

The 'Science' of Tuning
By: Chuck Raskin P.E. CMCS
9/2/98

My most often asked question is how to properly tune a motor. It has always been considered some form of 'Black Magic' to those who have not had many opportunities to perform a tuning operation, and a painfully learned technique to those who have. Although there are many good tuning methods and self-tuning algorithms available, most of them are focused to specific brands of motion controllers or specific types of operations. I have to this point not found a good generic tuning tutorial which covers ALL types of gain algorithms.

Personally, I hate to use math to solve problems and avoid it as much as possible, but sometimes you have no choice. Bode plots are interesting, but are not designed for novice users. As a matter of fact, the Bode plot puts a lot of effort into determining what MIGHT happen in a given system ONLY if ALL system parameters are known. But rarely, in motion systems, is everything known! This is because of the sheer number of parameters that affect the tune. Think about it. The type of grease on the bearings, the type of bearings, mechanics in general, the machine, the type of motor, amplifier, friction, environment, motion controller, computer, etc.

When tuning, rather than teaching the user how to dynamically tune correctly, each vendor has come up with his or her own special way of conquering their tuning situation. The result is that the user watches and does, but learns very little, and thus the dilemma. For example, what is Notch filtering, and when it needed? What are the benefits of Velocity and Acceleration FeedForward? Modeling of systems works great until an unforeseen condition is realized. You set up the filter conditions based on measurements taken, then HOPE that over time the conditions remain the same (or close to it). If they don't, then re-tuning is required to once again stabilize the system. I wonder just how many vendors have actually tried to tune a system in a hostile environment where the elements are stacked against you.

Finally, when tuning a system, there can be many different values entered into a given gain algorithm which will work, but only one that will work over the broadest possible range of load and motion variations. By achieving OPTIMUM gain values all other types of gain modifiers may not even be necessary (Notch filter, Adaptive tuning, FeedForward, etc.). But if they are, then they can simply be used to ASSIST the main gain structure in doing its job, rather than taking the place of the main structure.

The question here is how do we achieve the optimum primary gain control values?

The most popular style of gain algorithm in use today is the PID (Proportional-Integral-Derivative). What is so effective about the PID is its ability to compensate for continuous, long term, and instantaneous motion situations for a very broad range of applications, including process and motion control. Motors cannot react in zero time. They are smooth, continuous, devices. If a discon-

tinuous force is imparted to a motor, the motors velocity will change based on an overall system time constants (electrical and mechanical), and will do so in a continuous fashion (smoothly).

Is it possible then to speed up the reaction to a disturbance in order to maintain a more stable velocity profile? In addition, how can we ensure that the cancellation of a disturbance force is accomplished in such a manner that the motor will not impart an undesirable Jerk onto the machine or load?

The answer to the first question is yes, and the answer to the second is; With proper selection of the motor, amplifier, feedback, motion controller, and gain tuning values, which must all be coordinated in the big system picture. To simplify, with proper planning!

The scope of this write up, however, is not intended to cover system planning, but to define a method for tuning any PID gain algorithm in order to yield the optimum response to any system design. One thing to note is that regardless of the algorithm in use, the tuning direction is the same for its proportional, long term, and instantaneous reaction terms.

Defining the PID and Generally Used Terms

Since this tutorial is intended for the novice user, as well as the experienced, this section is intended to define the PID terms as I them throughout this discussion. It is also intended to show how the PID elements work using an analogy, I guarantee, you won't forget. For those of you who do understand the PID, skip to the next section at will.

Term Definition

K - A generally accepted variable used to denote gain, an arbitrary multiplier, or a constant. Here, it will be used as a gain multiplier. A lower case letter, such as Kp for Proportional gain will indicate the specific gain.

Update - An arbitrary but evenly spaced period of time, which controls how often the motion control algorithm will be accomplished. The DAC output will also be revised in this time interval.

Trajectory - The calculated, or desired motion.

Velocity move - No final stopping position is given to the motion controller. When a START command is issued, the motor will move indefinitely until it is commanded to stop. When STOP is commanded, the motor will follow a pre-calculated deceleration slope until it comes to a halt.

Position move - The position move is similar to the Velocity move except that a predefined stopping point is given to the motion control algorithm. Calculations will be done to determine when it is time to begin the deceleration profile to ensure the actual stopping point of the motion is at the desired target position.

