**KATHMANDU School of Engineering**

**UNIVERSITY Department of Electrical & Electronics Engineering**

CONTROL ENGINEERING LABORATORY EXERCISE-II

**Mathematical Modeling**

Reference: Sadat H., “*Computational aids in Control Systems using MATLAB*”, Milwaunkee School of Engineering, Milwaunkee

* 1. **Modeling a system**

**Example-1**

Consider the simple mechanical system as shown in the figure below. Three forces influence the motion of the mass, namely, the applied force, the frictional force, and the spring force.

*x(t)*

*f(t)*

*B*

*K*

*M*

Applying Newton’s law of motion, the force equation of the system is

Let and , then

With the system initially at rest, a force of 25 Newton is applied at time t = 0. Assume that the mass M = 1 Kg, frictional coefficient B = 5 N/m/sec., and the spring constant K = 25 N/m. The above equations are defined in an M-file **mechsys.m** as follows:

function xdot=mechsys(t,x);

F=25;

M=1; B=5; K=25;

xdot=[x(2);1/M\*(F-B\*x(2)-K\*x(1))];

Save the above instructions as **mechsys.m** and execute the following instructions:

tspan=[0,3];

x0=[0,0];

[t,x]=ode23('mechsys',tspan,x0);

subplot(2,1,1),plot(t,x)

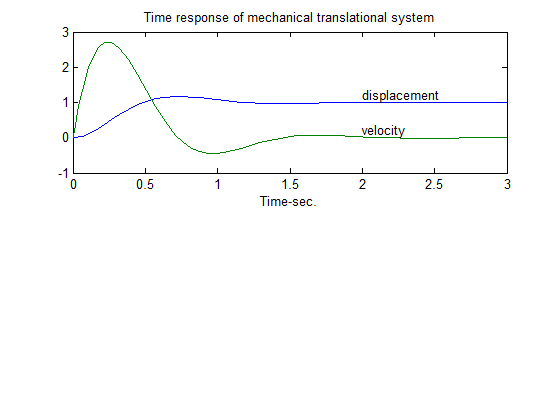
title('Time response of mechanical translational system')

xlabel('Time-sec.')

text(2,1.2,'displacement')

text(2,.2,'velocity')

The result of the simulation is as shown below:

****

**Example-2**

The circuit elements in the figure below are R = 1.4Ω, L = 2H, and C = 0.32F, the initial inductor current is zero, and the initial capacitor voltage is 0.5 volts. A step voltage of 1 volt is applied at time t = 0. Determine i(t) and v(t) over the range 0 < t < 15 sec. Also, obtain a plot of current versus capacitor voltage.

****

Applying KVL

and

Let

and

then

and

The above equations are defined in an M-file electsys.m as follows:

function xdot=electsys(t,x);

V=1;

R=1.4; L=2; C=0.32;

xdot=[x(2)/C;1/L\*(V-x(1)-R\*x(2))];

Save the above instructions as **electsys.m** and execute the following instructions:

x0=[0.5,0];

tspan=[0,15];

[t,x]=ode23('electsys',tspan,x0);

subplot(2,1,1),plot(t,x)

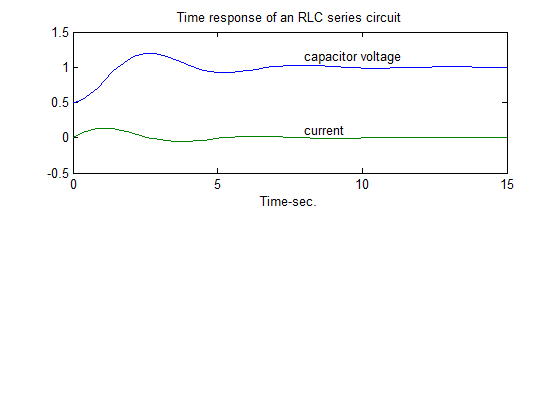
title('Time response of an RLC series circuit')

xlabel('Time-sec.')

text(8,1.15,'capacitor voltage')

text(8,.1,'current')

Result of simulation is as shown below:



**2.2 Introduction to Simulink**

SIMULINK is an interactive environment for modeling, analyzing, and simulating a wide variety of dynamic systems. A system in block diagram representation is built easily and the simulation results are displayed quickly.

