CHAPTER FOUR

Tuning

IN THIS CHAPTER

- PID Tuning – Description
- TQ10X Tuning Procedure
- Configuring an In Position Window

Tuning and Performance

You must tune the parameters on the TQ10X's *Proportional Integral Derivative* (PID) filter for optimum system performance. A properly tuned system will exhibit smooth motor rotation, accurate tracking, and fast settling time.

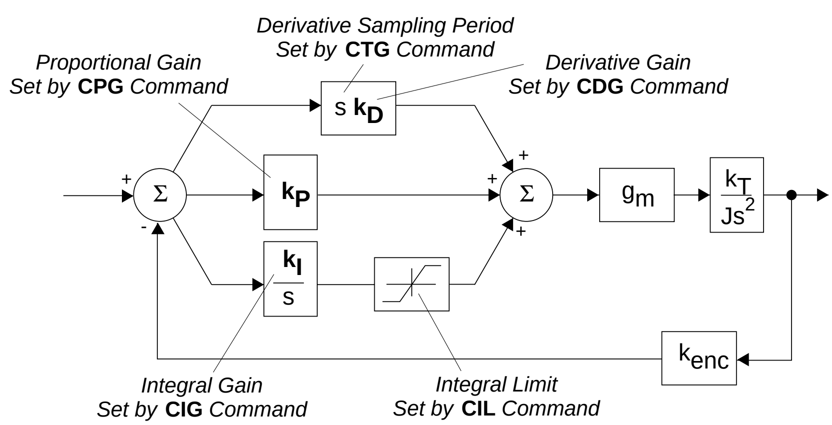
All tuning is performed via RS232-C communications.

PID Tuning

The TQ10X generates a move profile based upon the user supplied acceleration, velocity, and distance commands (**A**, **V**, and **D**). At each servo sampling period (every 266 microseconds), the TQ10X calculates the position the motor should reach as it follows the move profile. This is called *commanded position*, and is one of two inputs to a summing node. Position information from the encoder, which is called *actual position*, is the other input to the summing node. During a typical move, actual position will differ from commanded position by at least a few encoder counts. When actual position is subtracted from commanded position at the summing node, an error signal is produced. The error signal is the input to the PID filter.

The position specified by the distance command (**D**) is called the *target position*. During the move, commanded position is not the same as target position. The commanded position is incremented each sampling period. When it finally matches the target position, the move is over.

The servo block diagram is shown below.



As the figure shows, you can adjust five different parameters to tune the PID filter. The relevant commands are:

CPG	Configure Proportional Gain
CDG	Configure Derivative Gain
CTG	Configure Derivative Sampling Period
CIG	Configure Integral Gain
CIL	Configure Integral Limit

To tune the system, you will iteratively increase **CDG**, **CTG** and **CPG** to their optimum values. If necessary, you will also increase **CIG** and **CIL**.

In general, you will set **CDG** and **CPG** as high as necessary, and **CIG** as low as possible. Trade-offs between response time, stability, and final position error will dictate the values you select. For loads that vary during operations, you can download new parameters by using buffered versions of the five tuning commands (**BCPG, BCDG, BCTG, BCIG, BCIL**).

WARNING
During servo tuning, the system can undergo accidental and violent movement due to improper gain settings and programming errors. Please use extreme caution while prototyping.

Each tuning parameter is described in the following sections.

CPG – Proportional Gain

Proportional gain provides a torque that is directly proportional to the *magnitude* of the error signal. Proportional gain is similar to a spring—the larger the error, the larger the restoring force. It determines the stiffness of the system and affects the following error. High proportional gain gives a stiff, responsive system, but can result in overshoot and oscillation. Damping—provided by derivative gain—can reduce this overshoot and oscillation.

CDG – Derivative Gain; CTG – Derivative Sampling Period

Derivative gain provides a torque that is directly proportional to the *rate of change* of the error signal. The previous error is subtracted from the present error each sampling period. The difference represents the error's instantaneous rate of change, or *derivative*. The difference is multiplied by the value set by the **CDG** command, and the product contributes to the motor control output.

Derivative gain opposes rapid changes in velocity. It will dampen the resonance effects of proportional gain. With higher derivative gain, you can use higher proportional gain.

You can use the **CTG** command to make the derivative sampling period longer than the system's sampling period. The system sampling period—266 µsec—is the period between updates of position error, and cannot be changed. The derivative sampling period is an integer multiple of the system sampling period. It can range from 266 µsec to 68 msec, in increments of 266µsec (for example: CTG0 = 266 µs, CTG1 = 532 µs, CTG2 = 798 µs, etc.).

With a longer derivative sampling period, more time elapses between derivative error measurements. The difference between previous and present error is still multiplied by the CDG value. The product contributes to the motor control output every *system* sampling period, but is only updated every *derivative* sampling period. This gives a more constant derivative term and improves stability. Low velocity systems in particular can benefit from a longer sampling period.

Because of stability considerations, however, the derivative sampling period should be no longer than one tenth of the system mechanical time constant. This means many systems must have low values of **CTG**.

