ROBOTICS

Application manual

TuneMaster

Trace back information:
Workspace 20B version a3
Checked in 2020-04-24
Skribenta version 5.3.033

Application manual
TuneMaster
TuneMaster 6.11

Document ID: 3HAC063590-001
Revision: D

Keep for future reference.

Additional copies of this manual may be obtained from ABB.

Original instructions.

Table of contents

This page is intentionally left blank

Overview of this manual

About this manual

This manual describes how to use the PC-based software TuneMaster.

Usage

This manual can be used as a brief description of how to install, configure, tune and log additional axes and robots. It also provides information about related system parameters. Detailed information regarding system parameters, RAPID instructions, and similar, are not described here, but can be found in the respective reference manual.

Who should read this manual?

This manual is primarily intended for advanced users and integrators.

Prerequisites

The reader should...

- be familiar with industrial robots and their terminology
- be familiar with controller configuration and setup
- be familiar with the mechanical and dynamic properties of the controlled mechanism.

References

Reference	Document ID
Application manual - Additional axes and stand alone controller	*3HAC051016-001*
Application manual - Controller software IRC5	*3HAC050798-001*
Application manual - Servo Gun Setup	*3HAC065014-001*
Operating manual - IRC5 with FlexPendant	*3HAC050941-001*
Operating manual - RobotStudio	*3HAC032104-001*
Technical reference manual - RAPID Instructions, Functions and Data types	*3HAC050917-001*
Technical reference manual - System parameters	*3HAC050948-001*
Product manual - IRC5	*3HAC047136-001*
Product manual - Motor Units and Gear Units	*3HAC040148-001*
Product specification - Controller IRC5 with FlexPendant	*3HAC041344-001*
Product specification - Motor Units and Gear Units	*3HAC040147-001*

Revisions

Revision	Description
A	Released with TuneMaster 6.07. First edition.

Continues on next page

Revision	Description
B	Released with TuneMaster 6.08. • Added section *Tuning Uncalibrated Control Master 0 parameters on page 67*.
C	Released with TuneMaster 6.09. • Updated section *Log settings on page 19* and *Data logging on page 70*. • The name *Auto Tune* is changed to *Tune Estimate*.
D	Released with TuneMaster 6.11. • Information added in introduction that .NET Framework v4.8 must be installed when running TuneMaster.

Product documentation

Categories for user documentation from ABB Robotics

The user documentation from ABB Robotics is divided into a number of categories. This listing is based on the type of information in the documents, regardless of whether the products are standard or optional.

All documents can be found via myABB Business Portal, *www.myportal.abb.com*.

Product manuals

Manipulators, controllers, DressPack/SpotPack, and most other hardware is delivered with a **Product manual** that generally contains:

- Safety information.
- Installation and commissioning (descriptions of mechanical installation or electrical connections).
- Maintenance (descriptions of all required preventive maintenance procedures including intervals and expected life time of parts).
- Repair (descriptions of all recommended repair procedures including spare parts).
- Calibration.
- Decommissioning.
- Reference information (safety standards, unit conversions, screw joints, lists of tools).
- Spare parts list with corresponding figures (or references to separate spare parts lists).
- References to circuit diagrams.

Technical reference manuals

The technical reference manuals describe reference information for robotics products, for example lubrication, the RAPID language, and system parameters.

Application manuals

Specific applications (for example software or hardware options) are described in **Application manuals**. An application manual can describe one or several applications.

An application manual generally contains information about:

- The purpose of the application (what it does and when it is useful).
- What is included (for example cables, I/O boards, RAPID instructions, system parameters, software).
- How to install included or required hardware.
- How to use the application.
- Examples of how to use the application.

Continues on next page

Operating manuals

The operating manuals describe hands-on handling of the products. The manuals are aimed at those having first-hand operational contact with the product, that is production cell operators, programmers, and troubleshooters.

Safety

Safety of personnel

A robot is heavy and extremely powerful regardless of its speed. A pause or long stop in movement can be followed by a fast hazardous movement. Even if a pattern of movement is predicted, a change in operation can be triggered by an external signal resulting in an unexpected movement.

Therefore, it is important that all safety regulations are followed when entering safeguarded space.

Safety regulations

Before beginning work with the robot, make sure you are familiar with the safety regulations described in the manual *Safety manual for robot - Manipulator and IRC5 or OmniCore controller*.

This page is intentionally left blank

1 Introduction

TuneMaster

TuneMaster is an intuitive and easy to use PC tool for logging and optimizing the motion performance of additional mechanical units such as track motion, gantry robots, rotary axes etc. TuneMaster can tune both standard ABB robots and additional axes.

TuneMaster can perform movements on a target mechanical unit to find the optimal values for the controller parameters. TuneMaster can tune all relevant parameters for axes controlled by *Lag Control Master 0 (LCM0)*. Note that in TuneMaster, it is only possible to tune the parameters *Df* and *Mounting Stiffness* for an ABB robot.

The main features of TuneMaster are:

- Easy view and edit of relevant configuration parameters, see *Using the Cfg Info tab on page 29*.
- *Tune Estimate* function, automatically suggests motion parameter values, see *Using the Tune Estimate function on page 31*.
- Step by step guide for fine adjustment/optimization of each parameter, see *Tuning LCM0 parameters on page 38*.
- Logging of test signals from the system, see *TuneMaster - Log Signals and Servo Gun on page 69*.

The application is divided into different tabs for the different tasks, more information about the different tabs is found in this document.

Installation of the TuneMaster application

The installation can be downloaded from the *RobotStudio Online Community*, where it is included in the *RobotWare Tools and Utilities* package.

When running the TuneMaster application, the .NET Framework v4.8 must be installed.

This page is intentionally left blank

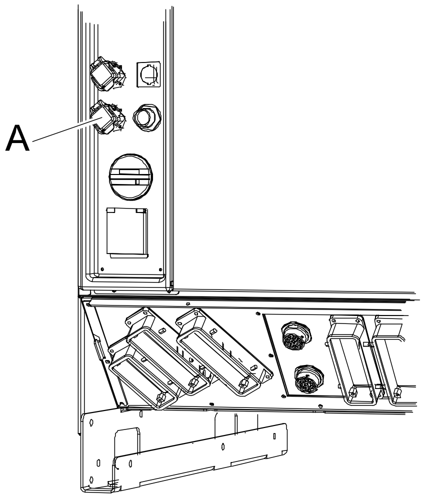
2 Working with TuneMaster

2.1 Connecting a PC to the service port

Procedure

Note

We do not recommend connecting the controller's service port to a LAN (local area network), since a DHCP server connected to service port would automatically distribute IP addresses to all units connected to the LAN. Please contact your local network administrator!

	Action	Note
1	Make sure that the network setting on the PC to be connected is correct. **When connecting to the service port:** • The PC must be set to “Obtain an IP address automatically” or set as described in **Service PC Information** in the **Boot Application** on the FlexPendant. **When connecting to the factory network port:** • The network settings for the PC depend on the network configuration setup by the network administrator.	Refer to the system documentation for your PC, depending on the operative system you are running.
2	Connect a network cable to the network port of your PC.	
3	**When connecting to the service port:** • Connect the network cable to the service port on the controller, or to the service port on the computer unit. **When connecting to the factory network port:** • Connect the network cable to the factory network port on the computer unit.	 connectb A Service port on the controller

2.2 Network settings

Overview

The information in this topic describes the network settings for a PC that shall be connected to a controller. Connecting the PC to the controller is necessary for working with TuneMaster. You can either connect the PC to the controller through an Ethernet network or through the controller's service port. When connecting to the controller's service port, you can either obtain an IP address for the PC automatically, or you can specify a fixed IP address.

Ethernet network connection

If the controller is connected to an Ethernet network, you can connect the PC to that network. When the PC and the controller is connected correctly, the controller will be detected by TuneMaster automatically. The network settings to use on the PC depends on the network configuration. To know how to set up the PC, contact the network administrator.

Service port connection with automatic IP address

The controller's service port has a DHCP server that will automatically give your PC an IP address if it's configured for this.

For more information on how to configure the PC to obtain IP addresses automatically, see Windows help.

Note

Obtaining an IP address automatically might fail if the PC already has an IP address from another controller or Ethernet device. To ensure that you get a correct IP address if the PC was previously connected to an Ethernet device, do one of the following:

- Restart the PC before connecting to the controller.
- Run the command *ipconfig /renew* from the command prompt after connecting the PC to the controller.

Service port connection with fixed IP address

Instead of obtaining an IP address automatically, you can also specify a fixed IP address on the PC you connect to the controller. Use the following settings for connecting with a fixed IP address:

Property	Value
IP address	192.168.125.2
Subnet mask	255.255.255.0

For detailed information about how to set up the PC's network connection, see Windows help on Configure TCP/IP settings.

Continues on next page

Connecting

Once the PC is correctly set up, enter the IP address of the controller and click **Connect**.

Property	Value
IP address of controller on service port	192.168.125.1

For information about how to connect the PC to the controller's service port, see *Connecting a PC to the service port on page 15*.

2.3 Plot settings

Options for plotting

Some settings of the plot can be changed. Click the **Tools** menu and select **Options**.

The following options for plotting are available:

- Number of significant figures
- Number of axis labels on plotter
- Line Width
- Show Grid
- Keep Zoom
- Background color

When switching between measurements in the list of measurements the **Keep Zoom** option can be useful. This will try to zoom the same area/limits when another measurement is selected. If this option is not active all data will be displayed when another measurement is activated.

The plot settings described above can be changed in two ways. Changes of this in the main window will not be stored when restarting the program. If changes will be done for further use, use these general Options.

It is possible to apply a low pass filter to the signal values in the plot. Filtering is not applied to exported or saved data, just in the plot. It is also possible to select if the plot should be updated during logging or only when the logging is stopped.

2.4 Log settings

Description

Select **Log Type**. **InfoStream** is the only supported type from RobotWare 6.09. To get 0.5 ms sampling rate Socket can be used on earlier versions.

Select **Sample time**. The actual sample time for each signal is adjusted to the nearest multiple of this sample rate.

Anti Alias Filter can be used if 4 ms is selected to filter signals with 0.5 ms sample rate.

Zero Hold can be used to hold the value until a new measurement is received if the signal has lower sample rate than selected. If **Zero Hold** is not used the value will be linearized between two measurements.

Log to file can also be activated. This can be used if log sequence will be very long. With this option all data will be saved to file but only the latest will be displayed by TuneMaster.

The max length of the log time buffer can be defined here. If the limit is zero, the buffer will have no limit. This option can be useful in combination with a signal trig to avoid that TuneMaster will use all the memory of the PC if logging is active for a long time.

2.5 Working with the plot

Zooming

There are a number of tools available for zooming by using the mouse and the keyboard.

To zoom in, click with the left mouse button. To zoom out, click with the right mouse button. The scrolling wheel can also be used. To zoom in on a part of the plot, click and drag to draw a square (box zoom). The box zoom can be changed to zoom in just X or Y with the **Zoom** button above the plot. Quick access of the zoom modes can be done with the X, Y or B keys on the keyboard.

To reset the zoom, double-click on the plot. To zoom out in X or Y direction, click **Reset X Zoom** or **Reset Y Zoom** above the plot. Quick access of the reset can be done with **SHIFT** and the **X** or **Y** key on the keyboard.

To pan or move the zoomed area, press **CTRL** and then click and move the mouse on the plot. Quick access of the move can be done with the **Move** button above the plot or the **M** key on the keyboard. Switch back to zoom mode with the **Z** key.

Limits for each axis can be changed manually by clicking on the end of each axis. Enter the axis limit in the displayed text box.

Copy data

The zoomed data can be copied in two ways. The plot can be copied as an image to be used in a mail or in a slide show etc. The data can also be copied for external calculations in a spread sheet program etc. Click the **Tools** button above of the plot to select copy method.

Crop log

With the **Tools** button it is possible to crop/cut the logged data by selecting **Crop Log**. Data will be reduced to the zoomed X axis. This can be useful if you have logged the same cycle several times but have different time, then you can use this function to reduce to the same X axis for all measurements and then compare these in an easy way.

Split log

With the **Tools** button it is possible to split the logged data into two measurements by selecting **Split Log**. The data will be split to two measurements, where the split will be in the right side of current zoom. This can be useful if you have logged the same cycle several times in one measurement but want to compare them with each other in the summary.

Data cursors and comments

To show the coordinates for a position, press **SHIFT** and click on the position in the plot. When clicking on the next position, the tool tip also displays the distance to the previous position.

Data cursors can be added with the **Add Data Cursor** button above the plot. Comments can be added with the **Add Comment Cursor** button. To change the comment, right-click on the cursor and select **Edit**.

Continues on next page

An added cursor can be moved or deleted when it is active. The cursor is active when it has bold lines. To move it, just drag and move with the mouse. To delete, press the **Delete** key or right-click on the cursor and select **Delete**.

Data cursors have a snap mode. The default mode is **Line**, which means it will snap to closest line. With the **Tools** button above the plot, this mode can be changed to **None** or **Point**. When **Point** is active it will snap to closest data point.

To copy the cursors in text format to use in another program, click on the **Tools** button and select **Copy Cursor data**.

2.6 Measurements

Finding measurements

In the **Edit** menu, it is possible to search for measurements/comments. All measurements can be listed as favorites, these can also be found with this tool. In the **View run** part of the program, it is also possible to enter a filter from the **Settings** button. When a filter is used, the list will only contain measurements with comments that contain the filter text.

Displaying information about measurements

The measurements also have additional information, such as **System name**, **Measurement date**, and **Log type information**. Click **Information** in the **View run** part to display the additional information.

Each logged signal also has information such as **Min**, **Max** and **Mean value**. Right-click on one signal in the signal table and click **Information** to get this information.

Comparing measurements

It is possible to compare measurements and signals in several ways. Each signal can be scaled to make it similar to other signals. Right-click on the signal in the signal table. A menu will be shown where it is possible to enter **Offset**, **Scale** and **Delay**. The scale factor will be written as a text in the first column of the signal table. In this menu it is also possible to change **Signal color** and **Derive signals** etc.

To compare two or more measurements, the **Summary View** can be used. This view is started from the **View run** part of the program, click the **Sigma**/**Summary** button. Select the measurements to compare and click the **Compare** button. It is possible to show/hide and scale all signal here too, equal to the signal table. Here it is also possible to synchronize several measurements, right-click on one signal to the left and select **Sync Logs with This**. This summary with several measurements can also be printed.

To make it easier to find some measurements the filter in the bottom of the view can be used to filter in the comments for each data set. It is also possible to add a comment to several measurements at the same time from this view.

Save data

To save all measurements in a project file, click **File** menu and select **Save Tune Project**.

Print data

To print the zoomed data, click the **File** menu and select **Print**. A report containing all, or a selected part of, the measurements can also be printed, for example to a PDF file, by selecting **Print Report**.

Continues on next page

Export data

From the **Tools** menu it is possible to export the data to a text file. Then it can be loaded from an external calculation program etc. It is only the visible zoomed data that will be exported. Note that any **Low Pass** filtering is not applied to the exported data.