PosFol/Err - The difference between the calculated trajectory position (desired), and the ctual motion (system, load, etc.) position.

DAC - The DAC (Digital to Analog Converter) is the electrical interface between the computer and the motor amplifier. DACs are available in three popular sizes: 8, 12, and 16 bit. The bit count partitions the total peak-to-peak output voltage swing into 256, 4096, or 65536 DAC steps, respectively. The more DAC bits, the finer the analog voltage resolution.

PID Element Definition

The classical PID structure consists of four independent elements used to produce the control signal that will be applied to the motor amplifier:

DAC output Value - This is the result actually applied to the motor control amplifier.

$$\text{DAC Value} = \text{SetPoint} + \text{Proportional} + \text{Integral} + \text{Derivative}$$

SetPoint - This is a user-preset value that can be used to offset the DAC output. For example, it can be used to offset the effect of gravity in vertical systems. If used, it is directly applied to the DAC output value.

$$\text{SetPoint} = \text{Constant value}$$

Proportional - This value is derived by multiplying the Kp gain term and the PosErr.

$$\text{Proportional} = K_p * \text{PosErr}$$

Integral - This value is derived by adding the newly computed PosErr to a PosErr totalizer, and then multiplying the total value by the Ki gain term.

$$\text{PosErr Total} = \text{Current PosErr Total} +/- \text{New PosErr}$$

Then: $\text{Integral} = K_i * \text{PosErr Total}$

Derivative - This value is derived by multiplying the Kd gain term and the difference between the PosErr determined in the current update and PosErr determined in the previous update.

$$\text{Derivative} = K_d * (\text{Current PosErr} +/- \text{Previous PosErr})$$

In addition, some motion controllers allow the Derivative element to be updated in a different time interval than other gain elements. This will generally increase the dynamic range of control capability.

A PID Analogy

To understand how to tune the PID, you first need to understand what the PID gain loop is up against when coordinating motion. The following analogy will allow you to better grasp how the elements interact.

Have you ever read about or tried to think of an analogy for the PID gain structure you could quickly relate to? The generally accepted analogy relating shock absorbers to Kd and springs to Kp has not allowed for good PID understanding since not everyone is mechanically inclined. In addition, an analogy for Ki to this point has yet to be solidly defined. So, let's begin understanding PID gain structures by relating the PID elements to something we are all intimately familiar with . . . our body.

Activating your Kp

Stand up straight with your arms at your sides. Carefully lift one foot off the floor, and place it behind the ankle of the other. Immediately, you will feel your supporting ankle reacting to changes in your body's balance. This is your body Kp error correction system reacting to instantaneous changes in balance your mind is sensing. The update period for this reaction is the time it takes for you to mentally record and correct for the changes in balance.

Your mind is using its learned Kp gain value to compensate for any shift in your balance. As you "sense/feel" your balance change, your mind multiplies your Kp value by your balance shift, or PosErr, and then applies the adjustment to your ankle. If you consciously put your learned Kp to zero, your ankle (system) will stop reacting and you will fall. If on the other hand you raise your Kp too high, your ankle will constantly overreact (i.e., oscillate out of control). This action is exactly how the Kp will respond in your motion system.

Activating your Kd

Slowly raise your arms until they are pointing to the left and right, parallel to the ground. Try to maintain them in this position for the remainder of the analogy. Note this position as Kd equal to zero. After a few moments, or minutes, you'll notice that your arms will have shifted (or be shifting) to new positions in order to help maintain your balance while standing on one foot. Also notice that your arms may maintain the new position for a period of time longer than the normal update period being used by your ankle Kp. This time interval is sometimes referred to as the Kd sample time or DST.

Your arm(s) 'adjust-and-hold-over-time' method of reacting to the body change in balance is its Kd reacting to a projected error based on previous error differences (or velocity changes for the motion controller). Note that your arms are responding to your sense of 'feel' of what you project will happen, not to what is actually happening as the Kp does. The anticipation of falling based on your body rate of change in vertical attitude is the same as a motion controller sensing a change in velocity over time (acceleration). Both cases are predicting what is expected to happen if the existing condition is allowed to prevail.

Activating your Ki

At this point, has your body torso also shifted in position? This is the body Ki. The Ki term builds slowly over time and as well, reduces to zero slowly over time. Note that as the Ki term is invoked, your ankles (Kp), and arms (Kd) never stop correcting. All three work as a team. Also note, however, that as one gain term takes over the control of the balance for a given moment, the other gain term effects are reduced, and might even go to zero.