CIG – Integral Gain; CIL – Integral Limit

Integral gain provides a torque that is directly proportional to the sum, over time, of the error values—the *integral* of the error. The controller reads the error value every sampling period, and adds it to the sum of all previous error values. The sum is multiplied by the value set by the **CIG** command (Integral * CIG), resulting in the *integral term* which contributes to the motor control output every system sampling period.

Integral gain can remove steady state errors that are due to gravity or a constant static torque. Integral gain can also correct velocity lag that can occur in a constant velocity system.

If error persists during a move, the sum of the error values may be quite high at the end of the move. In this case, the torque commanded by the integral gain can also be very high, and cause overshoot. This effect is called *integral windup*. Integral windup can sometimes cause very aggressive motion, you may want to limit this effect. CIL sets an upper limit on the integral, which in turn limits the integral term **(Integral * CIG)**.

Tuning the System

In the procedure given below, you will systematically vary the tuning parameters until you achieve a move that meets your requirements for accuracy and response time.

Special Considerations when Tuning with SM or NeoMetric Motors

If you use a high performance motor (peak current rating greater than three times the continuous current rating), be careful not to overheat the motor while tuning your system. If you accidentally choose tuning gains that cause motor instability, excess motor current can quickly overheat and damage the motor—even before the thermostat can trigger the motor overtemperatu re circuit.

CAUTION

For initial tuning with an SM or NeoMetric motor, set peak current DIP switches at twice the motor’s continuous rated current, or less. Otherwise, high peak currents during instability may cause motor overheating and damage.

To avoid damage, we recommend a tuning procedure that may differ from methods you have used before. Instability sometimes *does occur* during tuning; to avoid the damaging currents that instability can cause, reduce the peak current *before* you begin the tuning process. Then, as you refine your tuning values, you can gradually increase peak current. These steps are included in the tuning procedure described below.

Preparing the System for Tuning

Before applying power and tuning the drive, complete the following steps.

Setup Procedure:

① **Heatsink your motor:** This is especially important in temporary “bench top” procedures. SM and NeoMetric Motors dissipate excess heat through their faceplate; to ensure proper motor cooling, the faceplate must be mounted to a heatsink.

② **Reduce peak current:** Using the drive's DIP switches, set the peak current at a level that is less than twice the motor’s continuous current rating.

- 4.4 amps for SM motors with **-A** windings
- 7.4 amps for SM motors with **-B** windings
- 6.0 amps for N0701D and N0702E motors
- 8.9 amps for N0702F motor

This helps protect SM and NeoMetric motors from overheating when you begin tuning the drive.

③ **Prepare to Disable the Drive Quickly:** Wire a normally closed switch between the **ENABLE IN** input and ground. Use this switch to quickly and reliably disable the drive if the system becomes unstable during tuning.

Avoid Instability

After you change current settings, adjust a tuning parameter, or command a move, closely monitor the motor. If there is any sign of uncontrolled instability, use the switch described in *Step 3* above to quickly disable the drive, and avoid motor damage.

Tuning Procedure

You will achieve best results by making a short, repetitive test move, using high acceleration and high velocity. Because you will start with reduced peak current, the motor may not be able to make the move called for in your application. After you initially tune the system with the short test move, you can try your actual application move and choose final tuning values.

① **Issue a RETURN TO FACTORY SETTINGS command (RFS)**

The RFS command will reset the gains to their default values (CDG240, CTG0, CPG16, CIG2, CIL2, CPE4000)

② **Decrease CIG**

Set the integral gain to zero (**CIG0**).

③ **Command a Short, Repetitive Test Move**

Program the drive to continuously execute a short, repetitive test move. Depending upon your system's mechanics, the move can be back and forth, or in a single direction. Program a five second delay between each move. Use a high acceleration, such as A5000 (or the maximum you expect to accelerate the load); and use a high velocity, such as V100 (or the maximum velocity you expect to move the load). The motor may not reach the commanded velocity during the short move; this does not matter, and will not affect your tuning procedure. An example program is shown below.

Command	Description
CITØ	Sets “In Position” time to Ø (very short)
CEW1ØØØ	Sets “In Position Error Window” to 1ØØØ (very wide)
LD3	Disable limits (use only if appropriate for your system)
MN	Sets positioning mode to preset
A5ØØØ	Sets acceleration to 5ØØØ revs/sec^2 (very high value)
V1ØØ	Sets velocity to 1ØØ revs/sec (very high value)
D4000	Sets distance to one rev (for a 1ØØØ line encoder).
LØ	Loop continuously
G	Executes the move (GO)
T5	Wait 5 seconds at end of move
N	Ends the loop

If the mechanics of your system do not permit the moves described above, modify your test move accordingly.

④ **Observe the Results of the Move**

Closely observe the motor shaft and evaluate motor performance at the end of the move, as indicated by response time, end of position overshoot, following error, and so on. If the move is successful and your motion requirements are satisfied, you do not need to adjust tuning parameters or increase peak current any further—you may proceed to *Step 7*.

Prepare to disable the drive quickly if the motor becomes unstable!