Import data

From the **Tools** menu it is possible to import a text file. Data can also be imported from other TuneMaster projects from the **File** menu.

This page is intentionally left blank

3 TuneMaster - Robot

3.1 Tuning of Df

Introduction

CAUTION

It is recommended to use the *AutoTune* function. Be careful when doing manual tuning of *Df*. Low *Df* values could cause unstable movements. To minimize the risk, small gradual adjustments downward when *Df* is below 70 % is recommended.

Tuning *Df* can be used to improve the path accuracy by damping oscillations of the axes due to mechanical resonances. In most cases, it is not necessary to adjust *Df*. If the TCP is oscillating during motion, then the performance of the axes can sometimes be improved by adjusting *Df*.

It could be necessary to tune *Df* if the robot for example is mounted on a construction that is not stiff enough, for example when the robot is moved by an *IRBT* track motion or if the robot is a top loader. Another scenario could be that the user would like to have as optimal path accuracy as possible for a specific position within the working range. For this case it is possible to do the tuning close to the interesting position.

In the majority of cases when tuning is needed, that is when compensating for mounting flexibility, only axis 1 and 2 should be tuned. Normal tune values for these cases are 75-95%.

One rare exception is when the robot is holding a very flexible object. Then tuning of axis 4-6 with values in the range 50-95% could be used to reduce vibrations.

To get the best possible result, tuning should be performed with the maximal payload for the application. Tuning without any payload can result in bad robot movements when a payload is attached to the robot.

Tuning procedure

One simple way to tune *Df* is to use to use the **AutoTune**. Follow step 1 to 9 below and then select the **AutoTune** Mode before **Start** is pressed.

1. Enter the IP-address for the controller, or click **Browse** in the **Connect sub** menu to find all available controllers.
2. Click on **Connect** to connect to the controller. After a couple of seconds, the mechanical units connected to the controller should be visible to the left.
3. Select the **Tune DF** tab in the application.
4. Verify that the mechanical unit of interest is selected and activated.
5. Select a joint to tune and click **Choose Joint** or just double-click on the axis.
6. Select the current tool to the upper right. All tools that are defined and available on the robot controller are visible and possible to select.

Continues on next page

7 Select **Movement**: Define **Tune position** and **Tune move**. The robot will move from the tune position with the tune move to the end position and back to start position again. Recommended is one axis joint motion of 2-5 degrees.

8 Select **Speed** for the motion. Default and recommended is **vmax.**

9 Select **Single**, **Multiple** or **AutoTune** Mode. **Single Mode** will execute one single motion with defined *Df* value. **Multiple Mode** will execute a sequence of motions with *Df* values from **Start Df** to **End Df.** The number of motions is defined by **No of Df.** Recommended action is to start with doing a multiple sequence from 90% to 110% with **No of Df** set to 5.

10 The best value for *Df* among the executed sequences will automatically be calculated and presented in the **Result** part of the program.

Update system parameter

If *AutoTune* is executed you will be asked to update robot configuration with optimal value. Parameter can also be updated with help of the **cfg** button. If you update with new optimal value there is no need to update the RAPID program if you don't need a special tuning in a specific position within the working range.

Update RAPID program

To update with the optimal *Df* in a specific position within the working range, the RAPID program needs to be updated with instructions for adjusting the *Df* tuning. Add for example the following instruction before the interesting move instruction in the RAPID program:

```
TuneServo ROB_1, 1, 110 \Type:= TUNE_DF;
```

This will adjust *Df* to 110% for `ROB_1` axis 1.

After the move instruction, the following instruction could be added to reset the tuning parameters to default values:

```
TuneReset;
```

In most cases, the obtained *Df* values are generally valid and should always be used. `TuneServo` can then be executed in an initialization routine and there is no need to run `TuneReset`.

The RAPID instructions are described in detail in *Technical reference manual - RAPID Instructions, Functions and Data types*.

3.2 Tuning Mounting Stiffness

Introduction

Tuning *Mounting Stiffness* can be used to improve the path accuracy and reduce overshoots by damping oscillations due to a foundation with inadequate stiffness (that is the robot is mounted on a weak foundation). In most cases, it is not necessary to adjust the Mounting Stiffness.

A *Mounting Stiffness Factor* around 1.0 is gives the best behavior when the foundation is stiff according to the product manual for the robot, see requirement on foundation - minimum resonance frequency. A lower value will improve the robot behavior when the requirement on foundation is not fulfilled and a lower value means a more flexible foundation. There are three parameters for the x-, y-, and z-direction (torsional stiffness around base coordinate system axes).

To get the best possible tuning result, tuning should be performed with the maximum payload for the application. Tuning without any payload can result in bad robot movement when a payload is attached to the robot. Also, for best result, axis 2 and 3 should be positioned in a position that is typical for the application.

Tuning procedure

Select the **Tune Mounting Stiffness** tab. AutoTune is highly recommended. If you click on the **Start** button, the complete tune procedure below will be done at the same time if AutoTune is the selected mode. Note that the order of *Factors* is critical if manual tuning is performed.

1. Tune the *Y* factor while axis 2 is the active axis and axis 1 is in 0 degrees. Update the text box for **Y** with the optimal value.
2. Tune the *X* factor while axis 2 is the active axis and axis 1 is in ±90 degrees. Update the text box for **X** with the optimal value.
3. Tune the Z factor while axis 1 is the active axis.

Note

If the RobotWare option *Advanced Robot Motion* is installed, the **Motion Process Mode** can be changed during tuning. The default and recommended setting is **Low Speed Stiff Mode.** If it is not installed, and if the result of the obtained **Mounting Stiffness Factors** is not satisfactory, then try to change the **Motion Process Mode** to **Low Speed Stiff Mode** during the tuning procedure.

This page is intentionally left blank

4 TuneMaster - Additional Axis

4.1 Using the Cfg Info tab

Information

The **Cfg Info** tab is used to display and modify the configuration parameters for each mechanical unit in the system. The displayed data is read directly from the controller's configuration database. The **Cfg Info** tab does not display all the values of the system configuration, only the configuration parameters that are relevant for tuning of mechanical units.

Viewing the system configuration

Browse through the tree view by expanding the + signs to see the current configuration. To see more detailed information about each field, possible settings and a brief description, select the node and the corresponding information will be displayed in the info field to the right of the tree view.

Modifying the system configuration

To modify a value of the system configuration:

1 Select the corresponding node in the tree view.
2 Enter the new value for the parameter in the **New value** field.
3 Click the **Update controller Cfg** button.

When a configuration parameter is modified from the **Cfg Info** tab, the new value will be stored in the controller's configuration database.

Color codes in the Cfg Info tab

Values in the **Cfg Info** tab are displayed in three different colors, depending on how the parameter is handled by TuneMaster.

- *Green* parameters are automatically updated by TuneMaster for selected joint when performing tune runs, such as *Tune Kv* or *Tune Inertia*. The parameter is set at the beginning of each tune tune and reset at the end of the tune run. For green parameters it is therefore not necessary to immediately restart the controller. Changes to these parameters are only valid as long as TuneMaster is running, so when exiting TuneMaster it is necessary to restart the system for changes to take permanent effect.
- *Black* parameters can be changed from TuneMaster, but the changes will not take effect until the controller system is restarted.
- *Red* parameters indicate that a black parameter has been changed and is out of date. The value displayed in the **Cfg Info** tab is not the actual value that is currently used on the controller. Whenever a black parameter is modified it turns red, to remind the user that the controller needs to be restarted for changes to take effect. Red parameters turn black again the next time the controller system is restarted.

Continues on next page

Note

The color codes are only valid within TuneMaster. When TuneMaster is not running, green parameters are exactly the same as black parameters. Black parameters only become red if they are modified from TuneMaster. If a system parameter is modified outside of TuneMaster, from the FlexPendant or another program, the parameter will NOT become red in TuneMaster.

4.2 Using the Tune Estimate function

Information

TuneMaster can estimate the tuning of an additional axis. The Tune Estimate function performs a chirp excitation of the system. The result of the chirp excitation is analyzed in TuneMaster and suitable control parameter values are calculated. The Tune Estimate function will work for most axes, but it may also fail in some special cases. The most common reasons for the Tune Estimate function to fail are that the system is too stiff or too lightweight (resonance frequencies above 125 Hz). If the target system characteristics highly deviate from a two-mass system the Tune Estimate function may also fail.

Estimating the tuning of an axis

1. Select the joint you wish to tune in the joint selector tree view, located to the left in TuneMaster.
2. Select the **Tune Estimate** tab.
3. Click the **Start** button. The chirp movement will be performed by the axis.
4. If the estimated tuning calculation is successful, the recommended values for the controller parameters will be displayed to right of the plotter window.

Pressing the **Calc Params** or **Start** buttons only calculates and displays the values of the controller parameters, to actually update the controller configuration the user will have to manually enter the values in the **Cfg Info** tab. See *Using the Cfg Info tab on page 29*.

Adjusting the default chirp values

In some cases the default values will not give satisfactory excitation of the system. If the system shakes too much or too little, the amplitude parameter should be adjusted to compensate.

> **CAUTION**
>
> Do not exaggerate when increasing the amplitude, the axis should clearly vibrate but if the amplitude is increased to the point where the axis starts slamming the gearbox might be damaged.

Can the Tune Estimate values be trusted?

Pressing the **START** button causes two curves to be plotted.

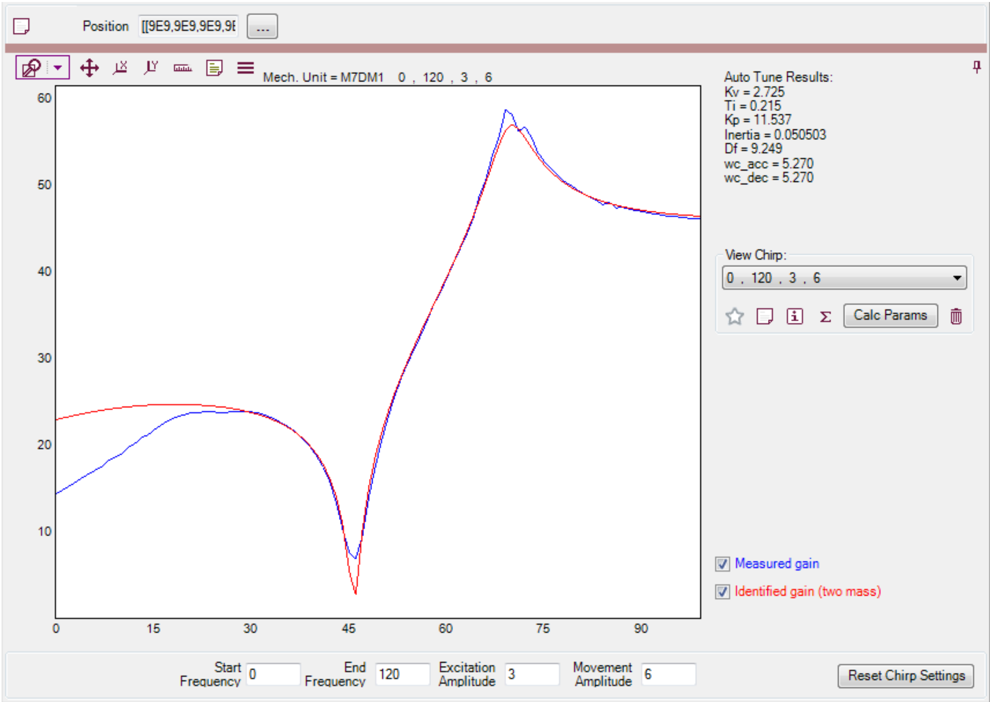
- Measured gain curve represents the measured transfer function of the system.
- Identified gain curve represents the transfer function of the approximated two mass model, used for the tuning calculations.

Comparing the two curves gives an indication of how good the approximation is, the more similar they are, the better the result.

Continues on next page

Reference graph

The graph below shows an acceptable result. The two curves are similar so the Tune Estimate results will be accurate.



xx1700001199

4.3 Measure noise

Information

The measure noise procedure is used to verify the noise level in the speed signal. A high noise level might be a source of bad control and accuracy of the mechanical unit. In some cases the noise levels also varies over one motor revolution caused by the mechanical and electrical construction. It is important to keep the overall (standard deviation) noise level low but also to verify that sudden noise pulses (peaks) are limited in number and magnitude.

Experiment and analysis procedure

Experiment

Only the motor speed (in rad/s) during the experiment can be set by the user. The motor speed should be set as low as possible to avoid too long transient periods during acceleration and deceleration. Each experiment performs a motion of two motor revolutions.

Analysis

There are two ways of analyzing data. First data can be analyzed versus time and second versus motor position (angle). The latter can be useful to identify if a noise pattern is correlated with the motor position or not. There are two limits, noise and peak, which can be modified by the user to analyze the noise level. The noise limit corresponds to the overall noise level while the peak value corresponds to noise pulses. As a rule of thumb the noise limit should be set to 3 rad/s and the peak limit to 5 rad/s. Beside the limit values a ratio of the samples exceeding the limits are computed. Both these values should be less than 10%.

Results

If the measured noise is much higher (>10%) then the noise limits (3 rad/s on average and 5 rad/s at peaks) the control of the mechanical unit may not work properly. The user can take some steps to try and correct a too high noise level. A high noise level can often result from problems with the resolver signal cables (the cable that connects the motor resolver to the SMB). The SMB can be located either on the mechanical unit (usually the case for robots) or in the controller cabinet (typical for most additional axes). Do not confuse the resolver signal cables with the digital signal cable between a standard ABB robot and the controller cabinet.

The following actions can reduce resolver noise:

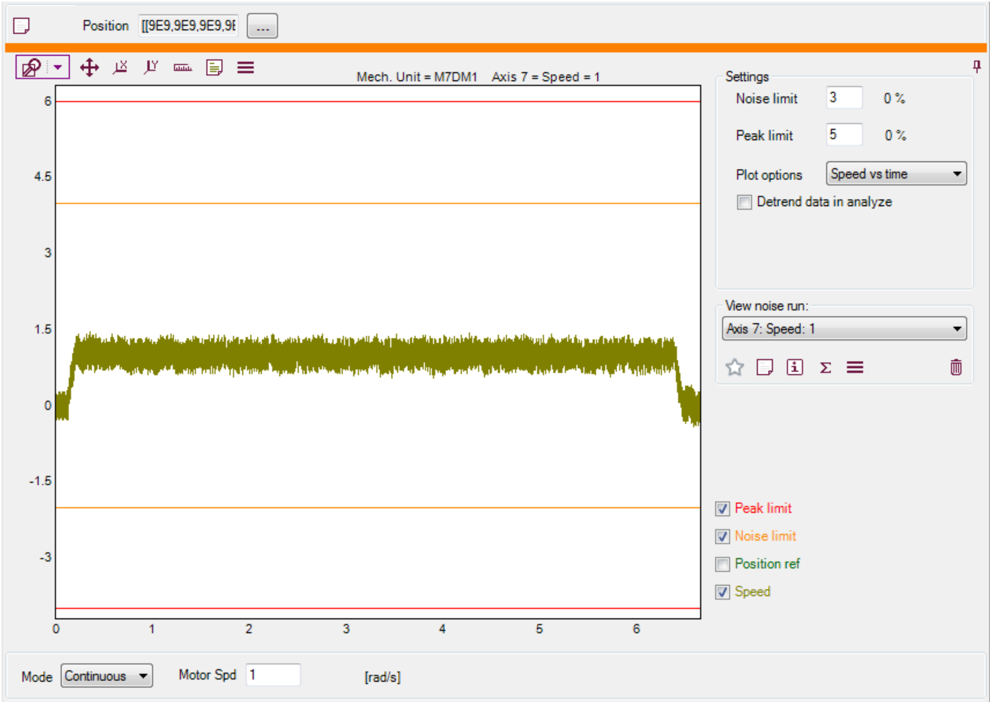
- Make sure that the signal cable is properly shielded, and that the shielding material is properly grounded.
- Make sure that the signal cable is separated from the power cable to the motor.
- Try to keep the signal cables as short as possible.
- When connecting the signal cable and the power cable to the motor, try to separate them by connecting the power cable from one side of the motor and signal cable from the other side.

