

# Motion System Tuning



---

## Summary of Changes

This publication contains the following new or updated information. This list includes substantive updates only and is not intended to reflect all changes.

Topic	Page
Added information about Kinetix 5300 drives	throughout
Define new command notch and adaptive tuning enhancements (available with Kinetix 5300 drives and Kinetix 5700 drives with firmware revision 13.001 and later)	throughout

## Table of Contents

---

<b>Preface</b>	
About This Publication .....	7
Download Firmware, AOP, EDS, and Other Files .....	7
Conventions Used in This Manual.....	7
<b>Chapter 1</b>	
<b>Background</b>	
Damping Factor .....	13
Drive Model Time Constant .....	14
Torque Scalar .....	14
Load Observer Feature.....	16
Benefits .....	16
How It Works .....	17
Sercos Drive Configuration .....	19
EtherNet/IP Drive Configuration.....	20
Adaptive Tuning Feature .....	22
Benefits .....	22
Resonances.....	22
How It Works .....	25
Configuration.....	26
Adaptive Tuning Modes .....	28
<b>Chapter 2</b>	
<b>Out-of-Box Tuning</b>	
Sercos Drives .....	35
Gain Calculation .....	35
Sercos Drive Recommended Settings .....	36
EtherNet/IP Drives .....	38
Gain Calculation .....	38
EtherNet/IP Drive Recommended Settings .....	39
Is Further Tuning Required? .....	41
<b>Chapter 3</b>	
<b>Autotuning</b>	
Sercos Drives .....	43
Bump Test .....	43
Gain Calculation .....	44
Using Load Observer with Autotune .....	48
EtherNet/IP Drives .....	49
Bump Test .....	49
Gain Calculation .....	50
Using Load Observer with Autotune .....	55
Is Further Tuning Required? .....	56

<b>Manual Tuning</b>	<b>Chapter 4</b>	
	Initialize the Axis (Optional) .....	57
	Tune the Velocity Loop .....	58
	Tune the Position Loop .....	61
<b>Filters and Compensation</b>	<b>Chapter 5</b>	
	Compensating for High Frequency Resonances .....	63
	Automatic Compensation .....	63
	Manual Compensation .....	64
	Notch Filters .....	65
	Sercos Drives .....	67
	EtherNet/IP Drives .....	67
	Configuration .....	68
	Low Pass Filter .....	73
	Sercos Drives .....	73
	EtherNet/IP Drives .....	74
	EtherNet/IP Drive Lag Filter .....	74
	Friction Compensation .....	75
	Static Friction .....	76
	Sliding Friction .....	76
	Viscous Friction .....	76
	Backlash Compensation .....	77
	Sercos Drives .....	78
	EtherNet/IP Drives .....	78
<b>Kinetix 300 Drive Tuning</b>	<b>Chapter 6</b>	
	Control Loops .....	79
	Autotune .....	79
	Tuning Method .....	80
	How It Works .....	81
	Manual Tune .....	82
	Manual Tune Example .....	84
	Filters .....	85
	Feedback Filter and Low Pass Filter .....	86
	Resonator and Notch Filter .....	86
<b>Create Move Profiles for Tuning</b>	<b>Appendix A</b>	
	Example 1 .....	89
	Example 2 .....	90
	Example 3 .....	91
<b>Create Trends for Tuning</b>	<b>Appendix B</b>	
	RSTrendX Properties .....	93
	Name Tab .....	93
	General Tab .....	93
	Display Tab .....	94
	Pens Tab .....	94

X-Axis Tab .....	96
Y-Axis Tab .....	96
Sampling Tab.....	97
Value Bar Information .....	98
<b>Appendix C</b>	
<b>Set Sercos Gains with IDN Write</b>	
<b>Messages</b>	
Set Load Observer Configuration Attribute .....	101
Set Load Observer Gains .....	102
<b>Appendix D</b>	
<b>Vertical Axis Tuning</b>	
Overview .....	103
Torque Offset Method.....	103
Integrator Gain Method.....	103
Considerations .....	104
<b>Appendix E</b>	
<b>Block Diagrams</b>	
Sercos Drives .....	106
EtherNet/IP Drives .....	107
Position Loop.....	107
Velocity Loop.....	108
Torque Loop.....	109
<b>Additional Resources</b>	



## About This Publication

This publication can assist you in tuning a Kinetix® drive system. This document is intended for motion control users with novice to advanced skill levels that are familiar with the following:

- Kinetix servo drives
- Sercos or EtherNet/IP communication
- Use of the Studio 5000 Logix Designer® application to create a motion axis
- Understanding how control loops work in motion control applications

Each component of the control structure is described in detail and out-of-box tuning, autotuning, and manual tuning techniques are presented.

## Download Firmware, AOP, EDS, and Other Files

Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes from the Product Compatibility and Download Center at [rok.auto/pcdc](http://rok.auto/pcdc).

## Conventions Used in This Manual

These conventions are used throughout this manual:

- Bulleted lists such as this one provide information, not procedural steps
- Numbered lists provide sequential steps or hierarchical information
- Integrated Motion on the EtherNet/IP network uses CIP Motion and CIP Sync technology from ODVA, all built on the Common Industrial Protocol (CIP). For convenience, drives that use this technology are referred to as EtherNet/IP drives.

**Notes:**

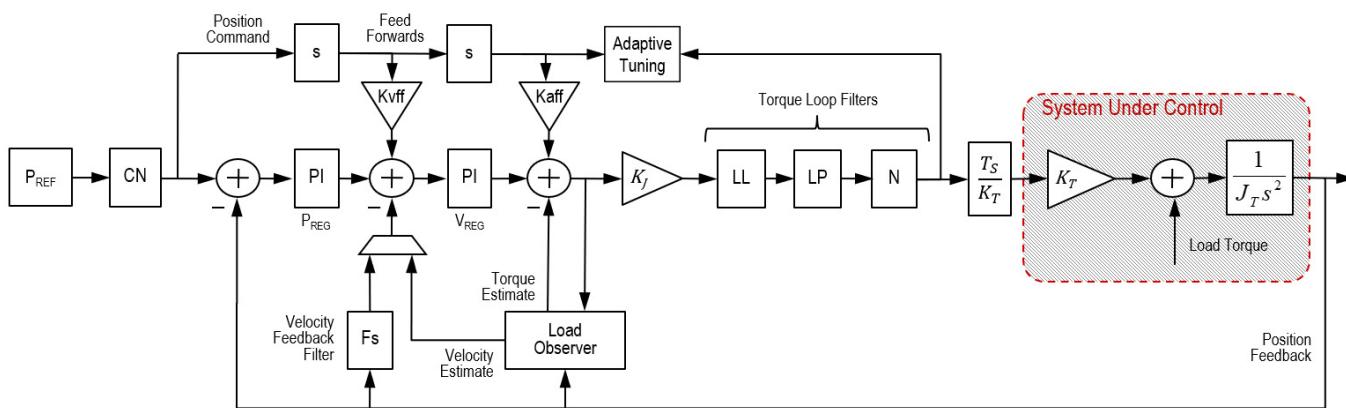
## Background

This chapter introduces you to the Kinetix® servo control loop structure, load observer, adaptive tuning features, and describes how you can use them for tuning your Kinetix drive system.

Topic	Page
Servo Loop Bandwidth	12
Damping Factor	13
Drive Model Time Constant	14
Torque Scalar	14
Load Observer Feature	16
Adaptive Tuning Feature	22

Kinetix servo drives implement an acceleration/torque loop, which is nested within a velocity PI control loop, which is nested within an outer position PI control loop. Each element in [Figure 1](#) is described in subsequent sections.

**Figure 1 - Kinetix Servo Control Loop Structure**



This type of control structure includes these advantages:

- Precise control of position, velocity, and torque
- Ability to switch between position, velocity, and torque modes without changing tuning gains
- Simple 'Inside-Out' tuning, described on page [57](#).

**Lead-lag Filter (LL)** - The lead-lag filter is new in drives that support integrated motion on EtherNet/IP and is not present in Sercos drives. See [page 74](#) for more information about the lead-lag filter.

**Position Command and Feedforward Commands** - The fine interpolators in EtherNet/IP drives generate a third order position command, a second order velocity feed forward, and a first order acceleration feed forward. In Sercos drives, the fine interpolators generate a second order position command, a first order velocity feed forward, and a zero order acceleration feed forward. A zero order acceleration feed forward signal means that it is actually updated at the coarse update rate (CUR). This fact is one reason why many Sercos applications do not benefit from the use of acceleration feed forward. Unlike Sercos drives, the EtherNet/IP drive fine interpolators also handle variable update times to compensate for non-deterministic networks.

**Command Notch (CN) Filter** - Two command notch filters are placed on command signals to remove load-side resonances and vibration that is common in robots, cranes, cantilevered loads, liquid sloshing, laser cutting, and material handling applications. See [Resonances on page 22](#) for more information. The frequency, width, depth, and gain of these filters can be adjusted. See [Notch Filters on page 65](#) for more information.

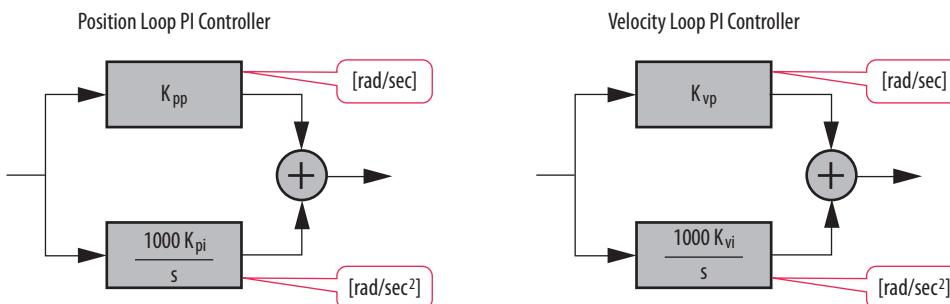
**Torque Scalar (KJ) and Motor Torque Constant (KT)** - The torque scalar affects tuning and overall motion performance. The motor torque constant affects the torque scalar. See [Torque Scalar on page 14](#) for more information on the torque scalar, motor torque constant, and the differences between EtherNet/IP drives and Sercos drives.

**Load Observer** - See [Load Observer Feature on page 16](#) for more information on the load observer.

**Adaptive Tuning** - See [Adaptive Tuning Feature on page 22](#) for more information on adaptive tuning and tracking notch filtering.

**Position and Velocity Loop Controllers** - In a Sercos drive, the position and velocity loop controllers are configured with a proportional term (top) in parallel with an integral term (bottom) as shown in [Figure 2](#). The proportional term equals the control loop bandwidth in units of [rad/sec]. However, the integral term has a squared relationship to the control loop bandwidth in units of [rad/sec<sup>2</sup>]. A factor of 1000 is applied to numerically keep the integral gain in the same range as the proportional gain and attempt to counteract the squared relationship.

**Figure 2 - Sercos PI Controllers - Parallel Form**

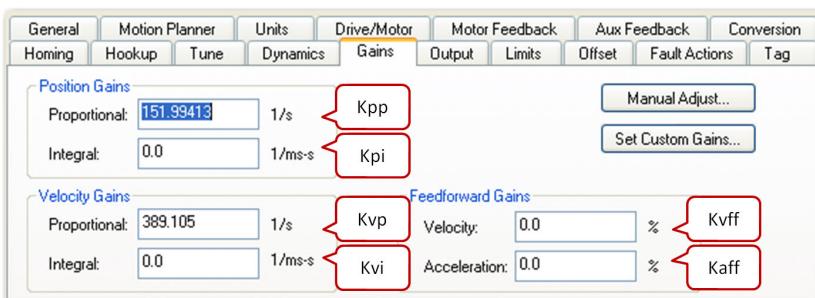


This configuration is standard in classical control theory, except the factor of 1000. The following definitions are control loop gains of a Sercos drive:

- $K_{PP}$  = Position Proportional Gain [rad/s]
- $K_{PI}$  = Position Integral Gain [rad/s/ms]
- $K_{VP}$  = Velocity Proportional Gain [rad/s]
- $K_{VI}$  = Velocity Integral Gain [rad/s/ms]
- $K_{VFF}$  = Velocity Feedforward [%]
- $K_{AFF}$  = Acceleration Feedforward [%]

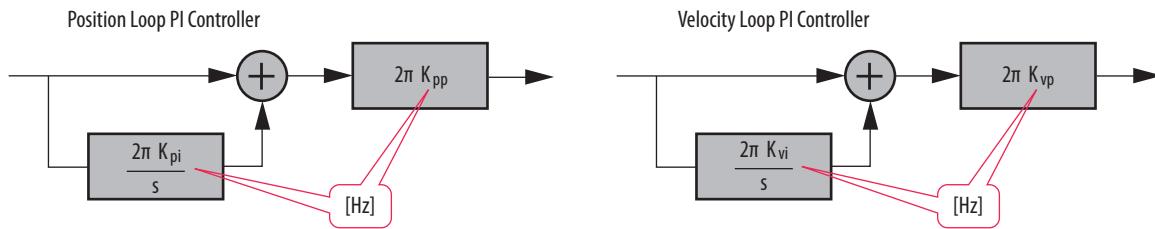
These gains can be accessed on the Gains tab of the Axis Properties dialog box in the Studio 5000 Logix Designer® application.

**Figure 3 - Sercos Control Loop Gains in the Logix Designer Application**



In an EtherNet/IP drive, the position and velocity loop controllers are configured with a proportional term (top) in series with an integral term (bottom) as shown in [Figure 4](#). A factor of  $2\pi$  is also applied to each gain. This configuration places all control loop gains in units of [Hz]. As a result, all gains are proportional to each other and represent bandwidths that relate directly to physically measurable signals that are easy to understand. More importantly, removing any squared relationships simplifies the math when tuning an axis, because all gains are related to each other by simple ratios.

**Figure 4 - EtherNet/IP PI Controllers - Series Form**

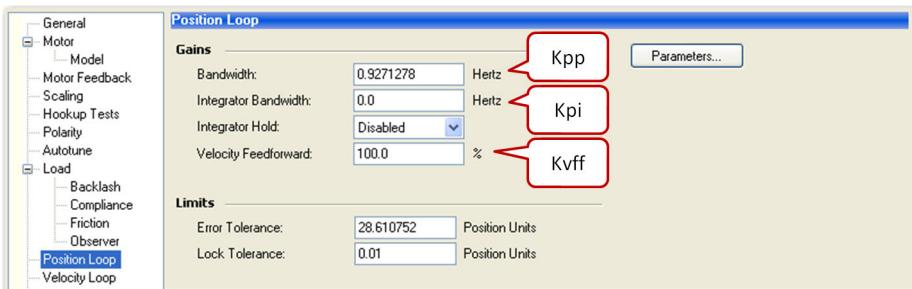


The following definitions are control loop gains of an EtherNet/IP drive:

- $K_{PP}$  = Position Loop Bandwidth [Hz]
- $K_{PI}$  = Position Integral Bandwidth [Hz]
- $K_{VFF}$  = Velocity Feedforward [%]
- $K_{VP}$  = Velocity Loop Bandwidth [Hz]
- $K_{VI}$  = Velocity Integral Bandwidth [Hz]
- $K_{AFF}$  = Acceleration Feedforward [%]

These gains can be accessed on the Position Loop and Velocity Loop tabs of the Axis Properties dialog box in the Logix Designer application.

**Figure 5 - EtherNet/IP Position Loop Gains in the Logix Designer Application**



**Figure 6 - EtherNet/IP Velocity Loop Gains in the Logix Designer Application**



Use the following equations to convert existing Sercos control loop gains to equivalent EtherNet/IP control loop gains

$$K_{PP(CIP)} = \frac{K_{PP(\text{Sercos})}}{2\pi}$$

$$K_{VP(CIP)} = \frac{K_{VP(\text{Sercos})}}{2\pi}$$

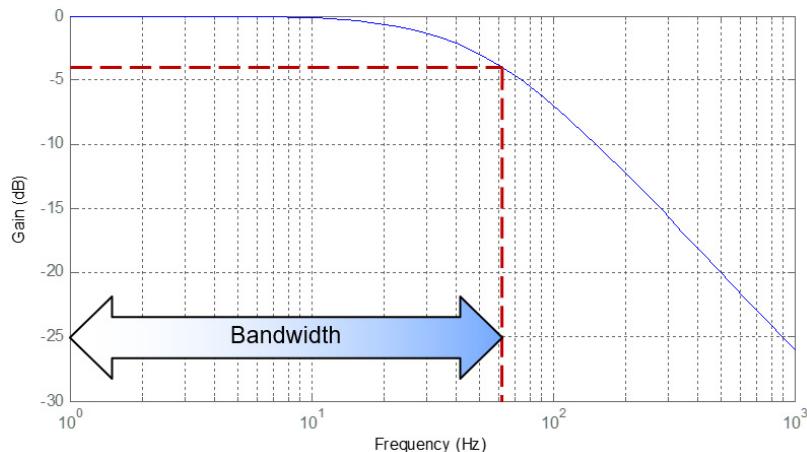
$$K_{PI(CIP)} = \frac{1000 K_{PI(\text{Sercos})}}{2\pi K_{PP(\text{Sercos})}}$$

$$K_{VI(CIP)} = \frac{1000 K_{VI(\text{Sercos})}}{2\pi K_{VP(\text{Sercos})}}$$

## Servo Loop Bandwidth

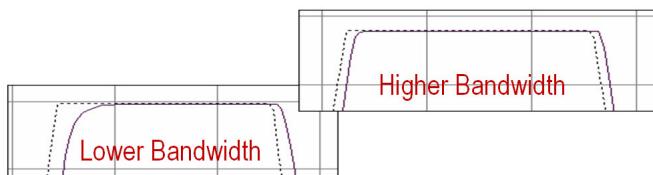
Bandwidth (BW) is a widely used term that is critical to servo drive performance. It is defined as the usable range of frequencies in [Hz] where the gain through the system is above -3 dB. Bandwidth indicates servo drive performance and directly equates to transient response and how fast the servo physically responds to the load. There must be a way to qualify performance for different servo axes, and the metric that makes the most sense is Bandwidth that is measured in [Hz].

**Figure 7 - System Gain versus Frequency**



Position, velocity, and torque loop bandwidth indicate the respective performance of each loop in a servo drive. Higher bandwidth improves transient response, decreases error, and makes the motor performance stiffer. [Figure 8](#) shows how bandwidth affects actual response with feed forwards disabled (solid) compared to its command motion profile (dashed).

**Figure 8 - How Bandwidth Affects Transient Response**



These factors affect servo drive bandwidth:

- Feedback resolution (higher is better)
- Motor-to-Load inertia ratio (lower is better)
- Drive loop update rate (higher is better)
- Load compliance (rigid coupling is better)
- Drive Model Time Constant (lower is better)

See [Drive Model Time Constant on page 14](#) for more information on this topic.

The drive, motor, and feedback device have a significant impact on the bandwidth that can be achieved on an axis through tuning. [Table 1](#) shows bandwidth values for various drive and motor combinations. Each has different loop update rates and feedback types.

**Table 1 - Bandwidth Comparison for Various Drive and Motor Combinations**

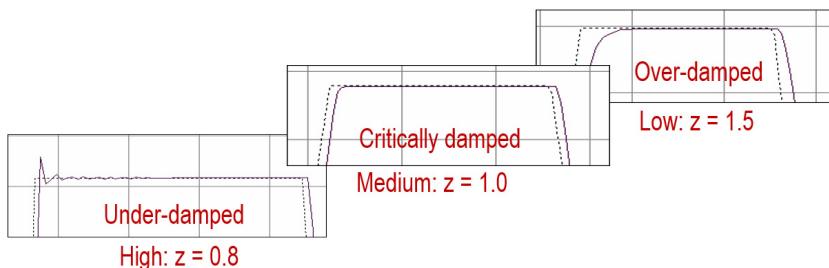
Drive or Servo Control Module	Loop Update Rate			Motor Bulletin Number	Drive Model Time Constant $\mu$ s	Torque Loop Bandwidth Hz
	Position $\mu$ s	Velocity $\mu$ s	Torque $\mu$ s			
Kinetix 350	500	500	125	MPL-S/M <sup>(1)</sup>	1003.9	158.53
				MPL-E/V <sup>(1)</sup>	1031.25	154.33
				MPL-H <sup>(2)</sup>	2024	78.65
				TLY	1062.5	149.8
Kinetix 5300	125	125	125	TLP	507	314.27
Kinetix 5500 Kinetix 5700	125	125	125	VPL	537	296.34
Kinetix 6000 Kinetix 6200 Kinetix 6500	125	125	125	MPL-S/M <sup>(1)</sup>	510.9	311.5
				MPL-E/V <sup>(1)</sup>	538.25	295.6
				MPL-H <sup>(2)</sup>	1531	103.95
1756-M02AE Servo Module	500	250	-	-	1502	106

(1) This motor has a high-resolution encoder.

(2) This motor has an incremental (low resolution) encoder.

## Damping Factor

Damping factor is commonly referred to as zeta ( $z$ ). It affects the rise time for a given bandwidth. [Figure 9](#) shows how the damping factor affects actual response with feed forwards disabled (solid) compared to its command motion profile (dashed).

**Figure 9 - How Damping Affects Transient Response**

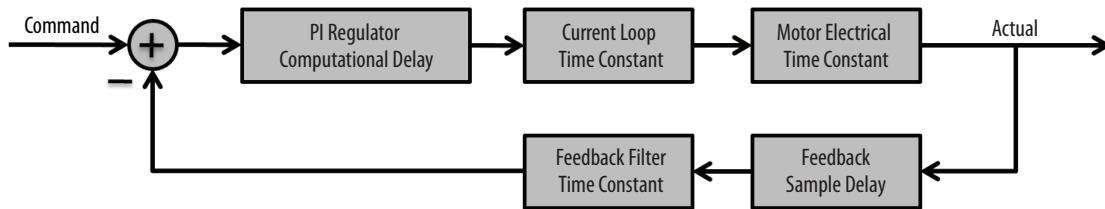
- A damping factor of  $z = 0.8$  produces high responsiveness, which is characterized by a faster rise time, but with overshoot.
- A damping factor of  $z = 1.0$  produces medium responsiveness, which is characterized by the fastest possible rise time without overshoot. We recommend this setting.
- A damping factor of  $z = 1.5$  produces low responsiveness, which is characterized by a slower response, similar to decreasing the bandwidth.

The default setting in a Sercos drive is  $z = 0.8$  and the default setting in an EtherNet/IP drive is  $z = 1.0$ . A lower damping factor decreases the spacing between the position, velocity, and torque loop bandwidths, generating under-damped responses in actual position and actual velocity trends. A higher damping factor generates over-damped responses.

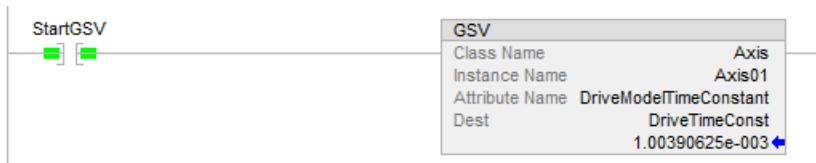
## Drive Model Time Constant

Drive model time constant (DMTC) is the sum of all delays around the torque loop for a given drive and motor. [Figure 10](#) shows the delays that are associated with the DMTC.

**Figure 10 - Delays Associated with DMTC**



DMTC values for different Kinetix drives are shown in [Table 1 on page 13](#). It can be obtained in the Logix Designer application through a GSV instruction as shown in this example.



The DMTC is used to calculate the torque loop bandwidth out-of-box and during an autotune. The position and velocity loop gains are then auto-calculated from the Torque Loop Bandwidth.

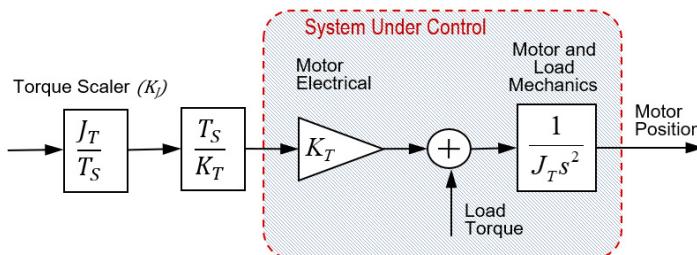
$$\text{Torque Loop Bandwidth } (T_{BW}) = \frac{1}{2\pi \text{ DMTC}} [\text{Hz}]$$

The Torque Loop Bandwidth displays a default of 1000 Hz under the Torque/Current tab in the Axis Properties dialog box of the Logix Designer application. It is important to note that this value actually represents the Current Loop Bandwidth and must not be confused with the Torque Loop Bandwidth described here.

## Torque Scalar

Torque scalar is a torque loop gain that accounts for load inertia. It calibrates the control loops so that all gains represent physically measurable bandwidths. As a result, it scales the system under control to unity gain. In other words, the motor and load have a gain of one when the torque scalar is applied.

**Figure 11 - Scaling Torque in the System**

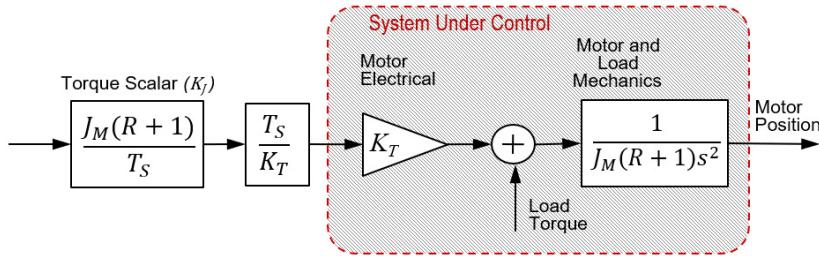


Scaling torque prevents the velocity loop response from being affected by motor gain or load inertia, and makes sure that the velocity loop response is represented by the velocity loop bandwidth set by  $K_{VP}$ . The torque scalar acts as an overall system gain. The following definitions are given:

- $K_T$  = Motor torque constant
- $J_M$  = Motor inertia
- $J_L$  = Load inertia
- $R = J_L / J_M$  = Load inertia ratio
- $J_T = J_L + J_M = J_M(R+1)$  = Total system inertia
- $T_R$  = Motor rated torque [NM]
- $T_S = T_R / (100 * 2 * \pi)$  = Scaled torque [% motor rated torque with respect to revolutions]

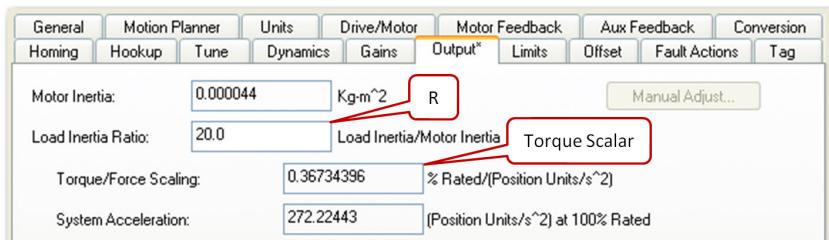
Torque scalar is a function of R (which is calculated by the autotune bump test) and two parameters,  $T_S$  and  $J_M$  (which are known by the Logix Designer application and queried from the motion database when a motor is selected).

**Figure 12 - Torque Scalar Calculation**



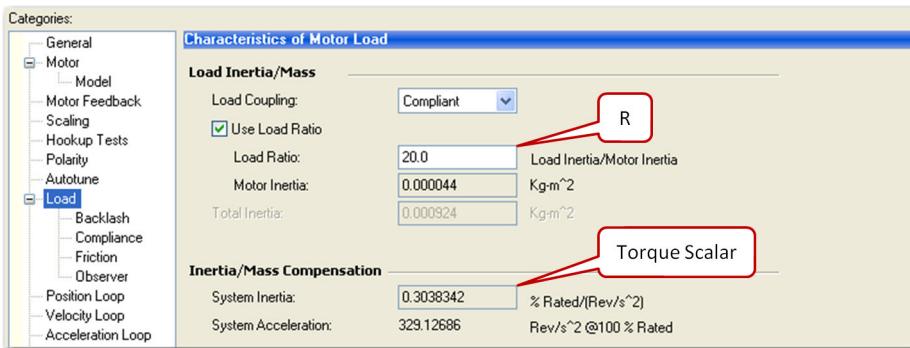
The torque scalar calculation is triggered when R is updated or a new motor is selected. For a Sercos drive, the Load Inertia Ratio is located on the Output tab of the Axis Properties dialog box in the Logix Designer application. The torque scalar is in the same location labeled Torque/Force Scaling.

**Figure 13 - Sercos Load Ratio and Torque Scalar**



For an EtherNet/IP drive, the Load Ratio is located on the Load tab of the Axis Properties dialog box in the Logix Designer application. The torque scalar is in the same location labeled System Inertia.

**Figure 14 - EtherNet/IP Drive Load Ratio and Torque Scalar**



You can enter R manually or you can run an autotune to calculate it automatically. Parameter values that are affected by the torque scalar are then updated.

Other considerations are given:

- When R is entered higher than the actual load ratio, the velocity loop bandwidth that is shown in the Axis Properties dialog box is artificially lower than its actual bandwidth. The opposite is also true.
- The actual value of  $K_T$  in the motor electrical decreases as the motor heats up. Sercos drives use a hot  $K_T$  value to compensate, which results in a torque scalar value that is approximately 20% higher than it is for similar EtherNet/IP drives.
- Since autotune is typically executed with a cold motor, EtherNet/IP drives calculate the 'cold motor'  $K_T$  to compensate. This calculation results in a more conservative torque scalar, approximately 20% lower than Sercos drives, guarding the system from instability.

## Load Observer Feature

The load observer feature is a control loop inside the drive that estimates the mechanical load on the motor and compensates for it. This allows the control loops to treat the motor as if it is unloaded and relatively easy to control. As a result, the load observer automatically compensates for disturbances and load dynamics, such as sudden inertia/torque changes, compliance, backlash, and resonances that are within the load observer bandwidth.

## Benefits

You can use load observer with out-of-box control loop gains, where the load is unknown and thus the Load Inertia Ratio = 0, or with autotuned control loop gains, where the Load Inertia Ratio is known or calculated by performing an autotune procedure.

When the load observer is enabled with the recommended out-of-box control loop gains, the load observer does the following:

- Provides relatively high-performance motion control without tuning
- Eliminates periodic retuning to account for machine wear over time
- Automatically compensates for changing vibration and resonances that are within the load observer bandwidth
- Eliminates periodic identification of in-band resonances to compensate for them

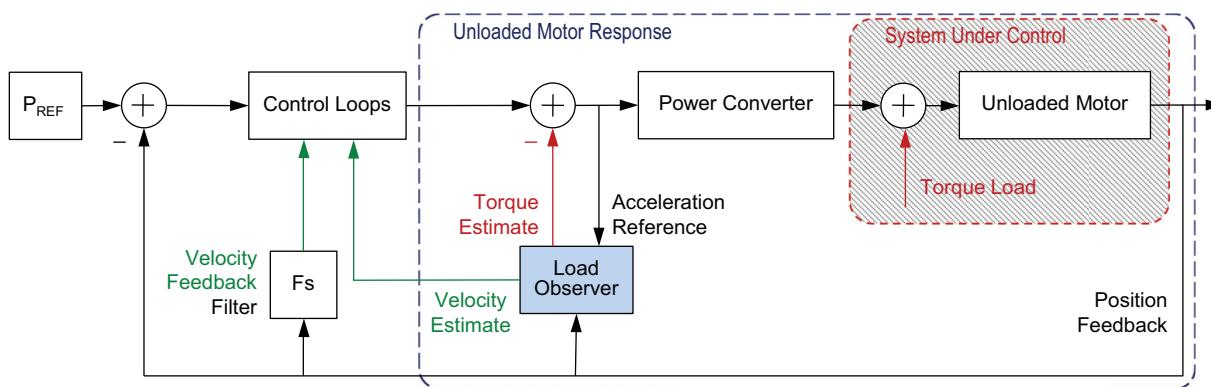
When used with autotuned control loop gains, the load observer does the following:

- Increases system bandwidth
- Reduces tracking errors, so line speeds can be increased
- Provides tighter control of moving parts, reducing wear and saving material costs

## How It Works

The load observer acts on the acceleration signal within the control loops and monitors the acceleration reference and the actual position feedback. The load observer models an ideal unloaded motor and generates a load observer torque estimate that represents any deviation in response of the actual motor and mechanics from the ideal model. This deviation represents the reaction torque placed on the motor shaft by the load mechanics. It is estimated in real time and compensated by closed loop operation.

**Figure 15 - Load Observer Block Diagram**



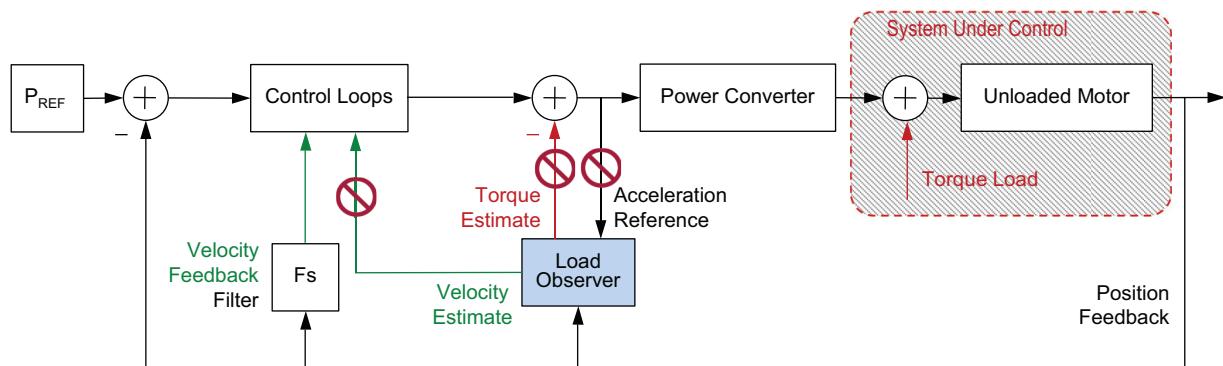
The load observer also generates a velocity estimate signal that you can apply to the velocity loop. The velocity estimate has less delay than the velocity feedback signal derived from the actual feedback device. It also helps to reduce high frequency output noise caused by the load observer's aggressive action on the acceleration reference. Together, the load observer with velocity estimate provides the best overall performance for positioning applications. [Table 2](#) describes the load observer mode settings that can be configured.

**Table 2 - Load Observer Configuration Settings**

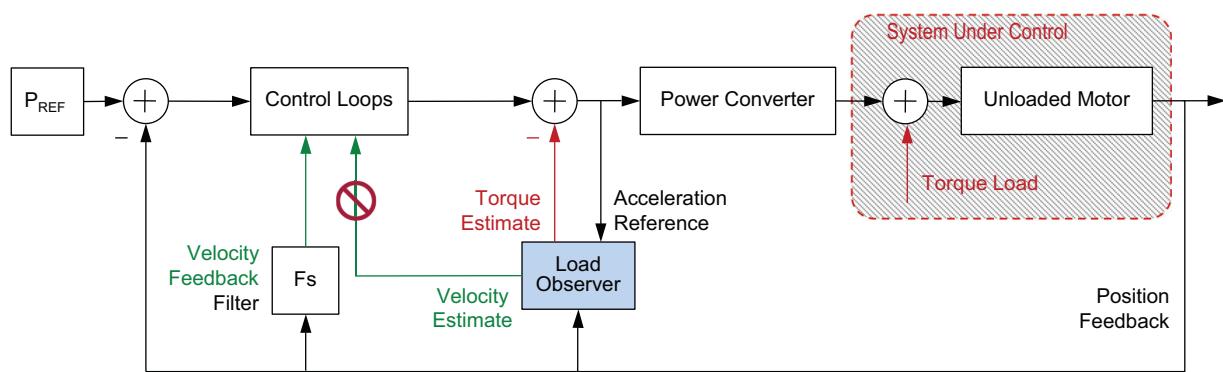
Mode	IDN P-431 Value	Description
Disabled (default)	0	Load observer is inactive.
Load Observer Only	1	Provides a Torque Estimate only This setting is similar to Acceleration Feedback but uses a second input signal, Acceleration Reference, for increased low frequency disturbance rejection (stiffness). It corrects error but is fairly aggressive. As a result, the bandwidth must often be cut in half or significantly reduced. It does not provide a Velocity Estimate in place of Velocity Feedback.
Load Observer with Velocity Estimate	2	Standard Operation: Provides Torque and Velocity Estimates This setting combines the best of the Load Observer Only and Velocity Estimate Only settings. Separately, velocity estimate provides a smooth response and reduces phase lag, but creates error, whereas load observer removes error, including steady state error in the velocity estimate, but it increases phase lag and is fairly aggressive. Together, they remove error and provide a smooth response. With the recommended out-of-box configuration, load observer performs extremely well in situations with changing inertia and unknown levels of compliance and backlash (vibration suppression).
Velocity Estimate Only	3	Provides a Velocity Estimate only This setting provides a Velocity Estimate in place of Velocity Feedback. This produces a smooth feedback signal but can add steady state error, generating a fictitiously lower velocity error. As a result of the potential error, it is not recommended to use in Velocity mode. Also, position integrator or observer integrator should be used with setting in Position mode.
Acceleration Feedback	4	Provides Acceleration Feedback by disconnecting the Acceleration Reference to Load Observer This setting creates a filtered acceleration feedback signal. It corrects errors but is fairly aggressive. As a result, the observer bandwidth must often be cut in half or significantly reduced for stable operation. This setting is similar to Load Observer Only but doesn't use an Acceleration Reference input signal to mitigate additional phase lag (delay) created by necessary filtering. It does not provide a Velocity Estimate in place of Velocity Feedback.

The following figures depict the high-level operation of each Load Observer mode.

