Criteria and Procedure for Controller Tuning I & II

R. Russell Rhinehart

Part I, Develop Your Potential Series in CONTROL Magazine, January, 2017, Vol. 30, No. 1, pp 54-55.

My perspective on this question is from experience in the process industries; where, characteristically, applications are nonlinear, relatively slow, with a modest level of signal noise, and substantial concerns related to safety.

Usually tuning is tested with set point changes (servo mode) using deviations from the new set point as the measure of goodness. And, usually, goodness of control is based on observing the controlled variable (CV) trend in time. If overshoot and undershoot (for example too hot, too cold) material is blended downstream, then Integral of the Error (IE) could be an appropriate measure of control goodness. "Error" is the controller actuating error, the deviation from set point. In the IE method, "+" and "-" errors balance each other. However, if either deviation is bad, then integral of the absolute value of the error (IAE) could be the right measure. For example, the customer may like a bit of extra purity, but the manufacturer may not like the extra energy it takes to make the higher purity. Here either deviation is undesired. But often, small deviations are inconsequential, and large ones are detectably bad. In this case integral of the squared error (ISE) is appropriate. ITAE (integral of time-scaled absolute error) and ITSE (integral of time-scaled squared error) penalize persisting oscillations (or any sort of deviation) from set point by multiplying the actuating error by the time-since-the-set-point-change. These metrics would seek to improve settling time in servo mode.

All of those measures quantify the impact of deviations from set point. But an older, still popular, criterion is damping rate, with quarter-amplitude damped (QAD) the custom. In QAD, the controller is tuned so that the second overshoot is ¼ of the first. The overshoot is the beyond-shoot. It could be either above or below the set point depending on the direction of the activating set point change. The overshoot is in the same direction as the set point change.

Although we use the term "integral" the actual metric is calculated numerically, from sampled data not continuum values, and over a finite time period not for infinite time. Here is a nominal definition for ISE, and its calculation using the rectangle rule of integration.

$$ISE = \int_{t=0}^{\infty} e^2 \, dt \cong \Delta t \sum_{i=1}^{N} {e_i}^2$$

ISE is my favorite of all of these sort of measures of CV response, because, when normalized by time, it represents the process variance during regulatory periods. Most of the time processes are not being changed, but they are being regulated, kept at the same set point. Process variance, which determines how close the set point can be to product specifications, would be an excellent choice for assessing control goodness. This expansion shows the relation to σ and ISE normalized by collection time, t.

$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - x_{SP})^2 = \frac{1}{N-1}\sum_{i=1}^{N} {e_i}^2 \cong \frac{1}{N}\sum_{i=1}^{N} {e_i}^2 = \frac{\Delta t}{t}\sum_{i=1}^{N} {e_i}^2 = ISE/t$$

If the product specification is $x \leq x_{Spec}$, and the CV standard deviation is σ, and the desire is to only violate the specification 95% of the time, then the set point needs to be

$$x_{SP} = x_{Spec} - 1.96\sigma = x_{Spec} - 1.96\sqrt{ISE/t}$$

But, in this regulatory case, testing goodness of controller tuning would not be in response to set point changes. To test in the regulatory situation, choose the controller coefficient values, collect data in an extended regulatory period, long enough to experience the control response to the range of vagaries that the process might experience.

Although ISE is my default choice, a particular application situation may make one of the other metrics (IE, IAE, ITSE, QAD, …) more appropriate. Understand your process and its context, then choose a right CV metric.

Goodness of control should also consider the impact on the manipulated variable (MV), the controller output. In my opinion, exclusively using the metrics of CV performance to tune a controller often result in a controller that aggressively moves the valves. This undesirably and excessively upsets utility streams, stresses equipment, and propagates upsets downstream. One measure of MV action is Energy, the sum of squared incremental changes in the MV at each sampling. Another is Travel the absolute value of the incremental MV changes since the last adjustment.

$$Travel = \sum |MV_i - MV_{i-1}| = \sum |\Delta MV_i|$$

If the MV needs to change from 50% to 60%, but does it by jumping to 80% then 55% then 75% then settling at 60%, it traveled 95% just to make the 10% change.