Continues on next page

If none of the above helps it is possible that the resolver itself is faulty, the motor should be changed. Another possibility is a malfunctioning SMB.

Reference graph

The graph below shows an acceptable result. The unit has a very low noise characteristic.



xx1700001200

4.4 Estimate friction

Information

The estimate friction procedure is used to identify parameters in a static model by means of a least-squares method for a single axes.

The parameters are:

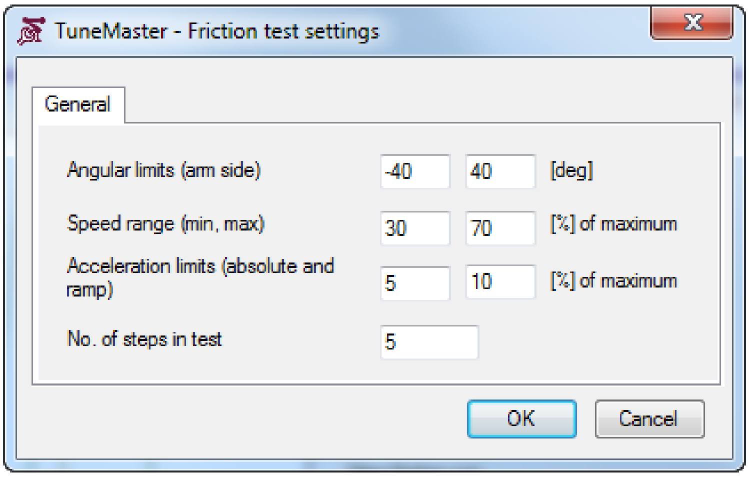
- *J* - Total inertia of motor, gear box and arm [kg m^2].
- *Fc* - Total coulomb friction for motor, gear box and arm [N].
- *Fv* - Total viscous friction for motor, gear box and arm [N].
- *Theta* - Angle between center of rotation and center of gravity [deg].

The values of the estimated parameters are shown in the **Result** box.

Measure and analysis procedure

Experiment

Each experiment performs a motion between two axes positions and the parameter steps determines the number of movements between the positions. If the button **Settings** is pressed, a window with parameters for the experiment opens.



xx1700001201

- **Angular limits** (**Distance limits**) - sets the two positions determining the motion. For rotational axes the parameters are given in degrees on the arm side and for linear axes in mm.
- **Velocity range** - Determines the speed range covered by the experiment in percent of maximum speed. For example, if the parameters are 50 and 60 and the parameter steps are set to 3 then the speed 50, 55 and 60% of the maximum speed will be applied, one for each step.
- **Acceleration limits** - sets the acceleration and ramp of acceleration for the experiment in percent of maximum acceleration and ramp. To avoid flexible modes both parameters should be kept as low as possible (< 20%).
- **No of steps in test** - Determines the number of motions between the positions given in **Angular limits** (**Distance limits**). The accuracy of the estimated parameters increases with the number of steps, at least five steps are

Continues on next page

recommended. The number of parameter steps can be set in both the setting window and the main window.

There are three different modes to choose from when performing the experiment.

- *Auto* - This is the default alternative and performs two identical experiments. In the first, the model parameters are estimated. In the second, the estimated parameters and new measured data (speed and position) is used to compute a simulated torque signal. The simulated torque signal is then compared with the measured torque signal.
- *Estimate* - Performs one experiment and estimates model parameters (first part of auto mode).
- *Validate* - Performs one experiment and uses the model parameters in the **Result** box along with data in the experiment to compute a simulated torque which is compared with measured torque (second part of auto mode). Note that a validation experiment can not be performed if the **Result** box is empty, that is, an auto or estimate experiment must be performed prior to the validate experiment.

Analysis

The most important part of the analysis is to see if the simulated **Torque sim** and measured **Torque ref** is similar. It is also important to compare the values in the **Result** box with physical knowledge to see if the value is probable or not.

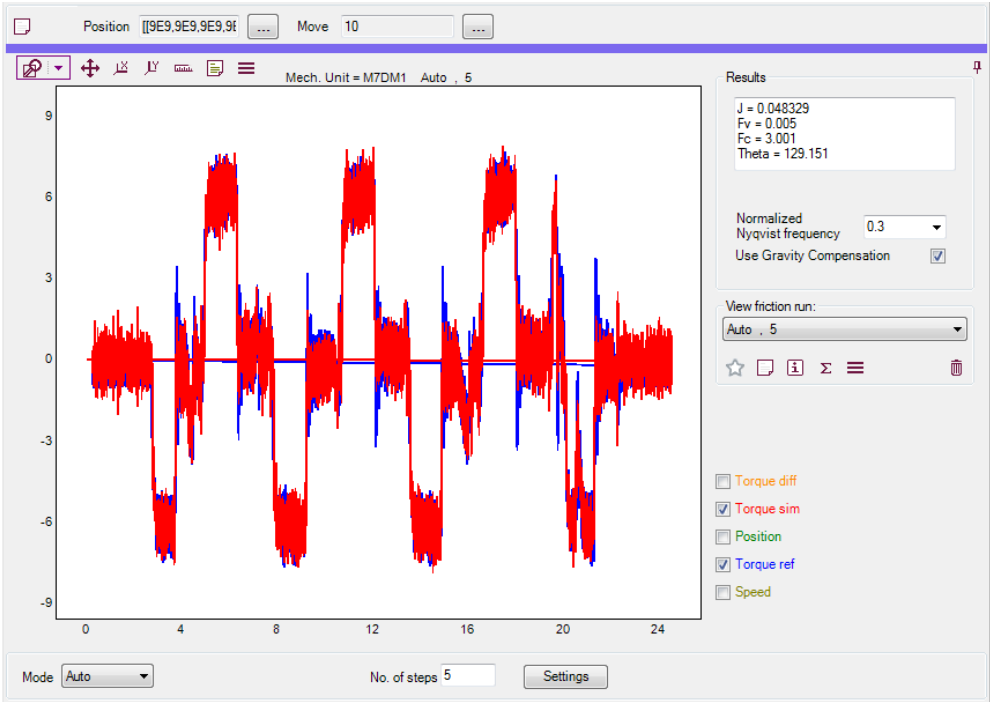
In some cases the match between the simulated and measured torque might be poor, and it can be necessary to perform the following actions (listed in priority order):

1. Increase the number of steps in the experiment. This yields more accuracy in the parameter estimation.
2. Change the low pass filter coefficient, **Normalized Nyqvist frequency**, in the **Result** box (a new experiment is not required). For some measuring devices the SNR (signal to noise ratio) is too bad and filtering is required before estimation of parameters. A proper filter coefficient is chosen when the noise level is similar for the simulated and measure torque.
3. Increase size of the experiment movement, that is, change **Angular limits** (**Distance limits**) in the settings window.
4. Change **Velocity range** in the **Settings** window.
5. Change **Acceleration limits** in the settings window. If it is assumed that flexible modes of the mechanical units are applied the limits should be decreased in other cases the limits should be increased.

Continues on next page

Reference graph

The graph below shows an acceptable result. The **Torque** and **Torque (sim)** curves are similar so the *Friction estimation* results will be accurate. Note that it was necessary to decrease the **Normalized Nyqvist frequency** to 0.3 to obtain an acceptable result in this case.



xx1700001202

4.5 Tuning LCM0 parameters

4.5.1 Tuning order for LCM0 parameters

Overview

The Lag Control Master 0, *LCM0*, can be used in three different modes, depending on the setting of the *ffw_mode* parameter. The options are:

- *FFW Mode = No (0)*, the simplest configuration.
- *FFW Mode = Spd (1)*, the controller receives information about the desired speed of the axis. As a result, the position lag is greatly reduced compared to the *No* mode.
- *FFW Mode = Trq (2)*, the controller uses the desired speed and acceleration of the axis to calculate the desired motor torque. This option normally gives the best performance.

The user must choose which type to use. Depending on the choice of *ffw_mode* different parameters need to be tuned. When working with TuneMaster it is recommended to use *LCM0* with *FFW Mode = Trq*, since this gives the best performance and should normally be only marginally harder to tune then the other modes.

Tuning order for *FFW Mode = No (0)*

Use the following tune order for *FFW Mode = No (0)*:

1. Tune the PID parameters:
   - Tune *Kv*, see *Tuning Kv on page 41*.
   - Tune *Ti*, see *Tuning Ti on page 43*. Make sure that the *Kv* property is set to the value obtained when tuning *Kv*.
   - Tune *Kp* with *ffw_mode = 0*, see *Tuning Kp on page 45*. Make sure that the *Kv* and *Ti* properties are set to the values obtained when tuning *Kv* and *Ti*.
2. Tune *Acc* and *Dec*:
   - Tune *Acc*, see *Tuning Acc on page 57*.
   - Tune *Dec*, see *Tuning Dec on page 60*.
3. If neccessary, *wc_dacc_ratio* and *wc_ddec_ratio* should be tuned to get a softer, but slower, movement. If the movement looks good after tuning *Acc* and *Dec*, the ratio parameters can be left at their default value 1.
4. After tuning *Acc* and *Dec*, make sure that there is no overshoot in the position signal. If there still is an overshoot it may be necessary to retune *Kp*. Another way to reduce overshoot is too reduce the *dacc_ratio* and *ddec_ratio* parameters.

Continues on next page

Tuning order for *FFW Mode = Spd (1)*

Use the following tune order for *FFW Mode = Spd (1)*:

1 Tune the PID parameters:
   - Tune *Kv*, see *Tuning Kv on page 41*.
   - Tune *Ti*, see *Tuning Ti on page 43*. Make sure that the *Kv* property is set to the value obtained when tuning *Kv*.
   - Tune *Kp* with *ffw_mode = 0*, see *Tuning Kp on page 45*. Make sure that the *Kv* and *Ti* properties are set to the values obtained when tuning *Kv* and *Ti*.
   - If there is an overshoot when tuning *Kp* even for low values, that is *Kp<10*, the bandwidth can be lowered to reduce the overshoot.
   - In rare cases the speed may lag behind its reference and the *delay_time* should be adjusted.

2 Tune *Bandwidth*, see *Tuning Bandwidth on page 55*.

3 Tune *Acc* and *Dec*:
   - Tune *Acc*, see *Tuning Acc on page 57*.
   - Tune *Dec*, see *Tuning Dec on page 60*.

4 If neccessary, *wc_dacc_ratio* and *wc_ddec_ratio* should be tuned to get a softer, but slower, movement. If the movement looks good after tuning *Acc* and *Dec*, the ratio parameters can be left at their default value 1.

5 After tuning *Acc* and *Dec*, make sure that there is no overshoot in the position signal. If there still is an overshoot it may be necessary to retune *Kp*. Another way to reduce overshoot is to reduce the *dacc_ratio* and *ddec_ratio* parameters.

Continues on next page

Tuning order for *FFW Mode = Trq (2)*

Use the following tune order for *FFW Mode = Trq (2)*:

1. Tune the PID parameters:
   - Tune *Kv*, see *Tuning Kv on page 41*.
   - Tune *Ti*, see *Tuning Ti on page 43*. Make sure that the *Kv* property is set to the value obtained when tuning *Kv*.
   - Tune *Kp* with *ffw_mode = 0*, see *Tuning Kp on page 45*. Make sure that the *Kv* and *Ti* properties are set to the values obtained when tuning *Kv* and *Ti*.
   - If there is an overshoot when tuning *Kp* even for low values, that is *Kp<10*, the bandwidth can be lowered to reduce the overshoot.
   - In rare cases the speed may lag behind its reference and the *delay_time* should be adjusted.
2. Tune the torque parameters:
   - Tune *Inertia*, see *Tuning Inertia on page 49*.
   - Tune *Df*, see *Tuning Df on page 52*. Make sure the value obtained when tuning *Inertia* is used when tuning *Df*.
3. Tune *Bandwidth*, see *Tuning Bandwidth on page 55*.
4. Tune *Acc* and *Dec*:
   - Tune *Acc*, see *Tuning Acc on page 57*.
   - Tune *Dec*, see *Tuning Dec on page 60*.
5. If neccessary, *wc_dacc_ratio* and *wc_ddec_ratio* should be tuned to get a softer, but slower, movement. If the movement looks good after tuning *Acc* and *Dec*, the ratio parameters can be left at their default value 1.
6. After tuning *Acc* and *Dec*, make sure that there is no overshoot in the position signal. If there still is an overshoot it may be necessary to retune *Kp*. Another way to reduce overshoot is to reduce the *dacc_ratio* and *ddec_ratio* parameters.

   When used in the *Trq* mode, there can be a small steady state error in the position signal, were the position comes to rest at a somewhat higher value then the position reference. This effect is normally very small and should not be confused with an overshoot. The steady state error can not be reduced and should be ignored.

4.5.2 Tuning Kv

Information

Kv, gain speed loop, the amplification of the velocity control. A high value gives better high frequency stiffness, better response speed and low overshoot. If the value is too high the axis will vibrate. *Kv* controls the amount of damping for the axis and is the most limiting of the parameters. A poor value of *Kv* will limit *Kp* and *Ti*, and the axis will not be fully utilized.

Tuning procedure

Start by pressing the **Generate start guess** button to let TuneMaster calculate an initial guess for *Kv*.

The start guess generated by TuneMaster is based on the motor inertia for the axis. If the motor inertia is not entered in the configuration, the start guess will be 0 and the user must provide an initial value. For a small motor use 1e-4, for a larger motor 0.1 is a reasonable start guess. Note that setting *Kv* to an extremely low value will not harm the system, the axis simply wont move. Setting *Kv* to a value that is several hundred times too high can cause heavy oscillation, and should not be done repeatedly.