Simulink is started from the MATLAB command prompt by entering the following command: *simulink.* The Simulink Library Browser window should now appear on the screen.

There are two major classes of elements in Simulink: blocks and lines. Blocks are used to generate, modify, combine, output, and display signals. Lines are used to transfer signals from one block to another.

**Blocks**

The subfolders underneath the "Simulink" folder indicate the general classes of blocks available for us to use:

• Continuous: Linear, continuous-time system elements (integrators, transfer functions, state-space models, etc.)

• Discrete: Linear, discrete-time system elements (integrators, transfer functions, state space models, etc.)

• Functions & Tables: User-defined functions and tables for interpolating function values

• Math: Mathematical operators (sum, gain, dot product, etc.)

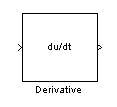
• Nonlinear: Nonlinear operators (coulomb/viscous friction, switches, relays, etc.)

• Signals & Systems: Blocks for controlling/monitoring signal(s) and for creating subsystems

• Sinks: Used to output or display signals (displays, scopes, graphs, etc.)

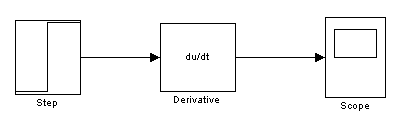
• Sources: Used to generate various signals (step, ramp, sinusoidal, etc.)

Blocks have zero to several input terminals and zero to several output terminals. Unused input terminals are indicated by a small open triangle. Unused output terminals are indicated by a small triangular point. The block shown below has an unused input terminal on the left and an unused output terminal on the right.



**Lines**

Lines transmit signals in the direction indicated by the arrow. Lines must always transmit signals from the output terminal of one block to the input terminal of another block.



A line can tap off of another line. This sends the original signal to each of two (or more) destination blocks. Lines can never inject a signal into another line; lines must be combined through the use of a block such as a summing junction.

**Building a System**

**Example-3**

The block diagram for a simple model consisting of a sinusoidal input multiplied by a constant gain is shown below:



This model will consist of three blocks: Sine Wave, Gain, and Scope. The Sine Wave is a Source Block from which a sinusoidal input signal originates. This signal is transferred through a line in the direction indicated by the arrow to the Gain Math Block. The Gain block modifies its input signal (multiplies it by a constant value) and outputs a new signal through a line to the Scope block. The Scope is a Sink Block used to display a signal (much like an oscilloscope).

To build a system, a new model window is brought, in which to create the block diagram. This is done by clicking on the "New Model" button in the toolbar of the Simulink Library Browser.

Building the system model is then accomplished through a series of steps:

