Application Note #5491

Manual tuning using the velocity zone method

General information

There have been many methods developed over the years for tuning the loop constants of a servo axis, ranging from various auto-tuning methods to step-by-step manual tuning. All of these methods offer users various benefits, but all have the same goal; to simplify the process of tuning a PID filter while giving the user optimum performance. One manual method of tuning which works very well when combined with the GalilTools scope is what we call a velocity zone tuning method. This application note will guide the user through this approach, ending in a well tuned servo axis.

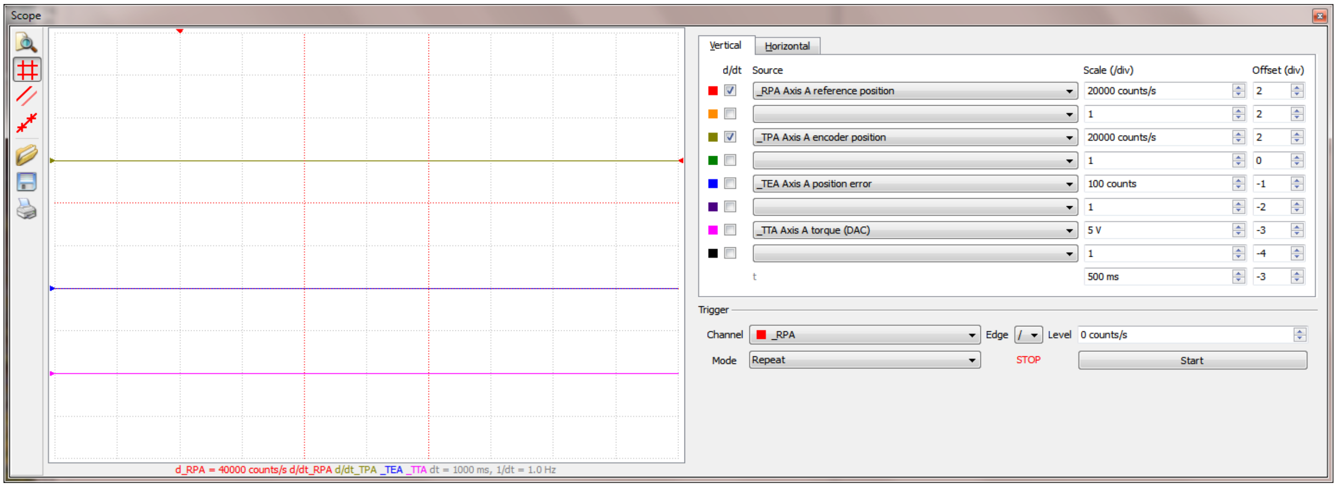
In traditional PID loop tuning, a position step response is plotted and the KD and KP term are increased up to a point just short of resonance or instability. Typically KD is increased first, followed by the KP term. However changes to KP may affect KD and vice versa, so many iterations are required before arriving at the final value. This approach results in a well tuned servo axis, but relies on a good deal of experience on the part of the user, who must be able to read the positional responses and determine which values to raise or lower.

The velocity zone tuning approaches tuning somewhat differently. The main feature of this tuning method is in isolating the velocity tuning parameters from the position parameters, and tuning those first. Specifically, the KD term which provides the derivative gain in the system as well as the FV term which applies a feedforward velocity gain are tuned first. These parameters are tuned based on the commanded velocity profile of the system. Once these values have been selected, the position error of the step response is used to select the appropriate KP (proportional gain) and FA (feedforward acceleration) terms. Finally, the position error information is again used to tune the KI (integral gain) and IL (integral gain limit) terms.

The benefit of this method is not that it necessarily yields better tuning results than traditional PID tuning. Rather, this method is easier for a basic user to follow along as a step-by-step manual tuning approach. That step-by-step approach is described below.