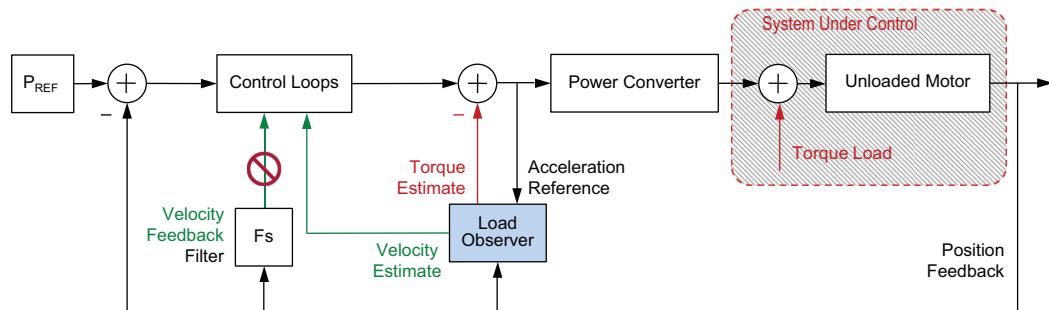
**Figure 16 - Load Observer Disabled Configuration (Value 0)**



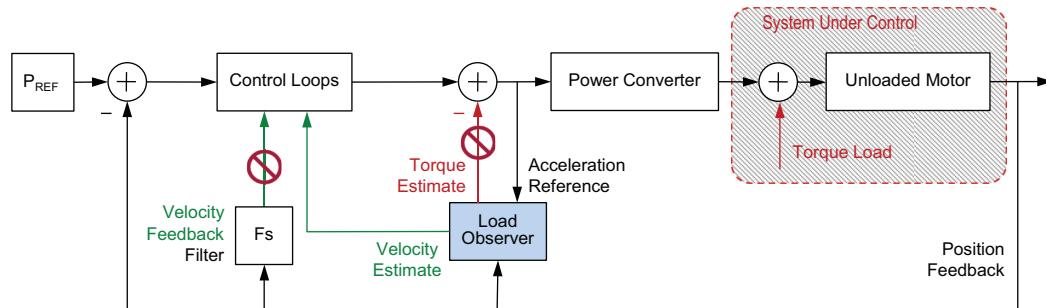
**Figure 17 - Load Observer Only Configuration (Value 1)**

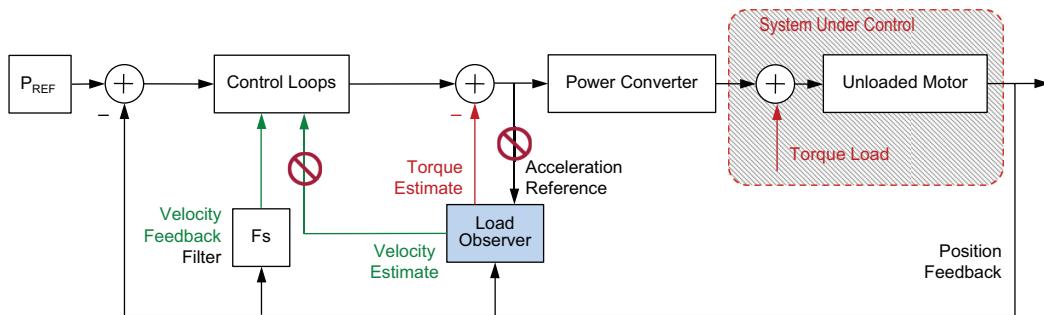


**Figure 18 - Load Observer with Velocity Estimate Configuration (Value 2)**



**Figure 19 - Velocity Estimate Only Configuration (Value 3)**

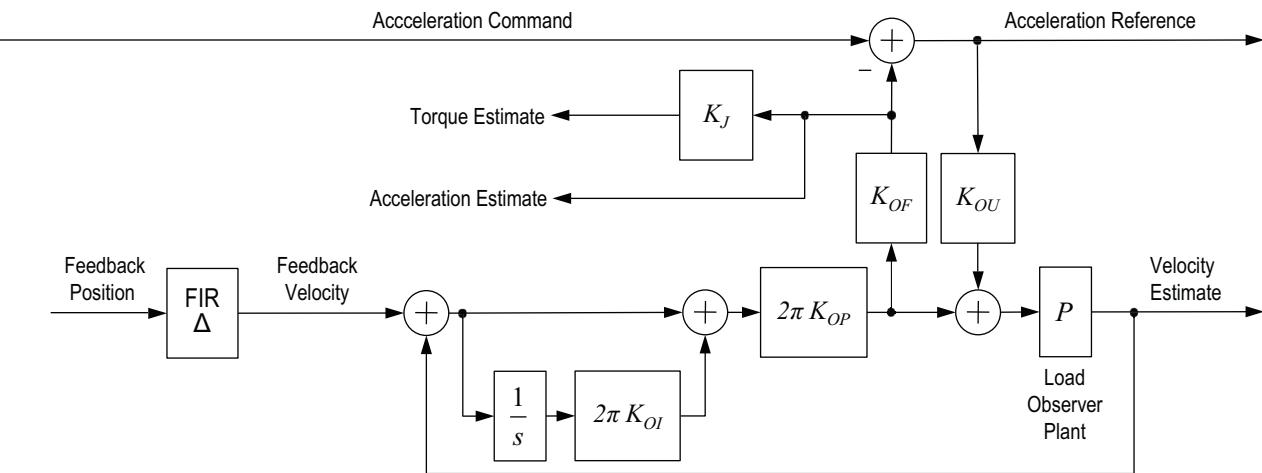


**Figure 20 - Acceleration Feedback Configuration (Value 4)**

## Sercos Drive Configuration

This section applies to the load observer feature in Kinetix 6000 and Kinetix 6200 drives. You can configure the load observer feature in a variety of ways by writing to a set of configuration IDN parameters. The overall behavior of the load observer is controlled by Load Observer Configuration (IDN P-431). This parameter is used to select the Load Observer mode. It can be set to the IDN values listed in [Table 2 on page 17](#).

For the remaining IDN descriptions, see [Figure 21](#).

**Figure 21 - Load Observer Internal Gains for Kinetix 6000 and Kinetix 6200 Drives**

Load observer gains that require user interaction are Load Observer Bandwidth ( $K_{OP}$ ) and Load Observer Integral Bandwidth ( $K_{OI}$ ). They are set by IDN P-432 and IDN P-433, respectively. Guidelines for setting these gains are provided in the following sections. In general,  $K_{OP}$  acts like a velocity integrator without windup and  $K_{OI}$  acts like a position integrator without windup. Typically,  $K_{OI} = 0$ .

Load observer gains that do not require user interaction are Load Observer Feedback Gain ( $K_{OF}$ ) and Load Observer Input Gain ( $K_{OU}$ ). They are automatically set internally based on the load observer configuration. However, when in Acceleration Feedback mode,  $K_{OF}$  can also be set manually by using IDN P-434 with typical values between zero and one.

**Table 3 - Load Observer Gain Parameters**

IDN	Name	Drive Variable	Units	Format	Value, min	Value, max
P:0:432	Load Observer Bandwidth ( $K_{op}$ )	$K_{op}$	Rad/s	16-bit unsigned integer	0	12,500 <sup>(1)</sup> 65,535 <sup>(2)</sup>
P:0:433	Load Observer Integral Bandwidth ( $K_{oi}$ )	$K_{oi}$	Rad/s			
P:0:434	Load Observer Feedback Gain ( $K_{of}$ )	$K_{of}$	-			200

- (1) This value applies to drive firmware revision 1.124 and 1.125.  
(2) This value applies to drive firmware revision later than 1.125.

The Acceleration Estimate and Torque Estimate signals are read with IDN P-435 and P-436, respectively. Definitions for these IDN parameters are given in [Table 4](#).

**Table 4 - Load Observer Output Signal**

IDN	Name	Units	Format	Value, min	Value, max
P:0:435	Load Observer Acceleration Estimate	Acceleration	32-bit signed integer	$-2^{31}$	$2^{31}-1$
P:0:436	Load Observer Torque Estimate	Torque	16-bit signed integer	$-2^{15}$	$2^{15}-1$

[Table 5](#) shows how the torque low-pass filter bandwidth is set in the drive based on what is entered in the Logix Designer application.

**Table 5 - Torque Low-pass Filter Bandwidth**

IDN P:0:065 Value	Bandwidth in the Logix Designer Application	Actual Bandwidth in Drive	Drive Firmware Notes
0	= 0	389 Hz	Operation before revision 1.116
	> 0	Limited to $\leq 389$ Hz	
1	= 0	Filter bypassed	Feature added to revision 1.116 and later
	> 0	Limited to $\leq 10,430$ Hz	

See [Appendix C on page 101](#) for more information on changing IDN parameter values in the Logix Designer application. Connected Components Workbench™ software can also be used to set IDN parameters.

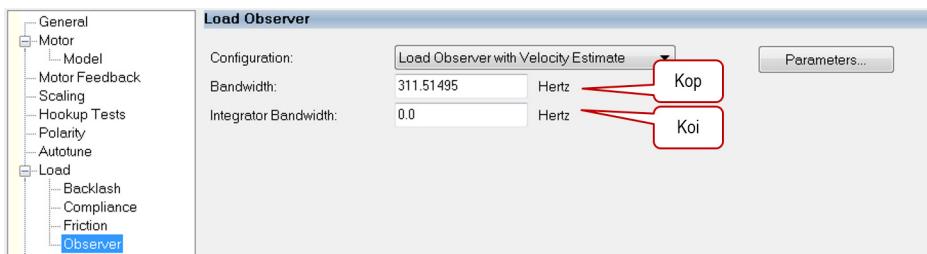
For more information on how to tune the load observer feature, see [Out-of-Box Tuning on page 35](#) and [Autotuning on page 43](#). [Table 6](#) summarizes the primary difference between the two tuning modes.

**Table 6 - Sercos Load Observer Tuning Mode Differences**

Tuning Mode	Description (Rad/s)
Out-of-box or unknown load Load Inertia Ratio = 0	Load Observer Bandwidth ( $K_{op}$ ) = 4 x Velocity Proportional Gain ( $K_{vp}$ )
Autotuning or known load Load Inertia Ratio > 0	Load Observer Bandwidth ( $K_{op}$ ) = Velocity Proportional Gain ( $K_{vp}$ )

## EtherNet/IP Drive Configuration

This section applies to the load observer feature in Kinetix 5300, Kinetix 5500, Kinetix 5700, and Kinetix 6500 drives. The load observer configuration is greatly simplified in EtherNet/IP drives compared to Sercos drives because it is natively supported by an interface in the Logix Designer application. Click the Observer tab in the Axis Properties dialog box. Here, the Load Observer mode can be selected with the Configuration pull-down menu. See [Table 2 on page 17](#) for descriptions of each setting. If load observer is enabled, the recommended Configuration setting is Load Observer with Velocity Estimate for positioning applications. Access to Load Observer Bandwidth ( $K_{op}$ ) and Load Observer Integral Bandwidth ( $K_{oi}$ ) is also shown. Typically,  $K_{oi} = 0$ .

**Figure 22 - EtherNet/IP Load Observer Configuration**

Gains are limited to 500 Hz in Kinetix 6500 drive firmware revision 2.16 and earlier. In Kinetix 6500 drive firmware revision 2.17 and later, the gain limits are increased to 10,430 Hz.

For more information on how to tune the load observer, see [Out-of-Box Tuning on page 35](#) and [Autotuning on page 43](#). [Table 7](#) summarizes the primary difference between the two tuning modes.

**Table 7 - EtherNet/IP Load Observer Tuning Mode Differences**

Tuning Mode	Description (Hz)
Out-of-box or unknown load Load Ratio = 0	Load Observer Bandwidth ( $K_{OP}$ ) = 4 x Velocity Loop Bandwidth ( $K_{VP}$ )
Autotuning or known load Load Ratio > 0	Load Observer Bandwidth ( $K_{OP}$ ) = Velocity Loop Bandwidth ( $K_{VP}$ )

## Adaptive Tuning Feature

This section applies to the adaptive tuning feature in Kinetix 5300, Kinetix 5500 and Kinetix 5700 drives. The adaptive tuning feature is an algorithm inside the drive that continuously monitors and, if necessary, adjusts or adapts the various filter parameters and control loop gains to compensate for the unknown and changing load conditions while the drive is running. The primary functions of adaptive tuning are:

- Monitor motor side resonances
- Automatically adjust torque loop notch and low pass filter parameters to suppress resonances
- Automatically de-tune control loop gains to avoid instability when it is detected

This section includes:

- [Benefits](#)
- [Resonances](#)
- [How It Works on page 17](#)

## Benefits

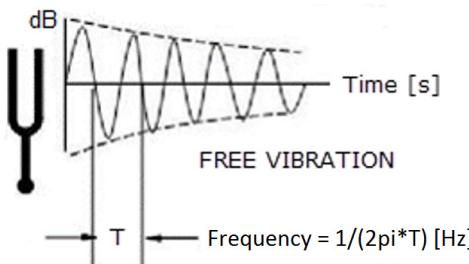
When adaptive tuning is enabled with recommended out-of-box control loop settings, adaptive tuning does the following:

- Automatically suppresses continuously changing resonances
- Eliminates the need for periodic identification of resonances and retuning
- Eliminates the need for a tuning expert
- Eliminates the need for periodic retuning of filters over time
- Reduces commissioning time, especially for high drive count
- Minimizes tracking error, machine vibration, and power consumption
- Allows bandwidth and line speeds to be increased
- Provides tighter control of moving parts, which reduces wear and saves on material costs

## Resonances

Resonance is the tendency for a system output to oscillate with greater magnitude at some frequencies than at others. Resonant frequencies are frequencies where the magnitude is at a relative maximum. Here, the system is sensitive to input and the gain is very high. Resonances are difficult to control because even small input forces can produce large magnitude oscillations in the output.

**Figure 23 - Tuning Fork Resonance**



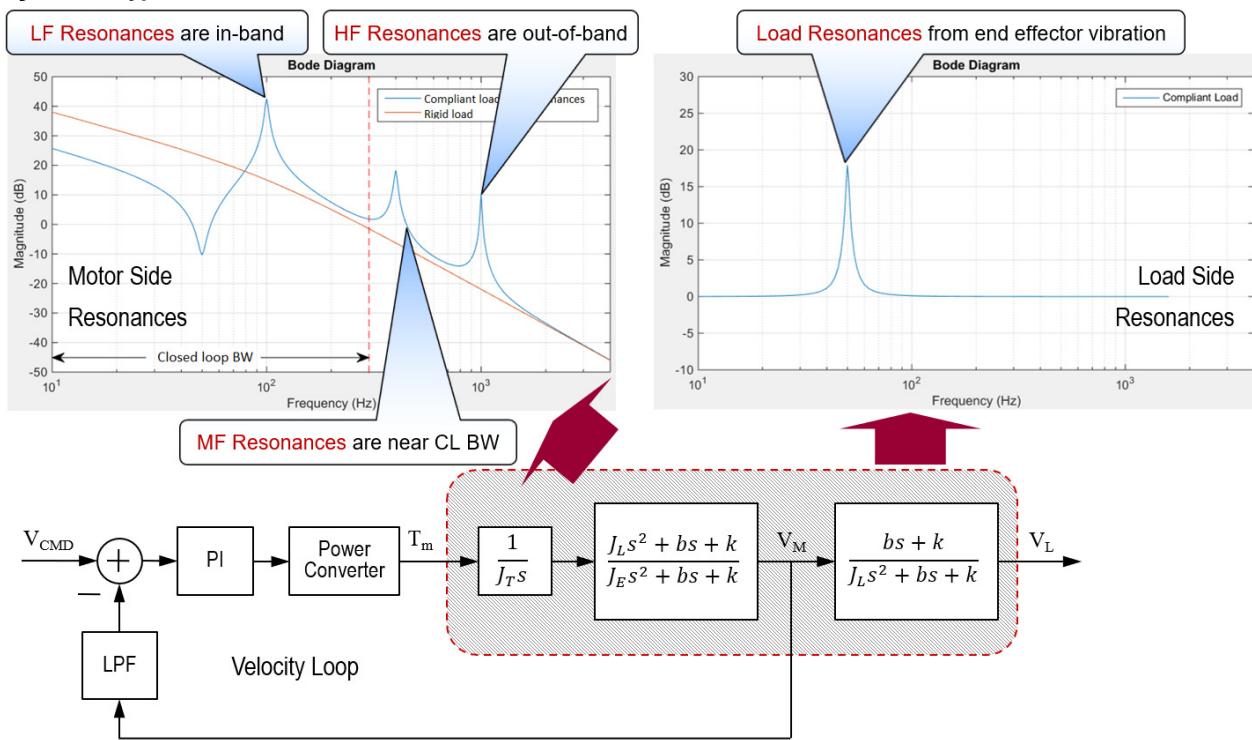
Anti-resonant frequencies are frequencies where the magnitude is at a relative minimum. Here, the system does not respond well to input and the gain is very low. Anti-resonances are difficult to control because even large input forces can produce little or no response in the output.

Compliance and backlash create resonances. Increasing the load inertia makes these problems worse. As a result, mechanical loads exhibit resonances that limit performance, damage hardware, consume energy, and generate noise. It is often left to the user to suppress these

resonances through manual tuning, a challenging and time consuming task. Resonances typically increase in number and severity as controller gains are increased. They range in frequency from a few Hz to a few thousand Hz. A typical approach is to reduce regulator gains to a low enough level so that the system bandwidth falls below the lowest resonant frequency. However, this approach results in limited performance.

There are two main types of resonances: motor side resonances and load side resonances. Motor side resonances are further broken down into three subtypes based on their frequency range: Low Frequency (LF) resonances, Mid Frequency (MF) resonances, and High Frequency (HF) resonances. [Figure 24](#) shows the types of resonances.

**Figure 24 - Types of Resonances**



Each type of resonance is handled uniquely during tuning and is described in the following sections:

- [Motor Side Resonances](#)
- [Load Side Resonances](#)

### Motor Side Resonances

Motor side resonances are resonances that are observable in the feedback signal and are detected by the feedback device on the motor. As a result, they are suppressed by tuning control loop gains, load observer gains, and torque loop filter parameters. Most mechanical resonances are motor side and most drive applications exhibit them. Furthermore, all applications ultimately exhibit resonance if gains are increased enough. Motor side resonances are categorized into three types, based on their frequency range.

**Low Frequency (LF) Resonances** – LF resonances have resonant frequencies below the Torque Loop Bandwidth. Since they are within the closed loop bandwidth (in-band), they are automatically suppressed when the load observer is applied with the recommended out-of-box settings. See [Load Observer Feature on page 16](#) for more information.

Otherwise, they can cause classical instability that generates an audible low-pitch growling noise. This instability requires detuning the control loop gains.

**High Frequency (HF) Resonances** – HF resonances have resonant frequencies above the Torque Loop Bandwidth. They typically generate an audible high-pitch squealing noise. Since HF resonances are outside the closed loop bandwidth (out-of-band), they are suppressed using

torque loop filters. Adaptive tuning addresses these resonances by automatically configuring these filters. See [Adaptive Tuning Feature on page 22](#) for more information.

Otherwise, it is left to the user to identify the audible frequencies and manually configure torque notch filters. If there are more HF resonances than there are notch filters available, a torque low pass filter is applied to suppress resonances with the highest resonant frequencies. A last resort is detuning control loop gains and load observer gains until the resonances vanish.

**Mid Frequency (MF) Resonances** – MF resonances are resonant frequencies near the Torque Loop Bandwidth. Here, torque loop filters are applied at frequencies close to the closed loop bandwidth. This close proximity allows phase lag generated by these filters to interfere with the closed loop dynamics and cause instability. Adaptive tuning addresses the issue by automatically narrowing torque notch filter widths as resonances approach the closed loop bandwidth. If instability occurs, adaptive tuning Gain Stabilization automatically decreases control loop gains and load observer gains to restore stability. See [Adaptive Tuning Feature on page 22](#) for more information.

Otherwise, it is left to the user to manually reduce notch filter widths, decrease control loop gains, and decrease load observer gains. Similarly, torque loop low pass filters impact stability when they are applied at frequencies as low as 3...5 times the closed loop bandwidth because they generate more phase lag than notch filters. As a result, only apply low pass filters if you run out of notch filters, and only reserve them for resonances with the highest resonant frequencies.

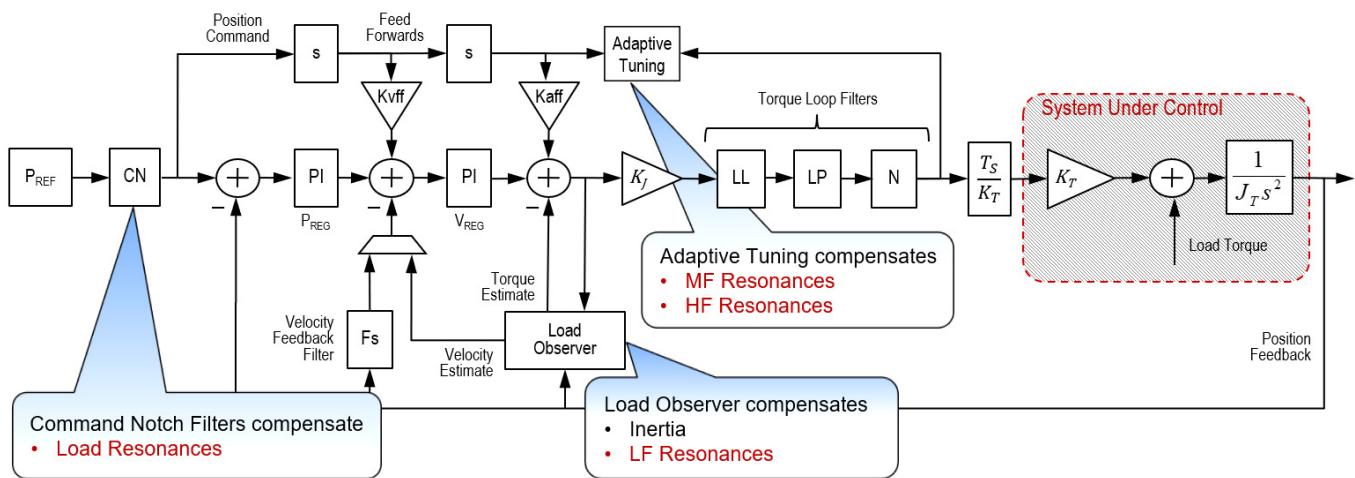
### *Load Side Resonances*

Even when all motor side resonances are suppressed and the motor shaft is tightly controlled using closed loop feedback, the load end effector can still oscillate at a few Hertz through a compliant mechanical connection or linkage to the motor. These oscillations are load side resonances that are unobservable in the feedback signal and are not measurable by the feedback device on the motor. Load side resonances result in end effector vibration that is common in robots, cranes, cantilevered loads, anti-sway, liquid sloshing, laser cutting, and material handling applications.

Load side resonance suppression requires one of the following techniques, or a combination of them:

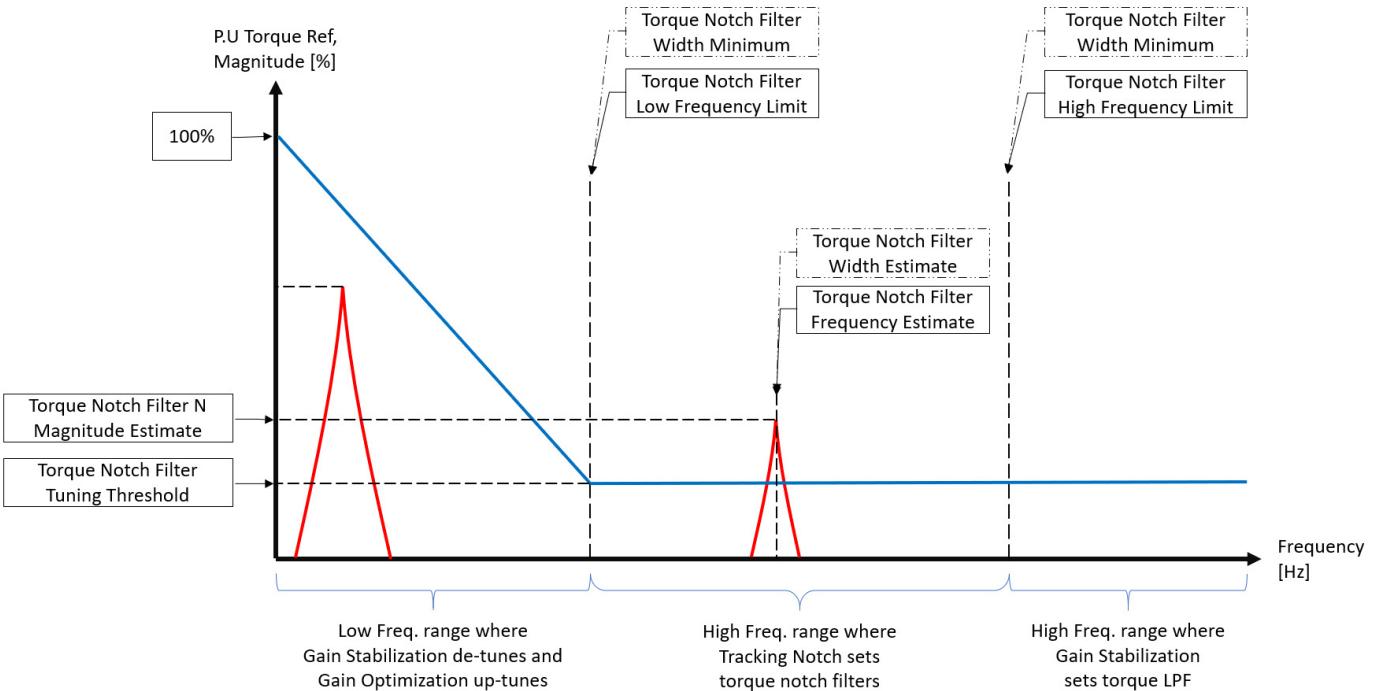
- Determine the load oscillation frequency with a stopwatch and apply a command notch filter at that frequency. Then select a smooth reference move profile. This technique is recommended, and is the simplest to implement.
- Determine the load oscillation frequency with a stopwatch and generate a smooth reference CAM profile that does not contain any load oscillation frequency content.
- Place a feedback device on the load and set the drive to load side feedback mode

[Figure 25](#) shows which features are used to control various resonances.

**Figure 25 - Controlling Resonance**

## How It Works

Adaptive tuning is always running in the background to detect motor side resonances. Every second, it analyzes the frequency response of torque loop signals to identify, track, and measure resonances. It also analyzes the frequency response of command signals to make sure dominant command frequencies are not mistaken for resonances. This is known as command rejection. The action taken to adaptively change tuning parameters largely depends on the adaptive tuning mode of operation. This approach is covered in [Configuration on page 26](#). An overview of how adaptive tuning works is shown in [Figure 26](#).

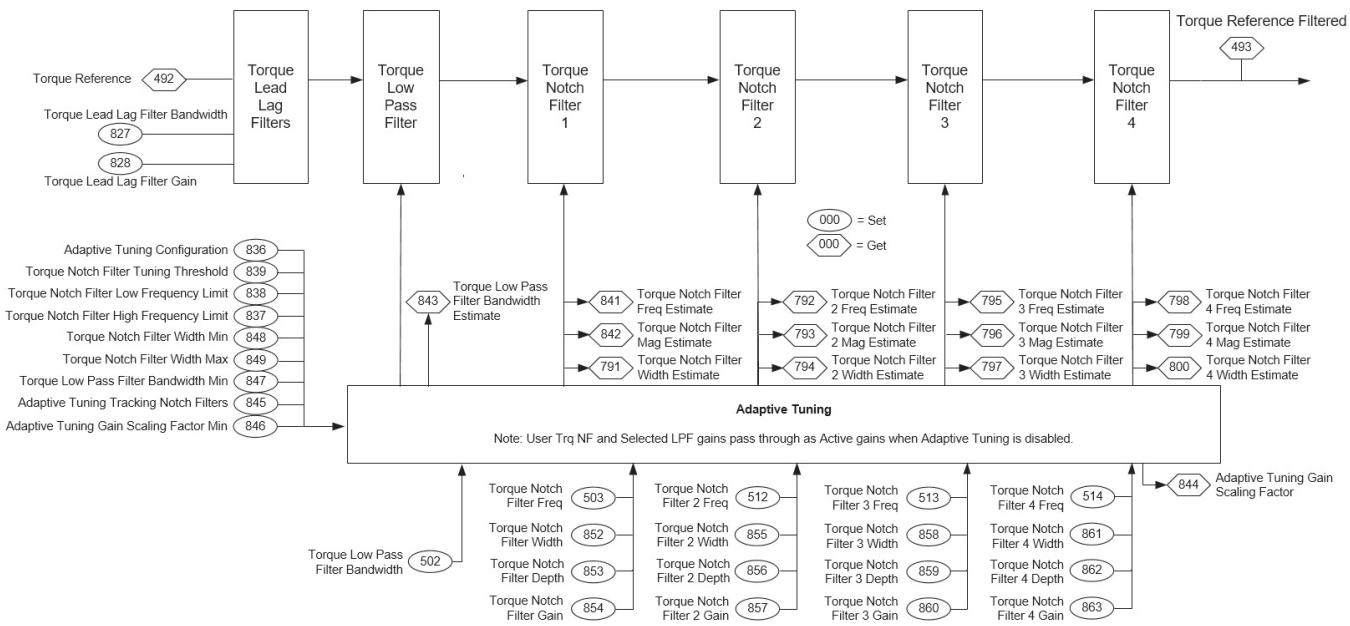
**Figure 26 - Adaptive Tuning Overview**

When the adaptive tuning feature is enabled, the drive will persist tuned values through a drive power cycle, network disconnect/reconnect, or controller download. The tuned values will not persist if the system inertia is changed by 10% or if the drive is power cycled within two minutes after an adaptive tuning update. Tuned values will be reset to user configured values if adaptive tuning is disabled.

## Configuration

[Figure 27](#) shows parameters that are relevant to adaptive tuning, preceded by their corresponding CIP Motion attribute numbers.

**Figure 27 - Adaptive Tuning Parameters**



All torque filter parameters go through adaptive tuning so that adaptive tuning can take control of these filters when required. Parameter settings are summarized in [Table 8](#), along with descriptions of how they work in various modes of operation.

**Table 8 - Adaptive Tuning Settings**

Parameter Name	Description	Default Value <sup>(1)</sup>	Range/Units
Adaptive Tuning Configuration	Controls the Adaptive Tuning feature mode of operation.	1 = Tracking Notch	0 = Disabled 1 = Tracking Notch 2 = Gain Stabilization 3 = Tracking Notch and Gain Stabilization
Adaptive Tuning Tracking Notch Filters <sup>(2)</sup>	The number of torque notch filters that are automatically tuned by Adaptive Tuning.	4	0...4 tracking notch filters
Torque Notch Filter High Frequency Limit	Adaptive Tuning identifies resonances that are not associated with the command between these low and high frequency limits with magnitudes above this tuning threshold.	4000	1...4000 Hz
Torque Notch Filter Low Frequency Limit		$4 * Z^2 * \text{Velocity Loop BW}$	1...2000 Hz
Torque Notch Filter Tuning Threshold		5	0...100% of motor rated torque
Torque Notch Filter N Frequency	Center frequency of each notch filter that is applied to the torque reference signal.	0	0...10000 Hz
Torque Notch Filter Width Minimum	The adaptive tuning Tracking Notch function adjusts torque notch filter widths proportional to the frequency estimate in relation to the high and low frequency limits. It sets torque notch filter widths equal to this minimum width when the frequency estimate is equal to the low frequency limit. It sets torque notch filter widths equal to this maximum width when the frequency estimate is equal to the high frequency limit.	0.707	0...max REAL
Torque Notch Filter Width Maximum			
Torque Low Pass Filter Bandwidth			
	Break frequency for the second order low pass filter that is applied to the torque reference signal.	5 * Load Observer Bandwidth or 5 * Velocity Loop Bandwidth when Load Observer is disabled	0...10000 Hz

**Table 8 - Adaptive Tuning Settings (Continued)**

Parameter Name	Description	Default Value <sup>(1)</sup>	Range/Units
Torque Low Pass Filter Bandwidth Min	Sets the lower limit on the Torque Low Pass Filter Bandwidth Estimate when the Adaptive Tuning function is active. Limiting the Torque Low Pass Filter Bandwidth Estimate reduces the potential impact that the filter's phase lag can have on control stability.	$2 * Z^2 * \text{Velocity Loop BW}$	1...10000 Hz
Adaptive Tuning Gain Scaling Factor	In modes with Gain Stabilization, adaptive tuning incrementally decreases this gain scaling factor from its default value to stabilize the system if required. The instability is caused from resonances that are not already suppressed by filters or it is caused by filter bandwidths that are too close to the closed loop bandwidth.	1	0...1
Adaptive Tuning Gain Scaling Factor Min	Sets the lower limit on the Adaptive Tuning Gain Scaling Factor when the Adaptive Tuning function is active. This value can be set to prevent the Adaptive Tuning algorithm from decreasing the bandwidth of the control loop to the point where there is insufficient torque to effectively control the load.	0.1	0...1

(1) Some of the default values differ depending on the drive type and firmware version that is used for the application.

(2) The option for the number of tracking notch filters differs depending on the drive type and firmware version that is used for the application.

Output parameters and their descriptions are summarized in [Table 9](#).**Table 9 - Output Parameters**

Parameter Name	Description	Default Value <sup>(1)</sup>
Torque Notch Filter N Frequency Estimate	Adaptive tuning sets this frequency estimate equal to the center frequency estimate of the current mechanical resonance that is identified by adaptive tuning. If multiple resonances exist with magnitudes above the threshold, then the one with the largest magnitude is selected. If two resonances with equal magnitude exist, then the one with the lowest center frequency is selected.	0
Torque Notch Filter N Magnitude Estimate	Adaptive tuning sets this magnitude estimate equal to the magnitude estimate of the current resonance that is identified by adaptive tuning. Adaptive tuning identifies resonances and estimates their magnitudes and widths.	0
Torque Notch Filter N Width Estimate	Adaptive tuning determines the optimal width of the Torque Notch Filter N. The Adaptive Tuning function adjusts the notch filter width to limit phase lag introduced by the filter while maximizing the range of attenuated frequencies.	0.707
Torque Low Pass Filter Bandwidth Estimate	In modes with Gain Stabilization, adaptive tuning decreases this bandwidth estimate from its default value in 200 Hz increments to suppress additional resonances above the low frequency limit if required. Additional resonances are ones that are not already suppressed by notch filters.	Torque low pass filter BW or 1500 when disabled

(1) Some of the default values differ depending on the drive type and firmware version that is used for the application.

Adaptive tuning status bits and their descriptions are given in [Table 10](#).**Table 10 - Adaptive Tuning Status Bits**

Bit	Name	Description
0	Torque Notch Filter Frequency Detected	This bit is set when the Adaptive Tuning function has detected a resonance frequency between the Torque Notch Filter Low Frequency Limit and the Torque Notch Filter High Frequency Limit with magnitude above the Toque Notch Filter Tuning Threshold. Otherwise, the bit is clear. This bit is cleared by the Adaptive Tuning function when the Axis state transitions to the Running state.
1	Torque Notch Filter Tuning Unsuccessful	When the Adaptive Tuning Configuration is set to Tracking Notch or Tracking Notch and Gain Stabilization, this bit is set when n+1 successive updates of the tunable Torque Notch Filters have failed to eliminate all resonances between the Torque Notch Filter Low Frequency Limit and the Torque Notch Filter High Frequency Limit with magnitude above the Toque Notch Filter Tuning Threshold. Here "n" is defined as the configured Track Notch Filters attribute value. This condition can occur when there are more resonances in the system than there are tunable Torque Notch Filters. Otherwise, the bit is clear. When this bit is set, and the Adaptive Tuning Configuration is set to Tracking Notch and Gain Stabilization, the Adaptive Tuning function decreases the Torque Low Pass Filter Bandwidth to attenuate the remaining resonances. This bit is cleared by the Adaptive Tuning function when the Axis state transitions to the Running state or, when in the Running state, the Adaptive Tuning Configuration transitions from Disabled to one of the Torque Notch Filter Tuning enumerations.
2	Torque Notch Filter Multiple Frequencies	This bit is set when, the Adaptive Tuning function, identifies multiple resonant frequencies that are between the Torque Notch Filter Low Frequency Limit and the Torque Notch Filter High Frequency Limit whose magnitudes are above the Toque Notch Filter Tuning Threshold. Otherwise, the bit is clear. This bit is cleared by the Adaptive Tuning function when the Axis state transitions to the Running state.

**Table 10 - Adaptive Tuning Status Bits (Continued)**

Bit	Name	Description
3	Torque Notch Filter Frequency Below Limit	This bit is set when the Adaptive Tuning function identifies a frequency that is below the Torque Notch Filter Low Frequency Limit but whose magnitude is higher than the configured Torque Notch Filter Tuning Threshold. Otherwise, the bit is clear. This bit is cleared by the Adaptive Tuning function when the Axis state transitions to the Running state.
4	Torque Notch Filter Frequency Above Limit	This bit is set when the Adaptive Tuning function identifies a frequency that is above the Torque Notch Filter High Frequency Limit but whose magnitude is higher than the configured Torque Notch Filter Tuning Threshold. Otherwise, the bit is clear. This bit is cleared by the Adaptive Tuning function when the Axis state transitions from disabled to enabled operation.
5	Adaptive Tune Gain Stabilization Active	This bit is set when the Adaptive Tuning Gain Scaling Factor is not equal to one. This indicates that the Adaptive Tuning function is actively adjusting servo loop gain values and the Torque Low Pass Filter Bandwidth value to improve system stability. This bit is cleared by the Adaptive Tuning function when the Axis state transitions to the Running state.

Adaptive tuning status bits let you create custom logic, trap errors, debug, or manually react to changes. This custom logic is useful for condition monitoring, diagnostics, and preventative maintenance purposes.

[Table 11](#) describes when output parameters get reset to their default values.

**Table 11 - Adaptive Tuning Reset Behavior**

Parameter Name	When Reset to Default Value
Torque Notch Filter Frequency Estimate	Transition to Disabled or Gain Stabilization modes
Torque Notch Filter Magnitude Estimate	A resonance is not identified
Torque Low Pass Filter Bandwidth Estimate	Transition to Disabled or Tracking Notch Filter modes
Adaptive Tuning Gain Scaling Factor	Transition to Disabled or Tracking Notch Filter modes

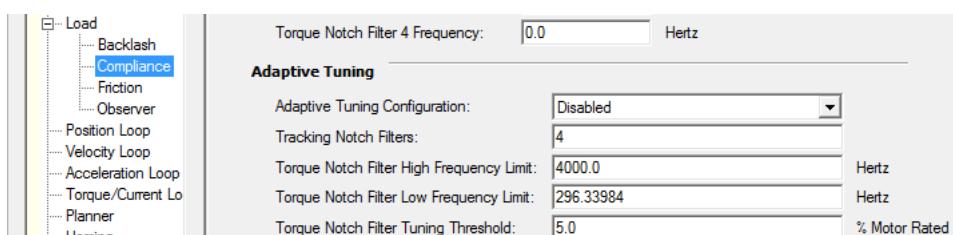
## Adaptive Tuning Modes

The following sections describe each adaptive tuning configuration mode in detail. The available configuration options differ depending on the controller and drive type you chose for your project. Resonances are characterized in the following ways:

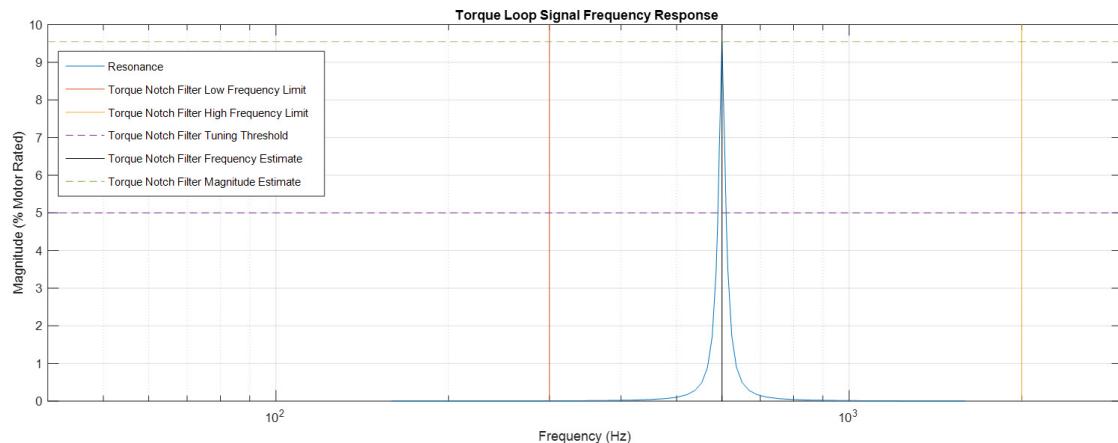
- High frequency (HF) resonances are above the low frequency limit.
- Low frequency (LF) resonances are below the low frequency limit.
- Mid frequency (MF) resonances are HF resonances that are slightly above the low frequency limit.

