SISO PID Simulator w Feedforward and Override
User Guide
Simulator Description, Instructions, Exercises

R. Russell Rhinehart
2025-09-20
russ@r3eda.com

Purpose

This simulator is designed for individuals to explore many features of SISO (single input single output) PID (proportional-integral-derivative) control, including:

- tuning,
- measurement noise and filtering,
- process disturbances,
- valve stiction,
- MAN-AUTO transitions,
- feedforward,
- integral wind-up and its reset-feedback solution,
- override,
- various modifications to the basic (standard) PID algorithm,
- options on the process,
- statistical filtering to temper the impact of noise, and
- generating FOPDT (first-order plus dead time) models.

I used this simulator in my classroom course on process control for both in-class demonstrations and assigned student homework. I thought it was very effective as a teaching/training tool. I also think it would be very useful for self-learners to be able to explore PID and associated options and procedures.

You are free to use this simulator however you wish (copy, modify, display, …) and share it with others. Please give credit to the source in any oral or written presentation.

The display is in an Excel spreadsheet, and the code is written in VBA (Visual Basic for Applications) as a macro attached to the Excel file. I think this makes it convenient for use, since most folks have access to Excel. Others may prefer MATLAB, or Python, or other environments. But, the programming environment is irrelevant to the ability to run the simulator or to evaluate the procedures. For those not familiar with VBA I have also posted a "Brief on VBA Programming". The simulator, "SISO PID w Feedforward and Override 2025-09-21, is also posted with this user guide.

In this operating guide, I presume that the reader understands basic numerical methods, FOPDT modeling, controller tuning, integral windup, output spikes, and noise filtering. So, I don't explain those basics.

I hope that you are willing to provide feedback on these materials. Send a note to me at my email, russ@r3eda.com.

PID

The basic PID algorithm is:

$$u = u_0 + K_C \left[e + \frac{1}{\tau_I} \int e \, dt + \tau_D \frac{de}{dt} \right] \qquad (1)$$

This is termed the standard (alternately ideal) PID algorithm, but ISA (or other standard setting bodies) never declared it as "The Standard", because alternate algorithms are also in use, are functional, and are championed by various users and vendors. One alternate PID version is the parallel (meaning parallel gains) version. The other major version is the series (alternately termed rate-before-reset, interacting, or physically realizable) version. Within each version there are other options such as P-on-x, D-on-x, reset feedback, proportional band, etc. The code contains options for you to experience many of the common modifications to the standard PID. I have also had students edit the simulator code to convert the standard PID to some of its other modifications.

The initial bias, u_0, in Equation (1), or its alternate of initializing the integral value when switching from MAN to AUTO, is often omitted in texts, which only show the algorithms in deviation variables. But initializing a controller while in MAN mode is essential for bumpless transfer.

The Process Simulator

The process simulator is 6^{th} order (6 sequential first-order lags) with a delay. You can change each time-constant, the process gain, and the delay. You can also choose a linear process simulator (constant gain) or a nonlinear process, in which the process gain is a function of the MV (manipulated variable, alternately CO for controller output). The nonlinear process has about a 6:1 gain ratio over the MV range.

The code could be rewritten to use any process model that you wish, such as a reactor, or flow system, or flash tank, … To make a lower order process simulator, simply enter zero for some of the 6 time-constants. You can easily (if familiar with VBA) edit the code to increase the process order.

The process has a disturbance which enters in the first lag. The control action enters in the third lag. You can also edit the code to change this.

By selecting options in the main spreadsheet, the disturbance can make noisy random walk drifts, or step changes, or no change. You can choose time-constant and range for the drifts. The drift in disturbance wanders above and below a nominal value. The time-constant is roughly $1/4^{th}$ of the time the drift spends above or below the nominal value, but the trend is random, it does not cycle. The range is essentially the maximum to minimum deviation. So, if you think the disturbance randomly wanders between a high of 17 and a low of 6 the range should be set to 11. And if you think the duration is about 16 minutes, the time-constant should be about $1/4^{th}$ of that persistence, about 4 min. Derivations for this are in my ISA book, titled Nonlinear Model-Based Control.

You can choose to add measurement noise to the CV (controlled variable, the process output to be kept at the set point). You can also have a signal discretization (resolution) feature.

