**Hot & Cold Water Mixing Process Simulator**

**A Process Control Challenge Problem**

R. Russell Rhinehart – 2018-09-20

[russ@r3eda.com](mailto:russ@r3eda.com), [rrr@okstate.edu](mailto:rrr@okstate.edu)

**Hot and Cold Mixing Overview**

As illustrated in Figure 1, the hot and cold water flow through valves to a common in-line mixing point, then flow in a common line to the exit. The 2x2 process control challenge is to adjust the signals to the hot and cold water valves to achieve desired values for the measured total flow rate and measured mixed fluid temperature.

Line 1 (Hot)



Mixing point

FCV1

ATO



ATO

FCV2



Line 3 (mixed)



Line 2 (cold)

Figure 1 – Process Illustration

Hot water enters through Line 1 from a header at pressure, P1, and height, h1. Cold water through Line 2, from a header at P2 and h2. Measured temperatures and flow rates of the inlet fluid could be transmitted to the control room as T1 and T2, and F1 and F2; but minimally only mixed temperature and total flow rate are measured. Flow control valves FCV1 and FCV2 are both modified-equal-percentage types with air-to-open (fail closed) actuators. The hot and cold fluid meet at the mixing point where they enter Line 3, and exit at P3 and h3. Mixed fluid temperature is measured by the sensor-transmitter (T3) at a distance LT downstream from the mixing point. Mixed flow rate and temperature are transmitted as F3 and T3.

I have coded this simulator in Excel VBA and posted it on my web site [1]. The code is open. You are free to use it, appropriate the code, or translate it to your preferred computing environment.

Although simple, this process offers several challenges characteristic of the CPI applications:

* It is nonlinear: 1) The equal-percentage valves have a nonlinear impact on flow rate, but the response is more linear than it would be using valves with a linear characteristic. 2) The pressure friction loss of the turbulent flow is a nonlinear (quadratic) response to flow rate. 3) The gain of how one valve affects temperature depends on the flow rates.
* It is interacting: 1) Changing either valve affects both temperature and flow rate. 2) Changing the flow rate of one fluid changes the mixed flow rate, which changes the exit line pressure drop, which impacts the inflow rate of the other fluid.
* The process has drifting disturbances: 1) The inlet fluid pressures and temperatures change in time causing disturbances. 2) The friction factors in the several pipe segments and the flow rate and temperature sensor calibration all drift in time. 3) The valves have sticktion.
* It is noisy: 1) On the flow rate measurements.
* It is nonstationary: 1) The transport delay (deadtime) on the temperature response changes with flow rate. 2) The drifts to the fluid flow pressure loss factors make the process response change in time.
* The measurements all have calibration faults: 1) At steady state, the material and energy balances don’t.
* It has constraints: 1) At a high flow rate set point, either high or low temperature set points cannot be simultaneously achieved; here, a controller needs to compromise on the two set points without windup.

Although this simulator includes many complications, the simulated process is relatively simple to understand (just material and energy balances, elementary fluid dynamics, and simple noise models) and simple to code.

**Elementary Deterministic Model**

The basics of the model are:

The fundamental mass and energy balances used to calculate mixed flow rate and temperature are:

(1)

(2)

This, of course, assumes constant density and specific heat of the fluid (water). Adding temperature dependence on fluid properties increases complication, but it is an inconsequential contribution to the control aspects. So, I choose to retain this simplistic model.

Since the impact of elevation is identical to that of pressure, all elevations are considered identical.

Since fluid acceleration has a much lower time-constant than the valve or temperature sensor responses, fluid acceleration () is not considered. Considering that the fluid acceleration as relatively instantaneous, the fluid flow models represent steady state. This simplifies the simulation, yet retains all of the control-related difficulties.

The steady state fluid flow equations represent mechanical energy balances on the fluid. Here, is the sum of all forces trying to accelerate, or decelerate, the fluid. Each force is a pressure (or pressure drop) times the associated area. Since the fluid acceleration is considered to be very fast relative to the speed with which valves change and temperature is sensed, the . For flow Paths 1 and 2, which both merge with Line 3:

(3)

(4)

In Equation (3) is the Line 1 entrance pressure driving the fluid, and is the exit pressure. Each line has a cross sectional area of . The terminology is similar for Line 2, Equation (4).

Here, the line losses, are calculated by a conventional friction factor relation, with equivalent-length accounting for the loss due to devices and fittings in the pipeline.

(5)

The simulator uses a Blasius model for the friction factor when the flow is turbulent

(6)

And, smoothly approaches the standard laminar flow model, , for transition and laminar conditions. If , then

(7)

Of course, is the Reynolds Number, .

The valve pressure drop usually needs a bit of explaining: First, the standard valve relation between flow rate and pressure drop is presented as:

(8)

where Pa and . (Note: Often the dimensional unifier, , is omitted from the equation, as is usually omitted from and from the conventional fluid mechanics equations. The industry uses a variety of temperatures as the standard temperature to assess specific gravity, within the range of 60 to 70 oF.) The fluid is water, and the temperature impact on density has a negligible effect on the simulator, so .

