

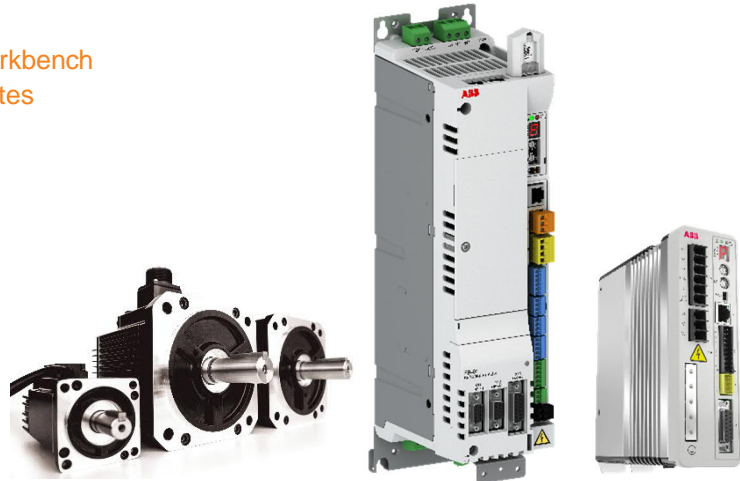
# Application note

## Tuning drives for servo motors using Mint Workbench

AN00250

Rev A (EN)

Setting up ABB motion drives using the Mint Workbench  
Commissioning Wizard results in motion in minutes



### Introduction

The purpose of this application note is to explain the steps required to get an ABB motion drive operating in conjunction with an AC servo motor and 'tuned' to achieve the best possible performance. Although some references are made to induction motor parameters, setup and tuning of induction motors is covered by a separate application note.

### MicroFlex e190 and MotiFlex e180 motion drives

The MicroFlex e190 and MotiFlex e180 are sophisticated servo drives which have been designed to meet the requirements of both de-centralised single axis intelligent solutions and centralised multi-axis systems, providing high performance in an economically priced device.

They provide an ideal solution to many machine applications and are ideal components for machine builders and system integrators.

In this application note we will describe the commissioning process for a MotiFlex e180. The process is almost identical for a MicroFlex e190 drive so references in this document to MotiFlex e180 are generally interchangeable with MicroFlex e190 unless stated otherwise.

### Equipment required:

- MotiFlex e180 servo drive running the latest firmware, with the correct feedback option module fitted
- BSM motor with either Incremental encoder, SSI, EnDat, SinCos, Hiperface, SmartAbs, Resolver<sup>1</sup> or BiSS feedback
- PC running Windows 7 or later
- Motor feedback cable terminated with suitable connectors at both ends
- Motor power cable with a suitable connector at the motor end
- Mains supply to the MotiFlex e180 (3 phase 200-480VAC or 1 phase 230VAC is also possible with frame sizes A and B providing the PHASELOSSMODE parameter is set to 0)<sup>2</sup>
- Ethernet Cable

<sup>1</sup> If using MicroFlex e190 a resolver based motor will require the use of a resolver adapter OPT-MF-201

<sup>2</sup> If using MicroFlex e190 the supply should be either 1 phase 115-240VAC or 3 phase 115-240VAC

## Contents

<b>1</b>	<b>Overview</b>	<b>3</b>
1.1	Recommended I/O connections	3
1.2	Feedback devices	4
1.3	Supply and motor power connections	5
1.4	Connecting a PC	5
<b>2</b>	<b>Commissioning wizard</b>	<b>7</b>
2.1	Entering application data	7
2.2	Viewing and changing parameters	15
2.2.1	Changing motor direction	15
2.2.2	Motor brake control	15
2.3	Understanding control rate and control loops	16
<b>3</b>	<b>Tuning the motor</b>	<b>18</b>
3.1	Checks to make before tuning	18
3.2	Initial tuning	18
3.3	Off load tuning	20
3.3.1	Measure stator resistance and inductance	21
3.3.2	Calculate current loop gains	21
3.3.3	Test the feedback	21
3.3.4	Measure the voltage constant	22
3.4	On load tuning	22
3.4.1	Measure the inertia	25
3.4.2	Calculate the speed and position loop gains	25
3.5	Fine tuning	25
3.5.1	Fine tuning the current loop	27
3.5.2	Fine tuning the velocity loop in velocity servo configuration	28
3.5.3	Fine tuning the position loop in velocity servo configuration	30
3.5.4	Fine tuning the position loop in torque servo configuration	34
3.6	Manual tuning	37
3.6.1	Manual tuning of the current loop	37
3.6.2	Manual tuning of the velocity loop in velocity servo configuration	38
3.6.3	Manual tuning of the position loop in velocity servo configuration	40
3.6.4	Manual tuning of the position loop in torque servo configuration	41
<b>4</b>	<b>Debugging and troubleshooting during tuning</b>	<b>42</b>
4.1	Debugging 'Measure stator resistance and inductance'	43
4.2	Debugging current loop tuning	43
4.3	Debugging 'Test the feedback'	43
4.4	Debugging 'Measure the voltage constant'	44
4.5	Debugging 'Measure the inertia'	45
4.6	Mechanical considerations	45
4.6.1	Inertia mismatch	45
4.6.2	Mechanical compliance and Adding torque filters	46
<b>5</b>	<b>Finalising the commissioning</b>	<b>49</b>
5.1	Further support	49

## 1 Overview

### 1.1 Recommended I/O connections

If there is any kind of safety requirement in the process then use of the two Safe Torque Off (STO) inputs on the drive which provide a TUV certified SIL 3 STO functionality should be made. Apart from these STO connections it is not essential to add any additional control connections although it is common to include a drive enable input when using the drive in combination with a motion controller providing hardwired analog and/or digital control connections (e.g. NextMove ESB-2).

The digital inputs on the drive allow connection of devices such as a home sensor or limit switches. Two of these inputs (1 and 2) can be used for high speed registration, capturing position in the order of nS to  $\mu$ S (depending on feedback type). Digital outputs can be configured for functions such as 'drive ready' and the relay output (DO4) can be used for functions such as motor brake control (remembering to follow the instructions in the installation manual to use an isolated 24Vdc supply for the motor brake circuit via this relay).



Note: The MicroFlex e190 does not have a relay output so if motor brake control is required with this product then an external intermediate relay will be required.

#### Drive enable signal:

The drive will need to be enabled to allow it to control the motor. The source of this enable signal can be defined during the commissioning wizard and can also be set using the Mint keyword 'DRIVEENABLEMODE' which defines if the drive is enabled via a hardware input only or whether some form of software interlock (e.g. a Mint program or a communications control word) is also included. If DRIVEENABLEMODE will be set to include a hardware input (other than the STO inputs) then the DRIVEENABLEINPUT keyword should be used to specify which input channel is used for this function during the commissioning process (described later). Drives running as intelligent single axis systems (i.e. running a Mint program) and Real-time Ethernet controlled drives (i.e. EtherCAT or Ethernet Powerlink) do not typically use a dedicated drive enable input and tend to rely only on the STO inputs for the hardware control of the drive's power stage.

#### Motor brake control output:

If the motor you are using has a brake then the drive has the ability to control a motor brake via any of the available digital outputs (Note that by default output 0 is pre-configured as a GLOBALERROROUTPUT – i.e. a drive status output – so it is common to use an alternative output for brake control). The OUTPUTACTIVELEVEL keyword should be used to ensure the specific output channel selected to control the motor brake is configured to be active low. Motor brake control is configured via drive parameters dedicated to this function (these keywords all start with the text MOTORBRAKE). It's important to consider where the 24Vdc supply for the brake will come from. This should be from a separate isolated power supply from that which is used to supply logical devices such as the drive control card, and other control equipment.

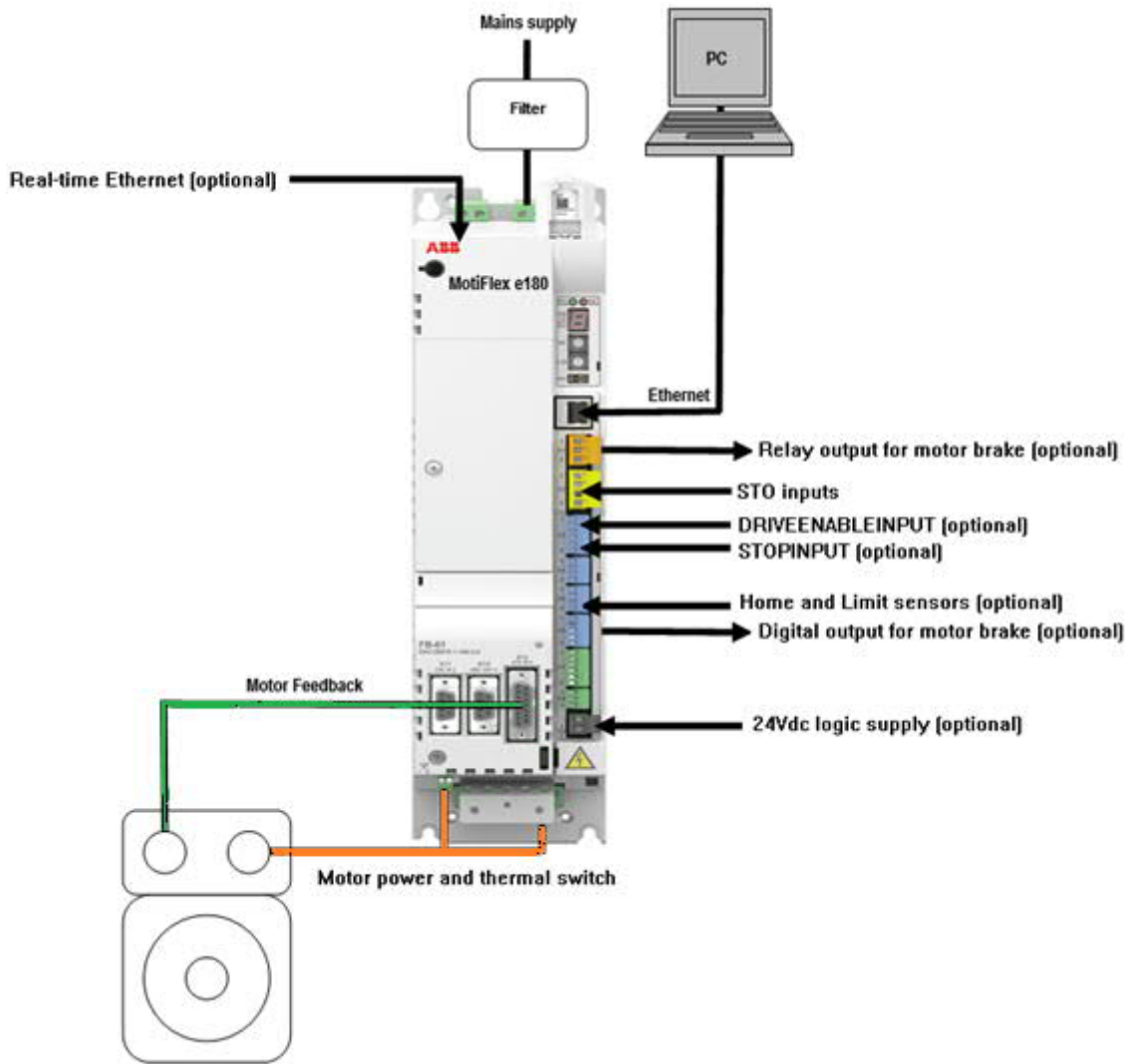
#### Emergency stop input:

When either or both of the SIL3/PLe STO inputs are activated the drive will turn off the power to the motor immediately. If the load has any kind of momentum and/or high inertia it may continue to 'free wheel' which may not be desirable. For this reason sometimes it is required for the drive to quickly decelerate the motor, in a controlled manner, before activating the STO function. If you have a programmable drive (with the Mint memory module option +N8020) it is possible to configure a STOPINPUT on the drive which can provide a means of quickly decelerating the drive. A contact from the system's safety circuit that opens instantly would be wired to the drive's configured STOPINPUT. For a drive controlled via a Real-time Ethernet network it is typical to connect this signal to the motion controller instead. Removal of the STO inputs would be delayed to allow the drive (or motion controller) time to profile the axis to a controlled stop before finally disabling the motor output via the STO function. Actual implementation must be determined by the user according to the result of a risk assessment performed on the system – please refer to application note AN00206, available on the ABB motion support website, for further details

#### Motor thermal switch:

The motor may be supplied with a sensor/switch that either opens completely or provides a high impedance when the motor temperature limit is reached. This should be wired directly to the dedicated motor temperature switch input (X10) on a MotiFlex e180 drive. If using a MicroFlex e190 drive use of this temperature sensor is optional, but if it is used it should be wired using an isolated 24Vdc supply (this can be the same supply used for a motor brake) and then included into the control scheme via an intermediate relay, contacts of which can either be used to disable the drive in some way or fed into the MicroFlex e190 as an input assigned to be the MOTORTemperatureINPUT.

The diagram below illustrates most of the possible connections to a MotiFlex e180 drive, please refer to the relevant drive (and motion controller if one is used) installation manual for full details:



**1.2 Feedback devices**

Motor feedback is essential for accurate “closed loop” control for both the speed and position loops that are built in to the drive. There are a number of different feedback devices commonly fitted to motors. For a brushless AC servo system the positional information is used to commutate the motor. The speed is derived by measuring the rate of change of position. A ‘+code’ must be added when ordering the MotiFlex e180 (+L5xx) to determine which of the 4 optional feedback modules will be fitted to suit the chosen feedback type;

- FB-01 (+L517): Incremental encoder with Halls(X13)**
- FB-02 (+L518): Serial interfaces + SinCos (X13)**
- FB-03 (+L516): Resolver (X13)**
- FB-04 (+L530): DSL (X13)**

MicroFlex e190 drives use a Universal Encoder Input (UEI) that supports a wide variety of encoder types, as well as resolver feedback via the resolver adaptor module OPT-MF-201.

### 1.3 Supply and motor power connections

When connecting the motor and supply power cables to the drive we must pay special attention to the termination of earths and power cable shields. Earths should be connected to the designated earth connections and shields should ideally be clamped using p-clips to the hardware provided with the drive for Electromagnetic Compatibility (EMC) purposes (e.g. the shield clamp plate provided with MotiFlex e180 or the threaded holes on the MicroFlex e190 heat sink). Please see the section 'Electrical Installation' in the drive's user manual for more information. In addition to the required AC mains supply an additional 24Vdc logic supply is typically a consideration if the control card is to be kept powered up when the mains has been turned off (e.g. to keep the control network in a healthy state).

### 1.4 Connecting a PC

As our programming/commissioning connection is achieved via standard Ethernet (TCP/IP) the configuration of the PC's network adaptor must be considered. A new drive (or any drive where dip switch SW1.1 is in the ON position) will be configured with the IP address 192.168.0.1, so to connect to it your PC must be configured to be able to access the 192.168.0 subnet. This is usually achieved by setting the PC's network adaptor to 192.168.0.[anything other than 1] and using a subnet mask of 255.255.255.0 (to ensure that only devices on 192.168.0 are accessible).

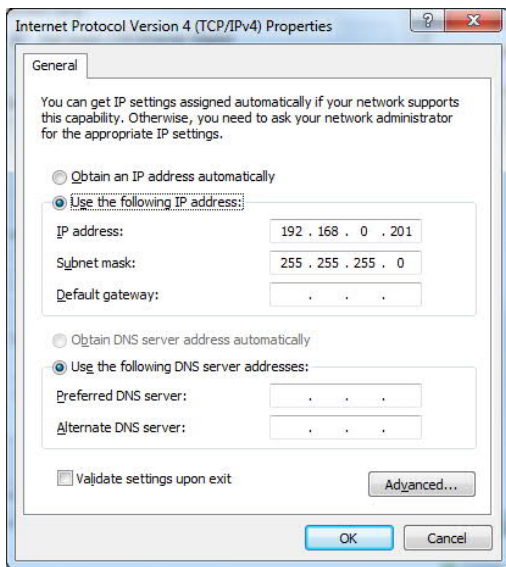
To access the network adaptor settings on your PC go to;

Control Panel > Network and Sharing Center > Change adapter setting > [Select network adaptor you are using] >

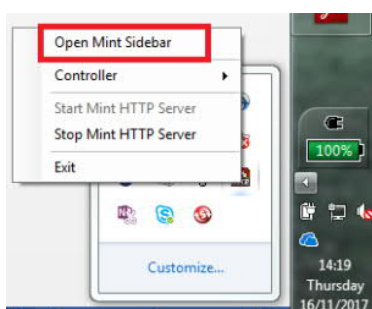


Right-click the adaptor and select 'Properties'.

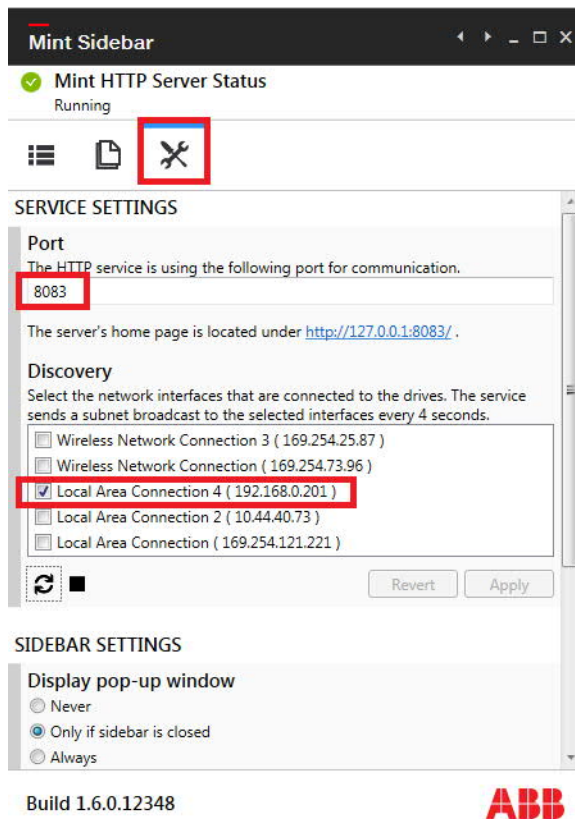
Select Internet Protocol Version 4 (TCP/IPv4) > Properties > Select 'Use the following IP address' > Add in the required IP configuration for your PC, for example...



Click OK and close the network adaptor properties dialog. Now the network adaptor is configured to use the same subnet as the drive it is possible to connect to the drive, however, for "Discovery" and access to Silverlight pages served by the drive (e.g. the network configuration screens) it is also necessary to configure the Mint HTTP server tray application / Mint Sidebar. The Sidebar runs as a service and is configured to run automatically when Mint Workbench is installed. Find the Mint HTTP server icon in the Windows system tray (typically in the bottom right corner of the screen) and right click this to access the Mint Sidebar...

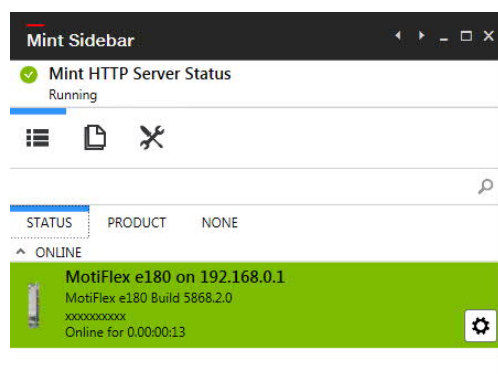


Once the Mint Sidebar is open it will show a list view of all connected drives but at this point this is likely to be empty. We must first tell the Sidebar which network adaptor should be used for 'Auto discovery' of connected motion products.



Click the settings icon to access the Service Settings page. On this page you can adjust the port number used for the HTTP service (8080 is the default but this is used by many Ethernet based applications so it may be necessary to change this to a different, less commonly used port, 8083 for example, the Sidebar will inform you if the port is already in use). You should see the PC's network adaptor you previously configured listed in the Discovery section (if not, click the refresh button underneath this list). Make sure you tick this adaptor.

If everything is working correctly and you connect an Ethernet cable between your PC and the drive's front Ethernet port (E3) and you switch back to the Controller List page you should see your connected drive (in a green box)



If you can't see the drive in the Sidebar recheck your adaptor and Sidebar settings. If you are sure the settings are correct then there could be a Firewall or Anti-Virus program preventing the HTTP server from accessing the required ports and protocols. Check with your IT department that it is possible for your PC to access the following ports:

- TCP Port 80
- UDP Port 5050

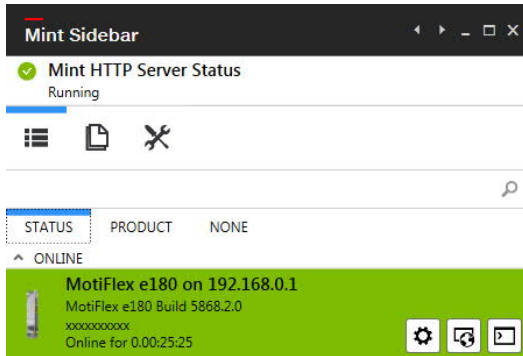
If the Sidebar is unable to discover the drive you can still launch Mint Workbench from the Windows Start menu and use “Add Specific Controller” to establish a connection to the drive, but you will be unable to access the ‘Configuration’ or relevant Real-time Ethernet pages in Workbench (these pages aren’t needed though to setup and commission a motor).


By default the Sidebar displays a button to allow Workbench to be launched once the drive has been discovered. To gain access to the drive’s web page and trace log (used for more advanced system diagnostics) switch back to the Service Settings page, scroll down and tick the Web page and Trace log options as shown below...

#### Show open button for

- Web page  
 Trace log

The Sidebar will now show the additional buttons in the Controller list...



You can now launch Workbench from the Sidebar controller list page, by clicking on  (or you can start Workbench in the normal fashion via the Windows Start menu).