I think that CV criteria for tuning often overlook the impact on the MV. The CV tuning criteria lead to aggressive controllers. However, because of safety, propagation of upsets in utility lines, thermal stresses, operator response to oscillating conditions, and robustness to process gain changes; operators usually prefer controllers to be more temperate and move the process in a fast-enough, but in an over-damped manner.

Although CV and MV issues are important, I think that three items related to controller tuning have even a stronger claim in evaluating tuning:

1. Operator/technician/engineer time/skill is the most important criteria. A tuning procedure should be fast, robust, and simple (no complicated calculations).
2. A tuning procedure should not be invasive. It should not upset the process. This is the second most important criteria.
3. Tuning must accommodate what the controller might encounter, not just what the process is at the instant it is being tuned. Processes are nonlinear, which means the gain, time-constants, and delays are not constant, but depend on operating conditions, which continually change. A tuning procedure should permit the tuner to balance CV and MV issues in the situation context over the likely future conditions.

If I wrote a textbook on process control, I would include many of the standard tuning approaches (ZN Ultimate, lambda, controller synthesis, FOPDT model based, Cohen-Coon, ATV, etc.). They have value in revealing analysis of dynamic systems and the historical context of the profession. But, I don’t use them.

They take too long, induce process upsets that are too large for comfort, presume the controller coefficient settings are properly calibrated, require calculations, use the too aggressive CV measures, and presume that the process is linear.

I use and teach a heuristic approach to tuning. It is based on the standard PID algorithm.

$$u = K_c\left(e + \frac{1}{\tau_i}\int e\,dt + \tau_d\frac{de}{dt}\right)$$

1. Adjust the gain in a P-only controller until there are about 3 detectible movement directions in response to a modest set point change. Make sequential set point changes in up-down-down-up pattern to minimize process deviation from the nominal value. Start with a low K_c value to minimize tuning-induced upsets. Double or halve the K_c value if initially searching, then when the range is identified, interval halve to fine tune. There is no need to get exactly 3 wiggles – 2.5 or 3.5 is fine. It may be easier to see the signal wiggles in the MV than in the CV. If there are more wiggles on one side than the other use the more aggressive side.
2. Set the integral time to half of the period of the oscillations.
3. In the rare case that derivative action is desired, accept the norm of the conventional tuning rules, and set the derivative time to ¼ of the integral time.
4. Set options that you would like (such as P-on-x, velocity mode, anti-windup).
5. Fine tune the PI or PID controller gain by testing with set point step changes over the operating range desired. Avoid oscillatory action. Accept sluggish control in one direction or one operating region, if that prevents oscillatory action in another region that might be encountered.

I find the procedure quick, easy, effective, and acceptable. It is my appropriation of the Ziegler-Nichols Ultimate method and ATV (Auto-Tune Variation). Both of those classic approaches induce oscillations in a P-only controlled process, and determine tuning values from the oscillation response. Both are robust; however, in my opinion, those classic approaches lead to overly aggressive controllers, and create undesired (or wholly unacceptable) process oscillations during the data-gathering phase.

This heuristic approach is fast, confidently done, has small impact on the processes, and includes human supervision. I suggest that you rise above any theory you may have been taught about one aspect of tuning, and practice heuristic tuning with a comprehensive set of evaluation criteria.

Part II, Develop Your Potential Series in CONTROL Magazine, September, 2017, Vol. 30, No. 9, pp 46-49.

Often one does not need to retune from the beginning. Usually, we come across a controller that was once tuned; but, because of process changes, it now needs some fine tuning. Then, one can spot an issue as it is revealed by patterns in the MV (manipulated variable, the controller output) and the CV (controlled variable, process response) with the active controller, and fix the tuning by adjusting the right coefficient, without starting from scratch. This article reveals some patterns to observe in undesired oscillations, and how to fine-tune PID controllers for open-loop stable processes.

It is a collection of insight I've received from several experts, and I am very grateful to those who have shared the practical side of process control, especially Harold L. Wade. Before I met Harold, everywhere I turned for help, I found academic research artifices and obscurities of Laplace-z-Fourier transforms and other delightful mathematical diversions. Such fundamentals are valuable; but, it is the simple things that are implementable, and which solve the real problems. Harold's insight made the concepts practicable and enabling. Consider acquiring his book, Basic and Advanced Regulatory Control: System Design and Applications, Third Edition, published by ISA. Hopefully, I can pass through such insight along to the reader.

