed in your local machine, reboot that machine. If the motion controller is installed in a remote system, reboot the remote system.

#### **Resetting Motion Controllers**

You can reset your motion controller to clear errors or to get out of the Emergency Shutdown state.

Complete the following steps to reset your motion controller:

- 1. Launch Measurement & Automation Explorer (MAX).
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then select the appropriate motion device.
- 4. Click the **Status** tab.
- 5. Click **Reset Device**.

Refer to <u>Motion Controller Status</u> for the power-up codes and descriptions for each code.

#### **Motion Controller Status**

The **Status** tab in the configuration view displays the status of bits in the communication status register of your motion controller. To access the **Status** tab in Measurement & Automation Explorer, select your device from the configuration tree. The tab appears below the configuration view.

When the motion controller is in the power-up state, the **Move Complete Status** (MCS) register contains a power-up code describing why the controller is in the power-up state. To access this code, execute the Read Move Complete Status VI or function.

Code	Reset Type	Cause
0x80	Bus reset	Normal PC power cycle
0x40	Power-up reset	Normal PC power cycle
0x20	Watchdog timeout	Fatal internal error
0x08	Shutdown	Shutdown (E-Stop) input active (refer to the Enable Shutdown function)
0x02	Software reset	Firmware download

When the controller is not reset and no motor is moving, the status value depends on the total number of axes on your controller. For example, if you have a NI 7340 with 4 axes, the status is 0x1E which indicates that the Move Complete status is True on all axes. Refer to the Read Communication Status function or VI for information about the communication status register.

## **Removing Motion Controllers**

# 73xx Controllers

Complete the following steps to remove a 73xx motion controller:

- 1. Remove the device from the system.
- 2. Launch Measurement & Automation Explorer (MAX).
- 3. Expand **Devices and Interfaces** in the configuration tree.
- 4. Right-click the 73*xx* motion controller you want to remove, and select **Remove Device**.

#### **NI SoftMotion Controller**

Use Add/Remove Programs in the Windows Control Panel to uninstall the NI SoftMotion Controller.

#### **Remote Motion Devices Mapped to the Local Machine**

Complete the following steps to remove a remote motion controller:

- 1. Remove the device from the remote system.
- 2. On the development machine, launch Measurement & Automation Explorer (MAX).
- 3. Expand **Devices and Interfaces** in the configuration tree.
- 4. Right-click the mapped controller you want to remove, and select **Remove Device from Remote System**.
  - Note If you uninstall NI-Motion or the NI SoftMotion Controller from the remote system, such as a PXI chassis, you must explicitly remove the mapped device. To use the device again, remap it after you reinstall NI-Motion or the NI SoftMotion Controller.

#### **Remote Systems and NI-Motion**

For an RT system, you can configure an NI motion controller on a remote PXI chassis through the remote configuration feature of MAX. You must install NI-Motion onto the remote system to use RT. Then, program the RT NI-Motion application exactly the way you would program any other NI-Motion application.

Complete the following steps to install NI-Motion onto the remote system.

- 1. Install one of the following software options onto the host system:
  - NI-Motion (73xx controller support)
  - NI SoftMotion Controller
  - **Tip** Refer to the *Getting Started: NI SoftMotion Controller* for Copley Controls CANopen Drives manual for information about this product.
- 2. Launch MAX.
- 3. Expand the **Remote Systems** tree.
- 4. Highlight the system on which to install NI-Motion.
- 5. Select the **Software** tab.
- 6. If NI-Motion is not already installed, right-click within the dialog box and select **Install Software**. A dialog appears that lets you select what to download. Make sure the checkbox next to NI-Motion RT is selected.
- 7. Click **OK** and wait for the software to download.

After the software downloads onto the remote system, complete the following steps to configure the remote NI motion controller.

- 1. Wait for the remote system to reboot so MAX is able to communicate with it.
- 2. Expand the **Remote Systems** tree and then expand the **Devices and Interfaces** tree.
- 3. Right-click the remote motion controller icon and select **Map to Local Machine**. This assigns a local board ID to the remote motion controller in the host system.

Mapping the remote controller into the local system allows you to configure the controller through MAX exactly as you would a

controller that is in the host system. You can initialize the controller, download firmware, and use the interactive and configuration panels exactly as you would on a controller installed in the host machine. You also can write VIs using the remote motion controller through the local board ID assigned to it.

This allows you to write and debug your VIs on the host, and then download them to the remote system when you are ready. All you need to change is the board ID in your VI from the locally assigned Board ID to the ID assigned by the remote system.

4. Browse to **Devices and Interfaces** under **My System**, where there is a shortcut icon next to a new controller name.

For example, if the motion controller on the remote system is a PXI-7334, and the remote system has an IP address of 123.456.789.000, then the shortcut device would show a name like **PXI-7334 (X) on 10.0.58.48 (Y)**.

*X* is the board ID assigned to the board by the remote system. Use this board ID for VIs that are downloaded to the remote system through LabVIEW RT.

*Y* is the board ID assigned to the remote motion controller by the local system. Use this board ID for any VIs that run on the host and use the remote motion controller.

To remove the mapped motion controller, browse to **My System** under **Device and Interfaces**. Right-click the mapped controller and select **Unmap Remote Device**. You should unmap devices when you no longer need to use them from the host machine.

# **Configuring Controllers for Steppers and Servos**

Before configuring the controller, install the following motion components:

- Motion controller or NI SoftMotion Controller from the appropriate CD
- Motors with encoders
- Encoders (optional for steppers)
- Drives
- Universal Motion Interface (UMI), if necessary
- Limit switches
- Home switches

Refer to the appropriate hardware documentation for installation instructions. For information about configuring and tuning a motion control system, visit <u>ni.com/support/motion</u>.



Note You must have a complete hardware setup to configure and test a servo system.

**Note** National Instruments recommends that you connect the motion controller axis inhibit outputs to the drive. Inhibit outputs are typically used to disable the servo or stepper drive for power savings, safety, or specific application reasons. For more information about inhibit outputs, refer to the *NI-Motion User Manual*.

# Configuring a Controller for a Stepper System or a System with No Motor

- 1. Power on all of the hardware in the motion control system, taking into consideration the order in which you must power on the devices.
- 2. Launch MAX and verify the motion controller firmware is updated.
  - P
- **Tip** This step is not applicable if you are using the NI SoftMotion Controller.
- 3. Select a configuration.
  - a. Expand **Devices and Interfaces** in the configuration tree.
  - b. Expand **NIMotion Devices**, and then select the appropriate device.
  - c. Click **Default** *xxxx* **Settings** to display the configuration options.
  - d. Select Open Loop Stepper.
  - e. If you want to customize any of the device settings, use the configuration tree items under each axis below **Default** *xxxx* **Settings** to view and configure items per axis, or for all axes.
    - To configure all of your axes at once, select the toplevel item for each configuration tree item below the Default xxxx Settings item, and configure the controller settings as appropriate. Click Save to save the customized settings to the MAX database.
    - To configure the axes individually, expand the toplevel item for each configuration tree item below the **Default xxxx Settings** item. Select the axis you want to configure, change its attributes, and click **Save**.

If you do not want to customize the device settings, initialize the controller.

• Axis Configuration—Use the Axis Configuration

tab to control the basic nature of your axis or axes, such as the type of motor or feedback device.

- Motion I/O—Use the Motion I/O tab to configure various aspects of the I/O conducted during the move, such as limits and switches.
- **Trajectory Settings**—Use the **Trajectory Settings** and **Move Constraints** tabs to set up various aspects of the move, such as operation mode, velocity, and following error.
- Find Reference Settings—Use the Find Reference Settings tab to set up home, index, forward, reverse, center, and sequence references for the move.
- **Gearing Settings**—Use the **Gearing Settings** tab if you are gearing an axis to another axis or an encoder.
- Control Loop Settings—Use the Control Loop Settings to set up PID loop parameters and filters.
- Compare & Capture Settings—Use the Compare & Capture Settings tab to set up high-speed captures and breakpoints.
- **Digital I/O Settings**—Use the **Digital I/O Settings** tab to set up port polarities and directions for the controller as it is connected to other devices.
- **ADC Settings**—Use the **ADC Settings** tab to configure the analog inputs on the controller.
- Encoder Settings—Use the Encoder Settings tab to configure the encoder settings for the system.
- **PWM Settings**—Use the **PWM Settings** tab to configure the Pulse Width Modulation (PWM) output settings for the motion controller.
- f. If you customized any of the controller settings, click **Save** on the MAX toolbar to save the changes to the MAX database on the local machine. If you want the changes to apply to all of the axes, click **Save to All Axes** on the MAX

toolbar.

- 4. Click **Initialize** on the MAX toolbar to send the saved controller settings to the controller. Initializing the motion controller resets it to a known starting state so that it is ready for commands.
- 5. Turn the motors by hand to check for holding torque. Motors that hold firmly in place exhibit holding torque, indicating that they are powered and responding to commands from the motion controller.
- 6. Perform a single-axis test.
  - a. Double-click the **Interactive** item in the configuration tree.
  - b. Expand the **1-D Interactive** item in the configuration tree.
  - c. Enter 10000 in **Target Position**.
  - d. Click **Apply** just below **Target Position**.
  - e. Click **Start** on the toolbar. As the motor rotates, **Position** increases to 10,000. **Position** changes to 10,000, regardless of whether a motor is attached.
  - f. Click Kill on the toolbar to allow the motor to rotate freely.
  - g. Turn the motor by hand. Notice that **Position** does not change. Stepper motors operating in open-loop mode cannot read a change in the position of the motor.
  - h. If you have encoders on the stepper motor, continue with the following steps. If you do not have encoders on the stepper motor, you are finished with the device configuration.
- 7. Change the motor type to closed-loop stepper.
  - a. Select **Default** *xxxx* **Settings** in the configuration tree.
  - b. Select the Closed Loop Stepper option button.
  - c. Set the values for **Encoder counts per revolution** and **Stepper steps per revolutions**. Refer to the motor or encoder documentation for the values.
  - d. Click **Save** on the MAX toolbar to save the changes to the MAX database on the local machine.
  - e. Click **Initialize** on the MAX toolbar to initialize the motion controller.
- 8. Perform a closed-loop stepper test.

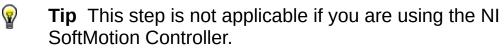
- a. Click **1-D Interactive** in the configuration tree.
- b. Click **Apply** at the bottom of the window.
- c. Click Kill on the toolbar.
- d. Rotate the motor by hand. **Position** changes as you turn the motor.

### Configuring a Controller for a Servo System



**Caution** A servo motor may move beyond a safe limit if it is not properly configured. This situation may occur if the encoder or motor is wired incorrectly or is poorly tuned. To help protect the system from damage, National Instruments recommends that you set the following error, or position error, to a value that will not allow damage to the system. You set the following error in the **Trajectory Settings** tab of the Trajectory Settings configuration item.

- 1. Power on all of the hardware in the motion control system, taking into consideration the order in which you must power on the devices.
- 2. Launch MAX and verify the motion controller firmware is updated.



- 3. Select a configuration.
  - a. Expand **Devices and Interfaces** in the configuration tree.
  - b. Expand **NIMotion Devices**, and then select the appropriate device.
  - c. Click **Default** *xxxx* **Settings** to display the configuration options.
  - d. Select the Servo button.
  - e. Select the commutation type:
    - For a brushed servo motor or brushless servo motor with drive commutation select Servo - External Commutation.
    - For a brushless servo motor select **Servo Onboard Commutation**.
    - Note The following steps assume you are using brushed servo motors. Refer to the *NI-Motion User Manual* for information about configuring brushless servo motor hardware.
  - f. If you want to customize any of the other device settings, use the configuration tree items under each axis below
     Default xxxx Settings to view and configure items per axis, or for all axes.
    - To configure all of your axes at once, select the toplevel item for each configuration tree item below the

**Default** *xxxx* **Settings** item, and configure the controller settings as appropriate. Click **Save** to save the customized settings to the MAX database.

 To configure the axes individually, expand the toplevel item for each configuration tree item below the Default xxxx Settings item. Select the axis you want to configure, change its attributes, and click Save.

If you do not want to customize the device settings, initialize the controller.

- **Tip** Refer to the motor or encoder documentation for the appropriate values to use.
  - Axis Configuration—Use the Axis Configuration tab to control the basic nature of your axis or axes, such as the type of motor or feedback device.
  - Motion I/O—Use the Motion I/O tab to configure various aspects of the I/O conducted during the move, such as limits, switches, and breakpoints.
  - **Trajectory Settings**—Use the **Trajectory Settings** and **Move Constraints** tabs to set up various aspects of the move, such as operation mode, velocity, and following error.
  - Find Reference Settings—Use the Find Reference Settings tab to set up home, index, forward, reverse, center, and sequence references for the move.
  - Gearing Settings—Use the Gearing Settings tab if you are gearing an axis to another axis or an encoder.
  - Control Loop Settings—Use the Control Loop Settings to set up PID loop parameters and filters.
  - Compare & Capture Settings—Use the Compare & Capture Settings tab to set up high-speed

captures and breakpoints.

