



33 South La Patera Lane
 Santa Barbara, CA 93117-3214
 ph (805) 681-3300
 fax (805) 681-3311
 tech@motioneng.com
 www.motioneng.com

APPLICATION NOTE 9D00-0184, REV. A

Scaling PID Tuning Parameters for Different Controller Sample Rates

INTRODUCTION

When tuning servo systems, it may be necessary to adjust the controller's sample rate for optimum performance. Changing the sample rate will directly affect the closed loop response of the PID control algorithm. To maintain stability and performance, the PID parameters must be scaled to match the change in sample rate. This document explains how to scale the tuning parameters properly. This document only addresses scaling for the PID closed loop algorithm. It does not compensate for resonances caused by increased system bandwidth or system nonlinearities.



Warning!

The system may experience stability problems when sample rates are changed from a higher sample rate to a much lower sample rate. This is due to additional phase lag of the zoh (zero order hold, the value that is held constant between sample periods) at a lower sample rate. It is necessary to take the appropriate safety precautions in order to prevent any sudden and unexpected movement of machinery from being hazardous to neighboring objects and/or personnel. Before entering or changing sample rates, verify that all personnel, hands, fingers, body parts, and/or property are clear of the movement area; otherwise, injury or death may result!

Document Revision History: Application Note 9D00-0184				
Rev.	Date	Description	DCR No.	DCO No.
A	25OCT2001	Initial release.	571	1458



NOTE: This document contains proprietary and confidential information of Motion Engineering, Inc., and is protected by federal copyright law. The contents of the document may not be disclosed to third parties, translated, copied, or duplicated in any form, in whole or in part, without the express written permission of Motion Engineering, Inc.

PID Loop and Sample Rates

When sample rates are changed, the performance of the servos will change. To understand why this relationship exists, it is helpful to look at the architecture of a PID control loop.

PID Feedback Loops

A simplified PID feedback loop is shown in the diagram below.

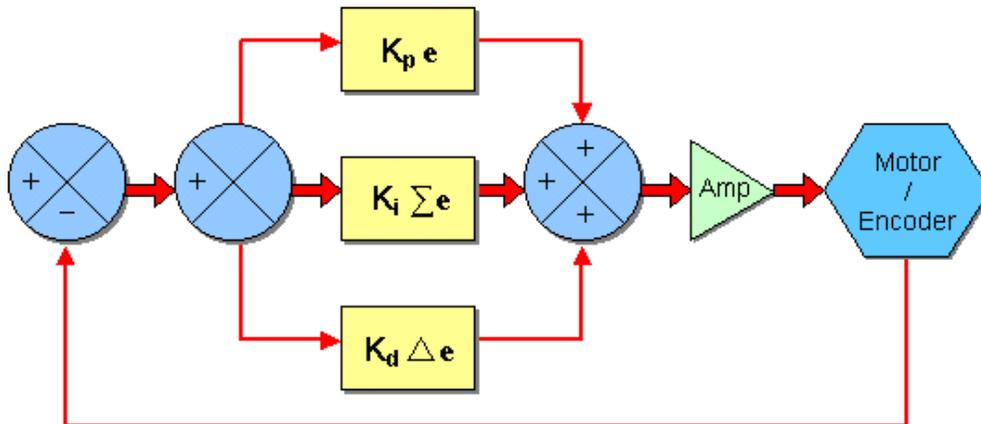


Figure 1. PID Feedback Loop

Major Components of the PID Feedback Loop, Theoretical vs. Practice

- Proportional Term - $K_p e$**
 This is automatically calculated in the XMP and is not affected by sample rate, since there is no time component factored into this calculation.
- Theoretical Integral - $K_i \int e dt$**
 In a digital sampling system with a finite sample rate, this is equivalent to $K_i \sum(e * t_{\text{sample}})$. In the XMP, the t_{sample} is not calculated to save computing time. As long as the sample rate is constant, the t_{sample} can be factored out. When sample rates are changed, K_i must be adjusted to maintain closed loop performance.
- Theoretical Derivative - $K_d * de/dt$**
 In a digital sampling system with a finite sample rate, this is equivalent to $K_d * \Delta e / \Delta t$. Δt is always equal to t_{sample} when the sample rate is constant. In the XMP, the t_{sample} is not calculated to save computing time. As long as the sample rate is constant, the t_{sample} can be factored out. When sample rates are changed, K_d must be adjusted to maintain closed loop performance.

Compensating the PID Feedback Loop for a Change in SampleRate:

Kp- Proportional Term

Kp does not need to be compensated for when using different sample rates because the proportional term has no time component.

Ki- Integral Term

The equation that the XMP uses for the integral term provides insight into how to compensate for the change in sample rate.