There are three functions in a PID controller. You need to understand their role to fine-tune controllers. Proportional, P, takes immediate action in response to the actuating error, the controlled variable (CV) deviation from the set point (SP). But, P cannot totally remove error. This persisting error is called steady-state offset. Integral, I, incrementally biases the controller output (the Manipulated variable, MV), to seek to remove SS offset. Although integral is essential to remove SS offset, do not think it is the primary action. I-action must be kept slow, because it remembers and tries to correct for long past events. Think of I-action as a helper to P; only there to remove SS offset. P&I join forces to push the CV toward the SP. Derivative, D, puts the breaks on. D overrides the P&I action when D anticipates that the P&I effort now, seeking to push the process toward the SP, will actually push the CV beyond the SP in the future. D-action looks at the rate of change of the CV, the derivative, to sense what might happen in the future. D is a good thing, but it is misdirected by noise and adds complexity. Mostly, PI is fully adequate.

In the standard controller the derivative time multiplies the rate of change of the CV, the integral time divides the integral (the accumulation, the memory) of past actuating errors, and the controller gain multiplies the sum of the actuating error and the I and D values. Note that the controller gain attenuates all three P, I, & D actions. However, there are many alternate forms for the PID controller. This discussion is based on the standard version. If your controller is based on an alternate version, you'll have to adapt the rules. Check what your controller is using. In calculus the standard version is represented as:

$$u = K_c\left(e + \frac{1}{\tau_i}\int edt + \tau_d \frac{de}{dt}\right)$$

In Laplace notation it is:

$$\hat{u} = \left(1 + \frac{1}{\tau_i s} + \tau_d s\right)\hat{e}$$

Notable, this is not an ISA Standard. The term "standard" is a term of conventional usage.

In debugging controllers, first simplify: Remove any D-action.

Most commonly, the undesired aspect is oscillations. Undesirable amplitude or persisting oscillations could be due to any number of situations, and patterns in the MV and CV can reveal where the problem is.

The oscillation might be induced from an up-stream process or disturbance, which is either oscillating or responding to an on-off or fill-empty action. If so, tuning a down-stream controller might temper the impact, but cannot eliminate it. Seek to temper the upstream cause with slower response, or intermediate inventory capacity.

The following examples are from a simulated PID-controlled flowrate process, representing the frequently encountered category of open-loop stable processes. Integrating processes (such as liquid level), processes dominated by deadtime, or inverse acting processes would have similar, but different character and solutions.

Figure 1 shows a PI controller that is too aggressive. The MV is the dotted line and starts at a value of about 0.58 (58%, displayed on the left axis). The SP is the dashed line and makes a change from 0.65 (65%, on the right axis), at a time of 70 (arbitrary units); and the CV (solid line) eventually settles to the set point. The first CV over-shoot (beyond-shoot is a better term because the CV goes on the other side of the SP) is about 4 times greater than the second, which might be confused with quarter-amplitude-damped tuning. However, note that the CV average is not at the new SP until after about 5 oscillations (eyeball an average CV value between any adjacent peaks or valleys in the cycle). Desirably, oscillations disappear about after 1 or 3 cycles; and, once the process changes direction, after a first full oscillation, the CV should be averaging at the set point.