Valve Cv values are a measure of valve capacity – the flow rate of water (here m^3 /s) through the fully-opened valve when there is a unit pressure drop (here 1 Pa) is across the valve. Following conventional practice, the valve Cv is chosen so that the pressure drop across a fully-open valve is about 50% of the line losses. Values of in Equation (8) are the valve characteristics, which indicate how the flow changes with valve stem position, . Characteristic and stem position have the same range: , and . If the valve has a linear characteristic, then . However, a valve with a linear characteristic, when installed in a pipeline has a nonlinear gain, because the pressure drop across the valve changes with pressure losses in the line, which changes with flow rate. In practice, an equal-percentage characteristic is commonly chosen to help linearize the flow rate response to valve stem position of an installed valve where the line losses change with flow rate.

Rearranging the valve equation, it can determine valve pressure drop required in Equations (3) or (4), given flow and valve properties:

(9)

Equation (10) presents a standard model of an ideal equal-percentage valve. In it, is the valve rangeability.

(10)

Note: In the nominal equal-% mathematical model, does not have a zero value when , which means the valve would not close. So, this simulator uses a modified equal-% valve characteristic, which is representative of industrial practice. If the valve characteristic is modeled to be linear with .

(11)

Also note: Rangeability in the equal-percentage model of Equation (10) indicates the ideal flow rate ratio between a fully-opened and fully-closed valve, which is a bit different from the alternate use of the same term representing the ratio of maximum flow rate to the minimum controllable flow rate. So, take caution when using the term.

It would be a rare confluence of conditions to have identical velocities in each of the three lines. At the mixing point fluid flow rate likely changes, due to the flow combination and/or pipe area. So, there is an associated pressure jump with the velocity change. Using a plug flow model, the Bernoulli effect is:

(12)

Valves are considered to have air operated actuators, and their stem position is modeled as a first-order response to controller output changes. Valve time-constants are  The valve stem ODE’s are:

(13)

Where O represents the 0 to 100% output (MV or manipulated variable) of the controller.

Note: This valve model represents ideal, perfect, calibrations in the sequence of final element devices. Ideally, 0 to 100% is translated by 4-20 mA (electric current) and 3-15 psi (pneumatic) signals. For instance, ideally, when the controller output is 0%, the signal transmitted to the i/p device is 4 mA, the i/p output is 3 psig, and the valve actuator spring and diaphragm pressure perfectly balance and hold the stem at exactly closed. But I have never been able to get this sequence to match properly, and commonly find, for instance, that the valve does not begin to open until the controller output is at 7% or that the valve is fully open when the controller output is 90%. Some controllers have limits of -6% and 106% to overcome calibration inaccuracy in the final element sequence. You could adjust the valve model to represent a non-ideal calibration situation. You also could use alternate time-constants for opening or closing the valve representing pneumatic filling and venting mechanisms. Again, I feel such a step toward greater rigor in the realism provides complexity that exceeds control relevance benefit. At the same time, including such transducer device calibration errors in the simulator would provide user understanding of the sequence of events that makes the final element change, which is usually missing in control instruction.

Nominally, the controller would send signals to the valves. Equation (13) would determine valve stem position, then Equation (11) the valve characteristic. If the Line 1 and 2 flow rates were known Equation (1) would reveal the Line 3 flow rate. Then Equations (5-12) would determine the pressure drops for each item. If the flow rates are correct, then Equations (3) and (4) will balance. I did not find a convenient arrangement of the nonlinear equations to explicitly solve for and , so I used a root-finding version of optimization. See Section 7. Once and values are obtained, Equation (2) calculates the mixed fluid temperature.

However, the mixed fluid temperature is measured downstream of the mix point. So, the mixed fluid T is stored in a vector so that the fluid T at the downstream sensor is based on the plug-flow transport-delayed value. The transport delay is flow rate dependent; and, if ideal plug flow . Accordingly, the temperature array item selected to represent the delayed T will change with flow rate. The temperature sensors are modeled as third-order, which provides an approximation to both thermowell sensor dynamics and the not-plug-flow in-line fluid mixing.

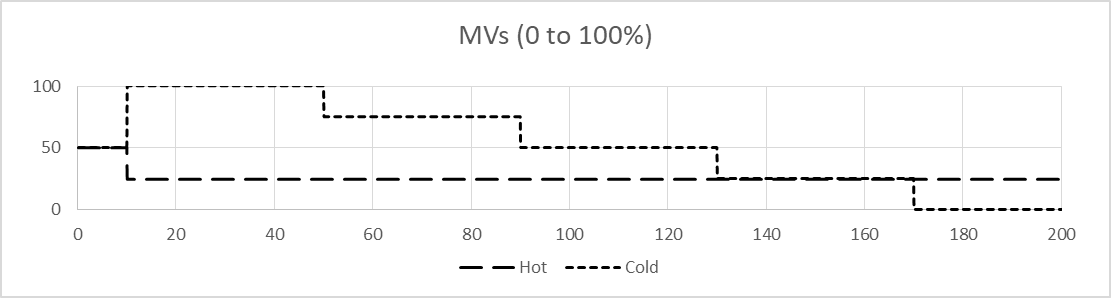
In a basic 2x2 controller simulation with two SISO controllers, the measured and values would be given to the controllers, compared to set points, and the controllers would calculate desired MV values, and , which would be sent to the valves.

