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

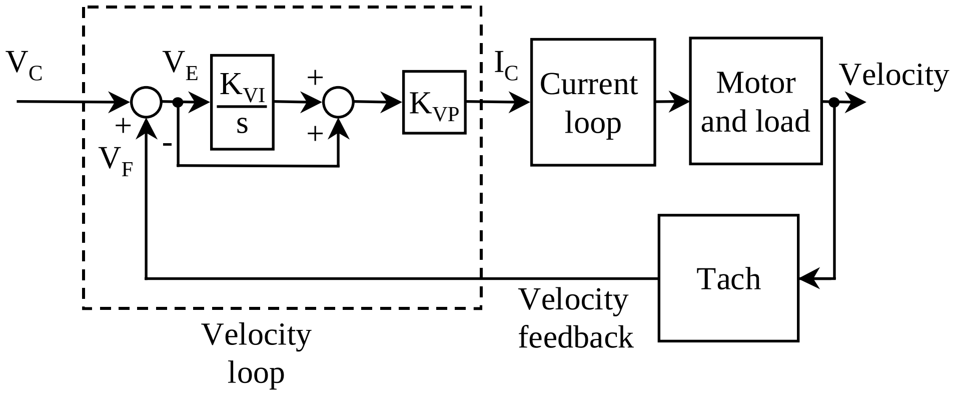
Cascaded position-velocity loops

July, 2000

This column begins a three-part series on position loops. This month we will cover the cascaded position-velocity loop. The next two installments will discuss the PID position loop and how feed-forward is used in these two loops.

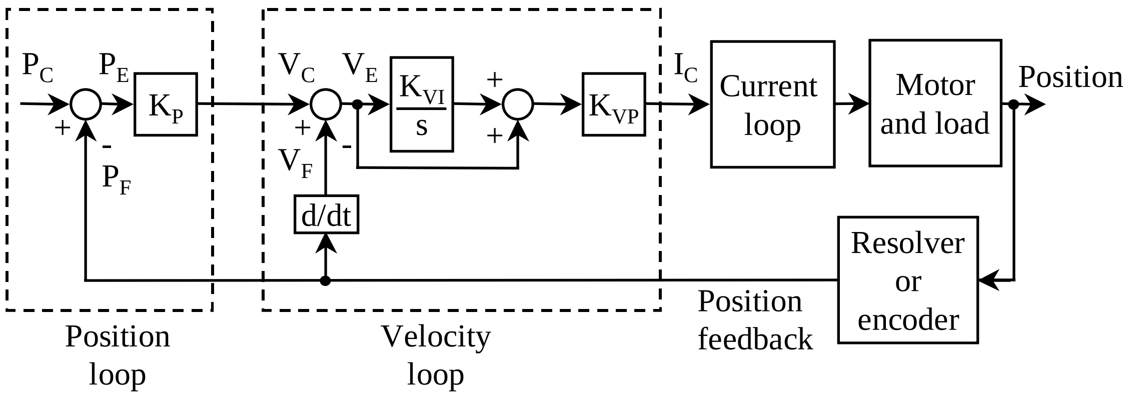
Theory

The velocity loop is the most basic servo control loop. In its simplest implementation, an analog command is compared to an analog tachometer output to generate current which produces torque which, in turn, will speed or slow the motor to satisfy the velocity command. The most common velocity loop is the proportional-integral or PI velocity loop. This loop has two gains, a proportional gain (K_{VP}) which reacts to the velocity error and an integral gain (K_{VI}) which reacts to the integral of velocity error. (Note that 1/s is the conventional depiction of an integration).



Block diagram of velocity loop.

A velocity loop fulfills many of the needs of a servo loop. It reacts to rapidly changing commands and it provides resistance to high-frequency load disturbances. However, a velocity loop cannot ensure that the machine stays in position over long periods of time. Since most machines require position control, the velocity loop must be augmented. One of the most common configurations of servo loops is to place a proportional position loop in series or *cascade* with a PI velocity controller. This configuration is shown below.



Block diagram of cascaded position-velocity loops

The position loop compares a position command to a position feedback signal, and calculates the position error or *following error*. Position error is scaled by a constant (K_P) to generate a velocity command. The principle here is simple: more position error generates a larger velocity command. A larger velocity command, fed to the velocity loop, creates more torque which, in turn, moves the motor to satisfy the position error.

Years ago the position and velocity loops had separate sensors. An encoder or resolver provided feedback for the position loop; an analog tachometer provided velocity feedback. Today, most servo systems rely solely on the position feedback sensor. Now the velocity loop derives velocity from sensed position; this is done by differentiation (velocity is the derivative of position) and is shown as *d/dt* in the block diagram.