You can select an option to add stiction (slip stick effect) to the valve stem position.

There are many other options.

The Controller

The controller can switch modes from MAN (manual, where the operator sets the MV) to AUTO (automatic, where the PID algorithm sets the MV). In MAN mode the controller is ON not OFF. And for bumpless transfer from MAN to AUTO, in MAN the controller initializes the SP (set point) to the CV, and initializes the bias (u_0 or its equivalent if using reset feedback).

The code in MAN mode is

```
Bias = PID                    ‘controller bias is the controller output
SP = CV                       ‘set point is the current CV value
Integral = 0                  ‘integral is initialized to zero
```

There are many controller options making the code for the PID subroutine seem quite complicated. Basically, the integral and derivative operations in the PID algorithm are solved using forward finite differences. In AUTO mode the uncomplicated code is:

```
e = SP – CV                                   ‘actuating error
Proportional = Kc * e                         'P-on-e, Proportional on error
Integral = Integral + Kc * e * dt / taui      ‘Integral incremental update
efiltered = ff * e + (1 - ff) * efiltered     ‘filter the error to temper noise on the derivative action
Dfiltered = Kc * taud * (efiltered - efold) / dt   'D-on-e, Derivative on filtered error
efold = efiltered                             ‘update for next controller action
PID = bias + Proportional + Integral + Dfiltered        ‘sum all terms for controller output
```

The controller limits the MV to the 0 to 100% range, and resets the integral value if the algorithm would want an MV value out of that range. This limits integral windup. One side of the code is:

```
If PID > 100 Then
   PID = 100
   Integral = 100 - bias - Proportional - Dfiltered
End If
```

You can choose to

- Filter the D mode to temper the impact of measurement noise
- Tune the Kc, taui, and taud terms
- Select P-on-x (proportional on measurement) to prevent output bumps upon set point changes
- Select D-on-x (derivative on measurement) to prevent output spikes upon set point changes
- Select erf (external reset feedback) which uses a first order filter on the MV instead of the integral on the actuating error to calculate the adjustable bias, which prevents integral wind up.
- Add a SPC (statistical process control) filter on the MV to temper the propagation of measurement noise onto the MV. An SPC filter holds the output at a constant value, until it is statistically confident that the noisy output from the controller is signaling a real change. Then it changes the signal to the process. The SPC filter is also explained in my ISA book, Nonlinear Model-Based Control.

- Add feedforward action to the feedback (PID) controller.

The feedforward action is calculated by the traditional ratio of FOPDT models for how the disturbance and MV affect the CV, and the feedforward action is added to the PID controller output.

The display

You enter options or numerical values for process or controller coefficients in the yellow highlighted fields on the "Simulation" worksheet. The blue fields describe events or trials that you can choose. Enter the Trial number in the yellow highlighted cell R24C19 (in R1C1 notation, alternately cell R24 in A1 notation) for the "Experimental Events".

The green highlighted cells reveal goodness of control metrics. They are calculated during the time range that you enter in the adjacent yellow highlighted cells. If you only want regulatory mode metrics, start the evaluation time after the controller gets the process to an initial steady state after startup. The integral metrics (IE integral of the error, ISE integral of the squared error, IAE integral of the absolute value of the error) and 3Sigma (effectively the 99% limits of the CV from SP) are normalized by the data collection time. Travel is the total valve movement (if the MV is 50% then 80% then 60% then the valve traveled a total of 30%+20%=50% to make a 10% position change from 50% to 60%). nPenalty is the time normalized violation of specification (integral of violation magnitude). You would want to tune and specify a controller option to minimize any of these metrics.

The graph displays key variables over the 400 minute controlled process simulation. The illustration in Figure 1 starts the controller in MAN mode then switches to AUTO and changes the SP at a time of 25 min. The CV (black fuzzy trend) has noise and is also influenced by the ever changing disturbance (blue). The black solid line is the set point. In MAN mode the SP tracks the CV, in AUTO it is changed to 90 CV units. The red line is the controller output, the MV, which continually changes to counter the disturbance effect to keep the CV near the SP. The dotted line at CV=100 is a specification limit on the CV, not to be violated. If the SP was set to the Spec Limit, then disturbances would cause violations, so the SP is 10 units below the Spec Limit to prevent violations. This deviation from the limit is termed quality give-away because on average the product has a CV value of 90 when the target is 100.

