Closed-Loop system simulation using MATLAB SIMULINK

Adapted from similar document by Dept. of Chemical Engineering, UC - Santa Barbara

Construction of an Open-Loop Block Diagram (Chapters 3,4)

1. Open Simulink and reproduce the following open-loop system simulation shown in Figure 1. In this example, a second order system with delay is to be controlled. The system output is also subject to a disturbance (or Load “L”). The input/output and disturbance/output transfer functions are assumed to be the same. Include a delay of 1 second in both transfer functions. (The transport delay icon is found on the continuous-time icon library of Simulink. You set the delay by double-clicking on the transport-delay icon).

![Open-Loop Block Diagram](image)

**Figure 1.** Open-Loop Block Diagram

2. You may also want to add scope blocks so that you can observe the response directly in the simulink environment.

3. Save the file as `examplesim` so that you can use it in constructing a closed-loop block diagram.

Construction of Closed-Loop Block Diagram

1. Open the file `examplesim` if it is not already open.

2. Click on the connection between the U block and Process TF block and delete it. Rename the U block to “R”. This block will be used to produce a step change in the set-point.

3. Place a copy of the Sum block to the right of R. Double click above the sum block icon and label the block “Sum1”. Open its dialog box and change the ++ sequence to +-. Instantly, the block’s icon will reflect this change. The top input will have a + located to the right of it while the bottom input will have a – located above it. Therefore, the output of Sum1 will be the top input
minus the bottom input. Connect the output of R to the top input of Sum1. Also, connect the output from Sum to the bottom input of Sum1. The output of Sum1 is the error between the set-point, R, and the controlled variable, C. Your model should look like figure 2:

![Figure 2. Partially Completed Closed Loop Diagram](image)

4. Now close the “Simulink” block menu and open the “Simulink Extras” block by right clicking on the block labeled “Simulink Extras”. Click on the “Additional Linear” block, then select the “PID Controller” and drag it to the right of the newest sum block. Connect the output of the Sum1 to the input of PID controller and the output of PID controller to the input of Process TF. Double-click on the PID controller and enter the IMC controller settings given in Table 12.1 of your textbook (tunings I or J should be appropriate). Please note that SIMULINK PID controller settings are P, I, and D where P = \( K_c \), I = \( K_c/\tau_i \), and D = \( K_c \tau_D \), so numerical values of P, I, and D should reflect these definitions. The model you have developed represents the closed-loop system. Your model should now look similar to figure 3:

![Figure 3. Closed-Loop Diagram](image)
Note that some text has been added to the block diagram shown on the previous page. Simply by double-clicking on a blank space in *examplesim* and typing. Text is added to the diagram at the point where you clicked. The E, P, X1, and X2 labels in the block diagram have no effect on simulation.

5. Now we are ready to simulate the closed-loop response of the system. We will start with the set-point response. Click on block L and set the Final value to 0 so that no step in the load will occur. Create a step in the set-point by clicking on R and setting the Final Value to 1. Start the simulation by selecting **Start** from **Simulation** menu. The resulting C and t variables in the workspace will be for the unit set-point response, because L (disturbance) has been disabled.

6. Select the MATLAB command window and type the following commands to obtain a labeled plot of the response.

   ```matlab
   plot(t,C)
xlabel('Time')
ylabel('C')
title('Set-point Response')
   ```

   Figure 9 shows the resulting plot. Once a new simulation is performed, SIMULINK will write new values to t and C. To avoid losing the old values, type `t1=t; C1=C;` to define the new variables t1 and C1.

![Set-point Response](image)

**Figure 4.** Unit set-point response for ITAE (load) settings.

7. Now simulate the unit load response. Double-click on R and set Final value to 0. Double-click on L and set Final value to 1. Again, select **Start** from the **Simulation** to begin the simulation. Type `plot(t,C)` to view the response.
Label this plot the same way as the previous one except replace the title Set-point Response with Load Response. Figure 10 shows the resulting plot.

![Load Response](image)

**Figure 5.** Unit load response for ITAE (load)

Before starting another simulation, type \( t2 = t; \ C2 = C; \)

8. Now simulate the unit load response for ITAE (set-point) by entering the set-point controller settings into the PID controller block and simulating. Type \( t3 = t; \ C3 = C; \) after the simulation has ended.

9. Next simulate the unit set-point response for ITAE (set-point) by setting Final value to 0 in L and Final value to 1 in R and simulating.

10. To reproduce Figure 12.8 on page 290 of the textbook, enter the following set of commands at the MATLAB prompt:

```matlab
subplot(211)
plot(t1, C1, '-', t, C, '--')
xlabel('Time')
ylabel('C')
title('Set-point Response')
subplot(212)
plot(t2, C2, '-', t3, C3, '--')
xlabel('Time')
ylabel('C')
title('Load Response')
```
The `subplot(mnp)` command breaks the Figure window into an m-by-n matrix of small rectangular panes and makes the p-th pane active. The `plot` command following `subplot` will place the plot in the active pane. To learn more, type `help subplot`. The `plot(x1, y1, 'linetype1', x2, y2, 'linetype2', ...)` command plots x1 vs. y1 with linetype1, x2 vs. y2 with linetype2 and, so on, on the same plot. To learn more, type `help plot`. For the set of commands listed above, Figure 11 shows the resulting MATLAB Figure window.

![Set-point Response](image1)

![Load Response](image2)

**Figure 6.** Comparison of PID controllers with ITAE settings for Example 12.7. Solid line – ITAE (load); Dashed line – ITAE (set-point)

Adding the legend can be tedious. Two commands, which can be used to generate a legend, are `text` and `gtext`. If you feel motivated, type `help text` and `help gtext` to learn how.