### Disabled

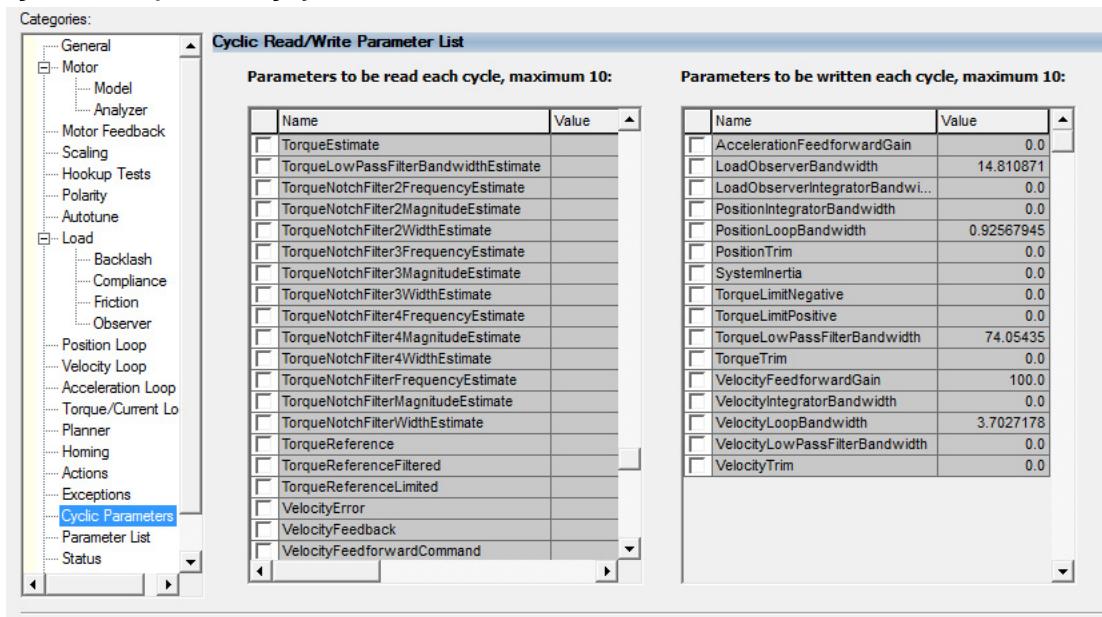
Adaptive tuning is always running in the background to identify motor side resonances, even when the feature is disabled.

**Figure 28 - Adaptive Tuning Disabled**

However, no action is taken to compensate for the identified resonances in this mode. The result is status only, which allows you to react to changes manually or with custom logic. This mode is useful for condition monitoring, diagnostics, and preventative maintenance purposes in tracking HF resonances that change over time. To illustrate the idea, [Figure 29](#) shows the frequency response of an identified resonance. Note that frequency response graphs like this are not available in the Logix Designer application; however, adaptive tuning does show the magnitude and frequency estimates.

**Figure 29 - Identifying One HF Resonance**

In Disabled mode, the high frequency limit, low frequency limit, and tuning threshold are read only and cannot be configured. As a result, you will have to temporarily enable adaptive tuning to change these settings. The adaptive tuning output parameters can be monitored in the Drive Parameters tab of the Axis Properties dialog box.

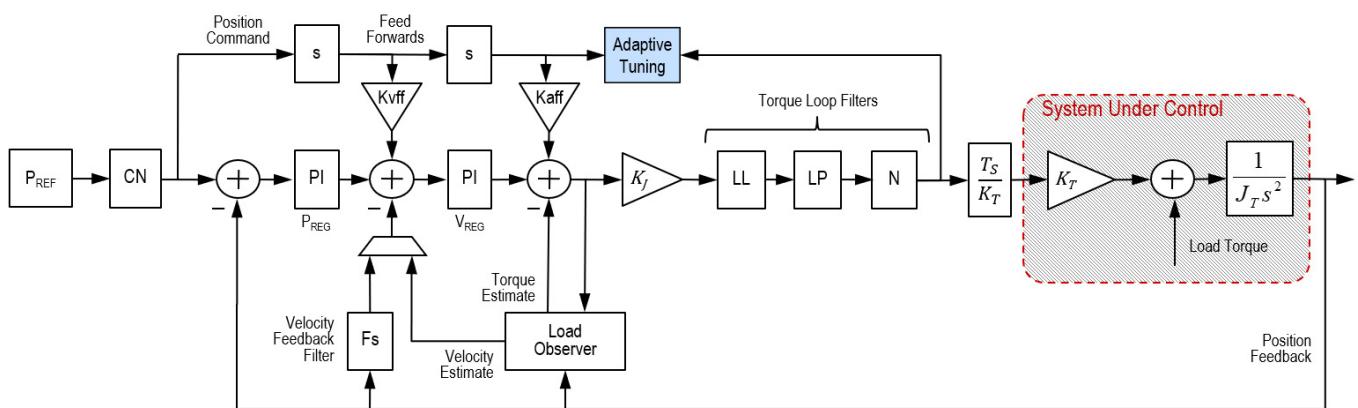
**Figure 30 - Adaptive Tuning Cyclic Reads**

### Tracking Notch

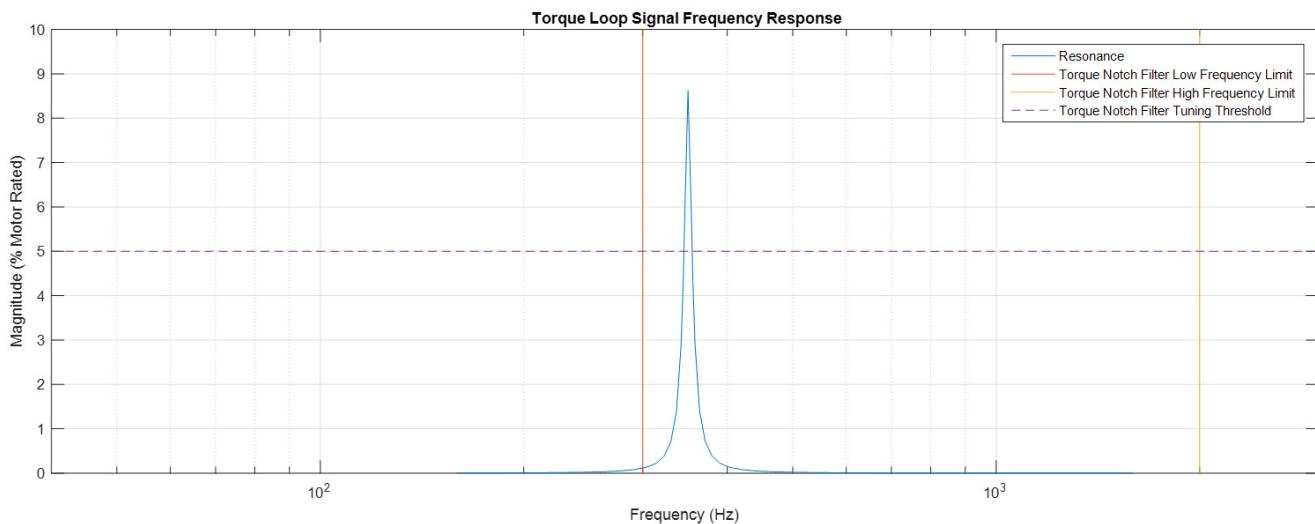
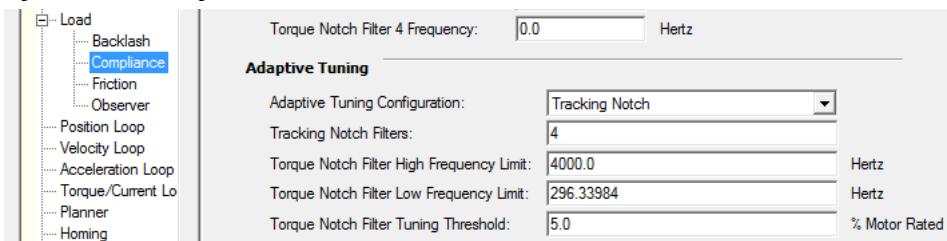
The adaptive tuning Tracking Notch function consists of four steps:

1. It applies the torque notch filter frequency estimate to the actual torque notch filter 1 frequency instead of the user entered value to suppress resonances if any are identified above the low frequency limit.

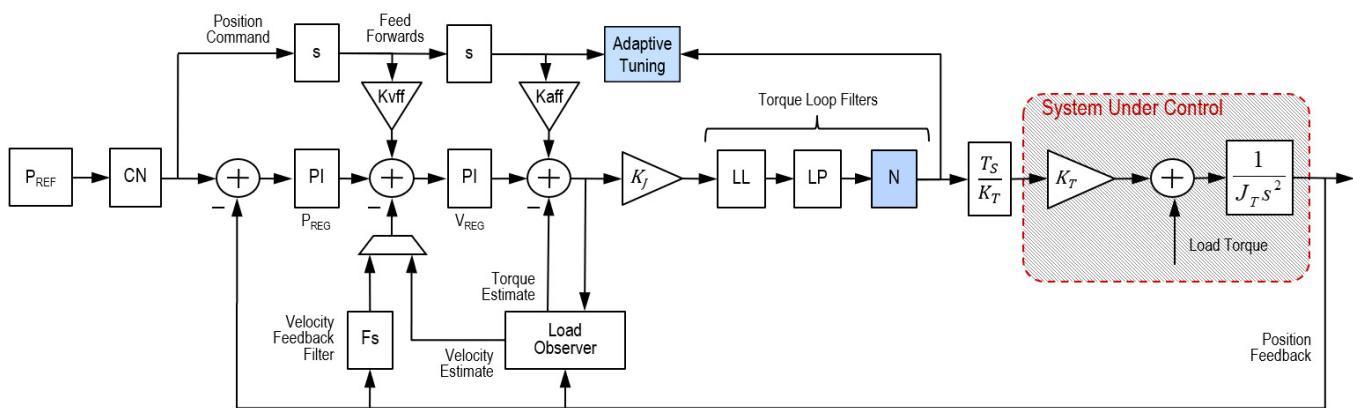
Parts of the control loop structure affected by this mode are highlighted in [Figure 31](#).

**Figure 31 - Tracking Notch Filter Configuration**

2. It sets the active torque notch filter 1 depth to 0 and the active torque notch filter 1 gain to 1. These settings help to ensure that the adaptive tuning Tracking Notch function configures the notch filter as a notch filter during operation regardless of the user entered settings.
3. It applies the torque notch filter width estimate to the active torque notch filter 1 Width. This allows tracking notch filters to suppress MF resonances without causing instability by adjusting the width in relation to the torque notch filter width high and low limits. [Figure 32](#) shows an MF resonance where the torque notch filter width estimate is slightly higher than the torque notch filter low limit. The default value of the low limit is small, which produces a narrow notch with little phase lag to interfere with stability.
4. It repeats [step 1](#) through [step 3](#) to set torque filters 2...4, depending on the value set by the Tracking Notch Filters tag and if more resonances are identified.

**Figure 32 - Identifying One MF Resonance****Figure 33 - Tracking Notch Filter Mode**

In [Figure 34](#), the parts of the control loop structure affected by Tracking Notch Filters are highlighted in blue.

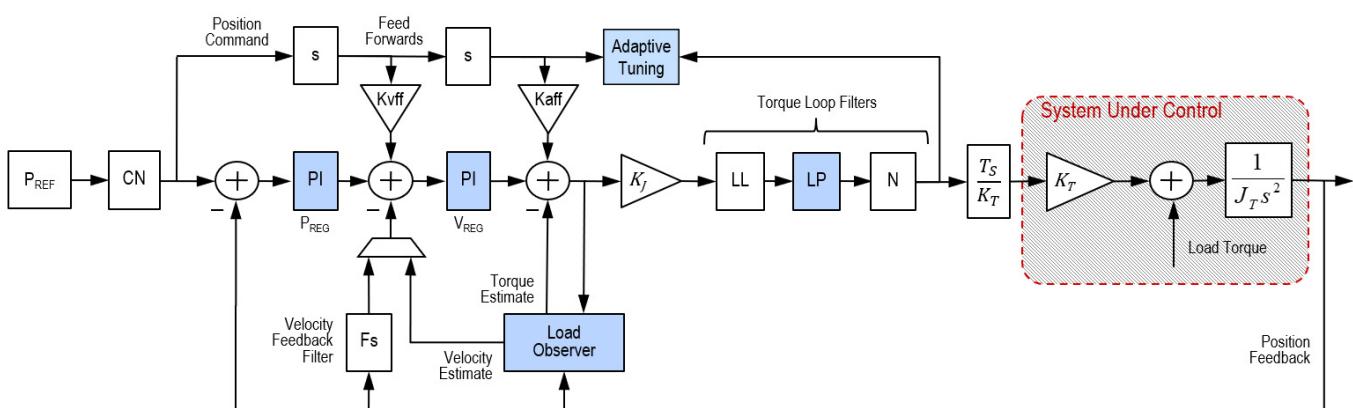
**Figure 34 - Tracking Notch Filter Configuration**

### Gain Stabilization

When a LF resonance is detected with a magnitude greater than the threshold line, see [Figure 26 on page 25](#), Gain stabilization performs two main functions until the LF resonance magnitude decreases below the threshold or an instability occurs:

1. It enables and tunes the low pass filter to suppress resonances if any are identified above the low frequency limit. Here, the Torque Low Pass Filter Bandwidth Estimate is applied to the torque low pass filter instead of the Torque Low Pass Filter Bandwidth. The bandwidth estimate is incrementally decreased from its default value until the identified HF resonances are suppressed or until an LF resonance or instability occurs.
2. Adaptive tuning detunes control loop gains to suppress any remaining resonances and stabilize the system if necessary. Here, the following gains are scaled by the adaptive tuning gain scaling factor (Axis.AdaptiveTuningGainScalingFactor).
  - Load Observer Bandwidth
  - Load Observer Integrator Bandwidth
  - Velocity Loop Bandwidth
  - Velocity Loop Integrator Bandwidth
  - Position Loop Bandwidth
  - Position Loop Integrator Bandwidth

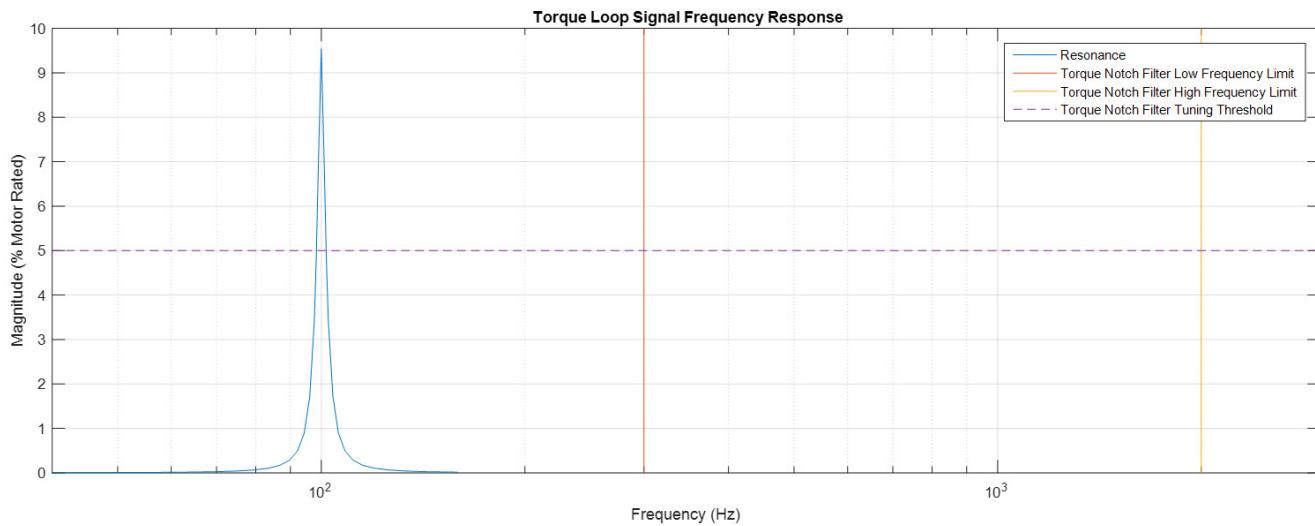
This scaling means that the actual control loop gains applied in the regulators are multiplied by the gain scaling factor internally. The scaling factor is incrementally decreased from its default value of 1 until the system is stable. When Gain stabilization is not enabled, the scaling factor is reset to its default value so that control loop gains are not affected. Parts of the control loop structure affected by this mode are highlighted in [Figure 35](#).

**Figure 35 - Gain Stabilization Configuration**

Gain stabilization is good for situations where there are more resonances than there are notch filters and for keeping the drive stable. Instability and audible noise are caused from the following situations:

- HF resonances that are not already suppressed by filters
- MF resonances that are suppressed by filters but the filter bandwidths are too close to the closed loop bandwidth
- LF resonances that result when load observer is not applied with the recommended out-of-box settings
- LF resonances that result from classical instability

**Figure 36 - Identifying One LF Resonance**

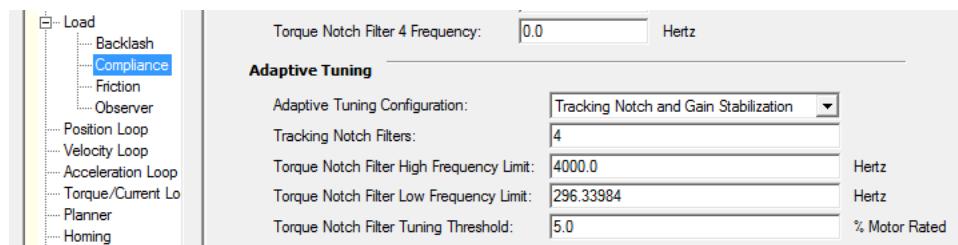


**IMPORTANT** Adaptive tuning gain stabilization is not recommended for vertical loads as detuning may cause load drops.

### Tracking Notch and Gain Stabilization

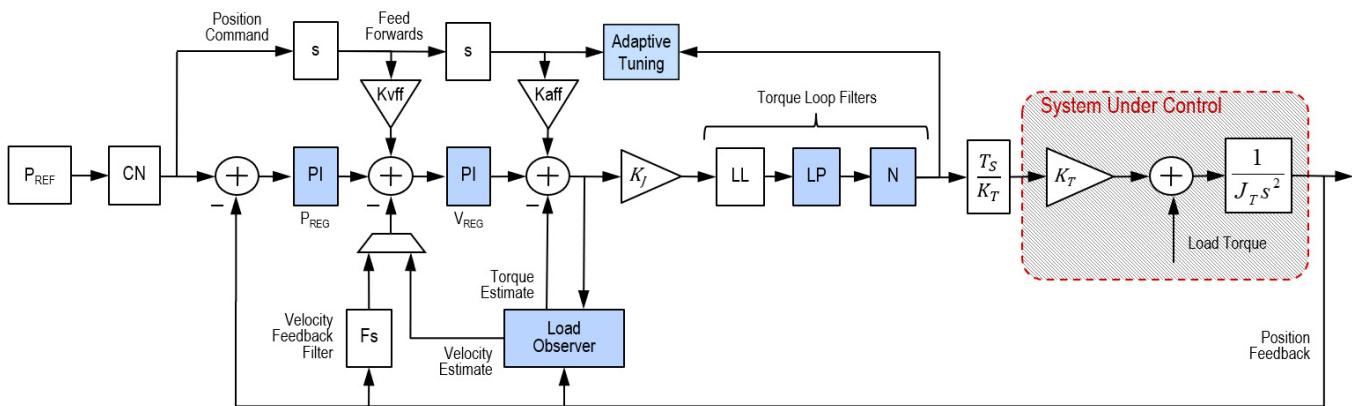
In this mode, adaptive tuning applies tracking notch filters if required, followed by gain stabilization if required.

**Figure 37 - Tracking Notch Filter and Gain Stabilization Mode**



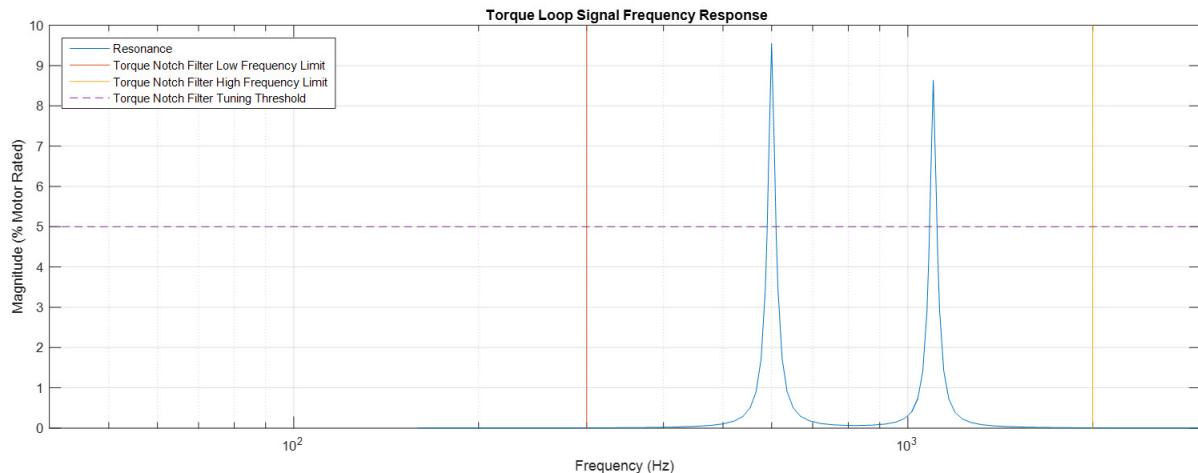
In [Figure 38](#), the parts of the control loop structure affected by adaptive tuning are highlighted in blue.

**Figure 38 - Tracking Notch Filter and Gain Stabilization Configuration**



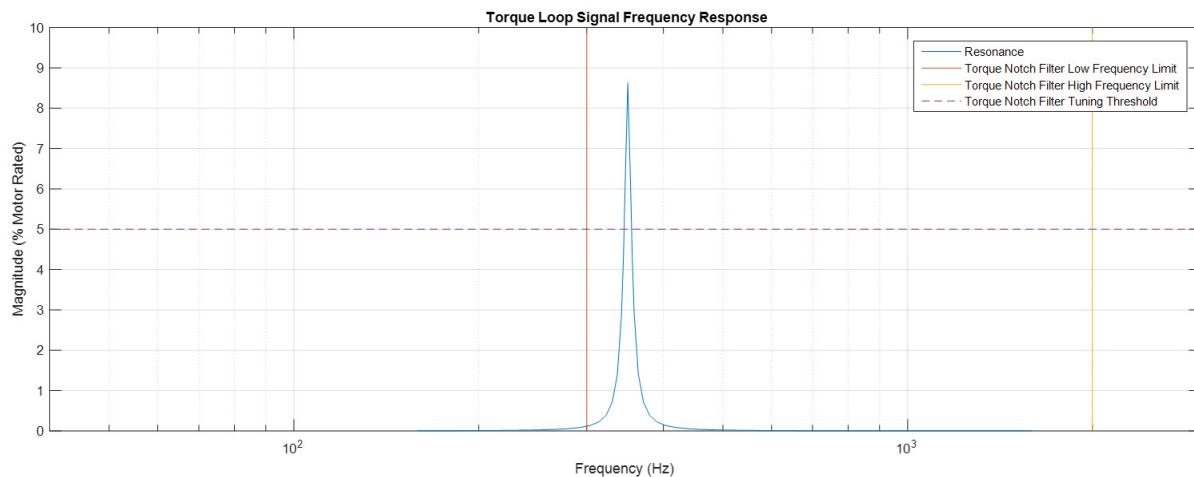
First, the Tracking Notch Filter sets the torque notch filters to suppress a HF resonance with the largest magnitude if one exists. Next, Gain Stabilization applies the low pass filter to suppress additional HF resonances if they exist. This is useful for suppressing more HF resonances than there are notch filters.

**Figure 39 - Identifying More HF Resonances than there are Notch Filters**



Finally, if the system is unstable, the Adaptive Tuning Gain Stabilization function incrementally detunes the position loop, velocity loop, and load observer gains to stabilize the system if necessary.

A good example is when a MF resonance is identified, as shown in [Figure 40](#).

**Figure 40 - Identifying One MF Resonance**

In this example, the first torque notch filter is set to suppress it if it is the only HF resonance or if it is the one with the largest magnitude. The system is detuned if one or more of the following conditions exist:

- The torque notch filter was set to suppress the MF resonance, but its width is wide enough or its frequency is close enough to the closed loop bandwidth to cause instability.
- The torque low pass filter was set to suppress the MF resonance, but its bandwidth is close enough to the closed loop bandwidth to cause instability.
- There are additional unsuppressed resonances present.

For information on parameters and status bits, see [Configuration on page 26](#).

## Out-of-Box Tuning

In this chapter, the sections on gain calculation describe how gains are currently calculated in released product when a new axis is created. However, the sections on recommended out-of-box settings proposed in this chapter should be used as a starting point when commissioning a new axis. They often yield satisfactory motion performance and no further tuning intervention is required.

Topic	Page
Sercos Drives	35
EtherNet/IP Drives	38
Is Further Tuning Required?	41

Out-of-box refers to default control loop gain settings that are pre-configured for a new axis when it is created. Since the load is unknown at this point, the motor is assumed to be unloaded and the load ratio  $R = 0$ . However, when the load is known or an autotune has been performed, the control loop gains are configured for a load ratio  $R > 0$ . This is the primary difference affecting out-of-box and autotuning rules. Thus, the phrase out-of-box implies  $R = 0$  and the term ‘autotune’ implies  $R > 0$ .

## Sercos Drives

The following subsections provide information for out-of-box tuning of Sercos drives.

### Gain Calculation

In a Sercos drive, the default damping factor is  $z = 0.8$ . This produces a loop spacing of  $4z^2 = 2.56$ . The proportional gains are calculated from the torque loop bandwidth ( $T_{BW}$ ) in units of rad/s and the integral gains are set to zero.

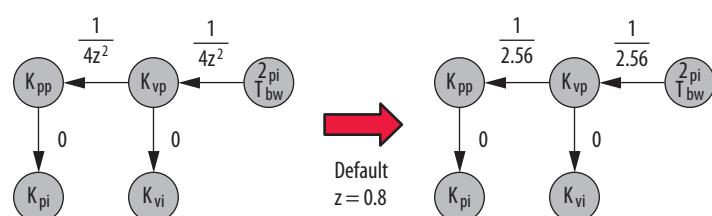
$$\text{Velocity Proportional Gain: } K_{VP} = 2\pi \times T_{BW} / (4z^2)$$

$$\text{Position Proportional Gain: } K_{PP} = K_{VP} / (4z^2)$$

$$\text{Velocity Integral Gain: } K_{VI} = 0$$

$$\text{Position Integral Gain: } K_{PI} = 0$$

**Figure 41 - Sercos Drive Out-of-Box Gain Relationships**



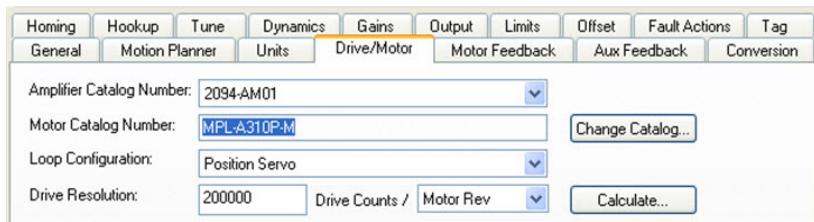
The velocity feed forward gain  $K_{VFF}$  and the acceleration feed forward gain  $K_{AFF}$  are set to zero by default as well.

## Sercos Drive Recommended Settings

This method of setting control loop gains applies to Kinetix® 6000 drives (firmware revision 1.124 or later) and Kinetix 6200 drives (firmware revision 1.049 or later). The method works for unknown loads or when an autotune is not performed and produces a relatively high level of performance in 90% of motion applications. Most of the time, there is no need to perform an autotune procedure or further optimize gain settings. Therefore, it has become the recommended out-of-box setting for Sercos drives.

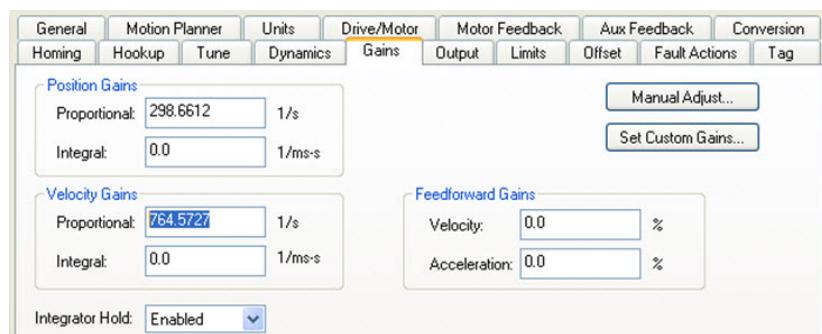
Follow these steps to configure the drive for high performance right out of the box. This procedure uses load observer to automatically account for the unknown load. As a result, you must be familiar with creating an axis in the Studio 5000 Logix Designer® application and accessing drive IDN parameters.

1. Create a new axis with type AXIS\_SERVO\_DRIVE.
2. Click the Drive/Motor tab in the Axis Properties dialog box and add a motor.

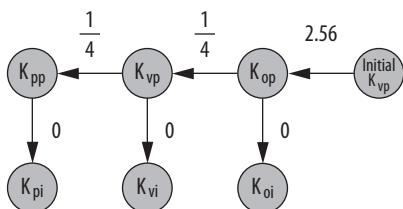


3. Click the Gains tab in the Axis Properties dialog box.

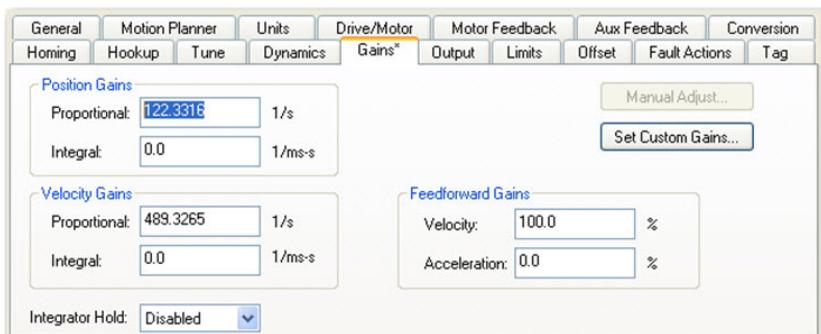
The Velocity Proportional Gain (Initial  $K_{vp}$ ) value is used to recalculate all gain values.



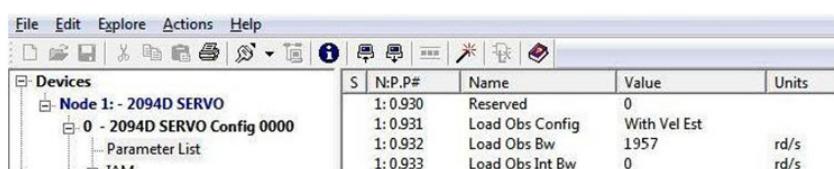
4. Make the following calculations:
  - a. Load Observer Bandwidth:  $K_{OP} = \text{Velocity Proportional Gain} \times 2.56$
  - b. Velocity Loop Bandwidth:  $K_{VP} = K_{OP} / 4$
  - c. Position Loop Bandwidth:  $K_{PP} = K_{VP} / 4$



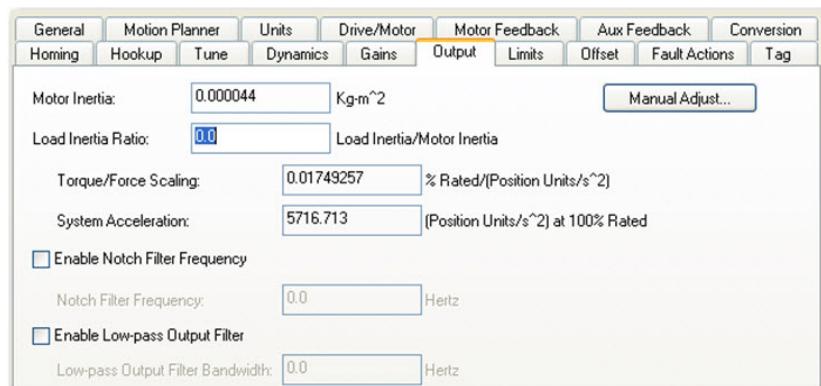
5. Configure these settings and values on the Gains tab.
  - a. Position Proportional Gain =  $K_{PP}$
  - b. Velocity Proportional Gain =  $K_{VP}$
  - c. Velocity Feedforward Gain = 100%
  - d. Integrator Hold = Disabled



6. Configure these IDN parameter values. Note that there is an offset of 500 in IDN values in the Connected Components Workbench™ software compared to Logix Designer MSG instructions.
  - a. IDN P-431 = 2 (Load Observer with Velocity Estimate)
  - b. IDN P-432 =  $K_{OP}$
  - c. IDN P-433 = 0
  - d. IDN P-065 = 1

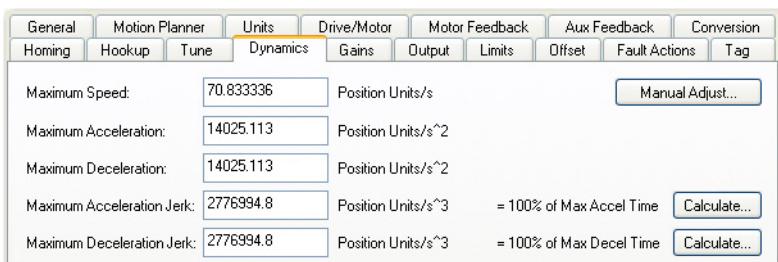


7. Click the Output tab in the Axis Properties dialog box and verify these settings.
  - a. Load Inertia Ratio = 0
  - b. Enable Low-pass Output Filter = Unchecked



8. **IMPORTANT:** Suppress resonances.
  - a. Briefly enable the axis with an MSO motion direct command followed by an MSF instruction.
  - b. If an audible noise exists at any time, use a smart phone app to identify resonant frequencies and set torque loop filters to remove them. See [Compensating for High Frequency Resonances on page 63](#) on to manually tune out resonant frequencies.
9. **IMPORTANT:** If required, reduce the Maximum Acceleration and Deceleration to meet application requirements and to protect the drive and motor from overload.

With Load Inertia Ratio = 0, acceleration limits are set to their maximum value, providing the best performance for an unloaded motor. However, the motor is loaded and may not be able to accelerate as fast. As a result, you may have to reduce the Maximum Acceleration and Deceleration to meet application requirements.



## EtherNet/IP Drives

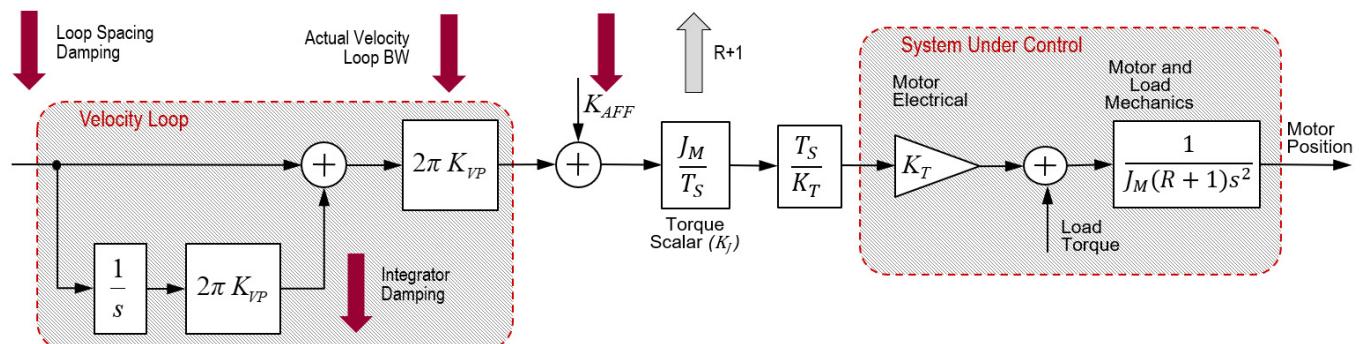
The following sections provide information for out-of-box tuning of EtherNet/IP drives.

### Gain Calculation

When the load ratio is set to zero in the Logix Designer application, but the actual load ratio R is non-zero, the torque scalar is lower than what is required by the servo system and must be increased by a factor of (R+1). This load disconnect has the following effects:

- The effectiveness of the acceleration feed forward that is applied is reduced with respect to the  $K_{AFF}$  value entered in the Logix Designer application.
- The torque scalar ‘steals’ a factor of (R+1) from  $K_{VP}$ , lowering the actual velocity loop bandwidth by a factor of (R+1) with respect to the  $K_{VP}$  value entered in the Logix Designer application. Reducing  $K_{VP}$  causes the spacing between it and  $K_{VI}$  to become smaller, which reduces the velocity loop damping associated with  $K_{VI}$ . Reducing  $K_{VP}$  also causes the loop spacing between it and  $K_{PP}$  to become smaller, which reduces the position loop damping associated with loop spacing.

**Figure 42 - Effects of Unknown Load R = 0**



To increase damping when  $K_{VP}$  is artificially lowered, all the other gains are decreased by an extra factor of 10. This provides satisfactory motion performance over a large class of unknown loads. In an EtherNet/IP drive, the default damping factor is  $z = 1.0$ . This produces a loop spacing of  $4z^2 = 4$ . The gains are calculated from the torque loop bandwidth ( $T_{BW}$ ) in units of Hz.

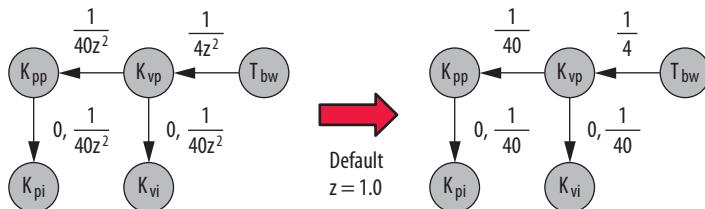
$$\text{Velocity Loop Bandwidth: } K_{VP} = T_{BW} / (4z^2)$$

$$\text{Position Loop Bandwidth: } K_{PP} = K_{VP} / (40z^2)$$

$$\text{Velocity Integral Bandwidth: } K_{VI} = 0 \text{ or } K_{VP} / (40z^2)$$

$$\text{Position Integral Bandwidth: } K_{PI} = 0 \text{ or } K_{PP} / (40z^2)$$

**Figure 43 - EtherNet/IP Drive Out-of-Box Gain Relationships without Load Observer**



Defaults for other relevant parameters are given.

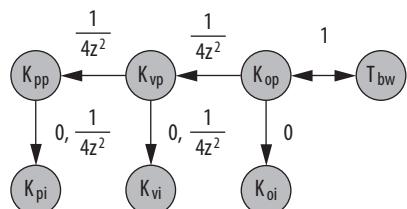
$$\text{Velocity Feedforward Gain: } K_{VFF} = 0, 100$$

$$\text{Acceleration Feedforward Gain: } K_{AFF} = 0, 100$$

$$\text{Low-pass Filter Bandwidth: } LP = 5 \times K_{VP}$$

When Load Observer with Velocity Estimate is applied, it automatically accounts for the arbitrary load disconnect created by the load, forcing it to function like an unloaded motor. As a result, standard 4 spacing sufficient for controlling an unloaded motor is applied when the load observer is enabled.

**Figure 44 - EtherNet/IP Drive Out-of-Box Gain Relationships with Load Observer**



When the low-pass filter is enabled, set it to a value greater than 5 times  $K_{VP}$  or  $K_{OP}$ , whichever is larger. This prevents additional phase lag created by the low-pass filter from being introduced into the system, causing instability. When the load observer is enabled, the low-pass filter default setting is given.

$$\text{Low-pass Filter Bandwidth: } LP = 5 \times K_{OP}$$

## EtherNet/IP Drive Recommended Settings

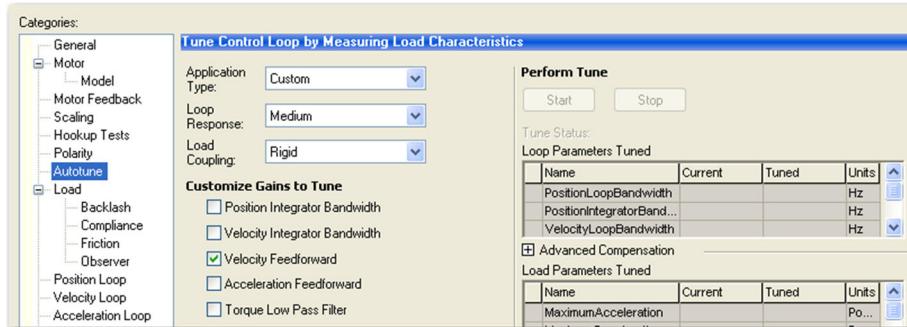
This method of setting control loop gains applies to Kinetix 5500, Kinetix 5700, and Kinetix 6500 drives. It works for unknown loads or when an autotune is not performed. It uses load observer and produces a relatively high level of performance in 90% of motion applications. Most of the time, there is no need to perform an autotune procedure or further optimize gain settings. Therefore, it has become the recommended out-of-box setting for EtherNet/IP drives.