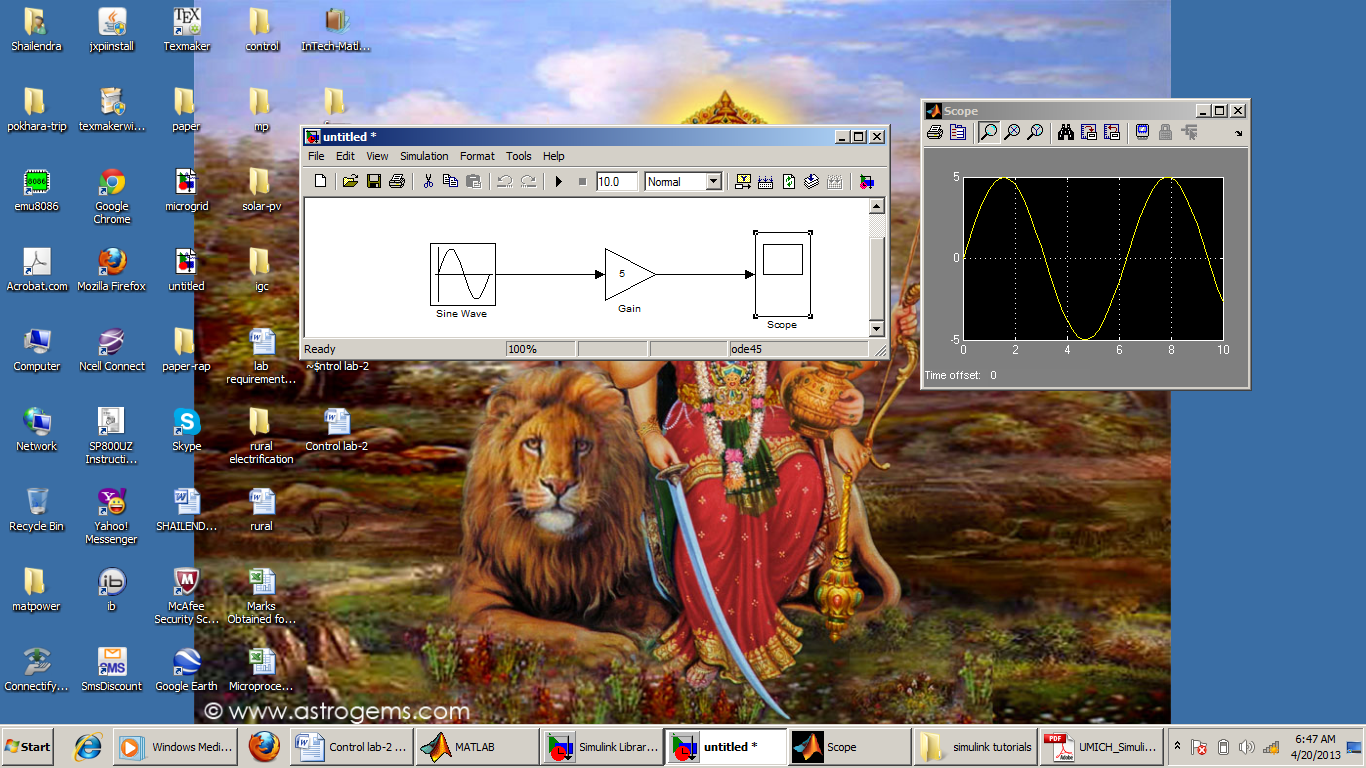
1. The necessary blocks are gathered from the Library Browser and placed in the model window.