The two SISO controllers, however are interactive. If the flow controller adjusts its valve to change flow rate, this upsets the mixed temperature. Then the temperature controller will adjust its valve, which upsets the mixed flow rate. A classic advanced regulatory solution to eliminate interaction is to use ratio control. Alternately, double ratio, feedforward, or decouplers could be used to fix the interaction. Also alternately, a model-based controller could be used to compensate for the interaction. In any case, if a controller strategy needs either additional devices or measurements of the , , , and/or values, its adds installed equipment cost and maintenance expenses, and the proposer of an advanced strategy should not expect it all to be free.

**Deterministic Simulator Responses**

This section is termed “deterministic” because it represents the basic process features with no noise, disturbances, or changes in coefficient values. In the following figures, the upper graph indicates the signals from the controller to the valves, and the lower graph shows the three flow rates and mixed temperature response.

Figure 2 reveals the effects of changing the signal to the cold water valve while keeping the signal to the hot water valve at a fixed value. Note several aspects: 1) Each drop in the cold flow rate and the total flow rate is fairly identical, because the equal-percentage valves substantially linearize the response. 2) The hot water flow rate increases with each drop in cold flow because the back pressure from the mixed flow in Line 3 is reduced. This shows one form of interaction. 3) The mixed fluid temperature shows a nonlinear response. Although the signal makes identical 25% changes, the temperature rise in the 60 s period is about 5 oC, but the temperature rise in the 190 s period is about 35 oC, about seven times greater. 4) Finally, when the total flow rate is high, in the 15 s period, the delay for the temperature change is about 2 s, but when the total flow rate is low, in the 190 s period, the delay is about 10 s, a five-to-one change. When the hot valve is also at a lower position this delay can be much larger. From maximum to a very low flow rate the temperature delay can range from nearly zero to over 20s.



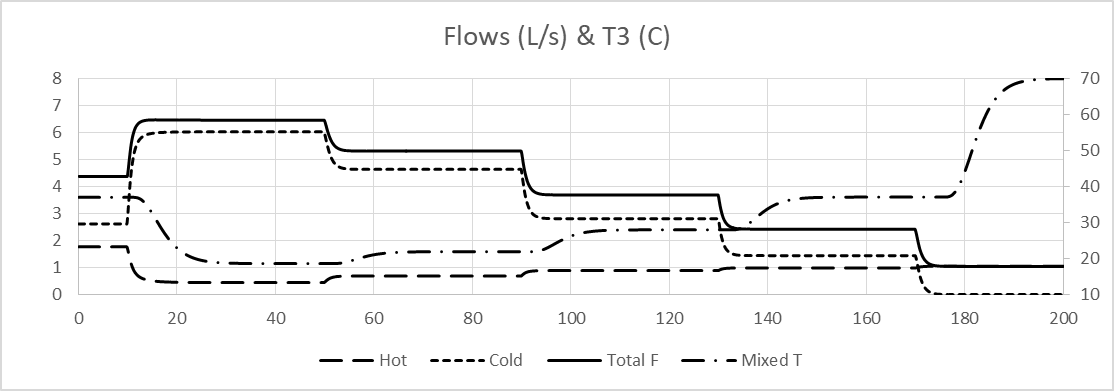
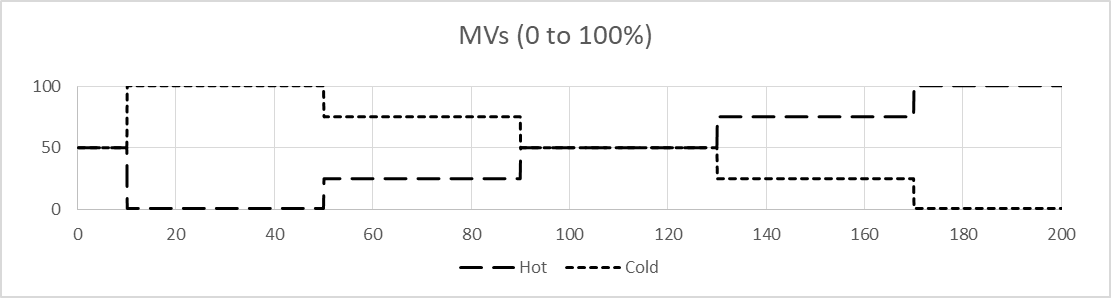


Figure 2 – Deterministic Response to Steps in One MV

Alternate phenomena are revealed in Figure 3. Here the MV values start at the extremes of 0 and 100% then simultaneously make 25% increments to the other extreme. If everything in Line 1 and 2 was identical, then the total flow rate would remain constant, but it drops with each initial change, then rises with the last change. Further, after the changes at 90 s and 130 s the total flow rate first drops then rises. This is because the valve dynamics, valve Cv values, and the Line 1 and 2 properties are not identical.