- **Digital I/O Settings**—Use the **Digital I/O Settings** tab to set up port polarities and directions for the controller as it is connected to other devices.
- **ADC Settings**—Use the **ADC Settings** tab to configure the analog inputs on the controller.
- Encoder Settings—Use the Encoder Settings tab to configure the encoder settings for the system.
- **PWM Settings**—Use the **PWM Settings** tab to configure the Pulse Width Modulation (PWM) output settings for the motion controller.
- 4. If you customized any of the controller settings, click **Save** on the MAX toolbar to save the changes to the MAX database on the local machine. If you want the changes to apply to all of the axes, click **Save to All Axes** on the MAX toolbar.
- 5. Click **Initialize** on the MAX toolbar to send the saved controller settings to the device. Initializing the motion controller resets it to a known starting state so that it is ready for commands.
- 6. Turn the motors by hand to check for holding torque. Motors that are "springy" and return to their original position exhibit holding torque, indicating that they are powered and responding to commands from the motion controller.
- 7. Perform a single-axis test.
  - **Tip** Depending on the servo system, you may need to tune the system before making a move. For information about tuning a servo system, refer to <u>Tuning Servo Motors</u>.
    - a. Double-click the **Interactive** item in the configuration tree.
    - b. Expand the **1-D Interactive** item.
    - c. Enter 10000 in Target Position.
    - d. Click Apply just below Target Position.
    - e. Click **Start** on the toolbar. As the motor rotates, **Position** increases to 10,000.
    - f. Click Kill on the toolbar to allow the motor to rotate freely.
    - g. Turn the motor by hand. **Position** changes as you move

the motor.

# **Configuration Options**

The following sections detail the configuration options available for your motion controller in MAX. This information is also available from within the help pane in MAX. Use the Show Help Show Help button on the toolbar to view the help pane if it is not visible.



 $\label{eq:Note} \textbf{Not all tabs are available for all motion controllers.}$ 

#### **Initialization Preferences**

Use the **Initialization Preferences** tab to configure the settings that are sent to the motion controller when you click **Initialize** or when the controller is initialized programmatically. Some options are always sent to the controller and cannot be disabled. These are the minimum configuration settings required for proper motion controller operation.



**Note** In some cases, clearing the checkbox for one initialization options group also clears the checkbox for another initialization options group and vice versa. For example, **IO Mapping** requires **Digital I/O Settings**, so clearing the **Digital I/O Settings** checkbox also clears the **IO Mapping** checkbox.

Shutdown Input

# **Shutdown Input**

When enabled, the controller goes into a shutdown state when the shutdown signal goes from a low to high transition. The following actions take place in the shutdown state:

- All the axes are killed. On servo axes, the control loop is disabled and the output DACs are zeroed, allowing frictional forces alone to stop the motion. On stepper axes, the stepper pulse generation is stopped. On both axis types, there is no trajectory profile. If enabled, the inhibit output is activated to inhibit (disable) the servo amplifier or stepper driver. You can enable the inhibit outputs and set their polarity as active high or active low with the Configure Inhibit Outputs function.
- All the axes, encoders and ADCs are disabled.
- All the digital I/O is re-initialized to defaults. If the user has saved defaults using the Save Default Parameters function, the digital I/O is re-initialized to the user defaults, otherwise it is re-initialized to the factory defaults.
- All currently running onboard programs are stopped.
- The controller does not accept any functions, except for the following:
  - Get Motion Board Information
  - Read Error Message
  - Enable Auto Start
  - Read Variable

The shutdown functionality is disabled by default on power up. This functionality has to be enabled every time the controller is powered up. You must enable this feature only after the shutdown (E-Stop) circuit is properly configured and connected to the controller. After shutdown (E-Stop) is enabled, it can be disabled only by resetting or power cycling the controller.



**Note** After the controller has shut down, you have to reset it or power cycle it before it can be used again.



**Note** The active state of the shutdown (E-Stop) line is low. To prevent the controller from shutting down prematurely, the line must be high (inactive) when you enable shutdown (E-Stop).

#### **Onboard Memory Manager**

Use the **Onboard Memory Manager** tab to view and manage objects stored in the onboard (FLASH) memory of a motion controller. The size and number of programs is completely flexible. It is ultimately limited by the 32 total memory objects in the Object Registry or by total available memory, whichever is reached first.

Adding Objects to the Onboard Memory Manager

#### **Saving Programs to Onboard Memory**

To save a program to FLASH memory, right-click the program in the configuration view and select **Save to FLASH**.



**Note** When you save a program to FLASH, it copies to FLASH from RAM memory and exists in both. However, your program appears in the configuration view as RAM until you free the program from RAM. Refer to the *Deleting Onboard Programs* section in this topic for information about freeing RAM.

# **Running Onboard Programs**

To run a program stored onboard, right-click the program in the configuration view and select **Run Program**.

#### **Deleting Onboard Programs**

To delete a program from FLASH (ROM), right-click the program in the configuration tree and select **Delete from FLASH**.

To delete a program from RAM, right-click the program in the configuration view and select **Free from RAM**.

# **Axis Configuration**

Use the **Axis Configuration** tab to control the basic nature of your axis or axes, such as the type of motor it controls.

#### Enabling and Disabling Axes



**Note** NI 7350 controllers allow you to choose between **External Commutation** for brushed servo motors or brushless servo motors with drive commutation and **Onboard Commutation** for brushless servo motors without drive commutation. Selecting **Onboard Commutation** enables the <u>Sinusoidal Commutation Settings</u> tab.



**Tip** Refer to <u>Resource Allocation</u> for information about using multiple resources on NI motion controllers.

# **Sinusoidal Commutation Settings**

Use the **Sinusoidal Commutation** tab to control the commutation settings of your servo axis or axes.

To configure the phase initialization, configure the type of <u>Hall Effect</u> <u>Sensor</u> to use or set the axis to <u>Shake and Wake</u> or **Direct Set**, and set the <u>counts per electrical cycle</u>.



**Note** When you are configuring sinusoidal commutation in MAX, sinusoidal commutation is available only on axes 1, 2, 5, and 6 of your 7350 motion controller.



**Tip** Refer to <u>Resource Allocation</u> for information about using multiple resources on NI motion controllers.

#### Hall Effect Sensors

If Hall Effect sensors are available in the system, the controller can estimate the commutation phase based on the state of the Hall effect sensors. After a Hall state transition, the controller calculates the phase angle based on the transition location.

To obtain the maximum torque, perform a move that is 1/6th of the encoder count per electrical cycle after system initialization. Refer to the *Connecting Brushless Servo Motors* section of Chapter 2, *Configuration and Installation*, in the *NI 7350 User Manual* for Hall effect sensor connection information.

#### Shake and Wake

During **Shake and Wake** initialization, the controller outputs a specified voltage for a specified duration to drive the system to the zero-degree phase position and establish it. During this process, the motor moves with very high torque. Make sure the system is away from any limits when performing **Shake and Wake** initialization.

If your system has a heavy load or if the motor is moving against gravity, increase the **Shake and Wake** voltage. If there is heavy jitter as an axis approaches zero, increase the time period.



**Caution** Phase initialization is performed when the axes are enabled. If you use **Shake and Wake**, the system can move during the initialization process.

## **Counts Per Electrical Cycle**

Electrical cycle is the command signal cycle that commutates the motor. To output in the correct phase, the controller must know the encoder counts per electrical cycle. This information is usually listed in the motor or stage documentation. Some motor manufacturers present this information as the number of poles and refer to electrical cycle as magnetic cycle. In such cases, convert number of poles to counts per electrical cycle with the following equation.

encoder counts per electrical cycle =  $2 \times \frac{\text{encoder counts per revolution}}{\text{number of poles}}$ 



**Caution** Counts per electrical cycle must be set correctly to avoid overheating and damaging your motor.

#### **Stepper Settings**

Use the **Stepper Settings** tab to configure settings for stepper motors.



**Tip** Refer to <u>Resource Allocation</u> for information about using multiple resources on NI motion controllers.

### Stepper steps per revolution

Steps can be full steps, half steps, or microsteps depending upon how you have the external stepper driver and motor configured.

Closed-loop stepper functionality relies on the ratio of counts to steps and not on the absolute values of counts or steps per revolution. For closedloop operation, any unit of measure (UOM) that allows you to enter both counts per UOM and steps per UOM that are within their valid ranges work.



**Caution** For closed-loop stepper controllers, steps per revolution/counts per revolution must be in the range of 1/32,767 < steps/counts < 32,767.

# **Stepper Loop Mode**

You can operate stepper axes in both open-loop and closed-loop modes. In open-loop mode, the stepper axis controls the trajectory profile and generates steps but has no feedback from the motor or actuator to determine if the profile is followed correctly.

In closed-loop mode, the feedback position is constantly compared to the number of steps generated to see if the stepper motor is moving correctly. When the trajectory profile is complete, missing steps (if any) are made up with a pull-in move. If, at any time during the move, the difference between the instantaneous commanded position and the feedback position exceeds the programmed following error threshold, the axis is killed and motion stops.

You use p-command mode for systems in which a servo drive receives step/dir (CW/CCW) from the motion controller. The drive closes the PID loop, and the motion controller provides the trajectory generation. Any time you use the axis in p-command mode and the axis goes from a kill state to a halt state, it is recommended to reset the axis position to synchronize the encoder and the step generation with the drive. Like in closed-loop mode, in p-command mode the axis trips on following error.



Note P-command mode does not use pull-in moves.



**Caution** For proper closed-loop and p-command operation, steps per revolution/counts per revolution must be in the range of 1/32,767 < steps/counts < 32,767. An incorrect counts to steps ratio can result in failure to reach the target position and erroneous closed-loop stepper operation.

To operate in closed-loop or p-command mode a stepper axis must have a primary feedback resource mapped to it prior to enabling the axis. Set these values on the **Axis Configuration** tab. You can operate an axis with a primary feedback resource in open-loop, closed-loop, or pcommand mode and you can switch the mode at any time. You can still read the position of the mapped feedback resource even when the axis is in open-loop mode.

# **Stepper Polarity**

Set stepper active polarity for active low or active high. For example, if the axis is configured for Step & Direction output mode, the stepper polarity determines whether a high direction output is forward or reverse. The polarity also determines the resting states of outputs when they are not pulsing. When configured as active low, the output is active when there is a low signal on the output pin. Conversely, active high means that the output is active when there is a high signal on the output pin.

You may need to change this setting so that the controller output matches the expected drive physical input.

#### Set Stepper Output Mode

NI-Motion supports the two industry standards for stepper control outputs. The most popular mode is **Step & Direction**, where one output produces the step pulses and the other output produces a direction signal.

In **Clockwise/Counterclockwise** (CW/CCW) mode, the first output produces pulses when moving forward, or CW, while the second output produces pulses when moving reverse, or CCW.

For the 7350 controller, you can set the stepper output Drive Mode to **Open Collector** or **Totem Pole**. On the 7330/40, stepper output is Open Collector and cannot be changed. On the 7390, stepper output is isolated so Drive Mode does not apply and the option is not selected and cannot be changed. For more information about Open Collector and Totem Pole drive modes, refer to the documentation for your 7350 device.

# **Enabling and Disabling Axes**

Only enabled axes are updated and there is a direct correspondence between the number of enabled axes and the fastest update rate allowed.

Number of Enabled Axes (7330/40/90)	Fastest Update Rate	Number of Enabled Axes (7350)	Fastest Update Rate	Number of Enabled Axes (NI SoftMotion Controller)	Fastest Host Loop Update Rate
1	62 µs	1–2	62 µs	1–8	10 ms
2	125 µs	3–4	125 µs	9–15	20 ms
3	188 µs	5–6	188 µs	16–23	30 ms
4	250 µs	7–8	250 µs	24–30	40 ms

The fastest update rate is achievable only when all axes are single-feedback servo axes and no extra encoders are enabled.



**Caution** Update rates that are too fast for the number of axes, stepper outputs, and/or encoders enabled generates an error and the previous setting remains in effect.

The Enable Axis function automatically enables the feedback devices mapped to the enabled axes. It is not necessary to explicitly enable the encoders or ADC channels before enabling the axes. For NI 73*xx* devices, axes must be killed before they are disabled or the error NIMC\_wrongModeError is generated. For the NI SoftMotion Controller it is not necessary to kill, or deactivate, an axis before disabling it.



**Tip** Refer to <u>Resource Allocation</u> for information about using multiple resources on NI motion controllers.

#### Motion I/O

Use the **Motion I/O** tab to configure the home switches, limit switches, software limits, and inhibits on your motion system. You also can set the active state of your inputs and outputs to active low/active open or active high/active closed.

Home & Limit Switches Software Limit Settings Inhibit Output Settings Drive Ready Signal Inhibit Input In-Position Input

#### Home & Limit Switches

A home switch is a physical switch placed somewhere within the range of travel of the motion system. You can search for the home switch to find a known reference location for your motion application.

A limit switch signals the ends-of-travel on a motion system. You can connect limit switches to the controller to protect the motion system from physical damage by indicating a hard limit of travel.

# **Software Limit Settings**

An enabled software limit causes the axis to smoothly decelerate to a stop when the limit position is reached or exceeded.

### **Inhibit Output Settings**

Inhibit outputs are typically connected to the Drive Enable signal on the servo or stepper drive to control the enable/inhibit function of the drive. This signal is used to disable the servo amplifier or stepper driver for power savings, safety, or other specific application reasons. You also can use inhibit outputs as general-purpose outputs.

When enabled, a per-axis inhibit output is linked to the motor off state of the corresponding axis. A killed axis (motor off) forces the corresponding inhibit output on. When the axis is active, the inhibit output is off.