## 2 Commissioning wizard

The Commissioning Wizard procedure we will run through shows a sequence of screens displaying the current drive settings and requesting information required to commission the drive and motor. When each page is complete click the Next button to continue through the Wizard. If Workbench is launched from the Sidebar the wizard must be launched from the Tools>Commissioning Wizard menu option. If Workbench is launched from the Windows Start menu you have the option to select “Launch Commissioning Wizard” as part of the controller selection process to start the wizard automatically. If you need to access the Mint Workbench command window at any time during the commissioning process simply right-click any blank area of the screen and select ‘Command Line’ from the resulting right-click menu.

### 2.1 Entering application data

#### Welcome page

The first screen describes the process and offers the opportunity to set all drive parameters to the factory default settings. Make sure the ‘Reset device to defaults’ option is selected and click the Next button. This will reset all drive parameters, except communication settings, to factory settings. If there is a Mint program on the drive this will be retained. If you have decided to re-use the wizard to check or modify an existing setup then be sure to deselect the ‘Reset device to defaults’ option.



Note: This action can also be achieved by typing ‘SYSTEMDEFAULTS’ in the command window at any time. If you wish to reset to the drive to factory defaults before commissioning, i.e. reset the communication settings, delete the Mint program, reset non-volatile data value as well as all drive parameters, you can type ‘FACTORYDEFAULTS’ at the Workbench command window.


#### Select motor type

Select the correct motor type from the list; AC brushless rotary, AC Brushless Linear, Induction (Rotary) or Induction (Linear) and click the Next button.

### Select motor

Workbench has a repository of ABB motor data stored within it so in this page we can select an ABB motor via either Catalog number or Specification number. When doing so Workbench will tell you the meaning of the part code selected. In the example below the motor selected is a BSM60R-240MT AC rotary servo motor;

Alternatively the data for a motor that is not in the repository or data for a “third party or Custom motor” can be entered manually although this may require more information from the motor manufacturer if the required data is not on the motor nameplate. Click the Next button.

 Note: The minimum amount of information that is required for a custom AC servo motor is the continuous stall current, peak current, maximum speed and number of poles. The Auto tune process can measure all of the other motor properties. For induction motors the minimum amount of information is rated current, peak current, rated voltage, rated frequency, rated speed, power factor, number of poles and magnetising current. Again the Auto tune process can measure all of the other motor properties

### Confirm Motor Information

The next screen allows you to confirm the motor and drive information. For an ABB motor this data will just be for information as the data is read from the information repository based on the motor specification or catalog number provided.

If you have selected a custom motor here is where you enter the data for it. The Motor Details section is free text so here you can record the part code. The Motor Nameplate Parameters are, as the name suggests, values that may typically be found on the motor nameplate. If possible enter all of these, but it is possible to leave voltage constant, inductance and resistance set to 0 if these are not known. When a field is selected text at the bottom of the screen tells you whether this value can be measured by the drive during auto tuning...

**Confirm Motor Information**

For a motor chosen from the database, confirm that the data matches that of your motor.  
For a custom motor, enter the data below. Some parameters can be set to zero here with the values being calculated during the autotune process.

<b>Motor Details</b>		
Spec. Number		
Cat. Number	BSM60R-240MT	
<b>Motor Nameplate Parameters</b>		
Continuous stall current	1.60	Amps (RMS)
Peak current limit	5.00	Amps (RMS)
Voltage constant	47.35	Vpk/kRPM
Inductance (L-L)	16.00	mH
Resistance (L-L)	5.00	Ohms
Max Speed	5000	RPM
Number of Poles	8	= 4 pole-pair(s)

**Voltage constant**  
Back EMF constant of the motor in Volts per 1000rpm. If this value is zero it can be calculated by the drive during autotuning.

< Back   Next >   Finish   Cancel   Help

Once the data has been checked and/or entered, click the Next button.

### Confirm Drive Information



The next screen allows you to confirm the drive rating zone and for certain drive sizes the switching frequency for the application. The default is for a 200% overload (for 3 seconds), the Pulse Width Modulation (PWM) frequency will depend on the drive size. You can change these settings to suit the application requirements by clicking on the box containing the required switching frequency and overload settings.

**Confirm Drive Information**

Confirm drive data. If applicable, choose a control type and drive rating zone.

Drive Data			Drive Rating Zone. From the table of switching frequency against percentage overload and duration, select the required continuous and (peak) current rating.						
Type	MotiFlex e180		Overload & Duration	300%, 3s	200%, 3s	150%, 60s	110%, 60s		
Peak Current	6.00	Amps (RMS)		Switching Frequency	8 kHz	2A (6A)	3A (6A)	3A (4.5A)	3A (3.3A)
Rated Current	3.00	Amps (RMS)				3A (6A)	3A (4.5A)	3A (3.3A)	
Bus Volts	640	Volts							
Motor Data									
Peak Current	5.00	Amps (RMS)							
Rated Current	1.60	Amps (RMS)							
Drive Data Information about the type and rating of the drive.									

Many motors have peak current ratings that are 3 to 4 times their RMS rating, so a 300% setting may allow the drive to extract the maximum performance from the motor. 150% and 110% ratings are not typically used for servo type applications, the overload (and recovery) time is more suited to industrial applications utilising induction motors.

Ensuring use of the latest firmware will provide improved 8kHz ratings for MotiFlex e180, the 4kHz and 8kHz ratings are then identical. Consideration should be given to the required maximum motor speed when selecting the PWM frequency. ABB suggest a minimum PWM: $f_{\text{motor}}$  (motor frequency) ratio of 32:1 for optimum current control.

Example:

Motor synchronous speed is found from  $v = 120 * f_{\text{motor}} / p$  (where  $v$  is speed in rpm,  $f_{\text{motor}}$  is motor frequency,  $p$  is number of poles)

For 4 kHz and a 32:1 ratio we can find  $f_{\text{motor}}$  from;  $f_{\text{motor}} = 4000 / 32 = 125$  Hz

For a BSM90N motor with 8 poles...

$$\Rightarrow v = 120 * 125 / 8 = 1875 \text{ rpm}$$

So we wouldn't advise running the BSM90N motor above 1875 rpm if 4 kHz switching frequency is used, to maximise the possible speed we would select 8 kHz instead. It is possible to run faster, but as the PWM: $f_{\text{motor}}$  ratio decreases the current stability will worsen (if the ratio falls below 16 you may experience intermittent overcurrent trips on the drive).

For these reasons it is more common to see 4 kHz selected when using induction motors where the maximum speed is often much lower. Note also that increasing the PWM frequency will lead to increased heating of the motor, so check that the motor is designed to cope with 8 kHz operation.

### Motor Feedback

In this screen we either define or check the information for the selected feedback device. In this example the motor has SmartAbs (Serial, single turn, absolute) feedback connected as was defined by the motor part number selected earlier (BSM60R-240MT). Other feedback types will have a different setup screen to the one shown.

## Motor Feedback

Please confirm that the feedback type and resolution are correct for your motor. No feedback type is required for V/f control. If no feedback device is fitted, select None.

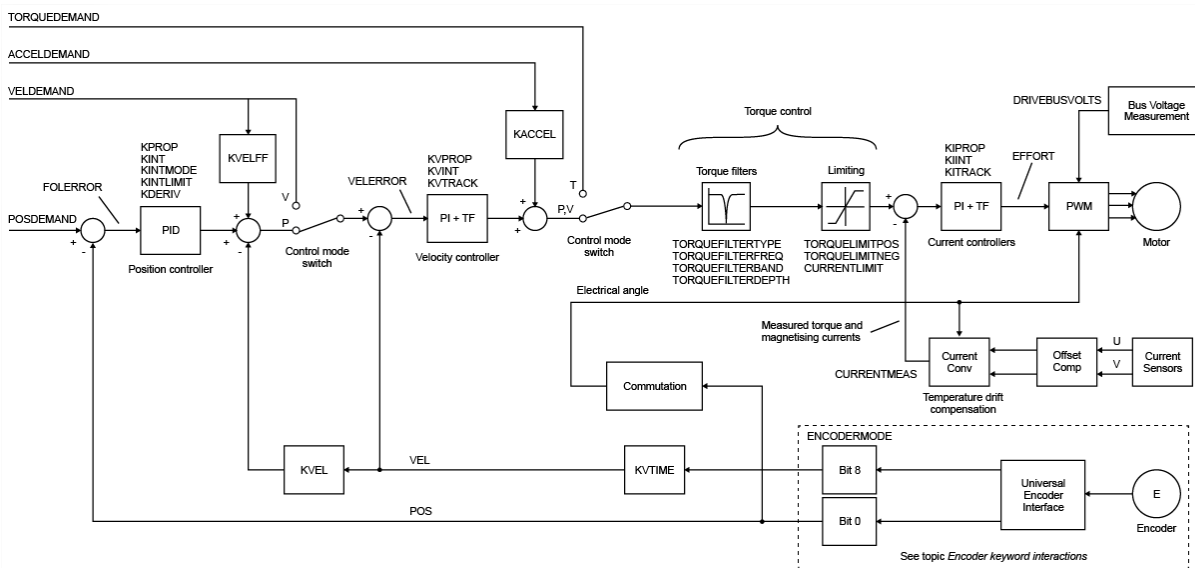
<b>Motor Feedback</b>		
Encoder Channel	Encoder 0	
Feedback Type	Smart Abs	
<b>Feedback Options</b>		
Encoder Info	Unable to read information from feedback device	
Single-turn Bits	17	bits (131072 counts per rev)
Multi-turn Bits	0	bits (1 revs)
Encoder Pre-scale	1	= range of -64.00 to 64.00 revs
Effective Resolution	131072	counts per rev
Battery Fitted	No	
<b>Simulated Encoder Output 0</b>		
Encoder source channel	None	
Resolution	1000	x 4 = 4000 counts per motor rev

For an axis requiring an absolute position range the 'Encoder Pre-scale' parameter (ENCODERPRESCALE in Mint) can be used to adjust the effective resolution. The screen shows both the number of motor revs that are possible before the absolute position range is exceeded as well as the effective resolution that results from a combination of the encoder's resolution and the encoder pre-scale setting. Encoder pre-scale can be adjusted from 1 to 128. If the pre-scale is changed in the future it will be necessary to reset the application maximum speed parameter (DRIVESPEEDMAX) as well as the speed and position loop gains, so it is a good idea to determine whether any pre-scaling of the encoder signal is needed or not at this stage.

From here we can also define the motor simulated output source channel and resolution. The resolution is the number of pulses per encoder revolution and the 'Encoder source channel' is which encoder on the drive the simulated encoder output will use as a reference. The choices for this are 0, 1 or 2, corresponding to encoder channel 0 (the Universal Encoder Input), 1 (the encoder input formed by the drive's fast digital inputs) or 2 (the secondary/master encoder input). In most cases the motor feedback is connected to the Encoder channel 0 and a setting of 0 will therefore emulate the feedback device fitted to the motor. Click on Next when ready.

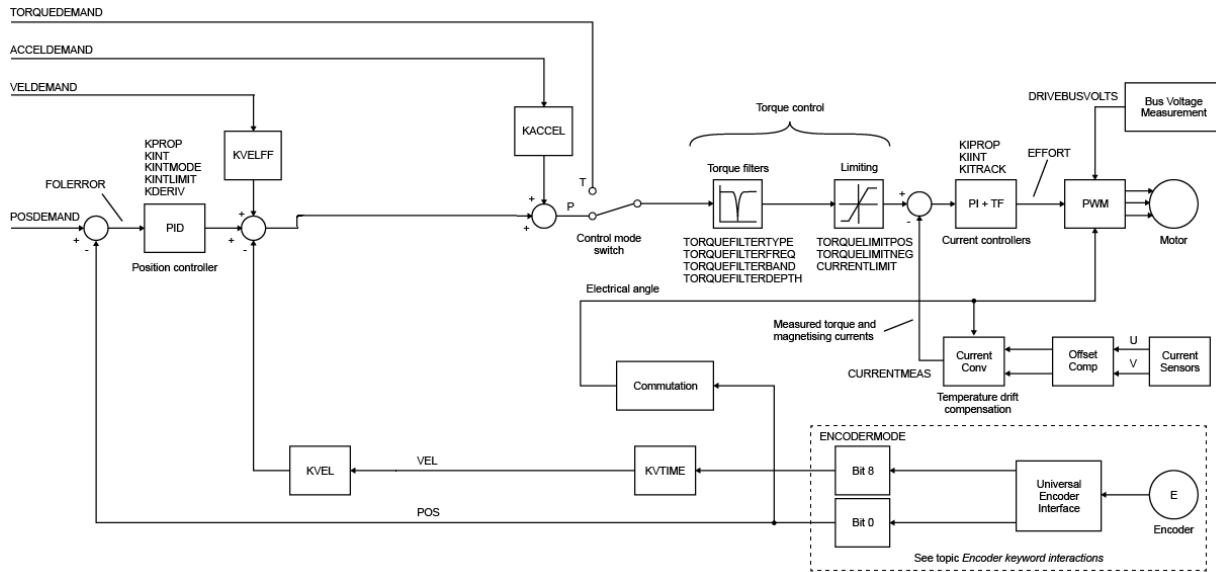
## Setting the servo configuration

By default the drive will operate as a "Velocity servo drive". This means that if the position loop is used the output of the position loop PIDVF controller will be a velocity demand that is then passed to the drive's PI velocity loop controller. The output of the velocity loop controller is then passed as a current demand to the PI current controller. "Velocity servo drive" is set by setting the drive parameter CONFIG(0) to 1 (or \_cfSERVO). The block diagram below illustrates the control structure for velocity servo drive configuration...



This setting is suitable for most general purpose applications, however, for applications requiring stringent accuracy and very low settling times it may be more suitable to use the drive as a "torque servo drive". Exceptions to this are applications where the velocity loop is essential (e.g. applications where the Mint VELREF motion command is used or the motion controller is operating in CSV mode).

The block diagram below illustrates the control structure for torque servo configuration:



If you feel that the application may benefit from the use of torque servo operation then right click anywhere in a blank area of the current Workbench screen and select the 'Command Line' menu option. In the command window now enter...

```
CONFIG(0) = 6 (or _cfTORQUE_SERVO)
```

If you aren't sure, leave the CONFIG set to its default setting of 1, it can always be changed at any later point in time. Remember that until prompted to save drive parameters (or unless Tools>Store Drive Parameters is selected) all drive settings are volatile and power cycling will return the drive to the last stored values.

**Select Operating mode and source**

The drive has three control loops; position, speed and current (torque). We therefore have the option to run the drive in either Position, Velocity or Torque control mode (as a default setting). The selection here should be made to suit the application. If the drive is operating as a network drive (i.e. via EtherCAT or Ethernet Powerlink) the selection will be based on whether the network master (i.e. the motion controller) is operating in either Cyclic Synchronous Position (CSP), Cyclic Synchronous Velocity (CSV) or Cyclic Synchronous Torque (CST) mode. If the drive is operating as an analog drive (receiving a +/-10Vdc demand from a motion controller) then the selection is likely to be either Velocity or Torque. If the drive is running a local Mint program the selection is likely to be Position.

In this instance we will select "Position" for the control mode.

Control Mode	<input type="radio"/> Velocity <input type="radio"/> Torque <input checked="" type="radio"/> Position
--------------	---



A Control Mode of Velocity is not valid when the servo configuration is set to 6 (\_cfTORQUE\_SERVO). Attempting to use this combination will result in no motion occurring.

The drive enable input mode determines what will happen if the drive enable input is removed whilst the drive is enabled.

Drive Enable Input Mode	<input checked="" type="radio"/> Crash stop, disable, no error <input type="radio"/> Crash stop, disable, error
-------------------------	--

For applications using an analog motion controller (e.g. NextMove ESB-2) the drive will be enabled/disabled as a normal part of the operation of the system (using an output on the motion controller) so in this case we would want to avoid the drive going into error/fault every time the enable is removed and we would select the "no error" option. This is also the typical setting for drives operating in real-time Ethernet mode. Only select the "error" option if you want the drive to enter the fault state every time the drive enable input is removed.



By default a DRIVEENABLEINPUT is not configured (as most applications will only use the STO inputs). If you need to define this input (e.g. if replacing an older analog drive such as a Baldor FlexDriveII or ABB Analog MicroFlex) then switch to the command window and enter DRIVEENABLEINPUT = n (where n is the input channel is used)

The Drive Enable Mode allows the user to select how the drive is enabled, allowing options to enable the drive from only a hardware enable input through to control of the enable via an on-board Mint program. Please refer to the Mint Help file topic DRIVEENABLEMODE for full details.

Drive Enable Mode	<input checked="" type="radio"/> Software enable <input type="radio"/> Cancel + Auto enable <input type="radio"/> Hardware enable <input type="radio"/> Auto enable <input type="radio"/> Cancel + Hardware enable
-------------------	--

For intelligent and network drive applications the 'Software enable' selection is typical. For analog drives the 'Hardware enable' option is most likely.

The reference source should be set to suit the application. The options are;

- Direct (Mint program/Profinet/ModbusTCP/EthernetIP/PC application control),
- RT Ethernet (EtherCAT or Ethernet Powerlink)
- Analog Input (Velocity or Torque reference via an analog input selected in Control Ref. Channel)

<b>Reference Source</b>	
Control Ref. Source	<input checked="" type="radio"/> Direct (Host/Mint) <input type="radio"/> RT Ethernet (CiA402) <input type="radio"/> Analog Input (direct)
Control Ref. Channel	Analog Input 0

**Application limits**

The application limits define the specific current and speed limits of the system and the behaviour of the drive in the event of an overload. The application peak current setting is automatically set to the lower of the motor or drive peak current ratings. It cannot be set higher than the lowest of these, but it can be set lower if there is a need to limit the torque applied to an element of the mechanical transmission between the motor and the load.

<b>Current Limit</b>		
Motor Rated Current	1.600	Amps (RMS)
Motor Peak Current	5.000	Amps (RMS)
Drive Rated Current	2.300	Amps (RMS)
Drive Peak Current	4.600	Amps (RMS)
App. Peak Current	4.600	Amps (RMS)

The motor and drive overload responses can also be set here. By default the drive will trip on any overload condition. Sometimes it may be desirable to limit the torque by folding back the current rather than putting the drive into fault, for example if the load is intended to be driven into an end stop. If the motor thermal switch is utilised it is also possible to ignore the drive's I2t motor overload algorithm completely (and just rely on the thermal switch to disable the drive). This can squeeze a little more performance out of the system, particularly if the motor is mounted onto something that acts as a large heatsink which the drive would not know about.

<b>Overload Protection Function</b>	
Motor Overload Action	<input type="radio"/> Ignore <input checked="" type="radio"/> Trip drive <input type="radio"/> Foldback current
Drive Overload Action	<input checked="" type="radio"/> Trip drive <input type="radio"/> Foldback current

At this point it is necessary to have a feel for how fast the motor is required to move in the application. If you don't know what the application max speed is likely to be at this stage then set this at approximately 75% of the max motor speed. If this value is changed later, after tuning has been performed, it will be necessary to recalculate speed and position loop gains, so try to set it correctly now if at all possible.

Application Max Speed		
App. Max. Speed	4000	RPM
Over Speed trip level	110	%
Max Motor Speed	5000	RPM
Max Theoretical Speed	7500	RPM

For applications that are designed to run at slower speeds it is advisable that the ‘Application Max Speed’ is reduced to maximise the speed resolution available, this will improve speed accuracy at lower speeds. But always set this parameter to a value slightly in excess of the maximum required to avoid unwanted errors. It is best not to set this below 500 rpm.



For very slow running applications it is advisable to select a motor with high resolution feedback (e.g. EnDat, SSI, SmartAbs or BiSS)

### Select scale factor

The position, velocity and acceleration user units are set here. If they are left as default (1.00) then all positions, velocities and accelerations used locally in the drive will be in encoder counts, but this is rarely useful in the real world. It is more common to enter values here that will scale these properties into units that will be used in the application (e.g. number of encoder counts in one mm of linear travel, or number of encoder counts in one degree of rotation - taking into account all gears, gearboxes and mechanical factors). The unit box is a combo-text box so it’s possible to type your required unit name into these if required and then enter the number of encoder counts per unit as shown below...

Select Scale Factor		
Select the units and scale factors you would like to use in your application.		
Scale Factor		
Position User Unit	mm	
Position Scale Factor	320.3	Count(s) per Position User Unit
Velocity Scale Unit	mm/s	
Velocity Scale Factor	320.3	Count(s) / s per Velocity Scale Unit
Accel Scale Unit	mm/s/s	
Accel Scale Factor	320.3	Count(s) / s <sup>2</sup> per Accel Scale Unit

Alternatively the combo-box has a built in selection for ‘revs’ as a unit and selecting this will automatically set the associated scale to the effective resolution that was set earlier in the wizard for motor revs (not necessarily the same as load revs)...

Select Scale Factor		
Select the units and scale factors you would like to use in your application.		
Scale Factor		
Position User Unit	Revs (r)	
Position Scale Factor	131072.00	Count(s) per Position User Unit
Velocity Scale Unit	Revs per second (r/s)	
Velocity Scale Factor	131072.00	Count(s) / s per Velocity Scale Unit
Accel Scale Unit	Revs per second sqd (r/s <sup>2</sup> )	
Accel Scale Factor	131072.00	Count(s) / s <sup>2</sup> per Accel Scale Unit

The values we set here are used for all commands issued locally on the drive during tuning and also affect the scaling of values displayed in the various Spy windows within Workbench. Motion related parameters within a Mint program on the drive (e.g. SPEED, ACCEL, MOVER etc...) would also be scaled by these values, but in this case it is quite common for the Mint program itself to contain the scale factors, which may or may not match the values set here in the drive’s parameters (e.g. someone may decide to tune the drive with it set for ‘revs’ but the Mint program may later operate in ‘linear mm’).