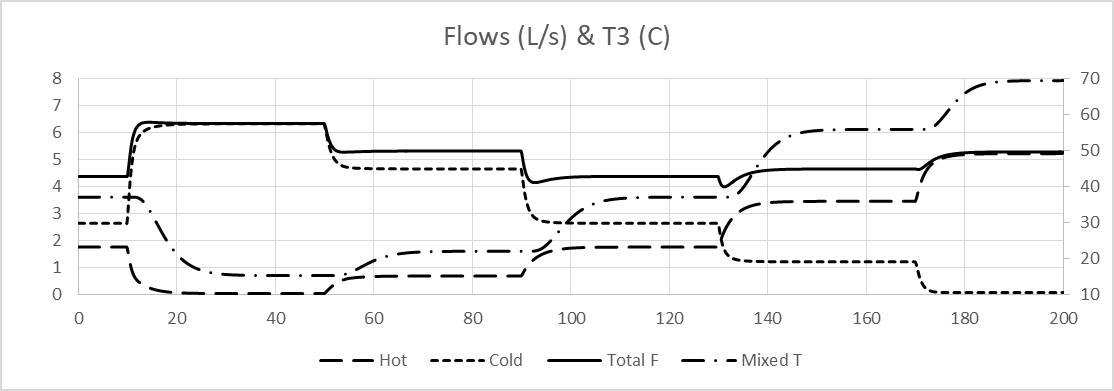


Figure 3 – Deterministic Response to Coordinated Steps in Both MVs

**Environmental Issues**

This section discusses several aspects that add realism, and non-deterministic (stochastic) aspects to the simulated process. From my experience, these are as relevant to control as the deterministic process features.

All transmitters lie. The true process value is biased by calibration drift and noise (modeled here as a uniform random perturbation on a first-order response, an ARMA(1,1) driven by a uniformly distributed random variate). The model is

(14)

Where represents the bias, is the sampling interval, is a uniformly distributed variate, , and alpha and beta are scaling factors to select persistence of past events and amplitude of the bias. The range of the drift value and the noise level on it depend on values of the alpha and beta factors. In this model, noise sigma, the sample-to-sample variation, is [2]. The drift value, however, retains an alpha-weighted portion of past perturbations, and makes a “random walk” path about a nominal value of zero. The standard deviation of the bias, , over a long period is , requiring for stability [2]. More rigorous models might be grounded in a Gaussian influence or higher-order or nonlinear ARMA models, but again the benefit of such rigor in simulating a control challenge problem is not as great as the complexity it adds.

Using and one realization of the deviation form Equation (14) is shown in Figure 4. The pattern will change with each randomization of the pseudo-random variate, . Any one particular outcome sequence is termed a realization, a possible outcome.

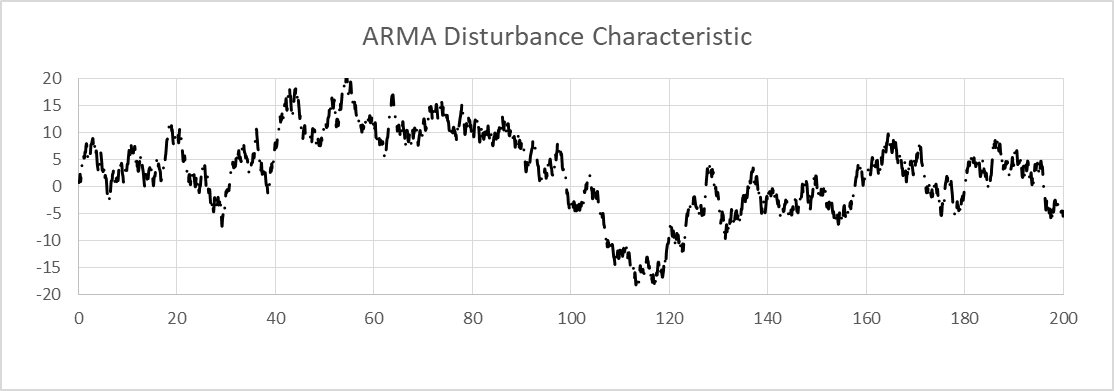


Figure 4 – A Disturbance Realization

Although the simulator will calculate a value of the and responses from the fundamental relations, these cannot be used for control. The simulator needs to use what might be measured. All measurements have similar models. Here are the two key ones.

(15)

(16)

Those are normal operating conditions. Other sensor failure or fault conditions are not included in the simulator, but could be if the user wishes to investigate such.