If you look at the difference between the XMP integral term and the ideal digital derivative term, you will find that the XMP derivative term should be multiplied by the sample (t_{sample}) to change it to the ideal digital derivative term. This means that you only need to multiply by the sample period (t_{sample}) or divide by the sample rate ($1/t_{\text{sample}}$) to compensate for a sample rate change.

$$\text{new Ki} = \text{old Ki} * \frac{\text{old SR}}{\text{new SR}}$$

$$\text{XMP Integral} = \text{Ki} * \sum e \qquad \text{Ideal Integral} = \frac{\text{Ki} * \sum e}{\text{SR}}$$

Example

To convert Ki from one sample rate to another, multiply by the new sample rate and divide by the old sample rate.

$$\begin{aligned} \text{old Ki} &= 0.01 \\ \text{old SR} &= 2,000 \\ \text{new SR} &= 10,000 \\ \text{new Ki} &= \text{old Ki} / (\text{new SR} * \text{old SR}) = 0.01 * 2,000 / 10,000 = 0.002 \end{aligned}$$

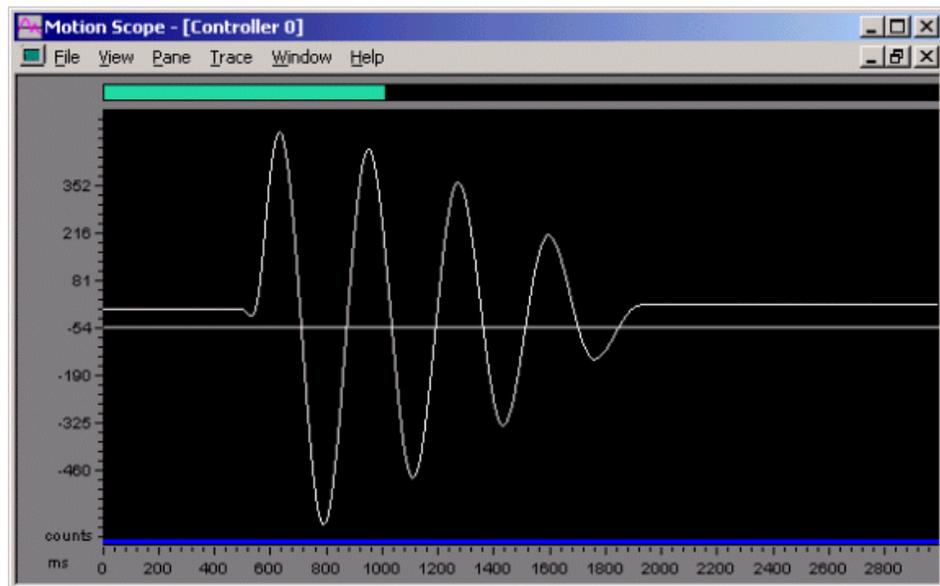


Figure 2. External Disturbance
Kp = 10, Ki = 0.01, SR = 20,000

The graph above shows a system that is almost in a state of instability due to a large integral term and a lack of a derivative term. Actual position is plotted vs. time showing an external disturbance.

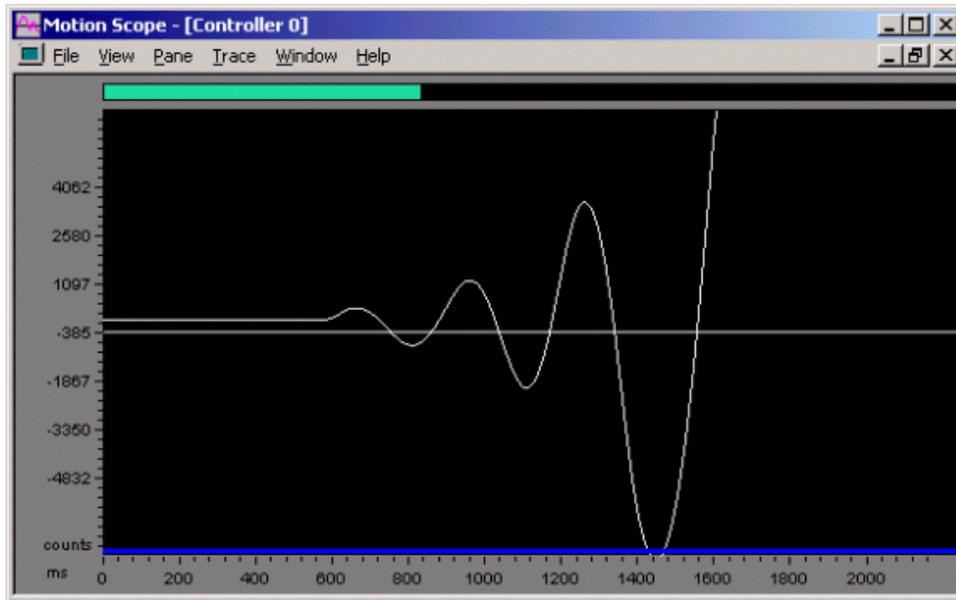


Figure 3. External Disturbance
 $K_p = 10$, $K_i = 0.01$, $SR = 10,000$

The integral term (K_i) is effectively larger at a 10,000Hz sampling rate than a 2,000Hz sampling rate. This increase in integral gain results in an unstable case, as shown in the diverging oscillation in Figure 3.