In summary, any motion commands initiated and profiled on the drive directly will use the scale factors set on the drive.

### Profile parameters

Current Control Profile Parameters			Position Control		
Rise Time	0	ms	Max Position Error	0.01	r
Fall Time	0	ms	Idle Position Tolerance	0.01	r
Error Fall Time	0	ms	Idle Velocity	0.2	r/s
Speed and Position Profile Parameters					
Default SPEED	30	r/s			
Accel Time to SPEED	500	ms			
Decel Time to SPEED	500	ms			

The current, speed and position profile parameters shown above are the default values used by the drive whenever it profiles motion for itself (i.e. if it is a Controlled Node (CN) profiled axis on an Ethernet Powerlink network, when it performs fine tuning test moves or when motion is initiated from a Mint program). It is common to set the default SPEED and accel and decel times to something that will likely be typical of the application requirements (however these values can all be adjusted later so it's not critical at this stage). As we scaled our example drive into 'revs' earlier note that the units for SPEED are now shown as 'r/s' (i.e. revs per second).

The 'Position Control' parameters must be set if the drive needs to operate in a position control mode at any point in time for the application. These all have default values that are originally set in counts, so after setting your scale factors earlier these can be fractional to a large number of places. It is common to round these up to the nearest significant digit. Don't decrease the value for 'Idle Velocity' as this will affect how reliably the drive detects the 'Idle' (motion complete) condition.

Max Position Error – sets the maximum deviation allowed between demand and measured position at any time (i.e. following error limit - FOLERRORFATAL). This value is scaled according to the POSSCALEFACTOR set earlier

Idle Position Tolerance – set how close the axis must be to the target position to consider a move to be complete (i.e. axis to be IDLE - IDLEPOS). This value is scaled according to the POSSCALEFACTOR set earlier

Idle Velocity – the speed below which the axis must be travelling in order to consider a move to be complete (i.e. IDLEVEL). The minimum speed resolution is 4000 counts/sec so it is typical to set a value equivalent to 20000 counts/sec or higher (the value used depends on the VELSCALEFACTOR set earlier).

Click "Next" to proceed.

### Analog Input Parameters

The analog inputs can be setup to suit different common configurations, the gain and offset parameters here can also be used to configure how the raw signal is interpreted by the drives on board analog to digital converter.

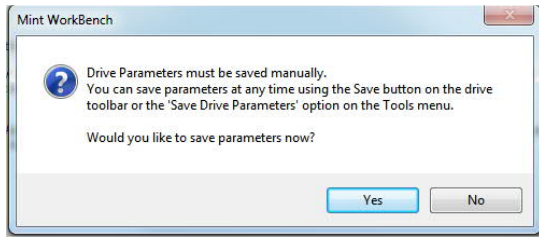
Analog Input 0		
On	<input checked="" type="checkbox"/>	
Common Settings...		
Gain	100.00	%
Offset	0.00	%
Tune Offset		
Current Mode Scaling	0.46	Amps per Volt
Speed Mode Scaling	399.90	RPM per Volt
Filter Time	0.00	ms
Analog Input 1		

If the drive is being used as an analog drive it will be necessary to tune out any offset on the analog input. Ensure the motion controller analog output is zero and click the 'Tune Offset' button. The drive will measure the analog input and set an appropriate offset to ensure a measured analog demand of zero.

If the drive is being used as an analog drive do not set a 'Filter Time', if however the analog input(s) are being used for general purposes (e.g. a reel diameter sensor may be wired as an analog input) you may wish to consider adding a filter to smooth the input signal and negate the effects of any electrical noise.

Click "Next" to proceed.

At the end of the wizard the user will be prompted to save the drive parameters that have been configured so far...



Click 'Yes' and Workbench will now automatically open the 'Auto Tuning' page and we are nearly ready to move on to auto tuning the drive.



Note: At power up the drive will initially load all configuration/parameter settings from the stored parameter table. If, however, the drive runs a Mint program it is possible for some, or all, of these to be overwritten at run-time

### 2.2 Viewing and changing parameters

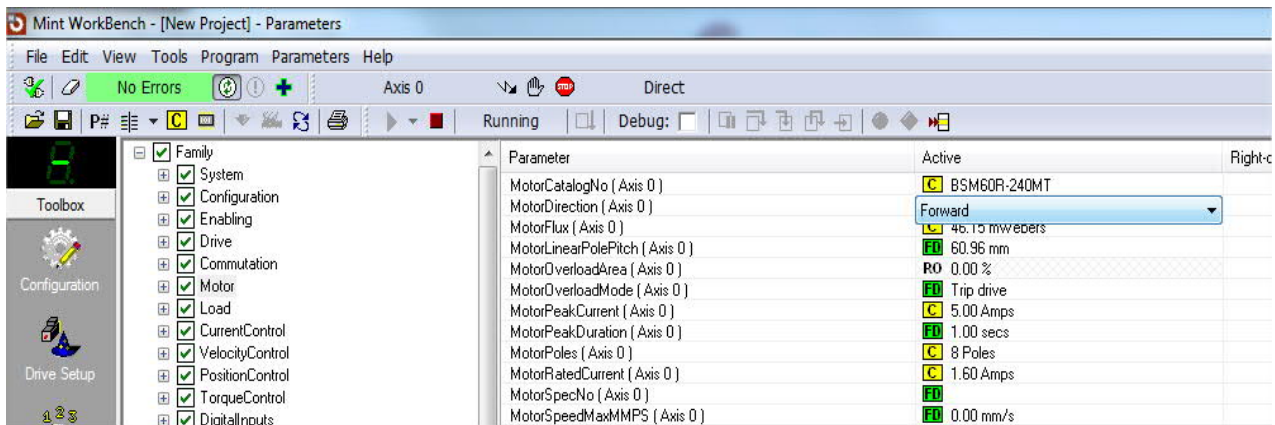


As we have been entering application data and saving this the drive has been setting internal drive parameters. As we continue with the auto tuning (and fine tuning) processes we may wish to examine or change some of these parameters (e.g. motor direction for a positive command or motor brake parameters). Drive settings can be changed by editing the drive parameter table directly (by clicking on the Parameters icon in the toolbox on the left hand side of Workbench) or by right-clicking in any blank area of the Mint Workbench and selecting 'Command Line' from the context menu where Mint keywords/parameters can then be entered.

In the following sections we will illustrate the use of the Parameters screen for modifying motor direction and motor brake parameters (the two most commonly used/adjusted settings that aren't an inherent part of the commissioning process).

#### 2.2.1 Changing motor direction

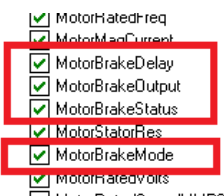
Select Family->Motor from the search tree on the left...



The motor direction parameter is listed in the right hand pane. Click on the active value (Forward or Reverse) and select the opposite direction if required. All of the listed parameters are shown in their Mint keyword form, so MOTORDIRECTION is a keyword that can be set in the Mint Workbench command window (MOTORDIRECTION = 0 or MOTORDIRECTION = 1).

#### 2.2.2 Motor brake control

Select Family/Motor and expand the parameter tree. In the list of motor related parameters you will find a number of motor brake related settings as shown below...



Motor brake delay comprises three separate channels:

- Channel 0 is the time allowed to engage the brake before disabling the drive
- Channel 1 is the time allowed to disengage the brake before starting motion
- Channel 2 sets the delay between enabling the drive and it establishing servo control

Motor brake output sets which digital output should be used as the brake output.

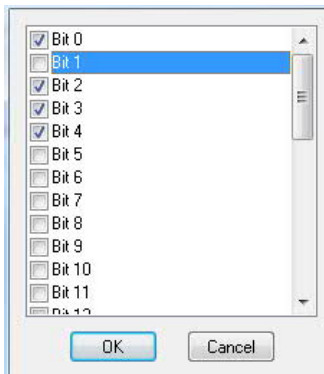


The motor brake output should be configured active low (and the electrical design should ensure the brake is engaged when the output is turned off)

To set the output active level to low either use OUTPUTACTIVELEVEL at the Mint Workbench command window or use Family>Digital Outputs>OutputActiveLevel in the Parameter viewer...



...and then set the appropriate output/bit to active low in the right-hand pane. The example below shows output 1 being set to active low...



Motor brake status just shows the current state of the brake control (i.e. whether the brake is considered to be applied or not).

Motor brake mode allows the user to enable or disable motor brake control.

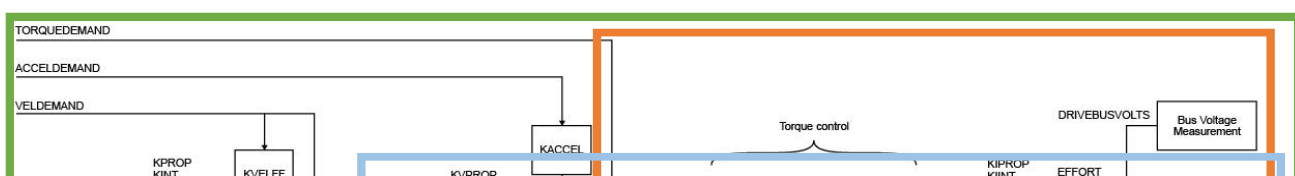
For further information on motor brake control read the Mint Help file: *Hardware->Input/Output Handling->Motor brake control.*

### 2.3 Understanding control rate and control loops

Before we look at the auto tuning process it would be good to understand how the drive makes use of the various internal control loops as this will help us later when fine tuning to achieve the ultimate performance from the motor/drive combination. The default settings for control rate and servo configuration are normally sufficient for the majority of systems using ABB motion products, however there could be some benefits in modifying these to suit the application so we will take a look at these in detail over the next few pages of this document.

#### Velocity servo control loops

The drive contains 3 nested control loops and all 3 of these are used in the default 'velocity servo' configuration (CONFIG parameter set to 1 or \_cfSERVO). The innermost loop is the current control loop (orange). This is enclosed within the speed control loop (blue). The outermost loop is the position control loop (green).





The current loop receives a torque/current demand from the speed control loop. The speed control loop in turn receives a speed demand from the position loop. The position loop generate a speed demand based on the demand and measured positions at any time. Note that if the drive is operating in velocity mode the position loop is switched out and the velocity demand is fed directly to the speed control loop and similarly if the drive is operating in current/torque mode the speed loop is switched out and the torque demand is fed directly to the current control loop.

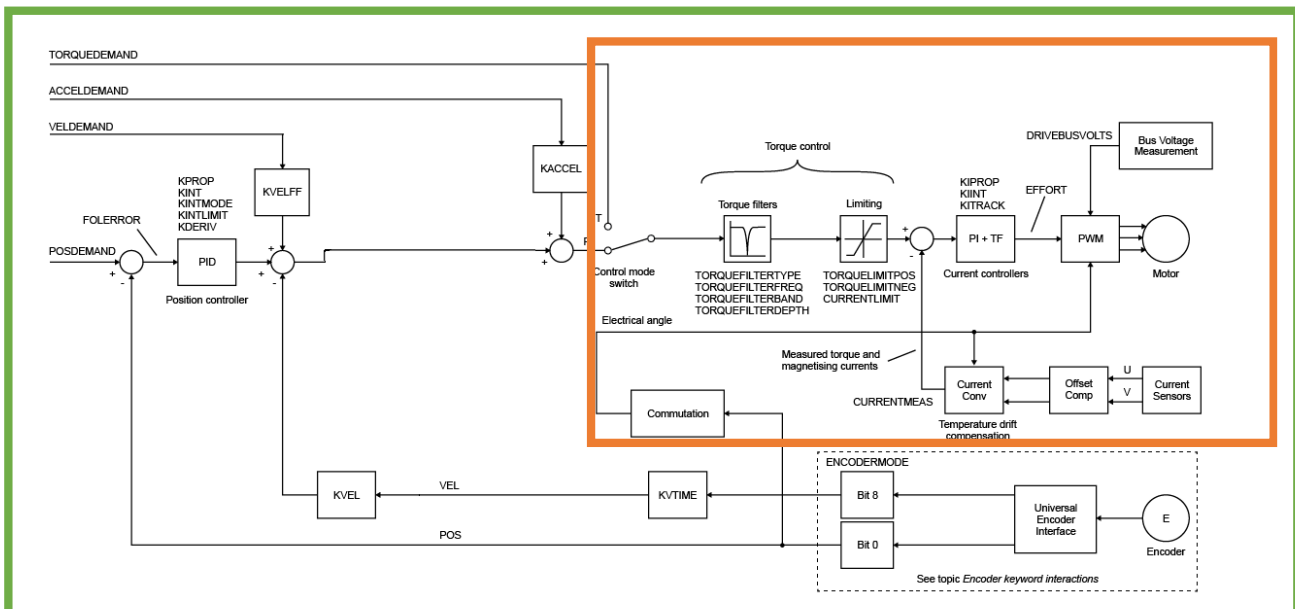
The current loop ultimately drives current into the motor which creates torque causing it to accelerate or decelerate. This loop must be tuned well to ensure good performance. If the current loop response is poor then no amount of velocity or position loop tuning will improve the situation.

The position and speed control loops are closed control loops which require feedback from the motor. A transducer fitted to the motor generates a signal which is used to determine the actual position and speed of the motor. Any difference between this and the demanded position and speed causes a demand on the current controller which in turn causes the motor to react to close the gap. Each of these control loops has some associated gain terms, each of which require tuning to the system load conditions.

**Torque servo control loops**

For applications requiring high dynamic performance, minimal settling time and improved resistance to oscillation at zero speed, operation in torque servo configuration may prove beneficial. Torque servo is configured by setting the CONFIG parameter to 6 (or `_cfTORQUE_SERVO`).

In torque servo configuration the cascaded velocity loop is removed and the position loop feeds a torque/current demand straight to the current control loop as shown below:





Torque servo is an invalid configuration for analog drives configured in velocity mode, if using the Mint VELREF motion command or if using a real-time Ethernet motion controller that needs to operate in CSV mode

### Control rates of the loops

The sampling rates for the three control loops (position, velocity and current) are fundamental to the operation of the drive, so are set to optimum default values automatically. However, the Mint Workbench 'Parameters' page or the Mint CONTROLRATE keyword can be used to read the values (as well as the profiler rate), as shown in the following table.

Item	Description	Default rate	Possible values
0	Profiler rate	1 kHz	125 Hz, 250 Hz, 500 Hz, 1 kHz, 2 kHz, 4 kHz
1	Position loop sampling rate	4 kHz	4 kHz
2	Velocity loop sampling rate	4 kHz	4 kHz
3	Current loop sampling rate	16 kHz	8 kHz or 16 kHz (Read only)

In addition to reading the sample rates the CONTROLRATE parameter/keyword can be used to adjust the profiler rate. This determines how often the drive generates a new target position/velocity/torque (depending on control mode). Other drive functions are linked to the profiler rate (e.g. processing of motion triggering) so in high performance Mint-based applications it may be necessary to increase the profiler rate – note that this increases the processor overhead so will slow down execution of the Mint application code.



Changes to CONTROLRATE take effect only after the drive has been power-cycled.

## 3 Tuning the motor

### 3.1 Checks to make before tuning

It is recommended that for the first stages of commissioning the motor is disconnected from the load. This is because some of the auto tuning tests should be performed off load and also in case of a wiring error or other such fault that may result in unexpected behaviour. If a brake is fitted to the motor either make sure 24Vdc is applied to the brake and check that the motor shaft is free to turn, alternatively motor brake control can be configured as previously described in section 2.2.2 [Motor brake control](#)



ABB motor brakes are not fitted with electrical suppression, so please ensure you fit a suitable diode inside the motor terminal box to suppress the brake coil, ensuring the correct polarity

Before the tuning process is started ensure the STO circuit (and other safety related parts of the control system) are operational so the drive can be disabled and put into a safe state if required. For the drive to enable as part of the tuning process the STO inputs and drive enable input (if defined) will need to be on.



If there is a load suspended vertically from the motor and the brake is disengaged while the drive is disabled then the load will fall without any inhibition.

### 3.2 Initial tuning

The following sections detail how to use the Mint Workbench 'Auto Tuning' page to tune the drive to suit a connected motor. The auto tuning tests can be split into those that can be carried out while the motor is not connected to the driven equipment (off load tests) and those which are carried out while the load is connected to the drive equipment (on load tests).

If you are using a third party or custom motor and there are gaps in the motor data you entered earlier (e.g. maybe you didn't know the motor inductance) it is necessary to complete all the auto tune tests that are listed below. Workbench will automatically select the tests that are required for you.

- Measure stator resistance and inductance
- Calculate current loop gains
- Test the feedback
- Measure the voltage constant (servo motors only)
- Measure the inertia

- Calculate the speed and position loop gains

If you are using a standard ABB motor selected from the database and the wiring is correct, it should only be necessary to complete the recommended auto tuning steps;

- Calculate current loop gains
- Test the feedback (Only required for certain feedback types)
- Measure the motor inertia
- Calculate the speed and position loop gains

We will provide more details on each of the auto tune tests in the following sections.

You should by now have already decided whether to use the default velocity servo configuration (CONFIG(0) = 1 or `_cfSERVO`) or the optional torque servo configuration (CONFIG(0) = 6 or `_cfTORQUE_SERVO`). The only auto tuning step that behaves differently based on this configuration is 'Calculate the speed and position loop gains' so should you decide to change CONFIG at any time be sure to either repeat this auto tune step (or recalculate these gains from the Position loop fine tuning screen).

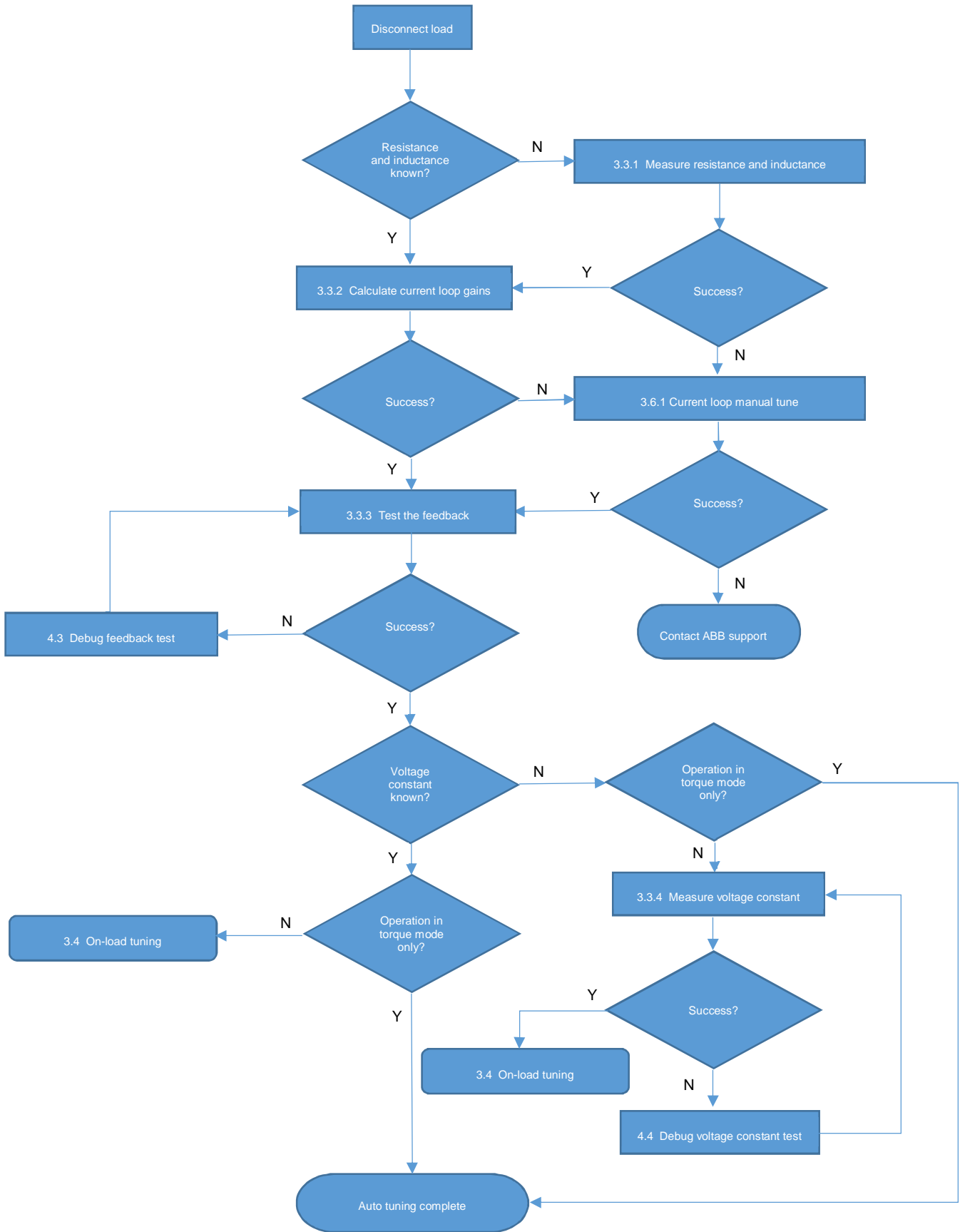


If the drive is operating in torque/current mode (as an analog drive or a network drive operating with a motion controller that is in CST mode) you do not need to complete any tests later in the list than 'Test the feedback'. All other tests relate to the speed and position control loops

To carry out an auto tune test select the appropriate check box or boxes and click the 'Start' button. To stop the process at any time click the 'Stop' button. As the tests complete a green tick adjacent to the test will be displayed. The test status will be displayed in the Mint Workbench output window.