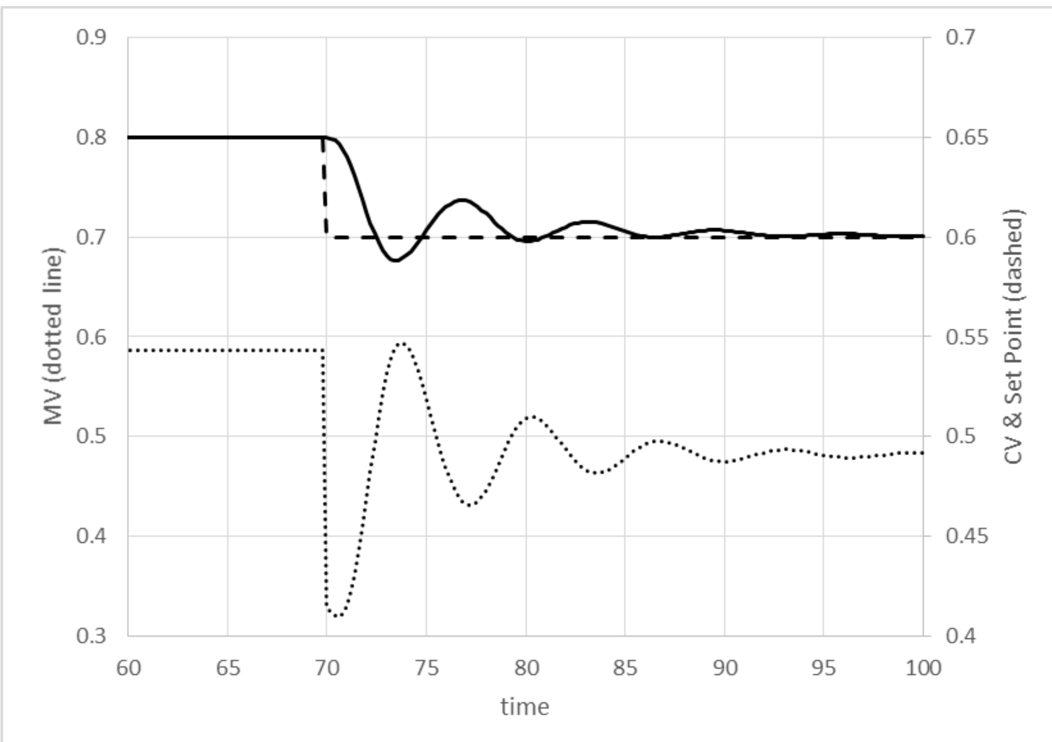

Figure 1 – Too aggressive and cycling in phase

In Figure 1 the controller is too aggressive, because the oscillations persist. And, yet, the integral action is not aggressive enough to remove SS offset within a full CV cycle. The problem is that the controller gain is too high, making the P-action too aggressive. A key clue that P-action is causing the oscillations is that the MV and CV are cycling nearly in phase (or nearly out of phase, depending whether the controller is set for reverse action or direct action). The peak of one signal corresponds in time to the peak (or valley) of the other. The solution is to reduce the gain.

The period of the CV oscillations is an indication of the time-constant for the process, of how quickly the process can be moved to and fro. I-action should be able to remove the offset within about a full CV cycle.

A nominal value for τ_i should be about ¾ of the half-period of the CV. In this case, the CV cycle period is about 6 time units, the half-period is about 3, and a good value for τ_i would be in the 2-3 range. “About” is fully adequate. Perfection is not justified.

For comparison, Wade recommends that τ_i be within the range 1/2 to 2/3 of the cycle period. This reveals personal preference, and gives permission to a reader to use their own feel.

In Figure 2, the controller is, also, undesirably aggressive. Note that the CV and MV are cycling about 90-degrees out of phase. The peak of one is between the valley and peak of the other. This indicates that the I-action is too aggressive. Increase the integral time-constant.

