**ES202 – Principles of Systems Engineering**

**Due Date: Week of 21 February 2011**

**LABORATORY INVESTIGATION #3: Introduction to Simulink**

**Task:** Become familiar with the Simulink numerical simulation package under MATLAB.

**Objectives:**

1. To gain familiarity with Simulink, a fundamental simulation tool used in most courses in Systems Engineering.
2. To investigate simulated solutions to differential equations using Simulink.
3. To understand the interaction between Simulink and MATLAB’s workspace.
4. To simulate a simple, real-world system and investigate the effects of simplifying assumptions on a numerical simulation.

**1.0 Overview**

In this lab, you will learn the basics of Simulink, MATLAB’s companion graphical simulation environment. The basics of Simulink are covered in **Introduction to SIMULINK** course notes. ***You need to study the notes before coming to lab.***

**2.0 Introductory Procedure**

**Step #1:** Following the procedure given in the **Introduction to SIMULINK** course notes, construct a Simulink model of the first order differential equation:



1. Start MATLAB; set current directory; open diary file
2. Start Simulink with MATLAB command: **» Simulink**
   1. Click the blank page icon in the upper left hand corner of *Simulink Library Browser* (‘Create new model’).
   2. Create Simulink model using the *drag & drop* technique as discussed in your SIMULINK intro packet.
   3. Connect the blocks (see the example in the SIMULINK intro).
   4. Double click on each block to set parameters such as:
      1. *Step* start time
      2. *Gain* blocks
      3. *To Workspace* variable names
      4. Initial conditions (the initial value of the integrator should by the same as *y*(0)).
      5. Always change the *To Workspace* save format to *array*
   5. Save your Simulink model
   6. From the Simulink menu bar use the *Simulation / Configuration Parameters / Solver* window to set a stop time of 3 seconds.
   7. From the Simulink menu bar in the *Simulation / Configuration Paramters / Data Import/Export* window uncheck the *Limit Points to Last* box. This will ensure that the size of the time vector, *tout*, is the same as the variable vector defined in the *To Workspace* block.
   8. Use the *Simulation / Start* menu or “play” button to run simulation
3. Return to MATLAB and plot data

Display your results using both a Simulink *Scope* and a MATLAB plot. Check to ensure that your results match the results given in **Introduction to SIMULINK** course notes. Always annotate your MATLAB plot with axes labels and the title. In this case the title should be:

**‘<your name>, Step #1: 1^{st} Order Step Response’**

Always print a hardcopy for your lab report.

Question #1: Do your results match the **Introduction to SIMULINK** course notes?

**Step #2:** Change yourSimulink simulation to model the equation:



Display your results using the Simulink *Scope.* Use the MATLAB command figure(2)to create a second figure window and plot your results. Annotate your MATLAB plot with the title:

**‘<your name>, Step #2: 1^{st} Order Step Response’**

Question #2: How do the results of the simulation of equation compare to results for equation ?

**Step #3:** Change yourSimulink simulation to model the equation:



Double click on the integrator block to change the initial condition *source* to *external*. From the *Simulink Library Browser* drag a constant block from *sources* to your model. Connect this to the initial condition input to the integrator, and set the initial condition value in the constant block. Display your results using the Simulink *Scope.* Use the MATLAB command figure(3)to create a third figure window and plot your results. Annotate your MATLAB plot with the title:

**‘<your name>, Step #3: 1^{st} Order Step Response’**

Question #3: How do the results of the simulation of equation compare to results for equation ?

**3.0 Pendulum Simulation**

Create a new Simulink model to simulate the motion of a simple pendulum shown in Figure 1.