### 3.3 Off load tuning

The following flow chart will guide the user through the complete off-load tuning procedure for an AC servo motor:





It is possible in certain circumstances that the system may not pass the Auto tuning procedure. If this is the case double-check the parameters you have used to set up the system and if necessary go to section 4 (Debugging and troubleshooting during tuning). If Auto tune continues to fail please contact your local ABB support office

### 3.3.1 Measure stator resistance and inductance

It is not necessary to do this test if the motor has been selected from the in-built database or if the motor’s stator resistance and inductance values have been entered from a datasheet. If this test is performed it will enable the drive and apply step changes in voltage to measure the response of the current in order to work out the time constant and therefore the resistance and inductance of the stator windings. This test may fail for motors with very low time constants. If you need to perform this test and it fails then you will need to manually tune the current loop parameters. If this test is successful then values for stator resistance and inductance will be displayed in the results window at the bottom of the screen. For example;

```
Test Started - Measure stator resistance and inductance.
Test Completed - Measure stator resistance and inductance.
Stator Resistance: 3.475734 ohms
Stator Inductance: 9.734374 mH
```

### 3.3.2 Calculate current loop gains

If this test is performed it will use the known values for resistance and inductance to calculate values for the current loop proportional (KIPROP) and integral (KIINT) gains. The motor will not move during this test. If this test is successful then values for these gains will be displayed in the black results window at the bottom of the screen. For example;

```
Test Started - Calculate current loop gains.
Test Completed - Calculate current loop gains.
KIProp: 0.172960
KIInt: 189.286362
```

The gain values will be calculated during the auto tune procedure using a defined bandwidth – the default is 2000 rad/s. But if needed this can be changed via the ‘Options...’ button to the right of the tick box. A lower bandwidth value may be needed if there is a very low inductance motor connected. If the drive cannot calculate these values you will need to manually tune the current loop.

### 3.3.3 Test the feedback

If this test is performed it will enable the drive to rotate the motor backwards and forwards an electrical cycle. This test allows the drive to measure the offset between the mechanical position of the magnets in the rotor and the zero point of the feedback device connected. In theory this test is not needed when using ABB servo motors with either resolver or incremental encoder with halls, as the offsets for these motors are retrieved from the database, but it is usually a good idea to do it as it will help to identify wiring faults early on in the test procedure. If this test is successful then the Mint Workbench will display information about the measured feedback offset:

SmartAbs encoder example...

```
Test Started - Test the feedback.
Test Completed - Test the feedback.
Feedback Offset: -178.591507 degs
```

Incremental encoder + halls example...

```
Test Started - Test the feedback.
Test Completed - Test the feedback.
Hall Table:
Hall State      1      2      3      4      5      6
Sextant        6      2      1      4      5      3
Hall Forward Transition Angles (deg):
Sextant 1-2  2-3  3-4  4-5  5-6  6-1
Angle      29.9  89.3  150.8 -149.4 -88.1 -30.7
Hall Reverse Transition Angles (deg):
Sextant 2-1  3-2  4-3  5-4  6-5  1-6
Angle     28.6  88.6  149.3 -150.5 -90.0 -31.1
```



The current loop gains must have been calculated successfully before the feedback can be tested.

### 3.3.4 Measure the voltage constant

If this test is performed it will enable the drive and rotate the motor a small amount (with direction, travel, speed and torque limits set by the Options button to the right of the test). The drive measures the back EMF produced by the motor and if the test is successful Workbench displays the result. This test is not required if the voltage constant has been entered earlier (e.g. if you are using an ABB servo motor that was in the motor database).



This test can be a useful way to diagnose a de-magnetised motor. If the voltage constant is measured and reads much lower than the catalog value then there is a good chance the motor has been de-magnetised and should be replaced

## 3.4 On load tuning

If the application requires that the drive operates in torque control mode only (as an analog drive or in combination with a motion controller operating in CST mode) you can skip straight to 3.5.1 Fine tuning the current loop. The response of the current loop is not affected by the mechanical load so there is no need to measure the load inertia or set speed and position loop gains in this case.

If the drive needs to use the speed and/or position control loops you should continue to read on...

So far we have completed the auto tuning tests that should be performed off-load. The remaining tests should eventually be done on-load (although it is common to also run these off-load initially to be sure there are no fundamental issues and to ensure that the motor's forward direction matches the application's forward direction before taking time to reconnect the mechanical load, particularly if the mechanics will only move in one direction). Use the Mint keyword MOTORDIRECTION to set the motor direction as required.

For the rest of this section we will assume these final tests have been done off-load, the motor direction has been set as required and the load has been reconnected. Before the drive can calculate speed and position loop gains automatically it must know the inertia of the connected load, so we must select both tests to successfully complete the on-load tests (if at a later time you need to recalculate the speed and position loop gains, for example, if you later modify the application maximum speed, there is no need to measure the inertia again if this test has already completed successfully).

Before starting the tests click the Options button to configure the range and performance limits of the tests to be done (for example, you may want to limit the direction to forwards only, or limit the maximum travel of the motor to avoid hitting an end stop).

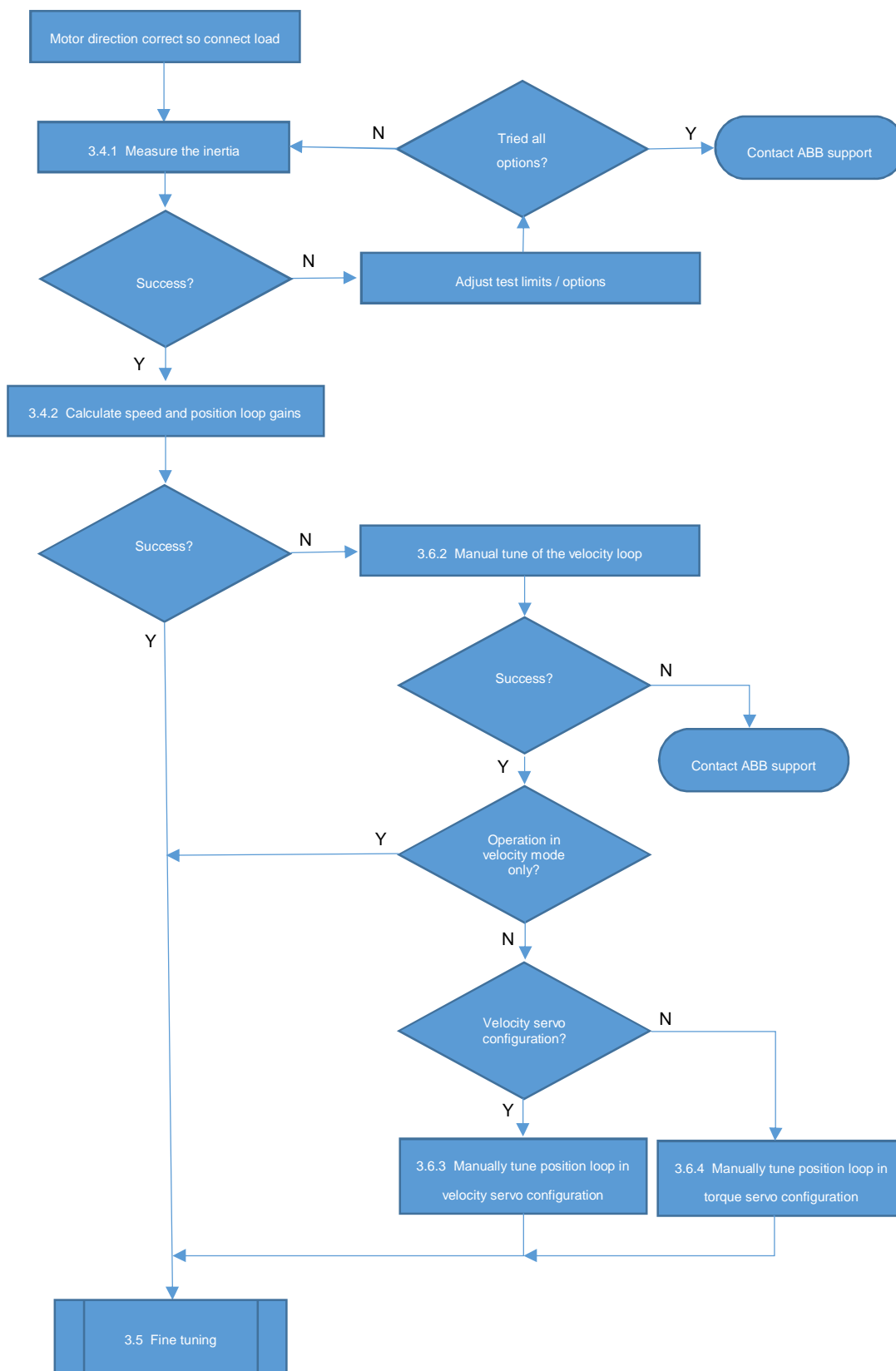
The Options button for the speed and position loop gain calculations sets a bandwidth (stiffness) for the system. The default is 150 rad/s. This will usually give a high level of performance. You may find for systems with high inertia mismatch, mechanical backlash or compliance, or low encoder resolutions that you need to reduce this bandwidth. Similarly for very rigid systems or motors with high resolution encoders fitted you may find you are able to increase the bandwidth (300 is a recommended maximum value).

When you have selected any options you require, click 'Start' to begin the auto tune process. To stop the auto tune process at any time, click 'Stop'. The output window at the bottom of the screen shows the progress of the auto tune tests and a green tick will be displayed when each test is completed.

Wait for the tuning tests to finish. This will be indicated by the 'Start' button becoming active again, and either the 'Auto tuning complete' or 'Auto tuning failed' message appearing above the output window.



The following flow chart will guide the user through the on load tuning procedure.



The auto tune process does not calculate every single gain, only those that are essential to the operation of a particular control loop. It may therefore be necessary to manually enter values for tracking factor gains, position loop integral gain, acceleration feedforward gain etc. depending on the application requirements



### 3.4.1 Measure the inertia

If this test is performed it will enable the drive and will apply a known torque and attempt to measure the resulting acceleration. Using the formula  $T = J\alpha$  (Torque = Inertia \* angular acceleration in rad/s<sup>2</sup>) the drive can measure the total inertia (connected load plus the motor itself). Use the Options button to adjust the test parameters. The default ranges/limits are usually acceptable but for large inertia loads/motors it may be useful to reduce the test torque, increase the test distance and set only one direction of travel. The drive calculates the load damping by measuring how long it takes to come to rest after removing the applied torque. If this test is successful then values will be displayed in the black results window at the bottom of the screen. For example;

```
Test Started - Measure the motor inertia.
Test Completed - Measure the motor inertia.
Load Inertia: 0.0000841 kgm²
Load Damping: 0.0003028 Nms/rad
```

### 3.4.2 Calculate the speed and position loop gains

If this test is performed it will use the data from all previous drive configuration parameters and tests (load inertia, damping factor, encoder resolution, encoder pre-scale, control rate and application max speed) to determine the settings for the gain terms that affect the speed and position loops. If this test is successful then gain values will be displayed in the black results window at the bottom of the screen.

For example in 'velocity servo' configuration;

```
Test Started - Calculate the speed and position gains.
Test Completed - Calculate the speed and position gains.
KVProp: 7.569449
KVInt: 574.791931
KProp: 0.103515
KInt: 0.000000
KDeriv: 0.000000
KVelFF: 6.001200
```

For example in 'torque servo' configuration;

```
Test Started - Calculate the speed and position gains.
Test Completed - Calculate the speed and position gains.
KVProp: 0.000000
KVInt: 0.000000
KProp: 2.172570
KInt: 0.032819
KDeriv: 57.569756
KVelFF: 0.000000
```

The Options button can be used to set the bandwidth (stiffness) used to calculate these values. By default the speed and position loop bandwidth is set to 150 rad/s. This may need to be decreased for systems with low resolution or high mechanical compliance for example, but it may be possible to increase this bandwidth for systems with high resolution and/or very rigid mechanical systems.

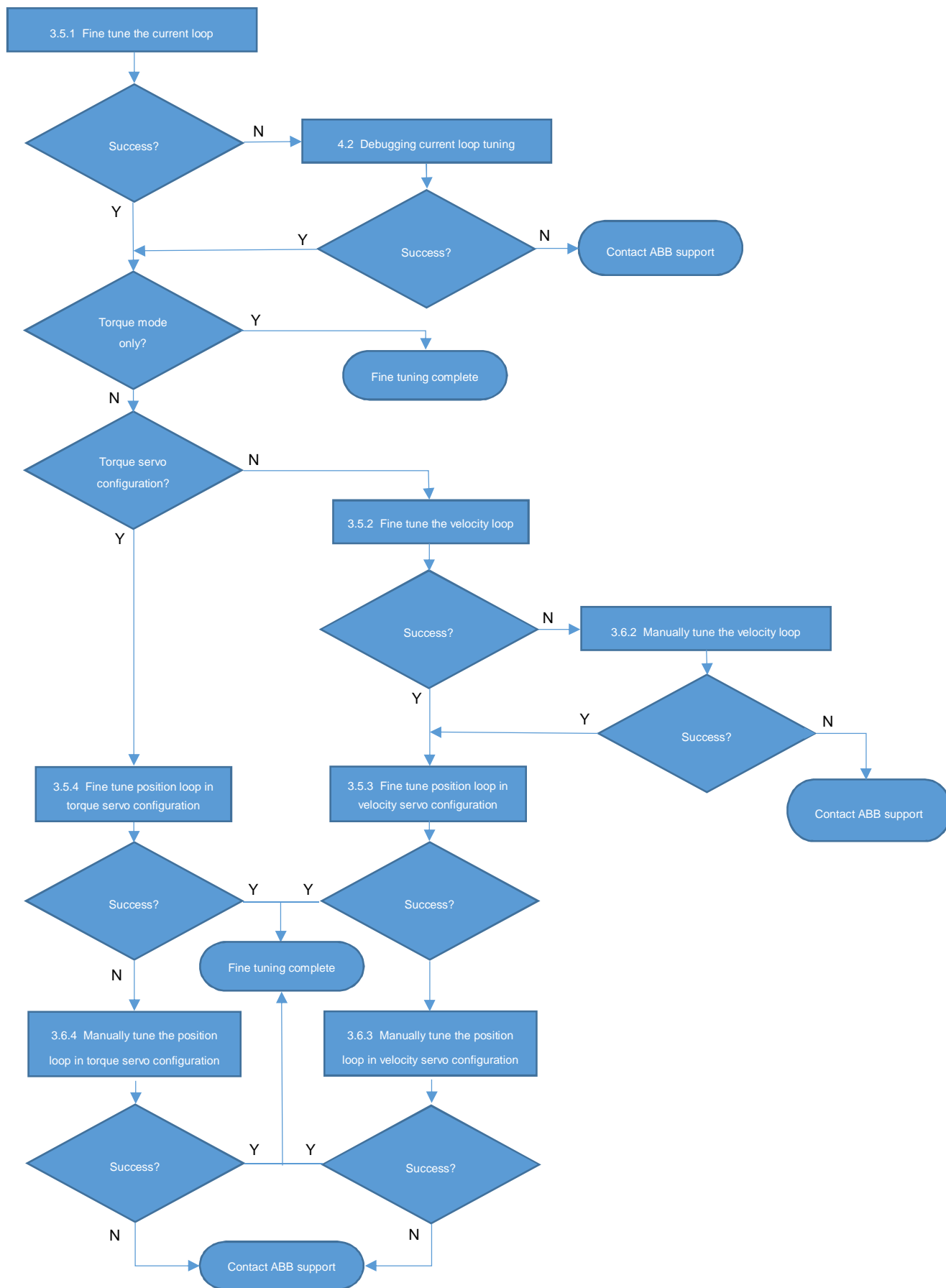


Calculating speed and position loop gains will overwrite any speed or position loop gains that have been entered manually (e.g. KACCEL) so only use this test if you are prepared to lose any fine tuning values you may have entered at a later stage in the process

## 3.5 Fine tuning

The following section presumes that the auto tune has been completed without errors. Using the fine tuning spy windows we will adjust the values that the auto tune calculated as a starting point. We always start with the innermost control loop (current) and work outwards. We only need to fine tune the loops that will be used (e.g. if the drive is operating in torque control mode there is no need to fine tune the velocity or position control loops).

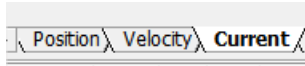
The flowchart below will guide the user through the fine tuning process:



We can access the Mint Workbench 'Fine Tuning' page from the Toolbox on the left side of the screen;



Once on the Fine tuning page select the relevant loop to fine tune from the set of tabs in the bottom right part of the screen;



When tuning the control loops it is wise to make use of the 5 different graphs you can use to record data to compare your results as you make adjustments to the control loop gains



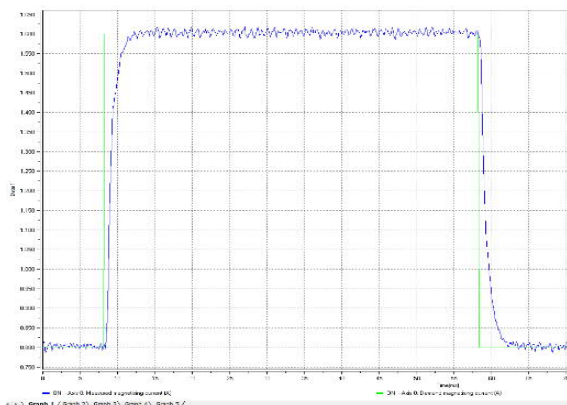
### 3.5.1 Fine tuning the current loop

Select the Current tab at the bottom right corner of the Mint Workbench screen. The drive's current loop will be tuned quite well during the auto tune but it is always worth checking it to see if we can make improvements. This is particularly important where a motor with a low inductance may be used (e.g. large frame size rotary motor or a linear servo motor) where the default current loop bandwidth may be too high, or in applications requiring very precise control where it might be necessary to ensure an optimal current loop response. If there are any errors during this fine tuning process refer to section 4.2 Debugging current loop tuning.

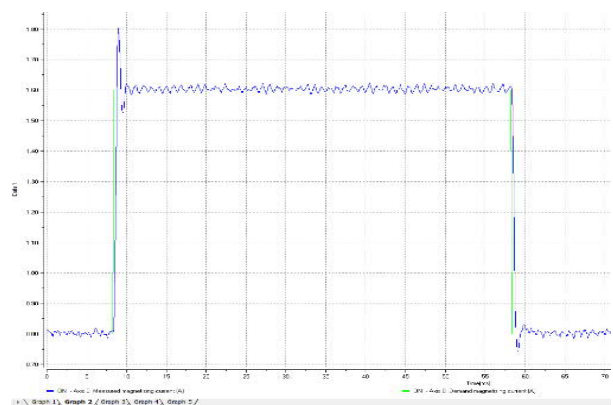
1. The tests can be started with the default parameters (with the current at 25% for 50ms and the test type set to 'Stationary'). Press Go to start a test "move". The motor may jerk slightly and the current loop test data will be uploaded to Workbench. You should see that the measured current tracks the demand current quite well, maybe with some curve at the end of each step change. Ignore this for now, at low current levels the actual performance can be masked slightly by the noise from the analog current sensors so always adjust the gains with 100% current
2. If the initial 25% test seems OK, set the test current to 100% and repeat the test (Note that MotiFlex e180 drives set a default current loop filter, KITIME, of 0.1ms to compensate for noise in the current sensors. For very high performance applications remove this filter – i.e. set KITIME to 0)
3. If there are large overshoots in the current response immediately then the bandwidth used during the auto tune could have been too high for the motor's (low) inductance. Either reduce KIINT and KIPROP (try 75% of their original values) or you can click on the 'Calculate' button in the spy window and reduce the bandwidth – the drive will recalculate the current loop gains automatically.

If the response looks generally OK but with curves on the edges move onto the next step

4. We can now start to adjust KITRACK (which the auto tune doesn't set). We are looking for the measured current (blue trace by default) to be as close to the demand current (green trace by default) as possible. As we can see below there will initially be some 'lag' in the response of the current loop which we can improve by increasing the KITRACK value. The units of KITRACK are percent. Increasing KITRACK will increase the response rate and decrease current tracking error. Too much KITRACK will result in overshoot/spikes or indication of over tuning, all of which are undesirable



KITRACK = 0, overdamped response



KITRACK = 80, excessive overshoot

Adjust KTRACK to find a happy medium between the initial overdamped response and an ideal square response, but avoiding large current overshoots.

- 5. Once satisfied with the response select Tools>Store Drive Parameters to save any changes permanently (if you don't do this then resetting the drive will result in the drive restoring the previous tuning parameters)

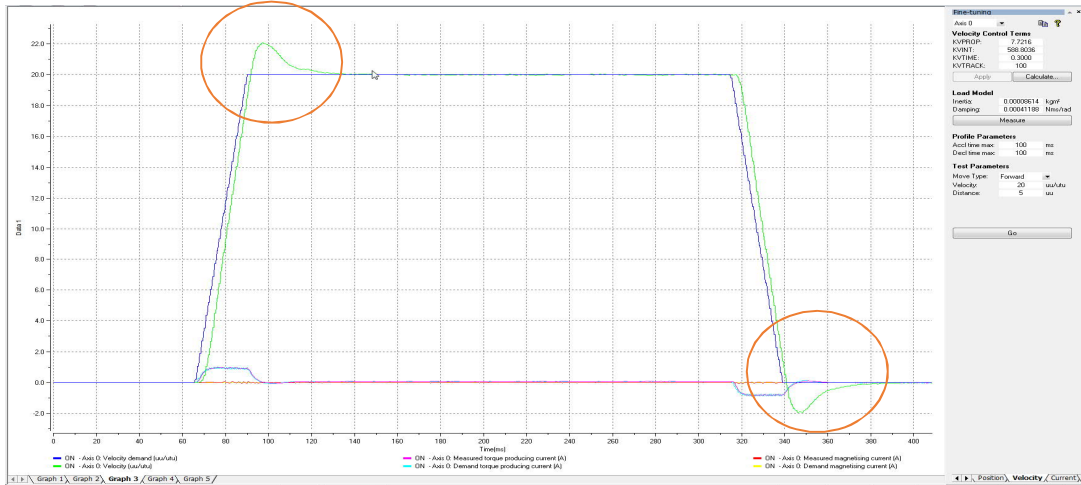
If the application shows signs of current loop instability when using a large motor with a relatively high electrical time constant (e.g. over 10ms) then it may be necessary to set a relatively damped current loop response by reducing the current loop bandwidth).

