This article is the August, 2000 installment 20-minute Tune-up which is published monthly in PT Design

PID position loops
August, 2000

This column is the second a three-part series on position loops. This month we will discuss the PID position loop. Next month we will take up the use of feed-forward in these two loops.

Theory

The velocity loop is the most basic servo control loop. However, since a velocity loop cannot ensure that the machine stays in position over long periods of time, most applications require position control. There are two common configurations used for position control: the cascaded position-velocity loop, as discussed last month, and the PID position controller, as shown below.

![Block diagram of PID position loop](image)

<Note to editor: can you replace my “1/s” in the block diagram with an integral symbol followed by “dt?”>

The position loop compares a position command to a position feedback signal, and calculates the position error, $PE$. In a PID controller, current command is generated with three gains: $PE$ is scaled by the proportional gain ($K_{PP}$), the integral of $PE$ is scaled by the integral gain ($K_{PI}$), and the derivative of $PE$ is scaled by the derivative gain ($K_{PD}$).

Tuning in Zones

Tuning is the process of setting control loop gains to achieve optimal performance. Higher gains improve responsiveness, but move the system closer to instability. Tuning PID position loops can be challenging because there are three servo gains: $K_{PP}$, $K_{PI}$, and $K_{PD}$. As with the cascaded position-velocity loop, each of the gains...
plays a different role in the servo system. Once you understand those roles, you can tune
the gains independently, saving time and ensuring consistency.

Each of the gains operates in one of three frequency “zones.” The highest
frequency zone is covered by the derivative gain ($K_{PD}$); typically, this zone ranges from
about 30 to 100 Hz, although it can be much higher. The proportional gain ($K_{PP}$) is most
important for frequencies between about 10 and 30 Hz. The position loop gain covers all
frequencies below that.

**Zone 1: Derivative gain**

Begin tuning the highest frequency zone. Start by eliminating the lower zones:
zero $K_{PI}$ and, if possible, $K_{PP}$. Many PID controllers do not allow $K_{PP}$ to be zeroed. If
that’s your case, fix $K_{PP}$ at a fairly low value while tuning $K_{PD}$. Now, prepare a
trapezoidal point-to-point move. These commands have three segments: acceleration,
cruise, and deceleration. When tuning Zone 1, you should set the acceleration and
deceleration rates as high as the controller will allow. In fact, a square velocity command
(unlimited acceleration) is ideal.

If the commanded move puts the current controller in saturation (that is,
commands more than the controller can produce), reduce the peak velocity of the move.
Usually a peak velocity of 0–250 RPM works well. Now, raise the derivative gain as
high as possible without generating overshoot in the velocity response. The three figures
below show the servo system response to a square-wave command when $K_{PD}$ is a) too
low, b) about right, and c) too high. Note that if you cannot zero the proportional gain
($K_{PP}$), expect some overshoot to a square wave. Overshoot due to $K_{PD}$ occurs on a much
shorter time scale and so is easily distinguished from overshoot caused by $K_{PP}$.

![Graphs showing servo system response to square-wave command](image)

*Zone 1: Velocity step command (black) and response (blue) with $K_{PP}$ and $K_{PI} = 0$, and
$K_{PD}$ a) low (0.6), b) about right (1.1) and c) high (2.0). Horizontal scale is 5mSec/div and
vertical scale is 50 RPM/div.*

Zone 1 is the hardest zone to tune. This is because two common problems seen in
servo systems, resonance (20-minute tune-up, October, 1999) and audible noise (20-
minute tune-up, July, 1999), are excited by this gain. You may need to use low-pass
filters to reduce these two problems. While low-pass filters are helpful, you should
always minimize the their use because they cause instability and force lower value servo gains.

Zone 2: Proportional gain

Now that the derivative gain is set, it’s time to tune the proportional gain. First, modify the position command. Lower the acceleration and deceleration to the highest rates that will be seen in the application under normal operation. Raise the integral gain until a slight amount of overshoot appears, and then reduce the gain to remove the overshoot. The responses for two proportional gain values are shown below with $K_{PP}$: a) about right and b) too high.

Zone 2: Trapezoidal position profile with $K_{PD} = 1.1$ and $K_{PP}$ a) about right (20.0) and b) a little high (40.0). Horizontal scale is 5mSec/div and vertical scale is 50 RPM/div.

Zone 3: Integral gain

The final zone to tune is the position-loop gain. Tuning integral gain is difficult because even a small amount causes overshoot. Several methods have been developed to deal with this shortcoming. First, most controllers allow you to clamp the maximum current the integral term can command. This makes sense because the primary reason to use integral gain in many applications is to overcome frictional loads; when that is the case, there is no need to allow current generated by the integral term to be much larger than the maximum friction load. Another technique is to force the integral to zero anytime the motor is commanded to move. Both of these techniques allow the integral gain to be raised to higher values than it otherwise could be. The figures below show the integral gain set to a level just below causing overshoot ($K_{PI} = 5$) and to value so high ($K_{PI} = 10$) that it causes oscillations at zero speed. Both figures depict a system with the integral zeroed when motion is commanded, and a clamp on the integral output of 2 amps.
**Zone 3:** Velocity trapezoid response with $K_{VP} = 1.1$, $K_{VI} = 2.0$, and $K_P$ a) about right (5.0) and b) high (10.0). Horizontal scale is 5mSec/div and vertical scale is 50 RPM/div.

**Laboratory**

Want to tune a PID position loop yourself? Then log onto www.motionsystemdesign.com and download this month’s ModelQ simulation program. Launch the program, select August’s model, and click “Run.” You should be looking at the square-wave response with just the derivative gain ($K_{PD}$). The value is set a little low (0.6), so raise it until the square wave response just overshoots, and then reduce the gain to eliminate the overshoot ($K_{PD} = 1.1$).

Now select a trapezoidal velocity command by choosing “Trapezoid” in the waveform generator at bottom left. Now raise $K_{PP}$, adjusting to the highest value that does not cause overshoot ($K_{PP} = 10$). Finally, raise $K_{PI}$ to 5 to get a complete set of PID tuning gains.

*George Ellis is a Senior Scientist at Kollmorgen.* [note to editor: we use “Kollmorgen” rather than “Kollmorgen, Inc.”]. *His book, “Control System Design Guide,” 2nd edition, was published in May, 2000 by Academic Press. He can be reached at gellis@kollmorgen.com*