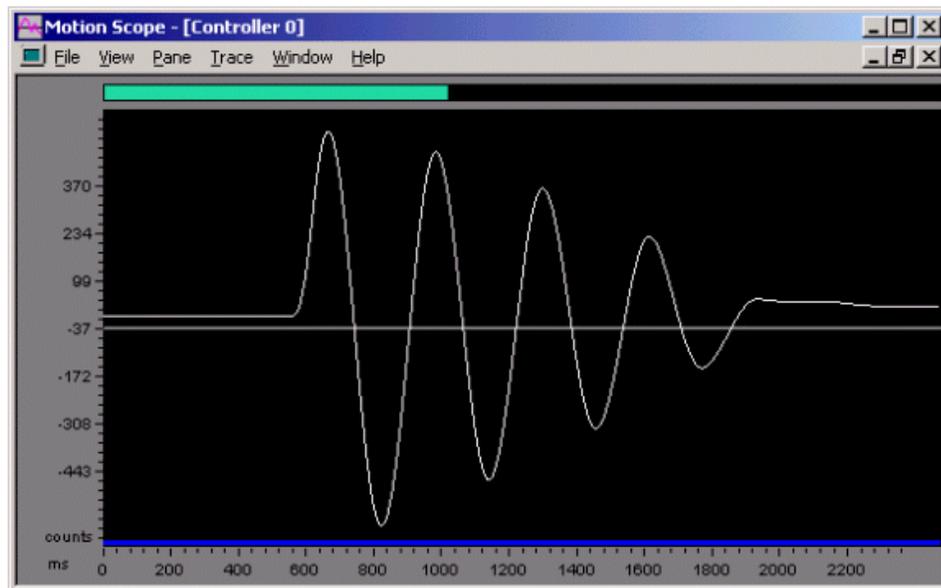


Figure 4. External Disturbance
 $K_p = 10$, $K_i = 0.002$, $SR = 10,000$

When the integral term is properly compensated for by using a smaller compensated integral term, it will result in performance that is very similar to the original performance. Notice the similarity between the graphs of Figure 2 and Figure 4.

Kd- Derivative Term

The equation that the XMP uses for the derivative term provides insight into how to compensate for the sample rate change.

If you look at the difference between the XMP derivative term and the ideal digital derivative term, you find that the XMP derivative term should be divided by the sample (t_{sample}) to change it to the ideal digital derivative term. This means that you only need to divide by the sample period (t_{sample}) or multiply by the sample rate ($1/t_{\text{sample}}$) to compensate for a sample rate change.

$$\text{new Kd} = \text{old Kd} * \left(\frac{\text{new SR}}{\text{old SR}} \right)$$

$$\text{XMP Derivative} = \text{Kd} * \Delta e \qquad \text{Ideal Derivative} = \frac{\text{Kd} * \Delta e}{\text{SR}}$$

Example

To convert Kd from one sample rate to another, multiply the old Kd by the new sample rate and divide by the old sample rate.

old Kd = 1,000
 old SR = 2,000
 new SR = 10,000

$$(\text{Kd} * \text{new SR}) / \text{old SR} = (1,000 * 10,000) / 2,000 = 5,000$$