⑤ **Adjust Tuning Parameters**

Vary the tuning parameters to improve motor performance and to achieve *satisfactory* motion—only tune for absolute maximum performance if the application requires it. Typical responses and suggested adjustments are:

Sluggish Motion – Increase proportional gain (**CPG**) or decrease derivative gain (**CDG**) to make the motor respond more quickly. Use caution when you increase **CPG**—too much will cause the system to oscillate and become unstable. If you increase **CPG** and your system becomes unstable, disable the drive immediately.

Oscillatory or Erratic Motion – Increase derivative gain (**CDG**) or decrease proportional gain (**CPG**) to help damp out oscillatory motion. Too much **CDG** will cause a sluggish or overdamped response; excessive **CDG** may cause the system to become unstable. If you increase **CDG** and your system becomes unstable, disable the drive immediately. You can try increasing the derivative sampling period (**CTG**) to control instability. For most applications, **CTG** will not need to be higher than 4.

Steady State Errors – See *Step 9* for information about controlling steady state errors by adjusting integral gain and integral limit parameters. Because integral gain reduces stability, it should only be adjusted after you determine settings for **CPG, CDG,** and **CTG**.

If you still need to improve performance after you adjust **CPG, CDG,** and **CTG**, proceed to *Step 6*.

⑥ Increase Peak Current

Using the DIP switches, increase peak current to the next level. Do not exceed three times the motor's continuous current rating. Closely monitor the motor immediately after you increase peak current. Be prepared to disable the drive if the system shows any signs of instability.

You can change the DIP switches with power applied—the drive will immediately sense the new current setting. For safety reasons, however, we recommend removing power from the drive before changing DIP switches.

⑦ Repeat Steps 4 – 6

With higher peak current, you can evaluate system performance and readjust tuning parameters if necessary. Continue repeating Steps 4 – 6 until you achieve satisfactory performance.

⑧ Choose Final Tuning Values

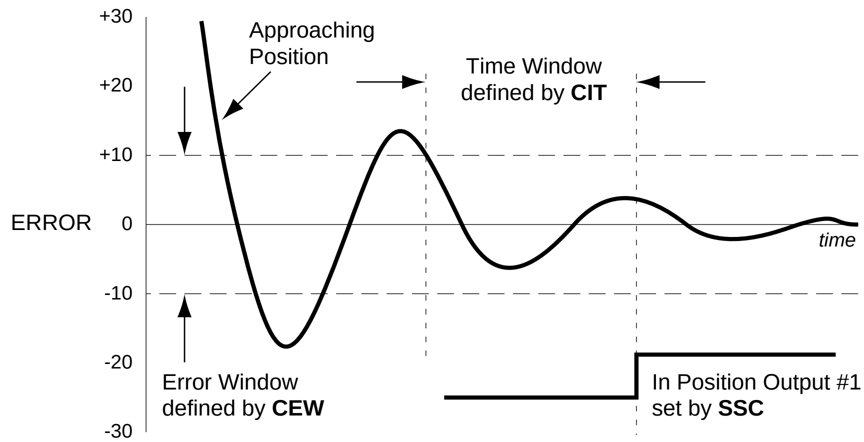
Program your applications's actual move, and select final tuning values based upon the performance you observe.

⑨ Determine Integral Gain (CIG) and Integral Limit (CIL) Values

Increasing integral gain **(CIG)** will reduce position errors at the end of the move and velocity errors during the constant velocity portion of the move. However, high values for **CIG** can cause the system to become unstable. If you increase **CIG** and your system becomes unstable, disable the drive immediately. In general, you should set **CIG** to the *lowest* value that will correct following errors and static position errors, but not increase overshoot or settling time. In a system without static torque loading, a **CIG** of zero may be appropriate.

Configuring an In Position Window

You can define an In Position Window, and use it to indicate that the preceding move is done. Two commands—**CEW** and **CIT**—determine the height and width of the window. A third command—**SSC**—can turn on output #1 when the In Position criteria are met.



As the drawing shows, **CEW** defines the position error window at the end of a move. **CIT** specifies the length of time the motor must be within the error window. The motor is In Position when three conditions are satisfied:

① The controller algorithm is finished (no input position command)
② Position error is less than that specified by the **CEW** command
③ Condition ② above has been true for the length of time specified by the **CIT** command

If **SSC** has been set to 1, output #1 will turn on when these three conditions have been met. You can use output #1 to trigger external hardware from the In Position condition. The output will stay on until the next move command is issued, such as **GO** or **GO HOME**.

(*Note:* If the motor is held (mechanically, or against an end stop), and **CPE** is greater than **CEW**, the motor may become "trapped" between **CPE** and **CEW**: it will not execute the next move. In this rare situation, two things are happening: 1.) **CPE** is not violated, and therefore no position error fault occurs; 2.) in position criteria are not met.

If you were to execute a **1R**, the response would be ***B**, which means the drive is "busy" waiting for the move to be over. Why doesn't the drive force the motor to finish the move? The motor is somehow held. To correct this situation, try touching the motor; this may complete the move, and the drive may execute the next move. Or, execute a **DPA** to read actual position, and verify that the move is not complete. You can also execute a **KILL** to reset the positions, and then do the next series of moves.)