Inlet and exit pressures, P1, P2, and P3 vary in time. (You may have experienced flow rate changes in a home faucet when other users or appliances use water.) These are also modeled as first-order drifts driven by uniform random variate. The drifting values make the flow rates vary, even with no change in the signals to the control valves. Inlet temperatures T1 and T2 also vary in time with similar ARMA(1,1) models. (Again, you may have experienced home hot or cold water temperatures changes as the use changes the inflow to water in the outside lines or dilutes the water heater tank.) Finally, the friction factor in each of the three lines, f1, f2, and f3 independently drift in time. (Industrially, silt, corrosion, entrained air, and many features change in time and affect friction factor in a line. Additionally, operator adjustment of manual valves, flow paths, and plugging of screening all impact equivalent length.) Since the terms are combined as a product in the equations, I chose to keep equivalent lengths fixed and attribute all such disturbances to the friction factor value. These are all normal environmental influences. The general model is:

(17)

(18)

Where x represents any of the nonstationary model coefficients or disturbances. The and coefficient values are unique to each variable, and although the resulting values change in time, the disturbance model is stationary. The alpha and beta coefficient values do not change with simulated time in this challenge problem. The random driver, , is independent for each variable and time interval. Other environmental influences could be included, such as pressure pulses (perhaps from water hammer effects on utility lines). Additionally faults could be included, such as plugged tap lines between orifice and dP cell, or a mechanical fault with a thermowell.

Since the inflow temperatures are allowed to drift in time, this creates two additional simulator aspects. First, the and sensors are considered to be in a thermowell, each with a third-order lag response, and with a measurement bias using Equations (17) and (18). The measurement lags behind the actual temperature. Second, the sensors “feel” the temperature upstream of the mix point, so the simulator uses two more arrays to delay the actual (not measured) temperature used in Equation (2) to calculate the mixed fluid T.

Noise is added to each flow measurement proportional to the square root of the flow rate, which simulates turbulence effects on an orifice device, and the measurement is filtered to temper the fluctuations.

Figure 5 reveals the impact of the environmental aspects when the signal to the valves is unchanged from the nominal initial 50% values. This is one realization. Since the random number generator is randomized with each trial, every trial is its own realization. However, the trial-to-trial influences have similar impact.

Note: The mixed temperature drifts from about 30 to 40 oC, has a smooth trace, and takes about 20 s to move from one value to another. By contrast, the flow rate has a noisy character (even with filtering) and has only a few seconds persistence in any value.

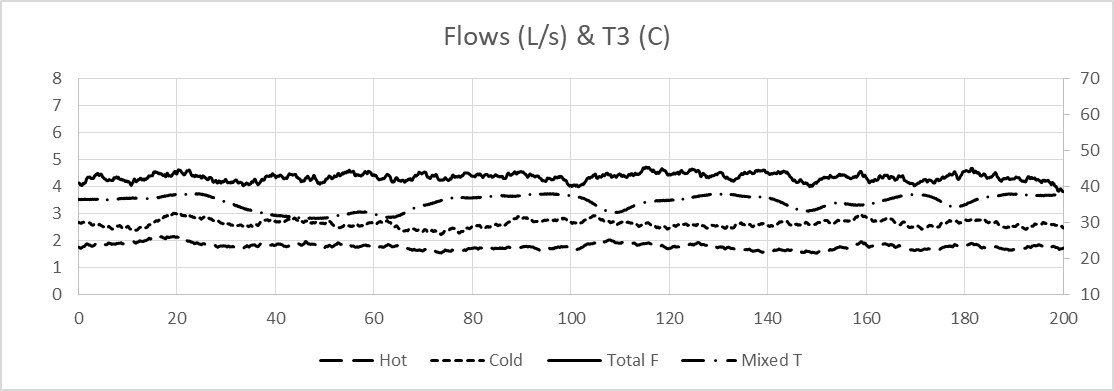
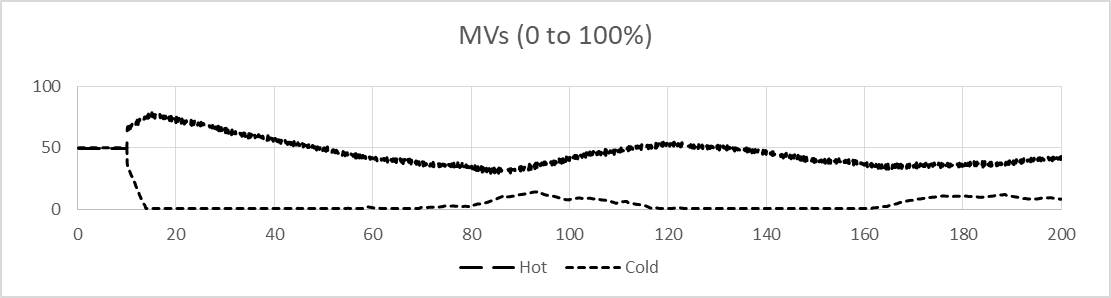


Figure 5 – A Realization with No Changes to the Valve Positions

Finally, completing the environmental afflictions, sticktion is added to the valve stem position. Valve stems enter the valve body through a hole that is filled with packing to prevent leakage of process fluid around the stem. The packing is fairly tight, causing static friction to resist actuator pressure in moving the stem. Often this grabbing feature is exacerbated by corrosion on the stem, or defects in any of the moving parts of the stem or actuator. When actuator air pressure overcomes the spring back force and the static friction, then the stem jumps to a new position. Equation (13) models the ideal valve position, but with sticktion the stem only changes if the position difference exceeds a deadband; and, if so, the stem jumps to a random position between the current location and the ideal target.