**Note** Totem Pole configuration for inhibit outputs is only available on NI 7350 controllers. Totem pole configures the inhibit output to sink and source current. Refer to the *NI 7350 Hardware User Manual* for more information. The output mode is not configurable on NI 7330/40/90 motion controllers.

# **Drive Ready Signal**

The drive ready input can be connected to the drive ready or servo ready output of the drive. The drive ready output is active during normal operation. If the drive ready input signal is inactive, calling Start Motion returns an error. All other VIs can be executed.



**Tip** Check the status of this input at power-on to verify that the controller is ready to start a move.

### **Inhibit Input**

Use the inhibit input to connect a drive alarm/servo alarm signal to the motion controller so that when an alarm or other drive fault occurs, the motion is stopped using a kill stop. A kill stop asserts inhibit-out, disabling the control loop and zeroing the DAC so that frictional forces stop the motion.

#### **In-Position Input**

Use the **In-Position** drive signal when the drive is closing the position loop, for example a servo drive that accepts p-command or digital signals. This signal tells the motion controller when the drive considers the motor to be at the commanded position.

When the In-Position drive signal is configured, the Move Complete status is linked to the state of the in-position input. The move is not considered complete unless the in-position input is active.



**Note** The in-position signal is only available in MAX for NI PCI-7390 motion controllers. For NI 7330/7340/7350 controllers, use the *Configure Motion I/O Map* VI or function to map the in-position input to a general-purpose I/O line and the *Write Trajectory Data* VI or function to add this signal to the move complete criteria.

#### **Trajectory Settings**

Use the **Trajectory Settings** tab to configure the settings for the move trajectory. Refer to the following topics for more information about each configuration option.

Move Status Settings

Move Complete Criteria

Velocity Filter Settings

# **Move Status Settings**

The following sections describe the Move Status Settings configuration options.

### **Following Error**

Defines the following error trip point in counts. If the following error exceeds this value, the axis is automatically killed. Loading zero (0) is a special case that disables the following error trip functionality.

# **Velocity Threshold**

When the velocity exceeds the threshold value, the **Velocity Threshold** status is **True**. Increasing the **Filter Time** reduces quantization noise in the threshold status but at the expense of increasing threshold status latency. Velocity threshold is typically used to monitor the acceleration and deceleration trajectory periods to see when or if an axis is up to speed.

# **Run/Stop Threshold**

Defines the velocity threshold above which an axis is considered running. The run/stop threshold has units of counts/sample period for servo axes or steps/sample period for open-loop and closed-loop stepper axes and is affected by the PID **Update Period**. If the velocity of an axis is above the specified step/count per sample period, the run/stop status is true. If it is below the threshold, the run/stop status is false.

# **Move Complete Criteria**

The effect of the move complete criteria parameters can be summarized with the following equation:

```
Move Complete = (Profile Complete [OR Motor Off])
[AND (Run/Stop == Stop)]
[AND (In Position == Active) ]
[AND (Delay == Done)]
```

```
[AND (| position - target position | < Deadband)]
```

where [...] = optional criteria.

If you enable the **Motor Off** checkbox, any condition that causes the axis to turn its motor off (a kill or exceeding the following error limit) satisfies this basic requirement for Move Complete. In other words, either Profile Complete OR Motor Off must be True for Move Complete to be True.

The next three criteria, Run/Stop, In Position, and Delay, are optional conditions that are logically ANDed to further qualify the Move Complete status. If the **Run/Stop** checkbox is enabled, the axis must also be logically stopped for the move to be considered complete. This value is set with using **Run/Stop Threshold** in the Move Status Settings.

If the **In Position** checkbox is enabled, the in-position input signal is used as a criteria for move complete, and the move is only complete if the inposition signal is active.

 $\overline{\mathbb{N}}$ 

**Note** The in-position signal can only be added through MAX for NI PCI-7390 motion controllers. For NI 7330/7340/7350 controllers, use the *Configure Motion I/O Map* VI or function to map the in-position input to a general-purpose I/O line and the *Write Trajectory Data* VI or function to add this signal to the move complete criteria.

If the **Delay** checkbox is enabled, the axis must wait a user-defined delay after the other criteria are met before the move is considered complete. The user-defined delay is typically used to wait the mechanical settling time so that a move is not considered complete until vibrations in the mechanical system have damped out. It also can be used to compensate for PID pull-in time due to the integral term. This pull-in is typically at velocities below the Run/Stop threshold.

Finally, if the **Deadband** checkbox is enabled, the axis checks its final stopping position versus its target position and only sets the Move Complete status if the absolute value of the difference is less than the

configured distance.

A non-zero value for **Minimum Pulse** guarantees the move complete status stays in the True state for at least this minimum time, even if another move starts immediately. You can use this feature to make sure the host does not miss a Move Complete status when checking for move complete status.

### **Velocity Filter Settings**

The Velocity Filter Settings options are used to configure the parameters for the velocity filter. Because the raw velocity data can change because of noise, jitter, or feedback quantization noise, a velocity filter is applied automatically to the reported velocity data. The velocity filter calculates an average velocity based on a specific time interval or travel distance, whichever comes first. In other words, when the axis is moving slowly the average velocity is calculated using the specified time interval and when the axis is moving fast enough, average velocity is calculated using the specified travel distance. Changing these values changes the time/distance used for calculating the velocity data. Also, you can disable the Filter Distance by entering 0. Entering 0 for Filter Time disables the velocity filter and allows you to read the raw velocity data.

### **Move Constraints**

Use the **Move Constraints** tab to configure various settings for the move, such as velocity and acceleration. Set the units for velocity, acceleration, and deceleration on the **Trajectory Settings** tab.

#### S Curve Time



**Tip** Refer to the <u>Trajectory Parameters</u> section of the *NI-Motion Help* for information about minimum and maximum allowable move constraint values.

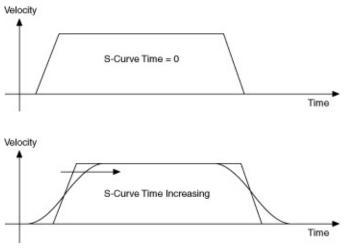
# S Curve Time

Adjusting the s-curve time smooths the acceleration and deceleration portions of a motion profile, resulting in less abrupt transitions from starting a move to acceleration, acceleration to constant velocity, constant velocity to deceleration, and deceleration to stop. Using s-curve acceleration limits the jerk in a motion control system.

Officially, jerk is defined as the derivative of acceleration (change of

acceleration per unit time) and is measured in units of counts (steps)/s<sup>3</sup>. This function, however, allows you to load s-curve time in update sample periods rather than have to deal with the obscure units of jerk.

With the default s-curve time of one (1) sample period, there is virtually no effect on the motion profile, and the standard trapezoidal trajectory is executed. As s-curve time increases, the smoothing affect on the acceleration and deceleration portions of the motion profile increase, as shown in the following figure. Large values of s-curve time can override the programmed values of acceleration and deceleration by sufficiently smoothing the profile such that the acceleration and deceleration slopes are never reached.



R

**Note** A large s-curve values makes the time to accelerate to the commanded velocity longer. Therefore, the time it takes to reach the target position is also longer. If the s-curve value is too large, the value is ignored and full acceleration is used. This happens when the acceleration and deceleration become too long to perform smoothly.

### **Find Reference Settings**

Use the tabs to configure the settings for subsequent reference moves for finding the home, index, forward or reverse limit switch, or the center of travel. Each tab has configuration options for the different reference move types.

# Smart Enable

The table below lists how the switches are enabled for each type of reference move when the **Smart Enable** checkbox is checked on each tab.

Switch Enabled	Home Switch	Forward Limit	<b>Reverse Limit</b>
Home	enabled	enabled	enabled
Index	disabled	enabled	enabled
Forward	disabled	enabled	no change
Reverse	disabled	no change	enabled
Center	disabled	enabled	enabled

### **Find Reference Index**

Use the **Find Reference Index** tab to configure the settings for subsequent reference moves for finding the index pulse from an encoder.

You can configure the initial search direction and the approach velocity, create an offset move, reset the position after the move, and determine the criteria for index pulse.

Refer to <u>Find Index</u> in the *NI-Motion Help* for more information about Find Index Reference Moves.

### **Find Reference Reverse Limit**

Use the **Find Reference Reverse Limit** tab to configure the settings for subsequent reference moves for finding the reverse limit of the range of travel of your axis or axes.

You can configure the initial search direction, final approach direction, and the approach velocity. You also can set an offset move that happens after the home is found, and whether the controller resets the position after the move.

Refer to <u>Find Forward and Reverse Limits</u> in the *NI-Motion Help* for more information about Find Reverse Limit Reference Moves.

### **Find Reference Forward Limit**

Use the **Find Reference Forward Limit** tab to configure the settings for subsequent reference moves for finding the forward limit of the range of travel of your axis or axes.

You can configure the initial search direction, final approach direction, and the approach velocity. You also can set an offset move that happens after the home is found, and whether the controller resets the position after the move.

Refer to <u>Find Forward and Reverse Limits</u> in the *NI-Motion Help* for more information about Find Forward Limit Reference Moves.

### **Find Reference Home**

Use the **Find Reference Home** tab to configure the settings for subsequent reference moves for finding the home switch. You can configure the initial search direction, final approach direction, the edge of the switch to stop on, and the approach velocity.

You also can set an offset move to occur after the home is found, and to determine if the controller resets the position after the move.

Refer to <u>Find Home</u> in the *NI-Motion Help* for more information about Find Home Reference Moves.

### **Find Reference Center**

Use the **Find Reference Center** tab to configure the settings for subsequent reference moves for finding the center of the range of travel of your axis or axes.

You can configure the initial search direction, final approach direction, and the approach velocity. You also can set an offset move that happens after the home is found, and whether the controller resets the position after the move.

Refer to <u>Find Center</u> in the *NI-Motion Help* for more information about Find Center Reference Moves.

# **Gearing Settings**

Use the **Gearing Settings** tab to configure the gearing settings for master and slave axes.

You can <u>configure the gearing master</u>, enable or disable gearing, gearing slave or slaves, <u>set the gearing mode</u>, and <u>configure ratio</u> of revolutions of the slave axis or axes per revolution of the master axis.



**Note** National Instruments recommends disabling gearing before resetting the position of the master axis or the slave axis. When the position of either the master axis or the slave axis is reset, the reset offset is treated as gearing offset, and the slave axis jumps to the new gear position in one PID period.



**Note** If you are running a slave axis move with a stepper axis, it is possible that the slave axis geared velocity exceeds the maximum step rate of the controller. In this case, the controller kills the axis and sets the following error status to true. Refer to the *Specifications* section of your controller user manual for the maximum step rate of your controller.

### **NI SoftMotion Considerations**

If you are using only one axis with the NI SoftMotion Controller for the Copley Controls CANopen Accelnet or Stepnet drives, this tab is not applicable because there is no other axis to gear to.

If you are using only one axis with the NI SoftMotion Controller for the Copley Controls CANopen Xenus drive, you can use this tab to configure the axis to gear to the ADC on the axis.

### **Gear Ratio**

Changing the absolute gear ratio causes the slave axes to jump to a new position specified by the new gear ratio. With servo motor systems the slave jumps immediately. With stepper systems, if the new slave position is within 4,000 steps, it jumps to the new position. Otherwise, the slave axis waits until the commanded slave position moves to within 4,000 steps and then jumps to the new position. The slave axis then resumes following the master axis according to the gear ratio.



**Note** If you are running a slave axis move with a stepper axis, it is possible that the slave axis geared velocity exceeds the maximum step rate of the controller. In this case, the controller kills the axis and sets the following error status to true. Refer to the *Specifications* section of your controller user manual for the maximum step rate of your controller.

# **Gearing Mode**

When you start gearing, the positions of the slave and its master are recorded as their absolute gearing reference. From then on, as long as the gear ratio remains absolute, every incremental change of the master position is multiplied by the absolute gear ratio and applied to the slave axis.



**Note** While changing an absolute gear ratio on the fly is allowed, you must be careful because the slave axis jumps with full torque to the position defined by the new ratio even when the master position has not changed. Similarly, resetting the position on a master axis causes its slave axes to jump to a new position relative to the new position of the master axis.

If a relative gear ratio is selected and loaded after gearing is enabled, the position of the master is recorded as its relative reference point and every incremental change from this reference point is multiplied by the relative gear ratio and applied to the slave axis or axis.

Relative gearing changes the position of the slave axis only when the master moves. It uses the length of the current move of the master axis to calculate how far to move the slave axis. At the end of the move, if the slave axis is not at the exact position, the position error is discarded. This may happen when the true target for the slave axis is a fractional position, which is impossible to actually reach. The position error can add up over time to become significant. If it become a problem, call this function with absolute gearing selected to correct for the accumulated error. The controller calculates where the slave is supposed to be relative to the offset of the master axis beginning from the position where gearing was enabled. With gearing ratios less than 1, this problem does not appear.

### **Gear Master**

When an axis is assigned as the master, its trajectory generator output (not its feedback position) is used as the master position command. This produces synchronized movements without taking the master's following error into account. This mode of operation can eliminate the following error skew between the master and slave axes and is especially useful in gantry applications. The master axis can operate in any mode (including being a slave to another master).

Selecting **Encoder** sets the axis as a slave to the master's encoder. This produces a following affect and takes all following error jitter and behavior of the master into account.

Selecting **ADC** is similar to selecting **Encoder**, but assumes that the master is using ADC feedback rather than encoder feedback.