Step-by-step approach to velocity zone tuning

1. *Setup GalilTools*
   The key information to display when tuning a servo axis is commanded/actual encoder velocity, encoder position error and servo torque. Figure 1 shows how the scope is set up for initial tuning. Values for “Scale” are selected based on move length, and may need to be adjusted during tuning. Set the trigger for “Repeat” and click on “Start” to get the scope ready to collect data.



d_RPA = 40000 counts/s d/dt_RPA d/dt_TPA _TEA _TTA dt = 1000 ms, 1/dt = 1.0 Hz

	d/dt	Source	Scale (/div)	Offset (div)
🟥	☑	_RPA Axis A reference position	20000 counts/s	2
🟧	□		1	2
🟫	☑	_TPA Axis A encoder position	20000 counts/s	2
🟩	□		1	0
🟦	□	_TEA Axis A position error	100 counts	-1
🟪	□		1	-2
🟪	□	_TTA Axis A torque (DAC)	5 V	-3
⬛	□		1	-4
		t	500 ms	-3

Trigger

Channel 🟥 _RPA Edge / Level 0 counts/s

Mode Repeat

Figure 1: GalilTools setup

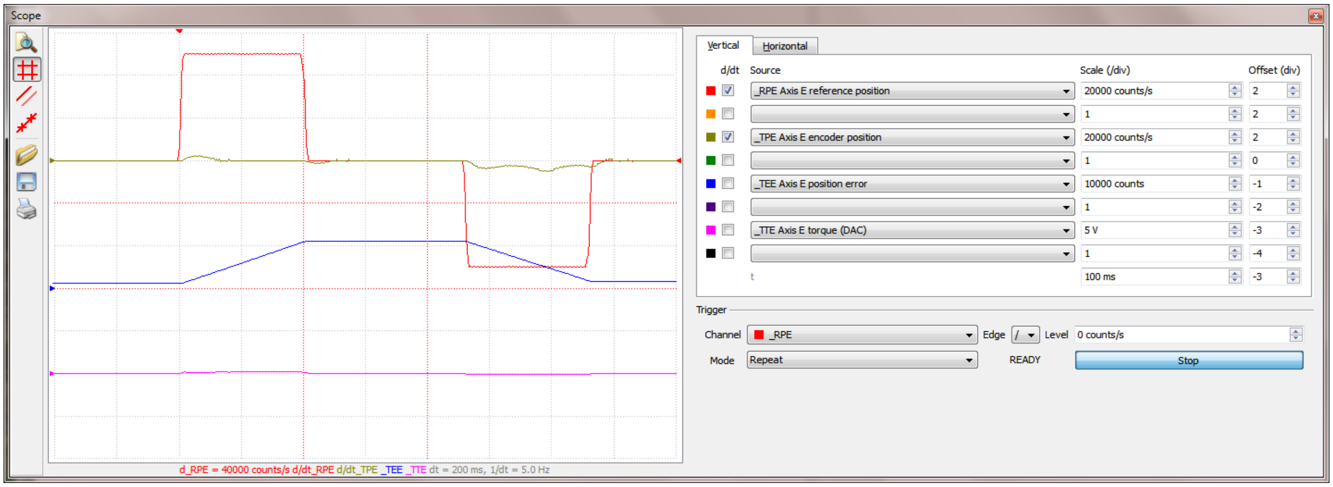
2. *DMC Code*
   This tuning method requires the motor to perform a short step move back and forth. The distance of this step move should be sufficient to break through static friction and fully accelerate an inertial mass. In addition, there should be some amount of steady state velocity in the profile (ie. No triangular profile). The velocity, acceleration and deceleration are selected based on maximums required for the application. Below is a sample program for this particular system being tuned. This program is then downloaded to the controller. Upon execution, this code will continually step the motor back and forth by the PA amount. The code may be stopped at any time via the "AB" or "ST" commands.

```
#A
AC 7500000
DC 7500000
SP 50000
PA 10000
BGX;AMX
WT 250
PA 0
BGX;AMX
WT 250
JP#A
EN
```

3. *Set initial parameters*
   Assuming the default gains are currently being used on the axis (KD64;KP6;KI0), set the KP term to 0 and leave KI at 0. Set KD to 10 to give a minimal derivative gain which will be acceptable for most systems.