With Logix Designer version 33 and later, these features are enabled by default for the following drives, so these configuration steps are not required:

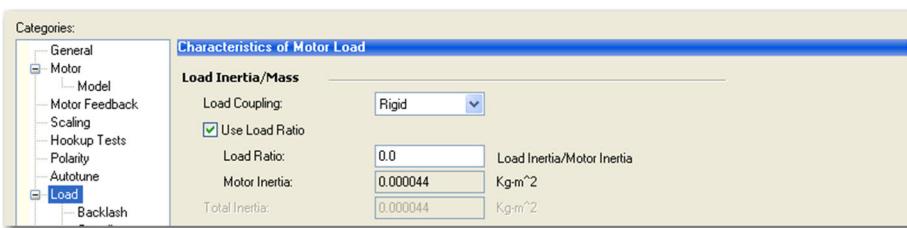
- Kinetix 5300 drives
- Kinetix 5700 ERS3 Series B drives, with firmware version 13 and later
- Kinetix 5700 ERS4 drives, with firmware version 13 and later
- Kinetix 5730 drives, with firmware version 13 and later

For drives that do not have these features enabled automatically, follow these steps to configure the drive for high performance right out of the box.

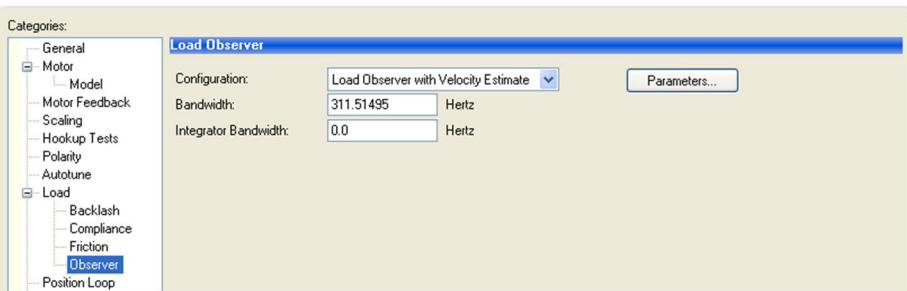
1. Click the Autotune tab in the Axis Properties dialog box.
  - a. From the pull-down menus for Application Type, Loop Response, and Load Coupling, choose Custom, Medium, and Rigid settings, respectively.
  - b. Verify that only the Velocity Feedforward box is checked.



2. Click the Load tab in the Axis Properties dialog box.
  - a. Check Use Load Ratio.
  - b. Set the Load Ratio = 0.

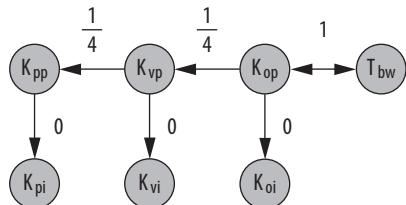


3. Click the Observer tab in the Axis Properties dialog box.
  - a. From the Configuration pull-down menu, choose Load Observer with Velocity Estimate.



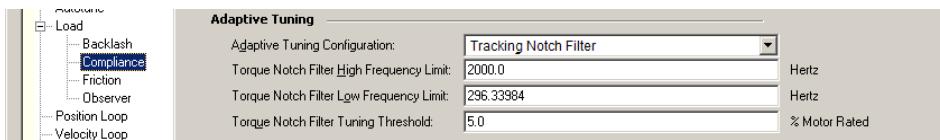
- b. Click Apply and then click Yes to update all dependent attributes.

The Load Observer Bandwidth and other gains are set automatically.



**4. IMPORTANT:** Suppress resonances.

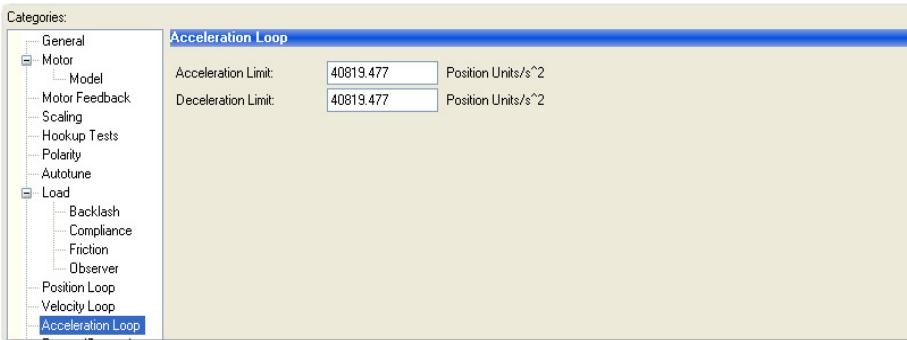
- a. For Kinetix 5300, Kinetix 5500, and Kinetix 5700 drives, choose Tracking Notch Filter from the Adaptive Tuning Configuration pull-down menu on the Compliance tab in the Axis Properties dialog box.



- b. For Kinetix 6500 drives, briefly enable the axis with an MS0 motion direct command followed by an MSF instruction. If an audible noise exists at any time, use a smart phone app to identify resonant frequencies and set torque loop filters to remove them. See [Compensating for High Frequency Resonances on page 63](#) to manually tune out resonant frequencies.

**5. IMPORTANT:** If required, reduce the Maximum Acceleration and Deceleration to meet application requirements and to protect the drive and motor from overload.

With Load Inertia Ratio = 0, acceleration limits are set to their maximum value, providing the best performance for an unloaded motor. However, the motor is loaded and may not be able to accelerate as fast. As a result, you may have to reduce the Maximum Acceleration and Deceleration to meet application requirements.



## Is Further Tuning Required?

Here are some observations that indicate the servo drive can produce satisfactory performance without additional tuning:

- Visibly smooth motion from a smooth cycle profile
- Little to no audible noise produced during and after a commanded motion
- Position and/or velocity errors are within the application requirements
- Position and/or velocity errors are repeatable
- The cycle profile is within the thermal limits of the drive and motor

If the load does not respond as intended, consider these factors before tuning:

- Investigate proper servo sizing. The best way to do this is to model the axis in Motion Analyzer software to verify the proper sizing of the motor and drive for the specified load and move profile. You can also compare various design options for gear ratio, load size, coupling configuration, high-resolution versus low-resolution feedback device, move profile types, and so forth.
- Simulate the axis in Motion Analyzer software, testing various control loop gain settings to conclude the desired motion can be achieved. Compliance and machine vibration can often be minimized by creating a more direct and stiff coupling between the motor and load. Helping to achieve this are quality couplings, gearboxes, actuators, and guides.

If the load requires additional tuning to further optimize motion performance, see [Autotuning on page 43](#) and [Manual Tuning on page 57](#).

## Autotuning

This chapter provides autotuning information for Sercos and EtherNet/IP servo drives.

Topic	Page
Sercos Drives	43
EtherNet/IP Drives	49
Is Further Tuning Required?	56

Out-of-box refers to default control loop gain settings that are pre-configured for a new axis when it is created. Because the load is unknown at this point, the motor is assumed to be unloaded and the load ratio  $R = 0$ . However, when the load is known or an autotune has been performed, the control loop gains are configured for a load ratio  $R > 0$ . This is the primary difference affecting out-of-box and autotuning rules. Thus, the phrase out-of-box implies  $R = 0$  and the term ‘autotune’ implies  $R > 0$ .

Autotune automatically performs two basic functions with minimal intervention:

- Autotune momentarily initiates motion in a ‘bump test’ to measure the load ratio ( $R$ ). The torque scalar and system acceleration are calculated from  $R$  and the axis dynamics and limits are calculated from these parameters.
- Control loop gains are calculated based on the torque-loop bandwidth ( $T_{BW}$ ), which is determined from the drive model time constant (DMTC), depending on the drive and motor selected.

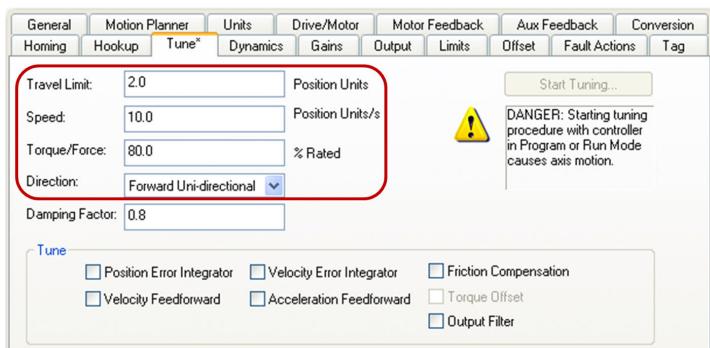
## Sercos Drives

The following subsections provide information for autotuning Sercos drives.

### Bump Test

When autotune performs the bump test, a momentary tuning torque is applied to the motor while the acceleration and deceleration times are measured. This information is then used to calculate the load ratio ( $R$ ), torque scalar, and system inertia. The top four parameters on the Tune tab of the Axis Properties dialog box are used to keep the motor and load within specified position, velocity, acceleration, and direction limits during the bump test.

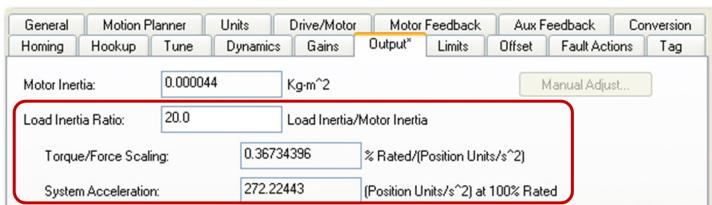
Figure 45 - Sercos Drive Autotune Bump Test Parameters



Set the Direction of the autotune test based on the application requirements. The default direction is Forward Uni-directional. Set the Travel Limit and Speed based on the physical limits dictated by the mechanics of the application. Start with Torque/Force = 50. If the bump test

fails, change the limits slightly and try again. The load ratio, torque scalar, and system acceleration are then calculated upon successful completion of the bump test.

**Figure 46 - Sercos Drive Autotune Bump Test Results**



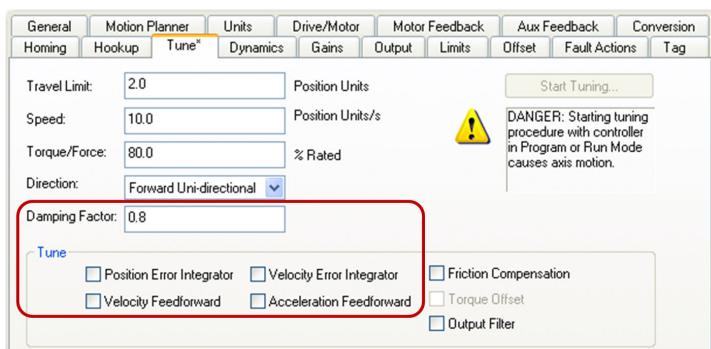
## Gain Calculation

The remaining parameters on the Tune tab of the Axis Properties dialog box are used to calculate control loop gains. The damping factor  $z = 0.8$  by default.

### Application Type

The Tune check boxes determine which integrator and feed forward gains are enabled and given a non-zero value.

**Figure 47 - Sercos Drive Autotune Gain Selection Parameters**



There are many combinations of integrator and feed forward gains to enable. [Table 12](#) shows some common applications and which integrator and feed forward gains are enabled for each.

**Table 12 - Sercos Drive Gain Selection Based on Application Type**

Application Type	Applications	K <sub>P</sub>	K <sub>V</sub>	Integrator Hold	K <sub>VFF</sub>	K <sub>AFF</sub>
Basic	Basic smooth motion				X	
Tracking	<ul style="list-style-type: none"> <li>Converting</li> <li>Printing</li> <li>Web</li> <li>Flying shear</li> <li>Coordinated motion</li> <li>Rotary knife</li> <li>Packaging</li> </ul>		X		X	X
Point to Point	<ul style="list-style-type: none"> <li>Pick and place</li> <li>Indexing</li> <li>Robotics</li> <li>Palletizing</li> </ul>	X		X		
Constant Speed	<ul style="list-style-type: none"> <li>Conveyors</li> <li>Line shafts</li> <li>Crank</li> </ul>		X		X	
Positioning	High performance position control	X			X	X

Autotune calculates the control loop gains when the controller is online and the Start Tuning button is pressed. The position and velocity integrators are calculated with non-zero values if they were selected on the Tune tab of the Axis Properties dialog box.

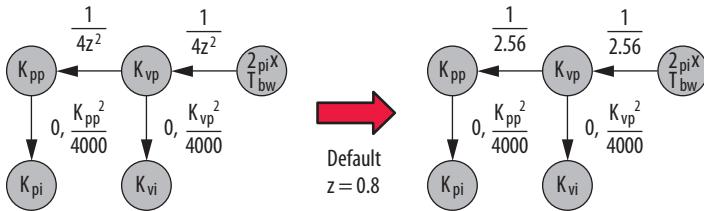
$$\text{Velocity Proportional Gain: } K_{VP} = 2\pi T_{BW} / (4z^2)$$

$$\text{Position Proportional Gain: } K_{PP} = K_{VP} / (4z^2)$$

$$\text{Velocity Integral Gain: } K_{VI} = 0, K_{PP}^2 / 4000$$

$$\text{Position Integral Gain: } K_{PI} = 0, K_{VP}^2 / 4000$$

**Figure 48 - Sercos Drive Autotune Gain Relationships**



In Sercos drives, damping factor affects loop spacing but not integrator spacing. A loop spacing of  $4z^2$  is sufficient for rigid loads but can be too hot for most loads, which are classified as compliant. As a result, you often must de-tune the control loop gains. Defaults for other relevant parameters are given.

$$\text{Velocity Feedforward Gain: } K_{VFF} = 0, 100$$

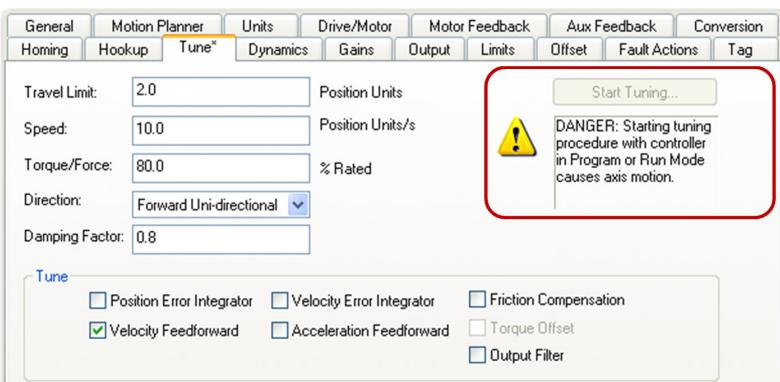
$$\text{Acceleration Feedforward Gain: } K_{AFF} = 0, 100$$

$$\text{Low-pass Filter Bandwidth: } LP = 4 \times K_{VP} / (2\pi)$$

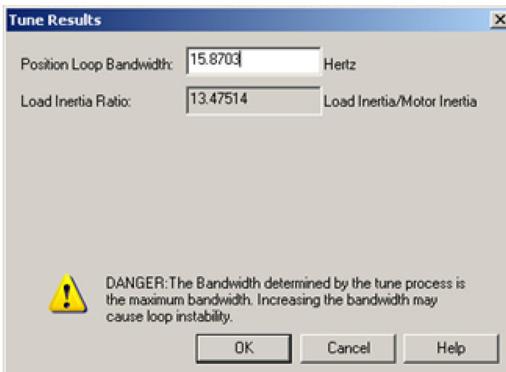
### Start Tuning

Follow these steps to execute an autotune. It may be beneficial to run a trend in conjunction with executing an autotune. See [Appendix B on page 93](#) for more information.

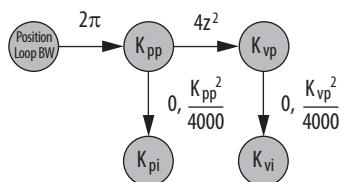
1. Click the Start Tuning button on the Tune tab in the Axis Properties dialog box. For variable inertia loads, perform an autotune at the point of lowest mechanical inertia. If you manually calculate the Load Inertia Ratio, enter the minimum load inertia on the Output tab.



After the autotune completes a bump test, a Tune Results dialog box displays the Load Inertia Ratio resulting from the bump test. The Position Loop Bandwidth =  $K_{PP} / 2\pi$  in units of Hz resulting from the gain calculation.



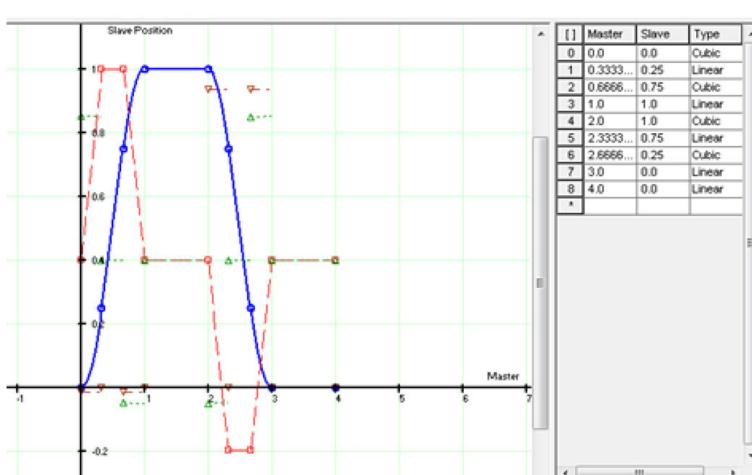
- At this point, you can increase or decrease the calculated Position Loop Bandwidth to recalculate the gains with higher or lower values.



Determine if the mechanical system attached to the motor is rigid or compliant and adjust the Position Loop Bandwidth accordingly.

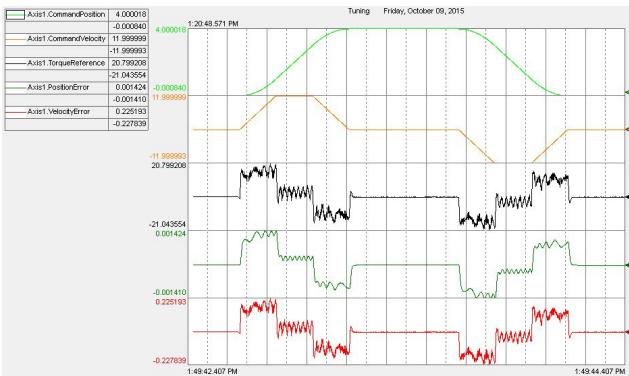
- Rigid systems typically involve high performance load mechanics that are tightly coupled directly to the motor shaft and there is no lost motion. Because the gains are auto-calculated for rigid loads, this type of load typically requires little to no adjustment of Position Loop Bandwidth to dial in the performance.
  - Everything else is compliant, including systems with belts and pulleys, long shafts, heavy loads, and couplings and gearboxes with backlash and/or lost motion. For this type of load, a good starting point is to reduce the Position Loop Bandwidth by a factor of the Load Inertia Ratio +1 to assure stability. However, performance is typically too low and you will have to incrementally increase Position Loop Bandwidth in successive autotunes to dial in the performance.
- Test the drive and observe mechanical performance and stability.
    - Create a move profile in the Studio 5000 Logix Designer® application to observe the behavior of the mechanics while tuning.

See [Appendix A on page 89](#) for more information on Creating Move Profiles for Tuning. An example CAM table for an MATC instruction is shown below.



- Create a trend in the Logix Designer application to monitor Command Position, Command Velocity, Torque Reference, Position Error, and Velocity Error.

See [Appendix B on page 93](#) for more information on how to Create Trends for Tuning. An example trend is shown below.



- Go online with the controller and have the drive in a Ready state.
- Enable the drive with an MSO instruction or motion direct command.

**IMPORTANT** If the drive has not been enabled before this step (new installation), verify that you have safeguards in place to safely remove power from the drive in case of an unstable situation where the drive can produce undesired motion.

- Execute the move profile.

Start the move cycles slowly at approximately 2...3% of the application's speed with a period of 4 seconds, increasing the speed to match your application requirements.

- Run the trend and observe the mechanical performance.

**IMPORTANT** If an audible noise exists at any time while tuning, use a smart phone app to identify the resonant frequencies and set torque loop filters to remove them. See [Compensating for High Frequency Resonances on page 63](#) to tune out resonant frequencies.

- Stop the trend.
- Stop the drive with an MAS instruction or motion direct command.
- Disable the drive with an MSF instruction or motion direct command.
- Repeat the previous steps, executing an autotune to incrementally increase the Position Loop Bandwidth until you achieve the highest performance within application requirements where the Torque Reference signal is not too oscillatory or noisy.

## Other Parameters

Other selection parameters on the Tune tab of the Axis Properties dialog box are listed below:

- Friction Compensation** - For more information, see [Friction Compensation on page 75](#).
- Torque Offset** - For more information, see [Appendix D](#) on Vertical Axis Tuning.
- Output Filter** - This check box enables the Low-pass Output Filter. It is used in conjunction with the Notch Filter to suppress audible vibration noise and high frequency resonances. For more information, see [Compensating for High Frequency Resonances on page 63](#).

When enabled, autotune calculates the Low-pass Output Filter Frequency (LP) in Hz.

$$\text{Low-pass Filter Bandwidth: } LP = 4 \times K_{VP} / (2 \pi)$$

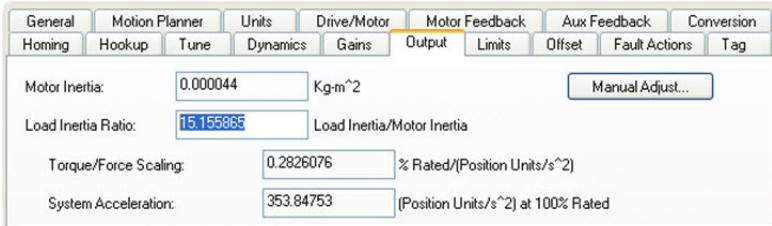
- Integrator Hold** - When Integrator Hold is enabled, the servo loop temporarily disables the position and velocity integrators while the velocity command is non-zero. This feature is used in point-to-point moves to minimize the integrator wind-up and phase lag during motion.

When Integrator Hold is disabled, all active position and velocity integrators are always enabled.

## Using Load Observer with Autotune

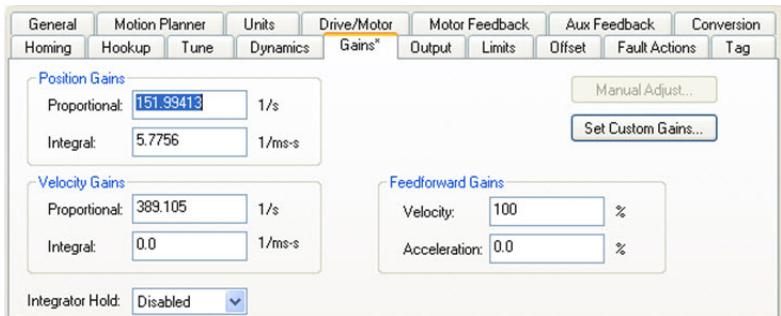
This procedure applies to Kinetix® 6000 and Kinetix 6200 drives. It involves configuring the load observer feature after executing an autotune. This method also works for any existing set of gains where the Load Inertia Ratio is known or manually calculated; that is, when the Load Inertia Ratio > 0. However, we advise that you first try the Sercos drive recommended settings. See [Sercos Drive Recommended Settings on page 36](#) for more information.

1. Click the Tune tab in the Axis Properties dialog box and perform an autotune. Follow all of the steps in [Start Tuning on page 45](#).
2. Click the Output tab in the Axis Properties dialog box and verify that the Load Inertia Ratio > 0. If you manually calculate the Load Inertia Ratio, enter the minimum load inertia.



3. Click the Gains tab in the Axis Properties dialog box.

Take note of the Velocity Proportional Gain value and set the Load Observer Bandwidth IDN P-432 equal to this value.



4. Configure these IDN parameter values.
  - a. IDN P-431 = 2
  - b. IDN P-432 = Velocity Proportional Gain value from the previous step
  - c. IDN P-433 = 0
  - d. IDN P-065 = 1

S	N:P,P#	Name	Value	Units
1: 0930		Reserved	0	
1: 0931		Load Obs Config	With Vel Est	
1: 0932		Load Obs Bw	389	rd/s
1: 0933		Load Obs Int Bw	0	rd/s

5. Enable the Low-pass Output Filter and set the Low-pass Output Filter Bandwidth equal to zero to properly disable it.

Sercos IDN P-065 has an impact on how the Low-pass Output Filter functions. See [Table 5 on page 20](#) for more information.

6. Execute a move profile and run a trend to observe the mechanical performance.

**IMPORTANT** If an audible noise exists at any time, use a smart phone app to identify resonant frequencies and set torque loop filters to remove them. See [Compensating for High Frequency Resonances on page 63](#) to tune out resonant frequencies.

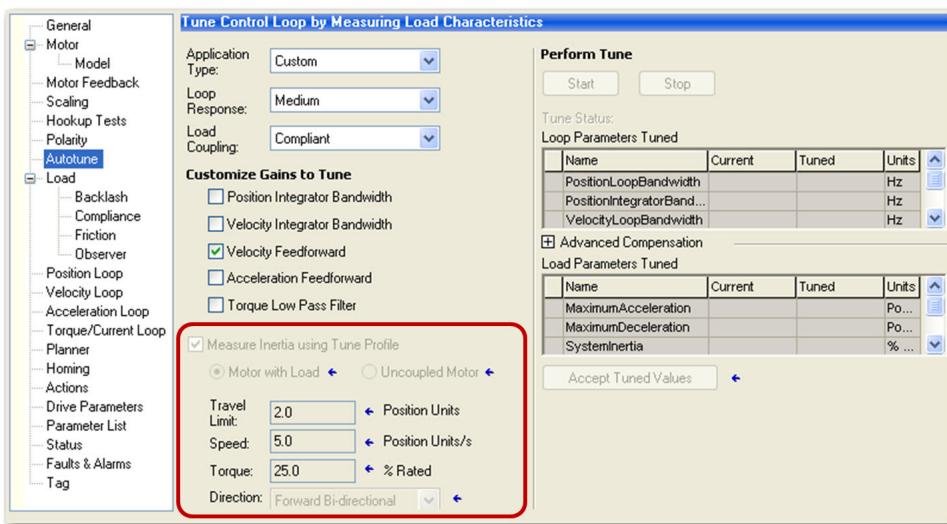
## EtherNet/IP Drives

The following subsections provide information for autotuning EtherNet/IP drives.

### Bump Test

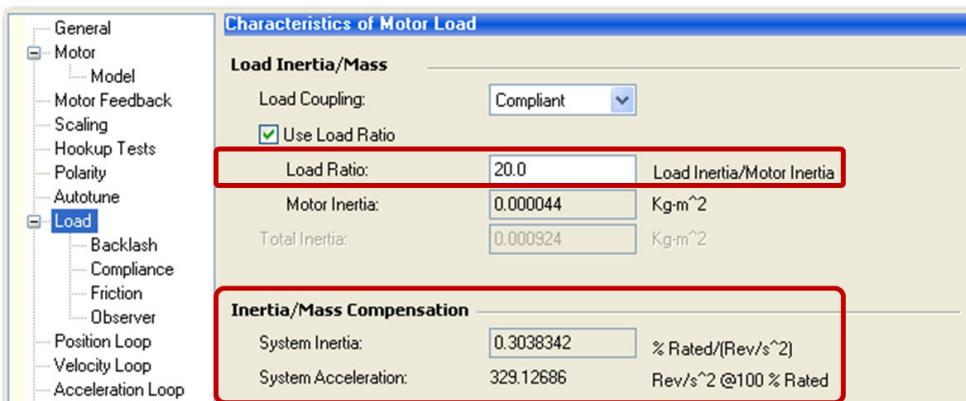
When autotune performs the bump test, a momentary tuning torque is applied to the motor while the acceleration and deceleration times are measured. This information is then used to calculate the load ratio (R), system inertia (torque scalar), and system acceleration. The bottom parameters on the Tune tab of the Axis Properties dialog box are used to keep the motor and load within specified position, velocity, acceleration, and direction limits during the bump test.

**Figure 49 - EtherNet/IP Drive Autotune Bump Test Parameters**



Set the Travel Limit and Speed based on the physical limits dictated by the mechanics of the application. Start with Torque/Force = 50. If the bump test fails, change the limits slightly and try again. The load ratio, system inertia (torque scalar), and system acceleration are then calculated upon successful completion of the bump test.

**Figure 50 - EtherNet/IP Autotune Bump Test Results**



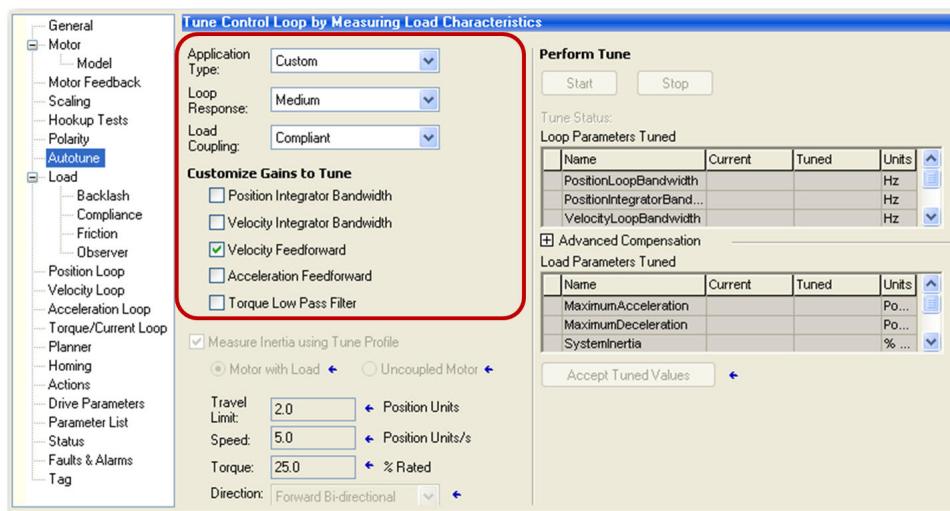
You can select whether you manually enter the load ratio or have it calculated during an autotune bump test. If you want to manually calculate the load ratio, clear the Measure Inertia using Tune Profile checkbox. For variable inertia loads, perform autotune at the point of lowest mechanical inertia. [Table 13](#) shows the different combinations possible by using the checkboxes in the previous two figures.

**Table 13 - Checkbox Combinations Affecting Load Ratio Calculation**

<b>Measuring Inertia using Tune Profile Checkbox</b>	<b>Use Load Ratio Checkbox</b>	<b>Result</b>
		Bump Test is not performed. Only the gains are calculated during autotune based on application type, loop response, and load coupling.
X		Bump Test is performed. Total Inertia is measured and calculated, but Load Ratio is not calculated.
	X	Bump Test is not performed. The entered Load Ratio is used in other autotune calculations.
X	X	Bump Test is performed. Load Ratio is calculated and used in other autotune calculations.

## Gain Calculation

The remaining parameters on the Tune tab of the Axis Properties dialog box are used to calculate control loop gains.

**Figure 51 - EtherNet/IP Drive Autotune Gain Selection Parameters**

### Application Type

There are many combinations of integrator and feed forward gains to enable. [Table 14](#) shows some common applications and which integrator and feed forward gains are enabled; that is, which ones are given non-zero values.

**Table 14 - EtherNet/IP Drive Gain Selection Based on Application Type**

Application Type	Applications	K <sub>PI</sub>	K <sub>VI</sub>	Integrator Hold	K <sub>VFF</sub>	K <sub>AFF</sub>
Basic	Basic smooth motion				X	
Tracking	<ul style="list-style-type: none"> <li>Converting</li> <li>Printing</li> <li>Web</li> <li>Flying shear</li> <li>Coordinated motion</li> <li>Rotary knife</li> <li>Packaging</li> </ul>		X		X	X
Point to Point	<ul style="list-style-type: none"> <li>Pick and place</li> <li>Indexing</li> <li>Robotics</li> <li>Palletizing</li> </ul>	X		X		
Constant Speed	<ul style="list-style-type: none"> <li>Conveyors</li> <li>Line shafts</li> <li>Crank</li> </ul>		X		X	
Custom <sup>(1)</sup>	High performance position tracking	X			X	X

(1) One of many possible combinations available with the Custom setting.

### Loop Response

Loop response sets the damping factor (z). For more information, see [Damping Factor on page 13](#). The loop response settings are described below:

- A High loop response sets the damping factor z = 0.8, which is characterized by a faster rise time with overshoot.
- A Medium loop response sets the damping factor z = 1.0, which is characterized by the fastest possible rise time without overshoot.
- A Low loop response sets the damping factor z = 1.5, which is characterized by a slower response, similar to decreasing the bandwidth.

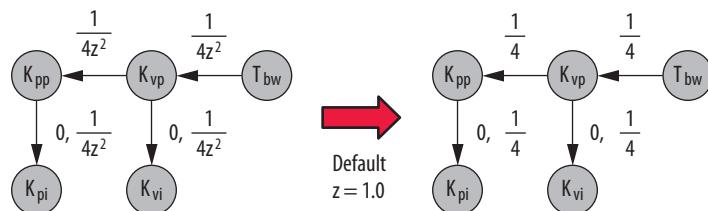
We recommend a Medium loop response setting.

### Load Coupling

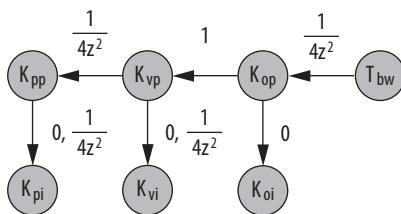
Load Coupling refers to whether the mechanical load connected to the motor is rigid or compliant, which is described as follows:

- Rigid systems typically involve high performance load mechanics that are tightly coupled directly to the motor shaft. The mechanics are stiff and there is no lost motion.
- Everything else is compliant, including systems with belts and pulleys, long shafts, short shafts with heavy loads, and couplings and gearbox with backlash and/or lost motion.

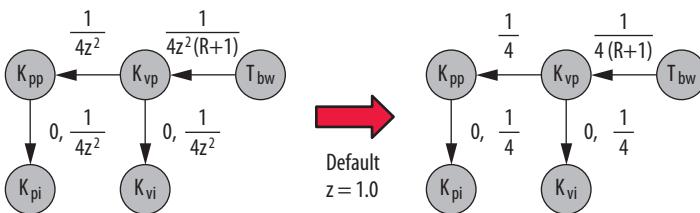
**Rigid Coupling** - A loop spacing of  $4z^2$  is sufficient for rigid loads. As a result, autotune performs the following control loop gain calculations. The damping factor z = 1.0 by default.

**Figure 52 - EtherNet/IP Drive Autotune Rigid without Load Observer Gain Relationships**

When the load observer is enabled on the Observer tab of the Axis Properties dialog box, autotune performs the following control loop gain calculations for a rigid load coupling. Note that  $K_{OP} = K_{VP}$  for all autotune rules.

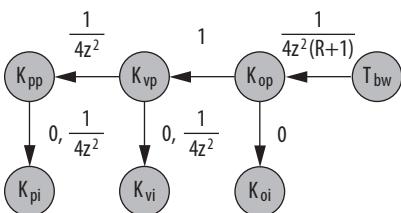
**Figure 53 - EtherNet/IP Drive Autotune Rigid with Load Observer Gain Relationships**

**Compliant Coupling** – Because compliant loads constitute such a large class of systems, all the gains are reduced by a factor of (R+1) to achieve stability under any circumstance. As a result, autotune performs the following control loop gain calculations. The damping factor z = 1.0 by default.

**Figure 54 - EtherNet/IP Autotune Compliant without Load Observer Gain Relationships**

Most of the time, these gain settings are too low and they can be manually increased to boost motion performance. For more information, see [Manual Tuning on page 57](#).

When load observer is enabled on the Observer tab of the Axis Properties dialog box, autotune performs the following control loop gain calculations. Note that  $K_{OP} = K_{VP}$  for all autotune rules.

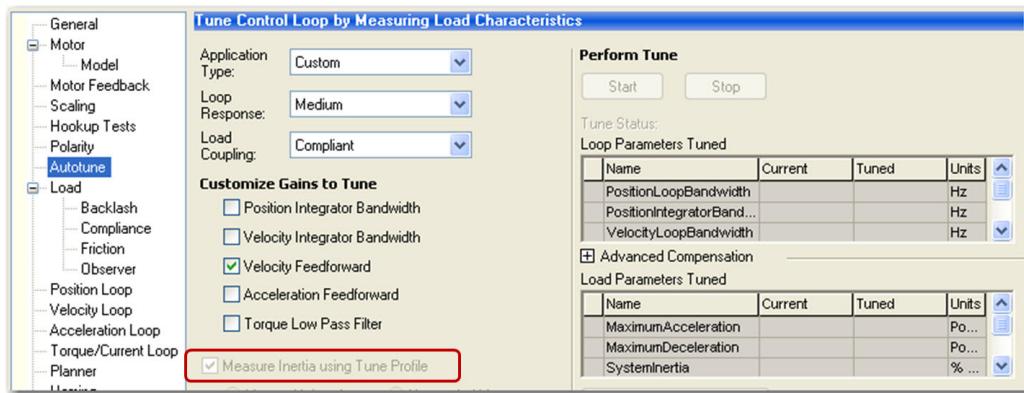
**Figure 55 - EtherNet/IP Drive Autotune Compliant with Load Observer Gain Relationships**

### Perform Tune

After choosing settings from the Application Type, Loop Response, and Load Coupling pull-down menus, you are ready to perform an autotune while online with the controller.

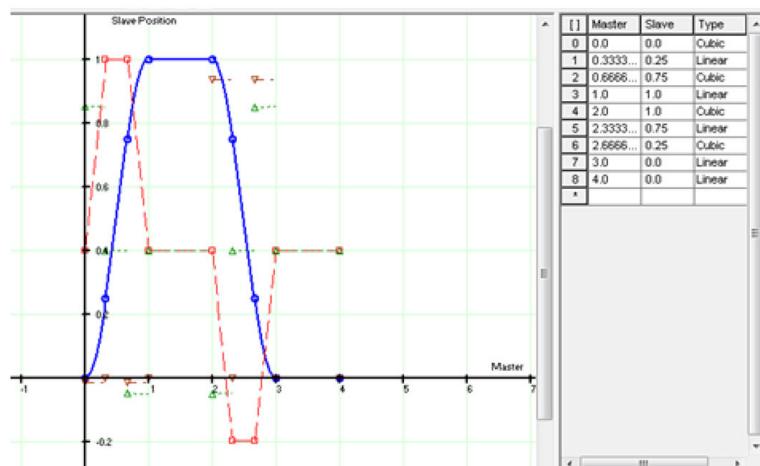
1. Click the Start button on the Autotune tab of the Axis Properties dialog box. For variable inertia loads, perform an autotune at the point of lowest mechanical inertia.

2. If you want to manually calculate the Load Ratio, clear the Measure Inertia using Tune Profile checkbox and enter the minimum load inertia on the Output tab.



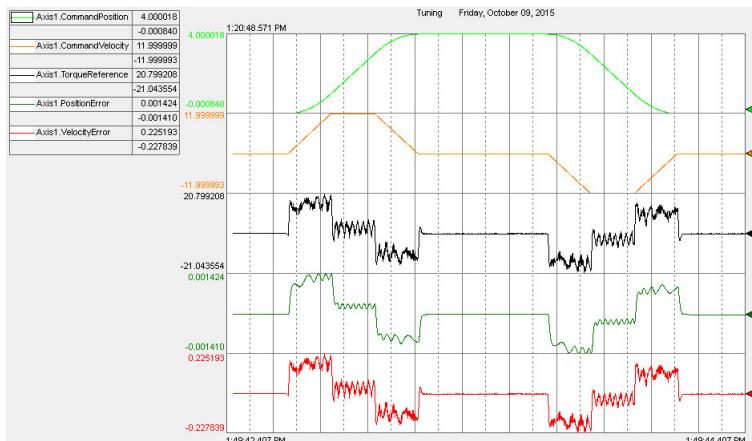
3. Wait for the tuning to complete.
4. Click Accept Tuned Values.
5. Test the drive and observe mechanical performance and stability.
  - a. Create a move profile in the Logix Designer application to observe the behavior of the mechanical load while tuning.