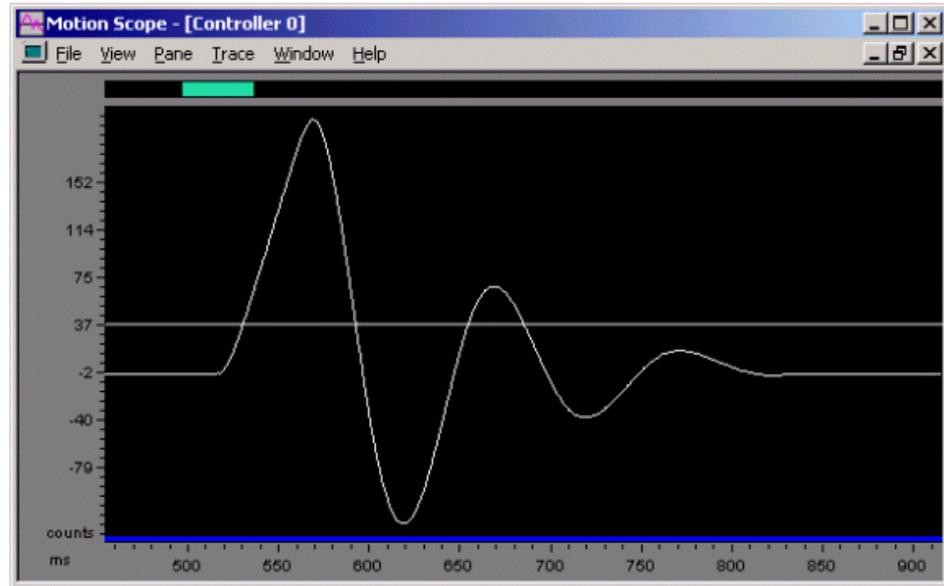


Figure 5. External Disturbance
Kp = 100, Ki = 0, Kd = 1,000, SR = 2,000

The graph above shows actual position vs. time where the system is subjected to an external disturbance. The system is tuned to be somewhat underdamped.

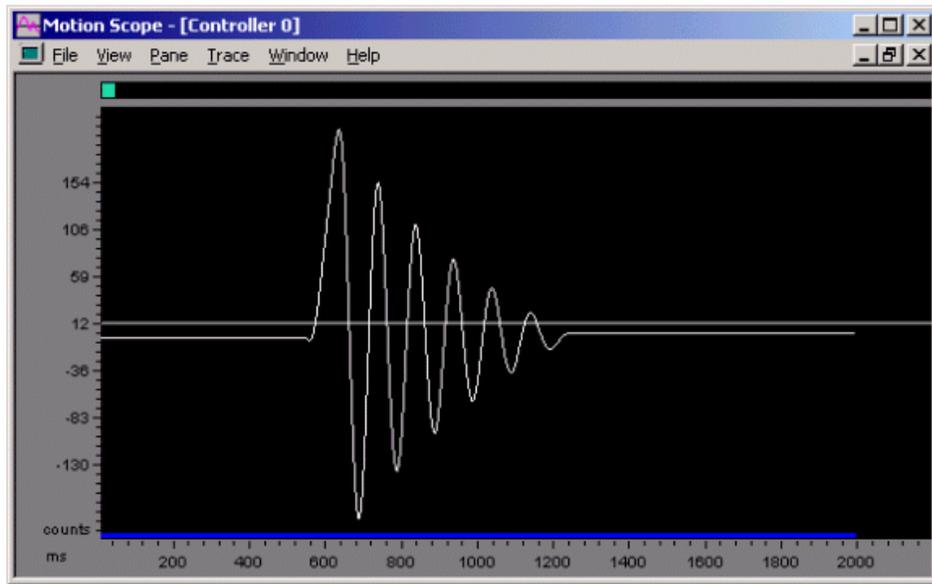


Figure 6. External Disturbance
 $K_p = 100$, $K_i = 0$, $K_d = 1,000$, $SR = 10,000$

Notice that the system has less dampening at a higher sample rate with the same tuning parameters. Next, we will compensate the derivative term for the change in sample rate.

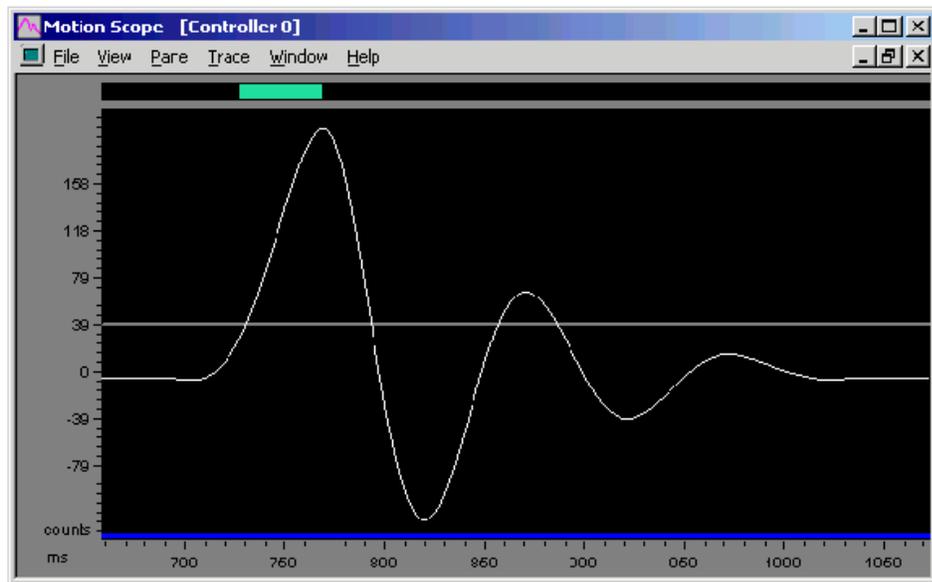


Figure 7. External Disturbance
 $K_p = 100$, $K_i = 0$, $K_d = 5,000$, $SR = 10,000$

Notice that the disturbance plots at 2,000Hz and 10,000Hz are nearly identical when the PID parameters are compensated for the sample rate.