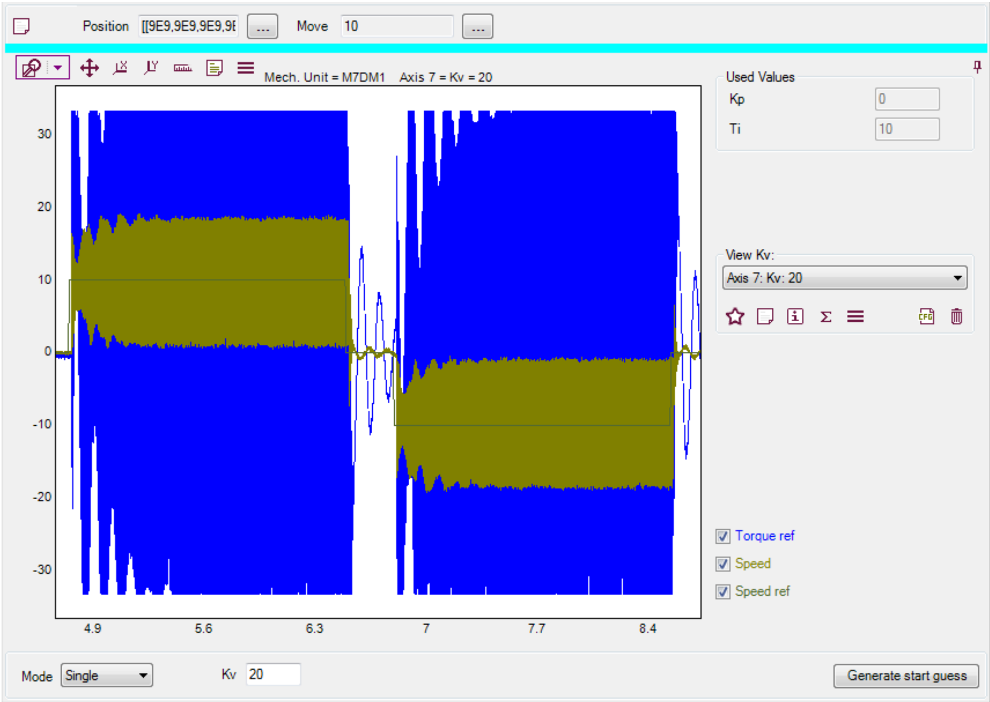
Increase the tune value for *Kv* in steps of 5% and observe *Torque ref* and *Speed*. Stop when the axis reaches instability. When the axis reaches instability it will vibrate, the signals for *Speed* and *Torque ref* will also jump rapidly back and forth between the max/min allowed value. Divide the tune value by 2.5 and run the axis again, while observing *Torque ref*. There should be at most one or two damped oscillations after the acceleration stage. If *Torque ref* oscillates more than this, then decrease its value somewhat.

Kv is a critical parameter. A large value will result in a stiff axis and a fast response. If *Kv* is too small *Kp* will also be limited, resulting in an under-utilized axis. When tuning *Kv*, the values for *Ti* and *Kp* are set to *Ti = 10* and *Kp = 0*. This is done automatically by TuneMaster and is reflected by the values in the **Ti** and **Kp** text boxes.

Continues on next page

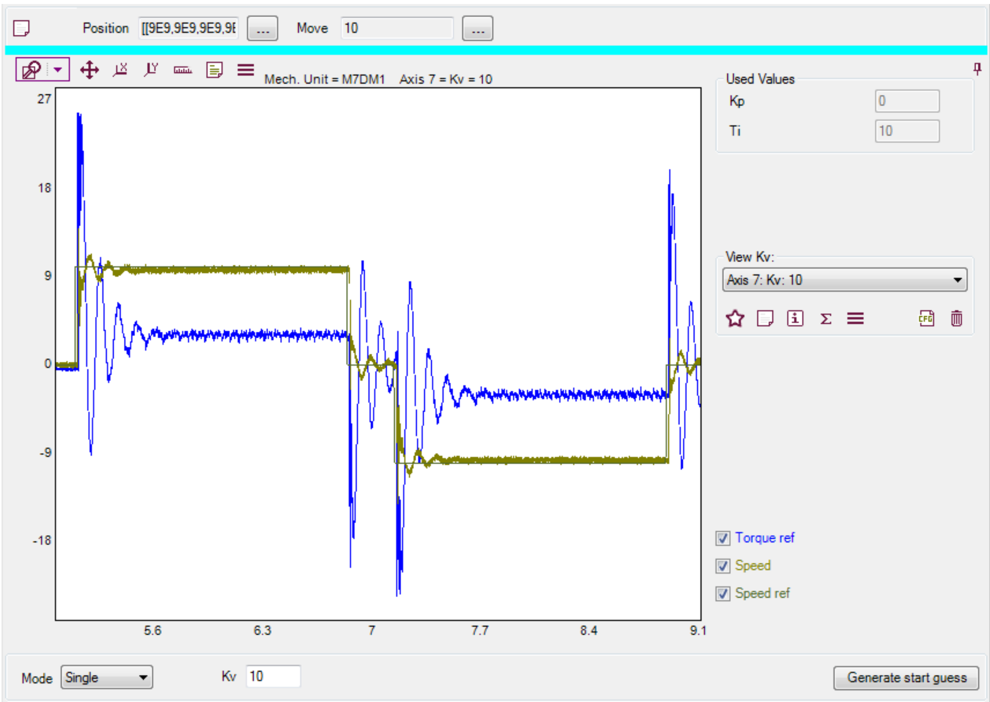
Reference graphs

Unstable *Kv* - The graph below shows the speed and torque signals for an unstable *Kv* value.



xx1700001203

Correct *Kv* - The graph below shows the plots for a suitable *Kv* value.



xx1700001204

4.5.3 Tuning Ti

Information

Ti, integration time speed loop, the integration interval constant of the velocity control. A typical value for Ti is 0.2. A low value gives low steady state error and better path following. If *Ti* is too small the axis will overshoot and the response will be oscillatory.

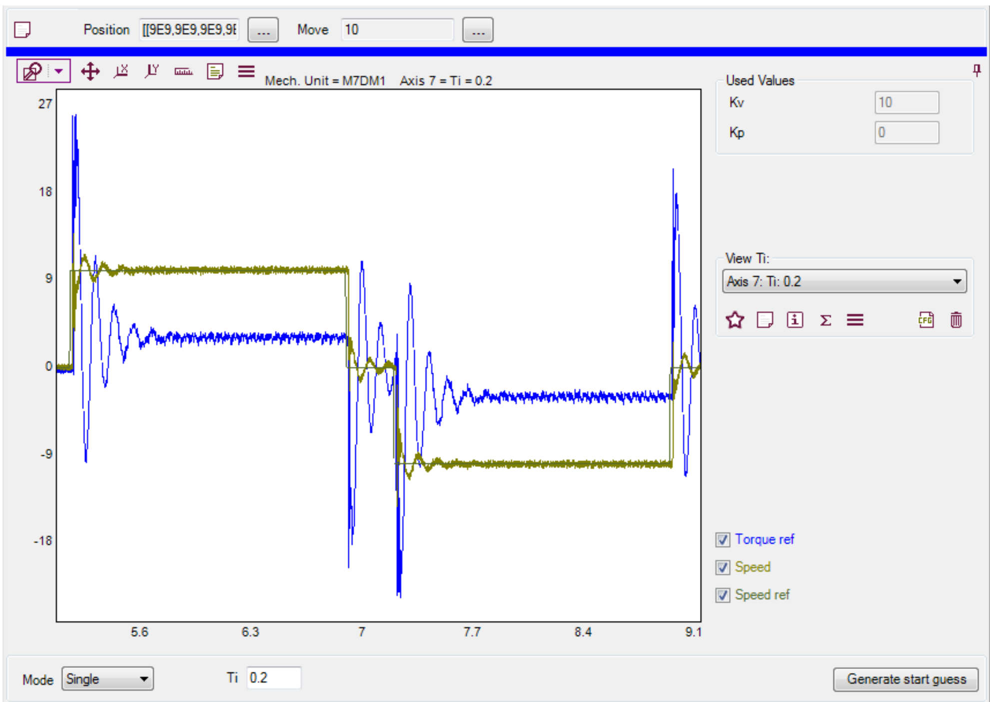
Tuning procedure

Before tuning *Ti*, make sure that the value previously obtained for *Kv* is entered in the **Kv** text box. The *Kv* value should be set to the actual obtained *Kv*, that is *Unstable Kv/2.5*. Start by pressing the **Generate start guess** button to let TuneMaster generate a suitable initial tune value for *Ti*.

Adjust the tune value for *Ti* downwards in steps of 10% until the effect can be seen on the chart recordings of speed as an increased overshoot. Increase the tuning factor by 5-10% to remove the effect. When searching for an increased overshoot it can be a good idea to compare the overshoot for a given tune value of *Ti* to the overshoot obtained when *Ti* = 10 (a tip is to use the **Kv** tab with the correct *Kv* value selected, since *Ti* is set to 10 when tuning *Kv*). The final *Ti* value should increase the the first peak overshoot by roughly 5-10% compared to *Ti* = 10.

Reference graphs

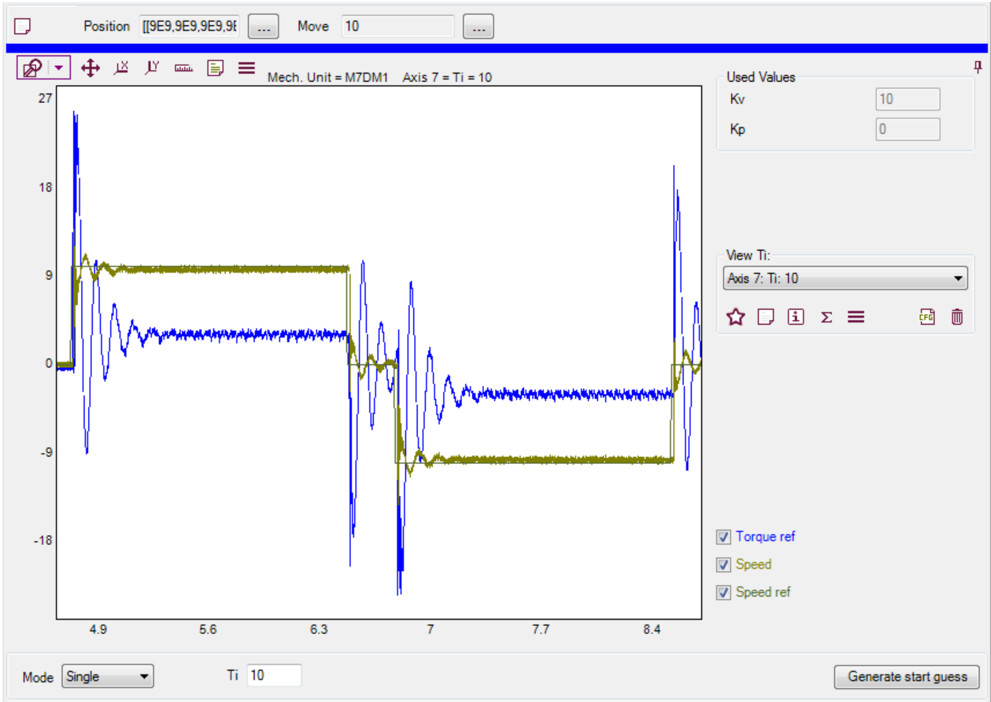
Correct *Ti* - The graph below shows the speed and torque signals for a correct value of *Ti*.



xx1700001205

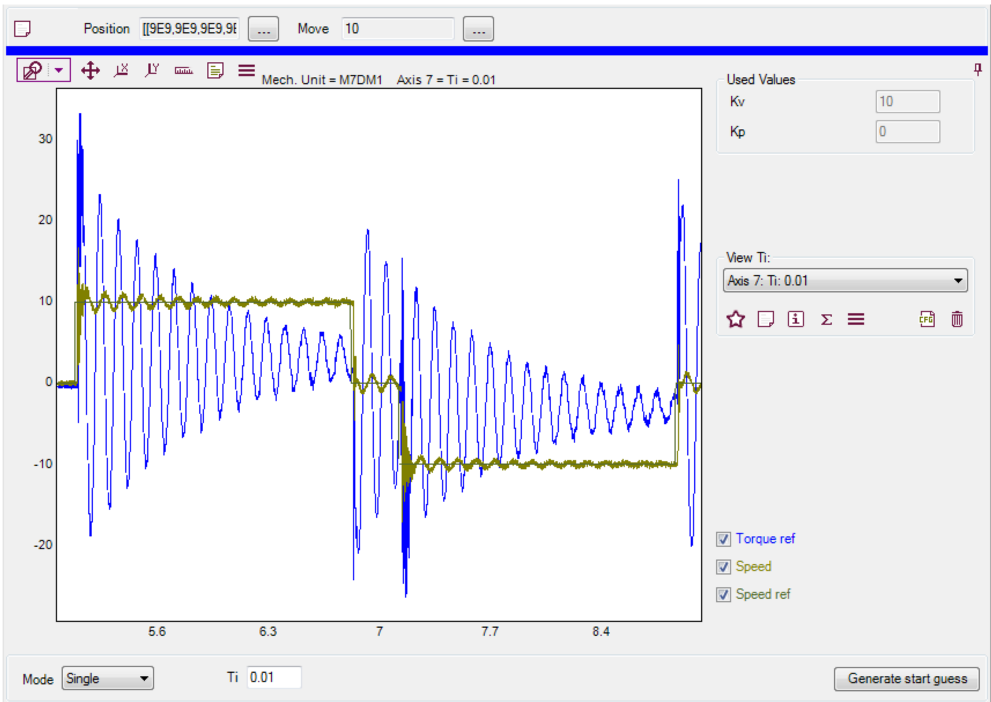
Continues on next page

Too high *Ti* - If the value of *Ti* is too high, there will be no integration effect.



xx1700001206

Too low *Ti*. If the value of *Ti* is decreased too much, the system will become oscillatory.



xx1700001207

4.5.4 Tuning Kp

Information

Kp, gain position loop, the amplification of the position control, for example 15. A high value will give a stiff axis that quickly assumes its new position. The value should be large without inducing overshoot in the position or oscillations of the axis.

Tuning procedure

Before tuning *Kp*, make sure that the **Kv** and **Ti** properties (textboxes) are set to their previously obtained values.