2. The parameters of the blocks are then modified to correspond with the system

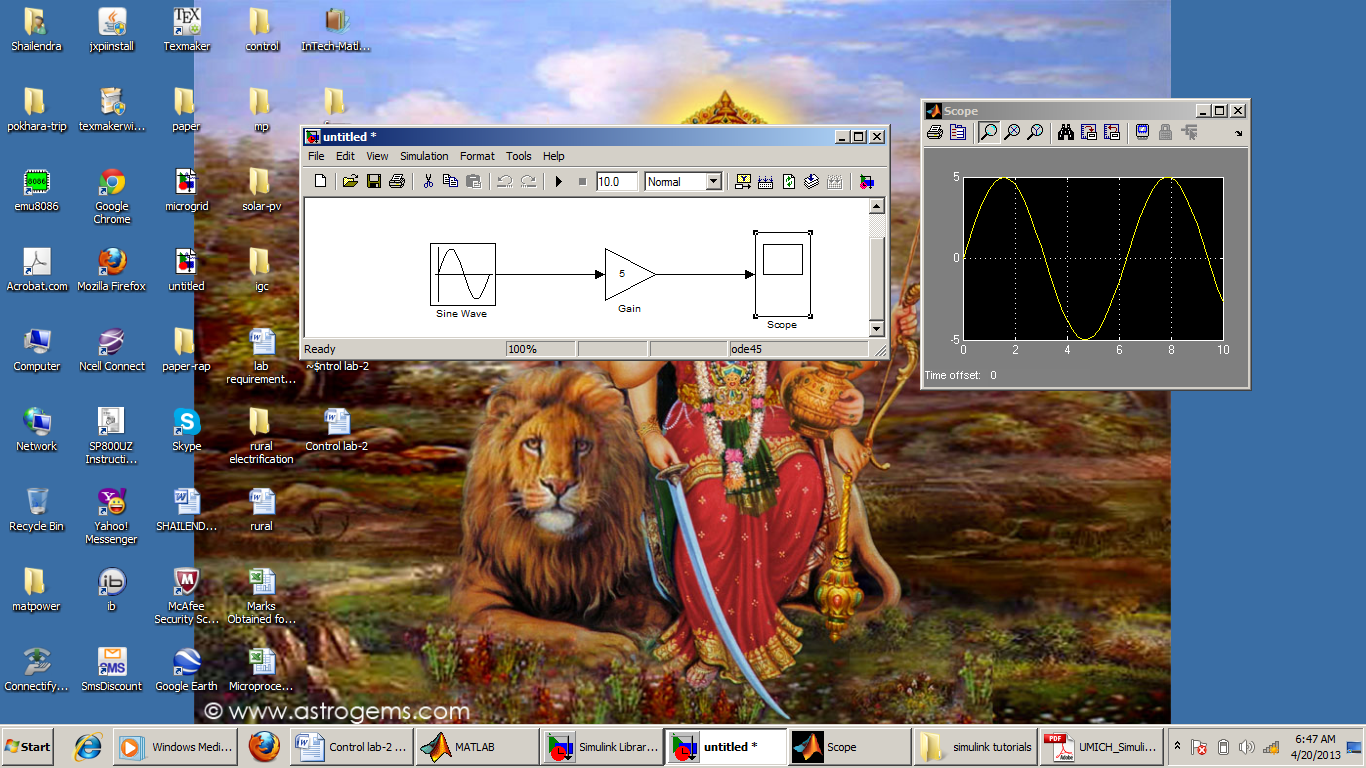
3. Finally, the blocks are connected with lines to complete the model.

**Running Simulations**

To simulate the system, go to the Simulation menu and click on Start, or just click on the "Start/Pause Simulation" button in the model window toolbar.



Double-click the Scope block to view the output of the Gain block for the simulation as a function of time. Once the Scope window appears, click the "Autoscale" button in its toolbar (looks like a pair of binoculars) to scale the graph to better fit the window.



**Example-4**

**Free Body Diagram and System Equation**

Consider the first-order model of the motion of a car. Assume the car to be travelling on a flat road. The horizontal forces acting on the car can be represented as shown in the figure below.

M

v

F

B

In this figure

* v is the horizontal velocity of the car (units of m/s).
* F is the force created by the car's engine to propel it forward (units of N).
* b is the damping coefficient for the car, which is dependent on wind resistance, wheel friction, etc. (units of N\*s/m).
* M is the mass of the car (units of kg).

The differential equation representing the system is

Assume that:

M = 1000 kg and b = 40 N\*sec/m

This system will be modeled in Simulink by using the system equation as above. This equation indicates that the car's acceleration (dv/dt) is equal to the sum of the forces acting on the car (F-bv) divided by the car's mass,

To model this equation, insert a Sum block, two Gain blocks and an integrator block into a new model window. Change the parameters of the blocks as per the requirement. Connect the blocks with lines as shown in the figure below.