### 3.5.2 Fine tuning the velocity loop in velocity servo configuration

First select the velocity tab at the bottom right corner of the Mint Workbench screen. We will now perform a velocity test move to determine how closely the actual velocity follows the demand velocity.

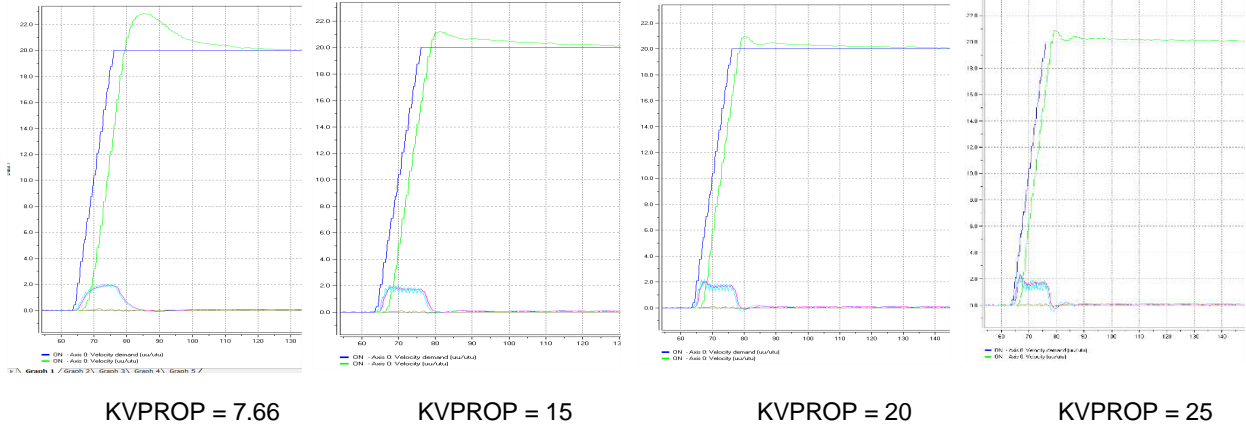
It is not necessary to fine tune the velocity loop if running in torque servo mode (CONFIG = 6 or \_cfTORQUE\_SERVO)

1. Try to use acceleration, deceleration and speed values that will be "typical" for your application. Units will be dependent on scaling (due to our scaling selections earlier we are using revs for distance and revs/s for speed). The test move type can be Forward, Reverse or Bi-directional. Forward and reverse are useful where the mechanics should only ever move in one direction. Bi-directional is useful when the axis has physical travel limits – you can ensure each test starts from the same place. In this example our test move is setup to do a forward move at a velocity of 20 rev/s, travelling for 5 revs with acceleration and deceleration time max set to 100ms (this is the time to the drive/application max speed, not the time to the defined test move speed)
2. Click the 'Go' button when these parameters have been typed in and you are ready to start the move. When the move is complete Workbench will display a graph. The graph below shows the response to the forward move we have programmed. The velocity demand (blue) and actual velocity (green) should closely match. As we can see there is an overshoot at the end of the acceleration ramp and at the end of the deceleration ramp. The steps that follow will allow us to improve this response somewhat. The aim here is to improve the response as best we can without 'over-tuning' the system which will result in oscillation, particularly at zero speed.

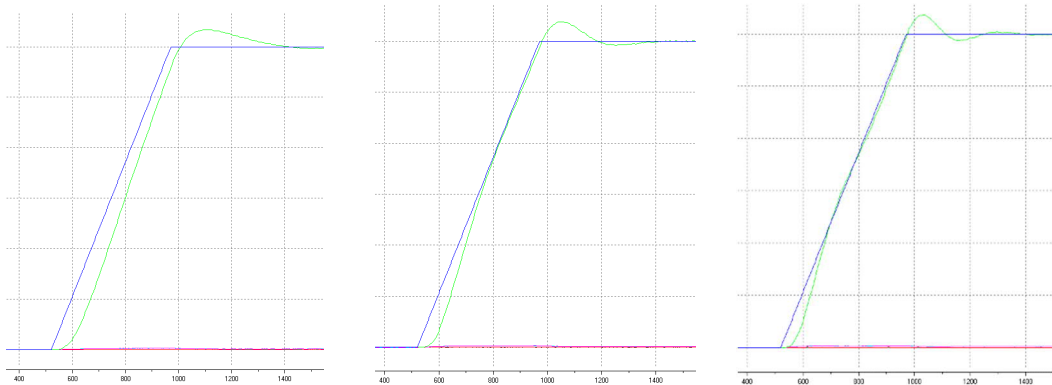


3. There are three gains that can be fine tuned; KVPROP which dampens overshoots, KVINT which sharpens the response and KTRACK which implements a general tracking factor. Auto tune will set KTRACK to 100(%) and it should typically be left like this, only adjust KVPROP and KVINT (as with the current loop, if you want to try just adjusting the bandwidth to generally stiffen or soften the system, the Calculate button allows you to adjust this and the drive will recalculate the gains – but be aware that this bandwidth is shared with the position loop so adjusting this here will also adjust the position loop gains). There is also a filter for the measured velocity, KVTIME. Typically this should be set to 0 for digital encoders and can be set to anything up to 2ms for analog (resolver) based feedback devices. When changing these tuning values it is always wise to make small incremental changes to the values you are changing in between tests as at some point you may go too far and create an unstable system
4. The auto tune procedure in our example system has given us a value of KVPROP = 7.6662. We can see from our initial response that we have some overshoot in the speed response that we'd like to dampen. To dampen overshoots we increase KVPROP. We can change this in small steps and press 'Go' after each edit to apply to run the same move as earlier. After each move we evaluate the response for actual velocity versus target velocity. We keep doing this until the actual value starts to match as closely as possible, all the time being careful to watch that the steady state speed

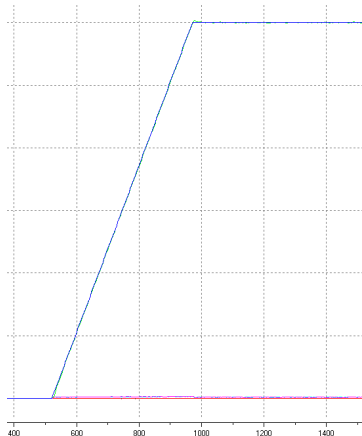
(especially zero speed) is not unstable (we might hear the motor buzzing for example). Back off KVPROP if you detect any instability. Typically only very small changes (5-10%) in KVPROP are needed, in our example we have artificially ensured KVPROP is low initially to allow us to demonstrate the effect of changing this value



- The auto tune procedure has given us a value of KVINT = 558.1609 we can change this value incrementally and press 'Go' after each edit to apply to run the same move as earlier. After each move we evaluate the response for actual velocity versus target velocity. Increasing KVINT will sharpen the velocity response (i.e. reduce the lag between demand and measured velocity....note that the effect of this is two-fold....the demand current will increase and the amount of overshoot will increase too (which could mean we need to return to adjusting KVPROP once we are happy with KVINT). To illustrate this the screenshots below show a sequence of test moves with KVINT being increased gradually....



You can see that as we increase KVINT the measured velocity (green) responds quicker, but at the expense of more overshoot as we reach constant velocity...we would therefore add more KVPROP to dampen this overshoot. The graph below shows our final result after adjusting KVINT and KVPROP in turn...



Our result looks perfect because we used an unloaded motor for our tests, so don't worry too much if there is a small overshoot in velocity at the end of the acceleration and deceleration sections of the test move – this is normal.

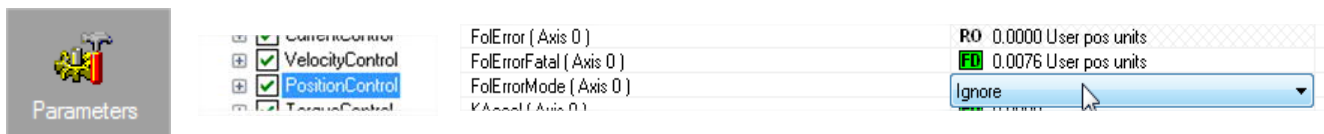
- Once complete you can save the parameters, the drive is now ready to have its position loop tuned.

### 3.5.3 Fine tuning the position loop in velocity servo configuration



This section assumes the motor is loaded within its limits and is free to rotate and requires that the drive is set in “velocity servo” mode i.e. CONFIG(0) = 1 or \_cfSERVO (this is the default configuration).

Select the ‘Position’ tab in the spy window of the Fine tuning section. Whilst carrying out position loop tuning on the drive, depending on the level you set for the fatal following error earlier in the commissioning process (FOLERRORFATAL) it is possible that you will get an error: ‘Fatal Following error exceeded’. Whilst tuning this can be a nuisance and so sometimes it may be desirable to turn the error detection off completely (FOLERRORMODE = 0 can be entered at the command window or you can navigate to Parameters>Family>Position Control>FolErrorMode and set this to Ignore).



Remember to turn FolErrorMode back to ‘Crash stop disable’ (or whatever mode is required) again after tuning is complete!

When tuning the position controller we will be adjusting some of the gain terms that effect its operation. We will describe the three most commonly adjusted parameters in the following paragraphs...

KPROP is the proportional gain of the PIDVF control in the position loop and its value is multiplied by the following error (the difference between demand position and measured position) to determine the contribution it makes to the output of the position controller. KPROP is normally used to generally reduce the following error and is the first gain that should be adjusted when fine tuning the position loop. It is recommended to set a low initial value, then increase it gradually until the following error becomes oscillatory. KPROP should be set to a value a little way before the following error begins oscillation. This scenario is called “critically damped”. Do not use KPROP to try and remove very small errors in final position, you will need a very high KPROP to do this and the result will be an unstable axis – refer to KINT below for the correct course of action.

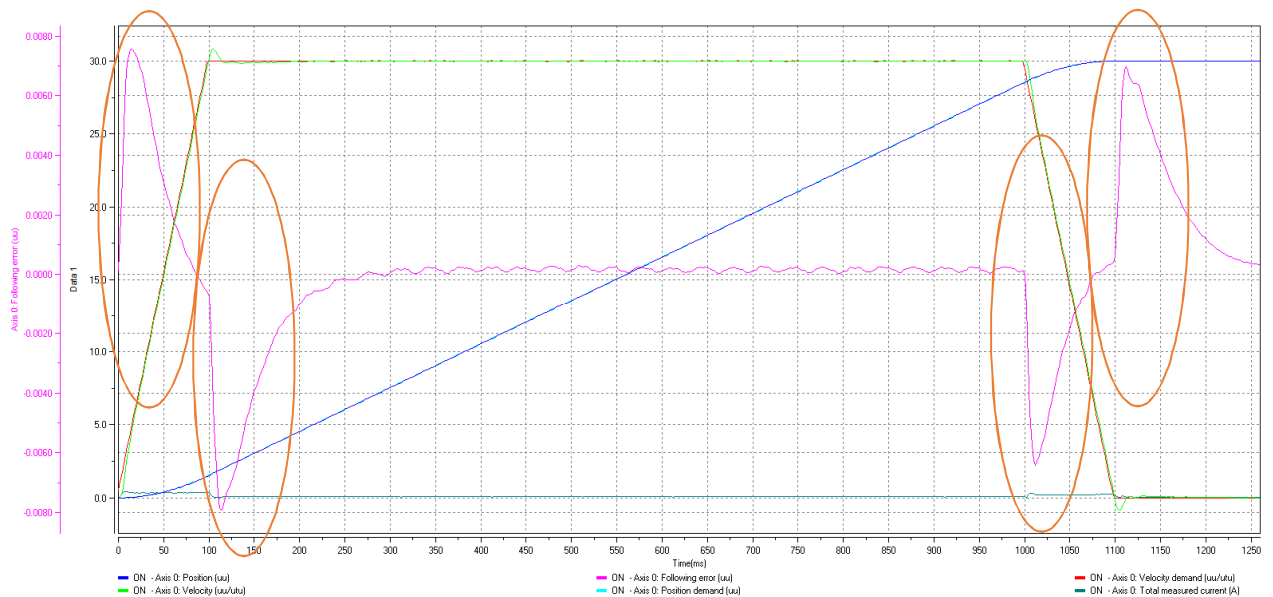
KACCEL is the position control loop acceleration feed forward gain term. KACCEL feeds a torque demand straight to the current controller proportional to acceleration and is useful for reducing velocity overshoots on high acceleration/deceleration moves. The auto tune does not calculate KACCEL so it will always be necessary to manually fine tune this for the optimum response. Depending on parameters such as application max speed and encoder resolution this gain can take a value anywhere from 1 up to its maximum value of 10000.

KINT is another important term that we may need to adjust. KINT is the integral gain for the position loop. The auto tune does not set a value for KINT when using velocity servo configuration. KINT can be used to overcome steady state errors that could arise from situations such as gravity acting on a vertical load or high friction preventing the axis from achieving a very accurate final position. KINT is used in combination with KINTLIMIT and KINTMODE. KINTLIMIT sets a limit on how much the integral term can contribute to the overall demand. Although the default is 100% this is not an advisable setting as the integral term can easily produce large oscillatory motion if it is set too high. It is suggested a value of 10-15 is used for KINTLIMIT initially. KINTMODE sets when the integral term is applied, it is best to set this to either ‘smart’ or ‘always’ if it is felt KINT is needed. KINT is usually a very small fractional value, so if you plan to use it start with a value of 0.0001 and work up in 0.0001 steps initially.

The sequence below describes how we fine-tuned the position loop on our example system:

- Set FOLERRORMODE to 0 (\_emIGNORE) at the command window initially, until you are happy that the response is good enough not to intermittently trip the following error detection. Once the following errors are within the defined range for FOLERRORFATAL be sure to set FOLERRORMODE back to 1 (\_emCRASH\_STOP\_DISABLE)

- When entering test move parameters on the Position fine tuning tab, we try to use acceleration, deceleration and speed values that will be “typical” for the application. In our example we set the test to do a forward move at a velocity of 30 rev/s for 30 revs with an acceleration and deceleration time of 100ms. This gives a total move time of 1.11seconds. When we perform the test (by clicking the ‘Go’ button) the drive will move the axis and will capture some key parameters automatically (demand and measured position, following error and total measured current for example). The auto tune process should have given us a relatively good level of performance, so demand and measured position are likely to look like they overlap with the graph fully zoomed out. We need to analyse the following error and total measured current in detail, so right click each of these legends and assign an independent Y axis to each of these so we can see them on their own scale. We examine the current to ensure the axis is not hitting a current limit (which we could spot if the current is ‘flat-lining’ during the movement). If a current limit is reached check the current limit is set correctly, try decreasing the acceleration and/or deceleration rates or investigate whether the motor/drive is undersized for the application. If the current shows no sign of limiting we can examine the following error in more detail
- As we can see from the graph below, the following error is at its greatest during changes in acceleration and deceleration (approx. 0.008 revs).

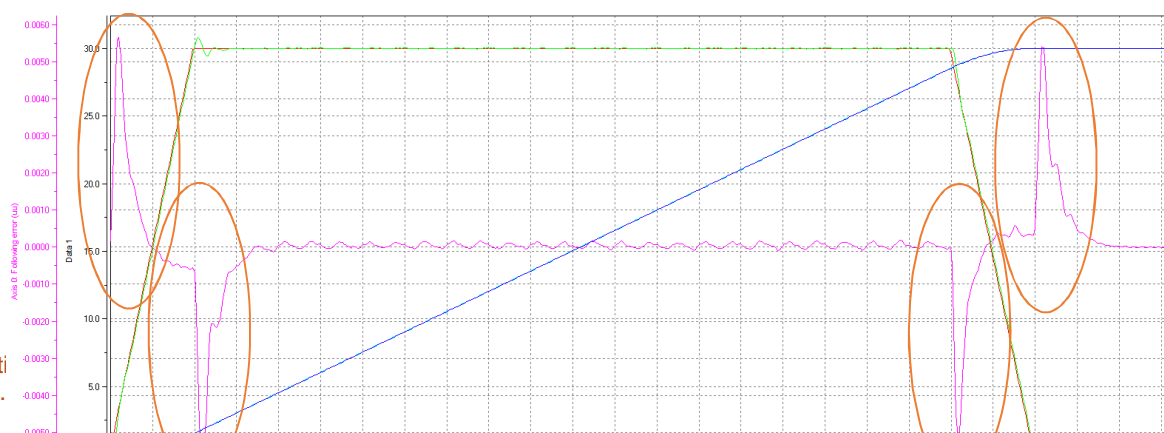


- Gradually increase the KPROP value, repeating the test move each time, using it to decrease the overall error and improve the settling time of the axis at the end of the move – it’s typical that KPROP can’t be increased too much without creating instability. Once the following error trace starts to show signs of oscillation back off KPROP slightly and leave this value set. It is sometimes useful to enable the axis before the test move so that it remains enabled at the end of the move....this way you might spot oscillation more easily

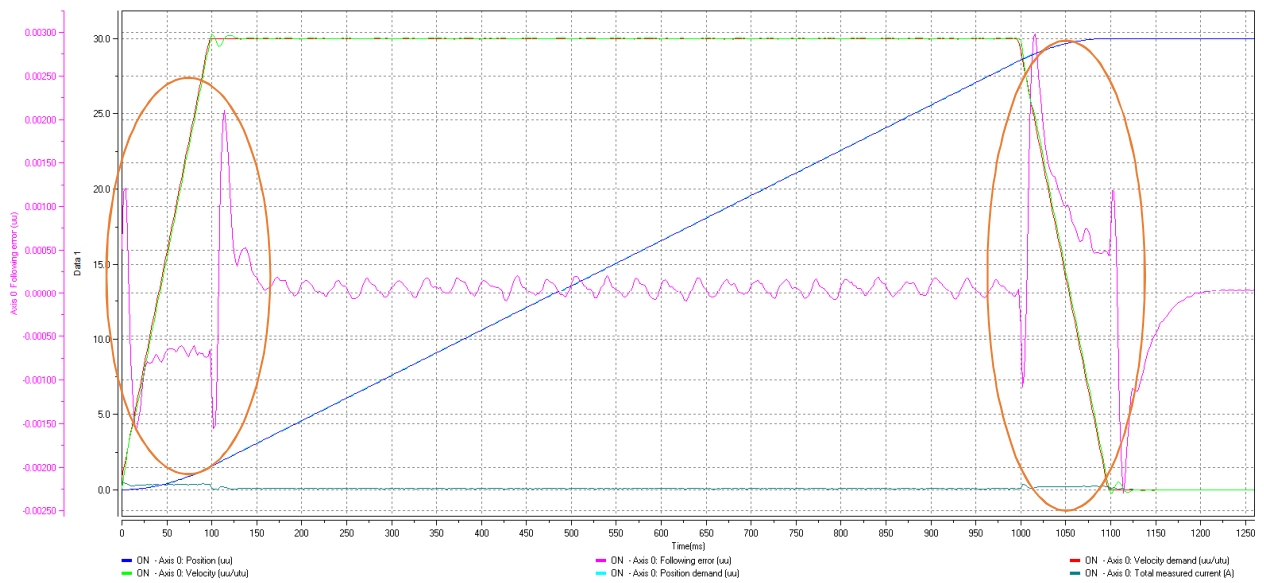


If the motor vibrates excessively or if the following error response looks excessively unstable before you even adjust KPROP the auto tune has calculated too high values – try reducing KPROP manually. It is **not** recommended to use the Calculate button to recalculate new values as this will also adjust the velocity loop gains we have just fine tuned

- Once the adjustment of KPROP is complete you should have a lower following error without excessive vibration from the motor. You will still have larger errors during the changes in acceleration and deceleration, but the duration of these peaks should be reduced. We will deal with these when we start to adjust KACCEL next...



- 6. In a lot of applications the following error during the move is not critical, it may only be important that the final position is accurate. If there is the need to specifically adjust the gains during acceleration and deceleration to minimise following error at all times then we can use KACCEL. As above, we can make incremental increases to the value in KACCEL until we see a desired response from the generated graphs after each test move. In our example we made initial changes of 0.1, increasing these steps to 1, 2 and 5 as we found no noticeable change with our steps at the start. We eventually reached a value of 185 for KACCEL. In the graph below you can see that the following error during acceleration and deceleration has now reduced (approximately half of the original error)...



- 7. At the end of the move, particularly if there is a lot of friction in the system, we may find that the axis either takes some time (maybe 100's of ms) to pull into position, or in some cases remains slightly out of position. This is where we can use KINT (and KINTLIMIT together with KINTMODE) to improve the time taken to reach the final position and/or increase the accuracy of the final positioning. The graph below illustrates a typical response at the end of a move with no KINT added to the control loop...



In this test move the profiler completes the velocity demand at 1100ms. There is a little overshoot during deceleration and our following error (the pink trace) maximises at approximately 1130ms into the move and then starts to be pulled back towards zero. At this point there is no demand velocity, so KVELFF which uses the velocity demand to generate an input to the control response has no effect. Similarly there is no demand acceleration, so KACCEL will also make no contribution to the response. The only gain acting at this point is KPROP. We can see that even after 100ms the following error has still not been pulled back to zero (and so our axis is not at the final target position). Depending on the magnitude of the following error, and the application requirements, this may not be an issue. If however we require very precise positioning with minimal delay achieving this we may need to add KINT to the control loop.





