Project work

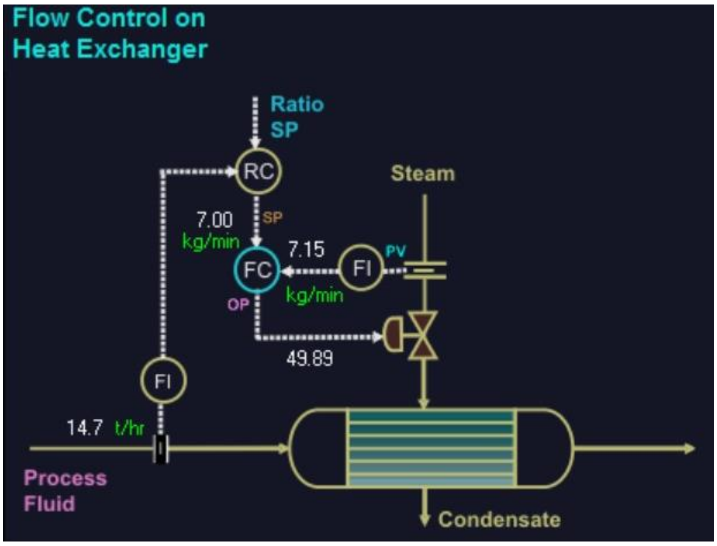
Tune Flow Control PID on a Heat Exchanger

Angela Mastrilli 0612202243

September 4, 2021

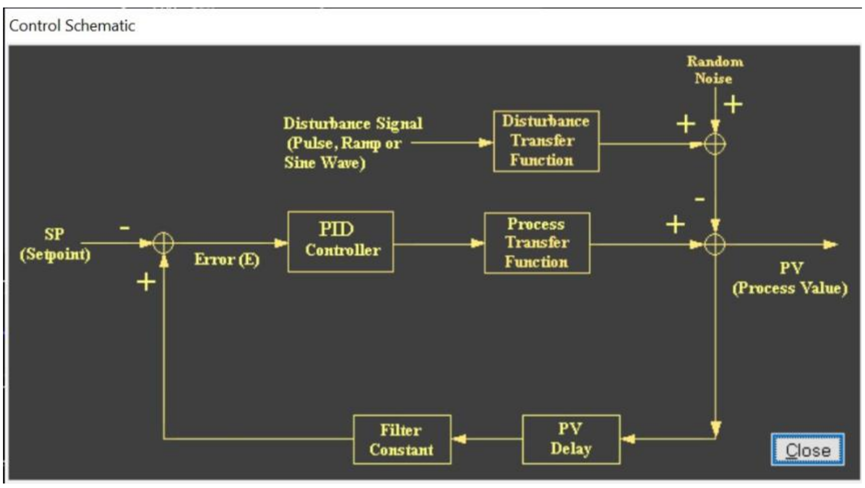
Heat exchanger system is widely used in chemical plants to transfer heat from a hot fluid to a cooler fluid, so temperature control of outlet fluid is of prime importance. To control the temperature of the outlet fluid a PID controller can be used.

The behavior of the assigned heat exchanger can be studied using the "tuning practice" section of SIMCET, a PID tuning training software.

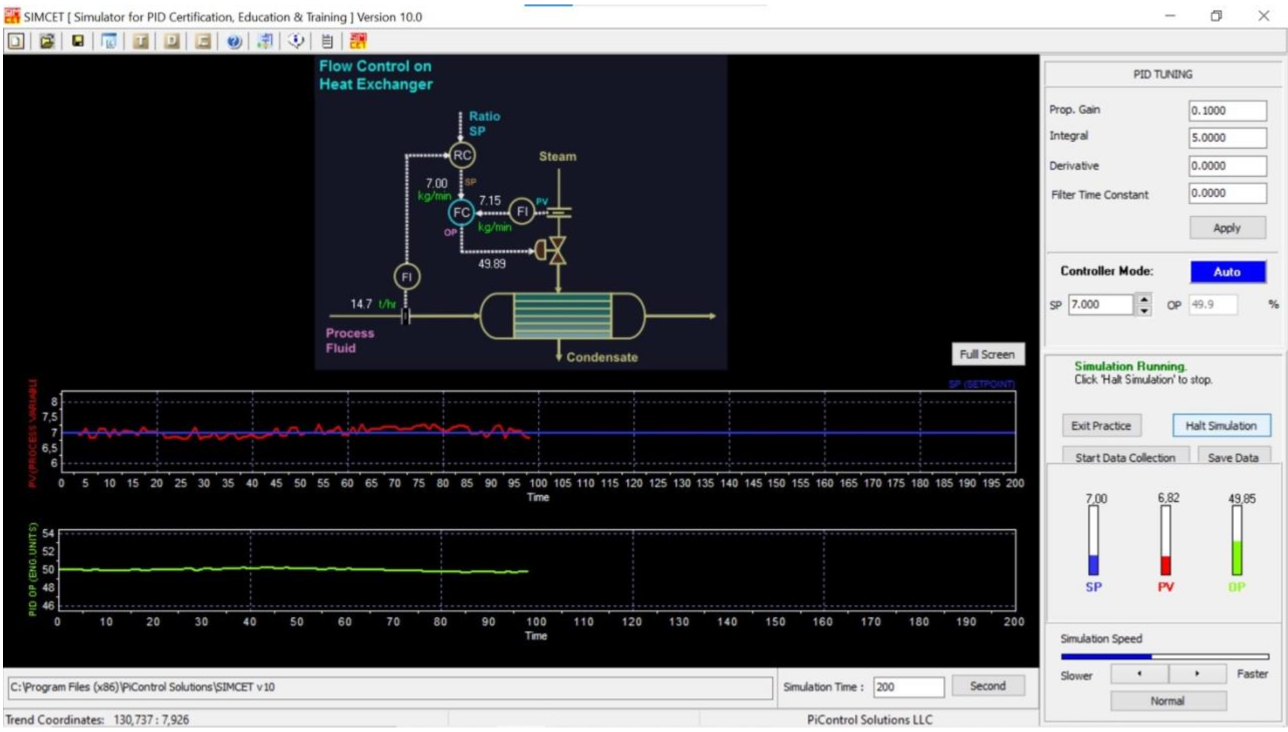


The process fluid is supplied to the heat exchanger, which heats up the fluid, to a desired set point, using steam. The heat exchanger consists of bundles of pipes enclosed within a cylindrical shell. The steam flows through the pipes, whereas the process fluid flows through the shell. After the steam heat up the process fluid, the condensate goes out of the system.

Proper devices measure the steam flowrate (our Manipulated Variable), and the process fluid flowrate. This information is sent to a flowrate controller. It compares the measured signal to the set point value and acts on a final control element, a valve, that regulates the steam flowrate entering the system.



Step#1



Once the window relating to the Flow control on heat exchanger is opened, the simulation starts to run. Below the process graphic, there are two trend display boxes. In the upper trend display, the red colored trend is the PV signal and the blue colored trend is the PID setpoint. The lower window shows the PID output trend in green color.

Near the top right corner of the simulation screen, we can see the PID tuning parameters, located below the label "PID TUNING":

- Proportional gain: or proportional band PB in the PID equation;
- Integral: it is the integral constant in the PID equation and is also called Reset;
- Derivative: it is the derivative constant in the PID equation;
- Filter time constant: it dampens a noisy signal, which can cause excessive jittery movement in the PID's output. This can result in undesirable control action including excessive wear on the valve. Filtering reduces excessive valve movement, but excessive filtering adds delay in the loop and harms control quality.

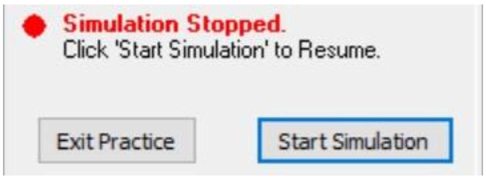
Going down we can see the label "Controller Mode" and a button that allows us to toggle between Auto and Manual modes. If the button label displays Auto, we can change the SP (set point) value, but not the OP (controller output). In this way the program allows us to act on the system by studying the *closed-loop* dynamics. If the button label displays Manual, we can change the OP value but not the SP. The PID's output OP goes directly to the control valve. 100% OP indicates a fully open valve. After changing the OP value in manual mode, we can observe the *open-loop* shapes of transfer functions.