Figure 6 reveals the impact of sticktion. Here two independent SISO controllers are used – PID for mixed fluid temperature, and PI for total flow rate. The temperature controller sends the signal to the hot water valve, and the total flow rate controller manipulates the cold water valve. Characteristic of sticktion is the cold flow rate response in the 80 to 100 s period and 170 to 200 s period. Note: In the upper graph, the signal to the cold valve makes a steady ramp up then down, but in the lower graph the actual flow rate holds a past value then makes abrupt steps up or down when the stem jumps to a new position.



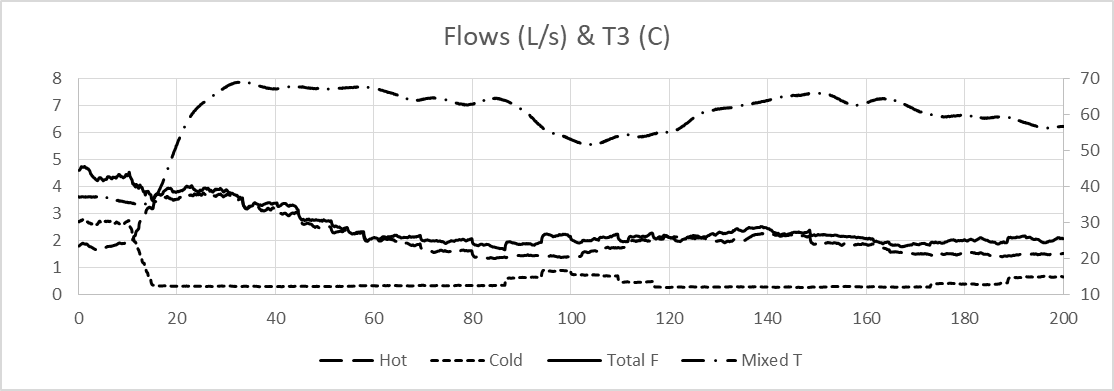


Figure 6 - Sticktion

Overall, the input-output sequence for the simulator is: Send a signal to a valve, this is the controller output (CO), or manipulated variable (MV), and is often labeled with the symbols or . My simulator uses the symbol O. This causes the stem position to change in a first-order manner, conflicted by sticktion. The fluid flow rate is determined by the mechanical energy balance of the two fluid paths; but at each sampling, new drift values for model coefficients lead to different flow rate solutions. The mixed T is calculated from the thermal energy balance, and it is also subject to the drifts on the inlet temperatures. The inlet temperatures are sensed at upstream sensors. The actual values need to be transported to the mix point, and the measured values lagged. The simulator stores the inflow temperatures in an array, which considers the in-line transport to be plug flow with no axial mixing. The mixed T is also stored in a vector so that the fluid T at the down-stream sensor is based on the plug-flow transport-delayed value. The temperature sensors are modeled as third-order, which provides an approximation to both sensor dynamics and in-line fluid mixing. Finally, any measurement represents the true simulator value corrupted by measurement drift. Flow rate measurements are filtered to temper noise.

**Coefficient Values**

System dimensions and nominal values for variables are listed in Table 1 (in SI units):

Table 1 – deterministic model coefficient values

|  |  |  |
| --- | --- | --- |
| D1 = D2 = D3 = 0.06 m | P1 = 500 kPa |  |
|  | P2 = 550 kPa |  |
|  | P3 = 150 kPa |  |
| Deadband = 0.05 | LT1 = 2 m |  |
| Lequ1 = 150 m | 1 L1 = 50 m | T1 in = 70°C |
| Lequ2 = 75 m | L2 = 25 m | T2 in = 15°C |
| Lequ­­­­­­­­3 = 400 m | L3 = 100 m | ρ = 1000 kg/m3 |
| Cv1 = 2 x 10-5 m3/s @ 1 Pa |  | μ = 0.001 Pa ⋅ s |
| Cv2 = 2.5 x 10-5 m3/s @ 1 Pa |  | gc = 1 kg-m/N-s2 |
| LT2 = 1 m |  | G=1 (specific gravity) |
| LT3 = 1 m |  |  |

The fluid is water, and the temperature impact on density has a negligible effect on the simulator, so , ρ = 1000 kg/m3, and µ = 0.001 Pa-s.

The internal diameter of all three pipes is 6 cm, but these could be assigned unique values. The pipe lengths are 50, 25, and 100 m and the distance between the mix point and the T3 sensor is 1 m. Lequi are the equivalent length of Lines 1, 2, and 3 due to devices and fittings.