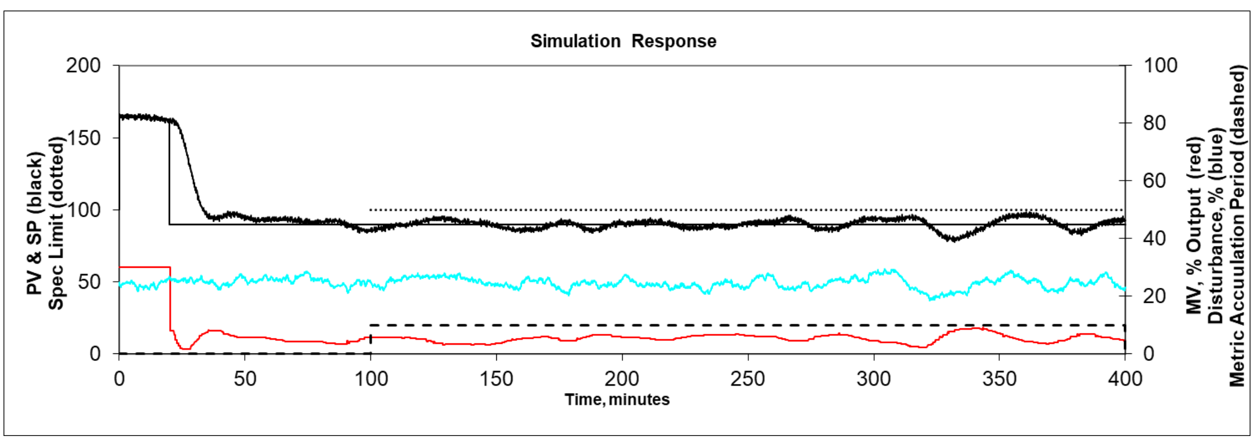

Figure 1 – Output Illustration Feedback Only

The dashed line that makes a step and hold at a time of 100 min indicates the time interval that the program accumulates goodness of control metrics.

Repeating the trial with feedforward, Figure 2, the control strategy does a much better job, and the SP can be moved nearer to the specification limit with no violations, with less quality give-away.

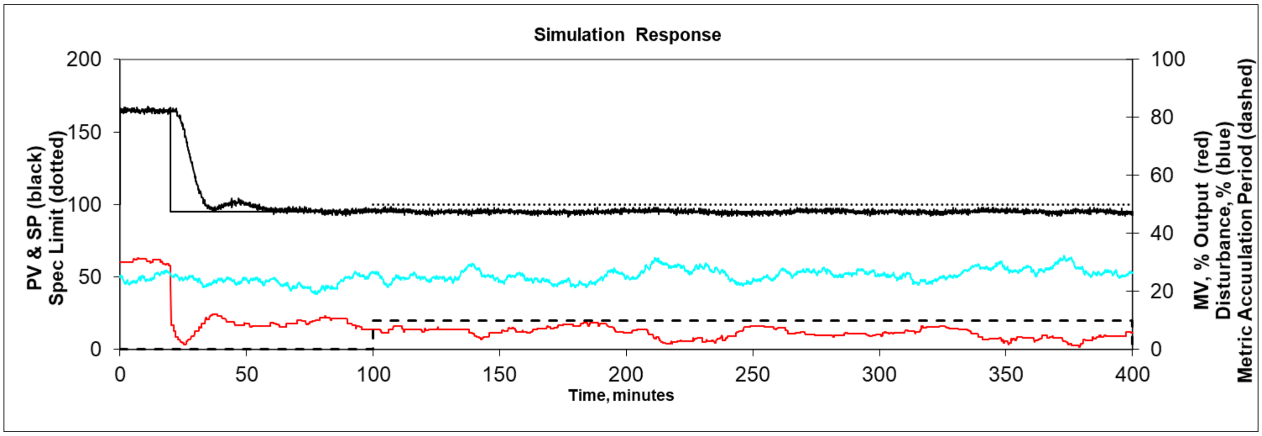

Figure 2 – Output Illustration with Feedforward

Goodness of Control Metrics

Goodness of control metrics quantify how well the controller does its job. Usually, textbooks use rise time and QAD responses to set point steps. But, I object. Tuning a controller to minimize rise time or to get quarter-amplitude-damped oscillations makes the controller uncomfortably aggressive. Aggressive control causes upsets to the utilities which upset other users, oscillations to the CV propagate downstream and upset others, oscillation makes managers wary of constraint violations and energy use, and I think it is a scary thing to hear the valves hissing opening and closing when alone at night surrounded by spooky equipment in the unit. Smooth but fast enough comfortable CV transitions to new set points is the desire.