4. *Perform step response*

With the DMC code downloaded to the controller, the step move is initiated by typing in the “XQ” command into GalilTools. The motor will make a small velocity move, trying to match the commanded velocity profile. For most systems, the KD of 10 will be insufficient to match the profile, so KD will need to be raised significantly (detailed in step 5). After the move has completed the scope will show the commanded velocity profile in addition to a trace for actual velocity profile, as shown in Figure 2:



	d/dt	Source	Scale (/div)	Offset (div)
🟥	☑	_RPE Axis E reference position	20000 counts/s	2
🟧	□		1	2
🟫	☑	_TPE Axis E encoder position	20000 counts/s	2
🟩	□		1	0
🟦	□	_TEE Axis E position error	10000 counts	-1
🟪	□		1	-2
🟪	□	_TTE Axis E torque (DAC)	5 V	-3
⬛	□		1	-4
		t	100 ms	-3

Trigger

Channel 🟥 _RPE Edge / Level 0 counts/s

Mode Repeat READY

d_RPE = 40000 counts/s d/dt_RPE d/dt_TPE _TEE _TTE dt = 200 ms, 1/dt = 5.0 Hz

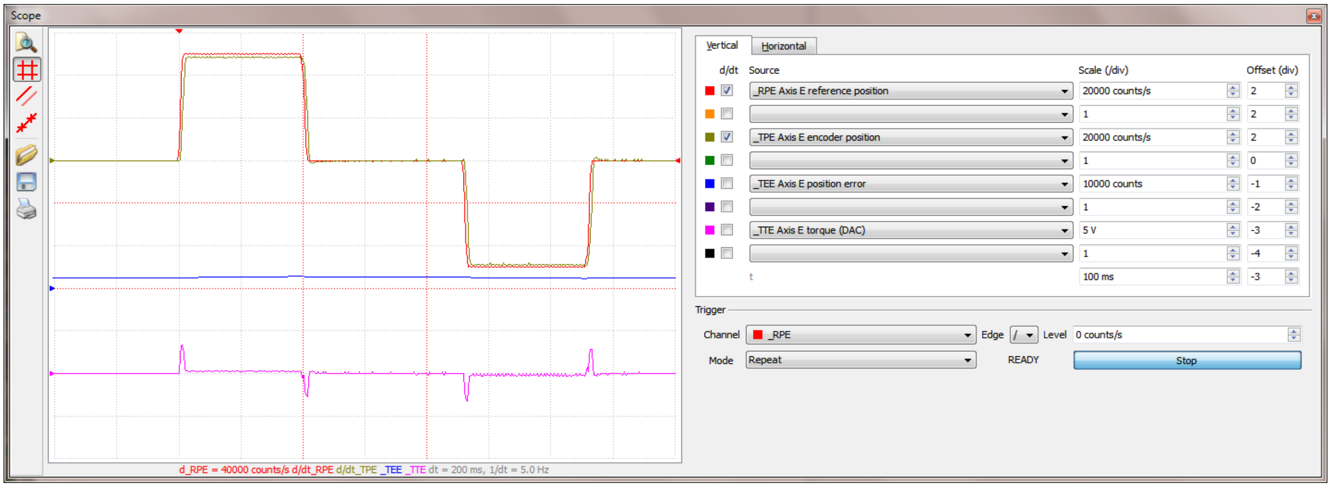
Figure 2: Step response with minimal KD

5. *Tune KD term*

The goal of this step in the tuning is to continually increase the KD term until the actual velocity profile nearly matches the commanded velocity profile (Figure 3). If the initial KD10 results in an actual velocity profile that is <50% in magnitude of the commanded velocity profile, the first steps can be large (ie. Increase from KD10 to KD50 and repeat). Otherwise steps of 5 or 10 are sufficient. As the actual velocity profile gets closer to the magnitude of the commanded velocity profile, the steps of the KD value should decrease to narrow in on a value. One of three things will indicate that the proper KD value has been selected:

1. The actual velocity profile is within ~5% of the magnitude of the commanded velocity profile.
2. Increasing KD no longer changes the shape or magnitude of the actual velocity profile.

