General information on Data Acquisition and Control has become harder to find. The field has split between expensive, proprietary products and hobby grade projects. The purpose here is to work toward regaining the middle ground.

Simple, Affordable Computer-in-the-loop Data Acquisition and Control

Background

Why can't you just use your computer for loop control? Precision loop control requires three basic components before you can get productively started.

1) You need high resolution monotonic inputs and outputs if your loop is to both handle a wide range of conditions and settle accurately to a specific value.

Let's break that down. Why monotonic? Non-monotonic systems will misbehave, sometimes catastrophically, when the feedback reverses polarity.
Why high resolution? Without enough resolution, you need to limit the range, which leads to gain switching, which tends toward non-monotonicity at the transitions.

2) You need an accurate timebase with low latency.

To accurately represent changing voltages, you need to know both the voltage and the time. Timing errors will disrupt a loop just as much as voltage errors.

3) You need an easy way to set up the control rules and tune them.

It makes a huge difference if the rules can be intuitive and easy to adjust. Once you find yourself tinkering in the dark, your chances of optimizing a control system in a finite amount of time drop drastically.

Plus, for reliability and safety, you need electrical isolation and proper grounding to keep digital, or high current outputs from interfering with analog inputs. If you have stray feedback paths from that sort of interference, loop control will always be problematic.

With the above capabilities in place, you can, in fact, use your computer for practical loop control. You need to invest a little for quality hardware, but the software can be free.

Alternative to PID control

There is no one definitive guide to PID (Proportional Integral Differential) tuning. Instead, there is a ongoing mix of conventional advice and new recommendations. That pattern is strong evidence that PID is not perfectable. PID is neccessarily a compromise between speed and stability. Here, we propose alternatives to the normal control topology that can improve performance and simplify the tuning of any loop, particularly a loop that suffers from delayed response.

Our approach:

Differential equations are a valid and necessary way to understand control loops, but they are of little benefit to the technician doing the loop tuning. Here, we will build an intuitive case that stays as close as possible to the practical matter at hand.

If the feedback is immediate, loop tuning is easier. The hard part is dealing with delays. The longer the delay, the more challenging the control task.

For cruise control on a passenger car, delays are short. If you take your foot off the gas, the car will begin to slow immediately. That is an example of a loop with relatively little delay. Cruise control for a ship is harder. You can entirely shut off the propulsion and the ship will long continue to travel at almost the same speed. Or, you can speed up the engines and the ship will continue at much the same speed. The intuitive takeaway from that situation is the present speed is more dependent on the past speed than on the current output of the engines. Heater control with long delays involves the same sorts of challenges. If you shut down the heater completely, the process temperature might even continue to rise for some period of time thereafter.

Take an even simpler example of control with delay. A dial controls a filtered voltage, such that it takes a long for the voltage to settle. An operator adjusts the dial so as to obtain a particular voltmeter reading. If the voltage follows the dial promptly, there is no difficulty. If it takes the voltage many seconds to settle to a new value, the operator will overshoot an attempted correction, and then undershoot on the second correction, causing the voltage to oscillate around the desired value. The operator will learn to make smaller adjustments and wait longer between them in order to damp the oscillation, but adjustment then becomes a slower and more tedious process.

The generalized dial analogy holds for any control loop with a large delay in the feedback. The voltmeter reading is here the process feedback variable, which in turn, is the input to the control loop. You may be used to thinking of the voltage as the output of a control loop, not the input, but in this case, the dial setting is the control output and the voltage reading is the process feedback.

Lets say the system is stable at the desired voltage and now you need to adjust for a different voltage. Even with the extra care learned from the original adjustment, unless you take extra time making small adjustments and waiting in between, you will again find the voltage alternating above and below the new set point.

How would an master dial operator get the voltage stabilized at the new set point as quickly as possible? Through earlier experimentation, the operator could have found the new dial setting and gone straight there. That would cause the voltage to exponentially approach the new setting, being a real improvement, but the long tail on the settling curve means it would still take a long while for the voltage to reachy the new setpoint value.

A masterful dial operator could reach stability at the new setpoint faster by setting the dial higher than the new setting for a period of time, then setting it below the new setting for a shorter period before eventually arriving at the new dial setting. Thereafter, only small continuing corrections would be needed.

Advance knowledge of the required dial setting is considered feed forward, as opposed to feedback. That advance knowledge is helpful, and does make the control task easier, but advance knowledge is not always available. Here, we will limit ourselves to the more challenging case where such advance knowledge is absent. How can we reduce the actions of that master dial operator to a set of quantifiable rules?

Nuts and Bolts

The starting point for any control decision is the voltage difference between the setpoint and the present voltage. That is the familiar error term used in any feedback system. The dial operator knows that the bigger the error, the larger the correction to apply. This term is conventional proportional control. Proportional control is always part of the solution. It works, but if it is applied too aggressively, it is bound to cause the oscillation around the setpoint described above whenever there is delay in the feedback.