See [Appendix A on page 89](#) for more information on Creating Move Profiles for Tuning. An example CAM table for an MATC instruction is shown below.



- b. Create a trend in the Logix Designer application to monitor Command Position, Command Velocity, Torque Reference, Position Error, and Velocity Error.

See [Appendix B on page 93](#) for more information on how to Create Trends for Tuning. An example trend is shown below.



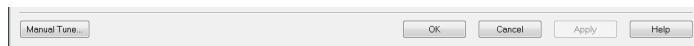
- c. Go online with the controller and have the drive in a Ready state.
- d. Enable the drive with an MSO instruction or motion direct command.

**IMPORTANT** If the drive has not been enabled before this step (new installation), verify that you have safeguards in place to safely remove power from the drive in case of an unstable situation where the drive can produce undesired motion.

- e. Execute the move profile.

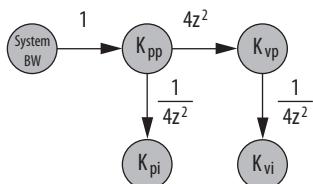
Start the move cycles slowly at approximately 2 to 3% of the application's speed with a period of 4 seconds, increasing the speed to match your application requirements.

- f. Run the trend and observe the mechanical performance.
  - g. Important: Suppress resonances. For Kinetix 5300, Kinetix 5500, and Kinetix 5700 drives, choose Tracking Notch Filter from the Adaptive Tuning Configuration pull-down menu on the Compliance tab in the Axis Properties dialog box. For Kinetix 6500 drives, if an audible noise exists at any time while tuning, use a smart phone app to identify resonant frequencies and set torque loop filters to remove them. See [Compensating for High Frequency Resonances on page 63](#) to manually tune out resonant frequencies.
  - h. Stop the trend.
  - i. Stop the drive with an MAS instruction or motion direct command.
  - j. Disable the drive with an MSF instruction or motion direct command.
6. Click the Manual Tune button in the low left corner of the Axis Properties dialog box.

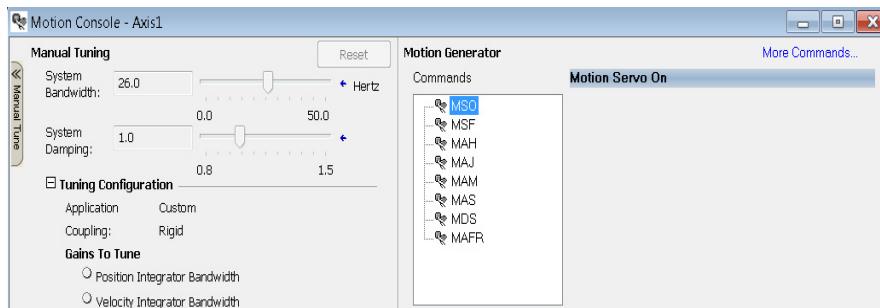


Similar to the Sercos Tune Results box, you can increase or decrease the System Bandwidth slider to recalculate the gains with higher or lower values based on the autotune rules. Rigid loads typically require little to no adjustment of System Bandwidth and compliant loads typically require an increase in System Bandwidth to dial in the performance.

**Figure 56 - System Bandwidth Gain Calculation**



7. Repeat [step 5](#) to test performance while adjusting the System Bandwidth slider until you achieve the highest performance within application requirements where the Torque Reference signal is not too oscillatory or noisy.



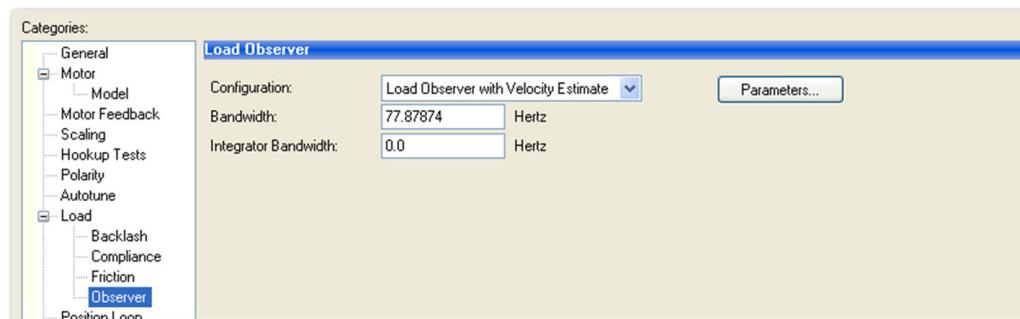
## Using Load Observer with Autotune

This procedure applies to Kinetix 5300, Kinetix 5500, Kinetix 5700, and Kinetix 6500 drives. It involves configuring the load observer feature after executing an autotune. This method also works for any existing set of gains where the Load Ratio is known or manually calculated; that is, when the Load Ratio  $> 0$ . However, we advise that you first try the EtherNet/IP drive recommended settings. See [EtherNet/IP Drive Recommended Settings on page 39](#) for more information.

1. Click the Start button on the Autotune tab in the Axis Properties dialog box and perform an autotune. Follow all of the steps in Perform Tune on [page 52](#).
2. Click the Load tab in the Axis Properties dialog box and verify that the Load Ratio  $> 0$ . If you manually calculate the Load Inertia Ratio, enter the minimum load inertia.
  - If the Measure Inertia using Tune Profile box was checked during autotune, the measured Load Ratio is shown in the Load Ratio box.
  - If you manually calculate the Load Ratio, use the minimum load inertia if it changes with position or over time.
  - Otherwise, enter the Load Ratio if it is known.



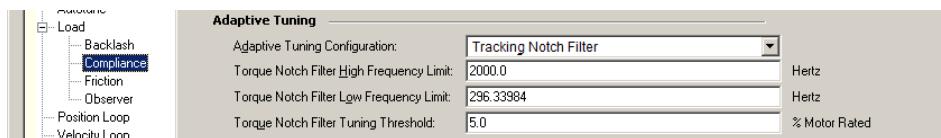
3. Click the Observer tab in the Axis Properties dialog box.
  - a. From the Configuration pull-down menu, choose Load Observer with Velocity Estimate



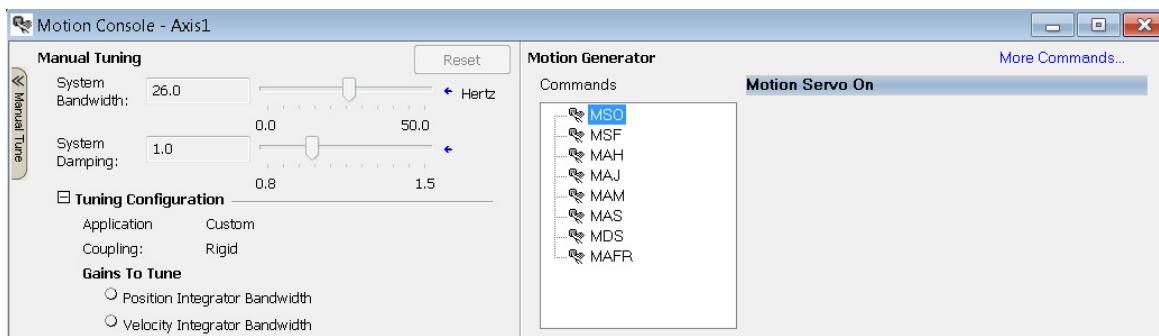
- b. Click Apply.

The Load Observer Bandwidth and other gains are set automatically.

4. Important: Suppress resonances.
  - a. For Kinetix 5300, Kinetix 5500, and Kinetix 5700 drives, choose Tracking Notch Filter from the Adaptive Tuning Configuration pull-down menu on the Compliance tab in the Axis Properties dialog box.



- b. For Kinetix 6500 drives, if an audible noise exists at any time while tuning, use a smart phone app to identify resonant frequencies and set torque loop filters to remove them. See [Compensating for High Frequency Resonances on page 63](#) to tune out resonant frequencies.
5. Click Manual Tune in the lower left corner of the Axis Properties dialog box and adjust the System Bandwidth slider to recalculate the gains with higher or lower values based on the autotune rules and achieve the required level of performance for the application.



6. Set the Load Observer Bandwidth equal to the Velocity Loop Bandwidth after each adjustment of the System bandwidth slider since the System Bandwidth slider does not affect the Load Observer bandwidth.



## Is Further Tuning Required?

Here are some observations that indicate the servo drive can produce satisfactory performance without additional tuning:

- Visibly smooth motion from a smooth cycle profile
- Little to no audible noise produced during and after a commanded motion
- Position and/or velocity errors are within what is required for the application
- Position and/or velocity errors are repeatable

If the load does not respond as intended, consider these factors before tuning:

- Investigate proper servo sizing. The best way to do this is to model the axis in Motion Analyzer software to verify the proper sizing of the motor and drive for the specified load and move profile. You can also compare various design options for gear ratio, load size, coupling configuration, high resolution versus low resolution feedback device, and so forth.
- Simulate the axis in Motion Analyzer software, testing various control loop gain settings to conclude the desired motion can be achieved. Compliance and machine vibration can often be minimized by creating a more direct and stiff coupling between the motor and load. Helping to achieve this are quality couplings, gearboxes, actuators, and guides.

If the load requires additional tuning to further optimize motion performance, see Manual Tuning, [Chapter 4 on page 57](#).

## Manual Tuning

This chapter describes a generally accepted method of manually tuning a servo drive. This method is sometimes referred to as inside-out tuning where the inner velocity loop is tuned, followed by tuning the outer position loop. It involves incrementally increasing control loop gains to the point of marginal stability, then backing them off by a given percentage. Typical ranges for various gains are also given to provide guidelines.

Topic	Page
Initialize the Axis (Optional)	57
Tune the Velocity Loop	58
Tune the Position Loop	61

The out-of-box and autotune rigid methods achieve relatively high performance. However, if you are comfortable with tuning, this method can help to further optimize performance from autotune-compliant settings or if maximum performance is required. This method can be used in the following situations where previously described methods do not produce the required level of performance:

- To optimize performance starting with out-of-box or autotuned settings
- To tune a difficult axis from a stable initial state

Other requirements include the following:

- Start with a set of gains that produces stable operation
- The load is attached to the motor while tuning
- The drive uses a nested velocity-position loop structure, for example, Sercos and EtherNet/IP drives

## Initialize the Axis (Optional)

If you have just configured the axis with autotune or out-of-box settings outlined in the previous chapters, but you still require more performance, skip this section and proceed to Tune the Velocity Loop on [page 58](#).

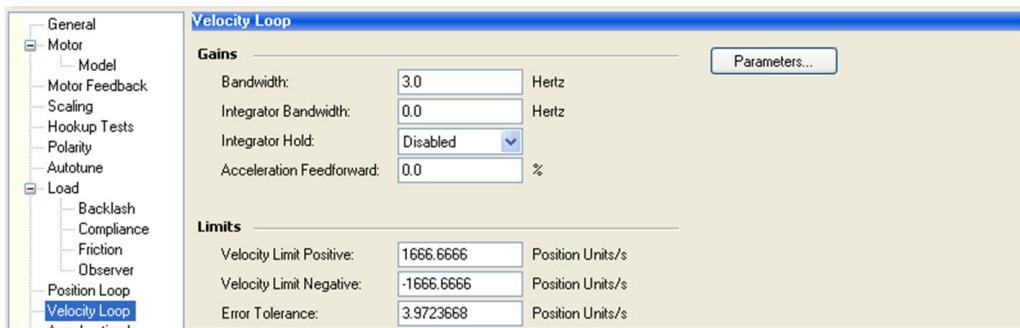
If your mechanical load is unstable or your control loop gain values are undesirable, then start with this section. However, we advise that you first attempt the recommended steps that are outlined in Out-of-Box Tuning, [Chapter 2 on page 35](#), and Autotuning, [Chapter 3 on page 43](#), as they will yield stable results.

Follow these steps to bring the axis back to an initial state with default settings.

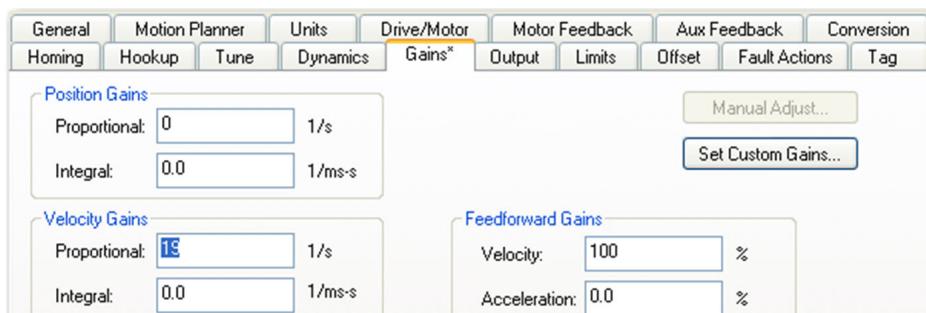
1. Go offline with the controller in the Studio 5000 Logix Designer® application.
2. Open the Axis Properties dialog box.
3. Reset the axis configuration back to default settings by using either of these methods:
  - Set the motor type to 'none' and reselect the motor by its catalog number.
  - Delete the axis and drive and re-create them.
4. Configure these applicable axis parameters.
  - a. Set the Loop Type = Position Servo
  - b. Set the damping factor  $z = 1.0$  (Loop Response = Medium in an EtherNet/IP drives)
  - c. Set  $K_{PP}$  and  $K_{PI} = 0$
  - d. Set  $K_{VP}$  and  $K_{VI} = 0$
  - e. Set Position and Velocity Integrator Hold = Disabled

- f. Set Load Observer Configuration = Disabled
  - g. Set  $K_{OP}$  and  $K_{OI} = 0$
  - h. Set  $K_{VFF}$  and  $K_{AFF} = 0$
  - i. Set the Low Pass Filter Bandwidth = 0
  - j. Set the Notch Filter Frequency = 0
  - k. Set the Lead-Lag Filter Gain = 1
  - l. Set the Lead-Lag Filter Bandwidth = 0
5. Configure the axis dynamic limits.
- Set the Torque limits to what is required for the application.
  - Change the Position Error Tolerance to a large value to avoid a fault.
  - For an EtherNet/IP drive, change the Velocity Error Tolerance to a large value to avoid a fault.
6. Configure the Load Inertia Ratio.
- If you know  $R > 0$  from a previous autotune, enter it into the Logix Designer application.
  - If you calculated  $R$  at the point of lowest mechanical inertia, enter it into the Logix Designer application.
  - Otherwise, set  $R = 0$  in the Logix Designer application.
7. Set an initial value for  $K_{VP}$ .

Use a small value because the error tolerance has been changed. For an EtherNet/IP drive, a value of  $K_{VP} = 3$  Hz is a good starting point.



For a Sercos drive, a value of  $K_{VP} = 19$  radians/second is a good starting point.

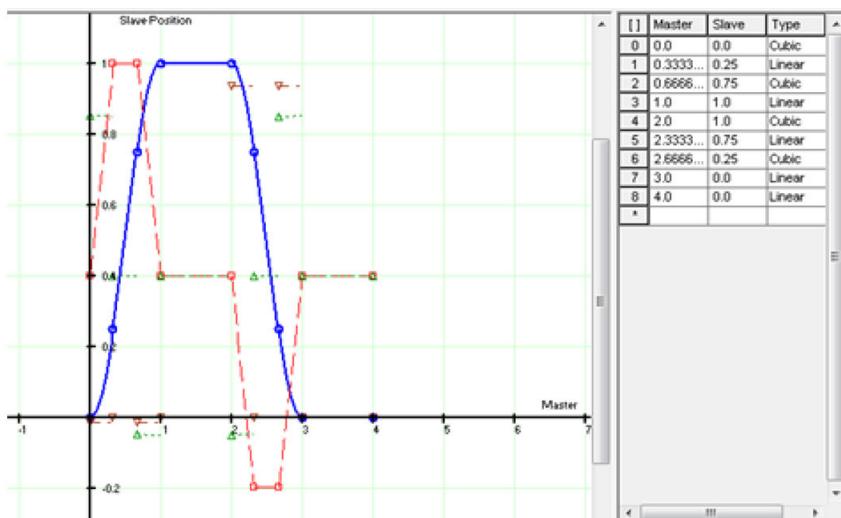


## Tune the Velocity Loop

Follow these steps to manually tune the velocity loop.

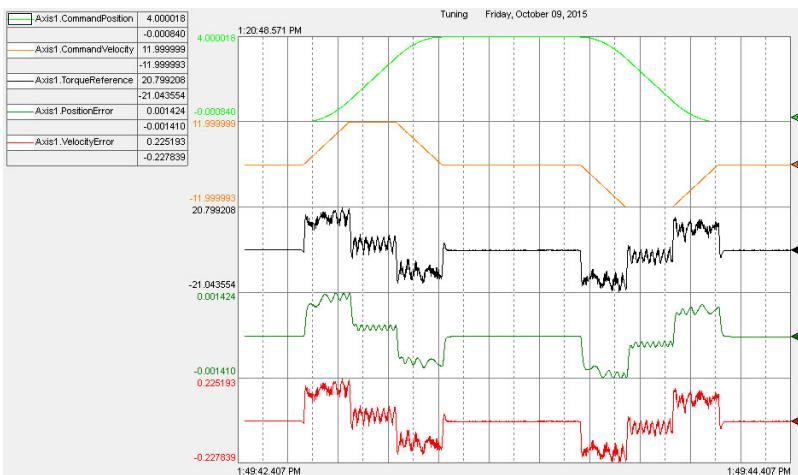
- Create a move profile in the Logix Designer application to observe the behavior of the mechanical load while tuning.

See [Appendix A on page 89](#) for more information on Creating Move Profiles for Tuning. An example CAM table for an MATC instruction is shown below.



2. Create a trend in the Logix Designer application to monitor Command Position, Command Velocity, Torque Reference, Position Error, and Velocity Error.

See [Appendix B on page 93](#) for more information on how to Create Trends for Tuning. The following is an example trend.



3. Isolate the velocity loop.
  - a. Note the position and feed forward gain values,  $K_{PP}$ ,  $K_{PI}$ ,  $K_{VFF}$ , and  $K_{AFF}$  in the Axis Properties dialog box.

You must change them temporarily to isolate the velocity loop and later restore them to the original values.

- b. Temporarily configure these settings in the Axis Properties dialog box:

- $K_{PP} = 0$
- $K_{PI} = 0$
- $K_{AFF} = 0$
- $K_{VFF} = 100$
- Low Pass Bandwidth = 0
- Notch Filter Frequency = 0

4. Go online with the controller and have the drive in a Ready state.
5. Execute motion.
  - a. Enable the drive with an MSO instruction or motion direct command.

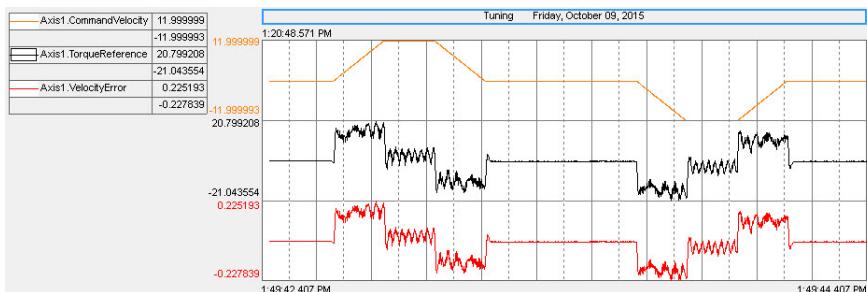
**IMPORTANT** If the drive has not been enabled before this step (new installation), verify that you have safeguards in place to safely remove power from the drive in case of an unstable situation where the drive can produce undesired motion.

- b. Execute the move profile.

Start the move cycles slowly at approximately 2...3% of the application's speed, increasing the speed to match your application requirements.

6. Run the trend.
7. Incrementally increase  $K_{VP}$  while observing a reduction in the Velocity Error trend.

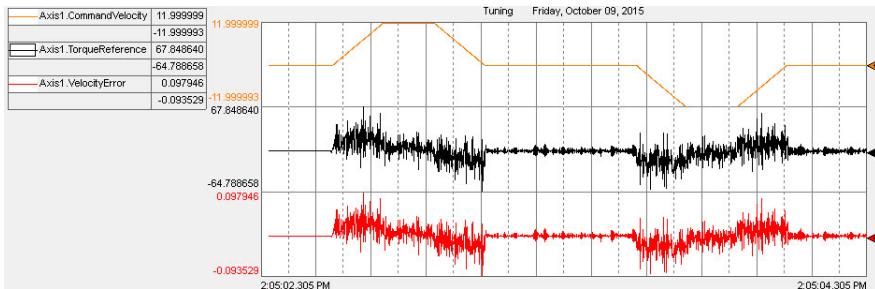
You may need to stop motion after each modification of  $K_{VP}$ .



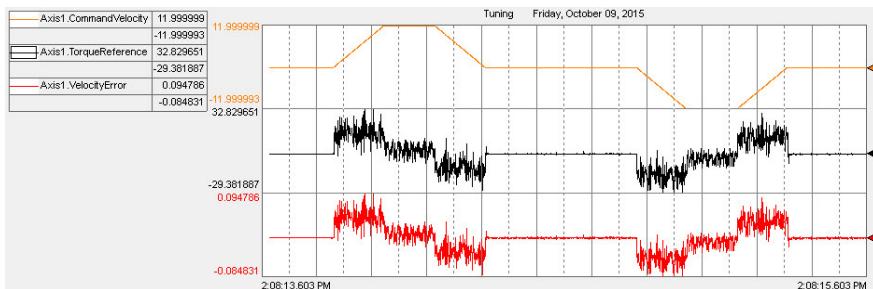
8. Optionally, you can enable Load Observer with Velocity Estimate in supported drives. See [Load Observer Feature on page 16](#) for more information.
  - a. If the load ratio  $R = 0$  and Load Observer with Velocity Estimate is enabled, set  $K_{OP} = 4.0 * K_{VP}$ .
  - b. If the load ratio  $R > 0$  and Load Observer with Velocity Estimate is enabled, set  $K_{OP} = 1.0 * K_{VP}$ . Note that  $K_{OP} = 0.5 * K_{VP}$  works better in some applications.
  - c. Every time you increase or decrease  $K_{VP}$  in the following steps, also increase or decrease  $K_{OP}$ , keeping the ratio between  $K_{VP}$  and  $K_{OP}$  constant.
9. Continue increasing  $K_{VP}$  until ringing occurs in the Torque Reference or an audible high frequency noise is present.

**IMPORTANT** If ringing occurs in the Torque Reference signal or an audible noise exists at any time while tuning, use a smart phone app to identify the resonant frequencies and set torque loop filters to remove them. See [Compensating for High Frequency Resonances on page 63](#) to manually tune out resonant frequencies. This may occur a few times as you increase  $K_{VP}$  before instability occurs.

10. Continue increasing  $K_{VP}$  until a low pitch growling sound occurs or when oscillation occurs in the Velocity Error Torque Reference signal as shown in the following example. This is the point where instability begins and  $K_{VP}$  cannot be successfully increased any further.



11. Decrease  $K_{VP}$  by dividing it by two.



12. Increase the dynamics of the move profile; for example, increase Time Scaling and Distance Scaling in an MATC or other move instruction to generate a more aggressive move profile that approaches the dynamic and travel limits of your application.
13. See [Table 12 on page 44](#) to determine if setting other velocity loop gains are required for your application.
  - For an EtherNet/IP drive, a typical range of values for various gains are given:

$$0 \leq K_{\text{AFF}} \leq 100$$

$$0 \leq K_{\text{VI}} \leq K_{\text{VP}} / 4$$

- For Sercos drives, a typical range of values for various gains are given. Note that many Sercos applications do not benefit from the use of acceleration feed forward and  $K_{\text{aff}}$  should remain equal to zero.

$$0 \leq K_{\text{AFF}} \leq 100$$

$$0 \leq K_{\text{VI}} \leq K_{\text{VP}}^2 / 4000$$

14. Stop the trend.
15. Stop the drive with an MAS instruction or motion direct command.
16. Disable the drive with an MSF instruction or motion direct command.
17. Restore the position and feed forward gains with the original values noted in step [3a](#) to re-enable the position loop.

## Tune the Position Loop

Follow these steps to manually tune the position loop after tuning the velocity loop in the previous section.

1. Go online with the controller and have the drive in a Ready state.
2. Execute motion.
  - a. Enable the drive with an MSO instruction or motion direct command.

**IMPORTANT** If the drive has not been enabled before this step (new installation), verify that you have safeguards in place to safely remove power from the drive in case of an unstable situation where the drive can produce undesired motion.

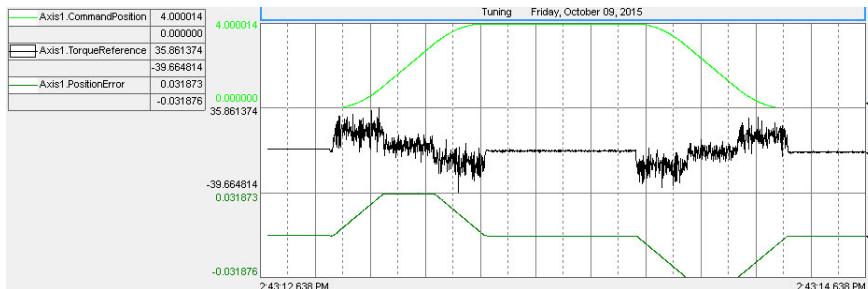
- b. Execute the move profile.

Start the move cycles slowly at approximately 2...3% of the application's speed with a period of 4 seconds, increasing the speed to match your application requirements.

3. Run the trend.
4. Incrementally increase  $K_{\text{PP}}$  while observing a reduction in the Position Error trend.

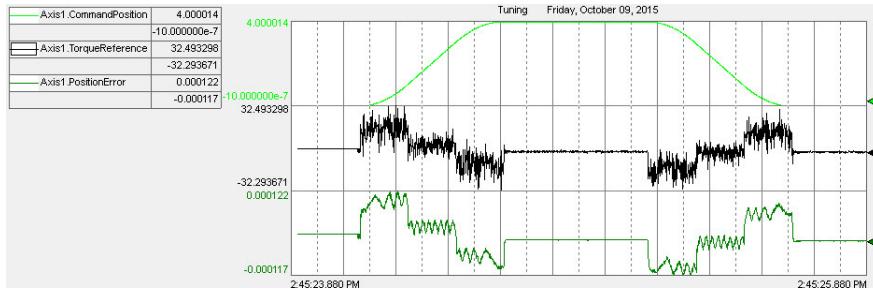
You may need to stop motion after each modification of  $K_{\text{PP}}$ .

5. Continue increasing  $K_{\text{PP}}$  until you hear an audible low pitch growling sound or until oscillation occurs in the Torque Reference.



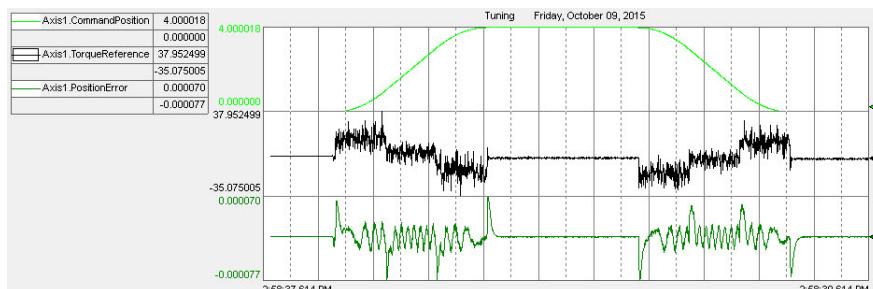
6. Decrease  $K_{PP}$  by dividing it by two.
7. See [Table 12 on page 44](#) to determine if setting other position loop gains are required for your application.
  - For an EtherNet/IP drive, a typical range of values for velocity feed forward gain is given:

$$0 \leq K_{VFF} \leq 100$$



- For an EtherNet/IP drive, a typical range of values for the position integral gain is given:

$$0 \leq K_{PI} \leq K_{PP} / 4$$



- For a Sercos drive, a typical range of values for various gains are given:

$$0 \leq K_{FF} \leq 100$$

$$0 \leq K_{PI} \leq K_{PP}^2 / 4000$$

8. Stop the trend.
9. Stop the drive with an MAS instruction or motion direct command.
10. Disable the drive with an MSF instruction or motion direct command.
11. With the drive disabled, adjust the Position Error Tolerance and Velocity Error Tolerance to values that are acceptable to the application and that result in a fault when required.

## Filters and Compensation

This chapter describes high-frequency resonances common to motion applications and methods used to compensate for them.

Topic	Page
Compensating for High Frequency Resonances	63
Notch Filters	65
Low Pass Filter	73
EtherNet/IP Drive Lag Filter	74
Friction Compensation	75
Backlash Compensation	77

Sercos drives provide the following filters:

- First-order low-pass filter
- Notch filter (tracking notch available in Kinetix® 6000 drives via message instructions)

Ethernet/IP drives Kinetix 5500 and Kinetix 5700 (using the Logix Designer application version 32 and later):

- First order low pass filter
- First order lead-lag filer
- One notch filter with tracking

Ethernet/IP drives Kinetix 5300, Kinetix 5500, and Kinetix 5700 (using the Logix Designer application version 33 and later):

- First order low pass filter
- First order lead-lag filer
- Four notch filters with tracking
- Two command notch filters

## Compensating for High Frequency Resonances

Approximately half of all motion applications exhibit high-frequency resonances that are apparent by an audible high-frequency squealing of the load mechanics. Furthermore, all applications will ultimately exhibit resonance when pushing gains to their limit while optimizing performance. See [Resonances on page 22](#) for more information on the various type of resonances and how to suppress them.

## Automatic Compensation

For drives with the Tracking Notch feature, see [Adaptive Tuning Feature on page 22](#) to automatically compensate for high frequency resonances.

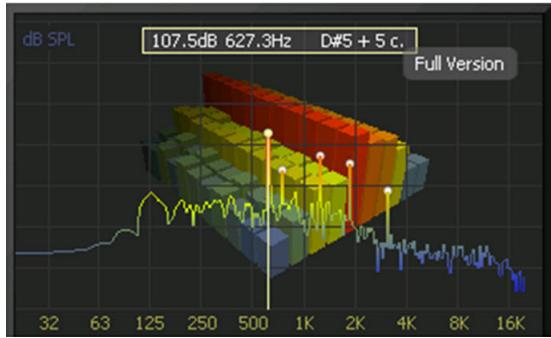
## Manual Compensation

Follow these steps to manually identify and reduce the presence of high frequency resonances.

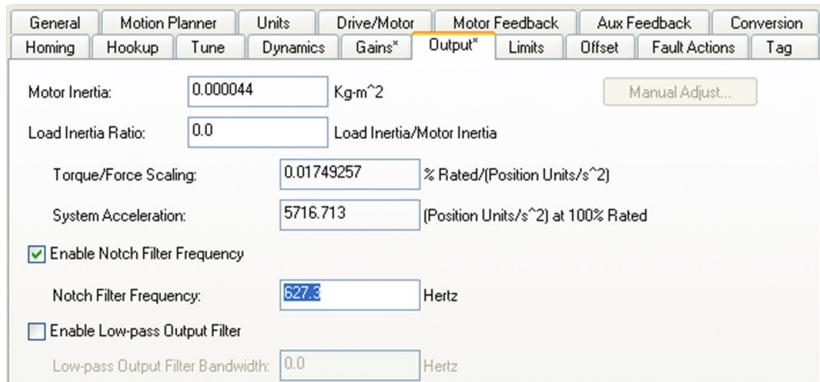
1. Perform the following move sequence by using Motion Direct Commands.
  - a. Enable the drive with an MSO instruction.
  - b. Slowly jog the axis with a MAJ or MAM instruction.
  - c. Stop the axis with a MAS instruction.
  - d. Disable the drive with an MSF instruction.

**IMPORTANT** Sometimes an audible resonance is heard before the axis is jogged, making the MAJ and MAS instructions unnecessary.

2. Determine if an audible high-frequency resonance exists in your motion application.
  - If an audible high frequency resonance is not present during the move sequence, skip the remaining steps and tuning is complete.
  - If an audible high frequency resonance is present during the move sequence, use an FFT smart phone or tablet application to identify the dominant resonant frequencies. This frequency is shown as the largest peak in the following iAnalyzer Lite example.



3. If resonances are below the torque loop bandwidth in Hz and a low pitch growling sound is present, then instability is present and you must decrease your control loop gains before continuing with the following steps.
4. For a Sercos drive, click the Output tab in the Axis Properties dialog box and do the following.
  - a. Check the Enable Notch Filter Frequency box.
  - b. Set the Notch Filter Frequency to the resonant frequency with the largest magnitude.
    - If multiple resonances have nearly the same magnitude, set the Notch Filter Frequency to the lowest resonant frequency.
    - c. If the problem persists, set the Notch Filter Frequency to the next highest resonant frequency.
    - d. If the problem persists, select the Enable Low-pass Output Filter box, set the Low-pass Output Filter Frequency to 2000 Hz, and decrement it until the ringing stops or until instability occurs. If instability occurs, detune control loop gains until the system stabilizes.
    - e. As you change control loop gains or the low pass filter bandwidth, resonant frequencies may shift or change. As a result, it may be necessary to periodically adjust torque notch filter parameters.
    - f. Click OK.

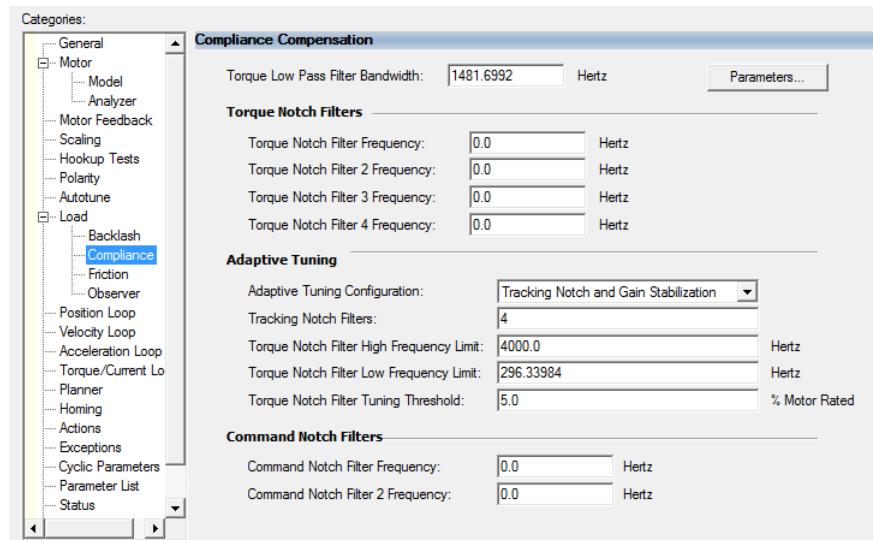


5. For an EtherNet/IP drive, click the Compliance tab in the Axis Properties dialog box and do the following.
- Set the Torque Notch Filter Frequency to the resonant frequency with the largest magnitude.

If multiple resonances have nearly the same magnitude, set the Torque Notch Filter Frequency to the lowest resonant frequency.

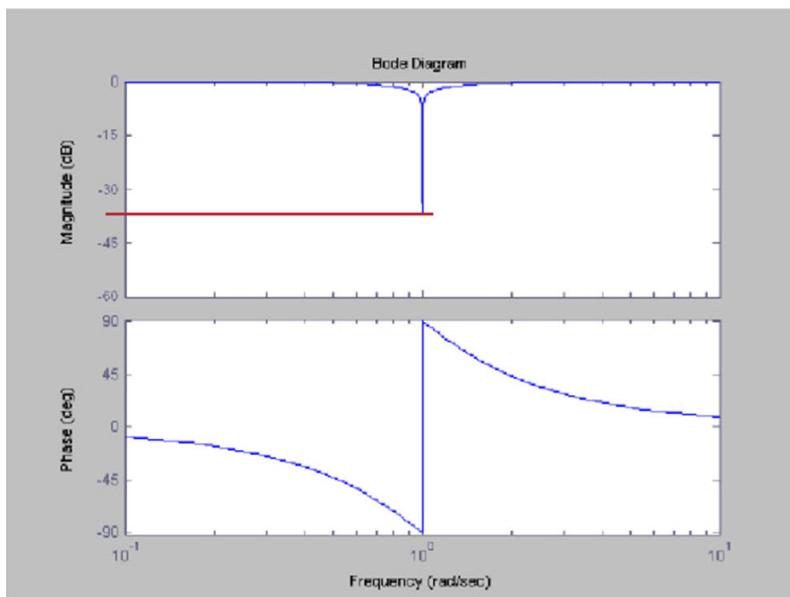
- If the problem persists, set the Torque Notch Filter Frequency to the next highest resonant frequency.
- If the problem persists, set the Torque Low Pass Filter Bandwidth to 2000 Hz and decrement it until ringing stops or until instability occurs. If instability occurs, detune control loop gains until the system stabilizes.
- As you change control loop gains or the low pass filter bandwidth, resonant frequencies may shift or change. As a result, it may be necessary to periodically adjust torque notch filter parameters.

e. Click OK.



## Notch Filters

This section applies to the torque notch filters in Kinetix drives. A notch filter is a filter that passes most frequency signals unaltered but attenuates signals within a specific range of frequencies. [Figure 57](#) shows gain and phase through the filter as a function of the frequency content of a signal passing through it.

**Figure 57 - Notch Filter Bode Plot**

Notch filters typically have a relatively narrow and deep attenuation band. Maximum attenuation is achieved at the notch filter frequency. The notch filter is second order with an attenuation of approximately 40 dB at the notch frequency.

As with any filter, its output is shifted in phase from the input. The shift is present for all frequencies. The output lags the input up to the notch frequency and leads the input for frequencies above the notch frequency.

The notch filter is effective in resonance control when the resonant frequency is higher than the control loop bandwidth. Like the low-pass filter, the notch filter works by significantly reducing the amount of energy in the device output that can excite high frequency resonances. It can be used even when resonant frequencies are relatively close to the control loop bandwidth. That is because the phase lag introduced by the notch filter is localized around the notch frequency. For the notch filter to be effective, the Notch Filter Frequency must be set very close to the natural resonance frequency of the load.

Notch filters are represented by the following second order transfer function.

$$G(s) = \frac{K^2 s^2 + 2KZ_D(2\pi F)s + (2\pi F)^2}{s^2 + 2Z_W(2\pi F)s + (2\pi F)^2} = K^2 \left( \frac{s^2 + 2Z_D(2\pi F_D)s + (2\pi F_D)^2}{s^2 + 2Z_W(2\pi F)s + (2\pi F)^2} \right), K = \frac{F}{F_D}$$

Coefficients are defined in [Table 15](#). They feature variable adjustment of width, depth, and gain to allow for a wide range of applications.

**Table 15 - Variable Notch Filter Parameters**

Parameter	Name	Minimum	Maximum	Default
F	Notch Filter Frequency	0 Hz	3999 Hz	0 Hz
K	Notch Filter Gain	-20	20	1.0
Z <sub>W</sub>	Notch Filter Width	0 = minimum width	$10^4$	0.707
Z <sub>D</sub>	Notch Filter Depth	0 = maximum depth	$10^4$	0

There are four torque notch filters in the drive. There are two position reference notch filters, two velocity reference notch filters, and two process PID reference notch filters in the drive. These three sets of reference notch filters share common parameters, which allow all command signals to be affected uniformly. The torque notch filters suppress MF and HF motor side resonances and the reference notch

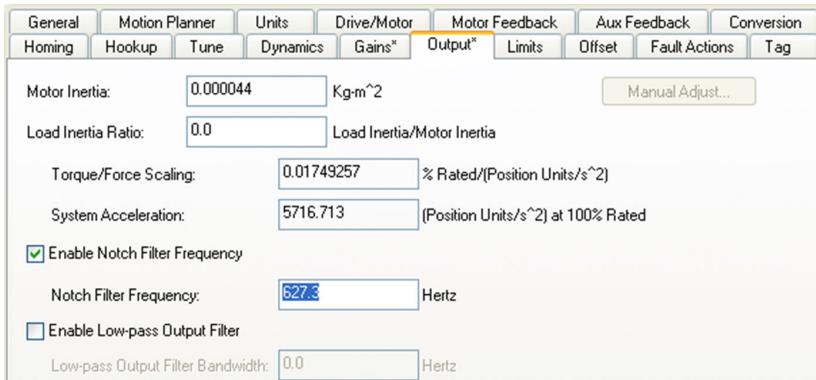
filters suppress load side resonances. See Resonances on page 15 for more information on the various types of resonances and how to suppress them.