3. Resonance in the system prohibits KD from increasing further. In this case, proceed to step 6 below for help in tuning PL term.



	d/dt	Source	Scale (/div)	Offset (div)
🟥	☑	_RPE Axis E reference position	20000 counts/s	2
🟧	□		1	2
🟫	☑	_TPE Axis E encoder position	20000 counts/s	2
🟩	□		1	0
🟦	□	_TEE Axis E position error	10000 counts	-1
🟪	□		1	-2
🟪	□	_TTE Axis E torque (DAC)	5 V	-3
⬛	□		1	-4
		t	100 ms	-3

Channel 🟥 _RPE Edge / Level 0 counts/s

Mode Repeat

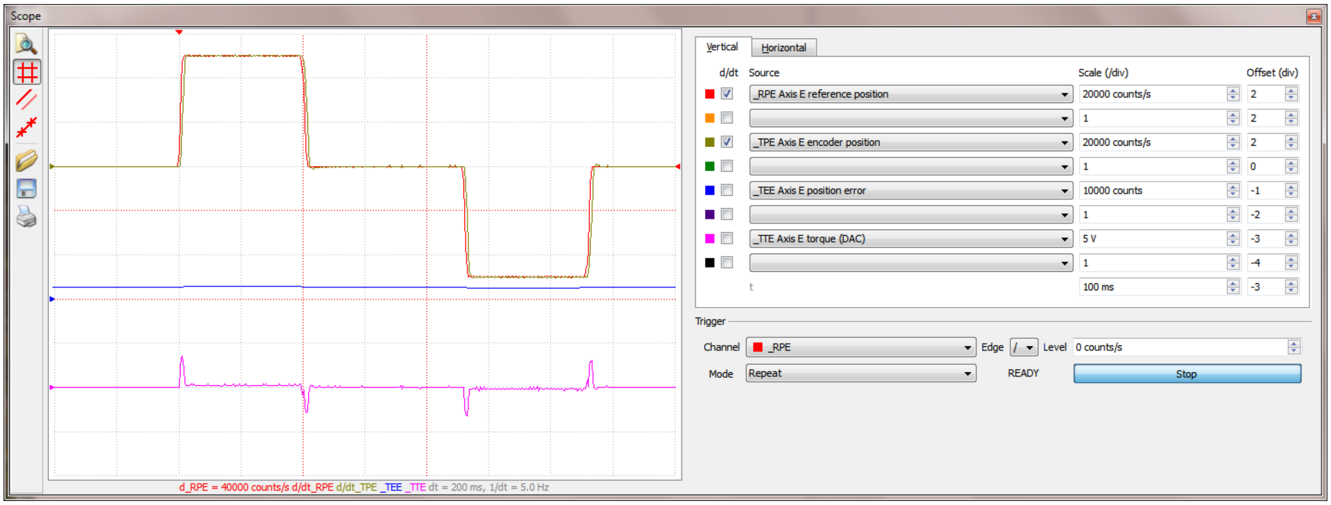
d_RPE = 40000 counts/s d/dt_RPE d/dt_TPE _TEE _TTE dt = 200 ms, 1/dt = 5.0 Hz

Figure 3: Step response with ideal KD

6. *If necessary, tune PL term*
   In certain systems, mechanical resonance can be magnified by the high frequency component of the KD term. This can turn minor resonances into actual position oscillations. The Galil motion controller has a low pass filter which is used to remove these resonances, thus allowing the user to increase the KD term for sufficient damping. This resonance can be heard as a high frequency buzzing, or can be seen on the actual velocity profile. If this situation occurs, the low pass pole filter is added, starting at a low value of PL0.1 and continually increasing while checking the step response. If the high frequency noise is removed, return to step 5 and increase KD. If the high frequency persists, raise to PL0.2 and check the step response. Continue this process until PL has removed the resonance, and KD has been raised to a sufficient value

7. *Tune FV term*

   While working with the velocity response from the system, it is convenient to also tune the feedforward velocity gain of the system. As seen in the plots of commanded and actual velocity in GalilTools, it is typically not possible to raise KD high enough to match the two plots 100%. The FV term is added such that the two plots match each other completely. Similar to the process for KD and PL, start FV with a low value of FV5. The actual velocity trace should increase. Continually raise FV until the two plots match (Figure 4).



	d/dt	Source	Scale (/div)	Offset (div)
🟥	☑	_RPE Axis E reference position	20000 counts/s	2
🟧	☐		1	2
🟫	☑	_TPE Axis E encoder position	20000 counts/s	2
🟩	☐		1	0
🟦	☐	_TEE Axis E position error	10000 counts	-1
🟪	☐		1	-2
🟪	☐	_TTE Axis E torque (DAC)	5 V	-3
⬛	☐		1	-4
		t	100 ms	-3