Tuning in Zones

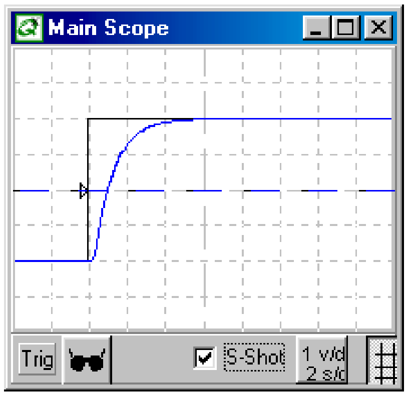
Tuning is the process of setting control loop gains to achieve optimal performance. Higher gains improve responsiveness, but move the system closer to instability. Normally, the goal of tuning is to raise the gains as high as possible without causing instability.

Tuning cascaded position-velocity loops can be challenging because there are three servo gains: the position loop gain (K_P), and the velocity integral (K_{VI}) and proportional (K_{VP}) gains. Using a "hit-or-miss" method of tuning with three gains can absorb a lot of time. Fortunately, 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.

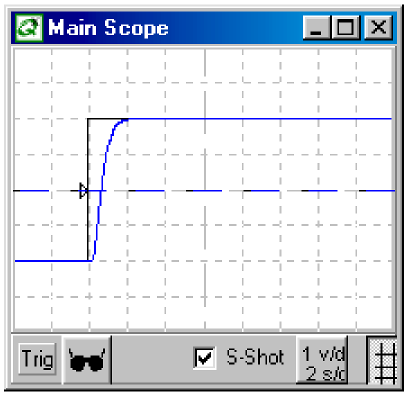
Each of the gains operates in one of three frequency "zones." The highest frequency zone is covered by the velocity-loop proportional gain; typically, this zone ranges from about 30 to 100 Hz, although it can be much higher. The velocity-loop integral gain is most important for frequencies between about 10 and 30 Hz. The position loop gain covers everything below that.

Zone 1: Velocity-loop proportional gain

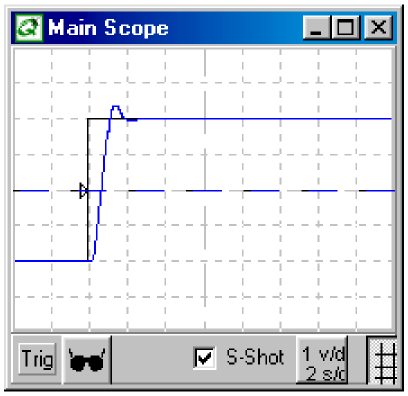
Tuning begins at the highest zone. Start by eliminating the lower zones: set K_{VI} to zero and put the system in velocity mode to eliminate the position-loop gain. Inject a square-wave velocity command; adjust the command magnitude to be as large as possible without saturating the current controller; you should ensure that the current controller remains below its maximum at all times. Usually a command magnitude of 0–250 RPM works well. Now, raise the velocity-loop proportional 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_{VP} is too a) low, b) about right, and c) too high.



a)



b)



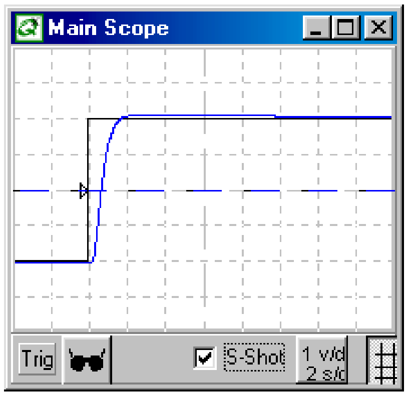
c)

Zone 1: Velocity step response with K_{VI} = 0 and K_{VP} 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.