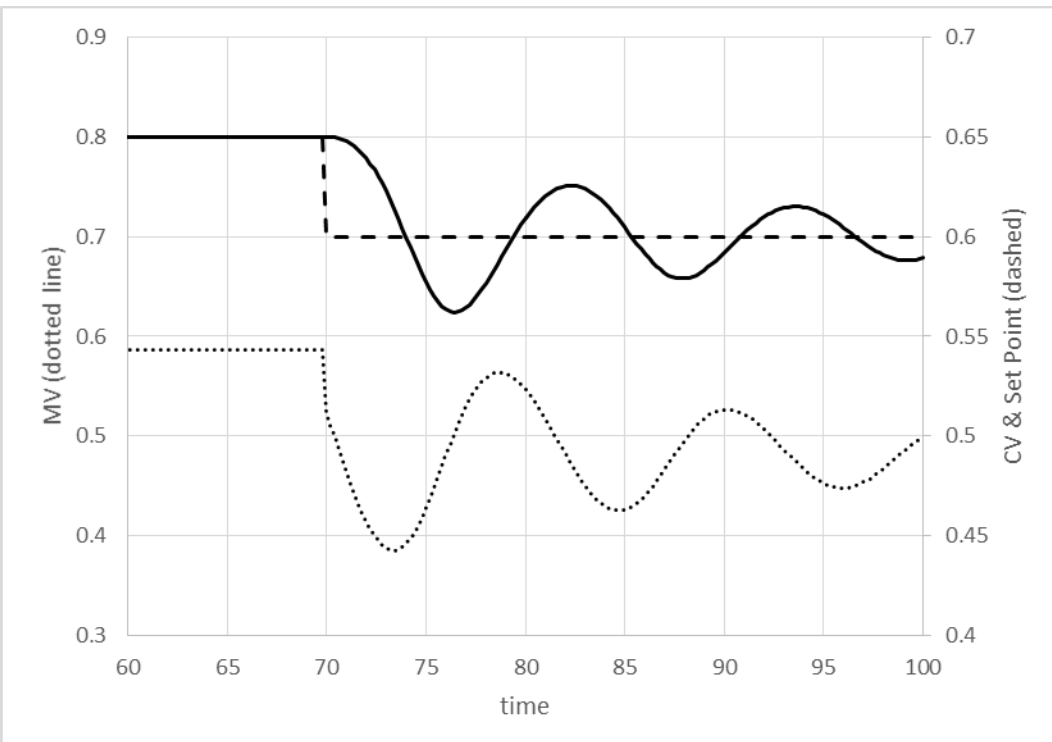

Figure 2 – Too aggressive and cycling out of phase

Whether the CV appears to lead or lag the MV, depends on the controller action (RA or DA). And, however it appears, of course, the CV must follow the MV.

In either too-aggressive case (Figure 1 or Figure 2), adding D-action, anticipatory action, could temper the oscillations. But, a proper solution is not to override bad tuning with D, and complicate the controller. Instead, the solution is to reduce either gain or integral action.

If the MV is oscillating, but the CV is not, as illustrated in Figure 3, this suggests that the derivative time is too large. The MV is the dotted line, and it settles to a value of about 0.5 (50%), and the CV moves toward the SP. Note that the settling time is about 10 units (the CV reaches the SP at a time of about 80, the SP change event was at 70, hence the transition time is 80-70=10); however, the oscillations in the MV have a much smaller period (of about 1.25 units). This means that the controller is changing its mind about how to push the process much faster than the process can change. Supporting this, also note that the MV oscillations are barely detectable in the CV. If this is the case, reduce the derivative time. Nominally, τ_d should be about ¼ of the integral time.

