Lab 1: Simulating Control Systems with Simulink and MATLAB

EE128: Feedback Control Systems

Fall, 1998

1 Simulink Basics

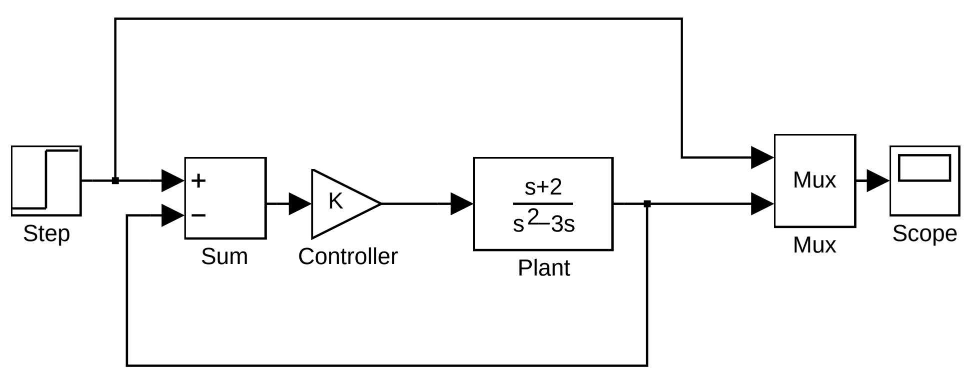
Simulink is a graphical tool that allows us to simulate feedback control systems. For most of the systems we will encounter, we only need to be concerned with a small fraction of Simulink's component library. In particular, the components you should be familiar with are:

- Sources library
  - step – generates a unit step signal
  - ramp – generates a ramp signal
  - sine wave – generates a sinusoid
- Sinks library
  - scope – used for viewing system output
- Linear library
  - gain – a constant gain
  - sum – used to add two or more signals
  - transfer fcn – used to add a system block in transfer function form
  - state-space – used to add a system block in state-space form
- Connections library
  - mux – used to multiplex signals together in order to plot several on one graph

The best way to learn Simulink is by *doing*. Let's try an example. . .

2 First Example

Load simulink by simply typing "simulink" at the MATLAB prompt. Once simulink has loaded, begin placing components on the empty window. Make your system look like the following:



Now, replace the variable "K" with a "1" in the constant gain controller. Run the simulation and see what happens. Is the output stable? Increase the gain to 2 and re-run the simulation. Continue increasing the gain to 10, and observe the results.

Task 1. For what values of K is the system stable? For roughly what value of K is the system completely oscillatory?

Now, change the gain to 8, and replace the step input with a ramp input. Re-run the simulation.

Task 2. The steady-state error is defined as the difference between the input and output signals when $t \to \infty$. Using the mouse, zoom in on the scope output at $t = 10$ seconds. Make an estimate of the steady-state error of this system due to a ramp input.

Task 3. What is the closed-loop transfer function for this system (leave K as a variable)? What is the characteristic equation? What are the locations of the poles (i.e. $s =$?)

3 Non-linear Example

Using the equation of motion for a *damped* pendulum given by:

$$\ddot{\theta} + \frac{c}{ml}\dot{\theta} + \frac{g}{l}\sin\theta = \frac{T_c}{ml^2},$$

construct a system with input T_c and output θ. Choose $l = 2.5$, $m = 0.75$, and $c = 0.15$. To construct this system, you will need to use the "trigonometric function" component which is found in the "Nonlinear library" of Simulink. The only other components you will need are: a summer, constant gains, and integrators.

Task 4. Print your Simulink representation of this system.

Task 5. Print the response of this system to a pulse having amplitude 20 and width 0.1 seconds. There are several ways to generate such a pulse; one easy way is to use a "pulse generator" with period= 100 and duty cycle= 0.1. When you print the graph, show the response for 50 seconds of time. To change the simulation time, choose "Simulation parameters. . ." from the "Simulation" menu.