# **Control Loop**

Use the **Control Loop** tab to configure the control loop used to handle feedback in a servo motion system.

Refer to the <u>Tuning Servo Systems</u> section of the <u>NI-Motion Help</u> and the <u>help associated with each control</u> for more information about configuring a control loop.

Any changes you make on the main **Control Loop Settings** item affect each control loop subitem in the list, if available. In addition, values on the main tab display the **Control Loop 1** settings only.



**Note** Clicking **Save** on the main **Control Loop Settings** item when you have not made any changes does not override any changed settings on the **Control Loop 1** or **Control Loop 2** subitems.

# **Control Loop Parameters**

The following section contains detailed information about each control loop setting option.

- Proportional Gain (Kp)
- Derivative Gain (Kd)
- Integral Gain (Ki)
- Velocity Feedback Gain (Kv)
- Derivative Sampling Period (Td)
- Integration Limit (ILim)
- Velocity Feedforward (Vff)
- Acceleration Feedforward (Aff)
- <u>Control Loop Update Period</u>

# **Proportional Gain (Kp)**

Determines the contribution of restoring force that is directly proportional to the position error. This restoring force functions in much the same way as a spring in a mechanical system.

For each sample period, the PID loop calculates the position error (the difference between the instantaneous trajectory position and the primary feedback position) and multiplies it by Kp to produce the proportional component of the 16-bit DAC command output.

The formula for calculating this proportional contribution is as follows:

An axis with zero or too small a value for Kp will not be able to hold the axis in position and will be very soft. Increasing Kp stiffens the axis and improves its disturbance torque rejection. However, too large a value of Kp can cause instability.

# **Derivative Gain (Kd)**

Determines the contribution of restoring force proportional to the rate of change (derivative) of position error. This force acts much like viscous damping in a damped spring and mass mechanical system (for example, a shock absorber).

The PID loop computes the derivative of position error every <u>derivative</u> <u>sampling period</u>. This derivative term is multiplied by Kd every PID sample period to produce the derivative component of 16-bit DAC command output.

The formula for calculating the derivative contribution is as follows:

where the time between t1 and t0 is the derivative sample period.

A non-zero value for Kd is required for all systems that use torque block amplifiers (where the command output is proportional to motor torque) for the servo loop operation to be stable. Too small a Kd value will result in servo loop instability.

With velocity block amplifiers (where the command output is proportional to motor velocity), you typically set Kd to zero or to a very small value.

# Integral Gain (Ki)

Determines the contribution of restoring force that increases with time and thus ensures that the static position error in the servo loop is forced to zero. This restoring force works against constant torque loads to help achieve zero position error when the axis is stopped.

Each sample period, the position error is added to the accumulation of previous position errors to form an integration sum. This integration sum is scaled by dividing by 256 prior to being multiplied by Ki. Therefore, the formula for calculating the integral contribution in the 16-bit DAC command output is as follows:

where LIMIT = shorthand for the effects of the integration limit described in the following sections.



**Note** The scaling by 1/256 allows the use of integer values for the integral gain even when only a small amount of integral contribution is required.

In applications with small static torque loads, this value can be left at its default value of zero (0). For systems having high static torque loads, this value must be tuned to minimize position error when the axis is stopped.

Non-zero values of Ki, while reducing static position error, tend to increase position error while accelerating and decelerating. This effect can be mitigated through the use of the Integration Limit parameter. Too high a value of Ki often results in servo loop instability. For these reasons, it is recommended that Ki be left at its default value of zero until the servo system operation is stable and then you can add a small amount of Ki to minimize static position errors.



Note Ki has no effect when Ilim is equal to zero.

# Velocity Feedback Gain (Kv)

Determines the contribution of restoring force (damping) proportional to the velocity of the motor/actuator. You can use a primary or secondary feedback encoder for velocity feedback. Setting velocity feedback gain (Kv) to a value other than zero (0) enables velocity feedback using the secondary encoder, if configured, or the primary encoder if a secondary encoder is not configured.

Kv is used to scale this velocity feedback before it is added to the other components in the 16-bit DAC command output. Kv is similar to derivative gain (Kd) except that it scales the velocity estimated from encoder resources only. The derivative gain scales the derivative of the position error, which is the difference between the instantaneous trajectory position and the primary feedback position. Like the Kd term, the velocity feedback derivative is calculated every <u>derivative sampling</u> <u>period</u> and the contribution is updated every PID sample period.

The formula for calculating the velocity feedback contribution is as follows:

where the time between t1 and t0 is the derivative sample period.

Velocity feedback is estimated through a combination of speed dependent algorithms. At high speeds, velocity is simply the change in position per sample. At low speeds, the estimator seamlessly transitions to a 1/T method that measures the time between encoder counts and then calculates the inverse. This method is used for smoother performance when estimating velocities less than one encoder count per sample derivative sample period.

Using Kv and a secondary feedback encoder creates a minor velocity feedback loop. This is very similar to the traditional analog servo control method using a tachometer and a velocity block amplifier and is commonly referred to as dual-loop feedback. Dual-loop feedback is most useful when the primary position sensor (encoder or analog transducer) is located on the end-effector for improved accuracy, and is separated from the motor by gears, ballscrews, belt drives, and/or other mechanical apparatus with potentially poor dynamics. In this case, it can be difficult to achieve a high performance, stable control system without using the minor loop velocity feedback from an encoder mounted directly on the back of the motor.

Typically, Kd is set to zero when Kv is used. However, NI-Motion allows you to use both Kv and Kd terms simultaneously for improved performance.



**Note** Operating with zero derivative gain (Kd) and either velocity feedback or a velocity block amplifier is often referred to as PIVff mode.

You can operate NI-Motion in PID mode, PIVff mode, or in a combination of both modes, by using Kd, Kv, or both.

# **Derivative Sampling Period (Td)**

Determines how often (in update samples) the derivative of position error is calculated.

The formula for calculating the derivative sample period from Td is as follows:

Derivative Sample Period = (Td + 1) x PID Sample Period

Because the range for Td is 0 to 63, the shortest derivative sample period is as follows:

Derivative Sample Period =  $1 \times 62.5 \ \mu s = 62.5 \ \mu s$ 

The longest derivative sample period is as follows:

Derivative Sample Period =  $64 \times 500 \ \mu s = 32 \ ms$ 

Adjusting Td provides greater flexibility in tuning the PID loop derivative term. As Td is increased, you can use a proportionally lower value of Kd for similar results. You must start the Td parameter at its default value of 2 and make small adjustments as required by the motion system configuration.

For low inertia systems, Td must be set to a small value (0 or 1) so that the derivative is calculated often enough to provide adequate damping for servo loop stability.

Systems with higher inertia can benefit from larger values of Td. Because the higher inertia means that the position error cannot change quickly, it is acceptable to calculate the derivative less often. This means you can use a lower value of Kd, have the same effective amount of damping and the system is smoother with less torque noise from the derivative term.

In higher inertia systems, using a Td of zero and therefore a larger value for Kd results in increased torque noise and motor heating without any improvement in system stability.

## Integration Limit (ILim)

Sets an upper bound for the contribution of the integral term in the PID loop. The scaled integration sum is compared to the integration limit and the lesser of the two values is multiplied by Ki to produce the integral term of the control output. This limiting function is indicated with LIMIT() in the following integral term output equation:

You can use Ilim to limit excessive restoring forces and to minimize the adverse effects that integral compensation has during acceleration and deceleration.



 $\label{eq:Note} \textbf{Note} \hspace{0.1 in the lim has no effect when Ki is equal to zero.}$ 

# Velocity Feedforward (Vff)

Determines the contribution in the 16-bit DAC command output that is directly proportional to the instantaneous trajectory velocity. This value is used to minimize following error during the constant velocity portion of a move and can be changed at any time to tune the PID loop.

Using velocity feedforward is an open-loop compensation technique and cannot affect the stability of the system. However, if you use too large a value of Vff, following error during the constant velocity portion can reverse, thus degrading rather than improving performance.

Velocity feedforward is typically used when operating in PIVff mode with either a velocity block amplifier or substantial amount of velocity feedback (Kv). In these cases, the uncompensated following error is directly proportional to the specified velocity. You can reduce this following error by applying velocity feedforward. Increasing the integral gain (Ki) also reduces the following error during constant velocity but only at the expense of increased following error during acceleration and deceleration and reduced system stability. For these reasons, increased Ki is not the recommended solution.

Velocity feedforward is rarely used when operating in PID mode with torque block amplifiers. In this case, because the following error is proportional to the torque required (not to the velocity), it is typically much smaller and velocity feedforward is not required.

# **Acceleration Feedforward (Aff)**

Determines the contribution in the 16-bit DAC command output that is directly proportional to the instantaneous trajectory acceleration. Aff is used to minimize following error (position error) during acceleration and deceleration and can be changed at any time to tune the PID loop.

Acceleration feedforward is an open-loop compensation technique and cannot affect the stability of the system. However, if you use too large a value of Aff, following error during acceleration and deceleration can reverse, which can degrade rather than improve performance.

# **Control Loop Update Period**

Sets the PID control loop and trajectory generator update rate. For stepper axes, this parameter also determines how often the step generator is updated. There is a direct correspondence between the number of enabled axes and the fastest update rate allowed as shown in the following table:

Number of Enabled Axes (7330/40/90)	Fastest Update Rate	Number of Enabled Axes (7350)	Fastest Update Rate
1	62 µs	1–2	62 µs
2	125 µs	3–4	125 µs
3	188 µs	5–6	188 µs
4	250 µs	7–8	250 µs

The fastest update rate is achievable only when all axes are single-feedback servo axes and no extra encoders are enabled.



**Note** You cannot set an update rates that is too fast for the number of enabled axes, stepper outputs, and/or encoders.

You can set the update rate slower than the maximum. This is useful in many applications to scale the effective range of the PID control loop parameters and/or to improve stability. Refer to the descriptions for each control loop parameter for more information about the PID parameters affected by the update rate.

# **Filter Settings**

Use the **Filter Settings** tab to enable and configure static friction (also known as stiction) compensation on your motion system. Stiction compensation adjusts for increased power needed to overcome inertia when beginning a move from a standstill.

#### <u>Deadzone</u>

Forward and Reverse Offset Voltages

Note Lowpass and notch filters are available only for the NI 7350.

### Deadzone

The maximum and minimum deadzones are the windows around a static position that determine its sensitivity to external forces that try to move it from its position.

Inside the maximum deadzone window, an axis can be moved without response. As soon as the movement causes the axis to exceed the maximum deadzone, the control loop responds by correcting the position back to within the window defined by minimum deadzone.

## Forward and Reverse Offset Voltages

The forward and reverse offset voltages are the voltages necessary to overcome static friction when beginning a move. The voltages are often the same for both directions, but can be different under some circumstances.

For example, if an axis moves against gravity in one direction and with gravity in the other direction, the offset voltage necessary to overcome static friction against gravity will be higher than that of the direction that moves with gravity.



Note The forward and reverse offset voltages are applied if Integral Gain (Ki)  $\geq$  1.

# **Torque Settings**

Use the **Torque Settings** tab to configure positive and negative torque limits and torque offset.

Positive and Negative Torque Limit

Torque Offset

#### **Positive and Negative Torque Limits**

By limiting the output range of a DAC, it is possible to control the maximum torque (when connected to a torque block servo amplifier) or velocity (when connected to a velocity block servo amplifier). The positive and negative torque limit values define a clamp for the primary servo DAC output and for the secondary DAC output when you are using a brushless motor. The clamp is useful for systems where the maximum torque output must be smaller than the +10 to -10 voltage range.

Primary and secondary DACs can have different limits, and the positive and negative limits can be both positive or both negative to limit the DAC output to a unipolar range. The only restriction is that a positive DAC limit cannot be less than the negative DAC limit.

# **Torque Offset**

N

The torque offset value applies an open-loop compensation for systems where the load in one direction is not equal to the load in the opposite direction. For example, in a system where the motor is mounted horizontally, the torque needed to move up is less than the torque needed for moving down, so this system might need a torque offset to maintain a stable PID control loop. Different torque offsets can be loaded for the primary and secondary DAC.

Note When an axis is killed, its DAC outputs are zeroed regardless of the torque offset loaded.

DAC offsets can be used in conjunction with DAC range limits to interface to servo amplifiers with unipolar input ranges (for example, 0 to 5 V or 0 to 10 V).

### **Breakpoint and Trigger**

Use the **Breakpoint and Trigger** tab to configure the breakpoint (position compare) and high-speed capture (trigger) settings on your motion system.

Refer to the <u>Synchronization</u> section of the *NI-Motion Help* for more information about the types of breakpoints you can configure.



**Note** NI-Motion does not support breakpoint and high-speed capture functionality on an axis when the primary feedback for that axis is an analog feedback.

Breakpoint Mode Breakpoint Action Breakpoint Pulse Width High Speed Capture Inputs

## **Breakpoint Mode**

All of the following modes are available for all NI motion controllers unless otherwise specified.

#### Absolute

This mode is available in both single and buffered operation. The breakpoint position is interpreted with reference to the zero position (origin) and can be anywhere in the 32-bit position range. This also applies to all the position data in the buffer for buffered operation.