Trigger

Channel: 🟥 _RPE Edge: / Level: 0 counts/s

Mode: Repeat READY

d_RPE = 40000 counts/s d/dt_RPE d/dt_TPE _TEE _TTE dt = 200 ms, 1/dt = 5.0 Hz

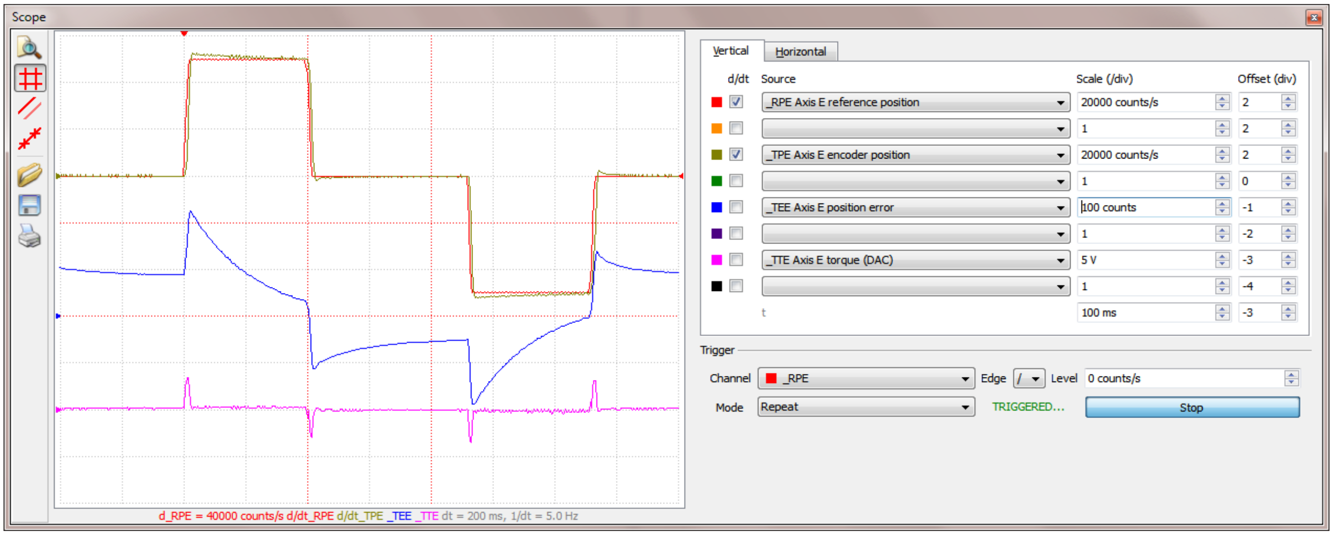
Figure 4: Ideal velocity profiles with FV and KD

8. *Tune KP term*

   At this point, the velocity terms are tuned properly and the positional terms can be tuned. For safety, the position error of the system should be zeroed at this point to prevent jumping of the motor when adding the initial KP. Stop the step response with “ST”, and issue “MO” followed by “SH”. This will zero position error and allow the user to assign an initial KP value of 5. Run the step response with “XQ”, and evaluate the position error trace. On most systems, the KP value of 5 will be insufficient for proper operation, so there will be excessive position error (ie. > 100 counts of error). Similar to the velocity terms, begin increasing the KP value in increments based on the amount of position error. If the position error is greater than 100, continue to increase KP in increments of 5 or 10. As the position error decreases, decrease the KP increments. The signs that the KP term is getting too high for the system are as follows. If these symptoms occur, back off from the KP value and move on to the next step.

   1. Position error trace shows unstable motor behavior.
   2. Position error peaks at acceleration and deceleration begin to increase, or are no longer decreasing.