The graph below now shows the difference in response after adding KINT gain (in this case we used a KINTMODE of 'Steady State' and a KINTLIMIT of 15%)...



As before, the profiler completes the velocity demand at 1100ms. The overshoot during deceleration occurs as it did previously and our following error (the pink trace) maximises at approximately 1130ms into the move. Now we have added KINT to the control loop we see that the following error is reduced to zero by 1250ms (previously it was still being reduced). In this example we actually had zero friction, so our KINT actually caused the following error to “overshoot” in a positive direction...with a real load this is less likely to happen and the final position would be achieved in even less time.

Below we describe the remaining position loop gains that are less commonly adjusted in velocity servo configuration:

KDERIV sets the position control loop derivative gain. This term acts on the rate of change of error so can be used to reduce overshoots (although these overshoots can be pre-empted by KACCEL which will typically prevent overshoots in the first place). If it is used it is recommended to adjust this gain after KPROP and KINT have been fixed.

KVEL is the velocity (feedback) gain of the position control loop. This value, multiplied by the measured velocity, is subtracted from the position PID output. It therefore acts as a general damping term at all times during the move.

KVELFF Sets the velocity feedforward gain for the position controller. The auto tune procedure calculates a value of KVELFF based on application max speed, encoder resolution and control rate. This value should **not** normally be changed. You may only want to change it to artificially introduce a lag or lead to the response for some reason (e.g. reducing KVELFF may remove an overshoot during acceleration at the expense of taking longer to reach the steady-state speed).

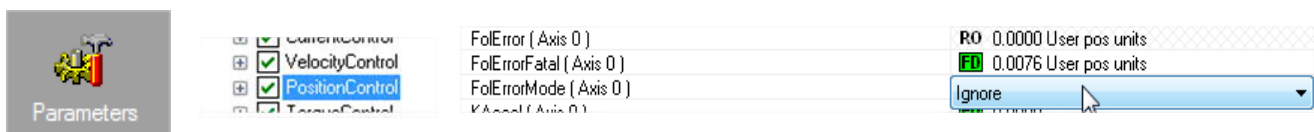
Please refer to the Mint Workbench help files and drive manuals for further information.

### 3.5.4 Fine tuning the position loop in torque servo configuration



This section assumes the motor is loaded within its limits and is free to rotate and requires that the drive is set in “torque servo” configuration - CONFIG(0) = 6 (or \_cfTORQUE\_SERVO)

Select the ‘Position’ tab in the spy window of Workbench. Whilst carrying out position loop tuning on the drive, depending on the level you set for the fatal following error earlier in the commissioning process (FOLERRORFATAL), it is possible that you will get an error: Fatal Following error exceeded. Whilst tuning this can be a nuisance and sometimes it may be desirable to either turn the error detection off completely during tuning (FOLERRORMODE = 0 can be entered at the command window or you can navigate to Parameters>Family>Position Control>FolErrorMode and set this to Ignore).



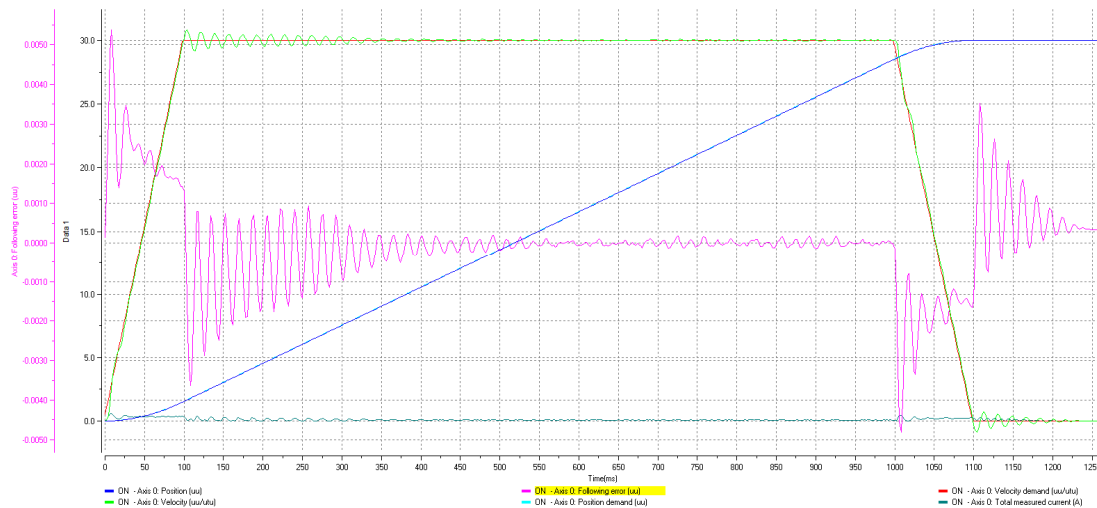
Remember to turn FolErrorMode back to ‘Crash stop disable’ (or whatever mode is required again after tuning is complete or at some point in the tuning process when you think the response is within your required FOLERRORFATAL limits!

The auto tune process for a drive in torque servo configuration will have automatically set starting values for the following gains:

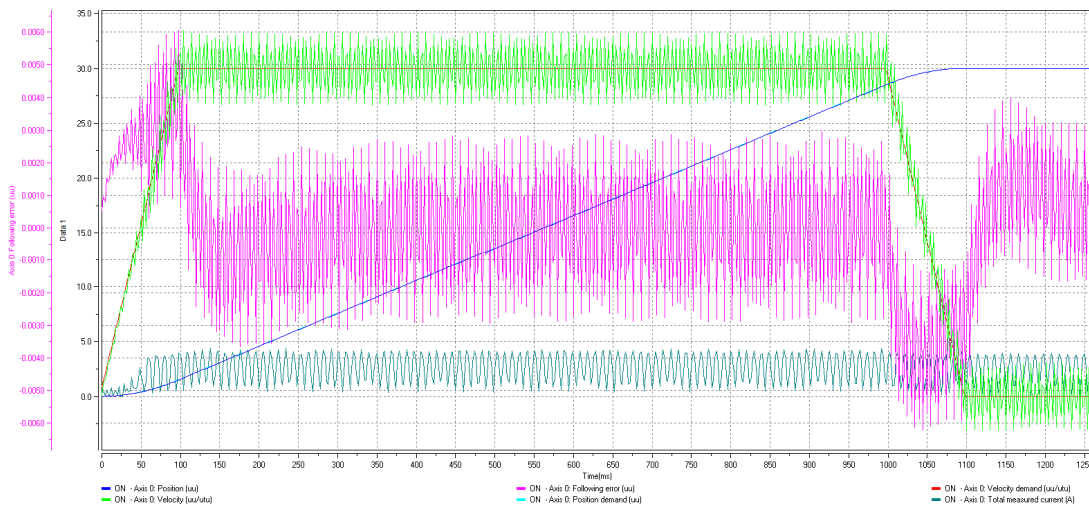
- KPROP
- KINT
- KDERIV

The two main terms that affect the response in general are KPROP and KDERIV. The graphs below show for example what the effect of KDERIV has on a torque servo configuration...

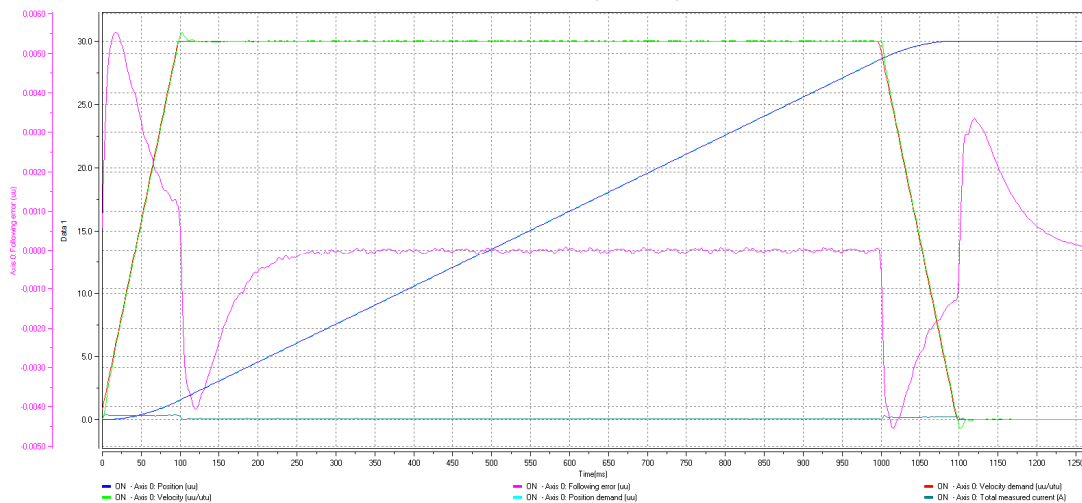
KDERIV too low, excessive (low frequency) oscillation results:



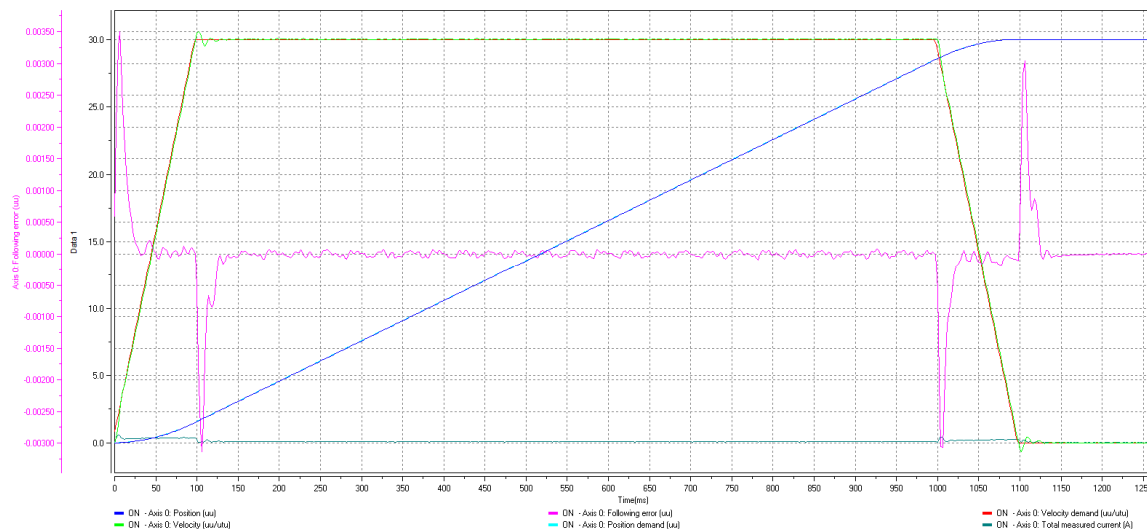
KDERIV too high, excessive (high frequency) oscillation results:



KDERIV set well, no oscillation and axis settles relatively quickly:

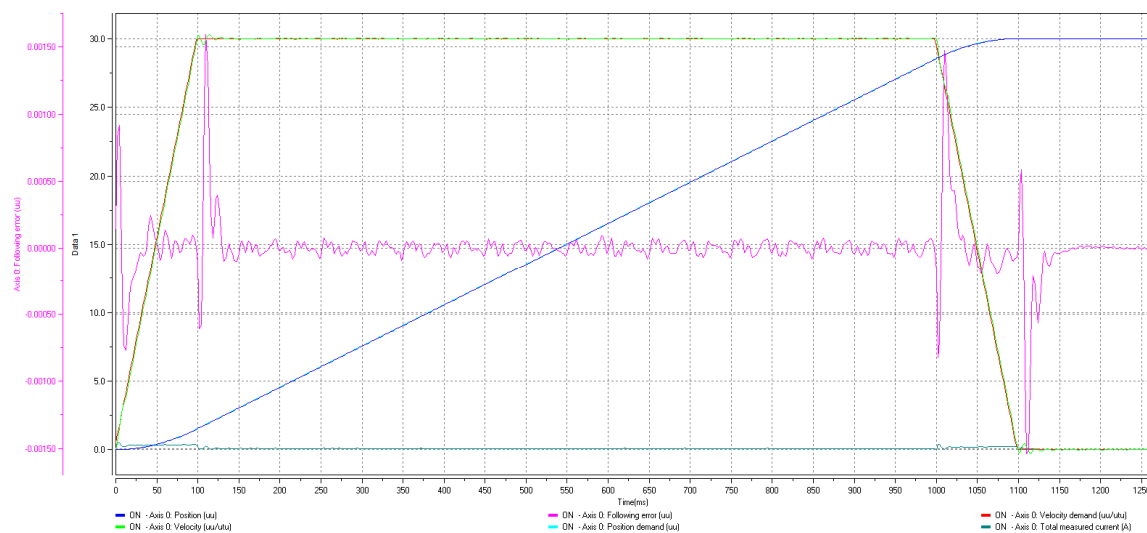


KPROP can be used to reduce the magnitude of the following error to some extent so that, together with increases in KDERIV, the best possible response using only these two gain terms can be achieved. You can see from the graph above that at steady state (i.e. constant slew or zero speed) the following error takes some time to reduce to zero. If you have read the previous section (Fine tuning the position loop in velocity servo configuration) you may recognise already that to improve this response and reduce the time taken to achieve zero following error requires the use of the integral gain KINT. By increasing KINT (again using very small fractional values) and adjusting KPROP and KDERIV together we can eventually achieve a response as shown below...



As the velocity loop is not used at all with this configuration the 'Calculate' button on the Position fine tuning tab (and its corresponding Bandwidth adjustment) is particularly useful in torque servo configuration. This can be used to automatically increase/decrease KPROP, KINT and KDERIV to stiffen or soften the system as required without having to worry about unwanted changes to the velocity loop terms.

Again, just as with the velocity servo configuration, we can attempt to remove the following error peaks during acceleration and deceleration by introducing a value for KACCEL. In the graph below we have managed to more than halve the following errors during these phases by adding KACCEL to the system...



Below we describe the remaining position loop gains that would not typically be used in torque servo configuration:

KVEL is the velocity (feedback) gain of the position control loop. This value, multiplied by the measured velocity, is subtracted from the position PID output. It therefore acts as a general damping term at all times during the move.

KVELFF sets the velocity feedforward gain for the position controller. This value should not normally be changed. You may only want to change it to reduce any lag in the response (e.g. adding a small amount of KVELFF may remove lag during acceleration). However it would be more typical to use KACCEL to reduce lag.

Please refer to the Mint Workbench help files and drive manuals for more information.

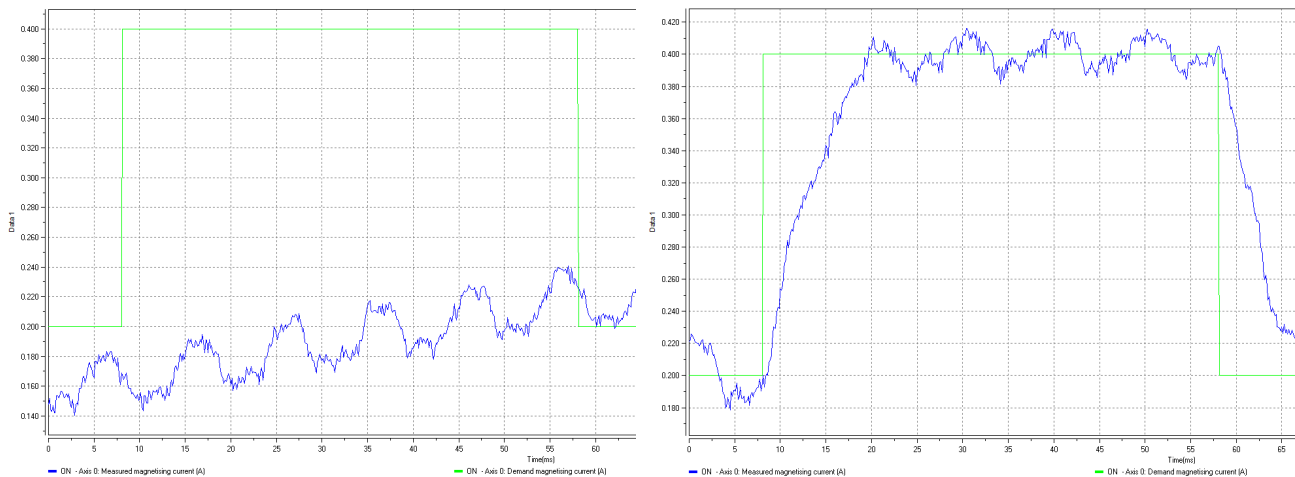
### 3.6 Manual tuning

The following section presumes that no auto tune can be completed but that all the motor data and scale factors have been entered (this can be done either by the wizard, the parameter table or keywords applied from a Mint program or via the command line window). As we haven't completed an auto tune, in each instance the gain terms for each loop will be zero so we will be completing a series of incremental changes to the gain parameters.

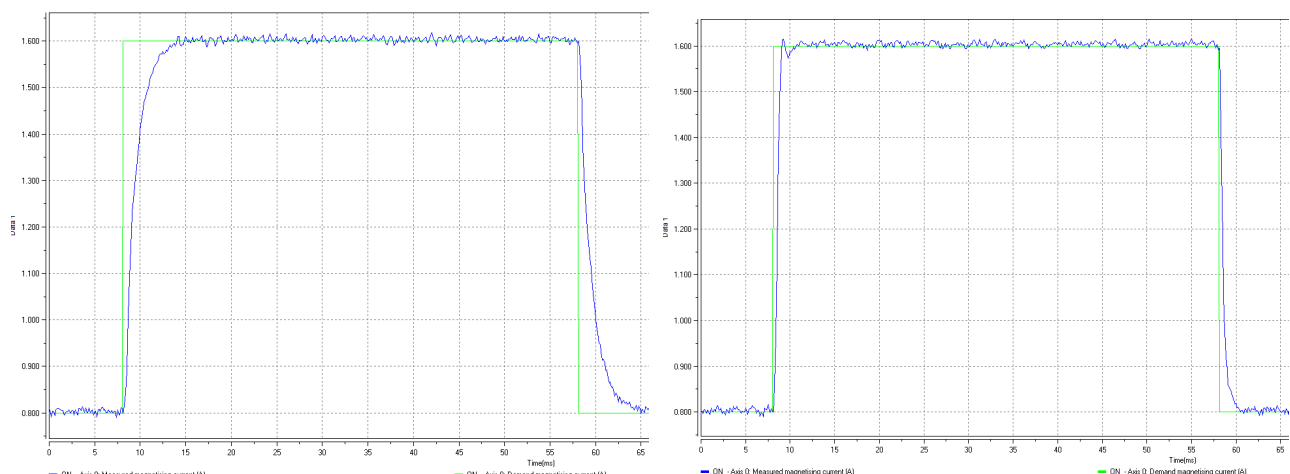
#### 3.6.1 Manual tuning of the current loop

Select the Current tab at the bottom right corner of the Mint Workbench screen to access the current loop test parameters. We are now free to commence the current loop tests. Here we are looking for the Measured current (blue) to be as close to the Demand current (green) as possible. The following steps will guide you through the tuning procedure.

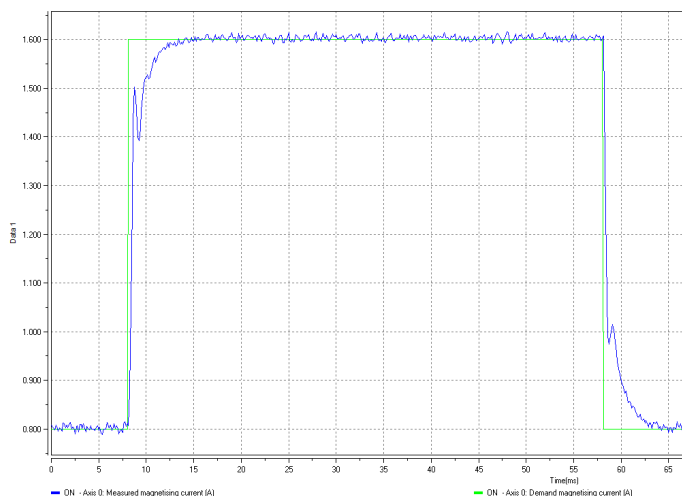
1. The drive current loop terms will be zero so first we must enter some initial values; As a general rule of thumb, KIINIT is usually at least a factor of 20 times larger than KIPROP, so we will start with a KIPROP of 0.05 and a KIINT 1 (KIPROP is usually a very small fractional value so increases in this value should also be fractional)
2. Initially leave the test current set to 25% for 50ms and click the 'Go' button. The motor will perform the move and the captured data will be uploaded and displayed on the screen. If there are no signs of response from the drive measured current at all then start increasing KIINT until there is some sign of a response...



3. If there are no large overshoots or signs of great instability increase the test current to 100%
4. Slowly increase KIINT to improve the sharpness of the response
5. As the response starts to overshoot and/or oscillate use KIPROP to dampen this
6. Keep increasing KIINT and KIPROP, one after the other until the response looks good with no large overshoots (try to listen to the motor during the test, if you detect too much audible noise then you have probably set too high values for KIINT and you may need to reduce both terms as a result)
7. Once the response is fairly good (with no signs of excessive audible noise either) you can now look to add KITRACK to remove the 'curves' from the response. The graph below on the left shows our initial manually tuned result and the one on the right shows the end result after adding KITRACK...



- If you experience a “kink” in the step response when adding KITRACK try reducing your value of KIPROP slightly, you may find that this is dampening the effect of the tracking factor too much. The graph below illustrates the effect of KIPROP being too high for the corresponding KITRACK...