The MV action is also part of the comfort issue. We want a progressive push, not an MV oscillation.

Consider: How do you accelerate your car when the speed limit changes? If you use a QAD approach, the acceleration-deceleration will upset passengers, other drivers will be annoyed with your speedup and slow down, and it wastes fuel. Tempering MV action is as important as rapid return of the CV to the set point.

However, most chemical processes operate at fixed set points, and the controller job is regulatory action (countering disturbances to keep the CV at the set point) not servo (following a changing set point).

Values in the green highlighted cells in Column 17 (Column Q) quantify the metrics. Although termed “goodness of control metrics” they are actually measures of undesirability that need to be minimized.

IE is the integral of the actuating error. Although called “integral” (which gives mathematicians great existential pleasure to use Laplace transforms to solve for the IE), it is not actually a calculus operation.

The irregularity of the vagaries of disturbances preclude any sort of analytical calculus assessment of IE. It is the sum of actuating errors from the start accumulation time to the stop accumulation time.

$$IE = \sum_{i\ start}^{i\ stop} e_i \tag{2}$$

If "+" deviations are balanced by "-" deviations over a time interval, then in-line blending downstream makes the CV appear to be at the set point. In this case IE is an appropriate metric. But if there is a positive or negative bias, then a longer accumulation period will lead to a larger IE value. Normalizing IE by the number of samplings makes it independent of the collection duration.

$$nIE = \frac{1}{N}\sum_{i\ start}^{i\ stop} e_i \tag{3}$$

If there is not adequate in-line blending, then both the "+" and "-" deviations will be objectionable. Further, either deviation may lead to constraint violation. In this case the integral of the absolute error would be an appropriate metric. Normalized

$$nIAE = \frac{1}{N}\sum_{i\ start}^{i\ stop} |e_i| \tag{4}$$

However, badness is usually proportional to the square of the deviation. Then the normalized "integral" of the squared deviation is

$$nISE = \frac{1}{N}\sum_{i\ start}^{i\ stop} {e_i}^2 \tag{5}$$

If the process averages at the set point, then the square root of $nISE$ is the CV standard deviation. And if the deviations in the actuating error are normal (Gaussian) then the 3-sigma value defines the deviation that the set point must be from a constraint (quality give-away) to be roughly 99% certain that there will not be constraint violations.

$$3\sigma = 3\sqrt{nISE} \tag{6}$$

Those metrics all relate to the CV. But the MV is also important.

Travel is the distance a valve must move, and since it continually moves in the regulatory mode the total Travel should be normalized by the number of samples. Whether it is a valve or cascaded set point, determine travel from the controller output

$$nTravel = \frac{1}{N}\sum_{i\ start}^{i\ stop} |u_i - u_{i-1}| \tag{7}$$

If there is a constraint or specification that should not be violated, then the number of times it is violated, and the magnitude of the violation would be a measure of undesirability

$$nPenalty = \frac{1}{N}\sum_{i\ start}^{i\ stop} \begin{matrix} 0, & IF\ CV_i \le Spec \\ CV_i - Spec, & IF\ CV_i > Spec \end{matrix} \tag{8}$$

Demonstrations and Student Exercises

Use Trial 1 to see the impact of step changes in the MV and disturbance on the CV. Generate FOPDT models of how the MV and disturbance impact the CV. Do it with noise off and on, drifts in the disturbance off and on, and with a linear and nonlinear process. Compare several methods of generating the FOPDT models and comment on the difficulty of determining steady state for starting and ending values for the calculation.

Use Trial 2, with the controller in MAN mode making a sequence of equal MV steps to reveal linearity or nonlinearity of the process gain. Select linear or nonlinear in cell R21C20. This should reveal that the FOPDT model depends on the MV value.