Noise and drift model coefficients are listed in Table 2.

Table 2 – stochastic model coefficient values

|  |  |  |
| --- | --- | --- |
| deadband = 0.05 (5%) | TSensor: α=0.99, β=0.2 |  |
| f: α=0.98, β=0.002 | Tinlet: α=0.99, β=2 |  |
| P: α=0.98, β=5,000 | Fsensor: α=0.99, β=0.02\*F |  |

**Solving for the Flow Rates**

Since the flow rate Equations (3) and (4) are nonlinear and interactive, an iterative approach is needed for the solution. I explored several arrangements of both successive substitution and Newton’s method as a root-finding method. These worked; but, I choose optimization because the root-finding techniques needed tempering to prevent the solution from jumping to excessive values, and finding a better rearrangement of the equations was not as easy for me as using optimization. I coded my simulator using a heuristic cyclic method [3] because it is both simple and robust to nonlinearities and discontinuities; but since the fluid flow solution is independent of the algorithm to obtain the solution to Equations (3) and (4), you could use any algorithm of your choice. Cauchy’s Sequential Line Search (CSLS), Incremental Steepest Descent (ISD), Levenberg-Marquardt (LM), Hooke-Jeeves (HJ), and Leapfrogging (LF) all work [3].

The objective is to minimize the squared error of Equations (3) and (4). The convergence on the solution could be based on incremental changes to the decision variables (DVs), which are the two flow rates. Alternately, convergence could be based on the objective function, or on both of its two elements. With any criterion, I chose the threshold on the solution to be an order of magnitude smaller than the threshold that permits visually detectable changes in the deterministic solution.

The OF has a single minimum. In assessing the optimizer work by number of function evaluations (NOFE) to converge with equivalent precision, Hooke-Jeeves used an average of 79, and the cyclic heuristic direct 82 NOFE. These were essentially tied for computational work to solve the equations, and had lower NOFE values than ISD (100), CSLS (210), LM (600), or LF (1,200). I chose CHD over HJ because it has equivalent NOFE and is a bit simpler to code and understand.

The cyclic heuristic direct (CHD) search starts with trial solutions (initial F1 and F2 values) that are the same as the most recent past values. If things are unchanged, these are the solutions, if things change somewhat they are near to the new solutions. This is the base case, and the OF is evaluated at the base case. One at a time, a trial solution value is perturbed from the base value, keeping the other at the base value, and the OF is evaluated. If the OF is better, then the new trial solution and OF are accepted as the new base values. This represents a step in the right direction and the perturbation for the next trial is increased by 20%. Alternately, if the new OF is worse, or if a constraint is violated, the perturbation is reversed in sign and halved. This is repeated for each DV, one at a time. When each have been perturbed once, the cycle repeats. Being a direct search technique (only using OF values, not gradients or surface models) this method is robust to many surface aberrations, and simple to implement and understand.

**A Control Response**

Figure 7 shows one realization of a simple model-based control run. The controller uses simple steady state models and presumes the process has first-order responses. The flow rate models in the controller are identical and linear. There is no feedforward of inlet temperatures or cascade to actual inlet flow rates. There is no model adjustment from on-line data. The flow set point is 5 L/s and the temperature set point is 60 oC. Note: In the upper graph, the signal to the hot water valve occasionally is limited by the upper value of 100% (hitting a constraint). Note: In the lower graph, the small steps in the cold flow rate at a time of 55 s and in the 100-120 s period, are revealing the impact of sticktion.



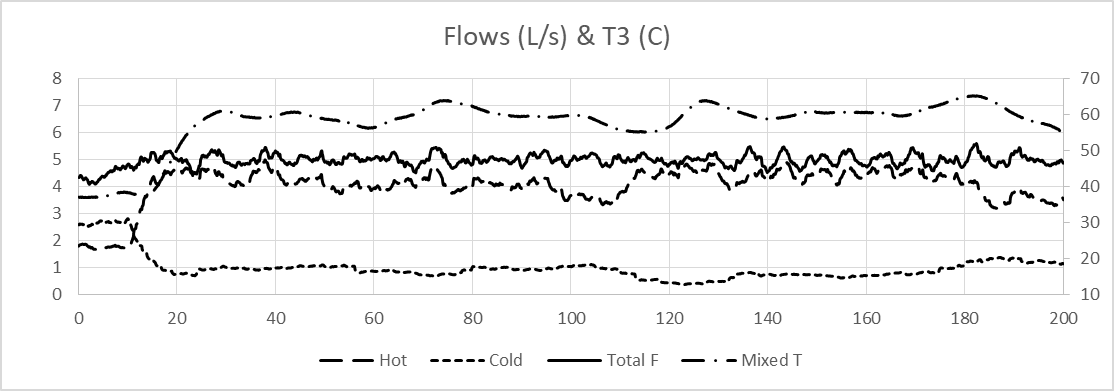


Figure 7 – A Control Trial with an Initial Set Point Change Followed by a Regulatory Period

**Discussion and Comments**

For convenience, fluid properties (density, viscosity, specific heat) are assumed independent of temperature and pressure at the values indicated. Including temperature dependence has an insignificant impact relative to other influences, and the complexity adds no benefit.