Near the bottom right corner of the screen, we can see color bars showing the SP, PV and OP values. The height of these bars corresponds to the actual values shown as a percent of the instrument measuring range.

We can also modify the "Simulation Speed" through the buttons Slower, Faster and Normal.

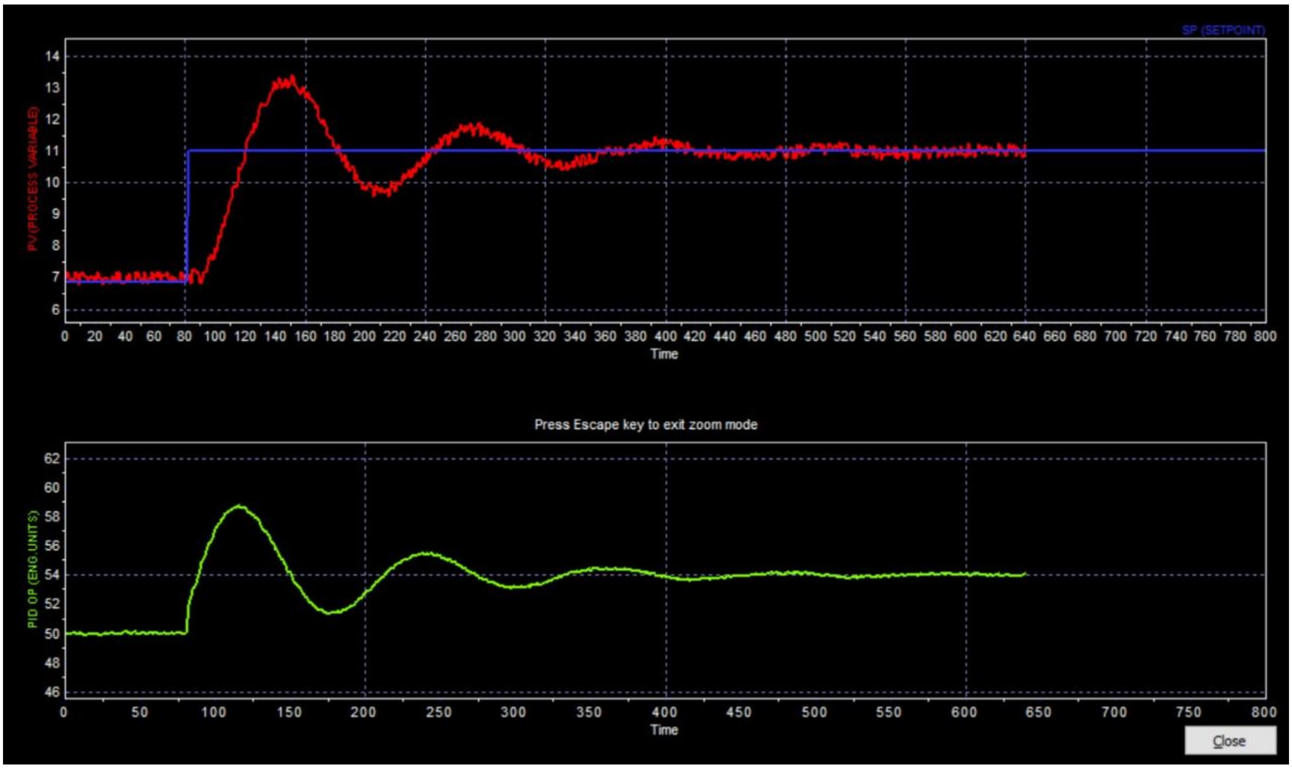
The "Simulation time" allows us to specify the time span of the simulation. It should be set longer for slow processes, in order to observe the complete dynamics after a set point change or disturbance.

Step#2



If the SIMCET program is running, a button displays Halt Simulation. Clicking on this button will stop the program execution and change the label to Start Simulation.

Step#3

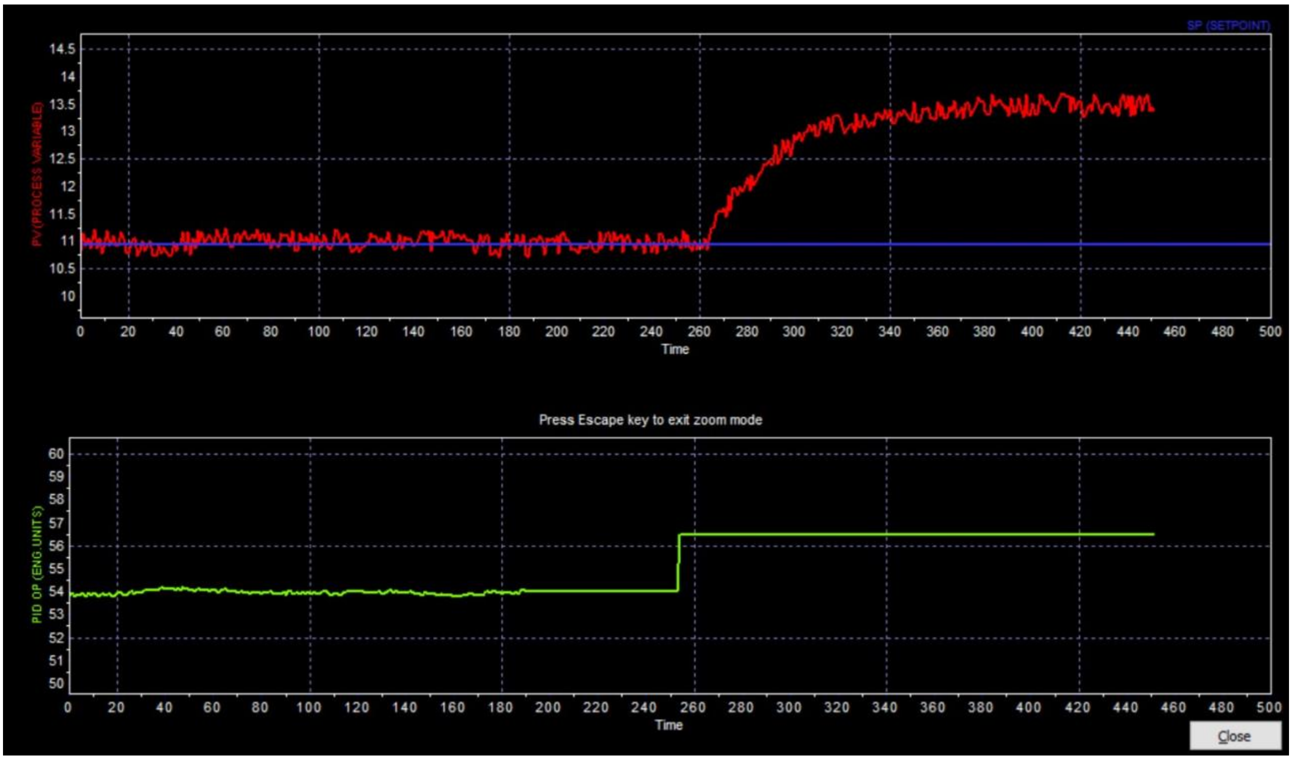


We click on Start simulation and re-start the program.

The purpose of the tests we are about to carry out is to reveal the relationship between the control loop's Process Variable (PV) and the associated Controller Output (CO). That can be accomplished either in open-loop, by manually adjusting the Final Control Element, or in closed-loop, by changing the Set Point (SP). To start, we want to focus on the closed-loop and we want to carry out a Step Test. We have to make a large enough change to the SP value, so that it results in a clear PV and CO response.