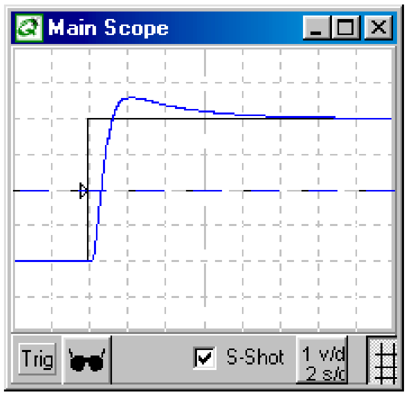
Zone 1 is the hardest zone to tune. This is because two common problems seen in servo systems, resonance (20-minute tune-up, October, 1999) and audible noise (20-minute tune-up, July, 1999), are excited by this gain. You may need to use low-pass filters to reduce these two problems. While low-pass filters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

Zone 2: Velocity-loop integral gain

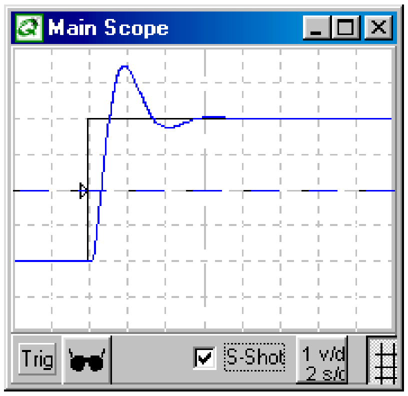
Now that the velocity-loop proportional gain is set, it's time to tune the velocity-loop integral gain. This gain is easily tuned. Leaving the command signal as it was, raise the integral gain from zero until there is about 15% overshoot. The responses for three integral gain values are shown below.



a)



b)

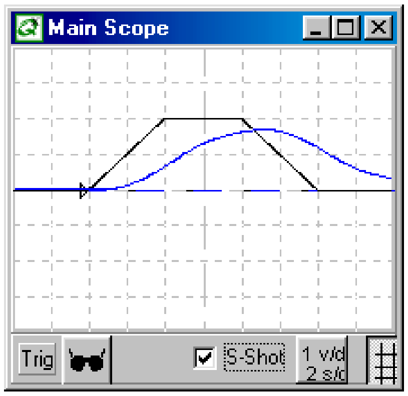


c)

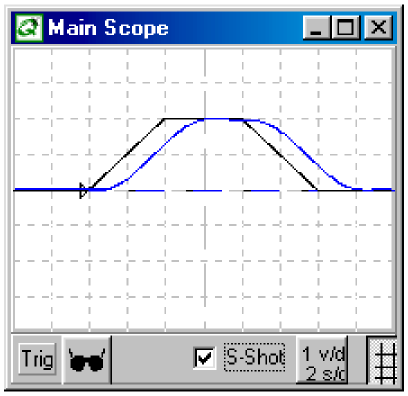
Zone 2: Velocity step response with K_{VP} = 1.1 and K_{VI} a) low (20), b) about right (100) and c) high (420). Horizontal scale is 5mSec/div and vertical scale is 50 RPM/div.

Zone 3: Position-loop gain

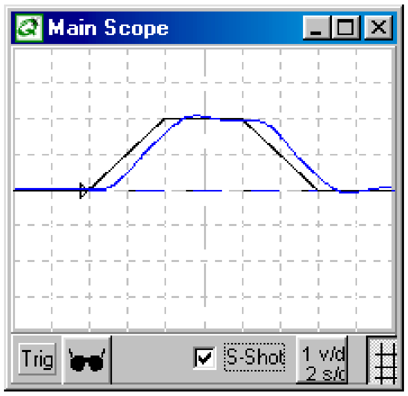
The final zone to tune is the position-loop gain. To complete this step, you must set the controller to operate in position mode. Most controllers provide a variable that will let you make this change. Now set the command to produce a point-to-point move with the acceleration rate set at the maximum that the servo loop will see in normal operation. Finally adjust the position-loop gain up until just before the response starts to overshoot. The figures below depict the response to a rapid trapezoidal command; note that since most applications require zero overshoot in normal operation, just the little overshoot apparent in “c)” is enough to eliminate this high tuning gain value (600).



a)



b)



c)

Zone 3: Velocity trapezoid response with K_{VP} = 1.1, K_{VI} = 100, and K_P a) low (100), b) medium (250) and c) high (600). Horizontal scale is 5mSec/div and vertical scale is 50 RPM/div.

Laboratory

Want to tune a cascaded position-velocity loop yourself? Then log onto www.motionsystemdesign.com and download this month’s ModelQ simulation program. Launch the program, select July’s model, and click “Run.” You should be looking at the square wave response with just the velocity-loop proportional gain. 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_{VP} = 1.1). Now raise K_{VI}, adjusting for about 15% overshoot (K_{VI} = 100). This completes the tuning of the velocity loop.

To tune the position loop, you must configure this model as a position controller. First, set the gain K_{VF} to 0% (it was set to 100% to simulate a velocity controller). Now select a trapezoidal velocity command by choosing “Trapezoid” in the waveform generator at bottom left. Finally, raise K_P to just below where the system overshoots (K_P = 250).