I explored several models to account for the transport delay and in-line dispersion of the fluid temperature. These included finite-difference PDE modeling of the fluid transport, and a more simple many mixers in series model with mixer time-constants dependent on the fluid flow rate to represent in-mixer residence time. These worked, and although they are more rigorous than the plug flow transport delay, they are still subject to idealizations that could be challenged. The simpler model still has the several transport delays dependent on the flow rates, and uses high-order lags for the temperature sensors and inline dispersion.

A standard array method to model transport would be to increment each value down one element then to insert the new value in the top of the array. However, a simpler method is to use one pointer (index value) to indicate where to input a new value, and another to indicate where to read a past value. The pointers are incremented each sampling and reset to 1 when they exceed the array size. The read pointer value is calculated from the delay and the put pointer.

If toggled “ON”, the simulator adds drifts to the pressure driving forces, drifts to the inlet temperatures, drifts to the friction loss coefficients, sticktion to the valves, bias to all measurements, and noise to all flow measurements. Set enviro=1 to create the confounding effects, and set enviro=0 to turn them off.

The simulation interval for the process is 1/10th of that for the data acquisition and control system sampling. I used a simple Euler’s (first-order explicit) solution to all of the ODEs and chose this simulation time step to be small enough so that the deterministic solution was not visually dependent on the time interval. The control and measurement is on a 10 Hz (0.1 s) interval, a characteristic (but relatively fast) scan rate for industrial flow rate and temperature control.

The default output includes F1 and F2. However, it is not normal to measure such extraneous variables. Normally, the process owners would only install devices to measure the essential T3 and F3 values. So, if you plan on using F1 and F2 in your controller, include a penalty for the cost (installation and maintenance) of the extra measurements.

**User Manual**

The Excel VBA file is named “Hot & Cold Mixing 2018-09-17.xlsm”. On Sheet 1 is data input and output. The graphs show the MV and process responses for a 200 sec simulated period. The yellow highlighted cells are for user input. Enter a 1 or a 0 in Cell(1,10) to turn the environmental effects on or off. Choose a test case by entering the number in Cell(2,10), and select either the conventional controllers or the simple Model-Based controller by entering PID or MBC in Cell(3,10). Press the run button to observe the result. If the environmental effects are off (a deterministic situation), the simulator just runs once. If the environmental effects are on (the stochastic situation), the simulator repeats the trials 100 times.

Cell(1,11) reveals the trial number. The first 4 rows of Column 17 indicate the goodness of control metrics for the most recent trial – nISE for T and F and Travel for O1 and O2. The units are oC2 and L/s2 and % and %. The first 4 rows of Column 18 indicate the average of the goodness of control metrics for the trials run. The goodness of control metrics are only calculated during the period that the controllers are in the AUTO mode.

Open the VBA editor (press CTRL-ALT-F11) to see or edit the code. The Main Subroutine runs the simulator for the 200 sec of simulated time. At each simulated time interval, the key calls are

Call Events – this controls what the user desires such as MAN-AUTO transitions or set points.

Call Process – this calls the process simulator to see the response.

Call Measure – this obtains measurement reported values from the process.

Call Filter – this filters the measurement values

If Control\_Choice = "PID" Then Call PID\_Control – this is the SISO PID controllers

If Control\_Choice = "MBC" Then Call MBC\_Control – this is the simple model-based controller

Call Evaluate – this accumulates the goodness of control metrics

Call Data\_Out – this outputs results to the Excel Sheet 1 for display.

Feel free to change the simulated time duration. If you lengthen it, you’ll need to adjust the range used for the Sheet 1 graphs.

The Events Subroutine schedules the events. Feel free to edit these as you wish. Presently testcase = 0 means do nothing. The controllers remain in MAN with the outputs each 50%. If you chose testcase 0 and environmental effects on, you can see the open loop variation in T3 and F3 due to environmental influences. Testcases 1 through 7 keep the controllers in MAN and perform several step tests on the MVs. These may be useful in generating data to tune the controllers. Testcases 8-12 place the controllers in AUTO and reveal their ability to move the process to desired set points. Testcases 13 and 14 keep one controller in MAN and the other in AUTO, which permits tuning one PID controller at a time.

**References**

[1] [www.r3eda.com](http://www.r3eda.com), “Hot and Cold Mixing Simulator”

[2] Bethea, R. M. and R. R. Rhinehart, Applied Engineering Statistics, Chapman & Hall/CRC, Boca Raton, FL, July 1991. ISBN 0-8247-8503-7. 401 pages.

[3] Rhinehart, R. R., Engineering Optimization: Applications, Methods, and Analysis, 2018, John Wiley & Sons, ISBN-13: 978-1118936337, ISBN-10:1118936337, 776 pages with companion web site [www.r3eda.com](http://www.r3eda.com).