We change the SP value from 7 to 11 and we can notice a step variation of the blue trend of the set point, which will move up to reach the new value. The PID controller (FC) reduces the OP, thereby putting less steam flow into the heat exchanger. The PV (steam flow) reduces and reaches the new SP.

Step#4



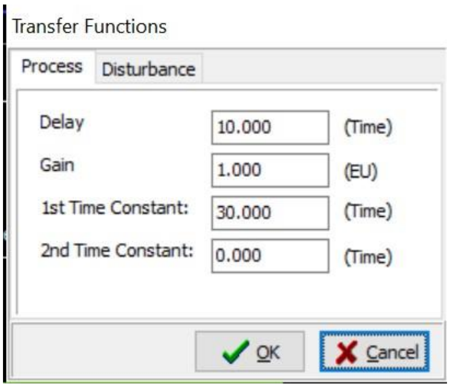
Now we focus on the open-loop case. We click on the Auto labeled button and we toggle from the auto mode to the manual mode. We can notice that the green trend draws a flat line because the PID is in manual mode and its OP value is fixed. Then we increase the OP value by about 5%, so we switch from 54 to 56.5. We have changed the steam valve position by about 5% and we will see the red trend (PV) changing steadily to reach the new value (self-regulating).
In this way we perform a step test, following the "*process reaction curve*" procedure:

- We make a step-wise variation in the controller output;
- We wait until another stationary state is asymptotized and we notice that the curve representing the PV does not start rising just at the same time when the step has been applied in the input variable, but there is a delay, a dead time.
- We can collect and save data in a file, we can get an FOPDT model of dynamic data fitting and make a tuning of the PID controller.

The transfer function for FOPDT model is:

$$G_{FOPDT}(s) = \left[\frac{K_p}{\tau_p s + 1}\right] e^{-t_d s}$$

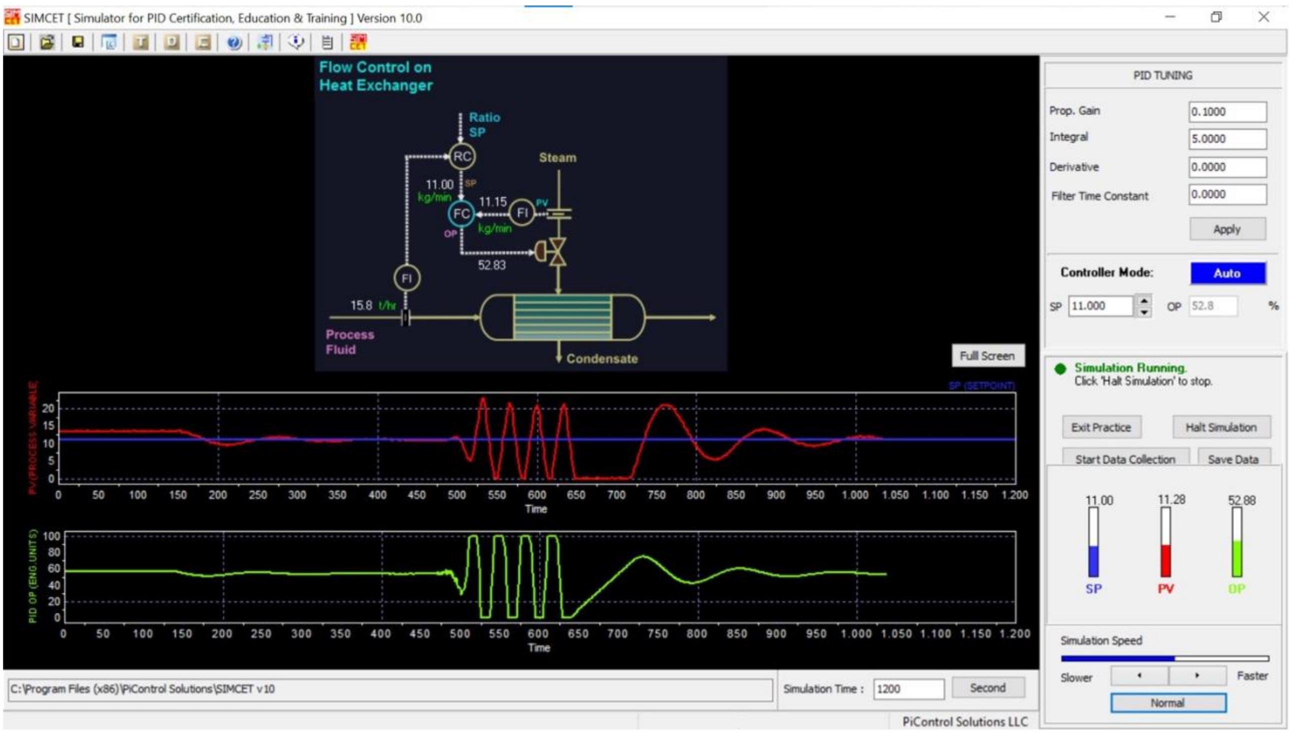
We click on the Transfer Function icon in the top-bar to specify the parameters:



We can also identify the parameters by visual examination of the trends:

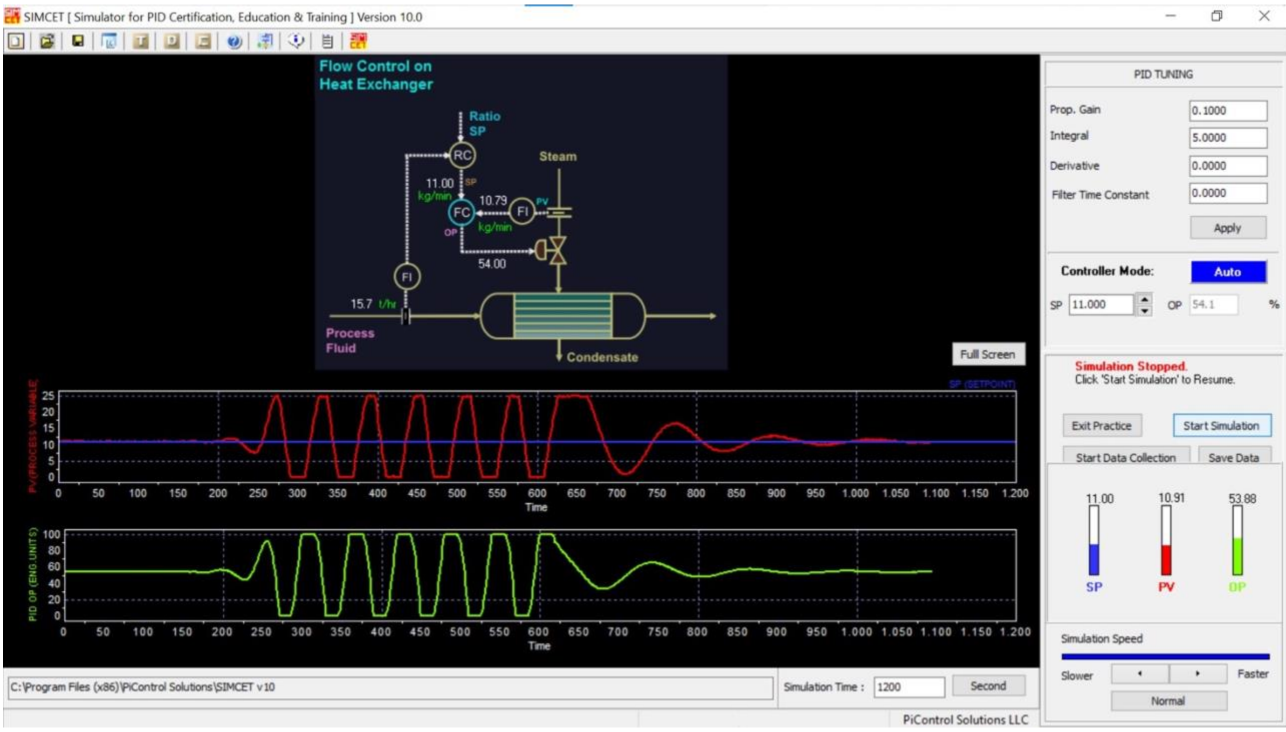
- Delay = t_d = time elapsed before the system response = 10s
- Gain = K_p = $\frac{output\ at\ steady\ state}{input\ at\ steady\ state} = \frac{13{,}5-11}{56{,}5-54} = 1$
- 1^{st} Time Constant = $\tau = \frac{output\ at\ steady\ state}{slope\ of\ the\ response\ at\ the\ inflection\ point} = \frac{13{,}5}{0{,}45} = 30s$