For more information on manually setting the notch filter, see [Compensating for High Frequency Resonances on page 63](#) and [Automatic Compensation on page 63](#).

## Sercos Drives

For a Sercos drive, click the Output tab in the Axis Properties dialog box. Check Enable Notch Filter Frequency and set the Notch Filter Frequency to the desired frequency in Hz.

**Figure 58 - Sercos Notch Filter Frequency**

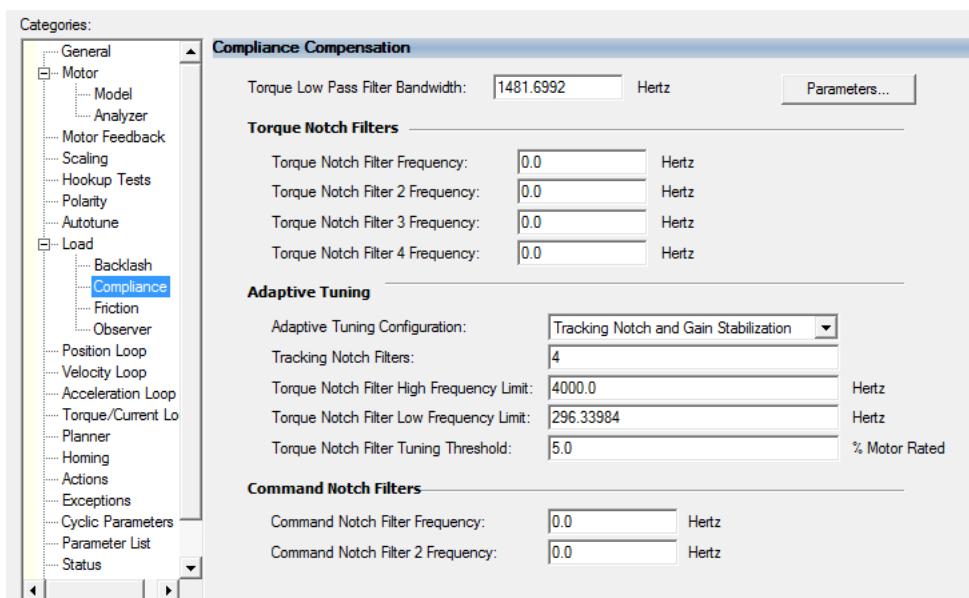


Sercos IDN P-065 has an impact on how the low-pass output filter functions. See [Table 5 on page 20](#) for more information.

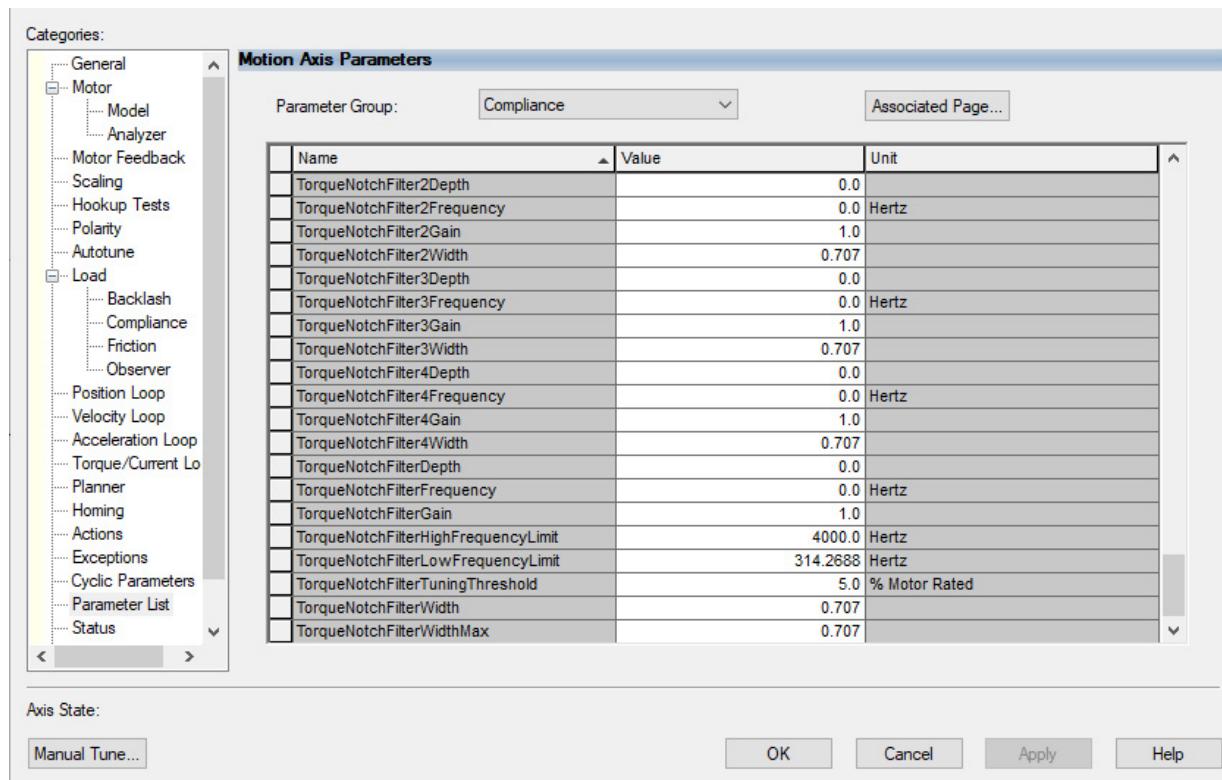
## EtherNet/IP Drives

For an EtherNet/IP drive, click the Compliance tab in the Axis Properties dialog box. Set the Torque Notch Filter Frequency to the desired frequency in Hz.

**Figure 59 - CIP Notch Filter Frequency**



From the Compliance tab, click the Parameters button to configure additional notch filter settings.



The torque notch filter can also be automatically set in Kinetix 5300, Kinetix 5500, and Kinetix 5700 drives. Choose Tracking Notch Filter from the Adaptive Tuning Configuration pull-down menu.

## Configuration

Notch filters have four modes of operation depending on the gain K setting, similar to lead-lag filters. The difference is that notch filters can only be configured as second order and lead-lag filters can only be configured as first order. The filter is disabled when F = 0 Hz, regardless of the other settings.

### Configured as a Notch Filter (K = 1)

When K = 1, the notch filter operates as a second order notch filter. This is the recommended and default mode of operation. Also, adaptive tuning sets K = 1 when tracking a notch filter.

The variable width and depth functionality of the notch filter provides more flexibility in achieving high performance in some situations. The width can be increased, which allows one filter to suppress two HF resonances that are close to each other. An increased width is also effective in suppressing a resonance that slightly moves as mechanics slowly degrade over time. This reduces the need for retuning. The width of a torque notch filter can be decreased to suppress an MF resonance that is near the closed loop bandwidth. This minimizes the impact of the filter's phase lag on stability margin and reduces the need for detuning. The depth can be adjusted only as deep as required to suppress resonances with magnitudes large and small. This also conserves phase and allows for higher tuning performance.

The recommended range for width  $Z_W$  is from a minimum width of 0 to a maximum width of 4. The default width  $Z_W = 0.707$  produces relatively good performance across applications.

The recommended range for depth  $Z_D$  is from a maximum depth of 0 to a minimum depth of  $< Z_W$ . The default depth  $Z_D = 0$  provides maximum notch attenuation. Also, adaptive tuning sets  $Z_D = 0$  when tracking a notch filter. Note that the filter is off when  $K = 1$  and  $Z_D = Z_W$  or when  $F = 0$ . Furthermore, the filter becomes an inverse notch (generating resonance) when  $Z_D > Z_W$ .

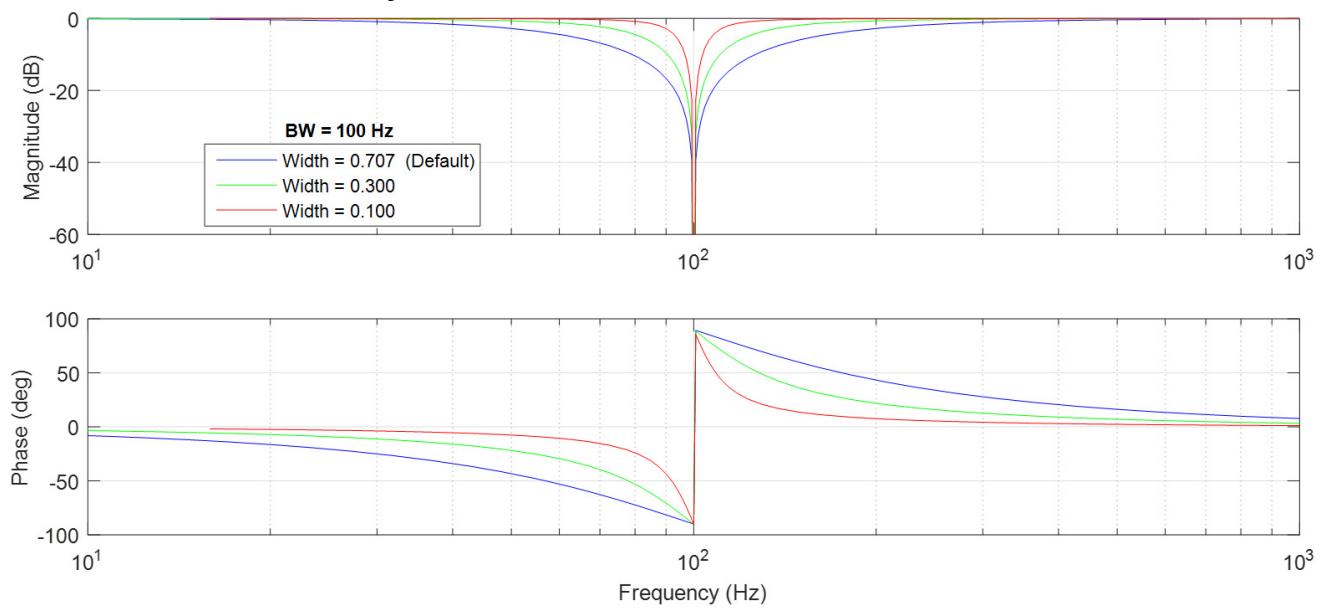
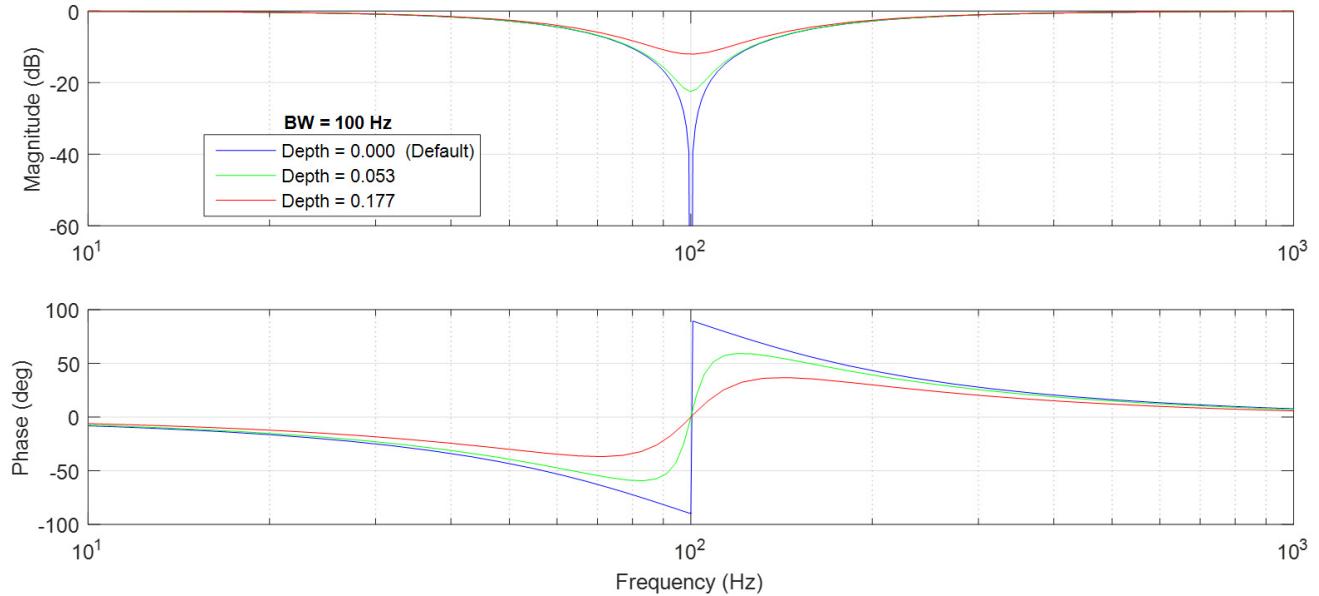
Notch filter width can be approximated in units of [Hz] as the range of frequencies impacted by the filter between -3 dB points.

$$\text{Width[Hz]} = 2FZ_W(1 - z^2 - 0.5z^4 - z^6), z = \frac{Z_D}{Z_W}$$

Notch filter depth can be calculated in units of [dB] at the center frequency of the filter.

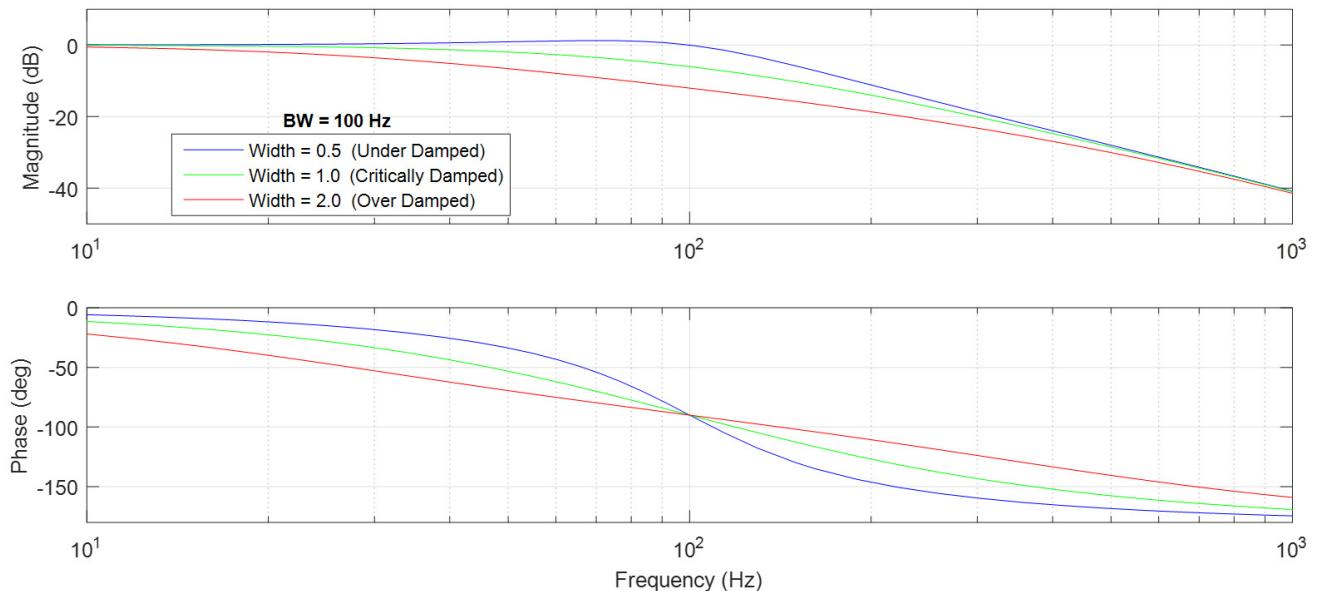
$$\text{Depth[dB]} = 20\log_{10}\left(\frac{Z_D}{Z_W}\right)$$

Bode plots with differing width and depth notch filter settings for  $K=1$  are shown in [Figure 60](#) and [Figure 61 on page 70](#).

**Figure 60 - Variable Width Notch Filter Configurations****Figure 61 - Variable Depth Notch Filter Configurations**

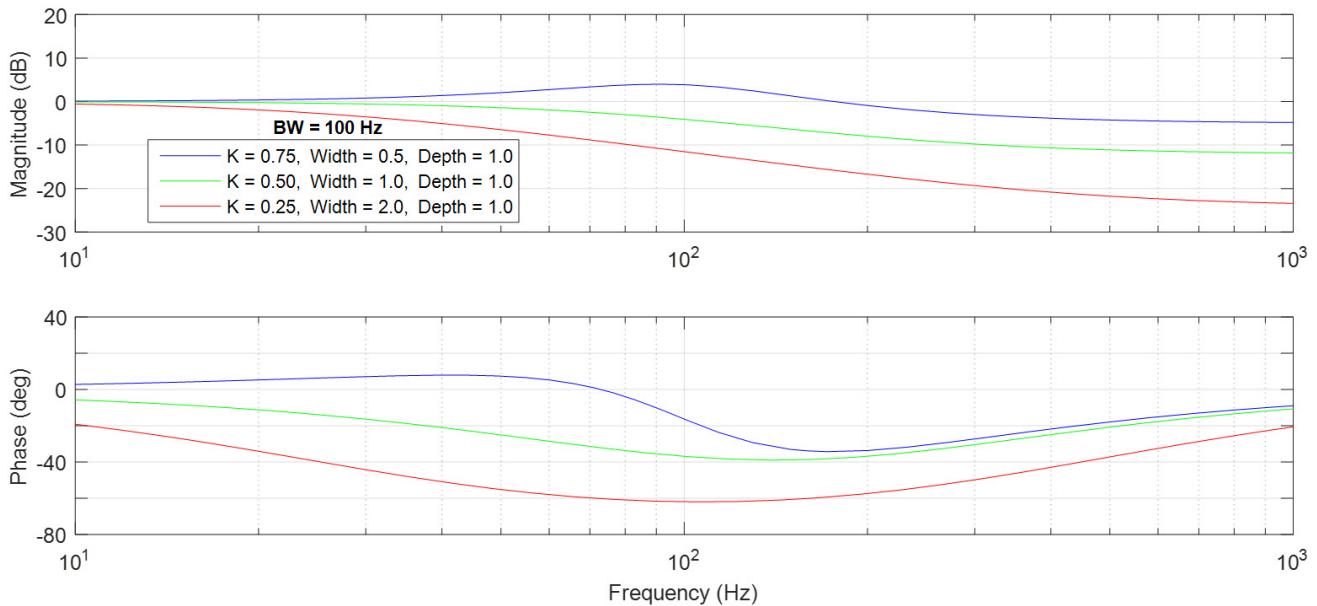
Configured as a Second Order Low Pass Filter ( $K = 0$ )

When  $K = 0$ , the notch filter is configured to operate as a second order low-pass filter with bandwidth  $F$  and damping  $Z_W$ . Bode plots with differing low-pass filter settings for  $K = 0$  are shown in [Figure 62](#).

**Figure 62 - Low Pass Filter Configurations**

*Configured as a Second Order Lag Lead Filter ( $0 < K < 1$ )*

When  $0 < K < 1$ , the notch filter is configured to operate as a second order lag-lead filter. It has two user configurable poles (lag) at  $F$  with damping  $Z_W$  and two user configurable zeros (lead) at  $F_D = F/K$  with damping  $Z_D$ . Bode plots with differing lag-lead filter settings for  $0 < K < 1$  are shown in [Figure 63](#).

**Figure 63 - Lag-Lead Filter Configurations**

This configuration can be used to compensate for compliant load mechanics by placing two poles at the anti-resonant frequency and two zeros at the resonant frequency using the following calculations.

$$K = \frac{1}{\sqrt{R + 1}}$$

$$F = K \times F_R$$

$$Z_W = \frac{B_T}{4\pi R J_M F}$$

$$Z_D = Z_W \sqrt{R + 1}$$

The following definitions are given.

$J_M$  = Motor Inertia in units of [kg m<sup>2</sup>]

R = Load Ratio

$F_R$  = an audible resonant frequency in units of [Hz] measured by a mobile application, such as iAnalyzer Lite.

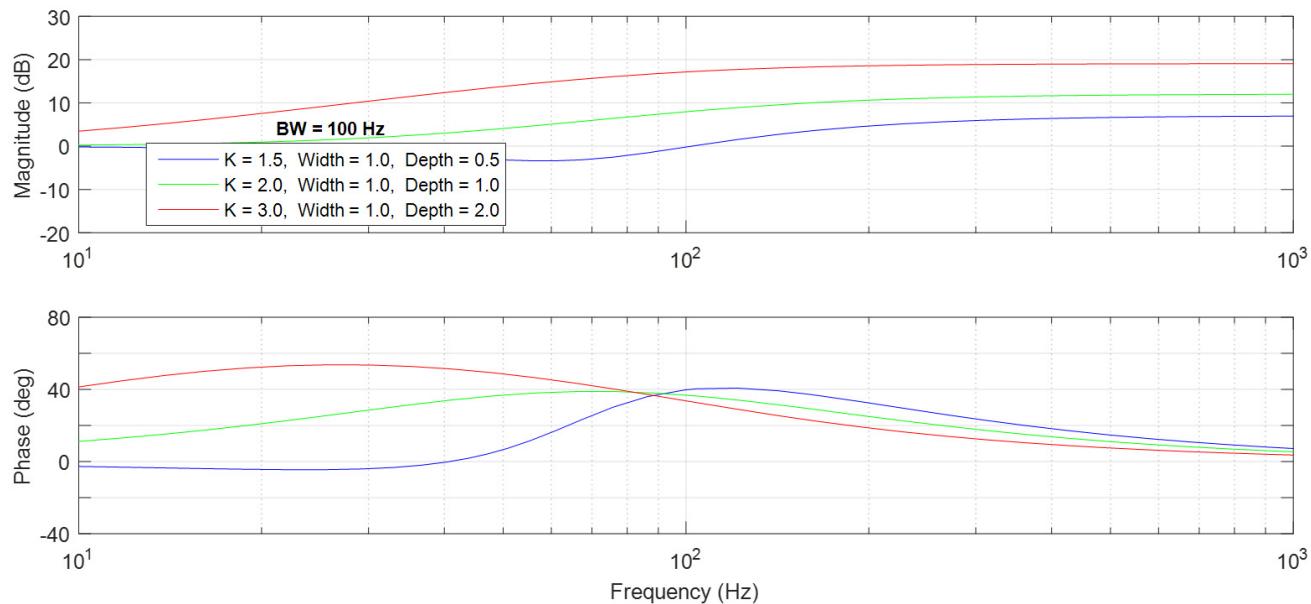
$B_T$  = the total mechanical friction in units of [Nm sec/rad]. This value is typically unknown and can be manually tuned. An initial value for a 580J motor is 0.01.

However, using the filter in this way can make the drive more sensitive to disturbances. Furthermore, the load observer is recommended to compensate for load compliance and disturbances because it typically does a better job without having to determine these parameters.

#### Configured as a Second Order Lead Lag Filter ( $K > 1$ )

When  $K > 1$ , the notch filter is configured to operate as a second order lead-lag filter. It has two user-configurable zeros (lead) at  $F_D = F/K$  with damping  $Z_D$ , and two user configurable poles (lag) at  $F$  with damping  $Z_W$ . Bode plots with differing lead-lag filter settings for  $K > 1$  are shown in [Figure 64](#).

**Figure 64 - Lead-Lag Filter Configurations**

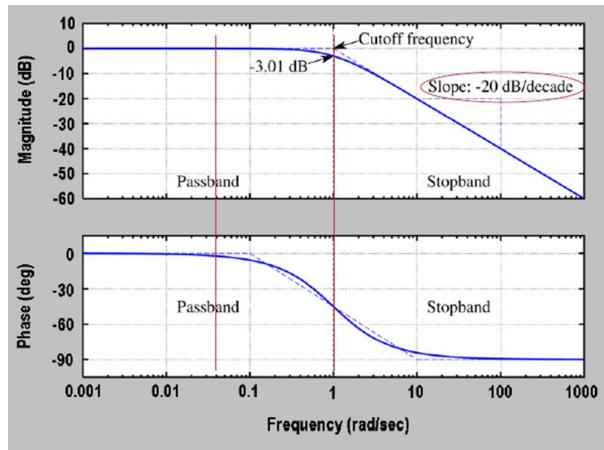


Initial recommended settings for damping are  $Z_W = 1$  and  $Z_D = 1$ . Lead-Lag filters have been used to boost velocity or acceleration loop bandwidth in order to improve performance.

## Low Pass Filter

This section applies to the torque low pass filters in Kinetix drives. A low pass filter passes low frequency signals but attenuates signals above the filter bandwidth. [Figure 65](#) shows gain and phase through the filter as a function of the frequency content of a signal passing through it.

**Figure 65 - First Order Low-pass Filter Bode Plot**



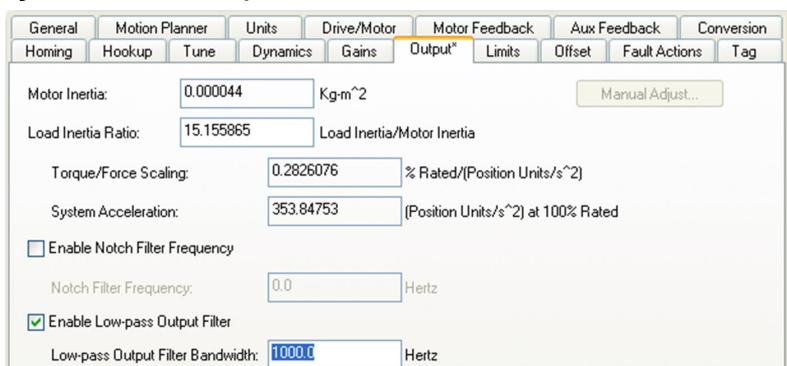
The signal is attenuated 3 dB at the filter bandwidth, also known as the cutoff frequency. The low pass filter is first order and has an attenuation of 20 dB/decade beyond the cutoff frequency.

As with any filter, its output is shifted in phase from the input. If the filter bandwidth is set too low when the filter is enabled, the filter's phase lag adds to the stack up of delays around the velocity loop and causes instability. This occurs when the filter bandwidth approaches either the velocity loop bandwidth or the load observer bandwidth when load observer is enabled, whichever bandwidth is highest. In this case, the low pass filter must either be disabled, its bandwidth increased, or the control loop gains decreased. For more information on setting the low pass filter, see [Compensating for High Frequency Resonances on page 63](#).

## Sercos Drives

For a Sercos drive, click the Output tab in the Axis Properties dialog box. Check the Enable Low-pass Output Filter box and set the Low-pass Output Filter Bandwidth to the desired frequency in Hz.

**Figure 66 - Sercos Low-pass Filter Bandwidth**

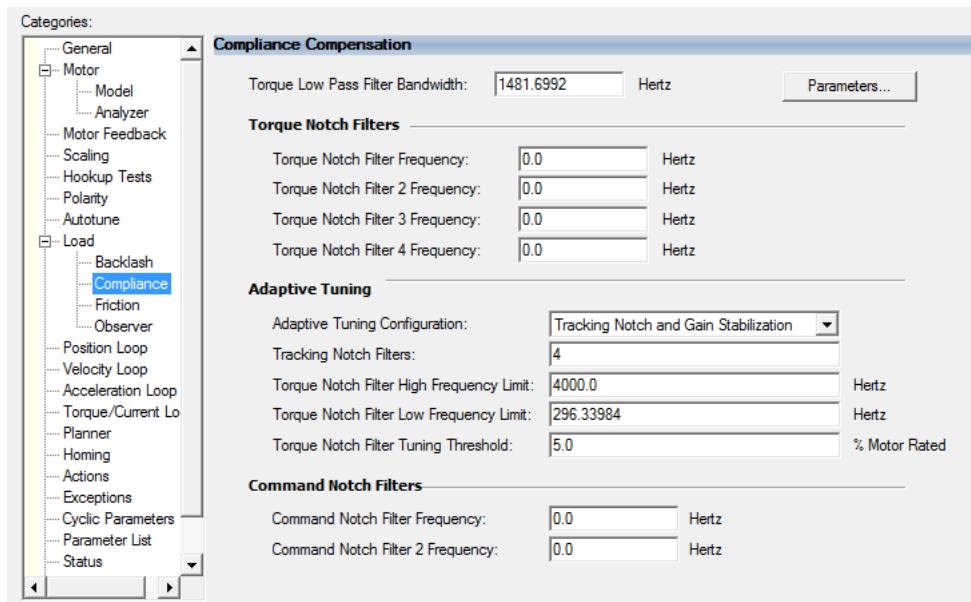


If the Low-pass Output Filter is enabled and load observer is not enabled, verify that the Low-pass Output Filter Bandwidth  $\geq K_{VP} / (2\pi) \times 5$ . If the Low-pass Output Filter and load observer are enabled, verify that the Low-pass Output Filter Bandwidth  $\geq \max(K_{VP}, K_{OP}) / (2\pi) \times 5$ . Sercos IDN P-065 has an impact on how the Low-pass Output Filter functions. See [Table 5 on page 20](#) for more information.

## EtherNet/IP Drives

For an EtherNet/IP drive, click the Compliance tab in the Axis Properties dialog box. Set the Torque Low Pass Filter Bandwidth to the desired frequency in Hz.

**Figure 67 - EtherNet/IP Low Pass Filter Bandwidth**



If the Torque Low Pass Output Filter is enabled and load observer is not enabled, verify that the Torque Low Pass Output Filter Bandwidth  $\geq K_{VP} \times 5$ . If the Torque Low Pass Output Filter and load observer are enabled, verify that the Torque Low Pass Output Filter Bandwidth  $\geq \max(K_{VP}, K_{OP}) \times 5$ .

## EtherNet/IP Drive Lag Filter

EtherNet/IP drives provide a lead-lag filter with the following first order transfer function.

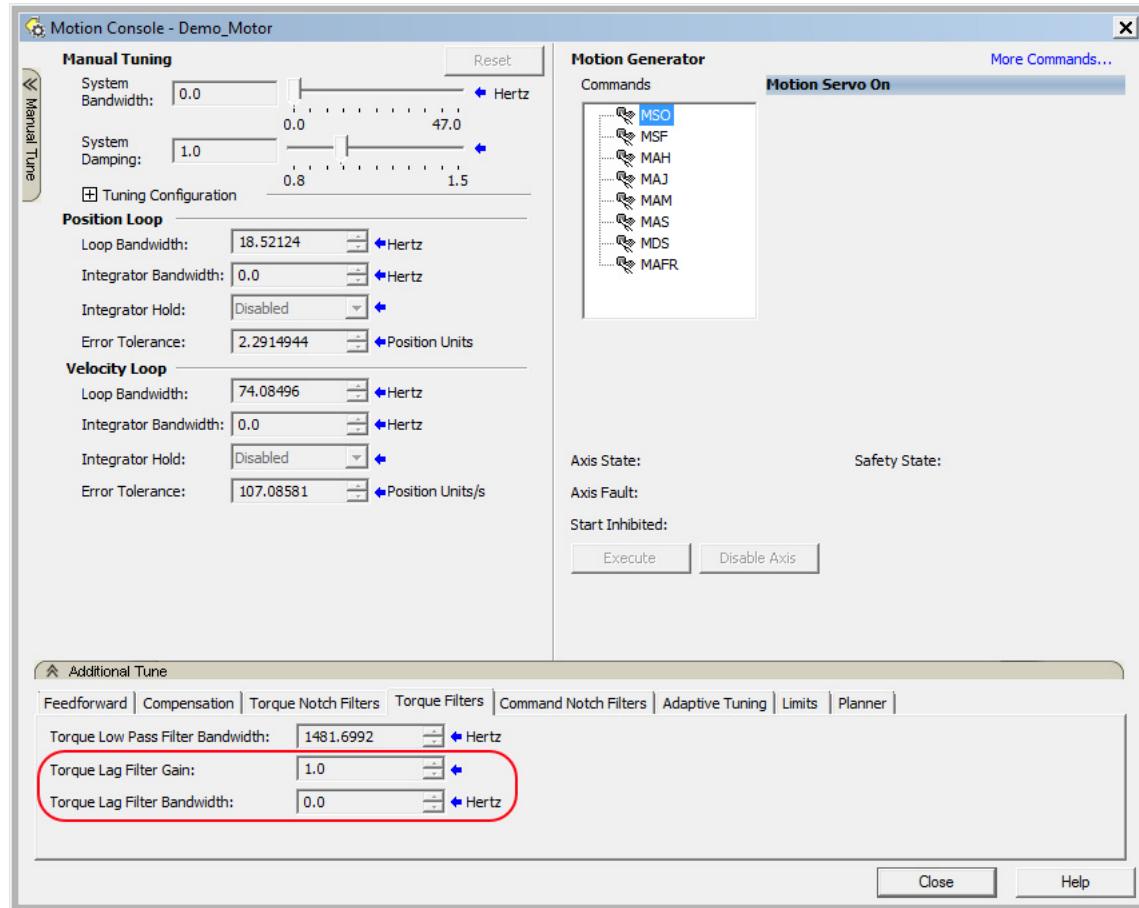
$$G(s) = \frac{Ks + 2\pi F}{s + 2\pi F} = K \frac{s + 2\pi F / K}{s + 2\pi F}$$

The filter has a DC gain of 1, a user-configurable pole (lag) at  $F$ , and a user-configurable zero (lead) at  $F/K$ . User-configurable parameters are given below.

$K$  = Torque Lag Filter Gain

$F$  = Torque Lag Filter Bandwidth in Hz

In the Logix Designer application version 32 and earlier, click the Compliance tab in the Axis Properties dialog box and click the Parameters button to configure the filter. In the Logix Designer application version 33 and later, the lag filter is configured in the Torque Filters tab of the manual tune window.

**Figure 68 - EtherNet/IP Drive Lag Filter Parameters**

There are four modes for this first-order lag filter:

- When  $K = 0$ , the filter is a low-pass filter.
- When  $0 < K < 1$ , the filter is a lag-lead. The initial use case for the filter was to use it in this mode to compensate for high frequency gain boost associated with compliant load mechanics. With a known load ratio ( $R$ ) and resonant frequency ( $F_r$ ), a pole is placed at the anti-resonant frequency and a zero is placed at the resonant frequency using the following calculations:

$$K = 1/\sqrt{R+1}$$

$$F = K \times F_r$$

However, using this filter can make the drive more sensitive to disturbances. We recommend using the Load Observer with Velocity Estimate mode to compensate for load compliance and disturbances.

- When  $K=1$ , the filter is off. We recommend this mode of operation.
- When  $K>1$ , the filter is a lead-lag. This has been used to boost velocity or acceleration loop bandwidth.

## Friction Compensation

Friction compensation applies a compensating directional torque or force to the motor to overcome the effects of friction in a mechanical system, thus reducing the control effort required. Individual attributes have been defined to support compensation for static friction, sliding (Coulomb) friction, and viscous friction. A compensation window attribute is also provided to mitigate motor dithering associated with conventional friction compensation methods.

## Static Friction

It is not unusual for a mechanical load to have enough static friction, commonly called ‘sticktion’, where the mechanical load refuses to move, even with a significant position error. Position integral gain can be used to generate enough drive output to correct the error, but this approach may not be responsive enough for some applications. An alternative is to use Static Friction Compensation to break the sticktion in the presence of a non-zero position error. This is done by adding or subtracting a fixed torque level, as determined by the Static Friction Compensation attribute, to the torque reference signal value based on its current sign. This form of friction compensation is applied only when the drive is static, that is, when there is no change in the position command.

The Static Friction Compensation value must be just under the value to overcome the sticktion. A larger value results in ‘dither’, a phenomenon describing a rapid back and forth motion of the load centered on the commanded position as it overcompensates for the sticktion.

To address the issue of dither when applying Static Friction Compensation, a Friction Compensation Window is applied around the current command position when the load is at rest. If the actual position error is within the Friction Compensation Window, the Static Friction Compensation value is applied to the servo drive output but scaled by the ratio of the position error signal to the Friction Compensation Window. Within the window, the position loop and velocity loop integrators are also disabled to avoid the hunting effect that occurs when the integrators wind up. Thus, once the position error reaches or exceeds the value of the Friction Compensation Window attribute, the full Static Friction Compensation value is applied. A Friction Compensation Window value of zero effectively disables this feature.

A non-zero Friction Compensation Window effectively softens the Static Friction Compensation as it is applied to the torque reference and reduces the dithering and hunting effects that it can create. This feature generally lets higher values of Static Friction Compensation to be applied, resulting in better point-to-point positioning.

## Sliding Friction

Sliding friction or Coulomb friction, by definition, is the component of friction that is independent of speed as long as the mechanical load is moving. Sliding friction is always less than static friction for a given mechanical system. The method of compensating for sliding friction is basically the same as that for static friction, but the torque level added to the torque reference signal is less than that applied to overcome static friction and is determined by the Sliding Friction Compensation attribute. Sliding Friction Compensation is applied only when the motor is being commanded to move.

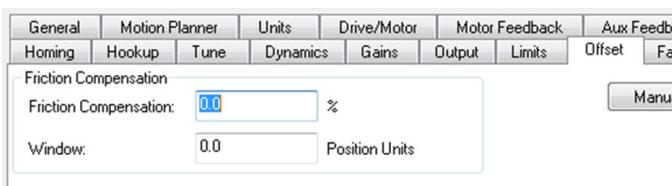
## Viscous Friction

Viscous friction is defined as the component of friction that increases linearly with the speed of the mechanical system. The method of compensating for viscous friction is to multiply the configured Viscous Friction Compensation value by the speed of the motor and apply the result to the torque reference signal. Viscous Friction Compensation is applied only when the motor is being commanded to move.

### Sercos Drives

For a Sercos drive, the friction-related values are located in the Offset tab of the Axis Properties dialog box in the Studio 5000 Logix Designer® application.

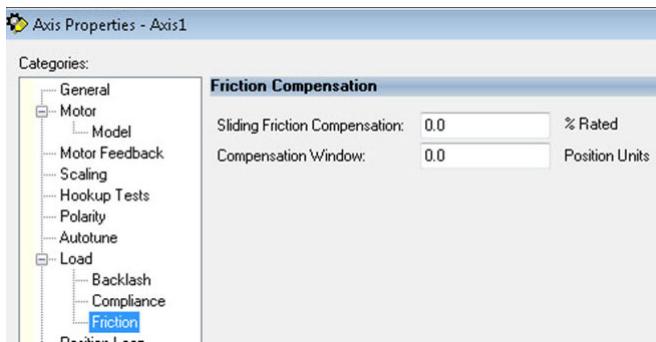
**Figure 69 - Sercos Drive Friction Compensation Attributes**



## EtherNet/IP Drives

For an EtherNet/IP drive, the friction-related values are located on the Load tab of the Axis Properties dialog box in the Logix Designer application.

**Figure 70 - EtherNet/IP Drive Friction Compensation Attributes**

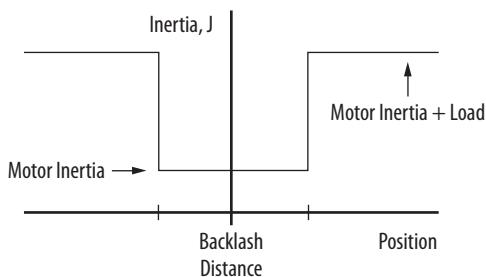


## Backlash Compensation

A number of important compensation features are included in the torque control diagram. Backlash Compensation is used to stabilize the device control loop behavior in applications with high load inertia ratios and mechanical backlash.