#### Relative

This mode is available in both single and buffered operation. For single operation, the breakpoint position is interpreted with reference to the encoder position when the breakpoint is enabled. For buffered operation, the first position data is interpreted with reference to the encoder position when the breakpoint is enabled. The subsequent position data is interpreted relative to the previous breakpoint such that:

Breakpoint[0] = Position on enable + Buffer[0] Breakpoint[n] = Buffer[n-1] + Buffer[n], where n>=1.

#### Modulus (7330/40/90)

This mode is available only in single operation. Modulus breakpoint is used to specify multiple breakpoints with a constant distance between them. The breakpoint position is interpreted with the modulus range such that

Breakpoint[n] = Breakpoint position + ( $n \times$  Breakpoint modulus), where n is any integer.



**Note** For modulo breakpoints, the magnitude of the breakpoint value must be less than the breakpoint modulus. If this range is exceeded, a modal error is generated when you execute the Enable Breakpoint Output VI or function.

When the breakpoint is enabled, the controller enables the two breakpoints closest to the current position, one in the forward direction, and one in the reverse. After a breakpoint occurs, you can re-enable the next breakpoint without loading a new position.

#### Periodic (7350)

This mode is available only in single operation. Periodic breakpoint is an enhanced version of modulus breakpoint. The period determines the distance between each breakpoint:

Breakpoint[n] = Breakpoint position + ( $n \times$  Breakpoint period), where n is any integer.

When you load a breakpoint position, it is interpreted as the first breakpoint. After this breakpoint has occurred, the breakpoint status is asserted and the subsequent breakpoint is automatically loaded and enabled by the controller.

#### **Breakpoint Action**

A breakpoint event can cause the state of the corresponding breakpoint output to change. The **Action** selection determines whether the output changes to low, high, toggles state (NI 7330/40/90 controllers only), pulses (NI 7350 controllers only), or does not change when the breakpoint event occurs. If the breakpoint output is presently in the state defined by **Action**, it is forced to the opposite state when the breakpoint is enabled. This guarantees that the specified transition occurs when the breakpoint is reached.

#### **Breakpoint Pulse Width**

 $\wedge$ 

**Caution** The breakpoint (position compare) frequency depends on the velocity and distance between breakpoints and is limited by the pulse width. The breakpoint frequency *must* meet the requirements specified below for correct operation.

The breakpoint pulse must not overlap with the next breakpoint event. If the breakpoint pulse overlaps with the next breakpoint event, breakpoint generation is stopped and the breakpoint is disabled. To recover from this situation change the pulse width, velocity, or distance between breakpoints. In addition, the breakpoint frequency cannot be greater than the maximum value allowed for the specified breakpoint type. Therefore, the breakpoint frequency, in hertz, at the specified pulse width, in seconds, must conform to the following two conditions:

breakpoint frequency (hertz)< 1 pulse width(seconds)

#### AND

Breakpoint Type	Maximum Frequency
Single	150 Hz
Buffered, PID rate of 62.5 µs to 250 µs	2 kHz
Buffered, PID rate greater than 250 μs	1 kHz
Periodic	4 MHz

### **High Speed Capture Inputs**

Active High, Active Low, Low-to-High Edge, and High-to-Low Edge modes define the state of the input signal that results in a high-speed capture event. Setting the trigger input mode to Active High means that an active high input causes a capture to occur. Conversely, Active Low means that an active low input causes a capture to occur. Notice that if the signal is already at the specified level when high-speed capture is enabled, the capture occurs immediately. Low-to-High Edge requires that the signal make a transition from low to high to trigger a capture; High-to-Low Edge requires that the signal make a transition from high to low to trigger a capture. In all of these modes, the input has latching behavior, meaning that after the high-speed capture input is enabled, if the input signal ever becomes active, the high-speed capture status is true until the input is reenabled

Active High Digital Input and Active Low Digital Input modes allow you to configure the high-speed capture input as a simple digital input independent of the high-speed capture circuitry. In these modes the inputs simply reflect the current state of the input signal. The input does not need to be enabled or reenabled at any time in these modes

# **Digital I/O Settings**

Use the **Digital I/O Settings** tab to configure the settings of the digital input/output bits on the motion controller.

Any changes you make on the main **Digital I/O Settings** tab affect all ports in the list. In addition, values on the main tab display settings for the first port in the list only. If you are using a PCI-7390, this applies to the **Input Port** and **Output Port** settings.



**Note** Clicking **Save** on the main **Digital I/O Settings** item when you have not made any changes does not override any changed settings on the **Port** subitems in the list.

# **ADC Settings**

Use the **ADC Settings** tab to enable or disable ADCs and set their appropriate <u>range</u> in volts.



**Note** With the NI 7350 motion controller, changing the range of an ADC changes all other ADCs to the same range. Other NI motion controllers can set the range independent of other ADCs.



**Tip** If you are using a fast PID rate, <u>disable</u> unused ADC channels. The ADC sample time increases when additional channels are enabled. Refer to <u>Enabling and Disabling ADC</u> <u>Channels</u> for more information.

Any changes you make on the main **ADC Settings** item affect all ADCs in the list. In addition, values on the main tab display settings for the first ADC in the list only.



**Note** Clicking **Save** on the main **ADC Settings** item when you have not made any changes does not override any changed settings on the **ADC Channel** subitems in the list.

# **Enabling and Disabling ADC Channels**

The NI-Motion Analog-to-Digital Converter (ADC) multiplexes between channels with a scan rate of approximately 40  $\mu$ s for the 7340 motion controller, and 25  $\mu$ s for the 7350 motion controller. Therefore, the time between samples for a specific ADC channel is as follows:

- **7330/40:** ADC sample time = 40 μs/channel x (number of enabled channels)
- **7350:** ADC sample time = 25 μs/channel x (number of enabled channels)

The 40  $\mu s$ /channel scan rate is fast enough to support analog feedback at the fastest PID update rates as long as no additional ADC channels are enabled.

# **ADC Range**

Input voltage range over which the ADC will convert input voltages to digital values. Voltages outside of the range will clamp at the extremes, which are -2048 or +2047 (7330/40) and -32,768 to +32,767 (7350) for the -5 to +5 V and -10 to +10 V ranges, and 0 or 4,096 (7330/40) and 0 or +65,535 (7350) for the 0 to +5 V and 0 to +10 V ranges. You can choose from the following ranges:

	Binary Values		
Range Values	7350	7330/40	
0 to 5	0 to +65,535	0 to +4,095	
–5 to +5	-32,768 to +32,767	-2,048 to +2,047	
0 to +10	0 to +65,535	0 to +4,095	
–10 to +10 (default)	-32,768 to +32,767	-2,048 to +2,047	

If you know that the input voltage falls within a more restrictive range, you can effectively increase the resolution of the measurements by selecting an appropriate range from the previous list.

 $\overline{\mathbb{N}}$ 

**Note** With the 7350 motion controller, changing the range of an ADC changes all other ADCs to the same range. Other NI motion controllers can set the range independent of other ADCs.

For example, if you are using a 7340 motion controller, the input signal ranges from –3 to +3 V, and you select the –5 to +5 V range, the 4,096 discreet values for the ADC is 2.44 mV apart instead of the 4.88 mV apart when using the –10 to +10 V range. If you are using a 7350 motion controller, the input signal ranges from –3 to +3 V, and you select the –5 to +5 V range, the 65,536 discrete values for the ADC is 152  $\mu$ V apart instead of the 305  $\mu$ V apart when using the –10 to +10 V range.

ADC ranges cannot be changed on any ADC while an axis that is using analog feedback is enabled. You must first disable the axis using analog feedback, then change the range.

### **Encoder Settings**

Use the **Encoder Settings** tab to configure the <u>encoder filter frequency</u>, polarities of the encoders, and so on. You also can enable or disable the encoder(s) on this tab when they are not assigned to an axis.



**Tip** If your encoder does not supply resolution in quadrature counts per revolution, determine quadrature counts per revolution by multiplying the encoder resolution in encoder lines or periods by four. Encoder counts per revolution is also referred to as counts per index. For example, a 500 line encoder has 2,000 quadrature counts per revolution.

Any changes you make on the main **Encoder Settings** item affect all encoders in the list. In addition, values on the main tab display settings for the first encoder in the list only.



**Note** Clicking **Save** on the main **Encoder Settings** item when you have not made any changes does not override any changed settings on the **Encoder** subitems in the list.

# **Encoder Filter Frequency**

Setting the maximum allowable count frequency for an encoder is useful for reducing the effect of noise on the encoder lines. Noise on the encoder lines can be interpreted as extra encoder counts. By setting the frequency to the lowest possible setting required for the motion application, you can ensure the highest degree of accuracy in positioning. In choosing the appropriate value, you must take into account the counts per revolution of the encoder and the maximum velocity for the axis in question.

For example, with a 20,000 counts per revolution encoder and a maximum velocity of 3,000 RPM (50 revolutions per second), the encoder signal could be as high as 1,000,000 counts per second. A maximum count frequency of 1.6 MHz would be appropriate in this case.



Tip When the encoder filter is changed, reset the position to prevent an erroneous count.

#### **PWM Settings**

Use the **PWM Settings** tab to configure the Pulse Width Modulation (PWM) output settings for the motion controller. You can load the clock frequency, configure the duty cycle, and enable or disable the PWM outputs.

The PWM outputs on the motion controller are digital pulse-train outputs that have a frequency specified by **Clock Frequency** and a duty cycle specified by **Duty Cycle**. These outputs can be used to control devices that require a PWM input, such as a laser whose intensity is controlled by a PWM signal, or can be used to generate isolated analog outputs by passing the PWM output through an optocoupler and then filtering the digital pulse train to produce an analog output voltage.

#### **PWM Clock Frequencies**

Any changes you make on the main **PWM Settings** item affect all PWMs in the list. In addition, values on the main tab display settings for the first PWM in the list only.



**Note** Clicking **Save** on the main **PWM Settings** item when you have not made any changes does not override any changed settings on the **PWM** subitems in the list.

#### **PWM Clock Frequencies**

The base clock frequency for the PWM outputs is 24.81 MHz on the 7340 and 7330 controllers, 24.58 MHz on the 7350 controller, and 20.48 on 7334/42/44 controllers. This base clock frequency is divided down depending on the clock value selected.

Clock Divider Value	Clock Value <sup>*</sup>	7334/42/44 Clock Frequency	7330/40 Clock Frequency	7350 Clock Frequency	
512	0	40.00 kHz	48.46 kHz	48.01 kHz	
1K	1	20.00 kHz	24.23 kHz	24.00 kHz	
2К	2	10.00 kHz	12.11 kHz	12.00 kHz	
4K	3	5.00 kHz	6.06 kHz	6.00 kHz	
8K	4	2.50 kHz	3.03 kHz	3.00 kHz	
16K	5	1.25 kHz	1.51 kHz	1.50 kHz	
33K	6	625.00 Hz	757.14 Hz	750.09 Hz	
External Clock/256 Hz	7	External Clock/256 Hz	External Clock/256 Hz	External Clock/256 Hz	
65K	8	312.50 Hz	378.57 Hz	375.05 Hz	
131K	9	156.25 Hz	189.29 Hz	187.52 Hz	
262K	10	78.13 Hz	94.64 Hz	93.76 Hz	
524K	11	39.06 Hz	47.32 Hz	46.88 Hz	
1048K	12	19.53 Hz	23.66 Hz	23.44 Hz	
2097K	13	9.77 Hz	11.83 Hz	11.72 Hz	
4194K	14	4.88 Hz	5.92 Hz	5.86 Hz	
External Clock/ 32,768 Hz	15	External Clock/ 32,768 Hz	External Clock/ 32,768 Hz	External Clock/ 32,768 Hz	
* These clock value numbers are used for purposes of discussion only.					

The following table describes the PWM clock frequency settings:

When you configure a PWM output, the clock frequency applies to both outputs (PWM 1 and PWM 2). If you configure one PWM output for a clock value of 3, and then the second PWM output for a clock value of 4, the value of 4 applies to both PWM outputs.

The only exception is when the clock settings for the two PWM outputs are 0 and 8, 1 and 9, 2 and 10, and so on, in which case each output has a different frequency. This is because clock values 0–7 and 8–15 are paired. When the clock settings for the two PWM outputs are 0 and 8, 1 and 9, 2 and 10, and so on, each output has a different frequency.

Clock values 0 and 8 use the common time base of 10.24 MHz

(12.8 MHz on the 7350). The frequency of clock value 0 is found by dividing the time base by 256, while clock value 8 is found by dividing the time base by 32768. If you have an NI 7344 controller and you switch channel B to clock value 10, the time base changes to 2.56 MHz. This change also switches channel A to 2.56/256 = 10 kHz.

If the value in one set changes, it switches the other channel to the pairing clock value. If the two values are in the same group, the two channels have the same frequency.

#### **1D Interactive Main**

Use the **1D Interactive Main** tab to test your motion system and configuration on the selected axis.

Operation Mode

## **Using the Interactive Environment**

To begin a move, click **Start** or **Blend**.

To stop the motion immediately, select **Halt**. To slow the motion to a stop, select **Decelerate**. To remove power to the axis by inhibiting the amplifier, select **Kill**.



**Tip** Refer to the <u>Trajectory Parameters</u> section of the *NI-Motion Help* for information about minimum and maximum allowable velocity, acceleration, and deceleration values.

# **Operation Mode**

There are four position modes and one velocity mode, as described below:

## **Absolute Position**

In absolute position mode, target positions are interpreted with respect to an origin, reference, or zero position. The origin is typically set at a home switch, end of travel limit switch, or encoder index position. An absolute position move uses the preprogrammed values of acceleration, deceleration, s-curve, and velocity to complete a trajectory profile with an ending position equal to the loaded absolute target position.