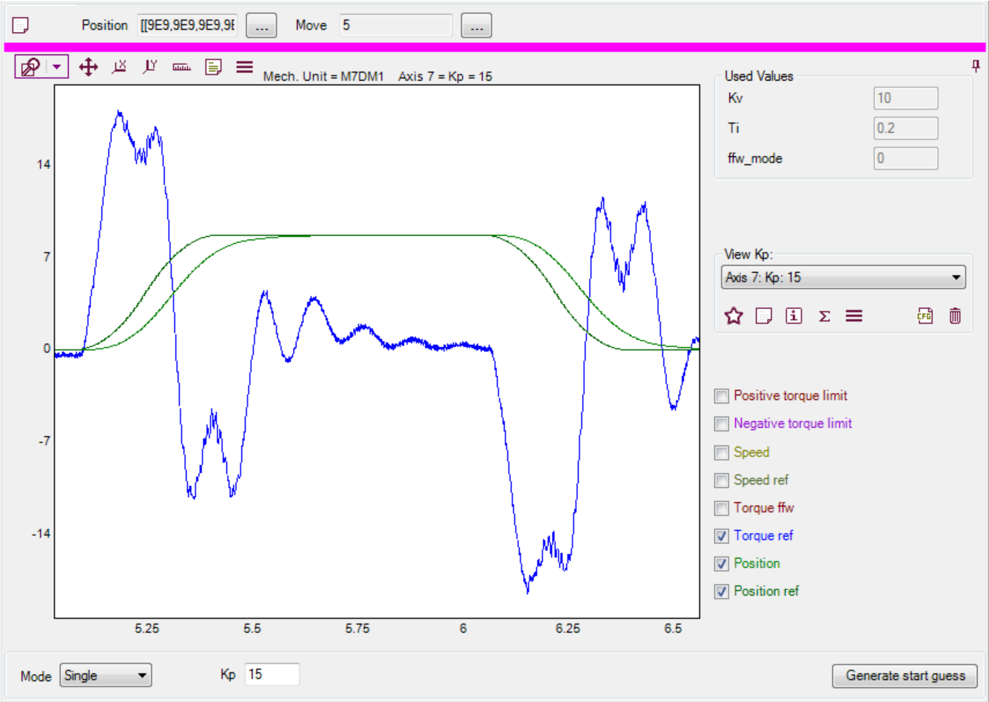
Start by letting TuneMaster generate a starting guess, by pressing the **Generate start guess** button. Execute a tune movement. Observe the position ref and position signal, increase *Kp* until there is an overshoot in the position signal or the *Torque ref* starts to vibrate. Typically the overshoot will occur before vibration in *Torque ref*, but if a large *Kp* is needed for overshoot *(Kp > 30)* then the *Torque ref* may become unstable before the overshoot is reached. For some axes, typically lightweight and stiff, it might be hard to find a value for *Kp* that gives an overshoot, in some cases it is even impossible, regardless of what value for *Kp* is used. In these cases, try to observe how well the position follows the position ref. As long as the gap decreases it is justified to keep increasing *Kp*, increase *Kp* until no more substantial improvement can be observed.

The position error (lag) is inversely proportional to *Kp*. Thus a large value for *Kp* is desirable.

Continues on next page

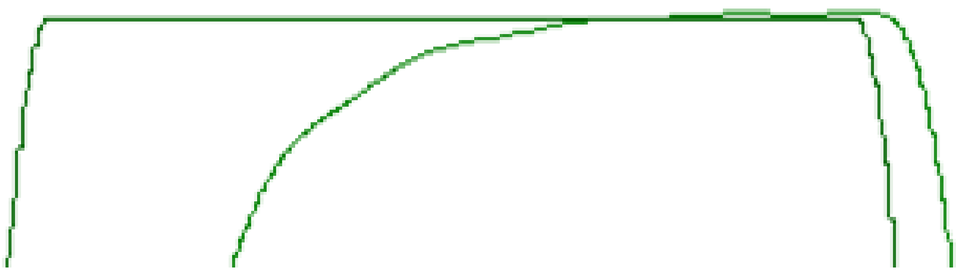
Reference graphs

Correct *Kp* - The graph below shows the position and torque signals for a correct value of *Kp*. The close-up of the top region shows that the *Position* signal is close to the *Position ref* but does not overshoot.



xx1700001210

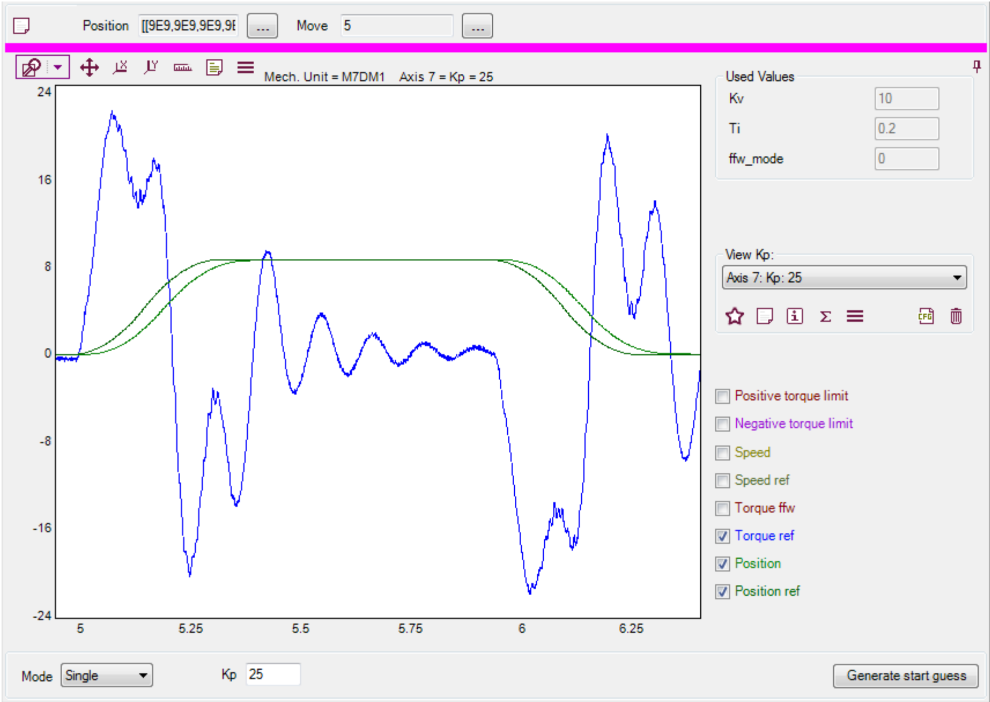
Close-up of top region.



xx1700001211

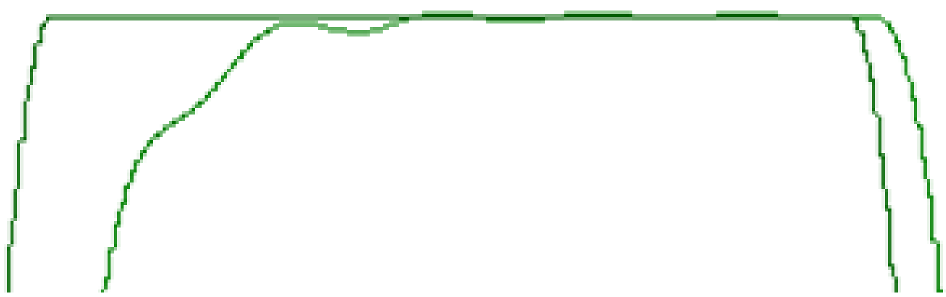
Continues on next page

Too high *Kp* - The graph below shows the position and torque signals for a "too high" value of *Kp*. It is hard to see in the first plot but the close-up of the top region clearly shows that there is an overshoot in the *Position* signal.



xx1700001208

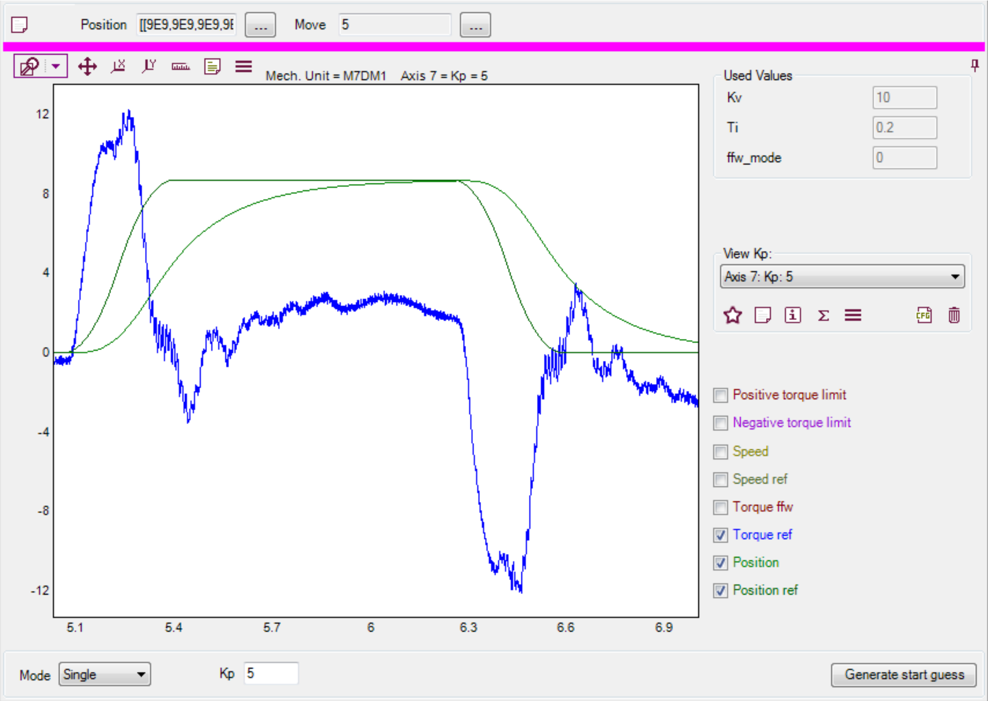
Close-up of top region.



xx1700001209

Continues on next page

Too low *Kp* - The graph below shows the position and torque signals for a "too low" value of Kp. It is clear from the close-up that it takes significant time for the *Position* signal to reach the target *Position ref* signal.



xx1700001212

Close-up of top region.

xx1700001213

4.5.5 Tuning Inertia

Information

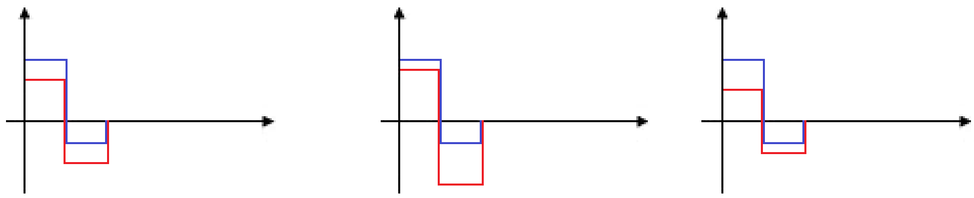
To obtain faster performance in an additional axis it is possible to use the torque feed forward mode (Trq) of *Lag Control Master 0*. The inertia must be tuned if the feed forward mode of LCM0 is set to torque, *ffw_mode=2*. In the tune inertia tab the feed forward mode is automatically set to torque. The user supplies values for inertia, which represents the total mass moment of inertia at motor side.

Tuning procedure

Start by choosing a low value for the inertia, typically 0.001. Adjust the inertia upwards or downwards in rather large steps, multiply or divide by 10. Use smaller steps once the *Torque ffw* signal and the *Torque ref* signals start to look similar. The tuning rule is that *Torque ref* should be as much higher than *Torque ffw* during acceleration as it is lower during deceleration. There are two different cases of interest:

- Axis without gravity - Only friction should differ between *Torque ref* and *Torque ffw*. See figure below.
- Axis with gravity - Difficult. Only friction should differ between (*Torque ref - Gravity torque*) and *Torque ffw*. *Gravity torque* will have to be estimated.

It can be helpful to set the `wc_acc` and `wc_dec` parameters for the joint to a low level, so that the torque plateaus during acceleration and deceleration are easier to see.



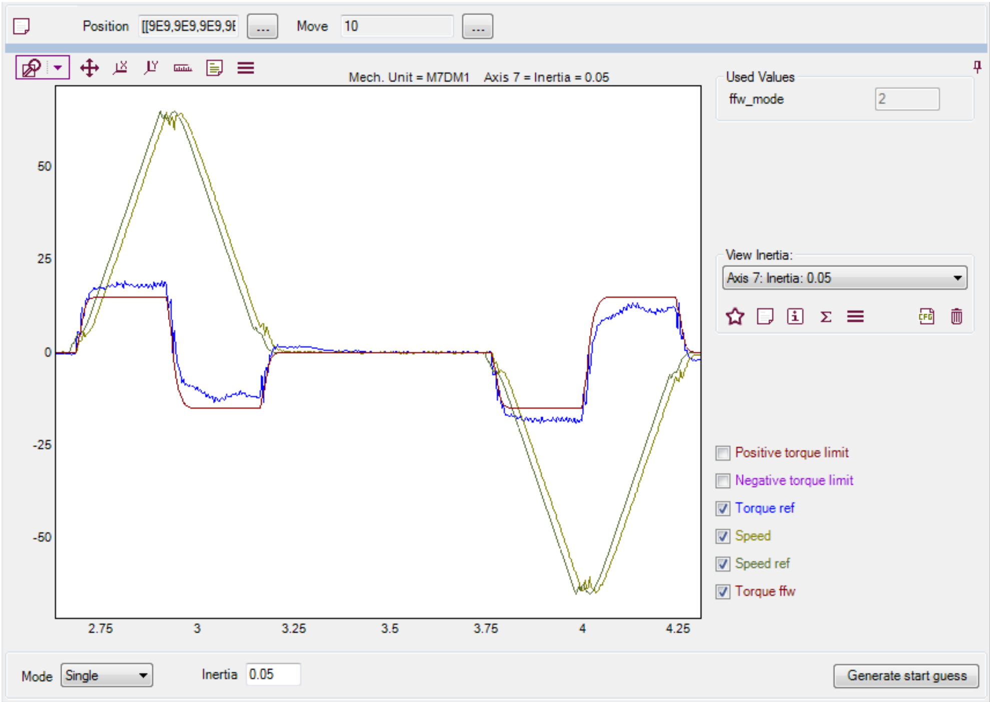
xx1700001214

Correct	Wrong - decrease *Inertia*.	Wrong - increase *Inertia*.