The long term Ki effect will readjust itself over time. As your body moves to 'catch up' to where your mind (the controller) says it should be (standing), the newly computed Ki output value may still be rather large (your body is still leaning or bent over). In time new position adjustments produced by the Ki will become relatively small in comparison to the total Ki reaction first encountered.

Once your body has stabilized, the Ki term will do what is required to restore you to your

initial upright position (zero following error). Notice that your Kp and Kd body elements meanwhile are still working to help you balance while the Ki element is working toward straightening you up.

It should be obvious that if the totaled Ki error is allowed to grow to a value too large for your body to handle, you will lose control and fall. In a motion system, however, too much Ki will react as a "rubber-band" oscillating back and forth around the desired position point. To prevent an excessive Ki reaction, your mind invokes a limiting value to the growth of the Ki result, generally called "Integral Limit" (IL). Your body will try not to lean beyond this limit.

Activating your FeedForward Gain

Actually, this gain loop has been in effect since the beginning of the exercise! As you were 'thinking' about your balance and what your Kp, Ki, and Kd were doing, you were also contemplating (projecting) balance problems. You were making adjustments for these projections before they even occurred! You were "looking ahead", so to speak, at potential instability and making corrections based on what your balance feedback device(s) (eyes, ears, and body 'feel', etc.) were telling you.

Tuning the PID

As previously stated, the purpose behind any gain algorithm is to produce smooth, stable motion. As systems become more responsive a more defined procedure for tuning the gain loop will be required. It is a must to use well-established tuning procedures instead of widely diverse methods, or guessing.

The following PID tuning procedure has been used extensively on both process and motion control systems. The key to this or any approach is to completely understand the gain algorithm and tuning methods, which makes the tuning intuitive. You must understand not only what the gain elements do, but in what order to adjust them and at what point on the trajectory profile to test them for proper adjustment. It's extremely important to learn how to "read" what the motion is "telling" you.

The Tuning Procedure

To expedite remembering all of the items, which will need to be found to complete this tuning approach, have a pencil and paper ready to record the results of each step.

1) Slowly rotate the motor to determine its resolution per revolution. This is the key to setting the correct amount of following error each gain term will be required to handle. Record this value in the KNOWN list of items to be determined.

Known 1: Counts per motor revolution

2) Determine the systems maximum velocity. The system maximum velocity should never exceed 90% of the motor top RPM. If the system must reach the top RPM of the motor purchased, then purchase a motor with a higher RPM! The system should never try to CONTROL a motor in excess of 90% of its top RPM!

Known 2: System Maximum RPM

3) Determine the system overall time constant. The system time constant is determined by simply

dividing the best system acceleration time from zero to top RPM by five. This means we must BUMP the motor under load, and measure the time it takes to achieve top RPM (item 2 above). The system time constant takes into account the electronics, the mechanics, and the available power of the motor/amplifier package. In many cases it is not desired to bump the motor allowing it to reach top RPM. In those instances, the motor can be bumped to 25% of the top speed, or until the acceleration ramp stops increasing, or levels off as a straight-line function. The time to top speed can then be projected.

Known 3: Time to Top Speed (Tt) in seconds

Known 4: System time constant = Tt / 5 seconds

4) Determine the proper acceleration time of the systems motion controller. This value is either 2.5 times the resulting value found in step 3 above, or, a slower rate (longer time) based on the system/load handling requirement given in the operating specification. If this value of acceleration is not 'fast' enough, purchase a motor/amplifier with more power, bandwidth, or response capability.

Known 5: System proper acceleration time (Ta) in seconds

5) Determine the Mode of Motor Operation. There are two basic modes of motor operation, Current and Voltage. Other terms used as an alias for these are torque, speed, velocity, etc. However, the actual mode (current or voltage) is determined simply by knowing what type of motor amplifier has been purchased, or what mode the motor amplifier is set to. The mode of motor operation must be known so that the gain values (Kp, Ki, and Kd) can be properly set.

Voltage mode operation will allow the motor to draw the current it requires based on the motor formula ($I = (E_t - E_g) / R$) without intervention by the motion controller. Thus, when using this mode, the motion controller bandwidth will have to be set lower than the motor amplifier bandwidth. However, when in the Current mode the motion controller is directly in control of motor current. Thus, the bandwidth of a motion controller operating a Current mode motor can be several times higher than would be possible with Voltage mode operation.