**Caution** Any single move is limited to between  $-2^{31}$  and  $2^{31}-1$  counts or steps. An error is generated if you exceed this limit by loading a target position too far from the current position.

The length of an absolute move depends upon the loaded position and the current position when the move is started. If the target position is the same as the current position, no move occurs.

### **Relative Position**

In relative position mode while motion is not in progress, loaded target positions are interpreted with respect to the current position at the time the value is loaded. A relative position move uses the preprogrammed values of acceleration, deceleration, s-curve and velocity to complete a trajectory profile with an ending position equal to the sum of the loaded relative target position and the starting position.

If a relative move is started while motion is in progress, the new target position is calculated with respect to the target position of the move already in progress (considered to be the new starting position), as if that move had already completed successfully. Motion continues to the new relative position, independent of the actual position location when the new move is started.

In relative mode, the new target position is calculated and doublebuffered when you enter a new **Target Position**. You must reload the relative target position each time before clicking the **Start** or **Blend** button.

# Velocity

In velocity mode, the axis moves at the loaded velocity until you stop motion, a limit is encountered, or a new velocity is loaded and you start motion again. Loaded target positions have no effect in velocity mode. The direction of motion is determined by the sign of the loaded velocity.

You can update velocity at any time to accomplish velocity profiling. Changes in velocity while motion is in progress uses the preprogrammed acceleration, deceleration, and s-curve values to control the change in velocity. You can reverse direction by changing the sign of the loaded velocity and clicking the **Start** button.



**Note** Clicking the **Blend** button in velocity mode has no effect because the move in process never normally stops. You must always click the **Start** button to update velocity in velocity mode.

### **Relative To Capture**

The relative-to-capture position mode is very similar to absolute position mode, except that the zero position reference (origin) is the last captured position for the axis or axes. A relative-to-capture position move uses the preprogrammed values of acceleration, deceleration, s-curve and velocity to complete a trajectory profile with an ending position equal to the sum of the loaded target position and the last captured position.

In relative mode, the new target position is calculated and doublebuffered when you enter a new **Target Position**. You must reload the relative target position each time before clicking the **Start** or **Blend** button.

In relative-to-capture mode, the new target position is calculated and double-buffered when you enter a new **Target Position**. These functions use existing values in the position capture register(s). You must load the target position after the capture event has occurred and before clicking the **Start** or **Blend** button.

This mode is typically used in registration applications. Refer to the <u>Gearing</u> section of the *NI-Motion Help* for information about using superimposed moves/registration applications. Also, refer to the <u>Rotating</u> <u>Knife</u> section of the *NI-Motion Help* for example code that includes superimposed moves.

## **Modulus Position**

In modulus position mode, the loaded target position is interpreted within the boundaries of a modulus range and the direction of motion is automatically chosen to generate the shortest trajectory to the target.

Modulus position mode is typically used with rotary axes or for other similarly repetitive motion applications.



**Note** Multiple revolution moves cannot be accomplished by indicating target positions greater than the modulus value. All moves are resolved to one modulus range.

### **1D Interactive Advanced**

Use the **1D Interactive Advanced** tab to control the advanced properties of the move.

<u>S Curve Time</u>

2

**Breakpoint Mode** 

#### **Breakpoint Action**

**Note** For closed-loop stepper systems in which the encoder counts-per-revolution differs from the steps-per-revolution, the breakpoint position is loaded in encoder counts.

#### **Using the Interactive Environment**

To begin a move, click **Start** or **Blend**.

To stop the motion immediately, select **Halt**. To slow the motion to a stop, select **Decelerate**. To remove power to the axis by inhibiting the amplifier, select **Kill**.

### **Plotting Your 1D Move**

Use the **Misc. Plots** tab to view the results of the move. This tab allows you to plot axes against variables. You can view the following information:

- Axis position vs. time
- Axis velocity vs. time
- Axis position and velocity vs. time
- Following error vs. time
- DAC outputs vs. time

Right-click anywhere on the graph to change the plotting parameters.

#### **2D Interactive Main**

Use the **2D Interactive Main** tab to test your motion system and configuration on two axes at a time.

Operation Mode

## **Using the Interactive Environment**

To begin a move, click **Start** or **Blend**.

To stop the motion immediately, select **Halt**. To slow the motion to a stop, select **Decelerate**. To remove power to the axis by inhibiting the amplifier, select **Kill**.



**Tip** Refer to the <u>Trajectory Parameters</u> section of the *NI-Motion Help* for information about minimum and maximum allowable velocity, acceleration, and deceleration values.

### **2D Interactive Advanced**

Use the **2D Interactive Advanced** tab to control the advanced properties of the move.

<u>S Curve Time</u>

#### **Using the Interactive Environment**

To begin a move, click **Start** or **Blend**.

To stop the motion immediately, select **Halt**. To slow the motion to a stop, select **Decelerate**. To remove power to the axis by inhibiting the amplifier, select **Kill**.

### **Plotting Your 2D Move**

Use the **Position Plots** tab to view the position plot of the move. This tab, in the 2D Interactive Environment, allows you to plot axes against variables. You can view the following information:

- X-axis position vs. Y-axis position
- X-axis position vs. time
- Y-axis position vs. time
- X-axis and Y-axis vs. time

Right-click anywhere on the graph to change the plotting parameters.

### **Plotting Your 2D Move**

Use the **Misc. Plots** tab to view the results of the move. This tab allows you to plot certain attributes of your axes. You can view the following information:

- Vector velocity
- X-axis velocity
- Y-axis velocity
- X-axis and Y-axis velocities
- Vector following error
- X-axis following error
- Y-axis following error
- X-axis and Y-axis following error

Right-click anywhere on the graph to change the plotting parameters.

## Working with Device Settings

You can use Measurement & Automation Explorer (MAX) to create, copy, rename, import, export, and delete device settings for your motion controller.

- <u>Creating New Device Settings Files</u>
- <u>Copying Device Settings</u>
- <u>Renaming Device Settings</u>
- Importing Device Settings
- Exporting Device Settings
- Deleting Device Settings

# **Copying Device Settings**

You can use Measurement & Automation Explorer (MAX) to copy and paste device settings in the Initialization Settings dialog box.

Complete the following steps to copy device settings:

- 1. Launch MAX.
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion controller.
- 4. Right-click **Default** *xxxx* **Settings** in the configuration tree.
- 5. Select Initialization Settings»Change Current Settings.
- 6. Right-click the settings you want to copy and select **Copy**.
- 7. Right-click anywhere in the Initialization Settings dialog box and select **Paste** to paste the settings file you copied.
- 8. <u>Rename</u> the copied settings file as appropriate.
- 9. Click **Change** to apply the newly renamed settings to the currently selected device.
  - Note If you do not want to apply the newly renamed settings to the currently selected device, click **Do Not Change**.

### **Copying Parameters**

You also can copy parameters from the settings for one device to the settings for another device.

Complete the following steps to copy parameters:

- 1. Launch MAX.
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion controller.
- 4. Right-click **Default** *xxxx* **Settings** in the configuration tree.
- 5. Select **Initialization Settings\*Copy Settings from**, and then select the initialization settings you want to copy the parameters from.

Copying the parameters from the initialization settings you selected changes all of the parameters in the current initialization settings to the parameters in the initialization settings you selected, as applicable. If the current initialization settings include parameters that are not included in the initialization settings you are copying from, the additional parameters are not overwritten.

#### **Creating New Device Settings Files**

Complete the following steps to create a new device settings file:

- 1. Launch Measurement & Automation Explorer (MAX).
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion controller.
- 4. Right-click **Default** *xxxx* **Settings**.
- 5. Select Initialization Settings»Change Current Settings.
- 6. Right-click anywhere in the Initialization Settings dialog box and select **Create New Settings**.
- 7. Type a name for the settings file.
- 8. Make sure the new settings file is selected in the Initialization Settings dialog box, and click **Change** to make the new settings active.
- 9. Work through the MAX navigation items for the device, and set up the appropriate initialization and configuration settings for the motion controller.
- 10. Click **Save** to save the new settings to the MAX database, and then click **Initialize** to download the settings to the controller.

#### **Changing Device Settings Files**

Complete the following steps to change the device settings file:

- 1. Launch Measurement & Automation Explorer (MAX).
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion controller.
- 4. Right-click **Default** *xxxx* **Settings** in the configuration tree.
- 5. Select Initialization Settings»Change Current Settings.
- 6. Select the device settings file you want to use, and click **Change**.

When you click **Initialize**, these settings are downloaded to the controller.

### **Deleting Device Settings**

You can use Measurement & Automation Explorer (MAX) to delete device settings files in the Initialization Settings dialog box.

Complete the following steps to delete a device settings file:

- 1. Launch MAX.
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion controller.
- 4. Right-click **Default** *xxxx* **Settings** in the configuration tree.
- 5. Select Initialization Settings»Change Current Settings.
- 6. Right-click the settings you want to delete and select **Delete**.
- 7. Close the Initialization Settings dialog box.

### Importing and Exporting Device Settings

# **Exporting Device Settings**

Complete the following steps to export motion device settings in XML format:

- 1. Launch Measurement & Automation Explorer (MAX).
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion controller.
- 4. Right-click **Default** *xxxx* **Settings** in the configuration tree.
- 5. Select Initialization Settings»Export Settings.
- 6. Navigate to the location you want to save the settings to and click **Save**.

#### **Importing Device Settings**

Complete the following steps to import device settings in XML format:

- 1. Launch MAX.
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion controller.
- 4. Right-click **Default** *xxxx* **Settings** in the configuration tree.
- 5. Select Initialization Settings»Import Settings.
- 6. Select a previously exported settings file, and then click **Open**.
- 7. In the Initialization Settings dialog box, select the settings you just imported and click **Change**.

**Note** If the imported settings are not valid for the device that you are importing them for, the Initialization Settings dialog box does not launch. You must copy the parameters from the imported settings into the current settings for the device. Refer to <u>Copying Device Settings</u> for more information.

P

N

**Tip** Refer to <u>Import/Export Wizards</u> in the *Measurement & Automation Explorer Help* for information about the MAX Import/Export Wizard. You can use the MAX Import/Export Wizard to import and export device configuration information in the NI Configuration Export Files format (\*.nce).

#### NI SoftMotion Controller Considerations

The following list includes considerations for using this function with the NI SoftMotion Controller:

- When you import settings for the NI SoftMotion Controller, MAX verifies if the axes in the imported settings are physically connected. If any of the axes in the imported settings are not physically connected or are not working, the axes are marked as disconnected. To reconnect the settings for a disconnected axis, ensure that the drive ID and drive type match the axis in the settings file. To reconnect the axis, right-click the axis and select **Reconnect Axis**.
- If the NI SoftMotion Controller is not in the reset state, you must reset it to connect the axis.

#### **Downloading Settings to the Controller**

Complete the following steps to change the device settings file:

- 1. Launch Measurement & Automation Explorer (MAX).
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then click the appropriate controller name.
- 4. When you have fully configured your controller, click **Initialize** to download the settings to the device.
  - **Tip** During configuration, if you want to revert the current settings in MAX to the most recent downloaded configuration, click **Refresh**.

### **Renaming Device Settings**

You can use Measurement & Automation Explorer (MAX) to rename device settings files in the Initialization Settings dialog box.

Complete the following steps to rename a device settings file:

- 1. Launch MAX.
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion controller.
- 4. Right-click **Default** *xxxx* **Settings** in the configuration tree.
- 5. Select Initialization Settings»Change Current Settings.
- 6. Right-click the settings you want to rename and select **Rename**.
- 7. Type a new name for the file and click **Change**.



**Tip** To rename the **Default** *xxxx* **Settings** configuration tree item, right-click the item and select **Rename**.

#### **Testing Motion Controller Resources**

Complete the following steps to test the resources associated with your motion controller:

- 1. Launch Measurement & Automation Explorer (MAX).
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then select the appropriate motion device.
- 4. Click the **Status** tab.
- 5. Click **Test Resources**.

Pass/fail information appears in the window below the **Test Resources** button.

### **Testing and Debugging Motion Controllers**

Use the **Interactive** configuration tree item in Measurement & Automation Explorer (MAX) to test and debug your motion controller. Using the interactive panels, you can test a single axis or multiple axes based on parameters that you set.

Complete the following steps to test and debug the controller:

- 1. Launch MAX.
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion device.
- 4. Right-click **Default** *xxxx* **Settings** in the configuration tree.
- 5. Click **Initialize** to download the current device settings to the controller.
- 6. Expand **Interactive** in the configuration tree and select the appropriate subitem depending on how many axes you want to test.
- 7. Click **Start**, and then review the status LEDs in the **Main** and/or **Advanced** tabs.

#### **Testing Moves**

Complete the following steps to test a 1D move in Measurement & Automation Explorer (MAX):

- 1. Launch MAX.
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then select the item for the appropriate motion device.
- 4. Click Initialize.
- 5. Expand Interactive in the configuration tree and select 1-D Interactive.
- 6. Select the **Main** tab.
- 7. Change the number in **Target Position** to a number that is different from the one under **Current Trajectory Data**.
- 8. Click Apply.
- 9. Click **Start**. The motor moves to the target position.
- 10. Repeat these steps for each remaining axis.