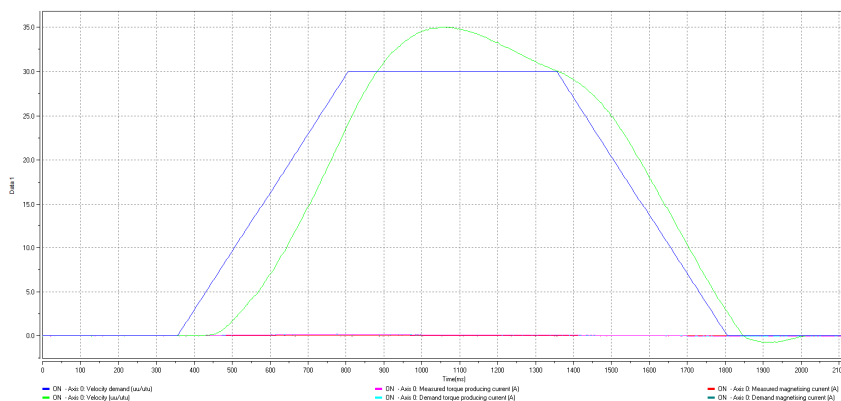


### 3.6.2 Manual tuning of the velocity loop in velocity servo configuration

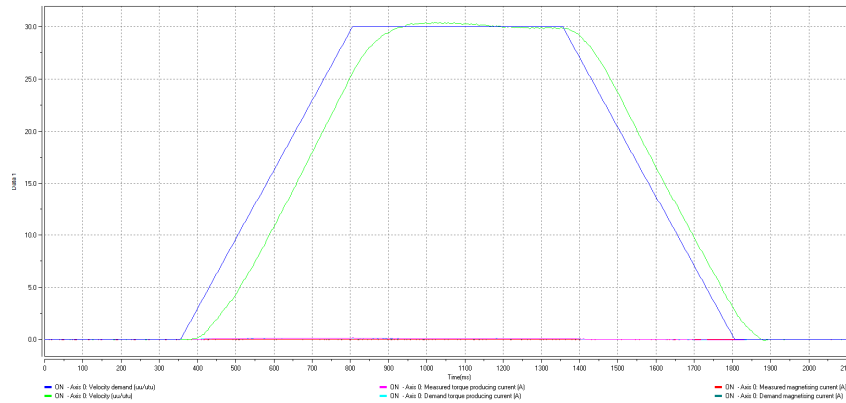
Before starting to manually tune the velocity loop it is critical that the drive is able to complete the feedback alignment test ('Test the feedback' on the 'Auto Tuning' page or via the 'Feedback Alignment – Test' button on the Current loop fine tuning tab). The feedback test should ideally be performed off-load. If this is not practical be aware that the motor's torque will be reduced by a factor proportional to the cosine of the angular error in the feedback alignment - e.g. if the feedback alignment is “wrong” by 10 degrees the motor will only achieve 98.4% (cosine of 10 degrees = 0.984) of its expected torque for a given current. It is not necessary to manually tune the velocity loop if running in torque servo mode (CONFIG = 6 or `_cftORQUE_SERVO`)

To manually tune the velocity loop, first select the 'Velocity' tab at the bottom right corner of the Mint Workbench 'Fine tuning' page. Here we are looking for the measured velocity (blue trace) to be as close to the demand velocity (green trace) as possible. The following steps will guide you through the tuning procedure.

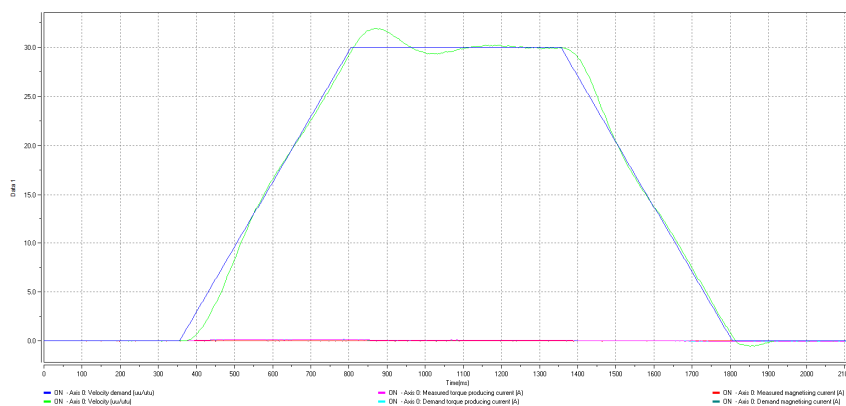
- The drive velocity loop terms will be zero so first we must enter some values to start with. Set KVPROP to 0.1, KVINT to 1, KVTIME to 0 and KVTRACK to 100
- Configure the velocity test move parameters to settings that will be “typical” for your application
- Click the Go button and examine the initial response. As with the Current loop, the integral gain (KVINT) will improve the response to the demand and the proportional gain (KVPROP) will dampen any overshoots. The graph below shows our initial response...



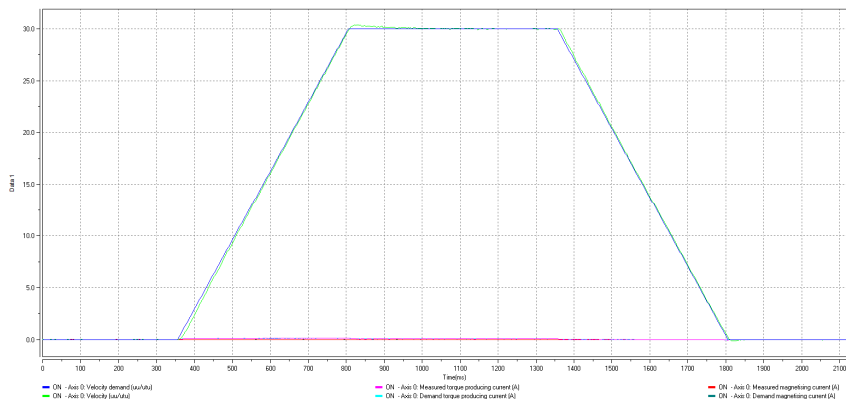
- As we can see there is some overshoot in the measured velocity we will initially increase KVPROP to remove this, but not adding excessive KVPROP such that the response becomes overdamped – a little overshoot at this stage is fine...



5. Now the response is generally the same sort of “shape” we can concentrate on removing the lag/delay in the response by slowly increasing KVINT. As we add KVINT we will improve the response but we will also re-introduce some overshoot into the velocity profile...



6. Increase KVPROP again to remove the overshoot. Keep repeating this cycle of increasing KVINT and KVPROP until you have the desired performance. It is often a good idea to start the test with the axis enabled so that it stays enabled at the end of the test move – this will help to reveal any signs of oscillation at zero speed. If the axis starts to oscillate/vibrate (or audibly buzz) at zero speed you should reduce KVPROP...



7. If you are using a resolver feedback device then you may want to consider using KVTIME to filter the measured velocity. This can help to smooth the velocity response as it will stop the drive reacting to high frequency measurement errors caused by electrical noise and analog conversion errors (Note this does not apply to drives using the resolver adaptor module, OPT-MF-201...this device provides the drive with a pure digital signal)
8. Once complete you can save the parameters, the drive is now ready to have its position loop tuned



Note: The load model parameters on the velocity fine tuning tab are irrelevant when using this manual tuning method. These values are only required to allow the auto tune to calculate velocity loop gains. If you get a message about the test velocity exceeding maximum velocity check the application maximum speed is set correctly in the Operating Mode pages

### 3.6.3 Manual tuning of the position loop in velocity servo configuration

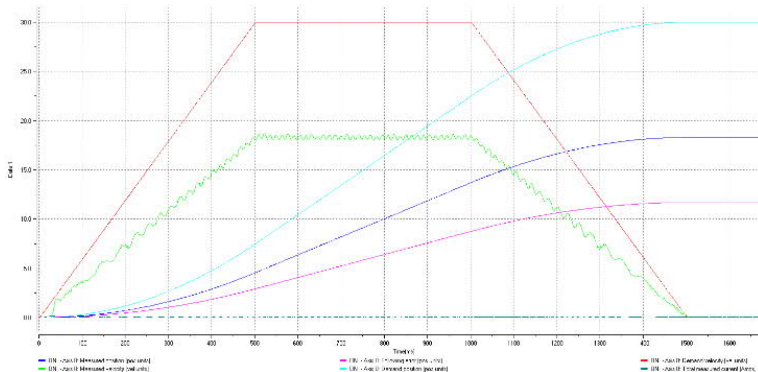
This section assumes the motor is on load, within its limits and is free to rotate in either direction and requires that the drive is set in “velocity servo” configuration (CONFIG(0) = 1 or \_cfSERVO).

Select the Position tab within the Fine tuning work area. We will now perform a test positional move to determine how closely the actual position follows the demand position.

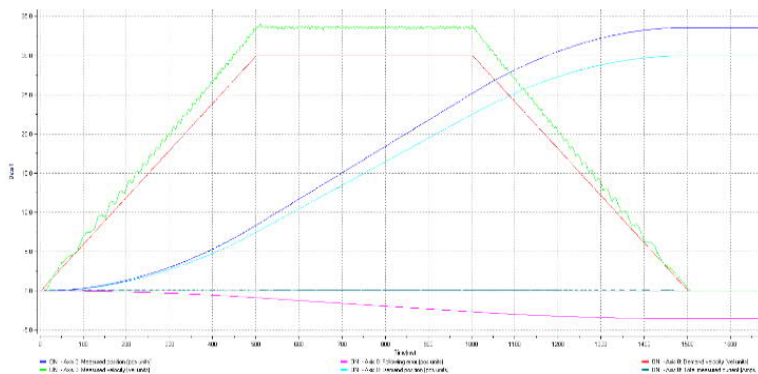
When analysing these graphs we need to check that position demand (turquoise) and actual position (blue) match closely. When they appear co-incident, when zoomed out, we can further emphasise the difference by scrutinising the ‘following error’ (pink) which can be plotted on an independent Y axis to give us detailed information.

When using the velocity servo configuration the most important gain to set first is KVELFF (the velocity feedforward term). When KVELFF is set correctly the demand and measured positions should almost overlap. Use the following procedure to manually tune the position loop...

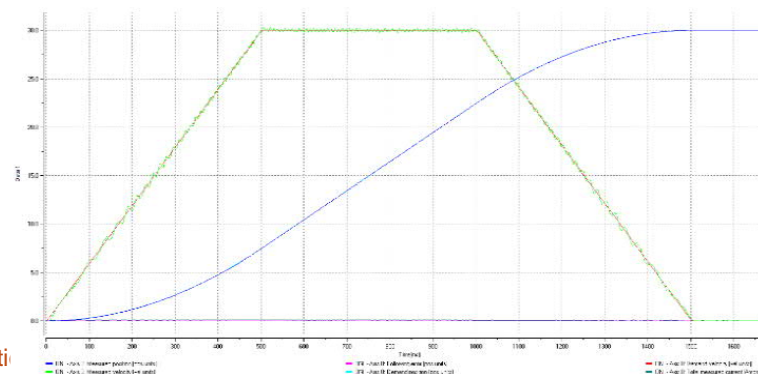
1. Set FOLERRORMODE to 0 (\_emIGNORE) initially. Once the following error trace shows that performance is generally within the limit set by FOLERRORFATAL you should set FOLERRORMODE back to 1 (\_emCRASH\_STOP\_DISABLE) or a mode suitable for the application
2. Set all position loop gains except KVELFF to zero....start with a KVELFF value of 1
3. Using acceleration, deceleration and speed values that will be “typical” for the application configure a test move and trigger this. The drive will perform the move and the captured data will be uploaded and displayed on the screen. This test will probably end up quite unresponsive (expect to see a large following error)
4. If the measured position falls short of the demanded position then increase KVELFF, if the measured position exceeds the demand position reduce KVELFF.....repeat the test move and adjust KVELFF until the traces appear to overlap



KVELFF too low – measured position does not reach the demanded position (and slew speed during the move is lower than the demanded velocity)



KVELFF too high – measured position exceeds the demanded position (and slew speed during the move is higher than the demanded velocity)



KVELFF set correctly – measured position and demanded position overlap (as does the slew speed during the move)



- Once KVELFF is set correctly we can start to add some proportional gain to remove the remaining following error (at this point you will want to give the following error trace its own Y axis on the graph). Start with a value of 0.01 and increase slowly, looking to minimise the following error but without causing instability in the axis. At this point you are effectively fine tuning the position loop so you can refer to section 3.5.3 (Fine tuning the position loop in velocity servo configuration) for further details about the effect of each position loop gain

### 3.6.4 Manual tuning of the position loop in torque servo configuration

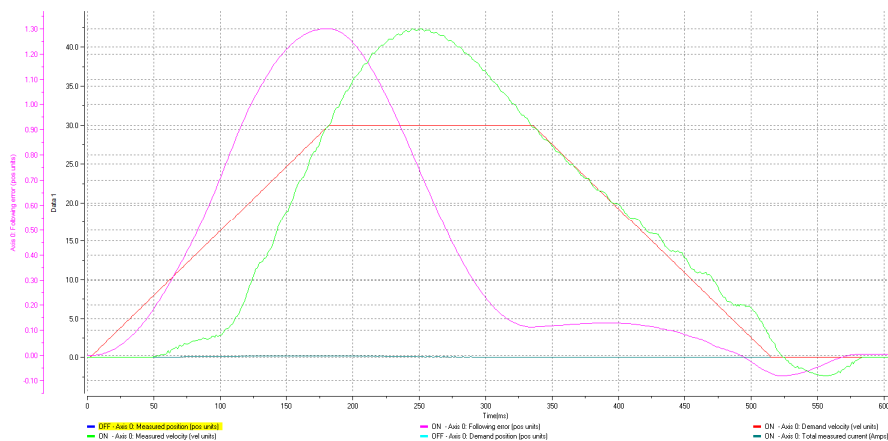
This section assumes the motor is on load, within its limits and is free to rotate in either direction and requires that the drive is set in “torque servo” configuration (CONFIG(0) = 6 or \_cFTORQUE\_SERVO).

Select the Position tab within the ‘Fine Tuning’ page. We will now perform a test positional move to determine how closely the actual position follows the demand position.

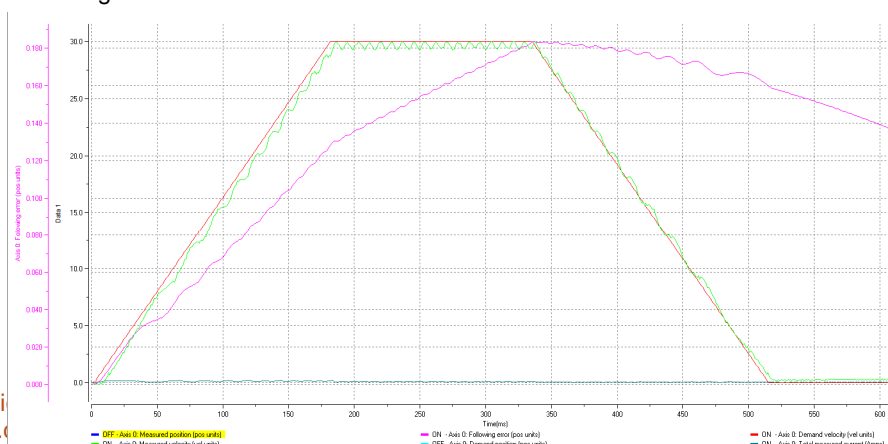
When analysing these graphs we need to check that position demand (turquoise) and actual position (blue) match closely. When they appear co-incident, when zoomed out, we can further emphasise the difference by scrutinising the ‘following error’ (pink) which can be plotted on an independent Y axis to give us detailed information.

When using the torque servo configuration the most important gains to set first are KPROP and KDERIV. Use the following procedure to manually tune the position loop...

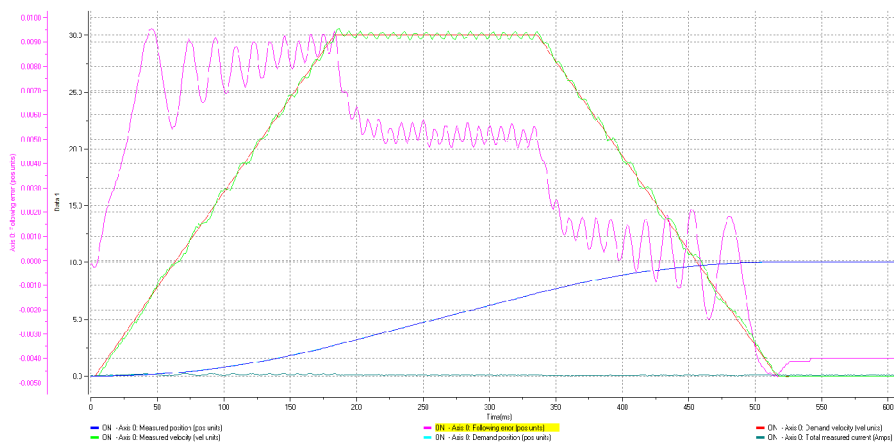
- Set FOLERRORMODE to 0 (\_emIGNORE) initially. Once the following error trace shows that performance is generally within the limit set by FOLERRORFATAL you should set FOLERRORMODE back to 1 (\_emCRASH\_STOP\_DISABLE) or a mode suitable for the application
- Set all position loop gains except KPROP and KDERIV to zero....start with a KPROP value of 0.1 and a KDERIV value of 1
- Using acceleration, deceleration and speed values that will be “typical” for the application configure a test move and trigger this. The drive will perform the move and the captured data will be uploaded and displayed on the screen. This test will probably end up quite unresponsive (expect to see a large following error)



- Keep increasing KDERIV to improve the response of the axis. You should see the measured velocity start to track the demanded velocity much more closely. Once increasing KDERIV stops having a noticeable effect it is time to start increasing KPROP

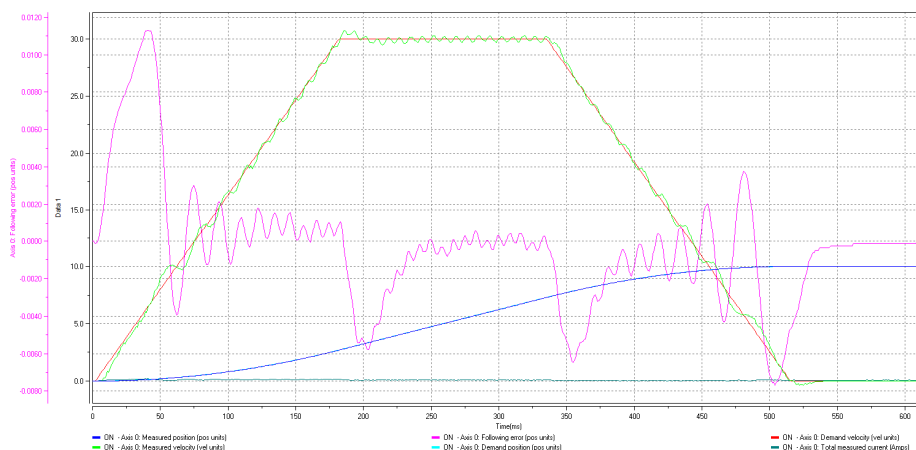


KDERIV adjusted until no further improvement – KPROP still set to a low value



KPROP adjusted until there is a small following error during accel and decel and a constant following error offset from zero during slew speed

- When KPROP is set correctly there will be some kind of following error during acceleration and deceleration (typically with opposite polarity) and a fixed (offset) following error during the slew portion of the test move. At this point we need to introduce the integral term KINT (be sure to set KINTLIMIT to 10-15% and KINTMODE to 'Always' before adjusting KINT. With KINT set correctly the following error at all times should be centred around zero



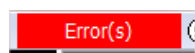
- If required the following error peaks during acceleration and deceleration can now be removed by adding acceleration feedforward gain (KACCEL). At this point you are effectively fine tuning the axis so refer to section 3.5.4 (Fine tuning the position loop in torque servo configuration) for further information about the effect of each position loop gain

#### 4 Debugging and troubleshooting during tuning

It is possible that, occasionally, the tuning procedures may 'fail' and report some kind of error. This might be to do with a mechanical concern (such as inertia mismatch or compliance issues), an installation/cabling issue or sometimes it can just be that the move profiles you have selected cannot be achieved by the system you have. The following sections provide some general guidance if you get an error during any part of the auto tuning process.

Be sure to check initially that the CURRENTLIMIT, TORQUELIMITPOS and TORQUELIMITNEG are set correctly and that the motor brake is released (or a MOTORBRAKEMODE, MOTORBRAKEOUTPUT etc... has been configured correctly) if a braked motor is used.

Some errors will be reported as auto tuning errors (error codes 4000-4999). However, whenever a drive error is generated as a result of unexpected behaviour during the auto tune (e.g. overcurrent trip or a digital encoder is not communicating) a red error banner will appear in Workbench.



Click on this to see further information about the error that is active. You can also use the Mint Workbench Error Log page to view information about past and active errors.



The information in the error log can be very useful in diagnosing the history of errors, and is especially useful when multiple errors occur at the same time. Sometimes the root cause of the problem is not apparent without this information.

#### 4.1 Debugging 'Measure stator resistance and inductance'

If this test fails the best step to a resolution is to manually tune the Current Loop (please see section 3.6.1 for more details). There is little we can do to debug a problem that is detected here. Sometimes the electrical characteristics of the motor (i.e. very low resistance or inductance) mean that the automatic tests for this will not work.

#### 4.2 Debugging current loop tuning

If the calculation of current loop gains fails check the Options dialog for this test (i.e. check the bandwidth is not too low or too high – usually the default value of 2000 rad/s is sufficient, but values in the range 1000-2000 are typical). If gains have been calculated but there is an error when fine tuning this loop (or if manually tuning this loop because the resistance and/or inductance was not known) then we should check for:

- Motor power cable wiring issues - check carefully for correct installation and shielding of the motor power cable
- Failed motor windings – remove the motor from the drive and test the motor windings for short and open circuits



Note: No capture data exists for the auto tune of the current loop so we cannot use Tools>Upload capture data to gather additional diagnostic information

If you cannot tune the current loop there is no point attempting to continue with the tuning process – please contact your local ABB support team for further assistance.