Continues on next page

Reference graphs

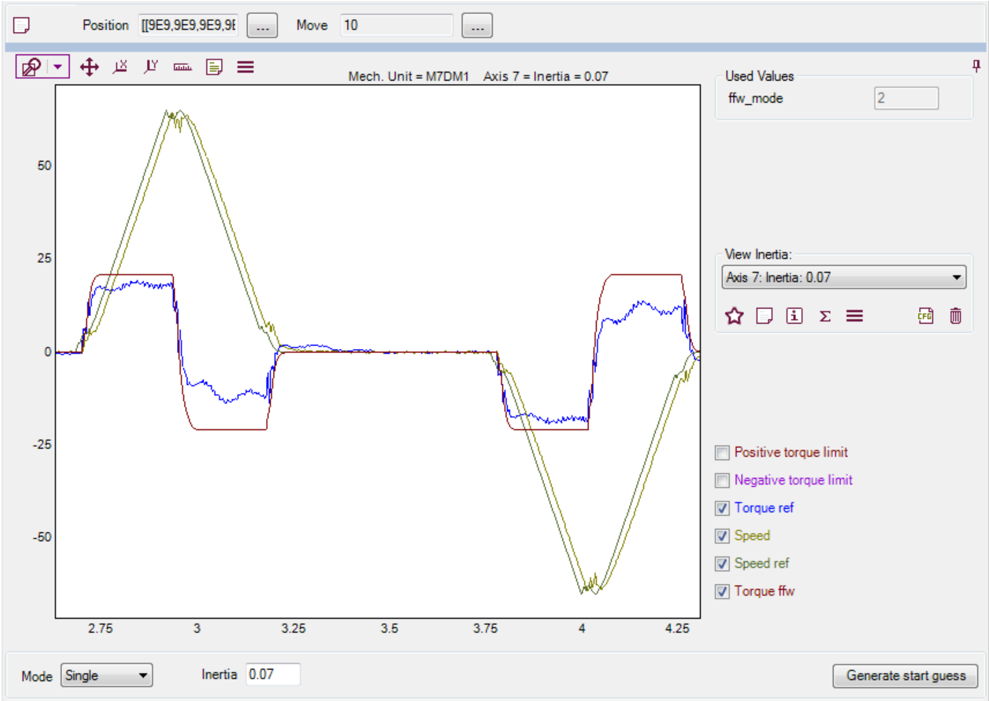
Correct *Inertia* - The graph below shows that the magnitude of the torque feed forward *Torque ffw* is as much lower than the actual *Torque ref* during acceleration as it is higher during deceleration.



xx1700001216

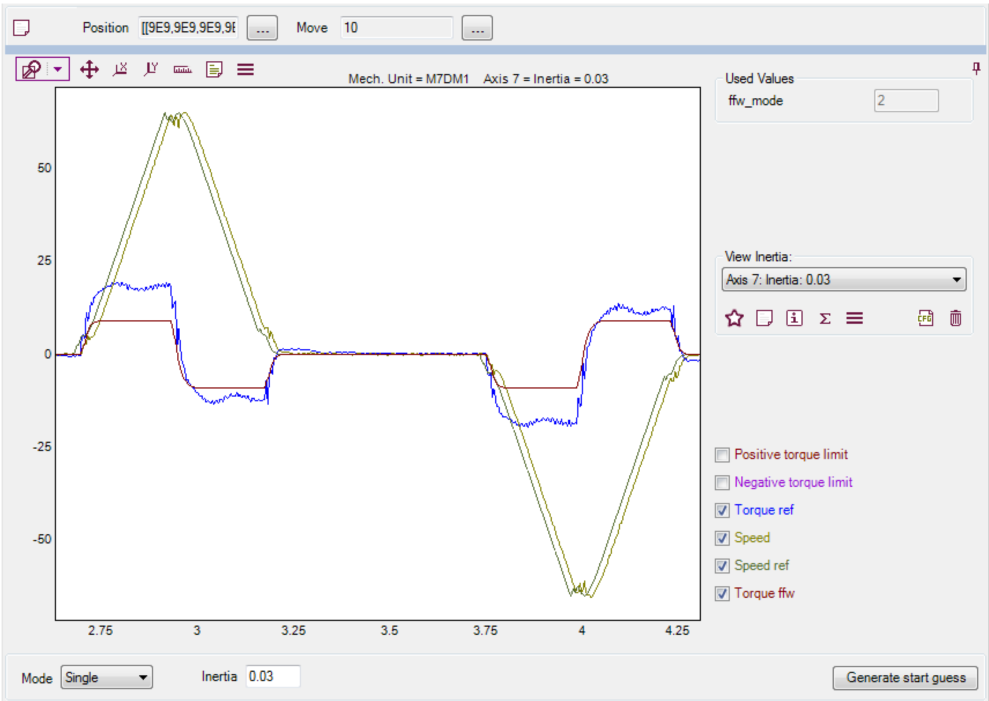
Continues on next page

Too high *Inertia* - The magnitude of the torque feed forward is greater then the actual torque, both during acceleration and deceleration.



xx1700001217

Too low *Inertia* - The magnitude of the torque feed forward is lower then the actual torque, both during acceleration and deceleration.



xx1700001218

4.5.6 Tuning Df

Information

The dynamic factor (Df) can be used to damp oscillations of the axis due to mechanical resonance. This parameter is only available if the *FFW Mode* is set to *Trq*. For most applications it is not necessary to adjust *Df*. If however, the *Torque ref* signal is oscillatory due to mechanical resonance, then the performance of the axis can be improved by adjusting *Df*. The *Df* parameter corresponds to the `resonance_frequency` parameter under **LCM0** in the **Cfg Info** tab.

Tuning procedure

Start by setting *Df* to 100 and performing one measurement at the **Tune Df** tab. To obtain a starting guess for *Df*, measure the time difference between two adjacent resonance peaks on the plot of the *Torque ref* signal. A rough estimate can be obtained by dividing one by the measured time difference.

Df = 1 / (T peak to peak)

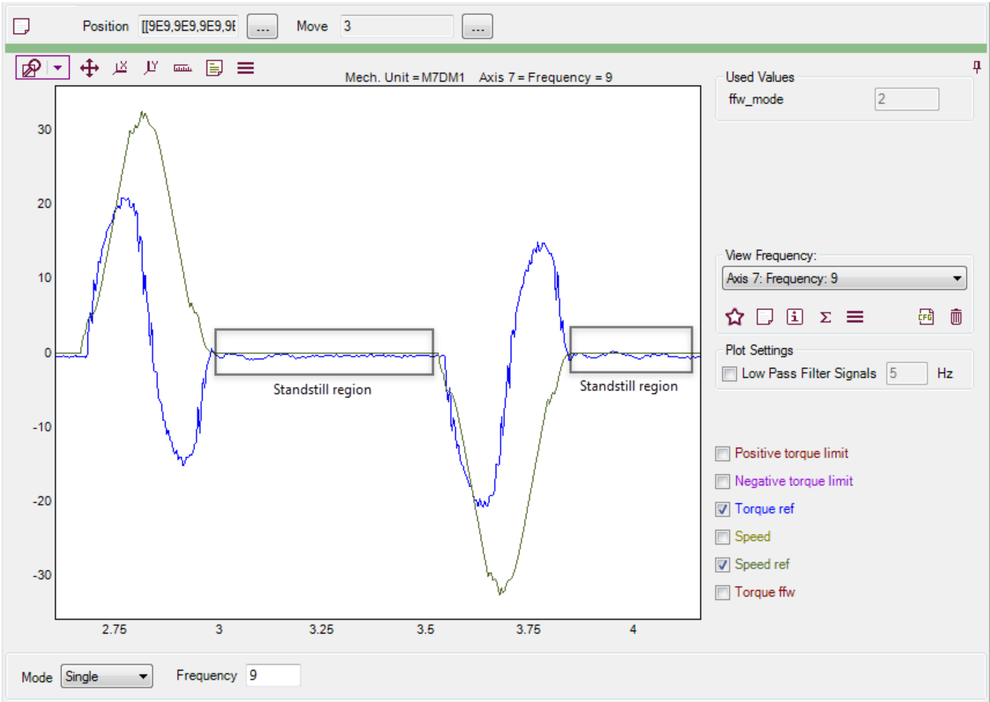
Df should be in the range 3 to 25. To tune *Df*, try values close to the start guess, both above and below. Choose the value that gives the least oscillation on the *Torque ref* signal. When judging how much oscillation there is in a particular graph it is important to zoom in and look at the interesting region, which is the part of the graph where the axis should be standing still. It is only this part that is relevant to look at, since the torque will oscillate during movement to make the axis reach a standstill faster. The standstill region is the part of the graph where the *Speed ref* signal has a value of 0. When tuning Df it is also easier to see the effects if a distinct movement is used, it may be helpful to set the `wc_acc`, `wc_dec`, `wc_dacc_ratio` and `wc_ddec_ratio` to high values during *Df* tuning. Especially the ratios should be set to 1 since they smoothen the movement when set to lower values.

The effect of tuning *Df* can vary allot for different mechanical units, depending on there stiffness, weight etc. Sometimes a very clear improvement can be obtained while in other cases the effect can be hardly noticeable even with optimal tuning. If there is no considerable oscillation to begin with it is recommended to leave *Df* at its initial value *Df* = 100.

Continues on next page

Reference graphs

Correct *Df* - The *Torque ref* signal reaches a steady state without much oscillation in the important standstill region.

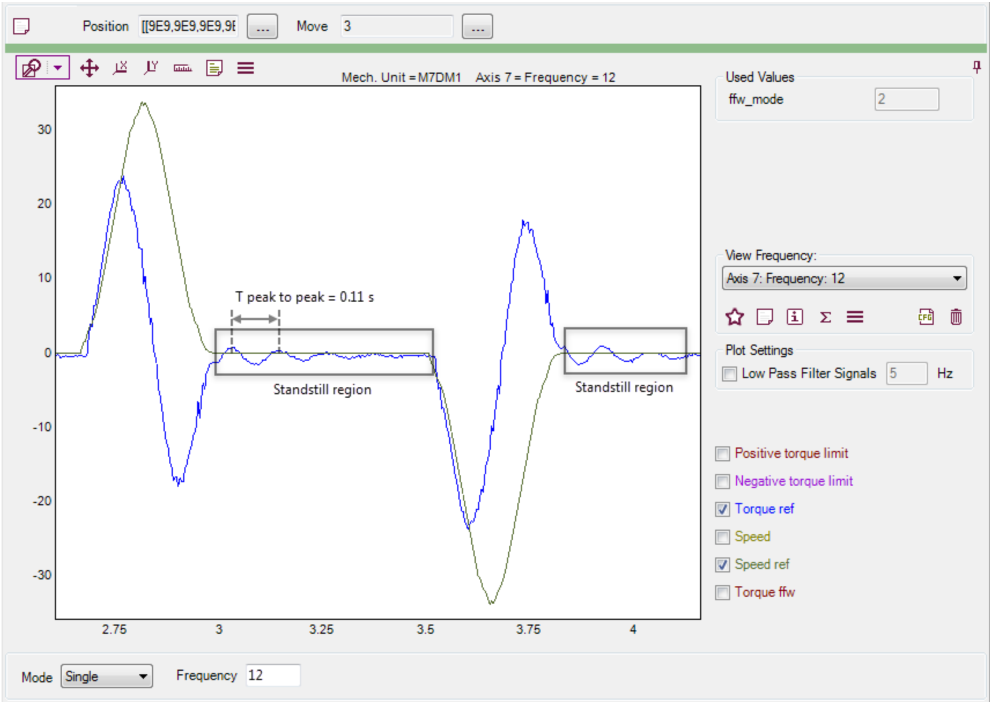


xx1700001219

Too high *Df* - Setting *Df* to a value above the optimal value gives a fairly clear oscillation on the *Torque ref* signal at the beginning of the standstill region. The

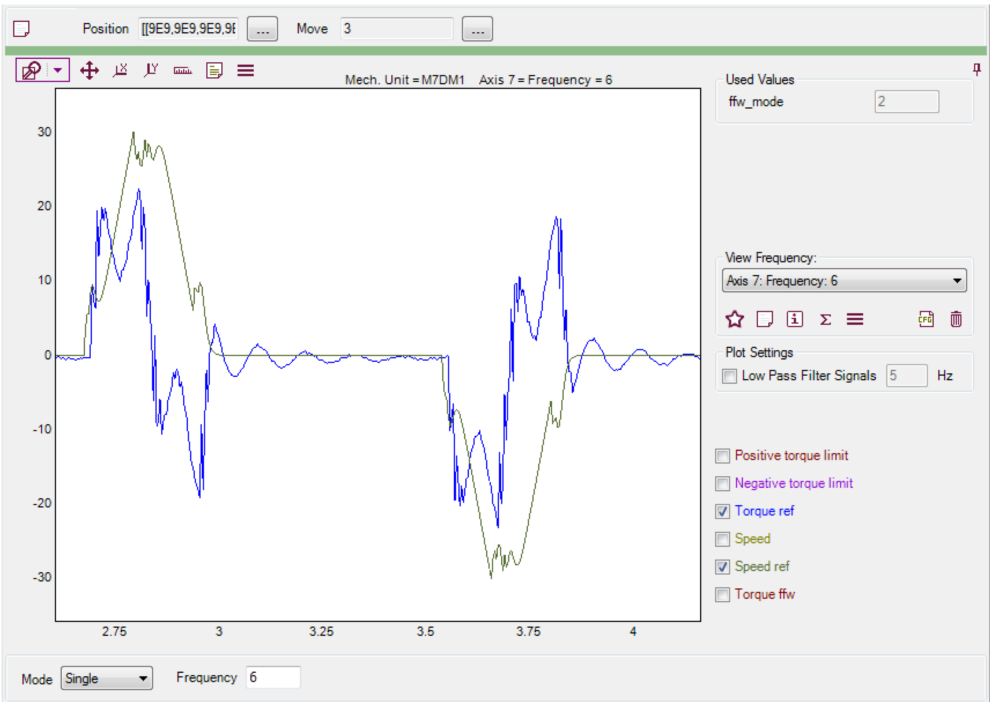
Continues on next page

peak to peak time of 0.11 s gives a starting guess of 1 / 0.11 s = 9.1 Hz, which is reasonably close to the optimal value achieved above.



xx1700001221

Too low *Df* - Setting *Df* to a value below the optimal value will cause the signals to be "choppy". A "klonk klonk" sound may also be heard during the movement. In general it preferable to choose a higher value if there is any uncertainty.



xx1700001223

4.5.7 Tuning Bandwidth

Information

The bandwidth parameter is only used for *FFW Mode = Spd (1)* or *Trq (2)*. A high bandwidth will result in faster control but increases the risk of vibrations and overshoot.

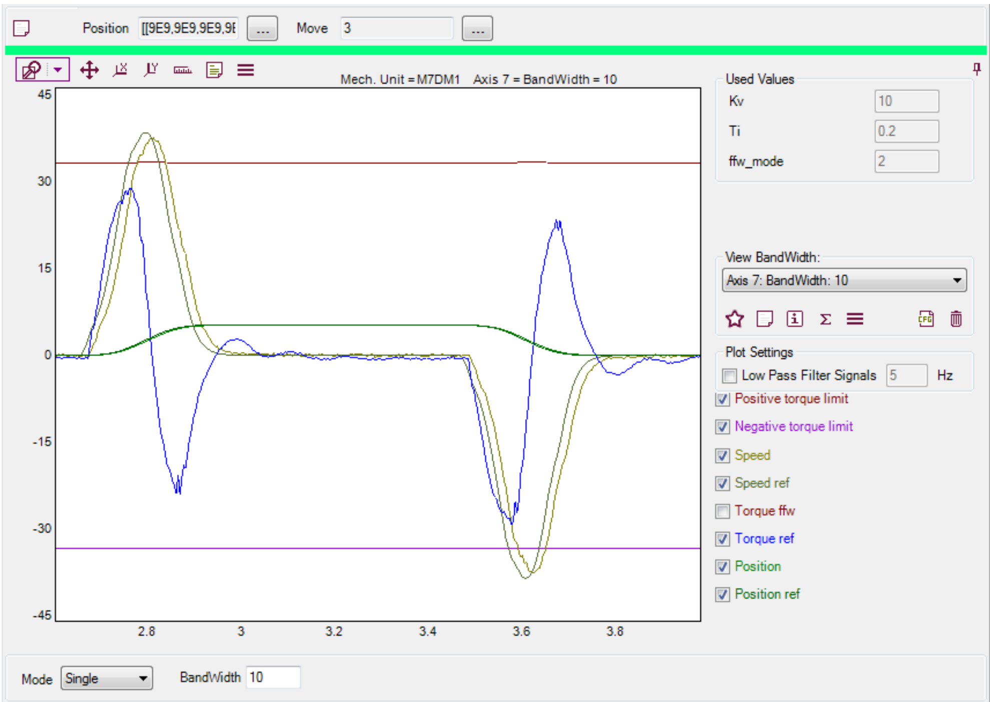
Tuning procedure

For *FFW Mode = Spd (1)*: Initially use bandwidth 25. Verify that there are no overshoots by using test signals and by visually observing the axis. Reduce the bandwidth if necessary.