Step#5



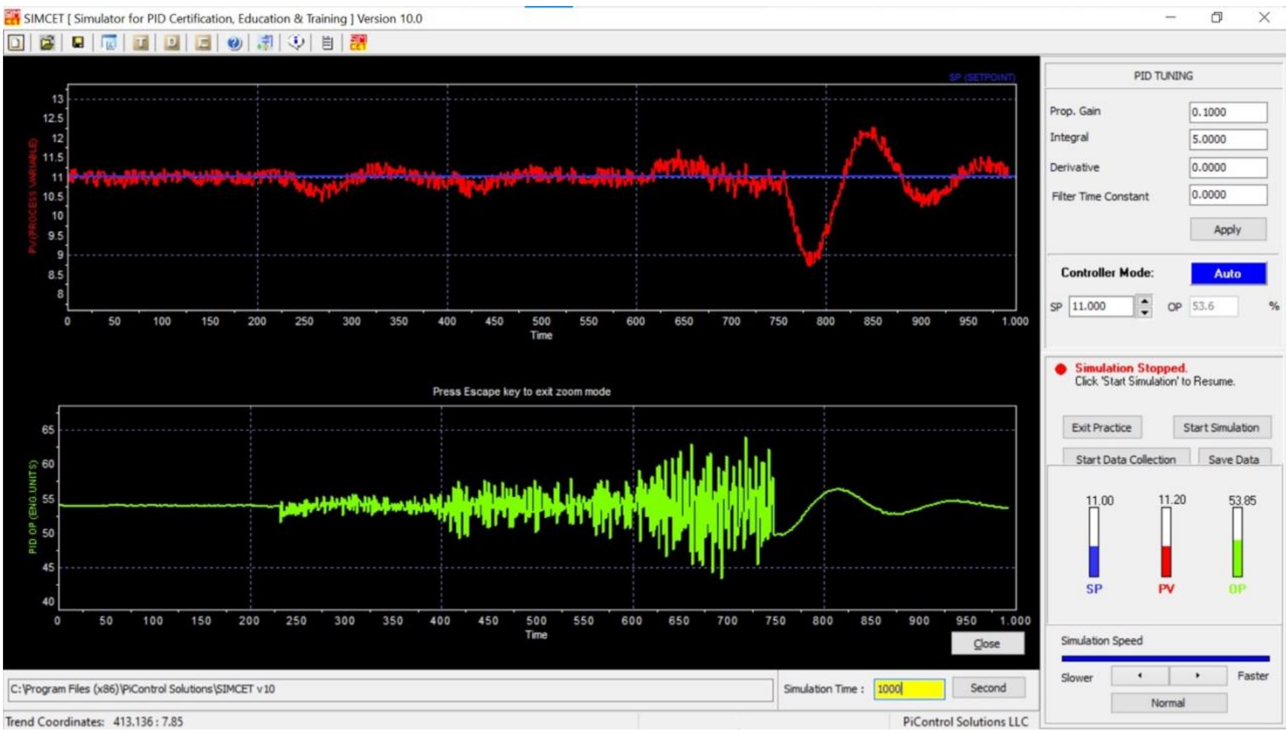
We click on the Manual button to toggle the PID to Auto mode again.
We notice that the PV once again goes back to the SP. Then we try to optimize the PID tuning parameters, starting with proportional gain.
If we change its value from 0.1 to 3, we can notice that the proportional action becomes too high and the controller starts to oscillate. The steam flow control valve goes from 0 to 100%. The FC is unstable. We will get a PV Low Alarm and a PV High Alarm as the PV gets too low or too high. If we change the proper gain back to 0.1, the controller becomes stable again.

Step#6



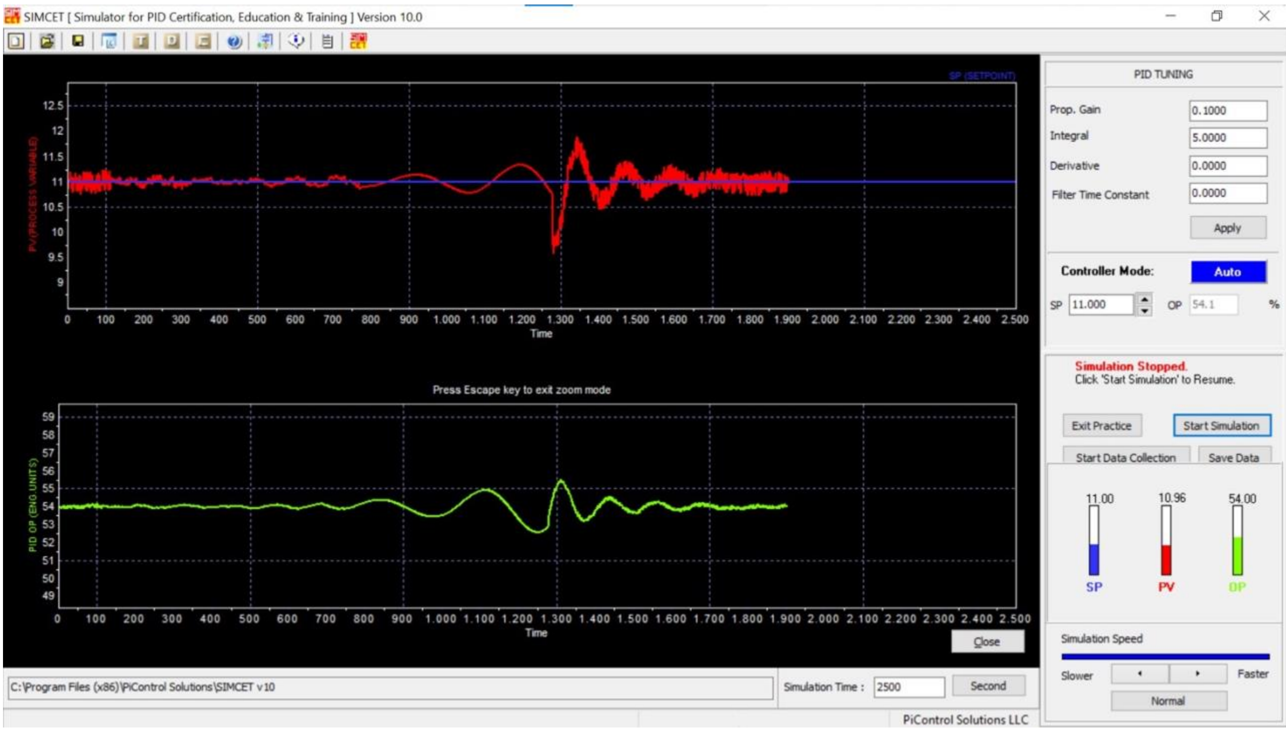
With the FC still in Auto mode, we change the Integral value from 5 to 0.5. We can notice that now the integral action is too high and the controller starts to oscillate. The steam flow control valve goes from 0 to 100%. The FC is unstable. We will get a PV Low Alarm and a PV High Alarm as the PV gets too low or too high. If we change the Integral back to 5, the controller starts to become stable again.

Step#7



With the FC still in Auto mode, we change the Derivative value from 0 to 10. We can notice that now the derivative action is too high and the controller output (control valve position) becomes very jagged and too sensitive to the noise. The steam flow control valve goes from 0 to 100%. The FC control action gets very erratic. Then we try with higher values, 25 and 50, and we can notice how jagged the control action becomes. At the end we set the Derivative back to 0.