3. Motor is getting audibly noisy, which can be seen as noise on the position error trace.


Scope

	d/dt	Source	Scale (/div)	Offset (div)
🟥	☑	_RPE Axis E reference position	20000 counts/s	2
🟧	□		1	2
🟩	☑	_TPE Axis E encoder position	20000 counts/s	2
🟩	□		1	0
🟦	□	_TEE Axis E position error	100 counts	-1
🟪	□		1	-2
🟪	□	_TTE Axis E torque (DAC)	5 V	-3
⬛	□		1	-4
		t	100 ms	-3

Trigger

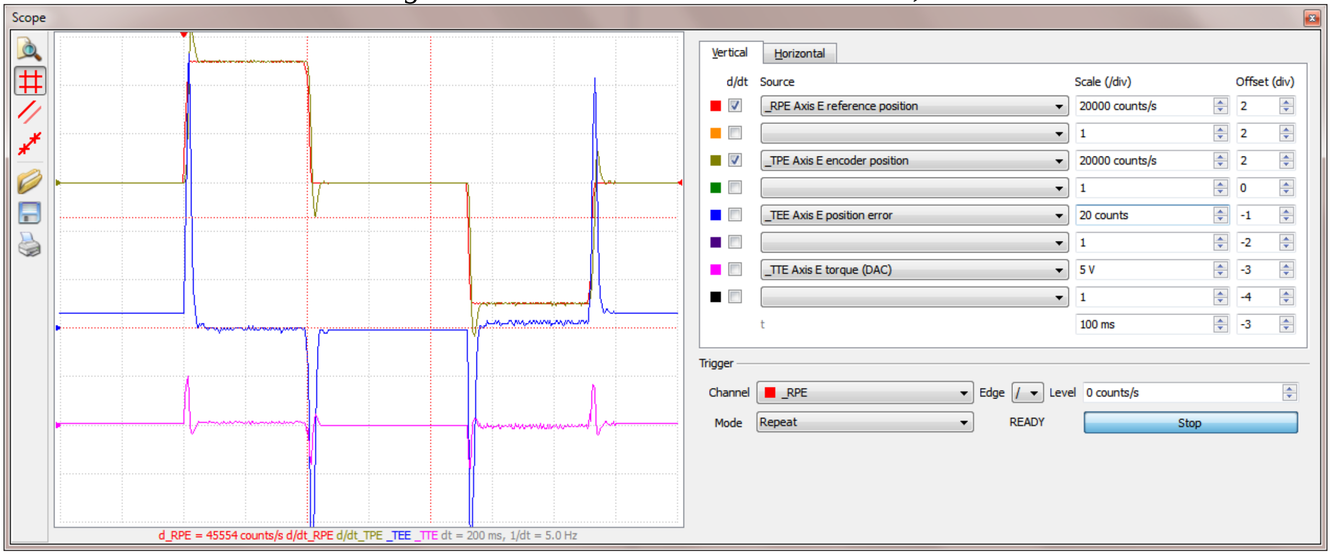
Channel: _RPE Edge: / Level: 0 counts/s

Mode: Repeat TRIGGERED...

d_RPE = 40000 counts/s d/dt_RPE d/dt_TPE _TEE _TTE dt = 200 ms, 1/dt = 5.0 Hz

Figure 5: Position error with KP5, Scale 100 counts/div

Figure 6: Position error with ideal KP80, Scale 20 counts/div


Scope

	d/dt	Source	Scale (/div)	Offset (div)
🟥	☑	_RPE Axis E reference position	20000 counts/s	2
🟧	□		1	2
🟩	☑	_TPE Axis E encoder position	20000 counts/s	2
🟩	□		1	0
🟦	□	_TEE Axis E position error	20 counts	-1
🟪	□		1	-2
🟪	□	_TTE Axis E torque (DAC)	5 V	-3
⬛	□		1	-4
		t	100 ms	-3