Use Trial 3 to see how the up-down-down-up sequence balances nonlinearity on FOPDT model coefficients and balances plus and minus CV deviations from the base conditions. Repeat with noise and disturbance on to realize the difficulty of determining steady state and the FOPDT coefficient values and the benefit of the 4 steps of the U-D-D-U method over the traditional one step reaction-curve approach.

Use Trial 4 to generate a skyline pattern in the MV with many steps and use regression to determine the best FOPDT model. This provides many, not just 4, steps and does not cause as large CV deviations form the base.

Use Trial 5 for tuning controllers by heuristic methods, and for testing the tuning results from FOPDT rules or the ultimate method rules. It makes a sequence of step changes in the SP (set point) value. Turn noise and disturbances off to remove those confounding effects to clarify the essence of the tuning method or tuned controller, but then add noise and disturbances to reveal the reality of getting FOPDT coefficients, determining when steady-state has happened, and to see the uncertainty of the goodness of control metrics.

Also, use Trial 5 with the external MV limit of 40% to compare the impact of an external override on integral windup (using erf or not). Choose no noise or disturbance to clearly see the delay in returning control with the wind-up of the integral when the override is active, or no delay using the erf (external reset feedback) option.

Use Trial 6 to see the impact of tuning and controller and strategy options and noise and disturbances on the quality give-away required to prevent specification deviations. Most chemical processes run at relatively steady conditions, and are continually subject to noise and disturbances. In this case, servo action (set point changes), are irrelevant to best tuning and goodness of control. In this case nISE and 3sigma metrics directly relate to quality give-away and manufacturing cost. Investigate how tuning and controller options best minimize nISE, 3Sigma, and nPenalty. Choose the specification limit value, and noise and disturbance effects. Tune the controller and choose FF ON or OFF. See how FF improves control (as in Figure 2) and quantify how the penalty for spec violations and quality give-away depend on tuning, noise and disturbance magnitude, etc.

Use Trial 7 (one step in the SP) and the ultimate method to generate controller tuning. Consider the impact of the ultimate method on down-stream processes and utility swings as you search for the ultimate gain. Compare how you accelerate your car when the speed limit changes to how the tuned controller changes the MV. Do you drive using aggressive pedal changes that cause oscillations that eventually damp out, or does your speed approach the speed limit in more of an over-damped, asymptotic approach? Consider the operators' reaction to aggressive oscillating controllers and the need for fine-tuning the PID controller after the ultimate or reaction curve approaches. Realize that you would never want to use the ultimate method in practice.

Use Trial 8 to reveal the controller response to a simple step in the disturbance. Turn off noise and disturbances to remove confounding aspects and clarify the impact of tuning choices and FF.

Trials 8 and 9 make feasible but large set point changes with or without a disturbance step. These could be used for testing controller options and tuning.

Trials 10 and 11 make an infeasible setpoint change and return and can also be used to quantify the benefit of erf over the classic integral.

For any of the Trial choices, there are many process options. The yellow highlighted cells in Column 20 (column T in A1 notation) permit you to change process dynamics, gain, or linear to nonlinear.

In the highlighted cells of Column 11 (K in A1 notation) you have a choice of many options on the controller. Values in Column 13 (column M) are suggested values to reveal the various features and options.