Compensating the PID Feedforward Loop for a Change in Sample Rate

A simplified diagram of the PID feedforward loop is shown below.

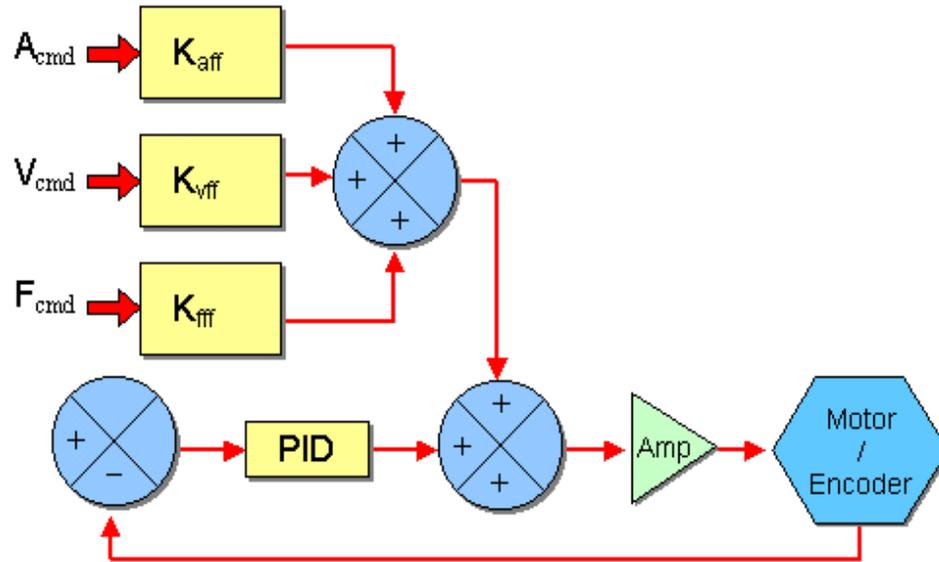


Figure 8. PID Feedforward Loop

Major Components of PID Feedforward

- Acceleration Feedforward - K_{aff}
- Velocity Feedforward - K_{vff}
- Friction Feedforward - K_{ff}
- Position Feedforward - K_{pff}

K_{aff}- Acceleration Feedforward

K_{aff} is calculated as the command acceleration counts / controller sample² multiplied by K_{aff}. In order to compensate for acceleration feedforward, convert from **counts / controller sample²** to **counts / second²** for the original sample rate and back using the new sample rate.

$$\text{new Kaff} = \left(\frac{\text{new SR}}{\text{old SR}} \right)^2 * \text{old Kaff}$$

Example

Let's say you have a sample rate of 2,000 Hz with a K_{aff} value of 100,000. To convert the original sample rate from **counts / controller sample²** to **counts / second²** and back using the new sample rate, multiply K_{aff} by the square of the new (SR / old SR).

old SR = 2,000
 new SR = 10,000
 old K_{aff} = 100,000

$$\begin{aligned} \text{new Kaff} &= (\text{new SR} / \text{old SR})^2 * \text{old Kaff} \\ &= (10,000 / 2,000)^2 * 100,000 = 2,500,000 \end{aligned}$$

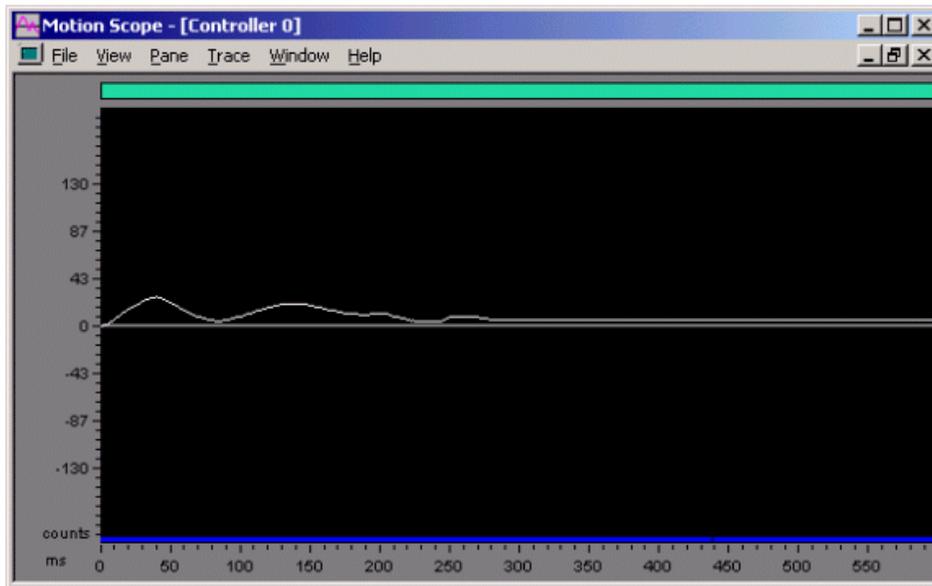


Figure 9. External Disturbance
 $K_p = 100$, $K_i = 0$, $K_d = 5,000$, $K_{aff} = 100,000$, $SR = 10,000$

The above graph shows position error vs. time for a commanded motion with an underdamped servo system. The underdamped nature of the system is difficult to see because the loads on the system are largely inertial. The acceleration feedforward nearly eliminates any inertial based position error.