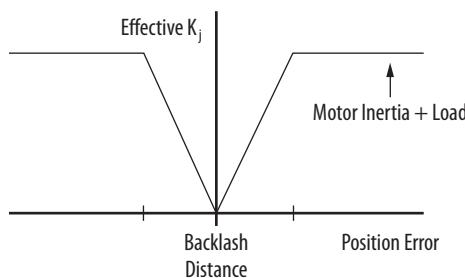
The Backlash Compensation Window attribute is used to control the Backlash Compensation feature. Mechanical backlash is a common problem in applications that use gearboxes and inexpensive couplings. This problem happens because until the input gear is turned to the point where its proximal tooth contacts an adjacent tooth of the output gear, the reflected inertia of the output is not felt at the motor. In other words, when the gear teeth are not engaged, the system inertia is reduced to the motor inertia.

**Figure 71 - Representation of Backlash**



If the velocity control loop is tuned for peak performance with the load applied, the control loops are under-damped at best and unstable at worst in the condition where the gear teeth are not engaged. In the worst case scenario, the motor and the input gear oscillate wildly between the limits imposed by the output gear teeth. The result is a loud buzzing sound, commonly referred to as 'gearbox chatter', when the load is at rest. If this situation persists, the gearbox will prematurely wear out. To prevent this condition, the conventional approach is to detune the velocity loop so that the axis is stable without the gearbox load applied. Unfortunately, system performance suffers.

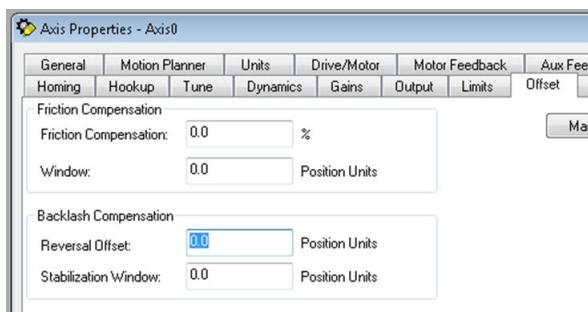
With a Backlash Stabilization Window value commensurate with the amount of backlash in the mechanical system, the backlash stabilization algorithm is very effective in eliminating backlash-induced instability while still maintaining full system bandwidth. The key to this algorithm is a tapered  $K_J$  profile, shown in [Figure 72](#), that is a function of the position error. The reason for the tapered profile, as opposed to a step profile, is that when the position error exceeds the backlash distance, a step profile creates a very large discontinuity in the torque output. This repulsing torque tends to slam the motor back against the opposite gear tooth and perpetuates the buzzing effect. The tapered profile can be qualified to run when only the acceleration command or the velocity command to the control loop structure is zero, that is, when not commanding motion that engages the teeth of the gearbox.

**Figure 72 - Backlash Stabilization Window**

Properly configured with a suitable value for the Backlash Compensation Window, this algorithm entirely eliminates the gearbox buzz without sacrificing any servo performance. The Backlash Compensation Window parameter determines the width of the window over which backlash stabilization is applied. In general, this value is set equal to or greater than the measured backlash distance. A Backlash Stabilization Window value of zero effectively disables this feature.

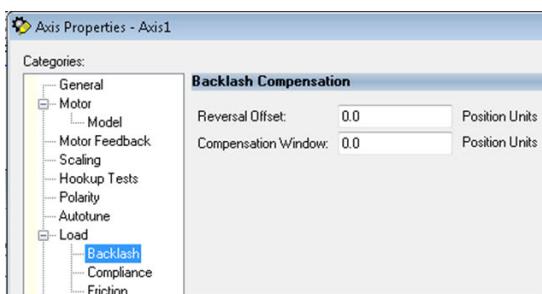
## Sercos Drives

For a Sercos drive, the backlash related values are located on the Load tab of the Axis Properties dialog in the Logix Designer application.

**Figure 73 - Sercos Drive Backlash Compensation Attributes**

## EtherNet/IP Drives

For an EtherNet/IP drive, the backlash-related values are located on the Load tab of the Axis Properties dialog box in the Logix Designer application.

**Figure 74 - EtherNet/IP Drive Backlash Compensation Attributes**

## Kinetix 300 Drive Tuning

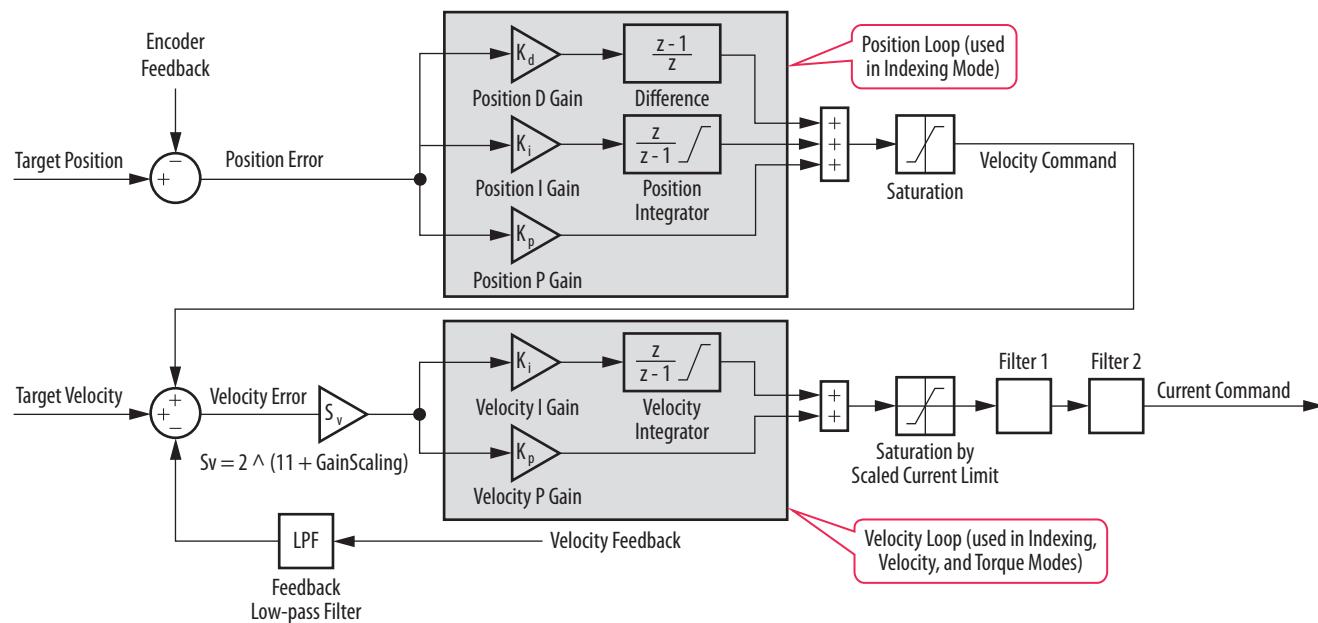
This chapter provides instructions to perform an autotune and explains the Kinetix® 300 drive control loops, filters, and how to use them. The autotune method described below works well for most applications.

Topic	Page
Control Loops	79
Autotune	79
Manual Tune	82
Filters	85

## Control Loops

The control loops within a Kinetix 300 servo drive are shown in the following image.

Figure 75 - Kinetix 300 Drive Control Loops



## Autotune

Depending on the load, it is generally beneficial to autotune without the load attached to achieve good control of the motor. This provides a stable motor before the load is attached in the case of a compliant load type.

## Tuning Method

Follow these steps to perform an autotune on a Kinetix 300 drive.

1. Type in the IP address of the drive in Internet Explorer software and in the General dialog box.
2. Set the drive mode to Auto Tune.



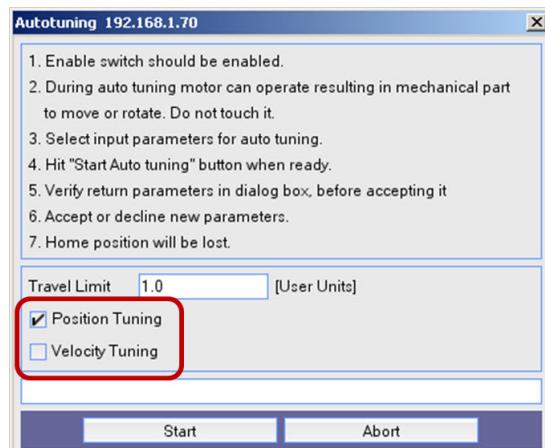
3. Enable the drive.
4. Click the Autotuning window.



5. Enter the Travel Limit.

This is the maximum Travel Limit (the motor does not necessarily move this distance).

6. Check the appropriate Tuning box depending on the drive mode.

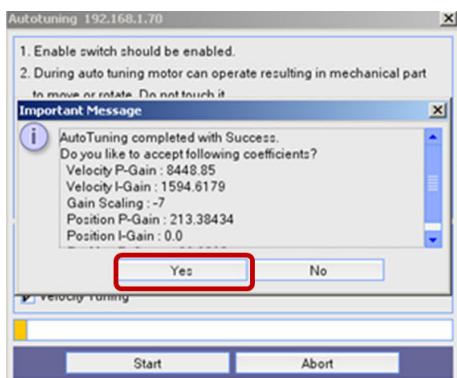


- If using an indexing mode or a positioning mode, check Position Tuning.
- If using a velocity mode, check Velocity Tuning.

7. Click Start.

**IMPORTANT** If the drive has not been enabled before this step (new installation), verify that you have safeguards in place to safely remove power from the drive in case of an unstable situation where the drive can produce undesired motion.

8. Click Yes to accept the tuned gains.



9. Enable the drive and use the drive mode to determine if the drive is stable.

Some values that you can modify are described below and can be accessed from the Dynamics dialog box.

Description	Value	Units	Min
Velocity P-Gain	3165.8632812		0.0000
Velocity I-Gain	184.2586975		0.0000
Position P-Gain	2192.2597656		0.0000
Position I-Gain	0.0000		0.0000
Position D-Gain	359.1798706		0.0000
Position I-Limit	0.0000		0.0000
Gain Scaling	-10	RPM	-16

Enable Velocity Integrator in Position Mode

Feedback Filter

Type: Off

Feedback Filter Time Constant: 2.0000 ms

Filter 1

Type: Off

Filter 2

Type: Off

- Gain values are loaded by default and are based on the motor nameplate and drive size. These values can change after an autotune is performed.
- Gain Scaling - every integer increase (this value is negative) **doubles** the system bandwidth. Every integer decrease **halves** the system bandwidth.
- Enable Velocity Integrator in Position Mode checkbox - this enables the  $K_{V_I}$  gain when checked. Experience suggests not enabling this in a positioning type application. However, if tracking during an index is required, then enabling  $K_{V_I}$  lets the servo drive track better during commanded motion.

## How It Works

It is important to note that the autotune function is not the same as the bump test in the Studio 5000 Logix Designer® application.

The Kinetix 300 drive autotune sequence works as follows:

- Several bi-polar steps are sent out and the response is measured.
  - Current steps are used in velocity mode.
  - Velocity steps are used in position mode.
- The PID gains are calculated based on the response.
- If you chose both velocity and position response, the shape and number of steps are quite different as it is possible both loops are not tuned.

**Figure 76 - Scope Trace of Step Pulses During Autotune**



## Manual Tune

After autotune completes successfully, the servo drive has relatively good performance. If you require better performance, follow these steps to manually tune the Kinetix 300 drive.

1. If possible, generate a bidirectional index command to the drive to test out the gains.

You can make manual adjustments to the gain values from the Dynamics tab.

The index can be unidirectional if the load cannot have bidirectional movement.

2. For steps 3 and 4, disable the drive before making any changes.

Gain changes with the drive enabled can cause the motor to go unstable.

3. Change the Gain Scaling value.

Using this value is similar to changing the system bandwidth described in previous sections with EtherNet/IP drives. The Gain Scaling is an integer and is set knowing these factors:

- Every integer increase (the Gain Scaling value is negative) effectively **doubles** the system bandwidth.
  - Every integer decrease **halves** the system bandwidth.
4. If the drive is in position mode, use the Velocity Integrator or Position Integrator to improve tracking error.

Start with a small amount and increase it while monitoring the velocity for overshoot and stability. A good starting point is 2% of the  $K_{Vp}$  or  $K_{Pp}$  gains.

5. Increase the index speed, acceleration, and deceleration to meet the application requirements.

If you have significant overshoot or instability, repeat [step 3](#). The response shown in the figure below is acceptable for our application.



At this point, there should be good response with an unloaded motor executing the required profile for the application.

6. Disable the drive and attach the load.
7. Enable the drive.
8. Slowly increase the dynamics of the index while you observe how the load responds.

If the load is fairly rigid and tightly coupled to the motor, the drive will still respond well. If the load is not responding well, perform an autotune with the load attached.

Any compliance in the load yields gain values that can be incorrect and misleading.

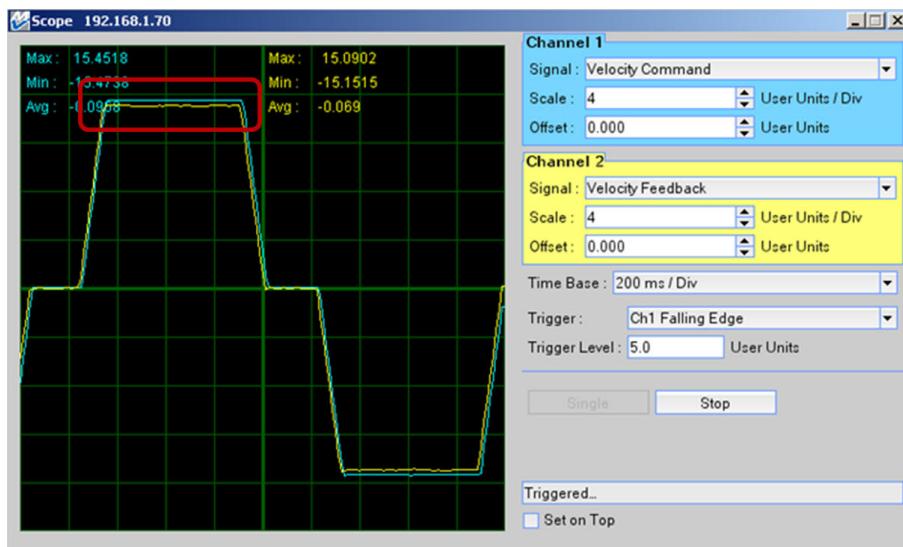
9. Depending on the drive mode, you can also continue at this point with the steps defined in [Chapter 4](#) for inside-out velocity and position loop tuning.

Also see [Compensating for High Frequency Resonances on page 63](#).

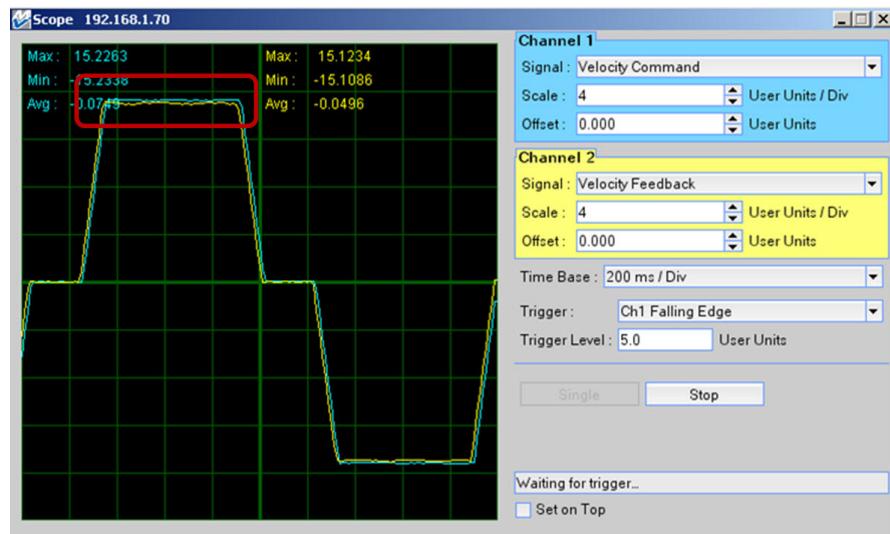
## Manual Tune Example

This example describes changing the Gain Scaling and observing the results.

- Autotuning without a load results in a stable index. The Gain Scaling is -7 and there is tracking error present during the index. This tracking error is shown in the following image.



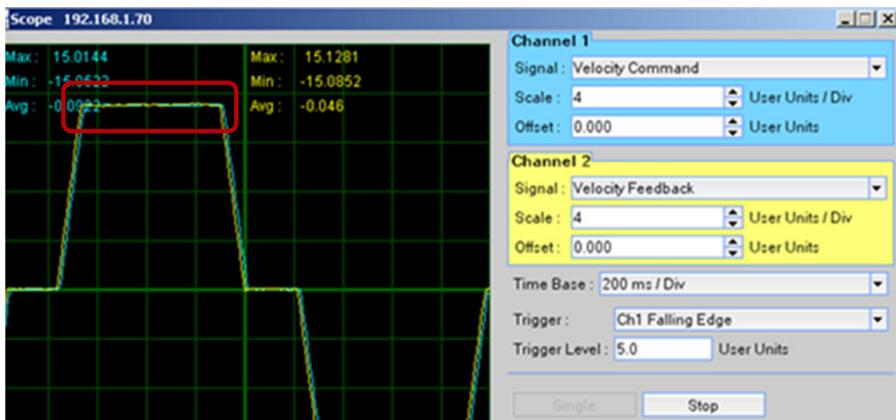
- The drive is disabled and the Gain Scaling is changed to -6. The changed tracking error is shown in the following scope trace.



- The Gain Scaling is changed to -5 and the motor becomes unstable.
- To improve the tracking error, select the Enable Velocity Integrator in Position Mode checkbox.

Enable Velocity Integrator In Position Mode

- The tracking error is minimal and the move profile is being followed as closely as possible. The tracking error is shown in the following image.

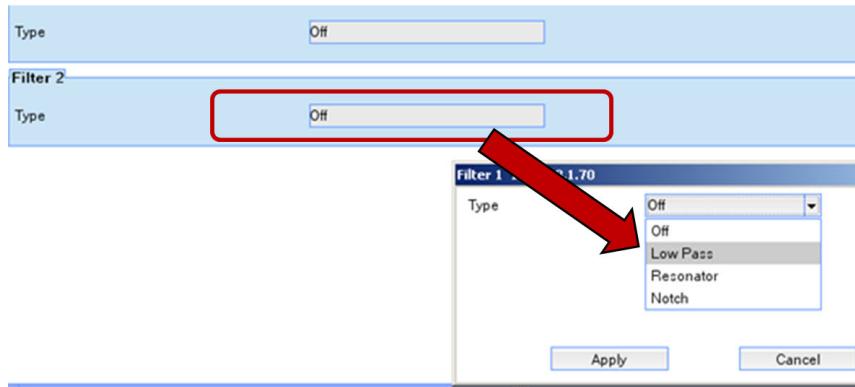


## Filters

The filters described below are not modified during autotune.

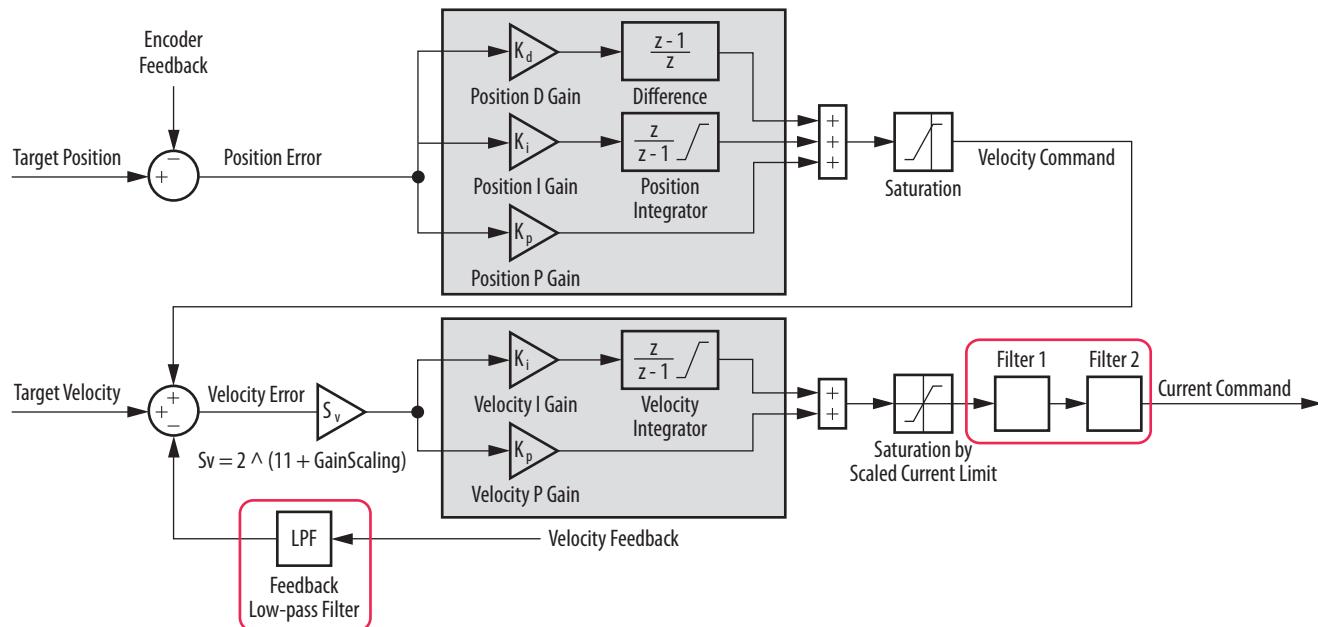
- The two filters can be used together with Filter 1 in series with Filter 2.
- The filters can be the same filter type with different values.
- See [Compensating for High Frequency Resonances on page 63](#) to help determine the resonant frequency of the load.

**Figure 77 - Accessing Filters from MotionView Software**



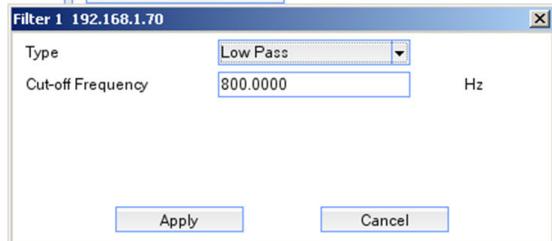
## Feedback Filter and Low Pass Filter

**Figure 78 - Filter Locations within the Control Loops**



- The Low-pass filters (Filter 1/Filter 2) can be used alone or together to accommodate different application resonance issues.

**Figure 79 - Low-pass Filter Selection**



- The Low-pass filter is a second order bi-quad filter that is used to filter resonant frequencies greater than the control loop bandwidth.
- The Feedback Low-pass filter (LPF) is a first order filter that is used to reject high frequencies coming from noisy feedback sources.

## Resonator and Notch Filter

Filter 1 and/or Filter 2 can be configured as a notch or resonant type filter. Both configurations implement band-stop filtering to solve certain mechanical compliance problems.

The resonator filter has a slightly different transfer function than the notch and guarantees the depth of attenuation. Therefore, the roll off will be different depending on the gain factor.

The notch filter has a fixed roll off, and the attenuation gain depends on the attenuation bandwidth (width of the notch).

A common problem is torsional resonance due to mechanical compliance between load inertia and motor inertia. Consider a long load shaft with an inert load at the opposite end. Such a system has a resonant frequency defined by the equation in [Figure 80](#).

**Figure 80 - Equation to Determine Resonant Frequency**

$$f_r = \frac{1}{2\pi} \sqrt{\frac{k_s}{J_p}} \quad [\text{Hz}]$$

where:  $J_l$  = load inertia [ $\text{kgm}^2$ ]

$J_m$  = motor inertia [ $\text{kgm}^2$ ]

$k_s$  = total stiffness of coupling and shaft [Nm/rad]

$J_p = (J_m \times J_l) / (J_m + J_l)$

$\pi = 3.1416$

Applying the resonant frequency in this configuration will cancel the resonance, effectively allowing higher overall gain.

The resonant filter lets the center frequency and gain to be set independently.

## Notes:

## Create Move Profiles for Tuning

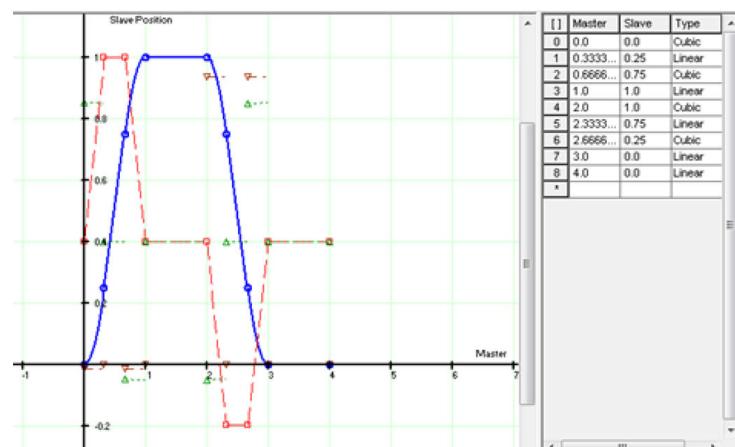
This appendix shows examples of creating move profiles in the Studio 5000 Logix Designer® application to observe the behavior of an axis while tuning by using motion direct commands, or JOG, MAM, or MATC instructions. See the Logix 5000™ Controllers Motion Instruction Reference Manual, publication MOTION-RM002, for more information on these instructions. Depending on the application, the move profile can be unidirectional or bidirectional.

Topic	Page
Example 1	89
Example 2	90
Example 3	91

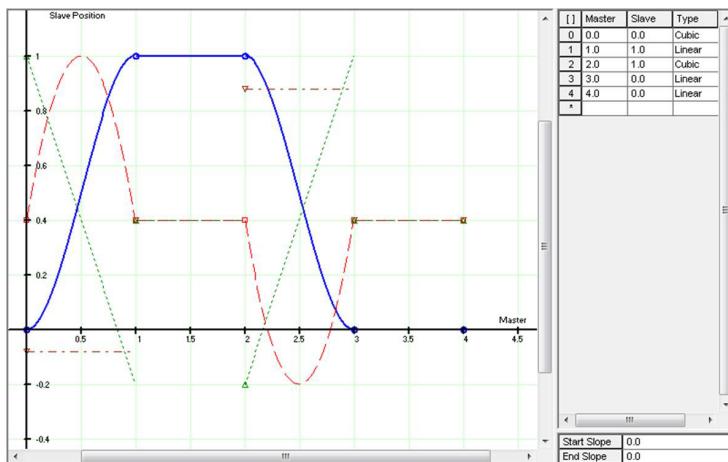
### Example 1

An example CAM table is shown that rotates the motor shaft back and forth 1 revolution in 4 seconds.

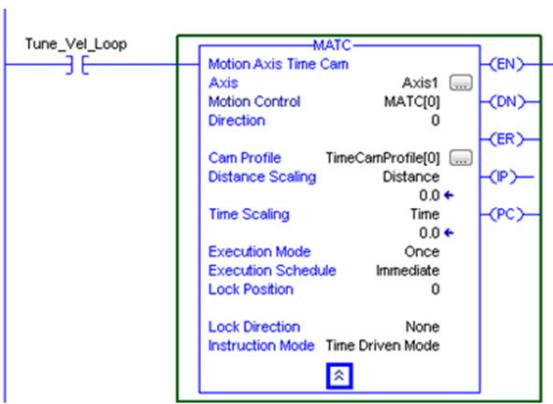
**Figure 81 - CAM Example, Part**



This produces a velocity trapezoid profile with high jerk at the trapezoid corners. If the machine mechanics are not robust enough to handle momentary high jerk transients, a CAM table is shown that produces smoother motion.

**Figure 82 - CAM Example, Part 2**

The CAM table above is used in an MATC instruction that executes physical motion of the motor.

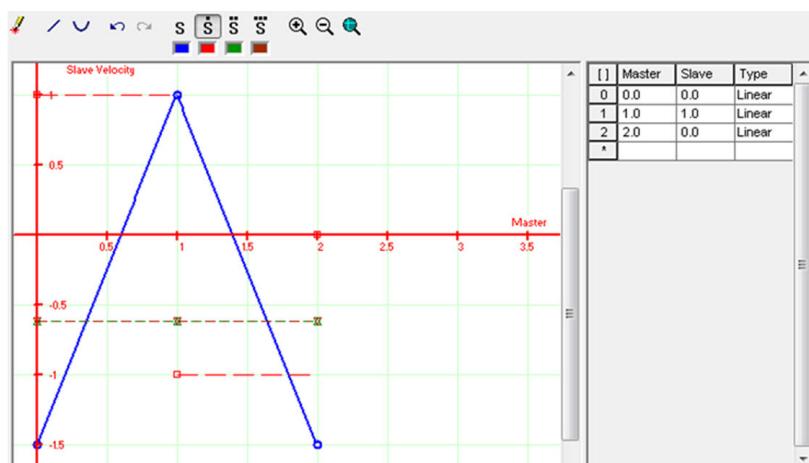
**Figure 83 - CAM Example, MATC Instruction**

Use scaling in the CAM editor to assure the CAM produces motion approaching, but not beyond, the dynamic travel limits of the machine:

- Use Distance Scaling to increase or decrease the travel distance from 1 revolution.
- Use Time Scaling to increase or decrease the period from 4 seconds.
- Set Execution Mode to 'Once' or 'Continuously' depending on the number of times you want to safely execute the move.

## Example 2

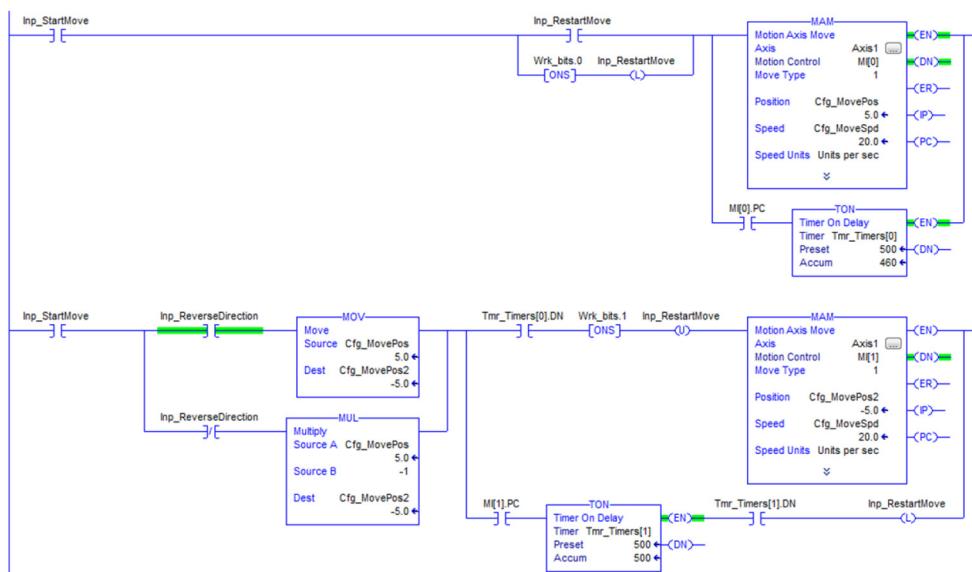
An example CAM table is shown that produces a velocity step command for tuning the velocity loop. This is the most aggressive motion profile you can create. As a result, we recommend that you use this one only if you are experienced in manual tuning, and the machine mechanics are robust enough to handle the high jerk.

**Figure 84 - CAM Example, Velocity Loop**

Use Distance Scaling to change the velocity step magnitude. Use Time Scaling to change the time period.

## Example 3

This is an example to perform a MAM instruction that rotates the motor shaft back and forth. If your application cannot perform a bidirectional move, then sequence a unidirectional move. An example for a bidirectional move is shown below.

**Figure 85 - Bidirectional Move**

You can also produce a velocity step with a MAM instruction by entering a very high Accel Rate and Decel Rate. However, be aware of the following things:

- Logix Designer application, version 19.00 and earlier - The default values for Speed Units, Accel Units, and Decel Units = [seconds], meaning that an Accel Rate = 0 seconds is a very high acceleration and produces a velocity step.
- Logix Designer application, version 20.00 and later - The default value for Speed Units = [units/second] and the default values for Accel Units and Decel Units = [units/second<sup>2</sup>], meaning that an Accel Rate = 1,000,000 units/second<sup>2</sup> is a very high acceleration and produces a velocity step.

## Notes:

## Create Trends for Tuning

This appendix shows how to configure different Trends in the Studio 5000 Logix Designer® application to observe the behavior of an axis while tuning.

Topic	Page
RSTrendX Properties	93
Value Bar Information	98

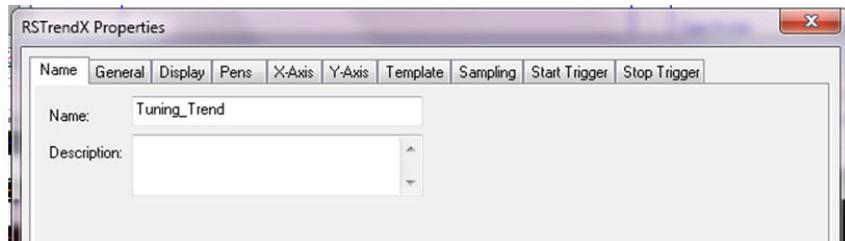
### RSTrendX Properties

The RSTrendX Properties dialog box provides access to attributes you can monitor or configure that affect the tuning of the axis.

#### Name Tab

The Name tab is where you can change the name of the trend itself, which is useful when you copy and paste a trend.

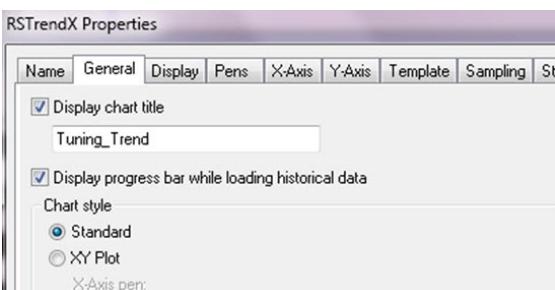
**Figure 86 - Name Tab**



#### General Tab

An important part of the General tab is that the trend is typically set against time, like an oscilloscope. You can have an X-Y plot that is useful when dealing with coordinated systems, like robots.

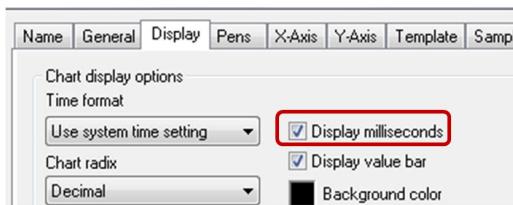
**Figure 87 - General Tab**



## Display Tab

Displaying milliseconds is useful when using the value bar of the trend. The other settings are defaults.

**Figure 88 - Display Tab**



## Pens Tab

The Pens tab is the most important tab in the RSTrendX Properties dialog box. You need to select the correct pens to use in your trend.

Click Add/Configure Tags to add and remove pens.

### Sercos Drives

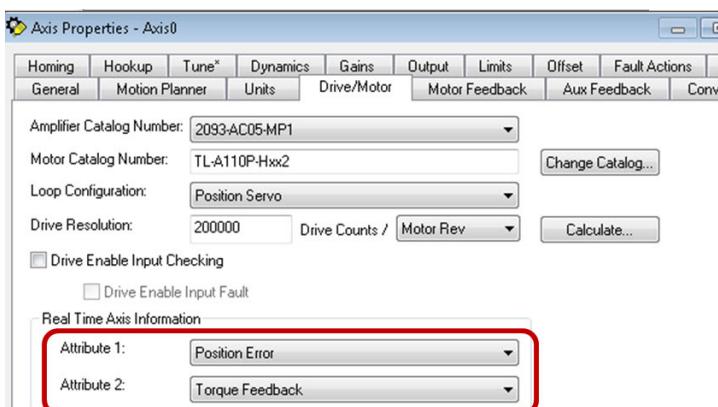
Typical pens used in tuning a Sercos drive are shown in [Figure 89](#).

**Figure 89 - Pens Tab**

	Tag\Expr.	Color	Visible	Width	Type	Style
1	Axis0.ActualPosition	Blue	On	1	Analog	.....
2	Axis0.ActualVelocity	Red	On	1	Analog	.....
3	Axis0.CommandVelocity	Red	On	1	Analog	.....
4	Axis0.PositionError	Magenta	On	1	Analog	.....
5	Axis0.TorqueFeedback	Black	On	1	Analog	.....
6	Axis0.VelocityError	Yellow	On	1	Analog	.....

Some of these values show zero if they are not set as real-time attributes. Select the parameters as shown in [Figure 90](#).

**Figure 90 - Axis Properties**



Notice that you can use only two real-time attributes at one time with a Sercos drive. This means that if you want to monitor Velocity Error at the same time as Position Error and Torque Feedback, you need to run two trends simultaneously and modify the real-time values on the fly as you require them.

## EtherNet/IP Drives

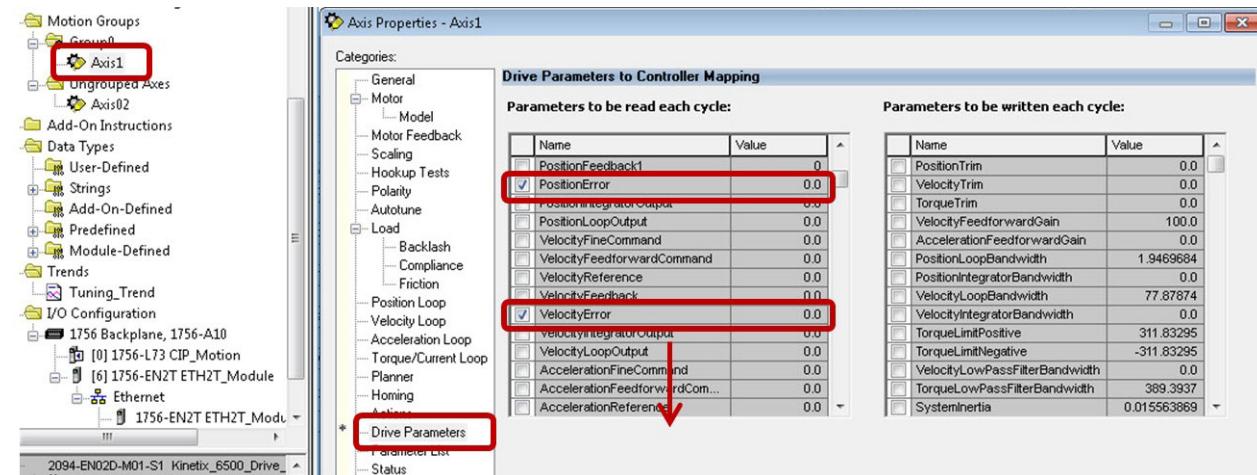
Typical pens used in tuning an EtherNet/IP drive are shown in [Figure 91](#).

**Figure 91 - Pens Tab**

Name	General	Display	Pens	X-Axis	Y-Axis	Template	Sampling	Start Trigger	Stop
<b>Pen Attributes</b>									
1	Axis1.ActualPosition	Blue	On	1	Analog	.....	.....	.....	.....
2	Axis1.ActualVelocity	Orange	On	1	Analog	.....	.....	.....	.....
3	Axis1.CommandVelocity	Red	On	1	Analog	.....	.....	.....	.....
4	Axis1.PositionError	Magenta	On	1	Analog	.....	.....	.....	.....
5	Axis1.VelocityError	Green	On	1	Analog	.....	.....	.....	.....
6	Axis1.CurrentFeedback	Yellow	On	1	Analog	.....	.....	.....	.....

Some of these values show zero if they are not set to be read each cycle. Select the Drive Parameters tab in the Axis Properties dialog box and check the parameters that you created pens for. You may need to scroll down to access all the values required.