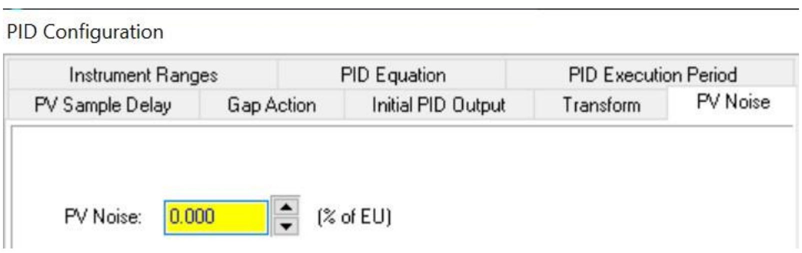
Step#8



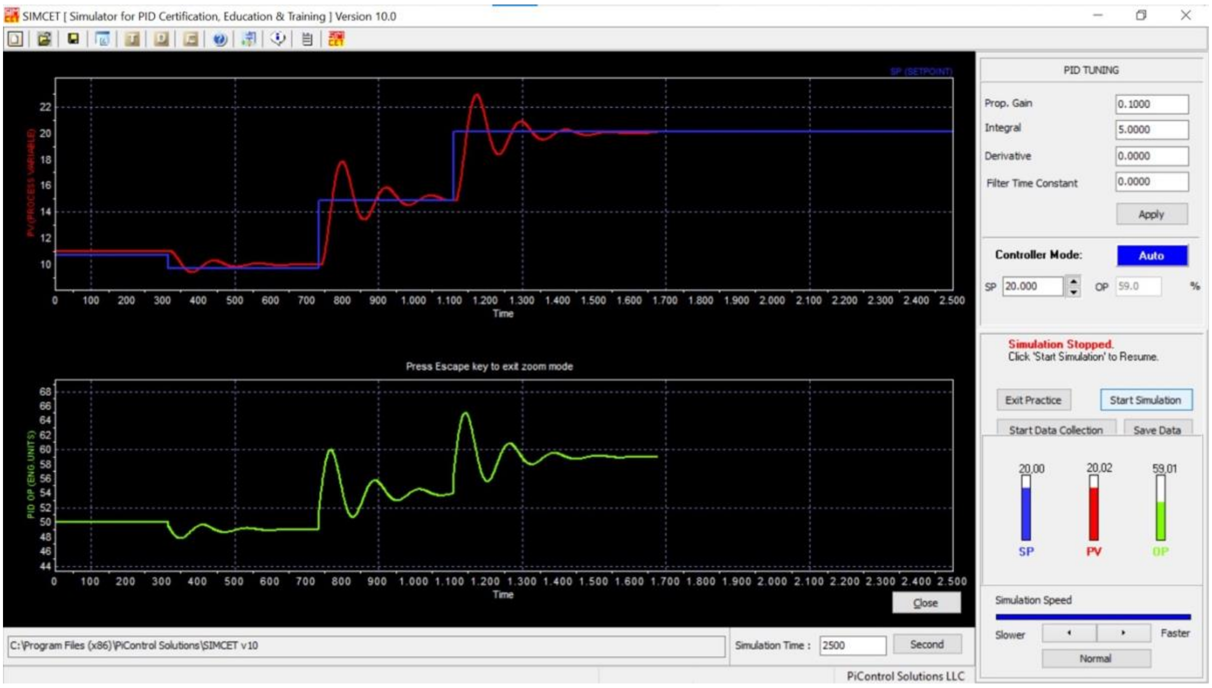
With the FC still in Auto mode, we change the Filter Time Constant value from 0 to 2. We can notice that now the PV signal, that was so noisy before, is more stable. So we increase the value to 5 and we wait for about half a minute and we notice the stabilizing effect on the previously noisy PV. If we increase the value up to 100, we see that the filtering is so excessive that it manifests as extra dead time in the loop and harms control performances. After we finish studying the filter time constant, we set it back to 0.

Step#9

Noise, the random variation in our data signal, is impossible to avoid. It can obscure a process response to change, and make it difficult to calculate an accurate model. When tuning a PID controller, we simply have to deal with it. This software allows us to modify the noise value and to study the process under different conditions.

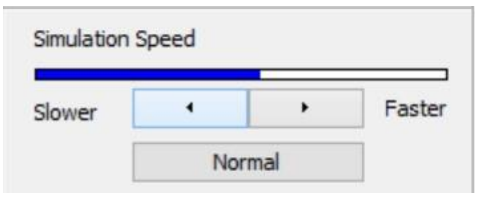


We click on the PID Configuration icon in the top-bar and change the PV noise value from 1.75 to 0.



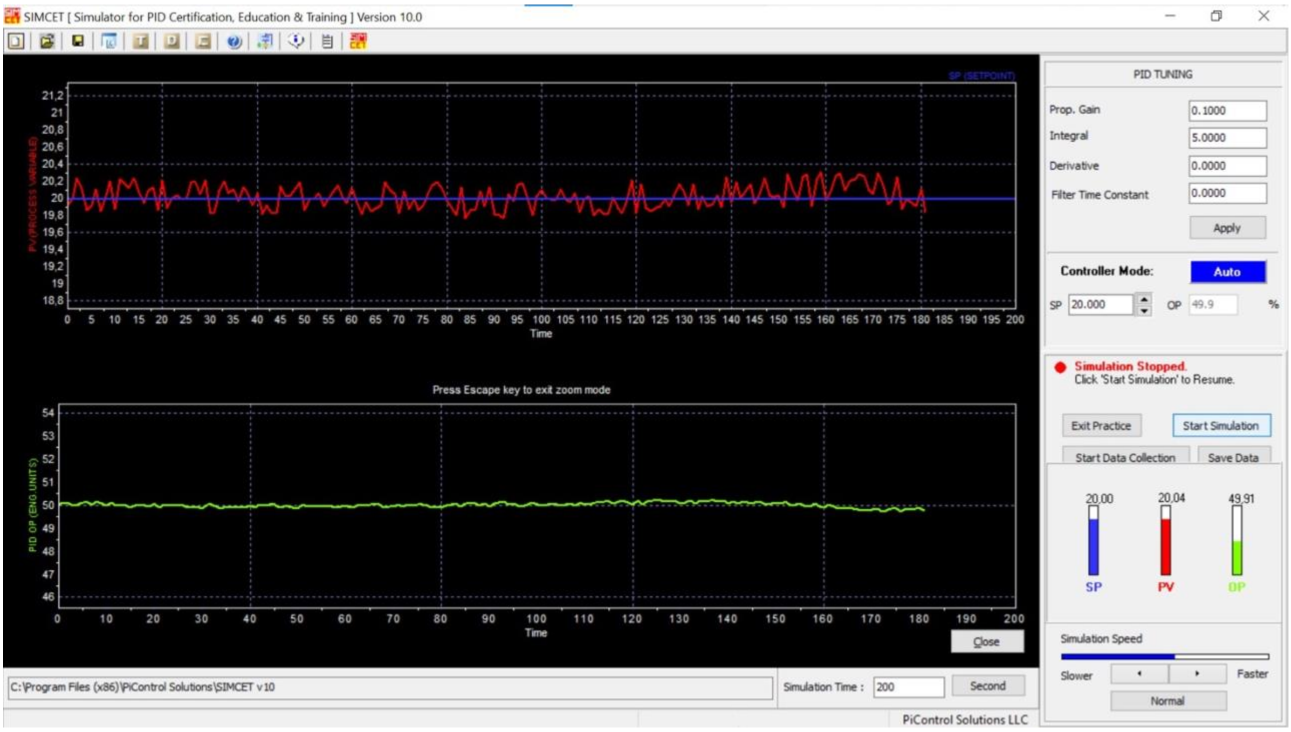
We can immediately notice that the noise in the PV is gone. Then we change the SP value to 10, 15 and 20, and we can see that the PV signal reaches the new SP without the previous noisiness. After we finish to study the PV noise, we set it back to 1.75.