Task 6. Replace the non-linear sin term with the linearization we discussed in class. Again print the response of this system to the pulse input. Is this system still stable? Is the linear approximation a good approximation?

Task 7. Now, put the non-linear sin term back into the simulation. Print the response of this system to a pulse having amplitude 200 and width 0.1 seconds. Again, show the response for 50 seconds. To what value is the output converging? Why?

Task 8. Again replace the sin term with the linear approximation. Print the response of this system to a pulse having amplitude 200 and width 0.1 seconds. To what value is the output converging now? Why? Is the linear approximation a good approximation?

4 MATLAB Basics

Before you begin the next sections, it would be a good idea to run the MATLAB Control System Toolbox demo. This is done by typing "ctrldemo" at the MATLAB prompt. Aside from the basic MATLAB plotting commands, you should become familiar with the following commands:

- **tf** – This command is used to enter transfer functions. For example, to enter the transfer function $H(s) = \frac{s+2}{s^2+5}$, you would type "H=tf([1 2],[1 0 5])". The first parameter is a row vector of the numerator coefficients. Similarly, the second parameter is a row vector of the denominator coefficients.

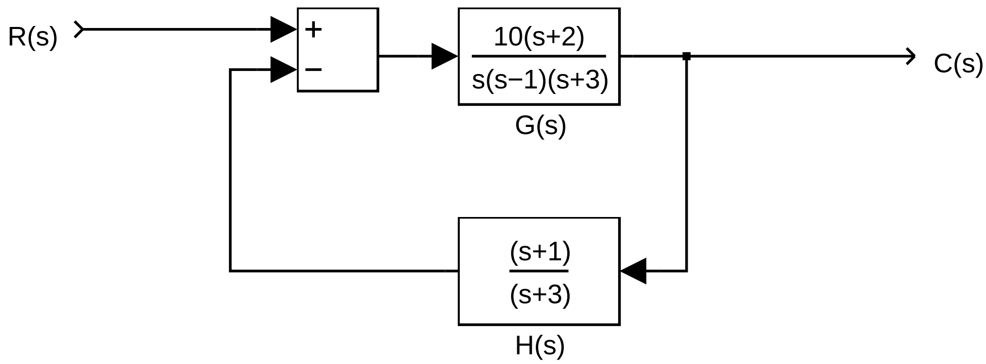
- **conv** – This command is used to convolve two polynomials. It is particularly useful for determining the expanded coefficients for factored polynomials. For example, this command can be used to enter the transfer function $H(s) = \frac{s+2}{(s+1)(s-3)}$ by typing "H=tf([1 2],conv([1 1],[1 -3]))"

- **series** or ***** – This command is used to combine two transfer functions that are in series. For example, if $H(s)$ and $G(s)$ are in series, they could be combined with the command "T=G*H" or "T=series(G,H)".

- **feedback** – This command is used to combine two transfer functions that are in feedback. For example, if $G(s)$ is in the forward path and $H(s)$ is in the feedback path, they could be combined with the command "T=feedback(G,H)".
- **step** – This command is used to plot the step response of a system. For example, "step(T)" would plot the step response of the system $T(s)$.
- **bode** – This command is used to plot the frequency repsonse. For example, "bode(T)" would plot the frequency response of the system $T(s)$.
- **rlocus** – This command is used to plot the root locus. For example, "bode(G*H)" would plot the frequency response of the system $G(s)H(s)$. Keep in mind that this command is used on the loop gain of the system as opposed to the closed-loop transfer function. For example, consider the standard negative feedback system with forward path G and feedback path H. The loop gain would be $G(s)H(s)$ whereas the closed-loop transfer function would be $\frac{G(s)}{1+G(s)H(s)}$.

5 MATLAB example

Consider the following system:



Task 9. Use MATLAB to find $T(s) = \frac{C(s)}{R(s)}$, the closed-loop transfer function. Show the MATLAB commands required to calculate $T(s)$. Make use of the commands *tf, conv,* and *feedback*.

Task 10. Print the step response, frequency response, and root locus for this system.