**Figure 92 - Drive Properties to Control Mapping**



## Useful Pens Tab Tips

Here are some useful tips about the Pens tab:

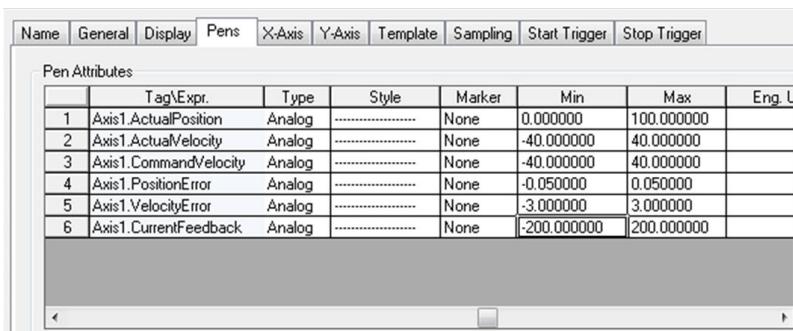
- You can turn the pen on/off by clicking 'On' in the visible tab, which is useful when the trend window is too busy with several pens.

Name	General	Display	Pens	X-Axis	Y-Axis	Template	Sampling	Start Trigger	Stop
<b>Pen Attributes</b>									
1	Axis1.ActualPosition	Blue	Off	1	Analog	.....	.....	.....	.....
2	Axis1.ActualVelocity	Orange	On	1	Analog	.....	.....	.....	.....
3	Axis1.CommandVelocity	Red	On	1	Analog	.....	.....	.....	.....
4	Axis1.PositionError	Magenta	On	1	Analog	.....	.....	.....	.....

- If you are trending a digital signal, that is, a start cycle, stop, e-stop button, and so forth, you can remove the rise and fall slope due to the sampling of the trend and change the Type from Analog to Digital.

Name	General	Display	Pens	X-Axis	Y-Axis	Template	Sampling	Start Trigger	Stop
<b>Pen Attributes</b>									
1	Axis1.ActualPosition	Off	1	Analog	.....	None	.....	.....	.....
2	Axis1.ActualVelocity	On	1	Analog	.....	None	.....	.....	.....
3	Axis1.CommandVelocity	On	1	Digital	.....	None	.....	.....	.....
4	Axis1.PositionError	On	1	Digital	.....	None	.....	.....	.....
5	Axis1.VelocityError	On	1	Analog	.....	None	.....	.....	.....
6	Axis1.CurrentFeedback	On	1	Analog	.....	None	.....	.....	.....

- Setting the Min and Max values can be very convenient if the Automatic Mode is not suiting the way the trend visually appears. In this case, you set the Min and Max value of the pen, for example Position Error and Velocity Error, where you want these to be relatively small. Make sure to balance the value out across the X-Axis by setting both Min and Max values the same.



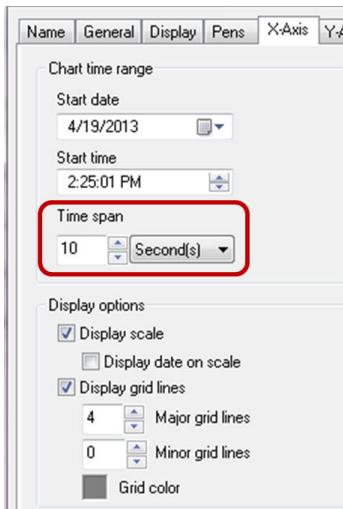
The screenshot shows the 'Pens' tab of a software interface. At the top, there are tabs: Name, General, Display, Pens, X-Axis, Y-Axis, Template, Sampling, Start Trigger, and Stop Trigger. The 'Pens' tab is selected. Below the tabs is a section titled 'Pen Attributes' containing a table. The table has columns: Tag\Expr., Type, Style, Marker, Min, Max, and Eng. L. There are 6 rows of data:

	Tag\Expr.	Type	Style	Marker	Min	Max	Eng. L
1	Axis1.ActualPosition	Analog	-----	None	0.000000	100.000000	
2	Axis1.ActualVelocity	Analog	-----	None	-40.000000	40.000000	
3	Axis1.CommandVelocity	Analog	-----	None	-40.000000	40.000000	
4	Axis1.PositionError	Analog	-----	None	-0.050000	0.050000	
5	Axis1.VelocityError	Analog	-----	None	-3.000000	3.000000	
6	Axis1.CurrentFeedback	Analog	-----	None	-200.000000	200.000000	

## X-Axis Tab

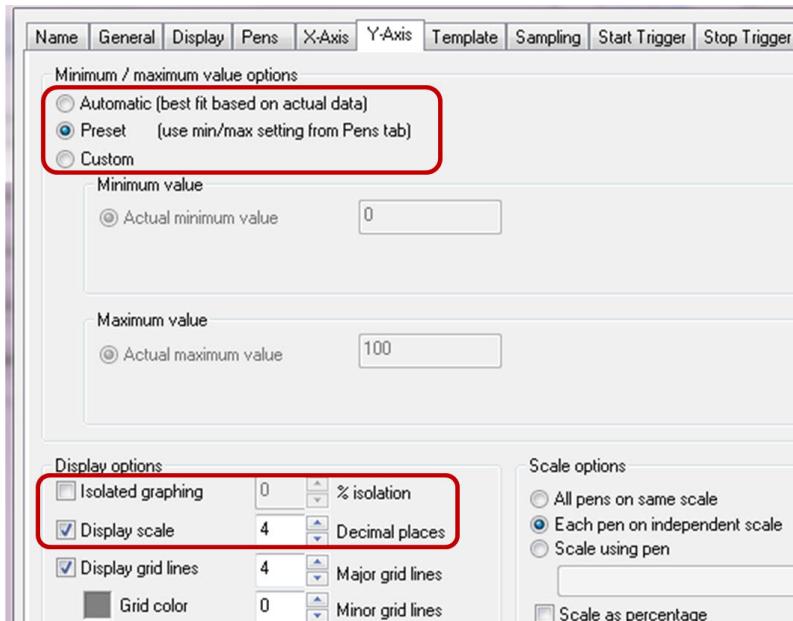
In the X-Axis tab, the only pertinent information for tuning is the scale of time for the axis. Change this to where you can comfortably see one or more cycles of the movements. You can always zoom in by clicking and highlighting the area.

**Figure 93 - X-Axis Tab**

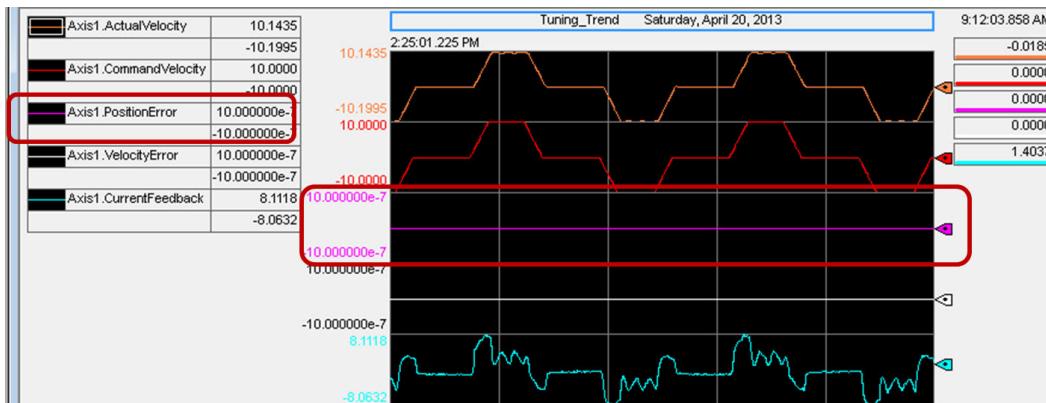


## Y-Axis Tab

The Y-Axis tab has some important settings that can be useful in different situations.

**Figure 94 - Y-Axis Tab**

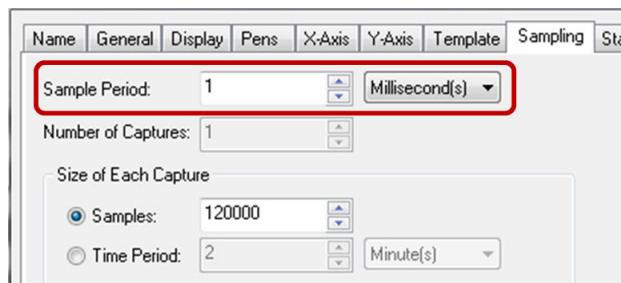
- Value options - clicking Automatic is the simplest, however, when tuning specific values, it can be more useful to click Preset with the Min and Max values you have previously set.
- Isolated graphing - checking this box displays all the pens in a time chart format. This is useful when plotting digital signals against motion. Also, when used with the Automatic value option, it can provide a window spanning the minimum and maximum values of the pen across the trend window, which is useful when manual tuning to reduce error. An example is shown below.



- Display scale - checking this box is important when you are fine tuning. You can choose how many decimal places to use for values like Velocity Error and Position Error pens, shown above. In this example, 4 decimals are selected, and the real value is shown on the trend to 4 decimal places.

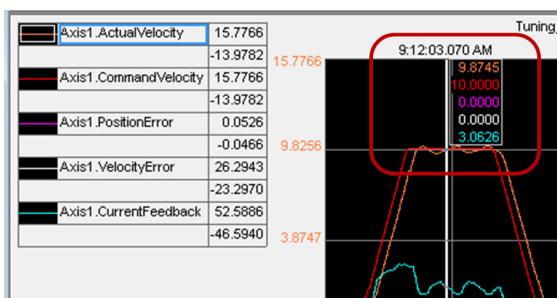
## Sampling Tab

The Sampling tab is used to set the sample rate of the pens. With tuning, you want the sampling rate set as high as possible without affecting the processor speed. Because the trend runs as a service in the Logix Designer application, you need to consider other online trends and processor utilization. A sample period set the same as the Coarse Update Rate of the Motion Group is recommended.

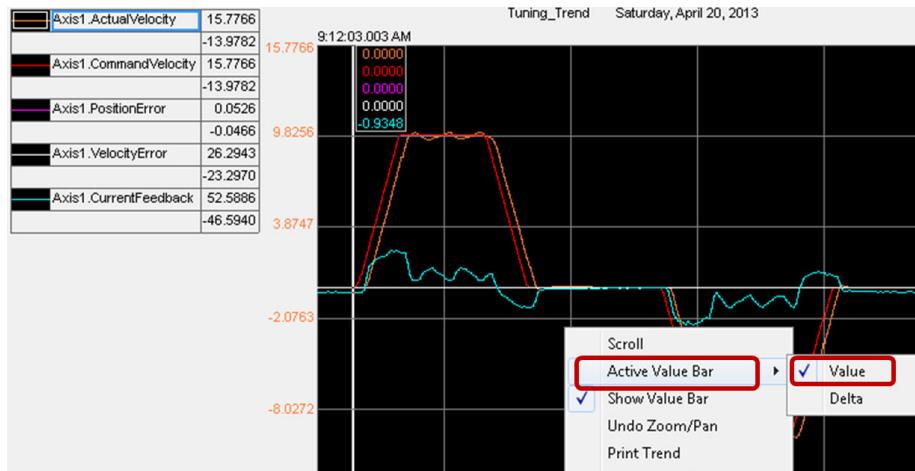
**Figure 95 - Sampling Tab**

## Value Bar Information

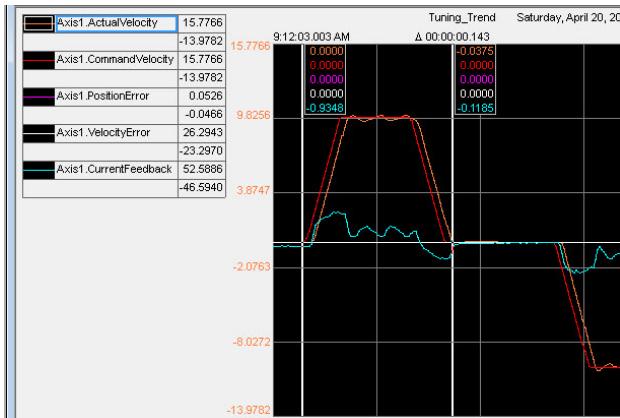
The value bar is important because it shows instant values of each pen at the same point in time. It also provides a time stamp that can be correlated to other trends if they are running simultaneously.

**Figure 96 - Value Bar**

Another useful way to use the value bar is in Delta mode where you can measure the width and magnitude differences of values in terms of time. To do this, right-click inside the trend window and choose Delta from the Active Value Bar.

**Figure 97 - Value Bar (Delta mode)**

After choosing Delta, the value bar locks on the starting point and when you click the second location, the delta value appears as time and the pen values are displayed (143 ms) as shown in [Figure 98](#).

**Figure 98 - Value Bar (Delta value)**

To clear the value bar and return to normal, right-click inside the trend window and choose Delta from the Active Value Bar. Then check value bar again.

## Notes:

## Set Sercos Gains with IDN Write Messages

This appendix describes how to set Sercos gains with write messages.

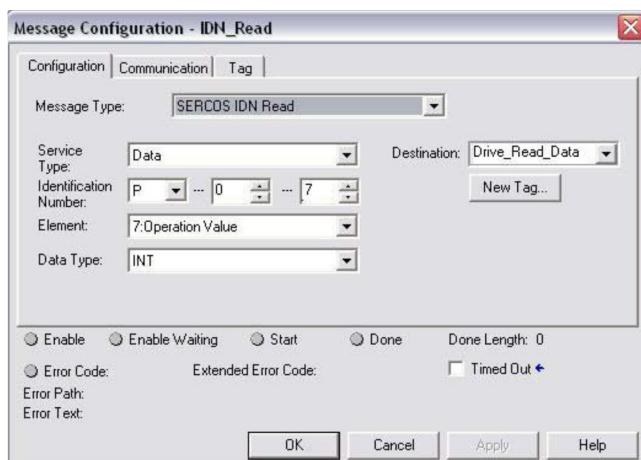
Topic	Page
Set Load Observer Configuration Attribute	101
Set Load Observer Gains	102

Write the load-observer configuration attribute and load observer gains each time the drive gets initialized after applying power.

### Set Load Observer Configuration Attribute

Use the Studio 5000 Logix Designer® application to perform the Sercos IDN read and write instructions.

- After initializing the drive, read the INT value of the load observer configuration with Sercos IDN P:0:431.

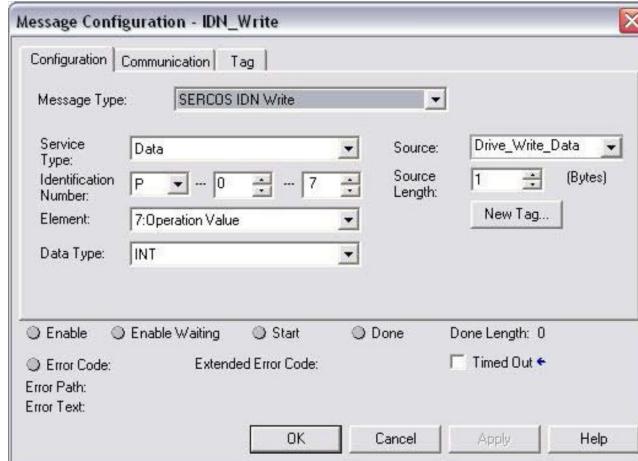


- If the read value is correct, skip to Set Load Observer Gains on [page 102](#).

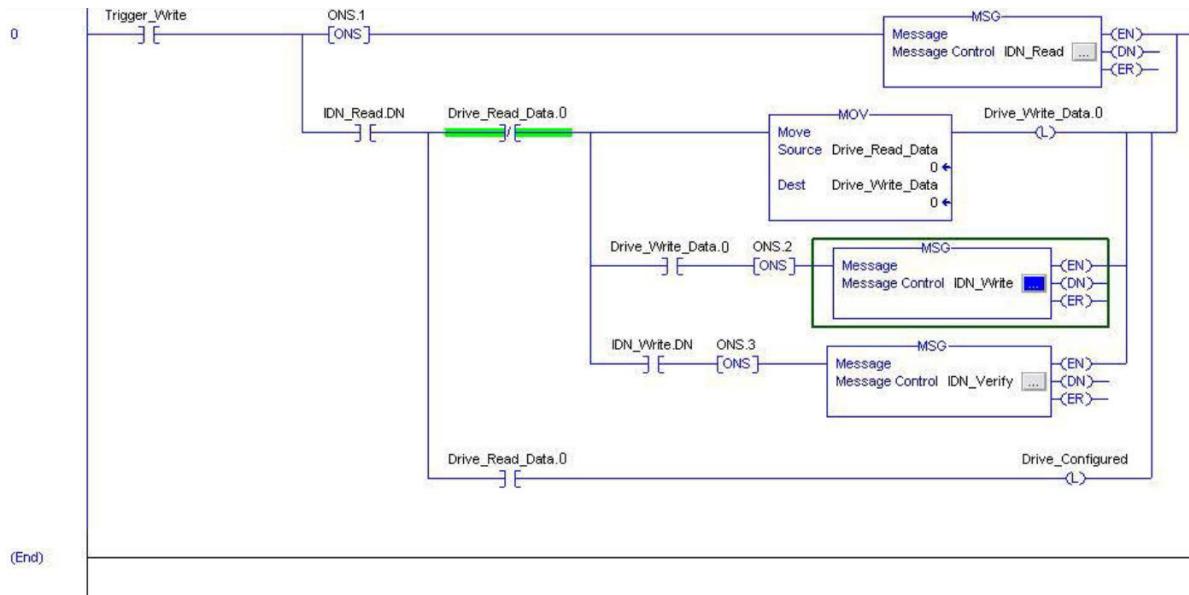
If you want to change the read value, do the following.

- Latch the value.
- Write the new value to the drive with Sercos IDN P:0:431.

Use INT again for the data type.



c. Read the INT value of the load observer configuration with Sercos IDN P:0:431 to verify the change.



## Set Load Observer Gains

Follow a similar procedure to the one previously described to set each of the load observer gains, as needed, with write and read messages.

# Vertical Axis Tuning

This appendix describes some considerations for vertical load applications..

Topic	Page
Overview	103
Considerations	104

## Overview

Vertical axes present a constant force of gravity on the load. This can present challenges with the load dropping while enabling or disabling the axis. Special consideration is required when tuning an axis to overcome these run time problems. Regardless of whether the load observer feature is enabled, we recommend applying one of the following methods.

### Torque Offset Method

Apply a constant torque offset to overcome the effect of gravity on a fixed inertia.

- This is the most common method.
- Create application code that calculates and applies a changing torque offset to overcome the effect of gravity on changing inertia.
- The proper amount of torque offset acts as an electronic counterbalance, allowing the control loops to strictly compensate for positioning.

See Vertical Load and Holding Brake Management Application Technique, publication [MOTION-AT003](#), for information on applying the torque offset method.

### Integrator Gain Method

Apply a small amount of position integral gain. Apply a small amount of load-observer integral gain if load observer is enabled.

- This method can be less effective than the torque offset method because the integrator often allows a small period of time to accumulate from the initial conditions until it can overcome the effect of gravity on the load. In this small period of time, the load can drop.
- This method can be more effective than the torque offset method regarding ease-of-use in overcoming the effect of gravity on changing inertia.

The torque proving feature in some Kinetix® drives can minimize the effect of load drops and, consequently, reduce the need for either of these methods. Kinetix drives can use the autotune function to tune a vertical axis with a holding brake.

## Considerations

Controlling a brake is not difficult if you use the tools provided. Here are some factors to consider:

- The holding brake is not a stopping brake.** If the application attempts to stop the load with the brake, the brake will not engage. However, if the zero speed detection occurs with significant torque at the brake, the brake can be damaged over time. You can use a mechanical stopping and/or regenerative means on your axis outside of the holding brake if you need stopping power. This can be modeled by using Motion Analyzer software.
- Brake drops on an E-Stop.** This is not necessarily tuning related. The brake uses a spline mechanism to engage and disengage. Use of the physical holding brake requires time to engage this spline. In the Studio 5000 Logix Designer® application, it can mean that brake engage and brake release times are required or at least need to be investigated. The following is an example of a few motor brake engage times and where these are located in the Logix Designer application for an EtherNet/IP drive.

MP-Series Low Inertia Motor Brake Specifications

Motor Cat. No.	Backlash, max (brake engaged) arc minutes	Holding Torque N·m (lb·in)	Coil Current at 24V DC A	Brake Response Time			Brake Rotor Inertia kg·m <sup>2</sup> (lb·in·s <sup>2</sup> )	Brake Motor Weight, approx kg (lb)		
				Release ms	Engage (using external arc suppression device)					
					MOV ms	Diode ms				
MPL-A/B1510V							0.0000099 (0.000088)	1.2 (2.6)		
MPL-A/B1520U		0.9 (8.0)	0.43...0.53	23	9	18	0.000015 (0.00013)	1.4 (3.1)		
MPL-A/B1530U							0.000026 (0.00023)	1.8 (3.9)		
MPL-A/B210V	0						0.000033 (0.00029)	1.8 (4.0)		

LoadRatio	U.U	Load inertia/motor
LoadType	Direct Coupled Rotary	
MasterDelayCompensation	True	
MasterPositionFilter	False	
MasterPositionFilterBandwidth	0.1 Hz	
MaximumAcceleration	14024.986 Position Units/s <sup>2</sup>	
MaximumAccelerationJerk	2776944.5 Position Units/s <sup>3</sup>	
MaximumDeceleration	14024.986 Position Units/s <sup>2</sup>	
MaximumDecelerationJerk	2776944.5 Position Units/s <sup>3</sup>	
MaximumSpeed	70.833336 Position Units/s	
MechanicalBrakeControl	Automatic	
* MechanicalBrakeEngageDelay	.03 s	
* MechanicalBrakeReleaseDelay	.2 s	
MotorPolarity	Normal	
MotionResolution	1000000 Motion Counts/A	
MotionScalingConfiguration	Control Scaling	
MotionUnit	Motor Rev	
MotorCatalogNumber	MPF-B330P-M	
MotorDataSource	Catalog Number	

- During tuning and commissioning, it may be necessary to control the brake manually.

See Vertical Load and Holding Brake Management Application Technique, publication [MOTION-AT003](#), for more information on holding brake management.

In summary, you can write to IDN parameters to manually engage and disengage the brake for both Sercos and EtherNet/IP drives. Any of the methods described in this publication can be used to tune a vertical axis. See [Chapter 2](#), Out-of-Box Tuning, for the recommended tuning settings for the drives. See Vertical Load and Holding Brake Management Application Technique, publication [MOTION-AT003](#), for more information on holding brake management and additional considerations for tuning vertical load axes.

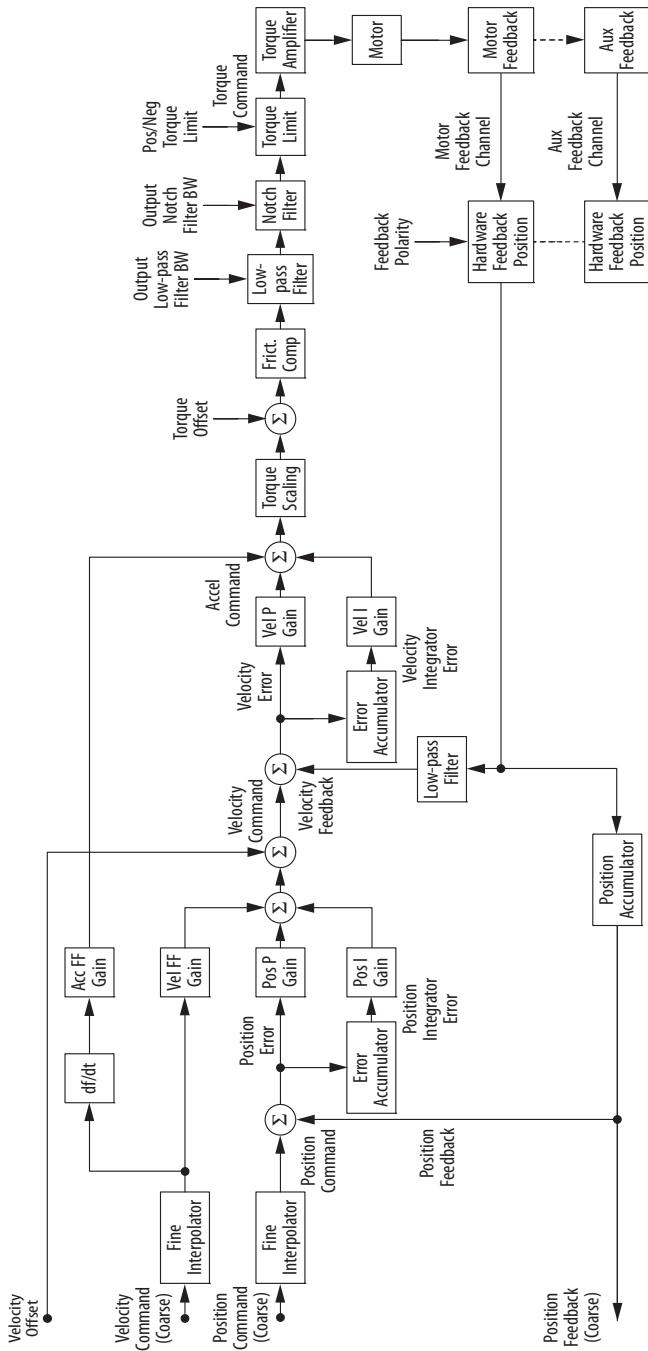
## Block Diagrams

This appendix provides block diagrams for Sercos and EtherNet/IP drives.

Topic	Page
Sercos Drives	106
EtherNet/IP Drives	107

## Sercos Drives

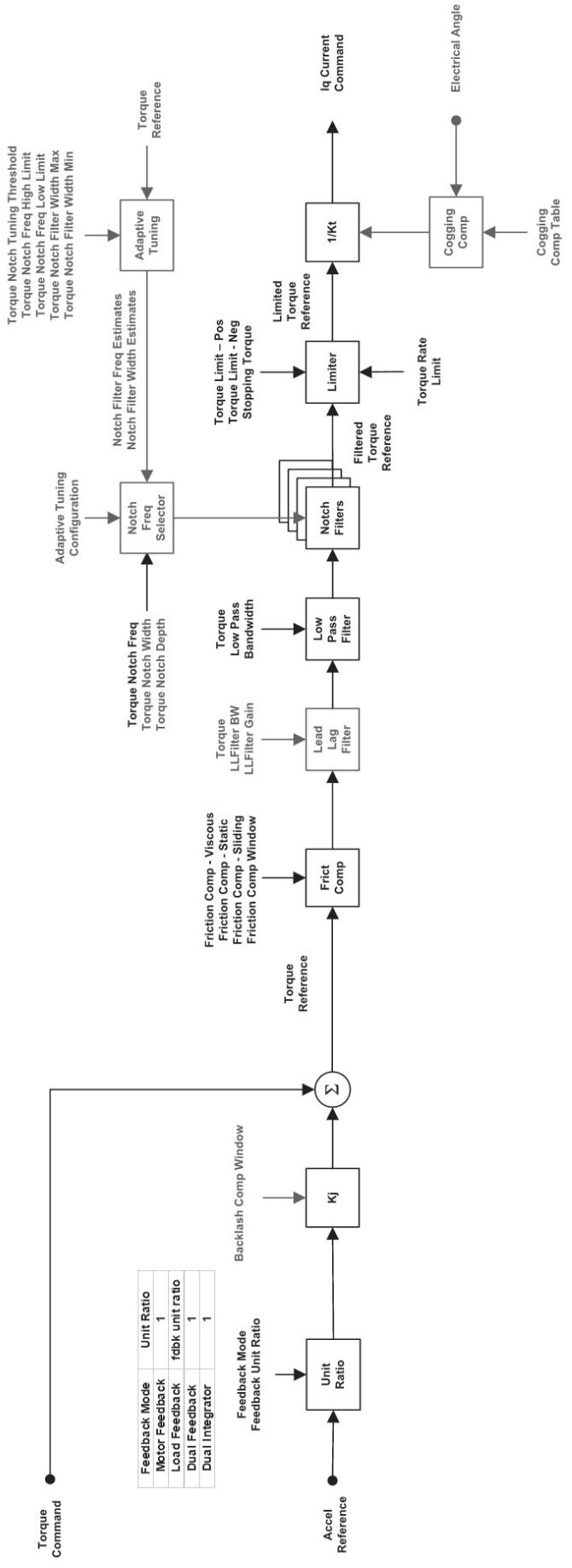
Kinetix® 7000, Kinetix 6000, Kinetix 6200, and Kinetix 2000 drive control loop architecture is shown in the following diagram.



## EtherNet/IP Drives

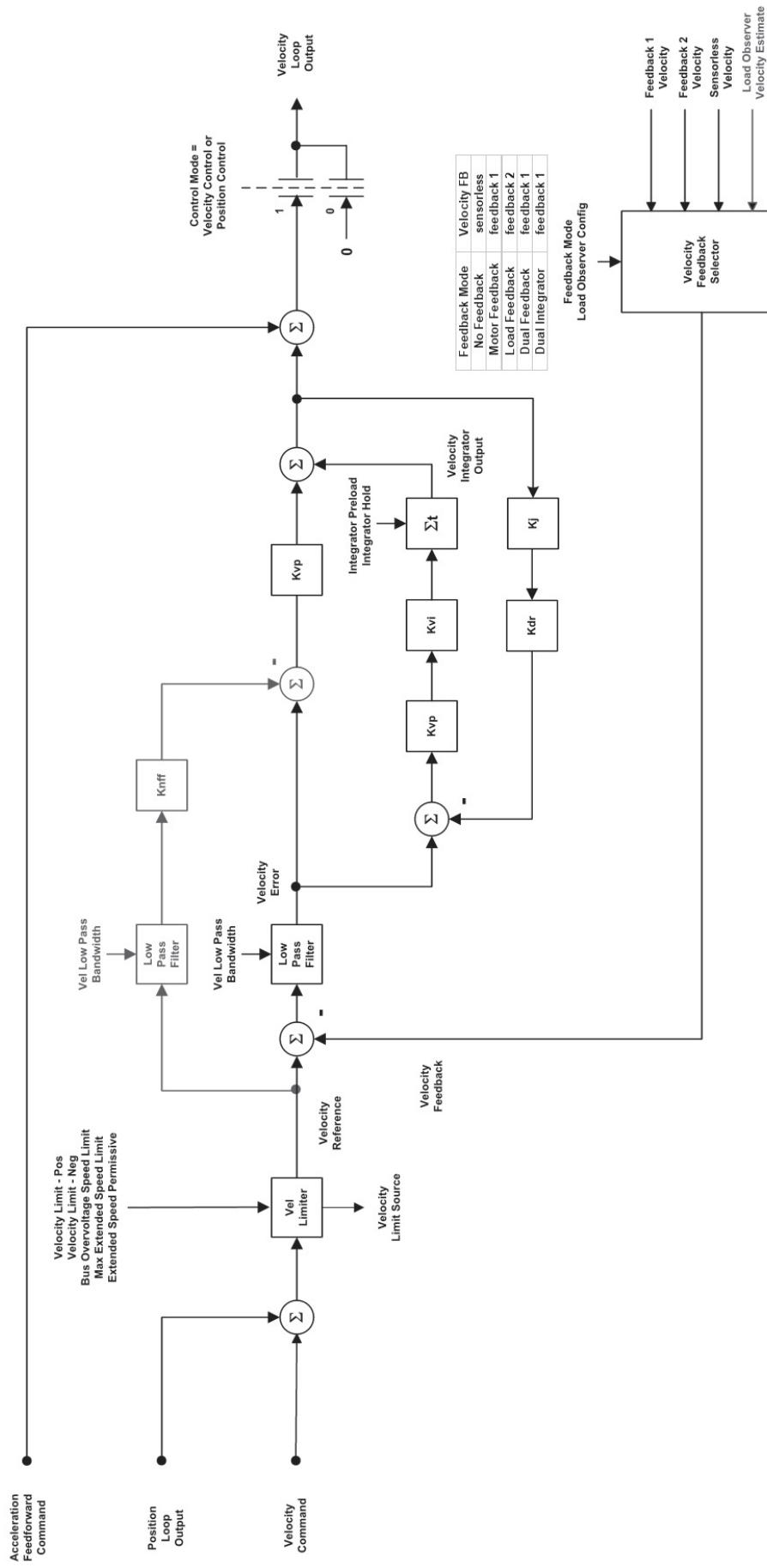
### Position Loop

Kinetix 6500, Kinetix 5700, Kinetix 5500, Kinetix 5300, and Kinetix 350 drive position loop architecture is shown in the following diagram.



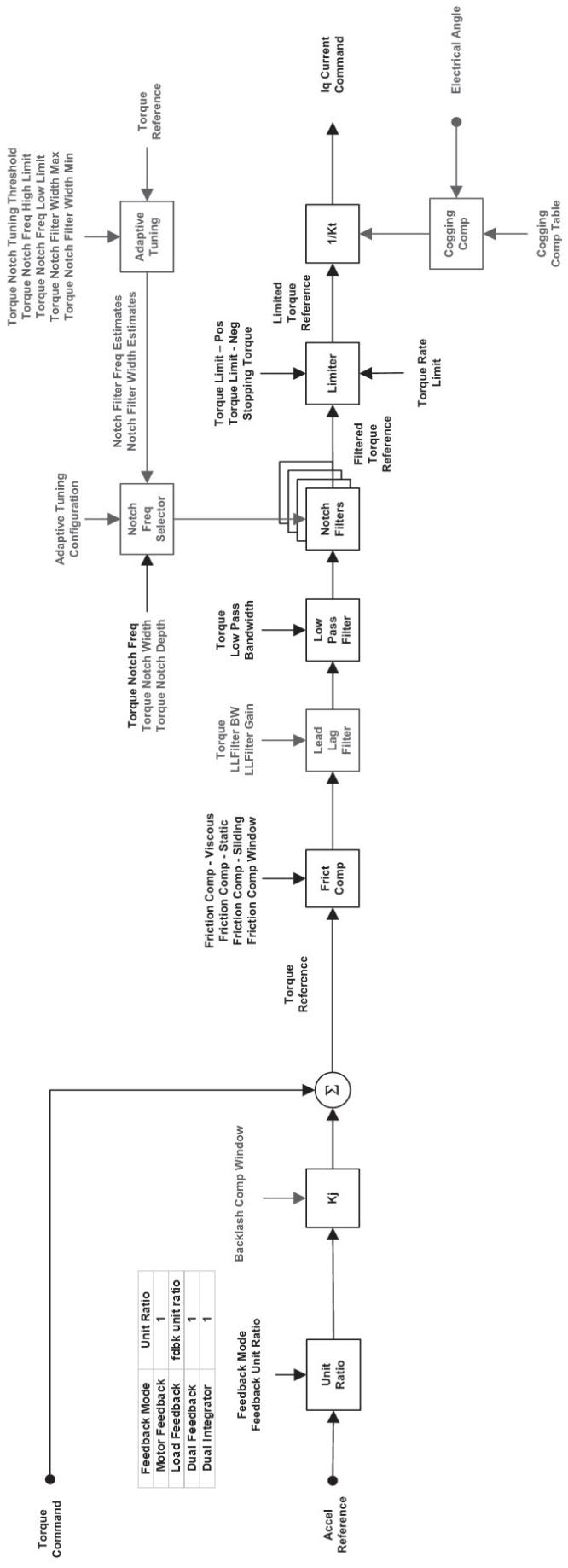
## Velocity Loop

Kinetix 6500, Kinetix 5700, Kinetix 5500, Kinetix 5300, and Kinetix 350 drive velocity loop architecture is shown in the following diagram.



## Torque Loop

Kinetix 6500, Kinetix 5700, Kinetix 5500, Kinetix 5300, and Kinetix 350 drive torque/current loop architecture is shown in the following diagram.



**Notes:**

# Additional Resources

These documents contain additional information concerning related products from Rockwell Automation.

Resource	Description
Logix5000 Controllers Quick Start, publication <a href="#">1756-OS001</a>	Start programming Logix5000™ controllers
Logix5000 Controllers General Instructions Reference Manual, publication <a href="#">1756-RM003</a>	Get detailed information about Logix-based instructions
Logix5000 Controllers Motion Instructions Reference Manual, publication <a href="#">MOTION-RM002</a>	Get detailed information about Logix-based motion instructions
Integrated Motion on the EtherNet/IP Network User Manual, publication <a href="#">MOTION-UM003</a>	Configure and startup integrated motion applications
Kinetix 6000 Multi-axis Servo Drives User Manual, publication <a href="#">2094-UM001</a>	Install, configure, and troubleshoot Kinetix® 6000 applications
Kinetix 6200/6500 Modular Multi-axis Servo Drives User Manual, publication <a href="#">2094-UM002</a>	Install, configure, and troubleshoot Kinetix 6200/6500 applications
Kinetix 6000 and Kinetix 6200/6500 Drive Systems Design Guide, publication <a href="#">GMC-RM003</a>	Determine what you need for Kinetix 6000 and Kinetix 6200/6500 applications
Kinetix 5300 Servo Drives User Manual, publication <a href="#">2198-UM005</a>	Install, configure, and troubleshoot Kinetix 53000 applications
Virtual Torque Sensor Application Technique, publication <a href="#">2198-AT003</a>	Get detailed information about the Virtual Torque Sensor, including parameter descriptions, methods to configure filters, and examples of applications.
Kinetix 5500 Servo Drives User Manual, publication <a href="#">2198-UM001</a>	Install, configure, and troubleshoot Kinetix 5500 applications
Kinetix 5500 Drive Systems Design Guide, publication <a href="#">GMC-RM009</a>	Determine what you need for Kinetix 5500 applications
Kinetix 5700 Servo Drives User Manual, publication <a href="#">2198-UM002</a>	Install, configure, and troubleshoot Kinetix 5700 applications
Kinetix 5700 Drive Systems Design Guide, publication <a href="#">GMC-RM010</a>	Determine what you need for Kinetix 5700 applications
PowerFlex 700S and PowerFlex 755 Drives Tuning Manual, publication <a href="#">DRIVES-AT004</a>	Tune PowerFlex® 700S/755 drives
Kinetix Tuningless Quick Start Guide, publication <a href="#">MOTION-OS001</a>	Quickly tune Kinetix drives

You can view or download publications at [rok.auto/literature](#).

# Rockwell Automation Support

Use these resources to access support information.

<b>Technical Support Center</b>	Find help with how-to videos, FAQs, chat, user forums, and product notification updates.	<a href="http://rok.auto/support">rok.auto/support</a>
<b>Knowledgebase</b>	Access Knowledgebase articles.	<a href="http://rok.auto/knowledgebase">rok.auto/knowledgebase</a>
<b>Local Technical Support Phone Numbers</b>	Locate the telephone number for your country.	<a href="http://rok.auto/phonesupport">rok.auto/phonesupport</a>
<b>Literature Library</b>	Find installation instructions, manuals, brochures, and technical data publications.	<a href="http://rok.auto/literature">rok.auto/literature</a>
<b>Product Compatibility and Download Center (PCDC)</b>	Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes.	<a href="http://rok.auto/pcdc">rok.auto/pcdc</a>

## Documentation Feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at [rok.auto/docfeedback](http://rok.auto/docfeedback).

Allen-Bradley, Connected Components Workbench, expanding human possibility, Kinetix, Logix 5000, PowerFlex, Rockwell Automation, and Studio 5000 Logix Designer are trademarks of Rockwell Automation, Inc.

EtherNet/IP is a trademark of ODVA, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.

Rockwell Automation maintains current product environmental compliance information on its website at [rok.auto/pec](http://rok.auto/pec).

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752, İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Connect with us.    

[rockwellautomation.com](http://rockwellautomation.com)

expanding human possibility™

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

ASIA PACIFIC: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication MOTION-AT005D-EN-P - November 2020

Supersedes Publication MOTION-AT005C-EN-P - May 2016

Copyright © 2020 Rockwell Automation, Inc. All rights reserved. Printed in the U.S.A.