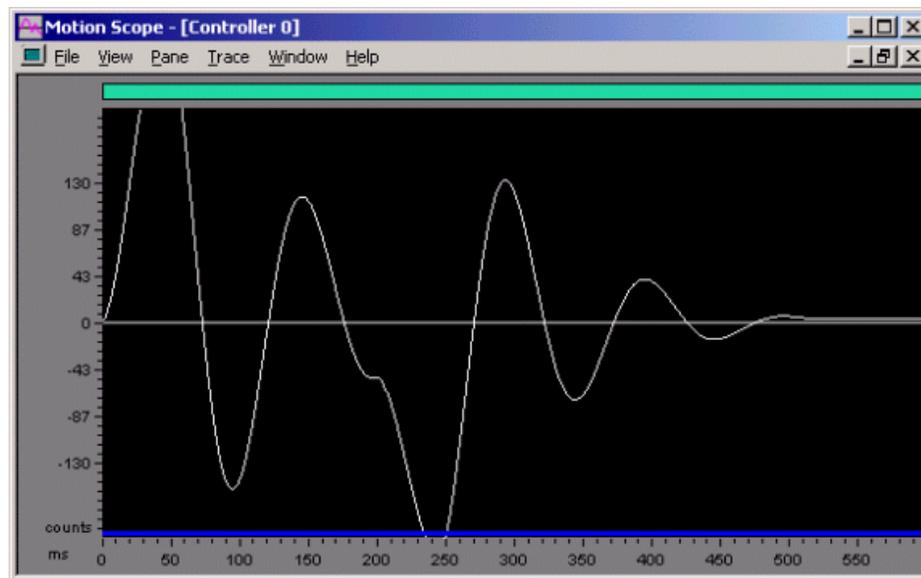


Figure 10. Position Error for a Commanded Motion
 $K_{aff} = 100,000$, $SR = 10,000$
PID parameters scaled from values used at $SR = 2,000$

This stage was intentionally tuned to be underdamped, highlighting an incorrect acceleration feedforward. Notice that large position errors occurred because the acceleration feedforward value was not scaled with the change in sample rate (SR).

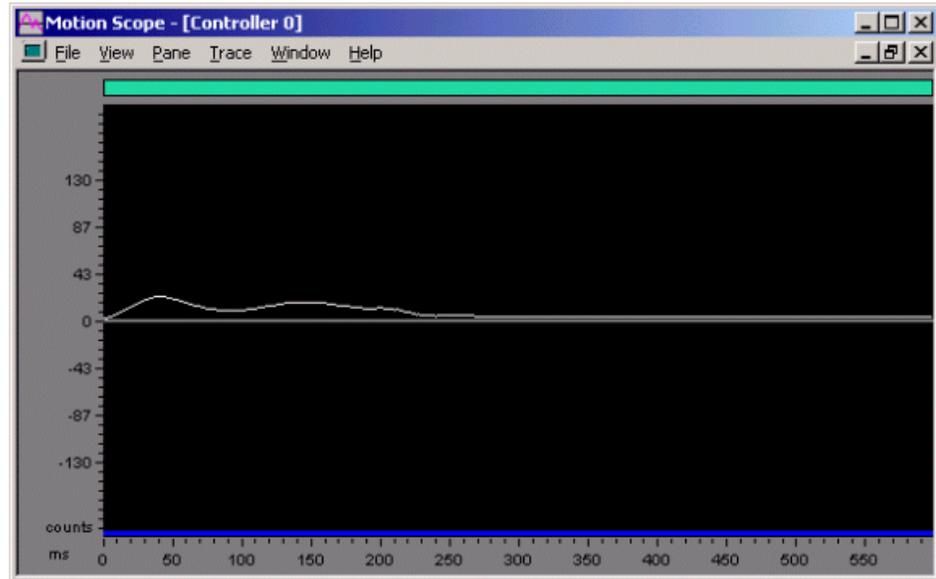


Figure 11. Position Error for a Commanded Motion
K_{aff} = 2,500,000, SR = 10,000
PID parameters and K_{vff} scaled for changed sample rate

Notice that when K_{aff} is changed, the position error is greatly reduced and a profile is produced that is very similar to the graph made at SR = 2,000. See Figure 9 and Figure 10.