Trigger

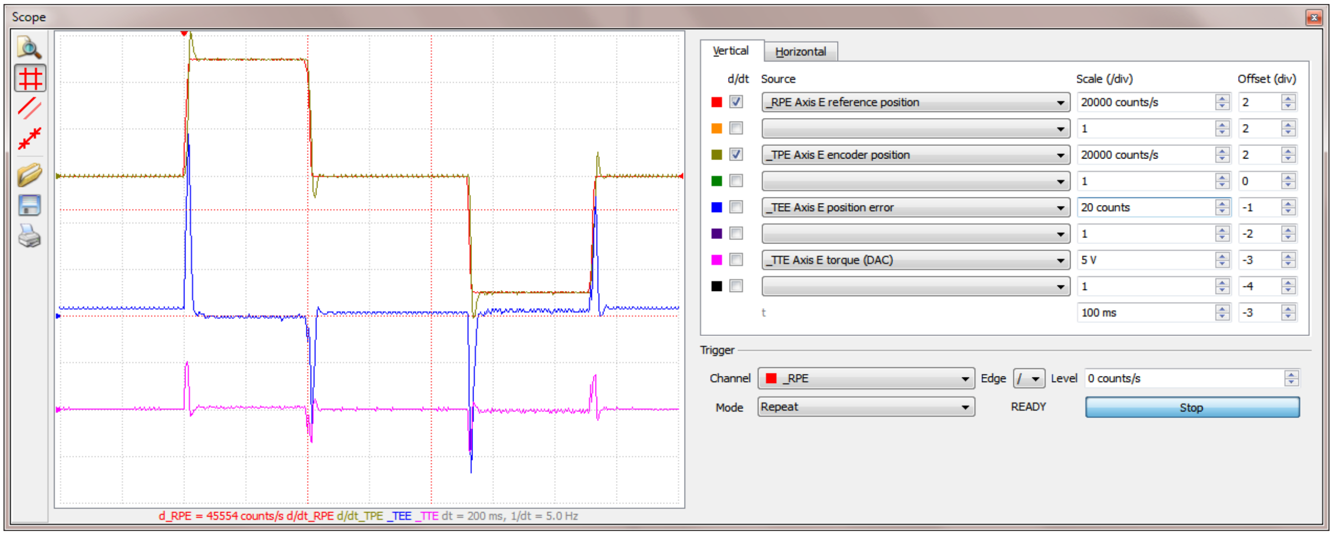
Channel: _RPE Edge: / Level: 0 counts/s

Mode: Repeat READY

d_RPE = 45554 counts/s d/dt_RPE d/dt_TPE _TEE _TTE dt = 200 ms, 1/dt = 5.0 Hz

9. *Tune FA term*

   The scope now shows that position accuracy during and after the move have improved significantly, and should be minimized for this system. However, due to the high KP term there is most likely a large position error spike at both the acceleration and deceleration phases of the profile (as seen in Figure 6). The feedforward acceleration gain (FA) on the Galil controller is implemented to help remove that spike. Similar to other filter parameters, start with a low FA of 0.5 and begin increasing. As FA increases, the positive and negative spikes at the acceleration and deceleration will begin to diminish. If the spikes ever invert from their original direction, this is a sign that FA has become too high.



	d/dt	Source	Scale (/div)	Offset (div)
🟥	☑	_RPE Axis E reference position	20000 counts/s	2
🟧	☐		1	2
🟫	☑	_TPE Axis E encoder position	20000 counts/s	2
🟩	☐		1	0
🟦	☐	_TEE Axis E position error	20 counts	-1
🟪	☐		1	-2
🟪	☐	_TTE Axis E torque (DAC)	5 V	-3
⬛	☐		1	-4
		t	100 ms	-3