Figure 1: Diagram of a simple pendulum Figure 2: Force diagram for simple pendulum

**3.1 Dynamics Of A Simple Pendulum**

There are two forces acting on the mass of the pendulum: tension *T* in the arm of the pendulum and the weight of the mass *mg*. Neglecting friction, Figure 2 shows that the tension on the arm of the pendulum is  and the tangential force (acting at a angle to the tension force) is:



The tangential force on the mass causes acceleration . The tangential acceleration, *aT* is the second derivative of the tangential position:



where is the arc length covered by the pendulum and *L* the lengthof the arm. Note the minus sign appears because the direction of increasing in Figure 1 and the direction of the force in Figure 2 are opposite. Combining equations and , the tangential force equation becomes:



Using the notation, assuming *L* is constant, and dividing by *m,* the tangential force equation yields the second order differential equation:



Using the small angle approximation  gives the linearized equation:



**3.2 Simple Pendulum Simulation**

Construct a Simulink model to solve linearized approximation, equation . Note that there is no input and the pendulum moves because it has an initial angle other than zero. You should have two integrator blocks connected in series. The first integrator block integrates  to get  and the second integrates  to get . The initial angle is set within the integrator for . Note that  is computed in radians, and that you will need to convert the output to degrees for display. Be sure to input initial conditions  in radians!

Use the MATLAB variables *g* and *L* to calculate the gain in your Simulink *Gain* block. You will set the parameters in the main MATLAB workspace and SIMULINK will use the current values. This way, you can adjust the gains in MATLAB without changing the SIMULINK diagram.

NOTE:

* From the Simulink menu bar, use the *Simulation / Configuration Paramters / Solver…* window to set initial and max. step sizes of 0.01 and a stop time of 10 seconds.
* Be sure to uncheck the *Limit Points to Last* box on the *Simulation / Configuration Paramters / Data Import/Export* window, otherwise you will get an error when plotting.

In the MATLAB workspace, define *m = 1* and *g = 9.81*.You will set *L* (also in the MATLAB workspace) to various values and examine the results. In the *To Workspace* block name the output variable *theta*.

1. Define  in the MATLAB workspace, and simulate the pendulum motion for an initial angle of  (NOTE: use *RADIANS* for the linearized system (8)… you will get bad results if you use degrees). Run your simulation and use MATLAB to plot the angle  in degrees versus time.

plot(tout,theta\*180/pi)

1. Define  in the MATLAB workspace and simulate the pendulum motion for an initial angle of . Use the MATLAB hold command to plot the b) results on the same plot as the a) results. Specify that the b) results be a dashed line using a plot command of the form:

plot(tout,theta\*180/pi,’--’)

1. Use the MATLAB *legend* command to distinguish between the two curves. The syntax is:

legend(’L = 0.5’,’L = 1.0’)

Question #4: How do the results of the simulation for L = 0.5 compare to results for

L = 1.0?

**3.3 Nonlinear Pendulum Simulation**

Now, add to your simple model from section **3.2** a second set of blocks, reflecting the nonlinear dynamics of (7). To do so:

1. Select the entire model from **3.2**.
2. Copy that model and paste it into the same window, below the original version.
3. Between the output of the second integrator and the input of the gain block, add a *Trigonometric Function* block. Choose ‘sin’ as the function. Now, the input to the *Gain* block will be sin() as opposed to .
4. Note that in the new *To Workspace* block the variable name has been changed in order to prevent a conflict with the original.

Using L=1, run the simulation with initial= 100 and compare the results of the two sub-models (note that the *To Workspace* block will have a different variable name as well as a different block name… make note of that new variable name). Try it again for initial = 450. ***Make appropriate plots to show the results***.

Question #5: How do the results of the simulation for (7) compare to results for

(8)? What is the cause of the discrepancy, if there is one?

**4.0 Lab Report**

The report should be in the form of a memorandum and contain answers to the five questions. Attach as enclosures, and carefully explain, the following:

1. A printout of the Simulink diagram for equation .
2. Three plots for the solution of equations , , and
3. A printout of the Simulink diagram for equation
4. One plot of angular position versus time for the pendulum given and 
5. A printout of the Simulink diagram for equations (7) and
6. Comparison plots of angular position versus time for the pendulum using equation (7) and for initial angle of  and .