- Watch: Enter "Y" to observe the output develop in real time, or "N" to show results faster.
- D-on: Enter "e" to have derivative on the actuating error, which leads to output spikes on a set point change, or "x" to have D action on the CV, which eliminates output spikes.
- P-on: Enter "e" to have proportional on the actuating error, which leads to an output bump on a set point change, or "x" to have P action on the CV, which eliminates output bumps. This is often termed set point response softening. Classic tuning requires P-on-e.
- Initialize U0: Enter "Y" to initialize u_0 in MAN mode or "N" to not. Use a MAN-AUTO Trial 6 to observe the bump (or not) when the controller mode changes.
- Tau Noise: Enter the time-constant for the noise, this defines autocorrelation in the noise added to the CV measurement.
- Range Noise: This defines the magnitude of the noise.
- erf: Enter "Y" to use external reset feedback, or "N" to use the classic integral. Ideally, when there is no constraint, the integral and reset feedback version are identical. But here, the slight difference is due to the simple numerical methods of approximating the integral or the first-order filter. You might want the students to explore the trapezoid rule of integration to make it one-step more ideal, but also have them judge whether the benefit is consequential in the presence of just a bit of noise or disturbance drift.
- Kc, Taui, and Taud: These are where you enter the tuning values for the three controller operations.
- D-Filter: This is the time-constant for filtering the derivative. The derivative action amplifies CV noise. The integral does not, and the proportional only slightly. To temper the impact of CV noise on control action, many filter the measurement, but this is unnecessary on P or I, and slows the P response. In spite of what texts might reveal, filter the D action only.
- OP Limit: This is where you would enter an external override to the controller output, the MV. For safety, auxiliary variable management, or select control features, an external override may constrain the output that actually goes to the process. When this happens, the integral will wind-up to a limit, and then take time to wind-down when the override is removed. erf is my choice to prevent integral wind-up, but commercially there are several alternate fixes.

- Tau Disturb: Enter the time-constant for the disturbance. This defines persistence of the random wandering of the disturbance above and below zero.
- RangeDisturb: This defines the magnitude of the disturbance above and below the nominal value.
- FF: Enter "Y" or "N" to include the feedforward feature to the controller.
- K-FF, FF-lead, FF-lag, FF-delay: These are the four feedforward tuning coefficients, obtained from the ratio of the FOPDT models for how MV and disturbance affect the process.
- Resolution: Many measurement devices or signal transducer/transmission systems have low resolution. It might be that the measurement range is very large relative to the bit length of the processor, or it might be the result of nonlinearity in high gain regions. Your digital watch has a resolution of 1 second. It does not display time continuously advancing, but only acknowledges when it has advanced each second. Time does not pause until the watch displays the next second. Similarly, instrument system resolution makes it appear that the CV does not change until its value rises above or falls below the resolution interval of the prior value. This aspect might be alternately termed discretization.
- Stick Band: Valves have a stick-slip behavior termed stiction. As the pressure on an actuator changes, the valve position remains fixed until the force balance between the spring and actuator overcomes the static friction of the packing grabbing the stem. Then the stem jumps to a new position. Stick band is the magnitude of this effect as a portion of full stroke.
- Spec Limit: Enter a value for the CV which represents a specification limit.
- Q GiveAway: This is the setpoint deviation below the Spec Limit.
- SPC Trigger: This is an alternate filter to temper the impact of noise. Rather than filtering the CV or the D action, this filters the output of the controller. And rather than using a first order filter, which causes a lag, this uses a Statistical Process Control (SPC) approach that holds the prior MV value until there is statistical evidence that the controller wants a new value. The value you enter is roughly the sigma level for taking action. A 2 sigma equates to being roughly 95% confident that a change is justified. A 3 sigma is roughly 99% confident, and takes longer, but reduces travel. A 1 sigma is roughly 70% confident, and takes action faster, but does not temper noise as much.
- Accumulation Time in Column 19 (Column S) is your choice of the simulation time to start and stop calculating the goodness of control metrics.

Explore the impact of controller and process options and coefficient values on set point response and regulatory action. Use the goodness of control metrics (green highlighted cells in Column 17, Column Q) to quantify the impact.

Other training teaching experiences

The above exercises and demonstrations are included in the programmed options. If you are willing to do some coding in VBA, here are some useful exercises:

- Convert controller gain to proportional band.
- Convert reset time (integral time) to reset rate.
- Filter the CV.
- Convert the controller to the parallel gains or the series version.
- Add lags to make the process simulator higher order.
- Explore the trapezoid or Simpson's rule of integration.

- Replace the elementary process simulator with one based on first principles models of a SISO process.
- Switch the input lag of the controller and disturbance. This will switch the delays on the FOPDT models which will generate a negative delay for the FF controller. A negative delay is infeasible to implement. So, enter zero for the delay, and adjust the lead coefficient to accelerate the initial FF action to compensate for the needed advance action.
- Add rise time and damping ratio to the goodness of control metrics.
- Add a trial with a ramp change in the set point or disturbance, to see the CV following lag then delay.
- Derive Equation (6)
- Convert nPenalty to reflect the squared deviation not the deviation.