Trigger

Channel: _RPE Edge: / Level: 0 counts/s

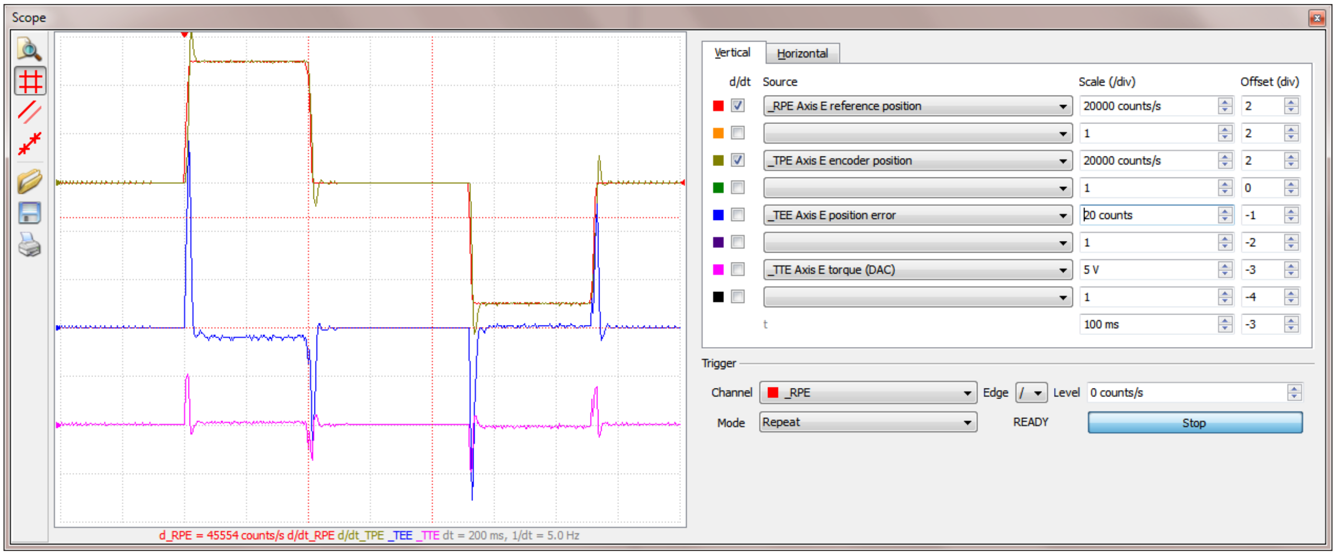
Mode: Repeat READY

d_RPE = 45554 counts/s d/dt_RPE d/dt_TPE _TEE _TTE dt = 200 ms, 1/dt = 5.0 Hz

Figure 7: Position error with FA

10. Tune KI and IL

The final step in tuning the servo motor is to increase the KI term to remove any steady state error. Due to friction and inertia in systems, motors when driven to a position may not settle on the exact commanded position (ie. TE ≠ 0). The KI term is used to overcome this friction and zero out the position error. Figure 7 above shows that during the rest between moves, the 2-3 counts of position error is never removed from the system. This will be even greater on systems with significant friction. The IL allows the user to set a limit on the integral term, as well as selectively "freeze" it out during motion. For this tuning, we will assume the goal is high speed step and settle. In those cases, freezing the integrator during motion is ideal. This is first accomplished by setting IL to a negative value. Start with "IL-9.9" to allow full torque from the integrator. Next, set a minimal KI value of 0.5 and view the effect on the position error. Continue to increase KI to minimize steady state position error, which is the 250msec between consecutive moves, to as close to zero as possible. If the position error begins to show oscillations, decrease IL until the oscillations are removed.



	d/dt	Source	Scale (/div)	Offset (div)
🟥	☑	_RPE Axis E reference position	20000 counts/s	2
🟧	☐		1	2
🟫	☑	_TPE Axis E encoder position	20000 counts/s	2
🟩	☐		1	0
🟦	☐	_TEE Axis E position error	20 counts	-1
🟪	☐		1	-2
🟪	☐	_TTE Axis E torque (DAC)	5 V	-3
⬛	☐		1	-4
		t	100 ms	-3

Channel 🟥 _RPE Edge / Level 0 counts/s

Mode Repeat READY

d_RPE = 45554 counts/s d/dt_RPE d/dt_TPE _TEE _TTE dt = 200 ms, 1/dt = 5.0 Hz

Figure 8: Position error with KI and negative IL

Additional Notes

1. The recommended GalilTools scope setup used for this tuning has a fourth trace enabled for Torque. This value represents the +/-10V output from the Galil to the amplifier, with +10V being full positive torque and -10V being full negative torque. This should always be monitored for saturation. If this value ever approaches the maximum +/-10V, this indicates that the system is being pushed too hard. Either decrease the profile parameters (SP, AC and DC) or change the amplifier gain in the system for more torque.
2. As mentioned in the KI tuning section, this system was tuned for high speed step and settle. In this case, the critical number is how quickly the motor moves and settles into position. Position error during motion is less critical. If a system needs to be tuned for accurate position during motion, it is advised to keep the KI term active at all times, which is done by keeping a positive value for IL (ie. "IL9.9").

In that case, tuning the KI term is similar, starting with a low value for KI and increasing until position error is minimized for both motion and settling. If the position error begins to oscillate, or KI can't be raised high enough to reduce the position error, the IL value should be reduced.

3. The motion controller also has a notch filter which can be used to remove system resonances. A notch filter is used in cases where a machine has a resonance of a certain frequency which prohibits proper tuning of the system. In those cases, the notch filter is used to remove the resonance and tuning is performed as above. For additional information on tuning the Galil notch filter, please see application note 2431 on the Galil website (link: http://www.galilmc.com/support/appnotes/optima/note2431.pdf)