K_{vff}- Velocity Feedforward

K_{vff} is calculated as the **command velocity counts / controller sample** multiplied by **K_{vff}**. In order to compensate for velocity feedforward, convert from **counts / controller sample** to **counts / second** for the original sample rate and back using the new sample rate.

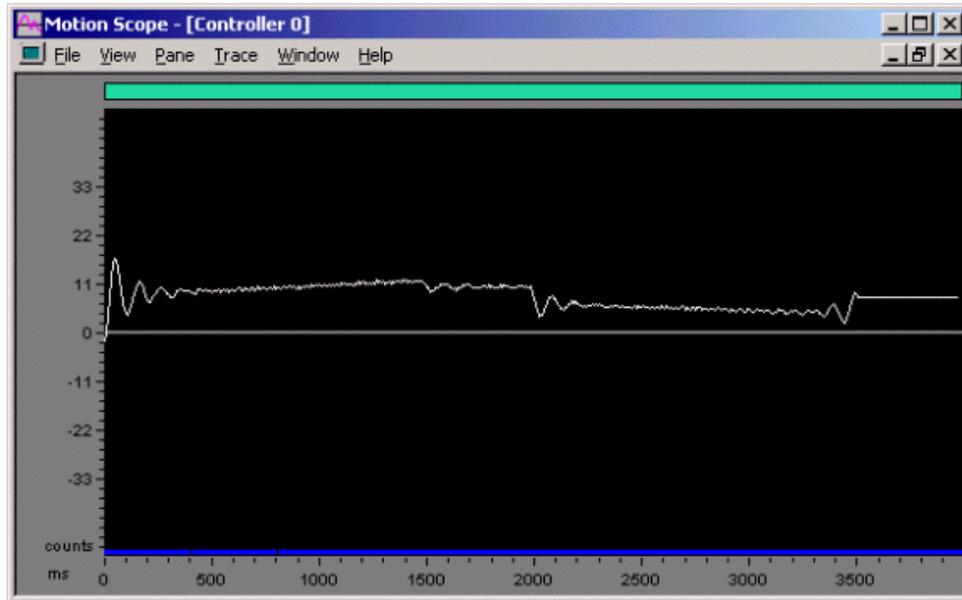
$$\text{new K}_{vff} = \left(\frac{\text{new SR}}{\text{old SR}} \right) * \text{old K}_{vff}$$

Example

Let's say you have a sample rate of 2,000 Hz with a K_{vff} value of 100,000. To convert from **counts / controller sample** to **counts / second** for the original sample rate and back using the new sample rate, multiply K_{vff} by the ratio of the new to old sample rate (SR / old SR).

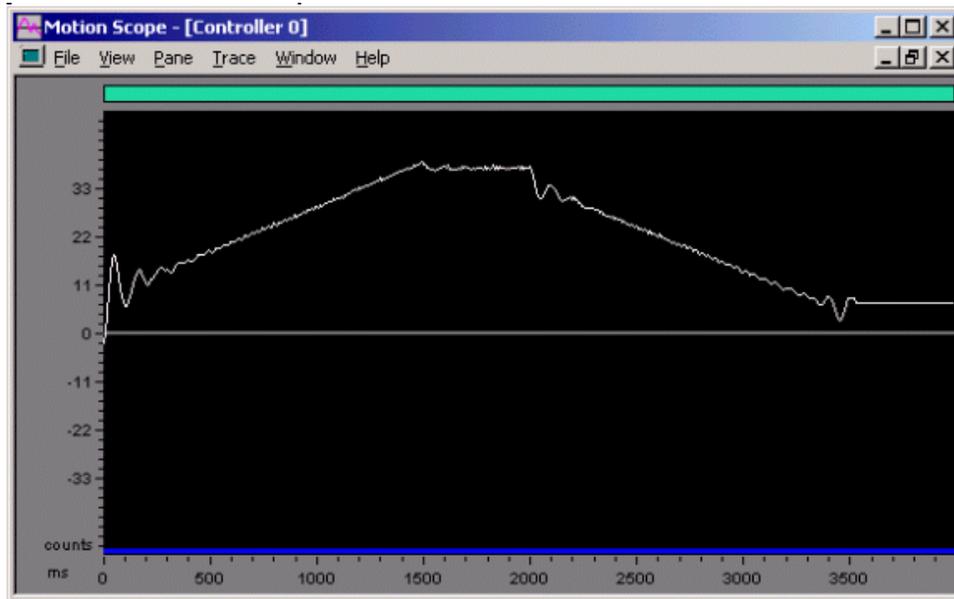
old SR = 2,000
 new SR = 10,000
 old K_{vff} = 100,000

$$\begin{aligned} \text{new K}_{vff} &= (\text{new SR} / \text{old SR}) * \text{old K}_{vff} \\ &= (10,000 / 2,000) * 45 = 225 \end{aligned}$$