Known 6: Mode of motor operation (E or I)

6) Determine the following error requirement for the tuning of the proportional PID term(s).

Current mode = 3 degrees / 360 degrees * Counts_per_motor_revolution

Voltage mode = 90 degrees / 360 degrees * Counts_per_motor_revolution

Note the following error difference. If the motor resolution was 2000 counts per motor revolution, the actual Current mode motor tracking would be $3/360 * 2000$ or 17 counts behind the generated trajectory path (desired position). If however, the motor was operating in the Voltage mode, then the actual motor position would be tracking $90/360 * 2000$ or 500 counts behind the generated trajectory path. This would be the following error that will be used when tuning the PID proportional terms.

Known 7: Kp following error requirement

(Current mode=17 counts.... Voltage mode=500 counts)

Once the proportional terms are set, the long term PID controls (Ki) will be set using the following guidelines. For the Current mode of operation, the following error value is zero, and for the Voltage mode, it is 80 degrees or 445 counts.

Known 8: Long term PID following error requirement for Ki

(Current mode=0 counts Voltage mode=445 counts)

Note: The reason for referring to the PID terms as proportional, or long term rather than Kp, or Ki, etc., is that a gain control algorithm may contain MORE than one proportional term, etc.

7) Tune the Kp PID term. Set the motion controller acceleration to the known best value known item 4), and the speed to the maximum RPM (known item 2). Set the move profile for a trapezoidal move (a flat topped profile).

Perform the profile move, and adjust the Kp term until the required (Kp) following error is achieved (known item 6) during the flat top section of the motion profile. If the motor begins to oscillate prior to achieving this condition, increase the Kd PID value to stabilize (or eliminate) this condition.

8) Tune the Ki and IL PID terms. Set the Ki value to the Kp value, and the IL value to 4 times the Kp value, as determined in step 7 above. Set the motion controller acceleration to the known best value (known item 4), and the speed to the maximum RPM (known item 2). Set the move profile for a trapezoidal move (a flat topped profile). Perform the profile move, and adjust the IL term until the required Ki following error is achieved (known item 7) during the flat top section of the move. If the motor begins to oscillate prior to achieving this condition, increase the Kd parameter value to stabilize (eliminate) this condition.

The object of the Integral term is to have the system track the desired profile as closely as possible. Theoretically, the Integral portion of the DAC output voltage should build during the acceleration slope of the profile, and should be at the proper value once the flat line portion of the profile (or stop) has been achieved (taking over DAC control completely).

The value of Ki determines the response of the Integral portion of control and needs to be optimized in order to ensure that it is stable upon achieving stable run operation (no excessive ringing at the knee of the accel/run or decel/stop positions). The position to determine if the Ki is stable will be at the knee plus one system time constant (known item 4). The position to determine if the IL is set properly is at the point of generated trajectory COMPLETE plus one system time constant (known item 3). If the actual stopping position is short of the desired position, increase the IL. On the other hand, if the actual position is beyond the desired stopping point, reduce the IL.

9) Tuning the Derivative Gain term 'Kd. The Derivative gain is used to maintain velocity stability by noting changes in the following error over time and then trying to stabilize further velocity excursions. The derivative gain can be considered a form of acceleration feed forward gain control. Adjusted properly, it can help maintain motion stability relative to the commanded velocity within a relatively small percentage of deviation.

Set the motion controller acceleration to the known best value (known item 6), and the speed to the maximum RPM (known item 2). Set the move profile for a trapezoidal move (a flat topped profile). The derivative term should be checked for proper operation by noting the following error half way up the acceleration slope of the desired trapezoidal profile. It is at this position that the system position should have completely recovered from any starting delay caused by the system time constant (known item 4).

Discussion

Tuning by itself is not the problem. Understanding the tuning algorithm, the interaction of the tuning elements, the order of element tuning, and where in the trajectory profile the test for velocity stability for each of the PID elements is!

Once an element has been tuned leave it alone. By readjusting it, you simply offset the dynamic response of the other elements causing them to over-react in order to compensate for improper settings of others. The result, a possible increase in stability for the set of existing conditions, and a less optimal tune for a wider dynamic range of inertial loading capability.

The Return of Science to Tuning is at Hand!