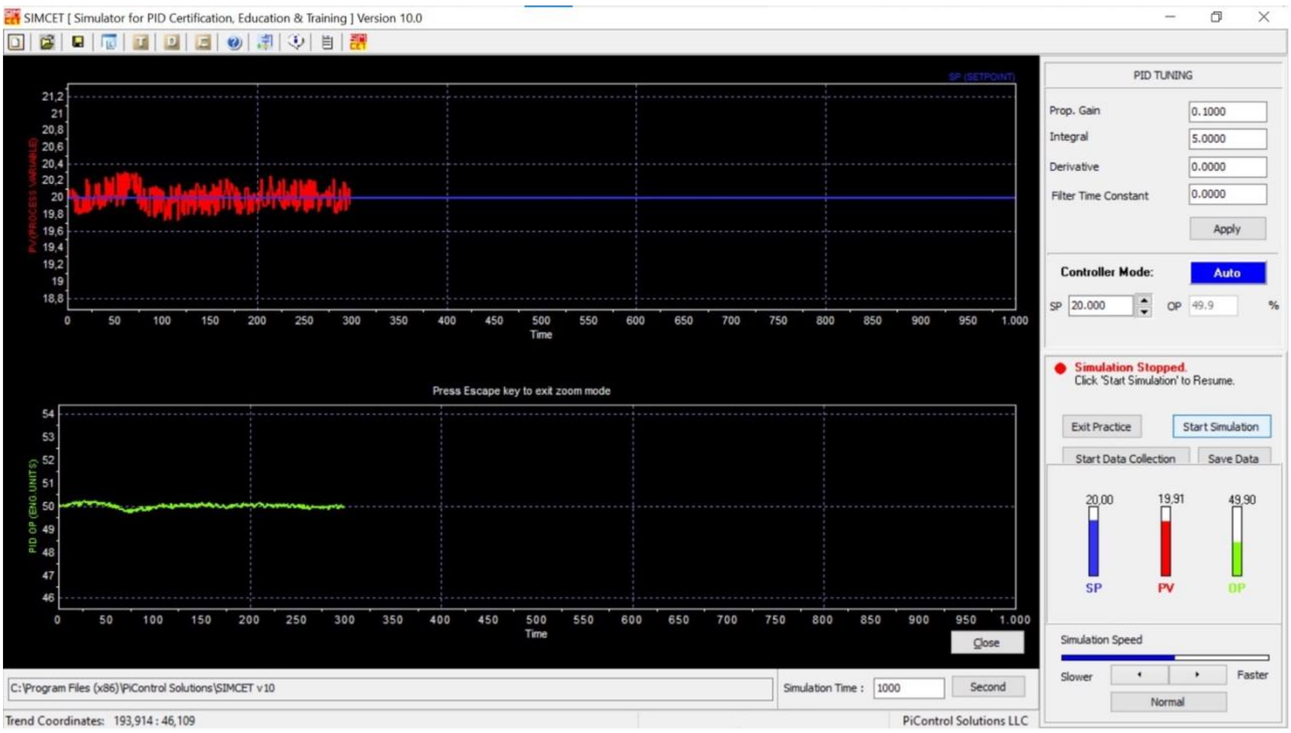
Step#10



The Simulation Speed control is located near the bottom right corner of the screen. We can click on Slower, Faster and normal, to slow down, speed up and reach the maximum simulation speed. When the simulation speed bar is fully blue, this means simulation speed is fastest. When the simulation speed bar is fully white, this means simulation speed is slowest.

Step#11

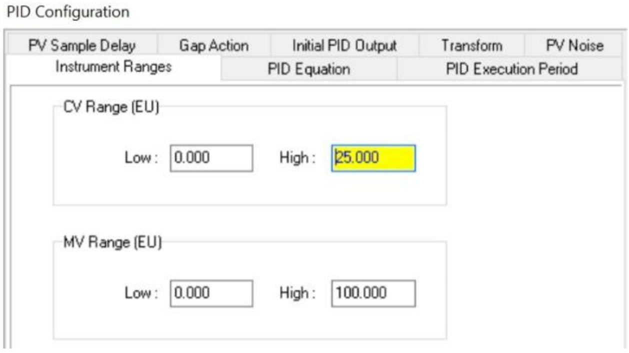




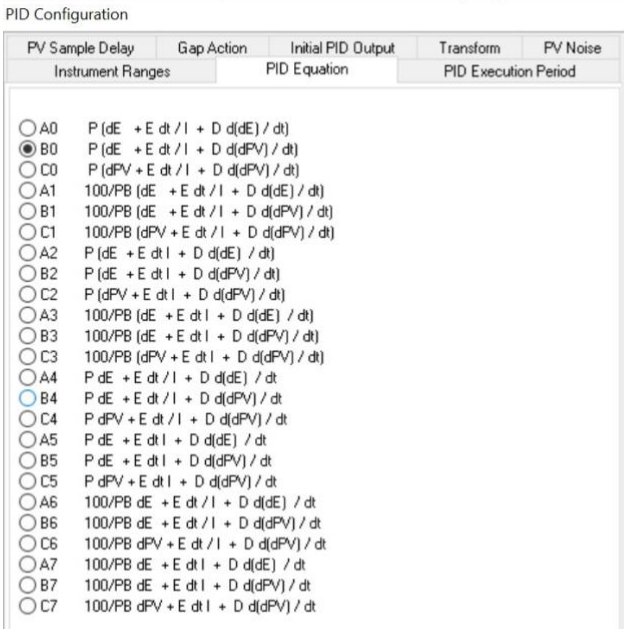
We can change the <u>simulation time</u> from 200 seconds to 1000 seconds, in order to make the simulation span longer. In this way we can see more PV, SP and OP data on the trend plot windows compared to before. Using Simulation Time and Simulation Speed controls, we can create a simulation to just precisely match our real process dynamics.

Step#12

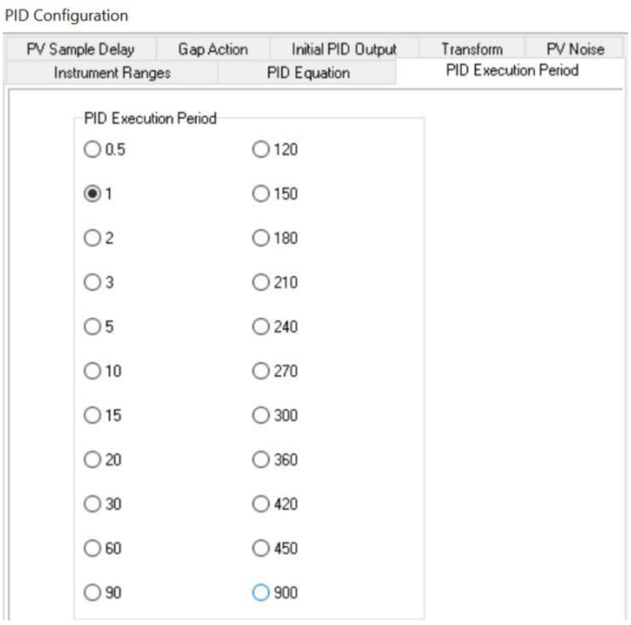
If we click on the PID Configuration icon, we will see all the typical PID configuration parameters.



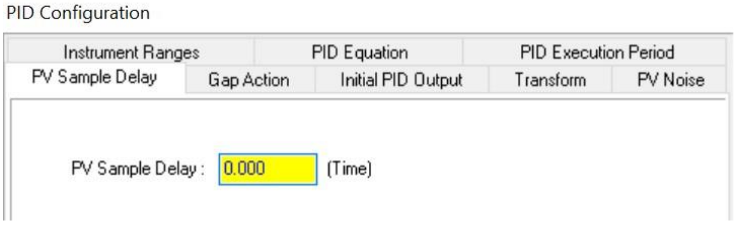
In this case we have a single flow controller (FC), so the CV range will be the flowmeter range and the MV range will be the valve range, which is normally 0 – 100%.