#### 4.3 Debugging 'Test the feedback'

Common causes for errors during test the feedback are;

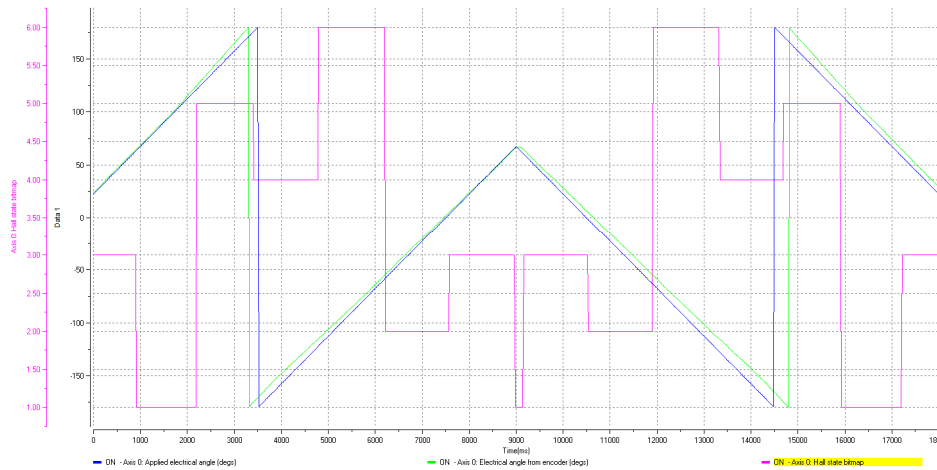
- Incorrect feedback device or motor part number selected
- Feedback wiring – check carefully for correct installation and shielding of both motor power and feedback cables
- Locked rotor shaft – check that the mechanics of the system are free to rotate (check brakes, gearboxes etc.)
- Coupling issue – the coupling between the feedback device and the motor could have failed or come loose
- Failed feedback device – feedback devices are often fragile and can be damaged by a number of factors including mechanical impact, electrical interference and hot plugging
- Incorrect number of motor poles or feedback resolution entered – if a custom motor has been configured the feedback test checks that the correct number of counts are detected when moving the rotor a known distance in each direction. If the number of poles or the encoder resolution is not entered correctly the drive will detect a “mismatch” and report an error
- Missing hall sensor signal (when using incremental encoder with halls as the encoder type) – the test will report ‘The hall sequence doesn't behave as expected through one electrical cycle’. Could be due to a faulty encoder or poor wiring connection



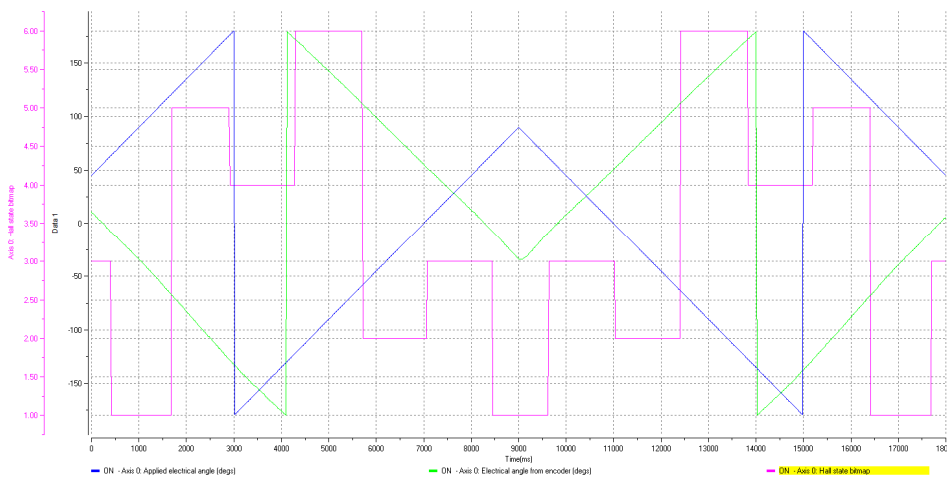
Note: If the motor or feedback is wired incorrectly (motor phases or encoder channels in the wrong order) then this may pass the test as the drive will “autocorrect” for this wiring. However, the resulting parameters will then only suit equally “miswired” motors in future

In the event of an error during test the feedback use Tools>Upload capture data to examine the response of the feedback device during the test, this may help to reveal the root cause of the fault. The data that is uploaded is the applied electrical angle, the angle of the encoder and, when using incremental encoder with halls, the hall state bitmap.

The important trend to look for is that the applied electrical angle and the electrical angle derived from the encoder be ‘parallel’ (note that the derived angle may be “flipped” in the Y axis if the drive detects that ENCODERMODE needs inverting, but if you were to imagine mirroring this it should follow the same general pattern as the applied electrical angle).

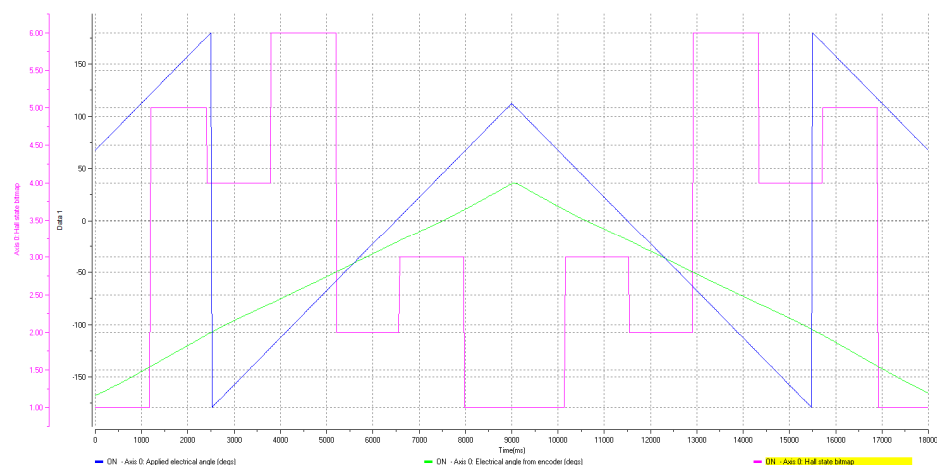


Capture data for encoder with halls when default ENCODERMODE is correct



Capture data for encoder with halls when default ENCODERMODE is incorrect (test will pass and drive will automatically set the correct value for ENCODERMODE)

If the two angles don't follow each other then either the entered feedback resolution or number of motor poles is incorrect. The hall state (if used) should show values in the range 1 to 6 only. If a 0 or 7 is shown then either the encoder is faulty or there is a wiring fault on the hall cabling.



In this capture upload the number of motor poles is wrong (too small) so the electrical angle derived from the encoder shows less travel

#### 4.4 Debugging 'Measure the voltage constant'

The voltage constant test is only required when the voltage constant of the motor is not already known. If the voltage constant cannot be measured then it will be necessary to manually tune the speed and position loop gains.

Possible causes of errors during this test are;

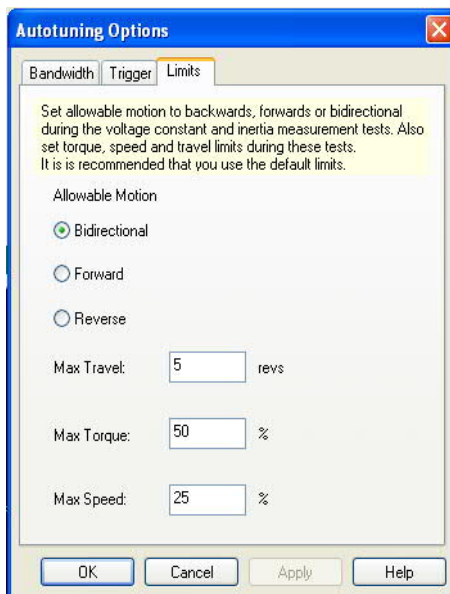
- Motor power cable wiring issues - check carefully for correct installation and shielding
- Failed motor windings – remove the motor from the drive and test the windings for short and open circuits
- Undersized motor or drive – upload capture data and check how much current is demanded to move the motor during the test. Also check the Options for the test and try increasing the 'Max Torque' setting for the test
- The Options may need to be modified for uni-directional testing only or the maximum distance may need increasing or the test speed decreasing. This can be the case if the mechanical system only allows movement in one direction or if the motor has high inertia for example

#### 4.5 Debugging 'Measure the inertia'

This test will rotate the motor backwards and forwards (depending on the test options that have been configured) several times to measure the response of the load to an applied (known) torque/current. By applying a known torque and measuring the resulting acceleration the drive can calculate the total inertia. The drive needs to complete this test successfully in order to automatically calculate the speed and position loop gains. If it's not possible to complete this test the speed and position loop gains must be tuned manually.

Possible causes of errors during this test are;

- If the load/inertia is high there may not be enough torque delivered during the test to move the motor - select the "Options" button for this test and then choose the "Limits" tab. Increase the percentage "Max Torque"
- If the load is very light / low inertia you may need to reduce the "Max Torque" setting or increase the "Max Travel" setting to prevent the test exceeding the max travel
- Can the load physically move in both directions? If not ensure that the correct direction settings have been made
- If the load has high inertia or high compliance it may not be able to reverse direction easily, so if an error occurs try setting the test direction to forward or reverse only



#### 4.6 Mechanical considerations

When using a servo drive/motor it is very important to ensure that the mechanical transmission system is suitably designed/constructed to suit the dynamics expected from it.

##### 4.6.1 Inertia mismatch

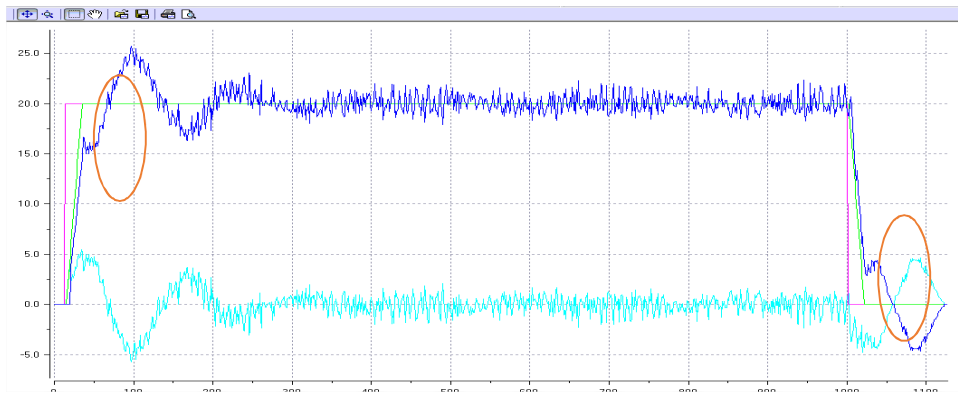
The total reflected load inertia should be well matched to the rotor inertia of the selected motor and there should be a very rigid coupling from the motor to the load. In general the higher the motor feedback resolution the more likely the drive is to cope with a higher inertia mismatch, but as a 'rule of thumb' it's good to aim for an inertia mismatch (load:motor) of between 1:1 and 3:1 for applications requiring very precise position control. Applications requiring reasonable position control or just very basic speed control can cope with a mismatch of 10:1.

### 4.6.2 Mechanical compliance and Adding torque filters

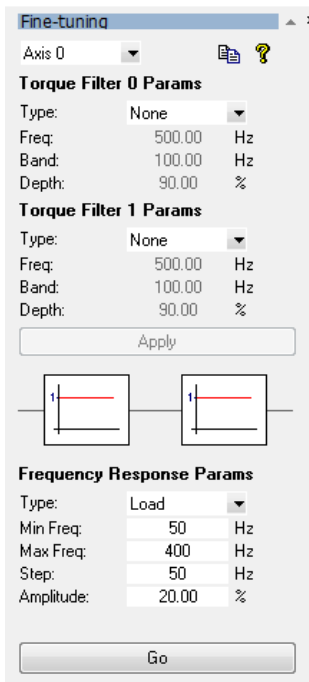
The quality of the tuning we can achieve is ultimately defined by the mechanical system connected to the motor shaft. If the mechanical connection between the tooling and the motor is not stiff we can see problems. Mechanical compliance is the inverse of stiffness in the system. It can be caused by poorly meshed or worn gears or drive belts that are incorrectly tensioned.



Poor mechanical compliance can reduce the performance and accuracy of the system and ultimately can hinder how well we can tune it. Below is a velocity loop tuning graph that shows a system with poor mechanical compliance. There is a loss of control during acceleration and during deceleration where the mechanics 'jump back'. If this problem is suspected then the machine should be powered down and the mechanics should be checked and improved if possible.



If it is suspected that there are mechanical resonant frequencies in the transmission then the 'Fine tuning' pages include a 'Filter' tab which allows the drive to apply a range of vibrations to the load at varying frequencies (changed in defined steps). The drive will then measure the resulting motor movement and plot a graph of motor measured movement against expected movement (as a ratio) over the range of applied frequencies.



In this example 'Filter' test we will oscillate the motor/load over the range 50Hz to 400Hz (in 50Hz steps) to measure the resulting movement of the motor shaft. The test sets a maximum frequency of 1kHz by default, but it's very rare for any mechanical system to resonate above 400Hz.

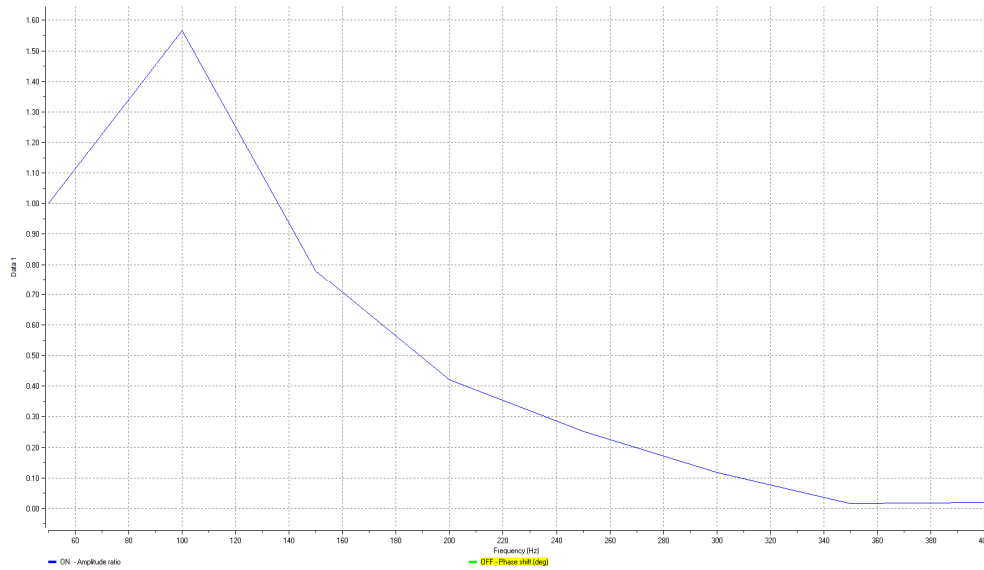
If there are any mechanical resonances in the system (e.g. caused by compliance in a drive belt) these will show up as a "spike" in the resulting graph uploaded automatically at the end of the test. If any "problem" frequency ranges are detected the test can be repeated with a more focused range and step size.

Spikes can be removed by applying a notch filter, bands of problem frequencies can be addressed by using a low pass filter.

The drop downs for the two available filters let you set these easily, they ultimately set the drive parameters TORQUEFILTERTYPE, TORQUEFILTERBAND and

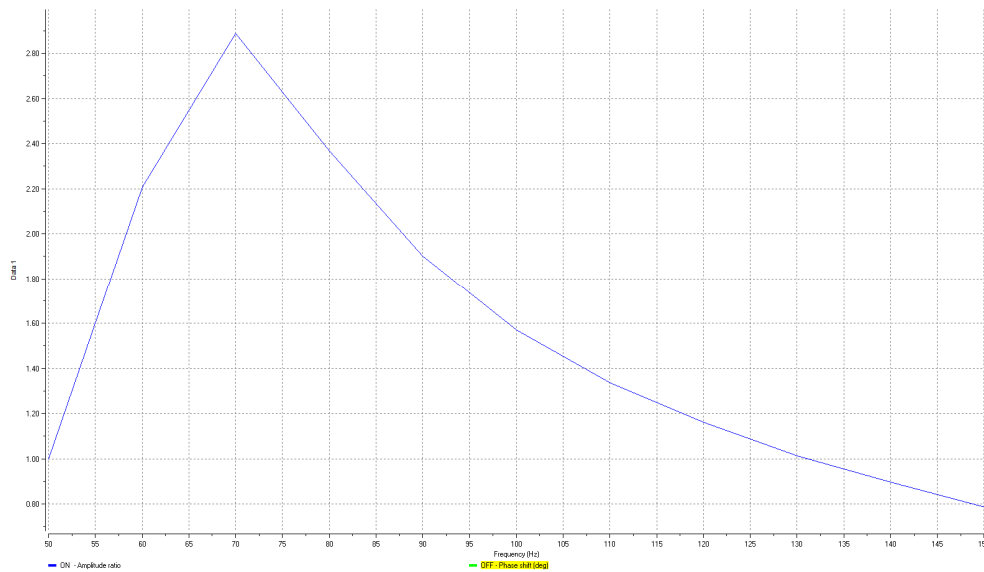


In our test results below we have turned off the 'Phase Shift' trace to examine the Amplitude ratio...

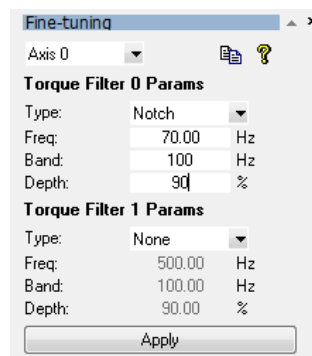


In a system with no mechanical resonance the amplitude ratio should start at 1 and decay to zero as the frequency increases. However, in our test we can see an increase in amplitude ratio at a frequency of 100Hz (our test was performed on a system where the motor was driving a flexible rubber belt via some pulleys and the belt resonates at somewhere close to 100Hz).

Now we know there is a problem between 50 and 150Hz we can repeat the test over this lower range and decrease the steps to 10Hz...



Now we have focused on this range we can see the actual resonant frequency is 70Hz (not 100Hz).





At this point we can apply a Notch filter, centred on 70Hz, with a band of 100Hz (50Hz either side of 70Hz) and a depth of 90% to filter the torque demand and ensure we don't excite our mechanical system into resonance.



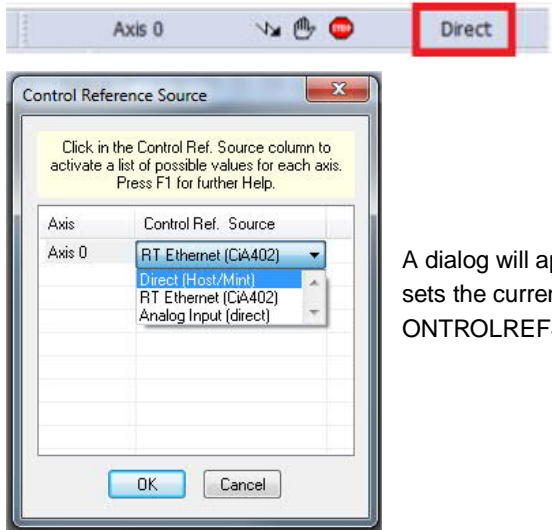
### 5 Finalising the commissioning

As long as no faults exist and all loops are responding correctly then we can consider that the tuning of the drive is complete.

 Remember to turn FOLERRORMODE back to the required application setting if you have set it to 0 (\_emIGNORE) during tuning

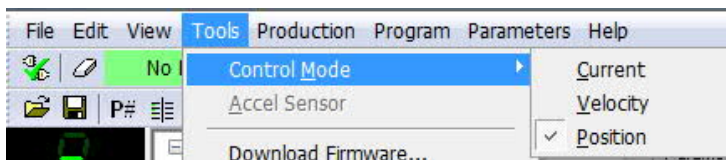
 Don't forget to save any parameter settings you have modified by clicking the save icon on the tool bar or going to Tools>Store Drive parameters

When the commissioning of the drive is complete the drive needs to be reconfigured to use its normal control reference and operating mode. The simplest way to do this is to power cycle the drive (the settings you've made during commissioning will then be applied), but it can also be done by clicking on the control reference source button in Workbench (which will currently show 'Direct') and changing this as necessary...



A dialog will appear where you can now select the required control reference (note that this sets the current CONTROLREFSOURCE parameter and not ONTROLREFSOURCESTARTUP which is what the commissioning wizard configures)....

From the Tools menu now check/select the required operating mode (Position/Velocity/Current)...



At this point it is a good idea to either upload the drive parameters (Tools>Parameter Table>Upload) or for a more comprehensive backup, create an archive of the drive.

To do this connect to the drive, select Tools>Controller Archive>Create, then select all the elements that you want to be included, select Prepare archive, find a suitable folder on the PC and give the archive file a name "Axis 1 Archive.zip" for example.

#### 5.1 Further support

If you have followed the advice in this application note and you are still struggling please contact your local ABB technical support team. The team may be able to provide remote support to help with tuning if necessary.

Please email;

Asia: [cn-motionsupport@cn.abb.com](mailto:cn-motionsupport@cn.abb.com)

Rest of World: [motionsupport.uk@gb.abb.com](mailto:motionsupport.uk@gb.abb.com).

#### Contact us

For more information please contact your Local ABB representative or one of the following:  
**[new.abb.com/motion](http://new.abb.com/motion)**

© Copyright 2017 ABB. All rights reserved.  
Specifications subject to change without notice.

[new.abb.com/drives](http://new.abb.com/drives)