

# Lab 1: Simulating Control Systems with Simulink and MATLAB

EE128: Feedback Control Systems

Fall, 2005

## 1 Simulink Basics

Simulink is a graphical tool that allows us to simulate feedback control systems.

### 1.1 Placing Components

Simulink can often be very finicky in regards to placement of components and connections. To place a component, drag it from the component browser to the model space. To make a connection, hold down CTRL and click on the arrows on each block that you wish to connect. To connect multiple lines to a single block, hold down CTRL and click on the line already attached to the block and then make the second connection.

### 1.2 Components

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:

- "Continuous" library
  - Integrator – integrates a signal
  - State-Space – used to add a system block in state-space form
  - Transfer Fcn – used to add a system block in transfer function form
- "Math Operations" library
  - Gain – a constant gain
  - Sum – used to add two or more signals
  - Trigonometric Function – used to place non-linear trigonometric elements
- "Signal Routing" library
  - Mux – used to multiplex signals together in order to plot several on one graph
- "Sinks" library
  - Scope – used for viewing system output
  - To workspace – used to transfer a signal to MATLAB
- "Sources" library
  - Ramp – generates a ramp signal
  - Sine Wave – generates a sinusoid
  - Step – generates a unit step signal

### 1.2.1 'Gain'

To modify a 'Gain' component, double-click on the component. The 'Gain' parameter can be either a number or, more useful, can be an expression. For instance, the 'Gain' parameter can be set as '9.8/2.5' instead of having to calculate and then enter '0.45'.

### 1.2.2 'Sum'

The 'Sum' component is equivalent to a summing junction in a block diagram. To add more input nodes or change the sign of an input node, double-click on the 'Sum' component and modify the text in the 'List of Signs' parameter. To make the sum have a '+' and a '-' node, change the 'List of Signs' parameter to '|+ -'. To add a third summing input node, change the text to '|+ +- '.

### 1.2.3 'To Workspace'

The 'To Workspace' component sends data to the MATLAB workspace where you can plot it or process it as you wish. To set this up, double-click on the component and change the 'Save format' parameter to 'Array'. Also, set the 'Variable name' parameter to something descriptive, such as 'yout'. Simulink also automatically exports the time in the MATLAB variable 'tout'. This allows you to plot the output in MATLAB using the standard 'plot' command, and also allows you to nicely label the plots.

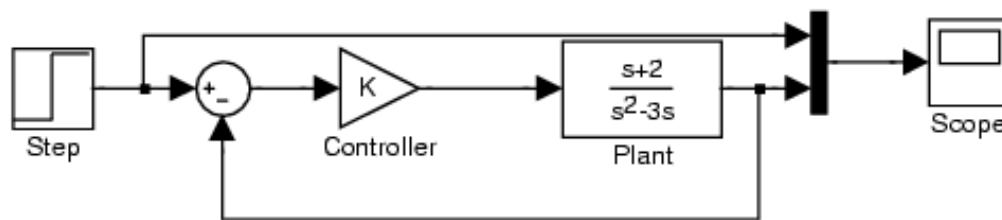
### 1.2.4 'Transfer Fcn'

To modify a 'Transfer Fcn' component, double-click on the block. The 'Numerator' and 'Denominator' parameters are the coefficients of the polynomials of the numerator and denominator of the transfer function. The coefficients are in order of decreasing power. For instance, if 'Numerator' is set to '[1 2 3]', then the numerator of the transfer function is  $s^2 + 2s + 3$ .

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, create a new model by going to File, New → Model (or alternatively press CTRL+N). Next, 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 by clicking on the triangle-shaped "play" icon in the toolbar and see what happens. Double-click on the scope to see the output of the simulation. Is the output stable? Increase the gain to 2 and re-run the simulation. Continue

increasing the gain to 10, and observe the results. Keep in mind that a linear system that is completely oscillatory, for a step or impulse input, is unstable. Convince yourself of this.

**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  $t \rightarrow \infty$ . 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.  $s = ?$ ) Leave  $K$  as a variable in the characteristic equation and pole locations. Also, evaluate the expression for the pole locations for any value of  $K =$  "unstable",  $K =$  "oscillatory" and  $K =$  "stable".

### 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  $\theta$ . 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 "Math Operations" library of Simulink. The only other components you will need are: a summer, constant gains, and integrators.

In this system, multiples of  $2\pi$  encode the same location. Is there any difference between  $\theta = 0$  and  $\theta = 2\pi$ ? Is there any difference between a system trajectory that starts at  $\theta = 0$  and goes to  $\theta = 0$  and a system that starts at  $\theta = 0$  and goes to  $\theta = 2\pi$ ?

**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 pulse width= 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 standard linearization for  $\sin\theta$ . The linearization arises by first considering the Taylor expansion:

$$\sin\theta = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \dots$$

and then disregarding higher order terms to yield the linearization (small-angle approximation):

$$\sin\theta \approx \theta$$

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?

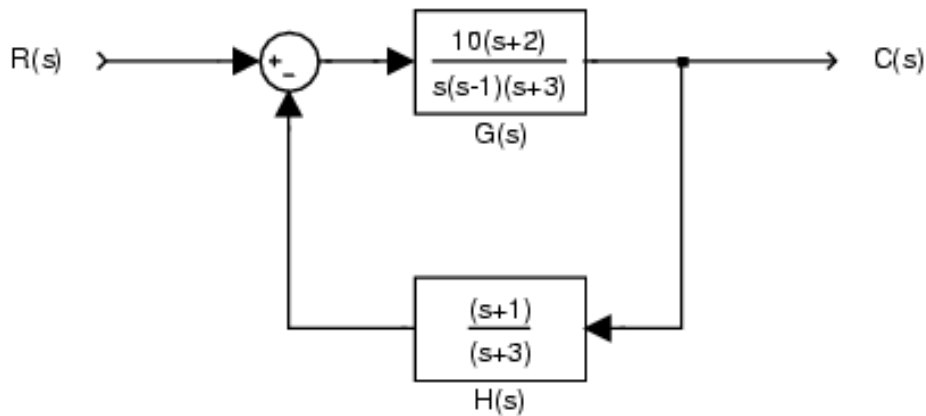
## 4 MATLAB Basics

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 “demo('toolbox','control')” 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 response. 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, “rlocus(G\*H)” would plot the root locus 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.