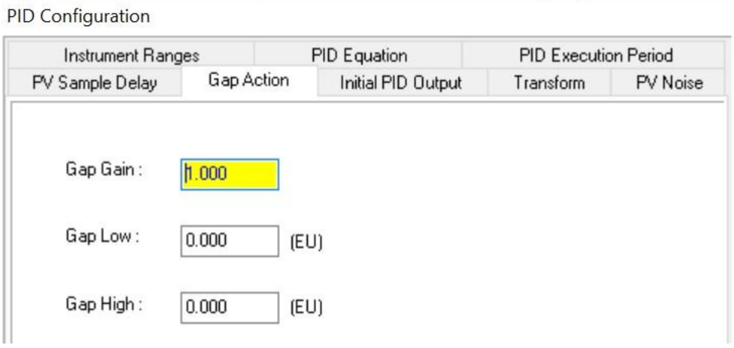
We have 24 different types of PID equations. The parameters P, I and D are the Proportional gain, Integral constant and Derivative constant. The term E is the error, defined as (PV – SP), the current deviation of the process variable from the setpoint. The term dt is the execution time (time scan) of the PID algorithm. The term dE represents the delta E, the change in the error between two consecutive PID executions. Equations of type B0 is one of the most common. In this equation the derivative acts on the deviation in the process variable, not the error.



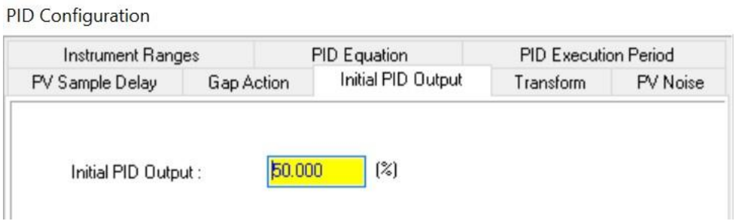
This parameter defines the execution frequency (scan time) of the PID controller. The default value is one time unit, so this means that the PID controller executes every one time unit (second, minute…) in the control system. We can select any of the other specified options as necessary for slower or faster scan rates.



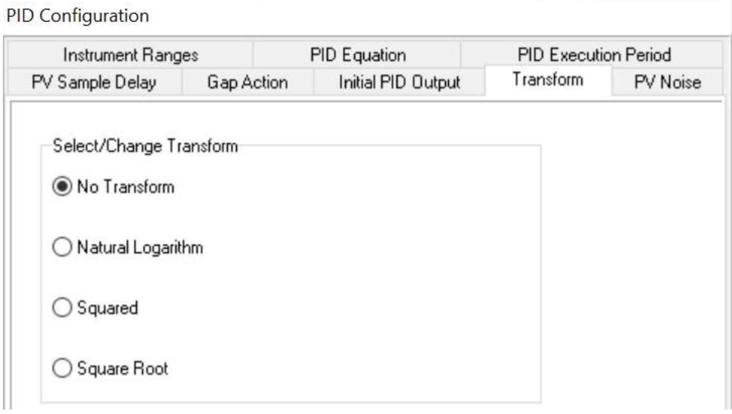
It is used to simulate a discontinuous signal.



The purpose of gap action is to dampen unnecessary PID control action when the deviation of the process variable from its setpoint is small. The amount of the dampening is specified by the gap gain and the amount of deviation by the gap low and gap high.

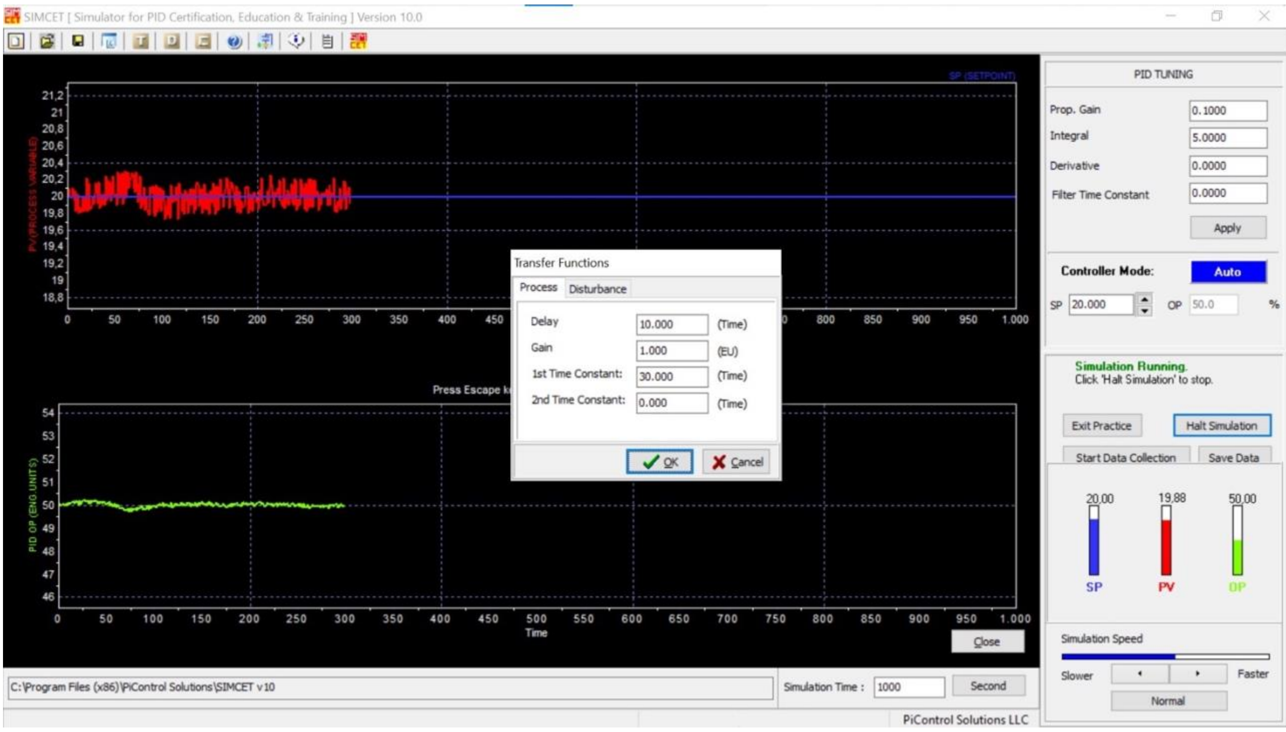


This is the PID output at the start time of the simulation (t=0). In this case the value is 50%, so It means that the valve is 50% open at the start time of the simulation.



Here we have four options. The transform applies to both the PV and SP before the PID error is calculated. With no transform, the raw PV and the raw SP are used to calculate the error. The other transforms are useful for linearizing nonlinear processes.

Step#13



If we click on the Transfer Function icon, we can see all the transfer function parameters and the disturbance transfer function parameters. We can notice that all the time constants are specified in the time unit (milliseconds, seconds or minute) as specified next the Simulation Time. In this case the unit of time is Second.