#### **Testing Limit and Home Switches**

Complete the following steps to test the limit and home switches:

- 1. Disconnect power to the motors and drive. If you have a National Instruments nuDrive or MID drive, turn off the Enable switch.
- 2. Launch Measurement & Automation Explorer (MAX).
- 3. Expand **Devices and Interfaces** in the configuration tree.
- 4. Expand **NI–Motion Devices**, and then select the item for the appropriate motion device.
- 5. Click Initialize.



- Note This step assumes that you have correctly configured the initialization settings. You configure limit and home switches on the **Motion I/O** tab.
- 6. Ensure that the switches are correctly powered.
- 7. Expand Interactive in the configuration tree and select 1-D Interactive.
- 8. Move the motor manually until it hits the limit or home switch. When the limit or home switch is active, LEDs on the **Main** tab of the **Interactive\*1-D Interactive** tab change from gray to red. As you move the motor away from the limit or home switches, these LEDs turn back to gray.
- 9. Repeat these steps for each motor that is attached to limit or home switches.

#### **Testing Motors**

Complete the following steps to test motors.

- 1. Disconnect power to the motors and drives. If you have a National Instruments nuDrive or MID drive, do not turn on the Enable switch.
- 2. Verify all the wiring between the motion controller and motor and between the drive and the motor. If you have servo motors, you may need to verify the direction of the voltage. If you have stepper motors, you may need to verify the polarity of the step and direction lines. Refer to the motor drive documentation for polarity information.
- 3. Launch Measurement & Automation Explorer (MAX).
- 4. Expand **Devices and Interfaces** in the configuration tree.
- 5. Expand **NI–Motion Devices**, and then select the item for the appropriate motion device.
- 6. Click Initialize.
  - Note This step assumes that you have correctly configured the initialization settings.
- 7. Expand Interactive in the configuration tree and select 1-D Interactive.
- 8. Select 1-D Interactive.
- 9. Click the **Advanced** tab.
- 10. Because the motor could run out of control if the motion control system is set up incorrectly, set the Following Error trip point to a low number to prevent damage to the motion control system. For example, if the length of your test system is 10,000 counts/steps and the start position is 5,000 counts/steps, you might set the following error trip point to 3,000 counts/steps to ensure that it occurs before the physical end of the system.
- 11. Power on the drive and motor.
- 12. Click **Halt** to energize the motor. The motion controller is sending information to the drive correctly if the motor maintains its position and resists movement by hand.
  - For servo motors, this step may cause the motor to buzz or

oscillate. If this happens, use the <u>servo tune panels</u> to tune the motors.

- For stepper motors, execute a move to identify correct behavior.
- 13. Repeat these steps for the remaining motors.

### **Testing Encoders**

If the motion control system uses servo motors or closed-loop stepper motors, you must test the encoders. If the motion control system uses open-loop stepper motors, skip this procedure.

Complete the following steps to test encoders:

- 1. Power off the motors and drive. If you have a National Instruments nuDrive or MID drive, do not turn on the Enable switch.
- 2. Ensure that the encoders are powered on.
- 3. Launch Measurement & Automation Explorer (MAX).
- 4. Expand **Devices and Interfaces** in the configuration tree.
- 5. Expand **NI–Motion Devices**, and then select the item for the appropriate motion device.
- 6. Click Initialize.



**Note** This step assumes that you have correctly configured the initialization settings.

- 7. Expand Interactive in the configuration tree and select 1-D Interactive.
- 8. If you are testing a stepper motor, skip to step 11. If you are testing a servo motor, select the **Main** tab.
- 9. Move the motor manually in the forward direction. The encoder count increments in the position indicator. Move the motor manually in the reverse direction. The encoder count decrements.
  - Tip If you see the encoder decrement instead of increment, swap encoder cable A+ with A- or B+ with B-, but not both. If you have single-ended encoders, swap channel A with B.
- 10. Proceed to step 12.
- 11. If you are testing a stepper motor, ensure you have closed-loop selected as the stepper loop mode in the **Main** tab.
- 12. Select the **Advanced** tab and make sure the encoder position increments in the encoder indicator.
- 13. Repeat these steps for the remaining encoders.

#### **Troubleshooting Motors**

Complete the steps in the following sections if your motors do not move as expected when you <u>test</u> them:

# **Check the Motion Controller**

- Ensure that the cable is not loose on either end.
- Check the polarities of the inhibits to make sure they match the hardware settings on the UMI or drive.
- Before turning on the drive, <u>check that the</u> <u>encoders work correctly</u>. When the encoders are working correctly, the encoder counts increment and decrement on the **1-D** Interactive tab in Measurement & Automation Explorer (MAX) when you move the motor manually.
  - Turn the motor clockwise, or in the forward direction, to increment counts.
  - Turn the motor counterclockwise, or in the reverse direction, to decrement counts.
    - **Tip** Moving the motor in a forward direction typically increments the count, and moving the motor in a reverse direction typically decrements the counter, but not always. Refer to the motor documentation for information about incrementing and decrementing the count.
- Turn on the drive.
- For servo motors, check the following error status to ensure the motor is not tripping out on following error immediately after the start of a move. This situation may occur if the encoders

or motors are wired incorrectly.

# **Check the Drive**

- Ensure that the drive is enabled. Most drives have a reset or enable switch.
- If you are using a UMI, ensure that 5 V is being supplied for encoder power.
- If you are using brushless servo motors and the drive is handling the commutation, ensure that the Hall effect sensors and encoders are connected to the drive correctly. Refer to the drive documentation for information about connecting the drive to motion controllers.
- Most servo drives must be configured. If the servo drive comes with tuning or configuration software, ensure you have installed it and configured the drive for torque/current, velocity, or p-command mode.

In most cases, the NI motion controller closes the position loop, so the position loop typically should not be closed by the drive. The following table includes considerations to make in other circumstances:

Using the drive in p-command mode for 73 <i>xx</i> controllers.	Use the drive configuration software to tune the position loop.
Using the NI SoftMotion Controller for CANopen.	Use the drive configuration software to tune the position loop.

The velocity loop can be optionally closed on the drive. Make sure that the gains for the drive are properly adjusted for the motor you are using.

• If you are using stepper motors, ensure that the step and direction lines are connected according to the requirements of the drive. In some cases, a pull-up resistor may be required.

#### **Tuning Servo Motors**

You can use Measurement & Automation Explorer (MAX) to tune a servo motor either manually or automatically, and you can tune a motor from scratch. Refer to the following topics for more information about various methods for tuning a motor:

- <u>Using MAX to Tune a Servo Motor</u>
- <u>Automatically Tuning Motors</u>
- Manually Tuning Motors
- Manually Tuning a System from Scratch
- Advanced Tuning Techniques

#### Using MAX to Tune a Servo Motor

Complete the following steps to tune a servo motor in MAX:

- 1. Launch MAX.
- 2. Expand **Devices and Interfaces**.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion controller.
- 4. Expand **Calibration** in the configuration tree, and select **Servo Tune**.
- 5. Select the **Control Loop** tab.
- 6. Configure the **PID Gain** and **Velocity Feedback Gain** settings as appropriate.
  - Proportional Gain (Kp) is the system stiffness. It determines the contribution of restoring force directly proportional to the position error. Restoring force is comparable to a spring in a mechanical system.
     A high proportional gain gives a stiff responsive system but can cause instability from overshoot and oscillation.
  - **Derivative Gain (Kd)** is the damping effects on the system. It determines the contribution of restoring force proportional to the rate of change, or derivative, of position error. This force is much like viscous damping in a damped spring and mass mechanical system, such as a shock absorber. Increasing derivative gain reduces oscillation at the commanded position. However, increasing Kd results in high acceleration, which can cause a ringing sound.
  - Integral Gain (Ki) is the static torque load on the system. It determines the contribution of restoring force that increases with time, ensuring that the static position error in the servo loop is forced to 0. This restoring force works against constant torque loads to help achieve zero position error when an axis is stopped.

Integral gain improves positional accuracy. High static torque loads need integral gains to minimize position error when stopped.

7. Click the **Step Response** tab to view the system step response.

- **Settling Time** is the time required by the response curve to reach and stay within a range that is approximately the final value of size. This value is specified by the absolute percentage of the final value, which is 2% to 5%.
- **Rise Time** is the time required by the response to rise from 10% to 90% of its final value. The faster the response time of the system, the faster the rise time.
- **Peak Time** is the time required for a response to reach the first peak of the overshoot.
- **Maximum Overshoot** is the maximum peak value of the response curve measured from the specified position. The maximum overshoot directly indicates the relative stability of the system.

Use the **Step Response** tab to plot the transient response of your system. Typically, the transient response is measured by first commanding a step, and then measuring how quickly a system reaches a steady state. Using the transient response, you can calculate the maximum overshoot, rise time, peak time, and settling time of your system.

You also use this tab to determine the relative stability of the system. A system is considered stable if the actual position is finite when the commanded position is finite. Similarly, a system is stable if a commanded position results in the motor coming to rest at a single position.

- **Tip** A system is considered unstable when any commanded position typically results in an exponential increase in position error, or the oscillations never dampen.
- 8. Analyze the step response plot.
  - **Over-damped system** produces a smoother, slower step response. An over-damped system is characterized by no overshoot and long rise and settling times.
  - **Under-damped system** produces a slight oscillatory response that eventually dampens out. An under-damped system is characterized by a large overshoot, a long settling time, and short peak and rise times.

- Critically damped system response provides a balanced medium between over- and under-damping. This type of system response balances response time and damping effects. A critically damped system is characterized by low overshoot, shorter rise and settling times when compared to over-damped systems, and longer peak times when compared to under-damped systems.
- **Oscillatory system response** produces a constantamplitude, continuous-position oscillation. An oscillatory system is characterized by infinite settling times.
- **Chattering system response** is an oscillatory response characterized by high-frequency, low-amplitude oscillations that are audible.

### **Automatically Tuning Motors**

When you automatically tune your servo system, use the **Control Type** and **Response Time** parameters on the **Main** tab to customize your system. Smooth controls have a slower response time. The smoother the control, the less the axis will overshoot the specified position before slowing. The more overshoot the system can manage, the faster the response times.

The motor connected to the axis oscillates to identify parameters, such as inertia and friction. Optimal control loop gains are calculated based on these values and the control type that you selected.



**Tip** By default, MAX saves control loop gain settings to the MAX database, and the settings are used for all subsequent initializations of your controller. If you do not want to save the control loop gain settings to the MAX database, deselect the **Save control loop gains to Initialization Settings after auto tune** option in the **Main** tab.

Click the **Step Response** tab to view the type of response Servo Tune achieved for the system. If the response time is unsatisfactory, change the control type and tune the motor again. Use the **Control Loop** tab to view the calculated control loop gains.

In most cases, auto tuning is 99% of the system tuning process. <u>Manual</u> <u>tuning</u> makes up about 1% of this process, and is used to tweak the PID gains to achieve an optimal response.



**Caution** If your system becomes unstable and unmanageable, including demonstrating rapidly fluctuating oscillations, immediately disable your drive and turn off the power. Disable the power switches on the drive before proceeding.

Complete the following tutorial for an example of how to automatically tune your servo motor:

- 1. Launch MAX.
- 2. Expand **Devices and Interfaces** in the configuration tree.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion device.
- 4. Expand **Calibration** in the configuration tree, and select **Servo Tune**.
- 5. Select the Main tab.
- 6. Select Axis 1 in Axis.
- 7. Adjust the **Control Type** to 70% to indicate smooth control and

slow response time.

- 8. Click **Auto Tune**. The motor oscillates as it determines the correct PID values for the system.
- 9. Click the **Step Response** tab and enter 500 in **Samples**.

A low overshoot and few, or no, oscillations occur. The Step Response plot shows that the system is slow to respond to commands, which is demonstrated by the slow migration to the commanded position.

- 10. Select the **Main** tab.
- 11. Select **Axis 1** in **Axis**.
- 12. Adjust the **Response Time** to 70% to indicate a fast response and stiff control.
- 13. Click **Auto Tune**. The motor oscillates as it determines the correct PID values for the system.
- 14. Click the **Step Response** tab. Compare this plot to the previous smooth control plot. The difference is that the system responds faster, but produces more oscillations and a greater overshoot.
- 15. Experiment with feeling the different responses by using your hand to move the motor to the off position, and feel the response when it snaps back.

#### **Manually Tuning Motors**

Use the **Control Loop** tab to view and edit the PID parameters. Auto Tune provides a tuned system, but for an optimally tuned system, it is necessary to fine-tune the final PID parameters. It may also be necessary to alter the PID parameters, depending on your specific circumstances.

- Proportional Gain (Kp)
- Derivative Gain (Kd)
- Integral Gain (Ki)
- Derivative Sampling Period (Td)

Complete the following steps to manually tune a motor:



**Note** This exercise assumes that Axis 1 is configured as a servo axis.

- 1. Launch MAX.
- 2. Expand **Devices and Interfaces**.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion device.
- 4. Expand **Calibration** in the configuration tree, and select **Servo Tune**.
- 5. <u>Auto tune</u> the motor first.
- 6. Click the **Control Loop** tab and note the values on the Kp, Kd, Ki, and Td controls. Write these values down for future reference.
- 7. In the **Control Loop** tab, increase the proportional gain (Kp) by a factor 2, and click **Save** in the MAX toolbar. Increasing by a factor of 2 works for most systems, but if your system becomes unstable, reduce this factor appropriately.
- 8. Click the **Step Response** tab to examine the Step Response plot. As you increase the proportional gain, the rise, peak, and settling times decrease, producing a stiffer, faster response. The maximum overshoot increases as well. Increasing by a factor of 2 typically results in a noticeably higher overshoot. If this is not true for your particular system, try increasing your gain again.
- 9. Decrease the proportional gain (Kp) by a factor of 2 from its original value or a factor of 4 from its current value. Decreasing by

a factor of 2 works for most systems, but if your system becomes too sluggish, reduce this factor appropriately.

