ET401 Lab  Self-Regulating Single Tank Level Control: Modeling and Tuning

Profs Sarnacki, Owen, Olivari, Tefft  Fall 2019

Background

Recall the generalized control system block diagram and Simulink controller you used last week as shown in Figure 1 and 2. So far you have run the system in open-loop and have demonstrated that the open-loop system is inherently stable. There is a risk that by closing the loop you can make it unstable if you design the controller poorly or guess and check your controller tuning on the fly with the system running. Typically in any industrial environment, you cannot run the risk of creating an unstable process or operating your system far from a desired setpoint. Serious money is at risk. Instead, it is standard practice to use system modeling to determine and validate proper controller tuning.

![Figure 1: Tank Control System Block Diagram](image1)

![Figure 2: Simulink Tank Controller](image2)
The System Model for Controller Tuning

We will start by creating an open-loop model of the tank system in Simulink to recreate our real step response results. Note: The step response tests were a deviation from a steady state. Our model needs to be built the same way.

To begin creating the open-loop model:

1. Open MATLAB then select the Simulink button on the Home tab to start Simulink.
2. Create a blank Simulink model.
3. Save your model as TopTankModelName1Name2 where Name1 and Name2 are your last names.
4. Add a step input block to the Simulink model. Double click the block and set the final value parameter to match a controller output (CO) step (change in CO) you used for one of your step response tests.
5. Next, just like your real system, the change in controller output is passed to your actuator, resulting in a change in pump flow to the tank. Create a Simulink gain block that results in a change in flow in LPM from a change in CO. Hint: This is based on your calibration results on step 2 of the last lab.
6. Use the bump test data to build a transfer function of the tank based on what you know about the process. The type of process will tell you what the transfer function needs to be. Note: Your input to the plant is change in flow and your output is change in level. Choose the correct gain based on your input and output to the plant transfer function block.
7. Create a sum block between your actuator and plant blocks. Double click on the sum block and change the second plus sign (+) to a minus sign (-). Add a constant block to the model and connect it to the minus sign on the sum block. You can leave the constant block as a zero for now. This will represent your disturbance test where you opened the disturbance valve after the tank had reached steady state, resulting in a lower steady state tank level. The constant block is the flow in LPM leaving the tank due to the disturbance valve. The disturbance flow is subtracted from the total flow entering the tank.
8. Place a scope block at the end of the model. Double click the block to open the plot window. Click on the Configuration Properties (gear) icon at the top of the plot window and change the number of input ports to 3 on the Main tab. On the Display tab check the Show legend check box. Click OK then connect the controller output, level, and flow to the scope block.
9. Double click on the lines connected to your scope block and label them CO, Level, and Flow so you can distinguish them on the scope plot.

Verify your open-loop model with an instructor: ____________________________

10. Choose a simulation time that will capture the time it took to run a step response test. Click the run button and observe the plot window from the scope block.

Verify your step response plot with an instructor: ____________________________

11. Validate your model with a disturbance test. Step 6 from the last lab included the Flow, level change, and time taken for the flow disturbance to reach a new tank level equilibrium. According to your average step response test gain and level change for the disturbance test, calculate the change in flow exiting the tank
valve. You recorded the flow entering the tank for the disturbance test. Calculate the disturbance flow as: 
(disturbance flow) = (flow entering tank)-(change in flow exiting tank valve). Add a disturbance flow to your model and run a disturbance test. Click the run button and observe the plot window from the scope block.

Verify disturbance plot with an instructor: ________________________________

12. Be sure to save your model when complete.

To finish the model, we need to make the model closed-loop:

1. Select Save As and save the model as TopTankModelwContName1Name2 where Name1 and Name2 are your last names.

2. Add a feedback, sum block, and PID controller to your model like that used in the Simulink controller of Figure 2.

Verify your closed-loop model with an instructor: ________________________________

3. Be sure to save your model when complete.

**Ziegler-Nichols Tuning Rule:** *It's a trap! - Admiral Akbar*

This obviously won't work on a first order system as it won't oscillate, but I think this is a useful learning experience. I actually attempted to use this to tune a system with a first order response before I knew what I was doing and it wouldn’t work. This led me to the direct synthesis method below.

Ziegler-Nichols is a common tuning method used to tune PID controllers. Use the model you just created to test the method given as follows:

1. Set Kd and Ki to zero and increase your proportional gain until you achieve a continuous steady oscillation of your second tank level. This gain is called the ultimate gain Ku.

2. If you can’t get a steady oscillation, reduce the proportional gain until you observe a ¼ cycle decay (each peak is ¼ the size of the previous peak). This is called the quarter cycle gain Kq. The ultimate gain is calculated as Ku=2Kq.

3. Measure the period between peaks. This is the ultimate period of oscillation Pu.

4. For a P only controller: Kc=0.5Ku. For a PI controller: Kc=0.45Ku, Ti=Pu/1.2. For a PID controller: Kc=0.6Ku, Ti=Pu/2, Td=Pu/8.

Discuss your findings with an instructor: ________________________________

**Direct Synthesis Tuning Rule**

Direct synthesis is a powerful PID tuning method based on system modeling (like you just did) and a lot of math. We won’t get into the math but instead use a tuning rule based on the full approach. Note the PID controller is a
parallel type with proportional, integral, and derivative actions on separate parallel paths. Proportional, integral, and derivative gains are given as $K_c$, $K_i$, and $K_d$. This tuning rule assumes the open-loop response of the system is first order. Is this the case? If not, the tuning rule may not work well. If the first order assumption is acceptable then the rule states:

$$K_c = \frac{1}{K_p \tau_{ratio}}$$

$$T_i = \tau$$

$$T_d = 0$$

Where $K_p$ is the average process gain calculated from your step response tests in cm/%, $\tau_{ratio}$ is a variable for specifying controller response speed (1=fast, 4=slow) chosen by you, and $\tau$ is the average process time constant calculated from your step response tests in seconds.

Verify $K_c$, $T_i$, and $T_d$ with an instructor: ____________________________

1. Convert $T_i$ and $T_d$ (standard PID algorithm) to $K_i$ and $K_d$ (parallel PID algorithm)

Verify $K_i$ and $K_d$ with an instructor: ____________________________

2. Enter the values of $K_c$, $K_i$, and $K_d$ into the controller of the closed-loop model. Run the model and examine the scope output. Does the controller guide the process to a stable equilibrium?

   Verify your model step change plot with an instructor: ____________________________

3. Test your tuned gains $K_c$, $K_i$, and $K_d$ on the real system. Start the controller in manual mode and adjust CO until the tank reaches a steady state near 50% level.

4. Change the Auto Level Setpoint to match the observed steady state level while in manual (open-loop) mode.

5. Switch the controller to auto mode, allowing the PID controller to control the process. Is the level maintained at the Auto Level Setpoint? Note the minor adjustments the PID controller is making to the controller output.

6. Enter in a change in level several cm above or below the current Auto Level Setpoint and observe the controller. Is the controller working?

   Verify your real step change plot with an instructor: ____________________________

7. Try opening the disturbance valve and verify that the controller reacts and maintains proper level.

   Verify your real controller disturbance plot with an instructor: ____________________________