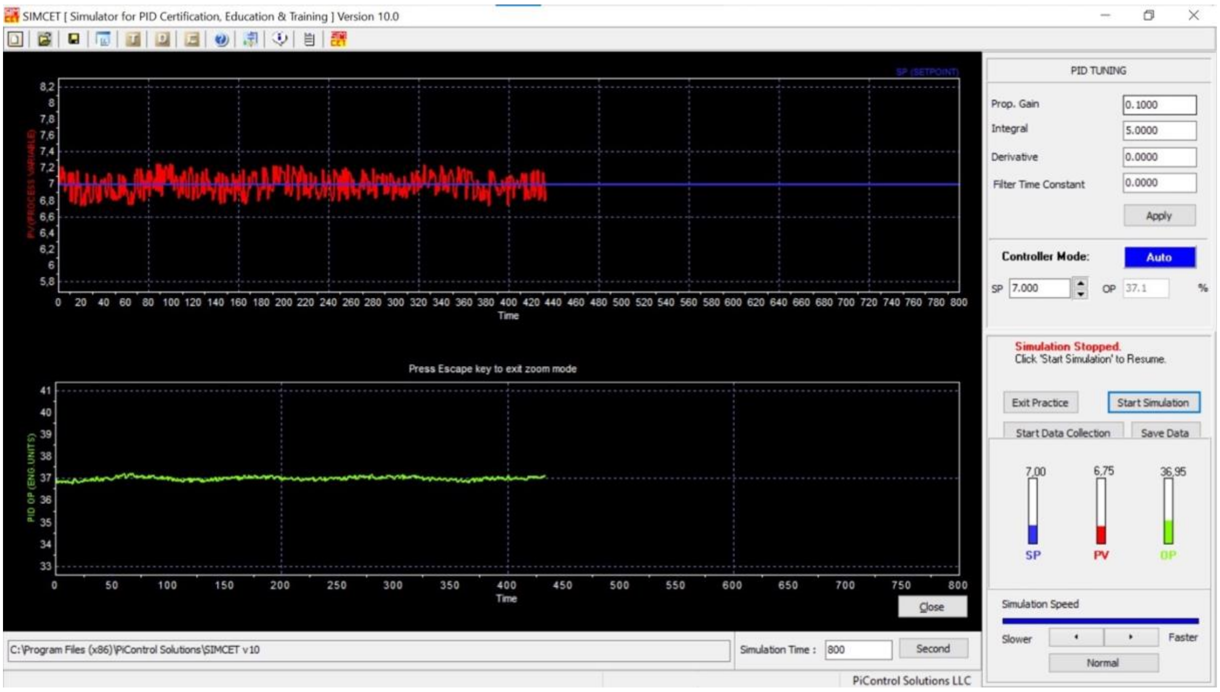
Step#14

At the top we have the top-bar menu icons.

Step#15

Now we try to improve the control action by adjusting the PID tuning parameters.

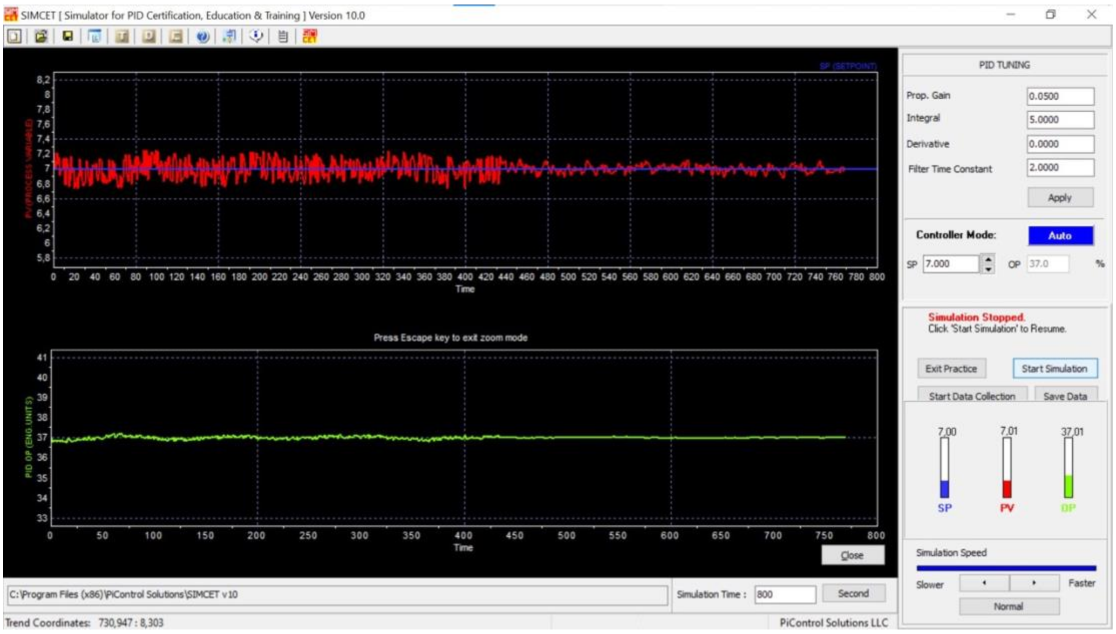
If we click on PID Configuration -> PID execution period, we can see that the default value is 1 second. This means that the PID controller executes every second in the control system. We can set the PID execution period as low as possible (0.5 seconds), so we ensure a faster scan rate.



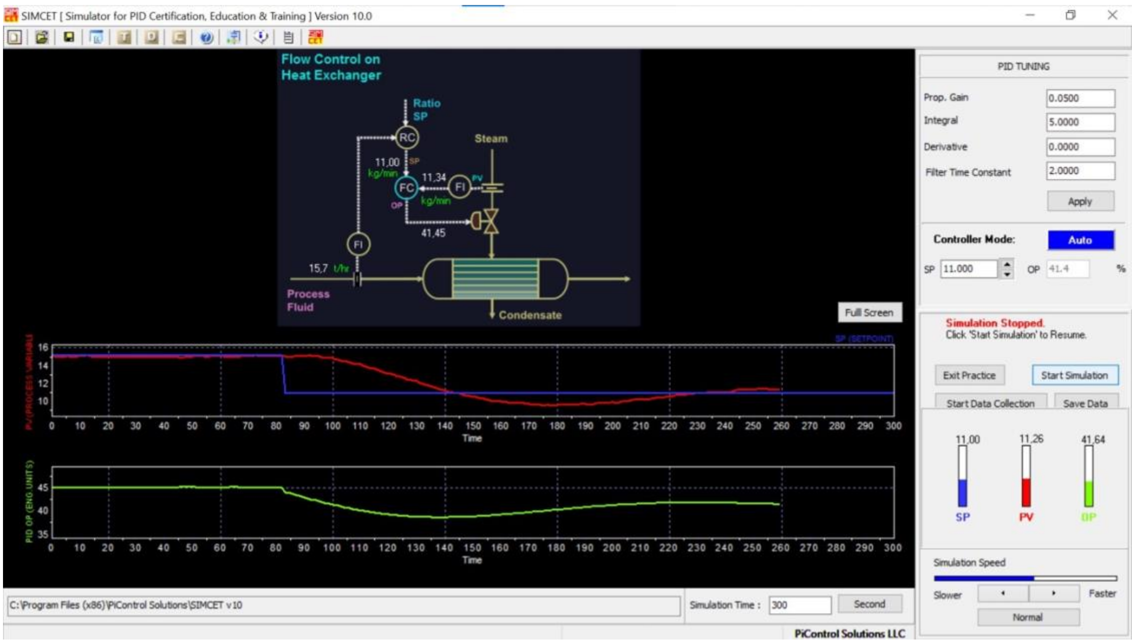
These trends correspond to the current values of proportional gain, integral, derivative and filter time constants.

- We do not increase the value of the proportional gain because we have seen that the proportional action becomes too high and the controller starts to oscillate, but if we decrease the value from 0.1 to 0.05, the green trend becomes a little more stable.
- We do not decrease the Integral value because the controller would start oscillating, but we also do not increase its value, because the trend does not vary significantly.
- We do not increase the Derivative value, because the controller output would become very jagged.
- We can increase the Filter time constant value to 2 and the red trend flattens out. We do not chose values that are higher than 2 because it would add some extra dead time.

With the changes introduced, the situation is as follows:



Then we can check the control quality by making some SP changes and then make iterative adjustments.



We can consider the parameters as optimal, because the process variable curve is quite precise.

Step#16

This software allows us to collect and save data, in order to create some data files that can be read into PiControl's PITOPS software for transfer function identification and PID tuning optimization. To use this function, before making the step tests, we click on Start Data Collection. When we are ready to save the file, we click on Stop Data Collection and on Save Data.