- 10. Click the **Step Response** tab to examine the Step Response plot. As you decrease the proportional gain, the rise, peak, and settling times increase, producing a smoother, slower response. As you decrease Kp, the maximum overshoot decreases as well. Decreasing by a factor of 2 typically results in a noticeably more damped system. If this is not true for your particular system, try decreasing the gain again.
- 11. Reset the Proportional Gain (Kp) to the original value you noted in step 6.
- 12. Increase the Derivative Gain (Kd) by a factor of 1.5. Increasing by a factor of 1.5 works for most systems, but if your system becomes too sluggish, reduce this factor appropriately.
- 13. Click the **Step Response** tab to examine the Step Response plot. As you increase the derivative gain, the maximum overshoot decreases and the settling time increases. The number of visible oscillations decreases as well. Increasing by a factor of 2 typically results in a noticeably more damped system. If this is not true for your particular system, try increasing your gain again.
- 14. Decrease the Derivative Gain (Kd) by a factor of 1.5 from the original value, or a factor of 2.25 from the current value. Decreasing by a factor of 1.5 works for most systems, but if your system becomes unstable, decrease this factor appropriately.
- 15. Click the **Step Response** tab to examine the Step Response plot. As you decrease the derivative gain, the maximum overshoot increases while the settling time decreases. The number of visible oscillations increases as well. Decreasing by a factor of 1.5 typically results in a noticeably higher overshoot. If this is not true for your particular system, try decreasing your gain again.
- 16. Reset the Derivative Gain (Kd) to the original value you noted in step 6.
  - Note There are subtle differences between the Kp and Kd values on your system. Increasing Kp increases the slope of the initial rise to the commanded position. As you increase Kp, you approach the commanded position faster,

and thus overshoot by a greater amount. Kd reduces the oscillations over a period of time after the initial rise. When Kp decreases, Kd becomes dominant, and when Kd decreases, Kp becomes dominant. When tuning your system, the goal is to find a comfortable balance between Kp and Kd such that there is adequate response time, which is controlled primarily by Kp, and minimal overshoot, which is controlled primarily by Kd, without having to significantly increase or decrease the gains. Increasing or decreasing the gains too much can create an unstable system and possibly damage the motor.

- 17. Decrease the Integral Gain (Ki) to 0.
- 18. Click the **Step Response** tab to examine the Step Response plot. You may notice a steady state error position delta on the Step Response plot. This is the steady-state position error.
- 19. Increase the Integral Gain (Ki), by increments of 1, beyond its original value. Be careful not to increase Ki too much.
- 20. Click the **Step Response** tab to examine the Step Response plot. Increasing the Integral Gain beyond what is necessary to correct for a steady state position error can result in an unstable system.
  - **Tip** As you become familiar with the effects of Ki on your system, you will notice that it corrects steady-state position error. Some simple systems may not require Ki.
- 21. Reset the Integral Gain (Ki) to the original value you noted in step6.

### Manually Tuning a System from Scratch

As Kp increases, you approach the commanded position faster, and thus overshoot by a greater amount. The opposite is also true. Kd reduces the oscillations over a period of time after the initial rise. When Kp decreases, Kd becomes dominant; and when decreasing Kd, Kp becomes dominant.

When tuning your system, it is necessary to find a comfortable balance between Kp and Kd such that there is adequate response time, which is primarily controlled by Kp; and minimal overshoot, which is primarily controlled by Kd; without having to increase or decrease the gains too much. Increasing or decreasing the gains too much can create an unstable system, and possibly damage the motor. Ki is used to correct for steady-state position error, and is typically the last step.

Complete the following steps to achieve a critically damped system from scratch:

- 1. Launch MAX.
- 2. Expand **Devices and Interfaces**.
- 3. Expand **NI–Motion Devices**, and then expand the item for the appropriate motion device.
- 4. Expand **Calibration** in the configuration tree, and select **Servo Tune**.
- 5. Select the **Control Loop** tab.
- 6. Set the integral gain (Ki) to 0. Set the proportional gain (Kp) to a reasonable starting value, such as 25.
- 7. Set the derivative gain (Kd) to a reasonable starting value, such as Kp \* 2, or 50.
- 8. Click the **Step Response** tab to examine the Step Response plot. The Step Response plot shows a significant overshoot. If the plot does not show a significant overshoot, increase the proportional gain by factors of 1.5 to 2 until the Step Response plot yields an overshoot.
- 9. After achieving overshoot, attempt to damp the system to reduce maximum overshoot and settling time, while still trying to maintain a low rise time.
  - a. Click the **Control Loop** tab and begin with a typical value

for the derivative gain of Kp \* 2.

- b. Increase the derivative gain by factors of 1.5 to 2 until the oscillations diminish or the system begins buzzing.
- c. When you hear buzzing from the motor, reduce Kd and proceed to the next step.
  - **Tip** If at any time the system is buzzing, the gains are too high. Reduce the gains and continue.
- 10. If the system is slow to respond to commands, increase the proportional gain and derivative gain by factors of 1.5 to 2 until you are satisfied with the step response. High rise time, peak time, and settling time are indicative of a sluggish machine. You can tweak your step response by adjusting the proportional gain or derivative gain.
- 11. If your system continues buzzing and you are unable to achieve a favorable step response after adjusting Kp and Kd, adjust the derivative sampling period (Td). Increase Td by 2, and adjust Kp and Kd accordingly. Depending on the step response graph you get out of the new settings, you may need start over at step 5.
- 12. After achieving a satisfactory step response, you can alleviate any steady state error using the integral gain (Ki). Not all systems need integral gain. Use the Step Response plot to determine the steady-state following error. If the final position at steady state is not the same as the commanded position, the solution may be to use integral gain. Adjust Ki, starting at 1, and increase incrementally by 1 or 2 until the steady state error on the step response is eliminated.
  - **Tip** Remember that if you increase Ki too much, your system may become unstable.

P

# **Advanced Tuning Techniques**

You can use bode plots and advanced control loop parameters to finetune your motion control system.

#### **Bode Plots**

Bode plots display the frequency response of your system. The **Calibration»Servo Tune»Bode** tab in MAX plots the bode diagram for your system. To perform the bode analysis, Servo Tune oscillates the axis to identify the system and to calculate the transfer function. Gain is the measure of the amplitude difference of the input to the system and output from the system. Phase defines the time shift between the input and output. Gain is plotted in decibels (dB), and phase is plotted in degrees.

#### Stability in the Frequency Domain

Use bode plots to measure system stability. At low frequencies, the gain is 0 dB for most systems and diminishes as frequency increases. A rise in gain before a fall in gain indicates marginal stability. For most systems, the allowable rise in gain before falling is below 6 dB, indicating approximately 50% overshoot. You can use the bode plots to ensure that at all significant velocities are stable given the PID parameters.

Gain margin is the gain of the system when the phase is at 180 degrees. Phase margin is the difference between the actual phase and 180 degrees when the gain is at 0 dB. Typically, the phase margin must be between 35 and 80 degrees for a stable and responsive system. The phase margin value must be as large as possible within this range. The gain margin must be between 10 and 25 dB.

#### **Advanced Control Loop Parameters**

The following **Control Loop** parameters are often necessary when using velocity or voltage amplifiers:

• Velocity feedback gain (Kv) is similar to derivative gain (Kd), except that it scales the velocity estimated from encoder resources only. The derivative gain scales the derivative of the position error, which is the difference between the instantaneous trajectory position and the primary feedback position. Like the Kd term, the velocity feedback derivative is calculated every derivative sample period, and the contribution is updated every PID sample period.

You can use a primary or secondary feedback encoder for velocity feedback. Setting Kv to a value other than zero (0) enables velocity feedback using the secondary encoder when using dual-encoder feedback, or the primary encoder if a secondary encoder is not configured.

The velocity feedback gain scales this velocity feedback before it is added to the other components in the 16-bit DAC command output. You can use velocity feedback gain for backlash compensation if your system has gears. In this example, you can configure the primary feedback to the linear encoder on your system, and configure the secondary feedback to the rotary encoder on the shaft of the motor.

• Velocity Feedforward (Vff) determines the contribution in the 16bit DAC command output that is directly proportional to the instantaneous trajectory velocity. Your system uses this value to minimize following error during the constant velocity portion of a move and can be changed at any time to tune the PID loop.

Because velocity feedforward is an open-loop compensation technique, it cannot affect system stability. However, if the Vff value is too large, the following error during the constant velocity portion can reverse—providing negative following error, which can degrade performance.

Velocity feedforward is rarely used when operating in PID mode with torque block amplifiers. In this case, following error is typically much smaller because it is proportional to the torque required rather than to the velocity. When operating in PID mode with torque block amplifiers, velocity feedforward is not required.

• Acceleration Feedforward (Aff) determines the contribution in the 16-bit DAC command output that is directly proportional to the instantaneous trajectory acceleration. Use Aff to minimize following error during acceleration and deceleration. You can change Aff at any time to tune the PID loop.

Because acceleration feedforward is an open-loop compensation technique, it cannot affect system stability. However, if the Aff value is too large, following error during acceleration and deceleration can reverse—providing negative following error, which can degrade performance.

# **Important Information**

<u>Warranty</u>

<u>Copyright</u>

<u>Trademarks</u>

Patents

Warning Regarding Use of NI Products

#### Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

# Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

#### Trademarks

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on ni.com/legal for more information about <u>National Instruments trademarks</u>.

FireWire® is the registered trademark of Apple Computer, Inc.

Handle Graphics®, MATLAB®, Real-Time Workshop®, Simulink®, Stateflow®, and xPC TargetBox® are registered trademarks, and TargetBox™ and Target Language Compiler™ are trademarks of The MathWorks, Inc.

Tektronix® and Tek are registered trademarks of Tektronix, Inc.

Other product and company names mentioned herein are trademarks or trade names of their respective companies.

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

#### Patents

For patents covering National Instruments products, refer to the appropriate location: **HelpPatents** in your software, the patents.txt file on your CD, or <u>ni.com/patents</u>.

# WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION. INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR

CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

### **Technical Support and Professional Services**

Visit the following sections of the National Instruments Web site at ni.com for technical support and professional services:

- <u>Support</u>—Online technical support resources at ni.com/support include the following:
  - Self-Help Resources—For answers and solutions, visit the award-winning National Instruments Web site for software drivers and updates, a searchable <u>KnowledgeBase</u>, <u>product</u> <u>manuals</u>, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on.
  - Free Technical Support—All registered users receive free Basic Service, which includes access to hundreds of Applications Engineers worldwide in the <u>NI Discussion</u> Forums at ni.com/forums. National Instruments Applications Engineers make sure every question receives an answer.

For information about other <u>technical support options</u> in your area, visit ni.com/services or <u>contact</u> your local office at ni.com/contact.

- <u>Training and Certification</u>—Visit ni.com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- <u>System Integration</u>—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your <u>local office</u> or NI corporate headquarters. You also can visit the <u>Worldwide Offices</u> section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

#### **Resource Allocation**

The following tables include information about using multiple resources on NI motion controllers. Click the links to view descriptions of the resource allocation.

#### **Default Resource Allocation**

#### Resource Allocation for Dual-Loop Feedback and/or Sinusoidal Commutation

- **Tip** Refer to *Load Commutation Parameter* in the *NI-Motion VI Help* or *NI-Motion Function Help* for information about setting commutation parameters for an axis.
- **Tip** Refer to *Configure Axis Resources* in the *NI-Motion VI Help* or *NI-Motion Function Help* for information about configuring resources for an axis.

### **Branch Offices**

Office	Telephone Number
Australia	1800 300 800
Austria	43 662 457990-0
Belgium	32 (0) 2 757 0020
Brazil	55 11 3262 3599
Canada	800 433 3488
China	86 21 6555 7838
Czech Republic	420 224 235 774
Denmark	45 45 76 26 00
Finland	385 (0) 9 725 72511
France	33 (0) 1 48 14 24 24
Germany	49 89 7413130
India	91 80 41190000
Israel	972 0 3 6393737
Italy	39 02 413091
Japan	81 3 5472 2970
Korea	82 02 3451 3400
Lebanon	961 (0) 1 33 28 28
Malaysia	1800 887710
Mexico	01 800 010 0793
Netherlands	31 (0) 348 433 466
New Zealand	0800 553 322
Norway	47 (0) 66 90 76 60
Poland	48 22 3390150
Portugal	351 210 311 210
Russia	7 495 783 6851
Singapore	1800 226 5886
Slovenia	386 3 425 42 00
South Africa	27 0 11 805 8197
Spain	34 91 640 0085
Sweden	46 (0) 8 587 895 00
Switzerland	41 56 2005151
Taiwan	886 02 2377 2222
Thailand	662 278 6777
Turkey	90 212 279 3031
United Kingdom	44 (0) 1635 523545
United States (Corporate)	512 683 0100