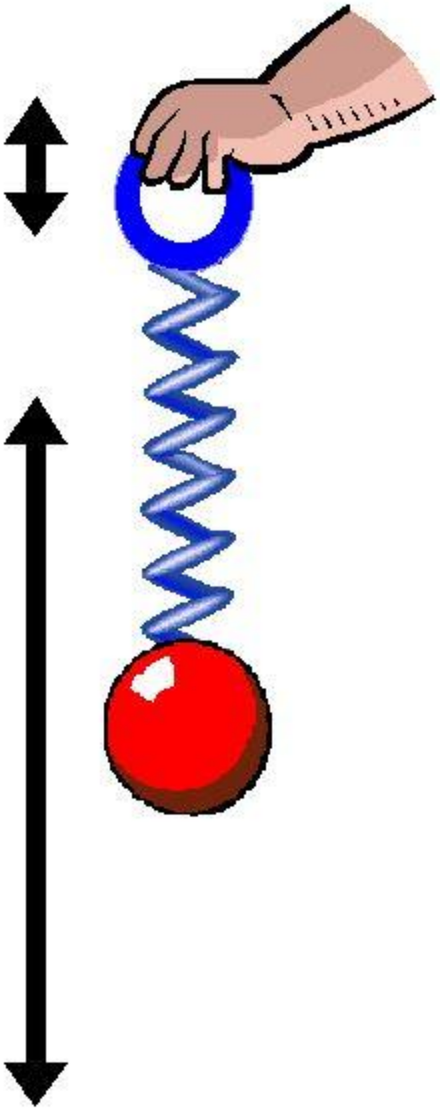
Back to Basics: Closed-loop stability

Tutorial: Stability is how a control loop reduces errors between the measured process variable and its desired value or setpoint.

Vance VanDoren, Ph.D., P.E., Control Engineering

08/17/2010



For the purposes of feedback control, stability refers to a control loop's ability to reduce errors between the measured process variable and its desired value or setpoint. A stable control loop will manipulate the process so as to bring the process variable closer to the setpoint, whereas an unstable control loop will maintain or even widen the gap between them.

With the exception of explosive devices that depend on self-sustained reactions to increase the temperature and pressure of a process exponentially, feedback loops are generally designed to be stable so that the process variable will eventually achieve a constant steady state after a setpoint change or a disturbance to the process.

Unfortunately, some control loops don’t turn out that way. The problem is often a matter of inertia – a process’s tendency to continue moving in the same direction after the controller has tried to reverse course.

Consider, for example, the child’s toy shown in the first figure. It consists of a weight hanging from a vertical spring that the human controller can raise or lower by tugging on the spring’s handle. If the controller’s goal is to position the weight at a specified height above the floor, it would be a simple matter to slowly raise the handle until the height measurement matches the desired setpoint.

Doing so would certainly achieve the desired objective, but if this were an industrial positioning system, the inordinate amount of time required to move the weight slowly to its final height would degrade the performance of any process that depends on the weight's position. The longer the weight remains above or below the setpoint, the poorer the performance.

Moving the weight faster would address the time-out-of-position problem, but moving it too quickly could make matters worse. The weight’s inertia might cause it to move past the setpoint even after the controller has observed the impending overshoot and begun pushing in the opposite direction. And if the controller’s attempt to reverse course is also too aggressive, the weight will overshoot the other way.

Fortunately, each successive overshoot will typically be smaller than the last so that the weight will eventually reach the desired height after bouncing around a bit. But as anyone who has ever played with such a toy knows, the faster the controller moves the handle, the longer those oscillations will be sustained. And at one particular speed corresponding to the resonant frequency of the weight-and-spring process, each successive overshoot will have the same magnitude as its predecessor and the oscillations will continue until the controller gives up.

But if the controller were to become even more aggressive, those oscillations would grow in magnitude until the spring reaches its maximum distention or breaks. Such an unstable control loop might be amusing for a child playing with a toy spring, but it would be disastrous for a commercial positioning system or any other application of closed-loop feedback.

One solution to this problem would be to limit the controller’s aggressiveness by equipping it with a speed-sensitive damper such as a dashpot or a shock absorber as shown in the second figure. Such a device would resist the controller’s movements more and more as the controller tries to move faster and faster. The derivative term in a PID controller serves the same function, though too much derivative damping can actually make matters worse.

See “Understanding Derivative in PID Control,” *Control Engineering*, February 2010.

See Tutorials Channel at www.controleng.com/tutorials.

Vance VanDoren, Ph.D., P.E., is *Control Engineering* consulting editor,
at controleng(at)msn.com. www.controleng.com

Source: http://www.controleng.com/single-article/back-to-basics-closed-loop-stability/846231f5c6d4e3fb0ff152f99f21c018.html