For *FFW Mode = Trq (2)*: If the dynamic factor *Df* is less than 10, use bandwidth 2.5**Df*. Otherwise use default bandwidth 25. Verify that there are no overshoots by using test signals and by visually observing the axis. Reduce the bandwidth if necessary.

Reference graphs

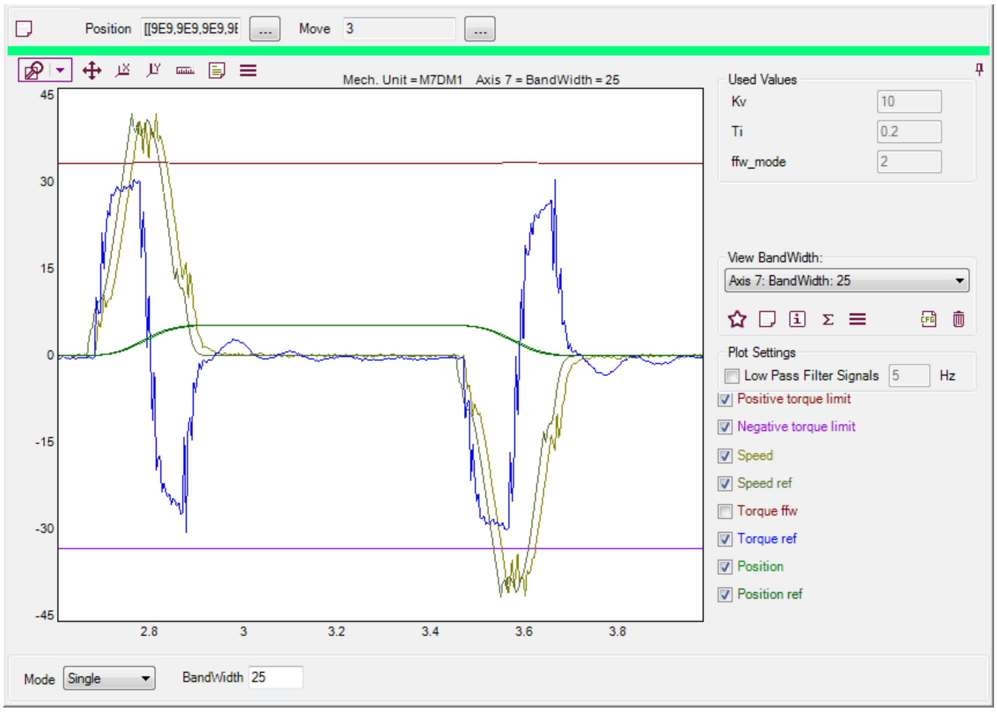
Correct *Bandwidth* – The graph below shows the test signals for a suitable value of the bandwidth.



xx1800000049

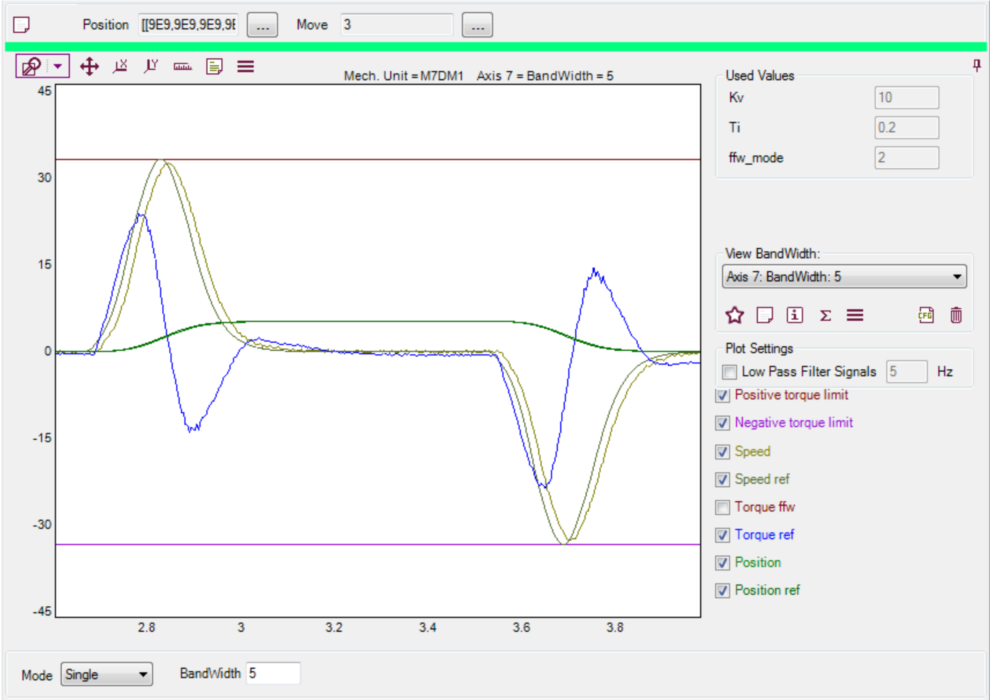
Continues on next page

Too high *Bandwidth* – The graph below shows the test signals for a too high value of the bandwidth, resulting in some vibrations of the axis.



xx1800000050

Too low *Bandwidth* – The graph below shows the test signals for a too low value of the bandwidth, resulting in a too large gap between the torque and torque limit due to filtering, leading to an underutilized axis.



xx1800000051

4.5.8 Tuning Acc

Information

The *Acc* parameter corresponds to the path acceleration used for the axis. High acceleration improves cycle time. If the value is set too high, the limit for torque output from the motor will be reached and the axis will not be able to follow its programmed path, resulting in poor performance and path errors.

Tuning procedure

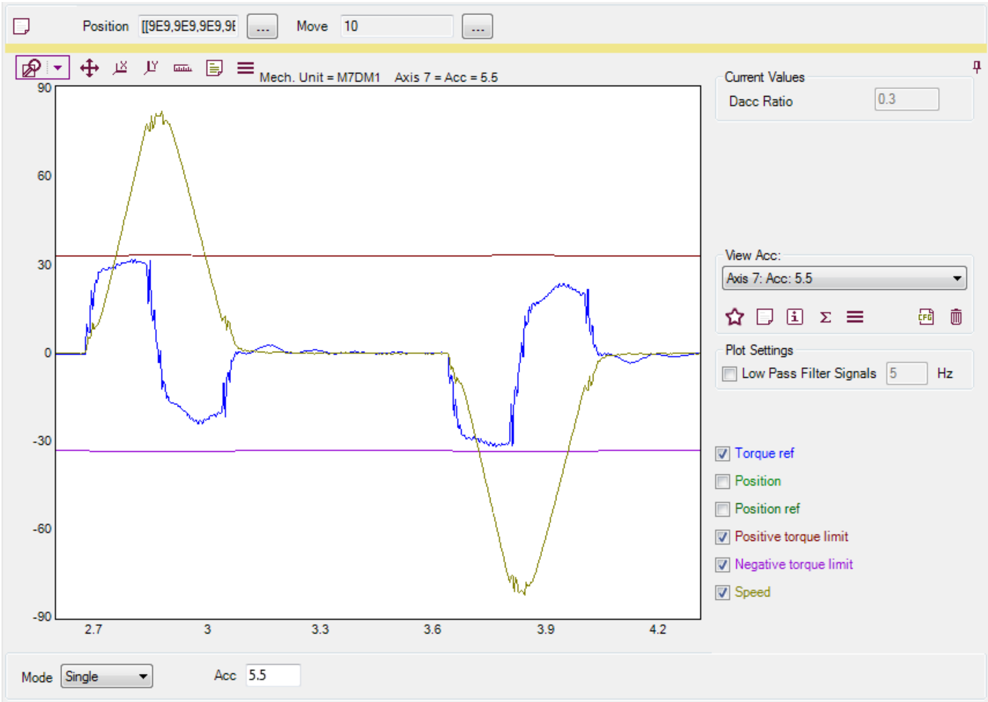
If an axis has a variable moment of inertia, *Acc* should be tuned with the maximum inertia. If gravity has an influence on the axis, then *Acc* should be tuned with a motion accelerating upwards against gravity. Due to friction the optimum value for *Acc* will typically be lower than the value for *Dec*.

Start with a medium value for *Acc*, typically 1. Observe if the *Torque ref* signal reaches either the positive or negative torque limit during acceleration, marked as the interesting region in the graphs below. Find a value for *Acc* that brings *Torque ref* close to the limits without exceeding them. If *Torque ref* does not reach the limits even for a high value of *Acc*, try increasing the tune move length. If that does not help, try increasing `wc_dacc_ratio` in the **Cfg info** tab. Note that depending on the sign of transmission on the axis, the acceleration plateau may be either positive or negative. In both cases it is still the first of the two plateau's that is the acceleration plateau.

Continues on next page

Reference graphs

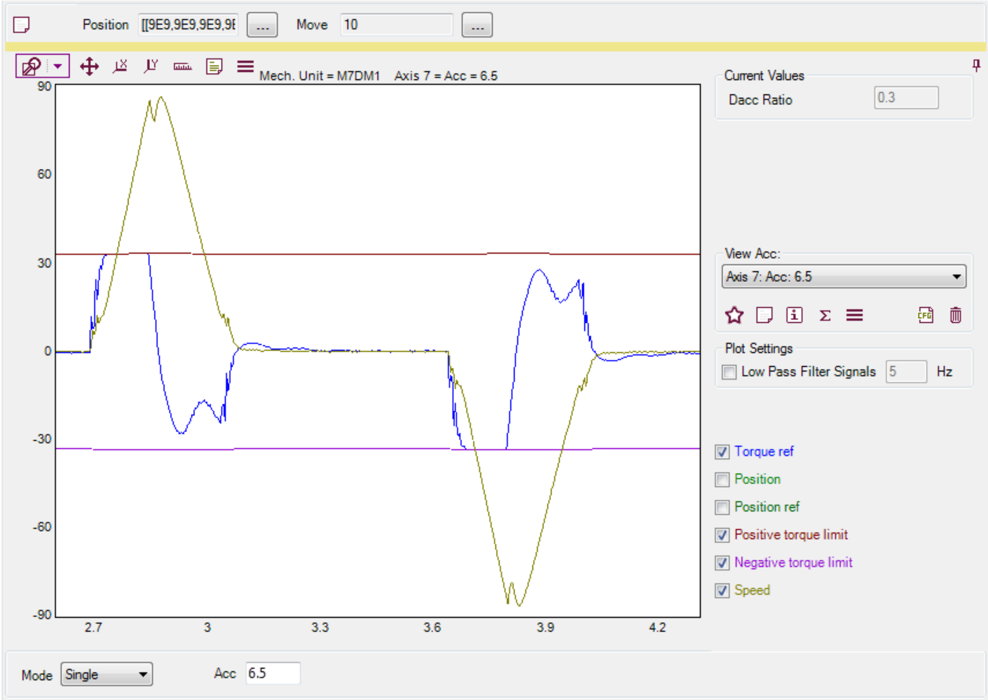
Correct *Acc* - The graph below shows that the *Torque ref* signal is close to the torque limit within the acceleration region, while keeping a small distance as a safety margin. The axis used has a positive transmission leading to a positive torque during acceleration.



xx1700001224

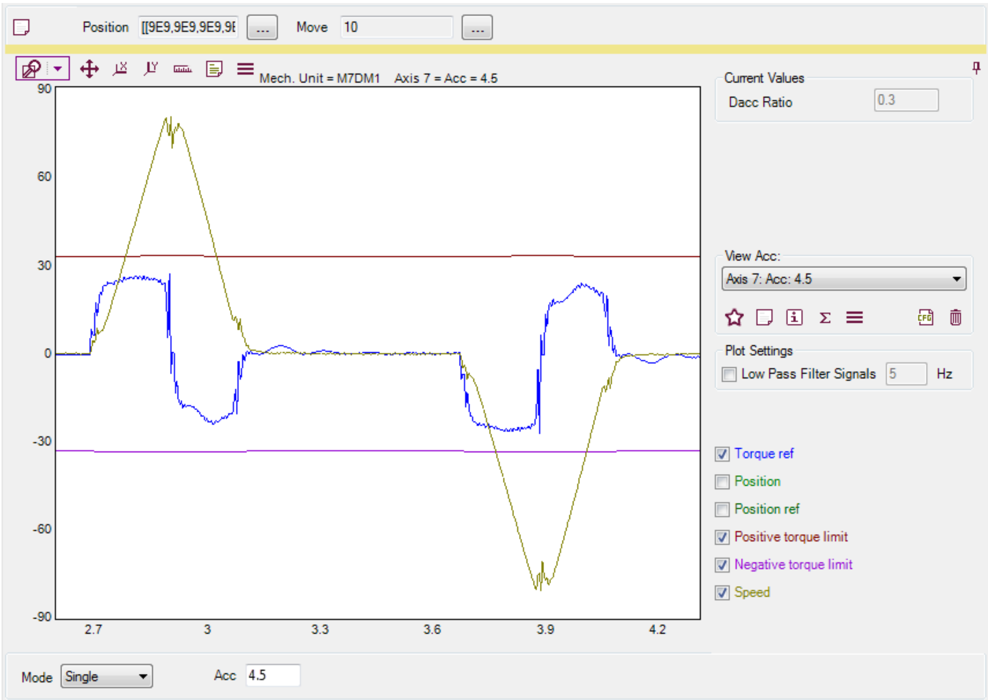
Continues on next page

Too high *Acc* - The *Torque ref* signal clearly hits the limits during acceleration.



xx1700001225

Too low *Acc* - There is an unnecessarily large gap between the *Torque ref* signal and the torque limits, leading to an underutilized axis.



xx1700001226

4.5.9 Tuning Dec

Information

The *Dec* parameter corresponds to the path deceleration used for the axis. High deceleration improves cycle time. If the value is set too high, the limit for torque output from the motor will be reached and the axis will not be able to follow its programmed path, resulting in poor performance and path errors.