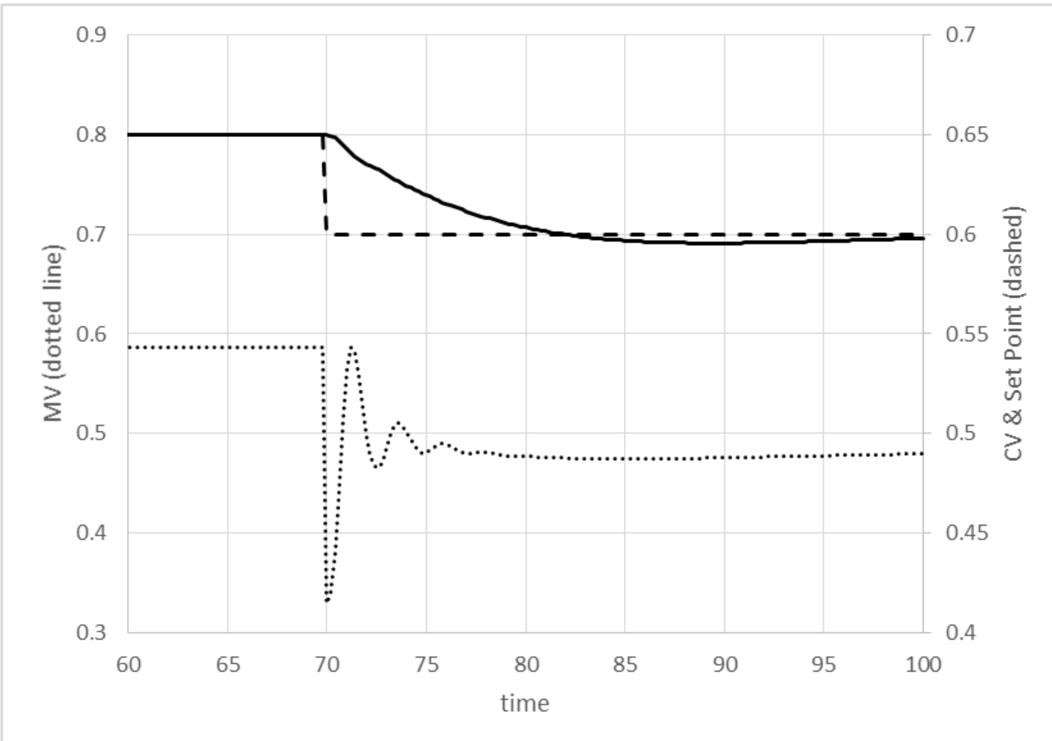

Figure 3 – Too much D action

Figure 4 reveals an alternate sort of persisting oscillation, one due to sticktion. Here, the controller is properly tuned for the process, but the valve temporarily sticks at one position until the pneumatic pressure in the actuator rises enough to make the stem overcome friction and change position. This is similar to deadband in a linkage to a final process influence. Note: The MV is the controller output, not the actual valve position. At a time of about 52, the controller was telling the valve to go to 0.7 (70%). When the valve did, the CV moved to too high a value. Consequently, the controller began reducing the MV. At time 55 the controller said, "Valve go to 0.65." But, even though the i/p device changed the pneumatic pressure on the actuator, the valve stem was held tight by the packing, and the valve did not change position. So, the CV remained too high, and the controller output progressively dropped until at a time of about 65, when it was directing the final control elements to make the valve go to 0.5, the pneumatic forces finally overcame the valve stem friction. Unfortunately, the valve jumped to 0.5, which is too low, and the CV dropped below the SP.

Ideally, sticktion makes a square wave response in the CV and a sawtooth wave in the MV. The abrupt change in the CV marks the reversal in direction of the MV. However, these idealizations may be masked by cycle-to-cycle vagaries and process dynamics, as in Figure 4.

You cannot eliminate such cycling by tuning the controller. If you see sticktion, fix your valve, perhaps add either a pneumatic positioner on the valve or a digital equivalent in the network.