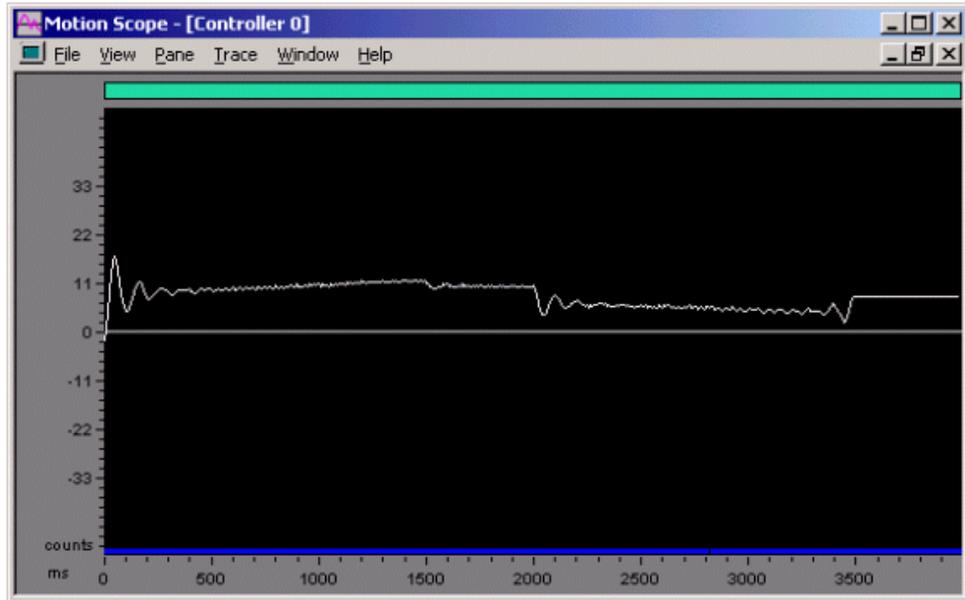
**Figure 12. Position Error vs. Time for a Commanded Motion
Kvff = 45, SR = 2,000**

Here is the same move on the same stage as used for the acceleration feedforward example was performed, except velocity feedforward was used.



**Figure 13. Position Error vs. Time for a Commanded Motion
Kvff = 45, SR = 10,000
PID parameters and Kaff scaled for changed sample rate**

Notice that a significant position error in the form of a velocity profile appears with the non-scaled Kvff term.



**Figure 14. Position Error vs. Time for a Commanded Motion
Kvff = 225, SR =10,000
PID parameters and Kaff scaled for changed sample rate**

The position error no longer shows the large velocity profile shaped component. The position error plot is similar in nature to the plot with the original sample rate (SR).

Kfff- Friction Feedforward

Friction feedforward is not affected by the sample rate, since it has no time component.

Kpff- Position Feedforward

Position feedforward is not affected by the sample rate, since it has no time component.

Unexpected Results

The control system bandwidth cannot exceed the *Nyquist Frequency* ($\frac{1}{2}$ sample rate). This means that there will only be DAC quantization distortions exciting resonances above $\frac{1}{2}$ the sample rate.

This means that if your stage has a large resonance at 3,000 Hz, and you run your controller at a SR = 2,000 Hz, you may never know that the resonance exists. If you change the sample rate to 10,000 Hz, you now have a control system that can create significant 3,000 Hz energy. It would not be surprising to find your stage oscillating at 3,000 Hz.

It is good to know where all the resonances are in your system. If you want to raise your sample rate to a frequency that covers a resonance in your system, consider applying a low pass or notch filter to reduce the magnitude of the resonance.

Do not be afraid of fine tuning the scaled tuning parameters (hint: write them down first). It is easier to fine tune scaled parameters than to start applying filters. Applying filters will most likely require the use of revised tuning parameters, so it is easier to fine tune scaled parameters instead.

Remember that this method of tuning parameter scaling only provides scaled tuning parameters. It does not take resonances or non-linearities into account, and is not intended to compensate for those effects.

PID Sample Rate Compensation Summary

Table 1: Conversion Equations

Term	Compensation
Proportional - Kp	None
Integral - Ki	$\text{new Ki} = \text{old Ki} * (\text{old SR} / \text{new SR})$
Derivative - Kd	$\text{new Kd} = \text{old Kd} * (\text{new SR} / \text{old SR})$
Acceleration Feedforward - Kaff	$\text{new Kaff} = (\text{new SR} / \text{old SR})^2 * \text{old Kaff}$
Velocity Feedforward - Kvff	$\text{new Kvff} = (\text{new SR} / \text{old SR}) * \text{old Kvff}$
Friction Feedforward - Kfff	None
Position Feedforward - Kpff	None