Tuning procedure

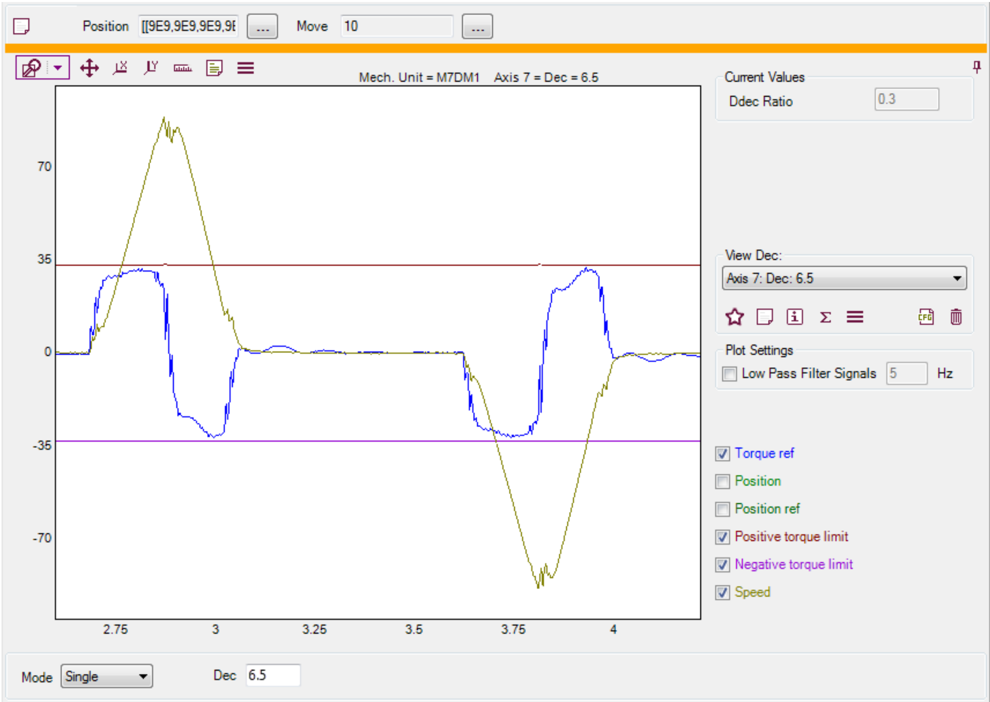
If an axis has a variable moment of inertia, *Dec* should be tuned with the maximum inertia. If gravity has an influence on the axis, then *Dec* should be tuned with a stopping motion (declaration) while moving downwards with gravity. Due to friction the optimum value for *Dec* will typically be higher then the value for *Acc*.

Start with a medium value for *Dec*, typically 1. Observe if the *Torque ref* signal reaches either the positive or negative torque limit. Find a value for *Dec* that brings *Torque ref* close to the limits without exceeding them. If *Torque ref* does not reach the limits, try increasing the tune move length. If that does not help, try lowering `wc_ddec_ratio` in the **Cfg info** tab. Note that depending on the sign of transmission on the axis, the deceleration plateau may be either positive or negative. In both cases it is still the second of the two plateau's that is the deceleration plateau.

Continues on next page

Reference graphs

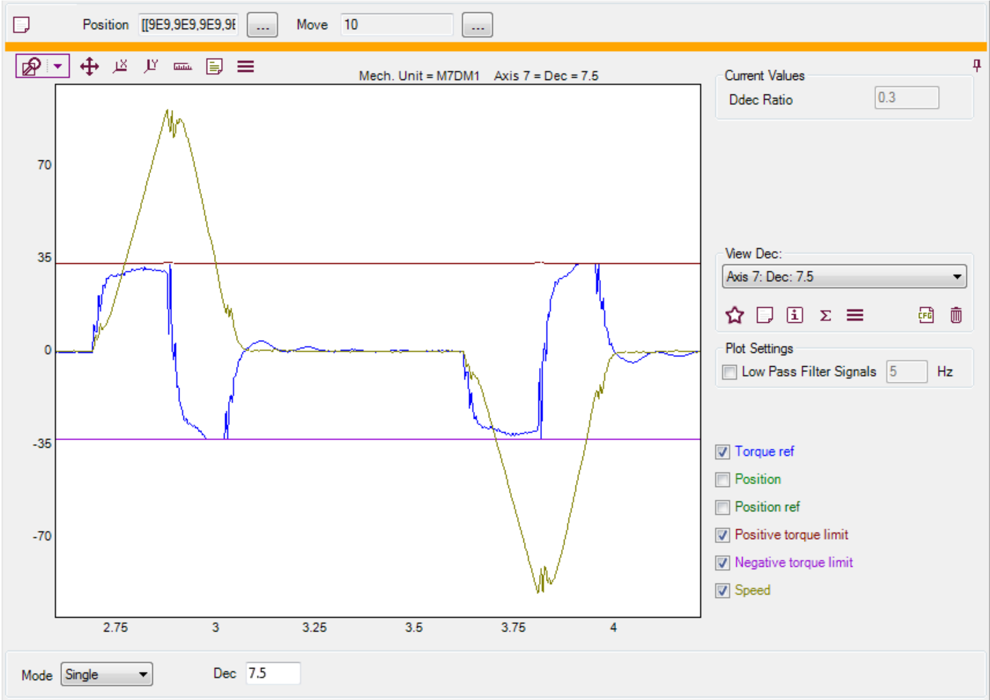
Correct *Dec* - The graph below shows that the *Torque ref* signal is close to the torque limit within the deceleration region, while keeping a small distance as a safety margin. The axis used has a positive transmission leading to a negative torque during acceleration.



xx1700001227

Continues on next page

Too high *Dec* - The *Torque ref* signal hits the limits during deceleration.



xx1700001228

Too low *Dec* - There is an unnecessarily large gap between the *Torque ref* signal and the torque limits, leading to an underutilized axis.

xx1700001229

4.6 Advanced tuning

4.6.1 Tuning Inpos Forced Control

Overview

Some axis may have a problem with "creeping" into an ordered position. The axis normally comes close to the ordered position and then starts to move slowly towards the final position. If this is a problem it is possible to reduce this effect by activating the *Inpos Forced Gain Control*. This will ramp up the control gains for positional feedback *(Kp)* and integral feedback, *(Ki = Kv / Ti)* when the axis is close to the ordered target, in the "inpos" region.

Note

Each axis has its own limits and forced factors, but change is made when all axis affected by forced gain control are inside its limit.

Inpos Forced Gain Control parameters

When activating forced gain control for an additional axis, two types in topic *Motion* must be considered. In *Lag Control Master 0 (LCM0)*, you can decide which axes should have forced gain control and in *Supervision*, you can decide which axes should affect forced gain control. All the axes that affect forced gain control must be within a certain position range from the end point before the forced gain control is enabled. This position range is also specified in *Supervision*

Type	Parameter	Description
LCM0	*Forced control active*	Determines whether forced gain control is active for this joint. If set to yes, *Affects Forced ctrl* in *Supervision* should normally also be set to yes for this joint (see below).
LCM0	*Forced factor for Kp*	The forced factor for *Kp*, if forced gain control is active.
LCM0	*Forced factor for Ki*	The forced factor for *Ki*, if forced gain control is active.
LCM0	*Raise Time for Kp*	The raise time for forced *Kp*, if forced gain control is active.
Supervision	*Affects forced ctrl*	Determines whether this joint affects forced gain control.
Supervision	*Forced on pos limit*	The upper position limit for forced gain control.
Supervision	*Forced off pos limit*	The lower position limit for forced gain control.

Tuning procedure

Start by selecting the joint you wish to tune and set *Forced Control Active* to true. To get reasonable staring values, click the **Generate Default Values** button. Click **Start** to perform the tune run.

The result of activating *Forced Control Active* should be that the axis moves faster into position the final few degrees and the "creeping" effect at the end of movement

Continues on next page

should disappear. If the system becomes oscillatory the forced factors should be reduced.

4.6.2 Tuning Soft Servo

Information

The option *Soft Servo* can be activated for additional axes which are configured with *Lag Control Master 0*. *Soft Servo* acts like a mechanical spring. When *Soft Servo* is active there is no integral action which means that steady state error is accepted. The stiffness of the servo is decided by the user. *Supervision* limits are automatically changed.

To activate/deactivate soft servo from a RAPID program use the commands `SoftAct` and `SoftDeact`.

In most applications soft servo is not used, and if it is used most often the default parameters are good enough, so no tuning is needed.

Soft Servo parameters

There are four system parameters to consider in *LCM0* when the soft servo is used for an additional axis.

Parameter	Description
K soft max factor	Determines the value of the product *Kp* Kv* when the *Soft Servo* is used with softness 0%. *K soft max factor* should be in the range 0.1 - 2.0 (default 1.0). When the *Soft Servo* is activated with 0% softness, the control parameters *Kp* and *Kv* will be tuned such that *Kp* Kv = (Kp* Kv)normal*K soft max factor*, where *(Kp * Kv)normal* is the product of *Kp* and *Kv* during normal operation.
K soft min factor	Determines the value of the product *Kp* Kv* if the *Soft Servo* is used with softness 100%. *K soft min factor* should be in the range 0.001 - 0.1 (default 0.01). When the *Soft Servo* is activated with 100% softness, the control parameters *Kp* and *Kv*are tuned such that *Kp* Kv = (Kp* Kv)normal*K soft min factor*.
Kp/Kv ratio factor	Factor used to alter the *Kp/Kv* ratio during soft servo. *Kp/Kv ratio factor* should be in the range 0.1 - 1.0 (default 1.0). In *Soft Servo* mode, *Kp* and *Kv* are tuned such that *Kp/Kv = (Kp/Kv)normal * Kp/Kv ratio factor*.
Ramp time	Default time for activation of the *Soft Servo*. The default value is 0.5 s.

Tuning procedure

Click the **Generate Default Values** button to get a set of values to start with.

Determine a maximum axis movement for which the axis should not move with softness 100%. Such a movement could be 0.1 rad on arm side for a rotating axis. Enter the value as *Move1*.

Determine a minimum axis movement for which the axis should move with softness 100%. Such a movement could be 0.2 rad on arm side for a rotating axis. Enter the value as *Move2*.

Click **Start**. The goal is to adjust the *Soft Servo* parameters so that the axis does not move for *Move1* but does move for *Move2*.

If the axis moves for both movements the axis is too stiff and the value of *K soft min factor* should be reduced. If the axis does not move for any movement the axis is too soft and the value *K soft min factor* should be increased. Keep tuning the

Continues on next page

value of *K soft min factor* until the axis moves for the longer movement and does not move for the short movement. If a ordered softness of 0% gives a too stiff, or too soft, servo the factor *K soft max factor* should be changed. If a ordered softness of 100% gives a too stiff, or too soft, servo the factor *K soft min factor* should be changed. The value of *Kp/Kv ratio factor* determines the stability margin for the axis. A value below 1 can increase the stability of the axis.

Tuning of Ramp time will change the activation/deactivation time of soft servo. A short ramp time can result in a jerk at activation.

4.7 Tuning Uncalibrated Control Master 0 parameters

Tuning procedure

	Action
1	Copy the values of *Kv, Gain Speed Loop*, *Kp, Gain Position Loop*, and *Ti Integration Time Speed Loop* from the type *Lag Control Master 0*.
2	Set the tuning value of *Speed Max Uncalibrated*. Maximum speed for uncalibrated axis (rad/s on motor side).
3	Set the tuning value of *Deceleration Max Uncalibrated*. Maximum deceleration for uncalibrated axis (rad/s^2 on motor side). Recommended value: *Nominal Deceleration* * *Transmission Gear Ratio*.
4	Set the tuning value of *Acceleration Max Uncalibrated*. Maximum acceleration for uncalibrated axis (rad/s^2 on motor side). Recommended value: *Nominal Acceleration* * *Transm Gear Ratio*.

This page is intentionally left blank

5 TuneMaster - Log Signals and Servo Gun

5.1 Working with signals

Adding test signals

One test signal is added by default in the signal table. Click **Add** to add more signals. For each signal, select the mechanical unit and axis and enter the signal number. To find the signal numbers for additional axes, use the **Find signal** button.

To switch between logging one, several or robot axes, click **Log several axes**, **Log only one axis** or **Log robot axes**. The axis will change for all defined signals if one axis is selected. If robot is selected signals for all axes will be added when double-clicking on the **Add** button. When changing a signal number all signals with the same number will be changed at the same time.

The check box for each signal defines if the signal should be visible or not in the plot. The **Check**/**Uncheck All** button can be used to change all signals at once.

To save the current signal definitions, click **Save signals**. Click **Open signals** to load saved signals.

I/O-signals

I/O-signals can be logged instead of normal test signals. Enter a signal name instead of a test signal number in the signal no column.

Signal functions

It is possible to plot a function of measured signals. Add a new signal last in the list of signals and enter a signal function instead of a signal number. Operators +, -, *, / and ^ can be used, see some examples below

- *f=s1-s2* is the difference between the first (s1) and second signal (s2) in the list
- *f=s1/s2* is the quota between signal 1 and 2
- *f=(s1^2+s2^2+s3^2)^0.5* is the distance if signal 1, 2 and 3 is x, y and z values

Deleting test signals

To remove a test signal, select the signal and click **Delete signal**.

5.2 Data logging

Start logging

Click **Start** to start logging of the defined signals.

Plot handling

By default all measured data is visible in the plot.

To limit the plot to only show the latest data, click and drag to draw a square that will set the limits. To show all data again, click **Reset X Zoom** and/or **Reset Y Zoom**. For more information, see *Working with the plot on page 20*.

Stop logging

Click **Stop** to stop the logging.

After logging has stopped, the measured data will be added to the open project and can be accessed from the list in the View Run part of the program where all measurements can be shown later on.

Trig logging

It is possible to trig/stop logging by defining a Trig for one/several signals. Right-click on the signal in the signal table. A menu will be shown where it is possible to define a Trig. Select **Trig Type**, define the **Limit**, **Post Time**, and if **Restart Logging** and **On Falling edge** if desired, and click **Set Trig** in the menu to finish. **Post time** is the time to stop logging after the trig occurred. **Restart logging** defines if a new log sequence should be started or not when a trig occurs. **On Falling edge** defines if trig should occur on falling instead of rising edge. This can be used if you want to log a sequence in a program. Use a digital output active during the sequence. Set the trig with type **GreaterThan**, limit to 0.5 and set **Restart Logging** and **On Falling edge** active.

It is also possible to restart logging when an event log is generated on the controller. Add event logs that should restart logging by the **Add Trigger for Event Log** above the log signals table.

Delete measurements

Click **Delete** in the View run part of the program to delete measurements.

Comments

It is possible to enter a comment for each measurement. To change a comment, click **Comment** in the View run part of the program. A comment can also be added before a measurement is done, then enter the comment in the textbox in the top part of the program. This textbox has a link that is deactivated by default, when this is active the text will be updated with the comment for the measurement that is displayed. To activate/deactivate the link, click **Link**.

Index

ABB AB, Robotics
Robotics and Motion
S-721 68 VÄSTERÅS, Sweden
Telephone +46 (0) 21 344 400

ABB AS, Robotics
Robotics and Motion
Nordlysvegen 7, N-4340 BRYNE, Norway
Box 265, N-4349 BRYNE, Norway
Telephone: +47 22 87 2000

ABB Engineering (Shanghai) Ltd.
Robotics and Motion
No. 4528 Kangxin Highway
PuDong District
SHANGHAI 201319, China
Telephone: +86 21 6105 6666

ABB Inc.
Robotics and Motion
1250 Brown Road
Auburn Hills, MI 48326
USA
Telephone: +1 248 391 9000

abb.com/robotics

3HAC063590-001, Rev D, en