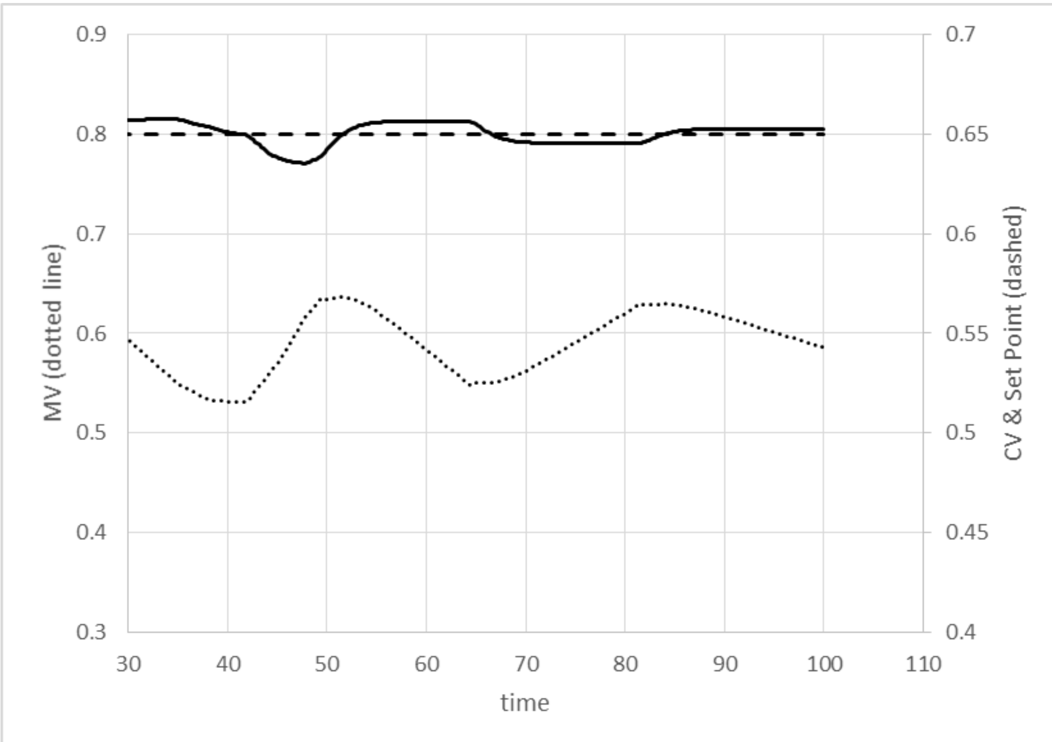

Figure 4 - Stiction

Processes are often nonlinear, and Figure 5 reveals the nonlinear response to two set point changes. The first down from 0.65 to 0.55 has the same CV change as the second, back up to 0.65; however, the first moves to a valve position of about 40% while the second moves to about 60%. The controller is the same. Note that the tuning is relatively aggressive for the first SP change, but relatively sluggish for the second.

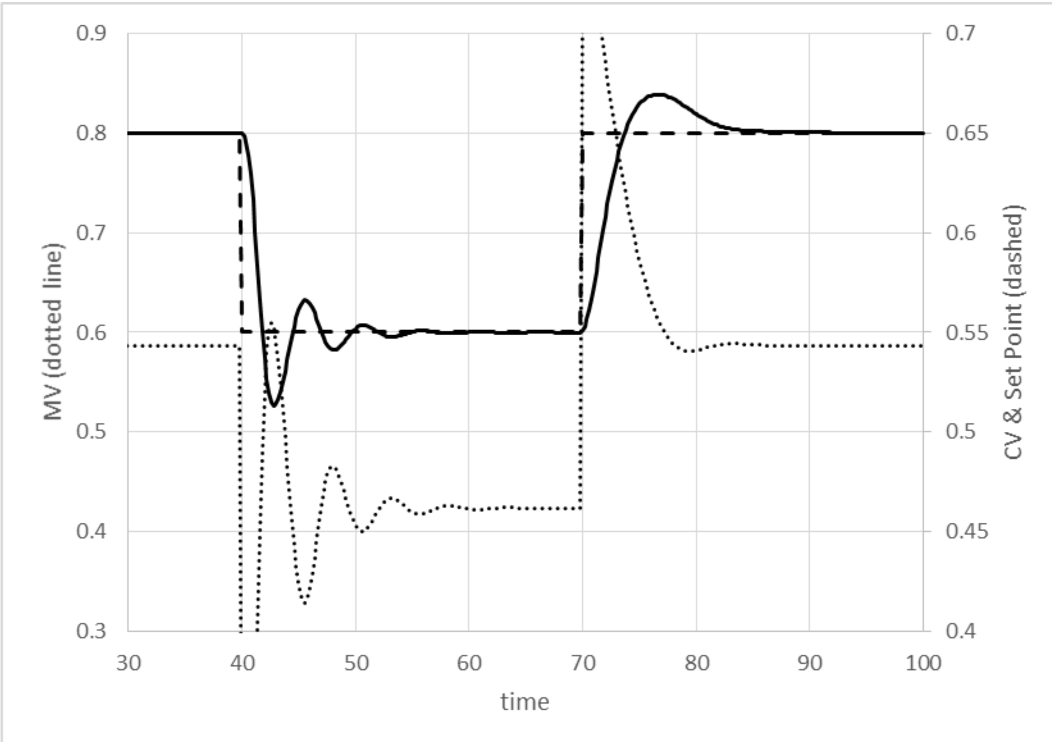

Figure 5 - Nonlinearity

If your process is nonlinear, when you are tuning the controller, stay in a local region. Don't move out of one operating region to another, from one process behavior to another.

If your process is nonlinear, you may need separate tuning values for each operating range. One cannot tune the linear PID controller with one set of coefficient values, and achieve desired performance in both operating ranges. Consider "gain scheduling" as a solution. When the controller moves from one region to another, use values for K_c, τ_i, τ_d that are appropriate for that region. Alternately, tune for the region that is more critical and accept sluggish or oscillatory behavior for the other region.