To stop oscillation, anticipation is needed. That takes the form of an indication of rate of change of the feedback. This term is the differential term. An output can change faster or more slowly, but cannot instantaneously step up or down. The dial operator needs to anticipate future voltage based on rate of change and loop delay. The delay in a system is not always constant, but it tends to be mostly baked in, so a general idea of how much delay to expect is good enough in order to anticipate effectively.

Predictive controls respond to the anticipated error at a future time. The rate of change of the output multiplied times the system delay yields a predictive guess of how much the voltage will have changed by a future time. Add that change to the instantaneous output to obtain the expected eventual output, (in the absence of additional influences,) at some future time. Predictive controls apply the proportional gain correction to the anticipated error at that future time, instead of to the instantaneous error.

The limiting factor for success in a predictive system is noise. Differential terms are very sensitive to noise, while integral terms are filtered, by definition, and are therefore more stable. By taking the differential of the integral we can capture the differential information without introducing too much destabilizing noise.

Stated another way, a little noise superimposed on an otherwise stable voltage will cause the predicted future voltage to be alternately higher and lower than the stable value, causing a tendency to overreact to even small amounts of noise. So, we first do a running average, and obtain the average rate of change from those averaged values. Better predictions with less sensitivity to noise is the result. With that improved prediction, the control can act faster than controls that require the integral term to dominate.

Further refinements

The loop, once adjusted, will stabilize whenever the system is in a static condition. It may not, however, stabilize at exactly the setpoint because the gain is necessarily limited. That means a secondary correction is needed. It can be done with a filtered term that adjusts the setpoint to obtain the desired output based on a longer-term average. We call this type of term the bias. In a computer-in-the loop system, the bias can be self-learned. That is, the system tends toward the output that has worked to bring about regulation in the recent past.

Finally, it is useful to know how much correction has been applied that has not yet shown up at the voltage output. That term is the change of the applied correction over a period of time. The exact period will depend on how much delay is built into the system. Adding a small amount of this term, because it gives advance notice of future changes at the output, can stop oscillation before it gets started.

Now we have all the pieces.

The control can be done in hardware or software. For fast controls, hardware is better. For heaters, or slower control loops needing only a few updates per second or less, computer-in-the-loop controls are just fine.

The dial and delayed voltage analogy represents any system with built-in delay. To the extent that the delay is constant, the loop can be tuned to near-ideal response. The prediction is tuned by looking ahead nearer or further into the future. Too much look-ahead causes undershoot, and too little causes overshoot. You will need to do a little trial-and-error tuning to find the right value. The predictive factor and the proportional gain factor will intact somewhat, but they are independent enough to make the tuning process straightforward. More proportional gain will correct more aggressively. At some point, too much proportional gain will introduce a tendency toward oscillation. Any amount of gain up to the edge of oscillation is workable.

Note that there are two different forms of disturbance. A setpoint change is an instantaneous disturbance, while a process change is more incremental. A tuned control will minimize undershoot or overshoot and speed settling time in both instances.

Specifics of tuning using the Lawson Labs general purpose template

Tuning summary:

Pick an avg count for the running average, make it larger for slower, noisier processes.
Turn up the gain until oscillation starts, or turn it down until oscillation almost stops
Tune the prediction to stop the oscillation.
Increase the gain, tune prediction
Continue until the highest proportional gain consistent with good stability is found
Add a touch of the D term for extra stability margin.

Why does it work?

The underlying problem with loops with long delays is oscillation. Feedback easily gets out of phase, since time constants can change with conditions. You can go to great lengths to try to match the response to the delay, and some of that is essential, but the match will never be perfect, so the amount of gain the system will tolerate will be less than the ideal gain, because of the tendency toward oscillation.

When we settle for reduced gain to optimize the dynamic response, the DC error gets bigger, so you often need an integral term to improve the DC performance. If the integral term has a short time constant, it will act much like more proportional gain, again causing oscillation. With too long a time constant for the integral term, precise response will be sluggish after a change in conditions.

So, instead of an integral, we use a self-learned offset to correct the DC response. With computer-in-the-loop control, a self-learned bias that tends to center the response on the correct value is easier and more controllable than trying to fine tune the gain. The self-learning should only applied when the system is stable to prevent an upset from causing the self-learning to temporarily detune the system.

For a comprehensive overview of the controllable factors in the free template, see the comntrol section of the manual, at

http://www.lawsonlabs.com/model30x/ExcelTemplate/General_Purpose_Template_Instructions.pdf

Questions and comments invited at

https://lawsonlabs.com/bboard/

Tom Lawson
September 24, 2020