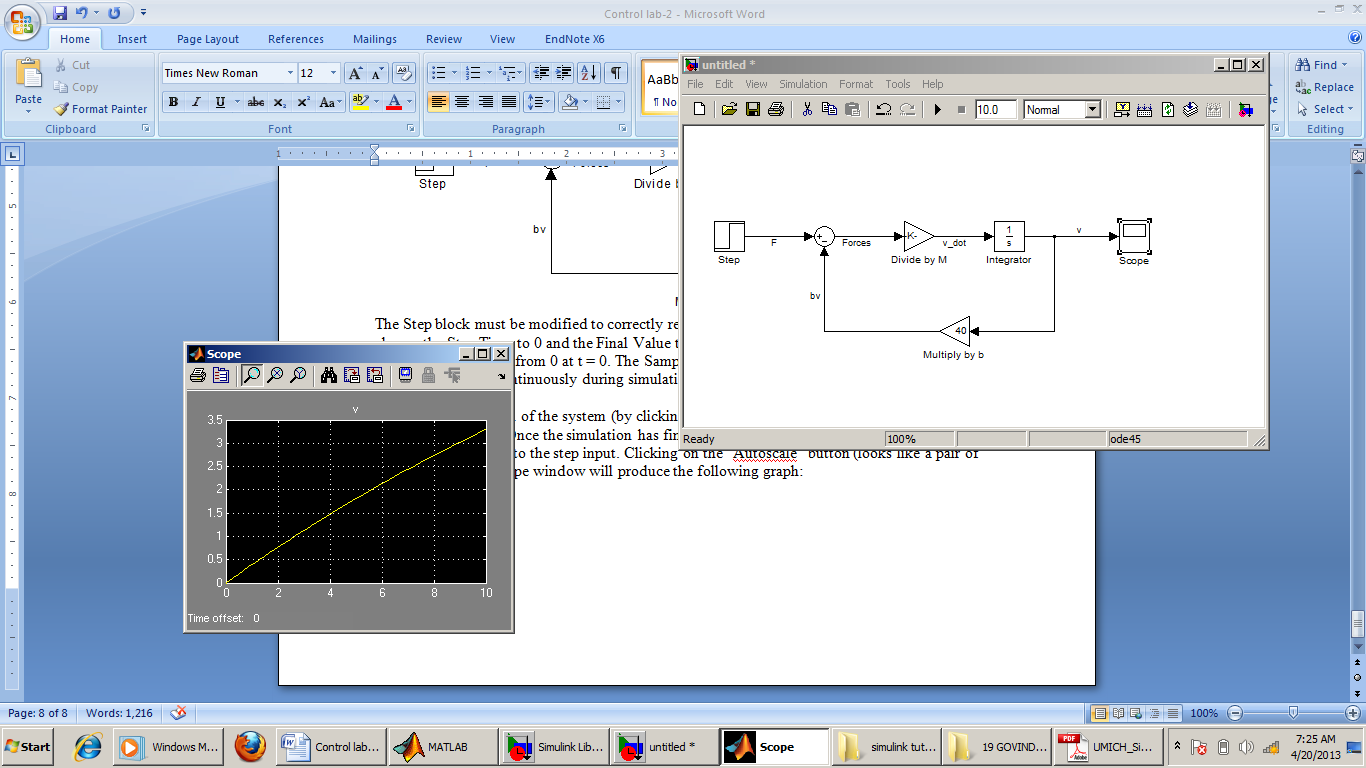
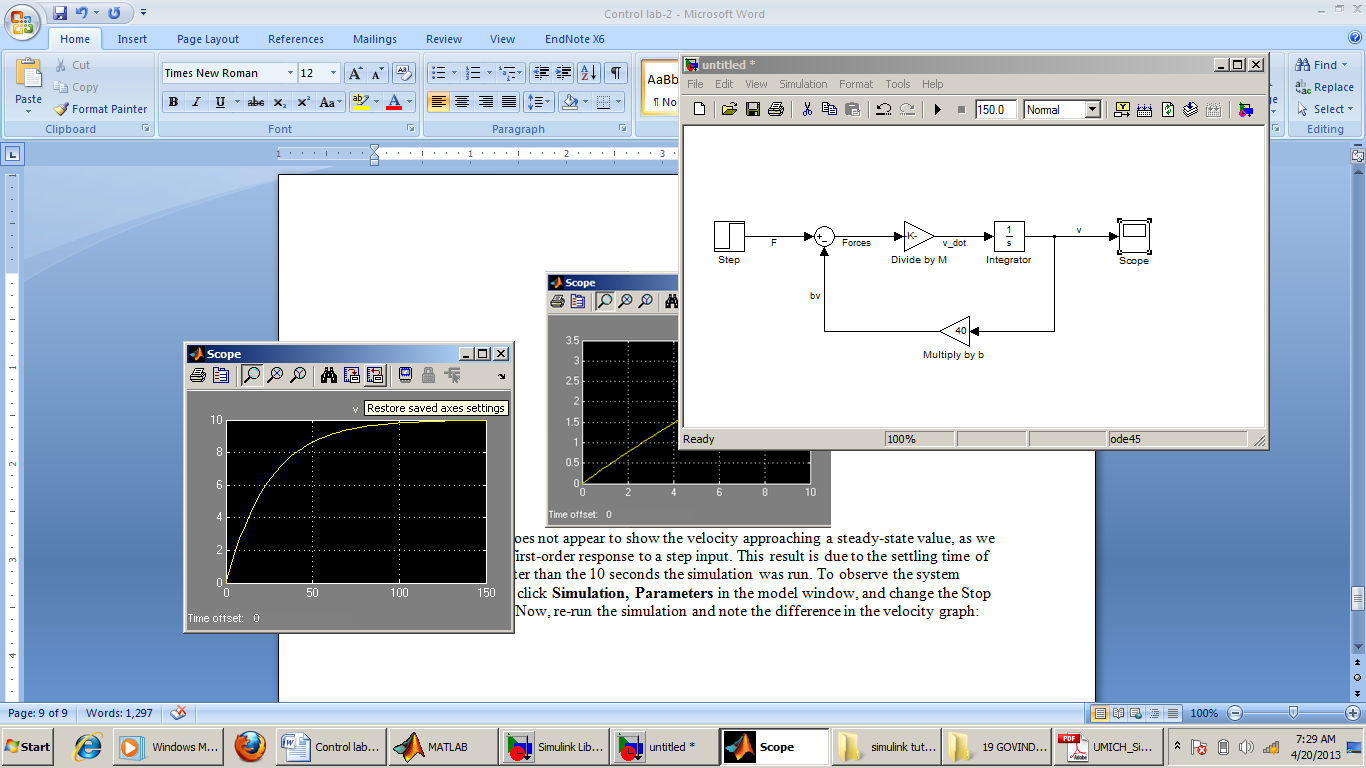
*System Response to Step Input*

To simulate the system the applied input F is to be specified. Assume that the car is initially at rest, and that the engine applies a step input of F = 400 N at t = 0. This is approximately equivalent to the car's driver quickly pushing down and holding the gas pedal in a steady position starting from a stoplight. Insert a Step block from the Sources subfolder into the model, and also add a Scope block from the Sinks subfolder to monitor the system's velocity, v.



The Step block must be modified to correctly represent the system. Double-click on it, and change the Step Time to 0 and the Final Value to 400. The Initial Value can be left as 0, since the F step input starts from 0 at t = 0. The Sample Time should remain 0 so that the Step block's input is monitored continuously during simulation.

Next, run a simulation of the system (by clicking the "Start/Pause Simulation" button or selecting **Simulation**, **Start**). Once the simulation has finished, double-click on the Scope block to view the velocity response to the step input. Clicking on the "Autoscale" button (looks like a pair of binoculars) in the Scope window will produce the following graph-(a).

1. (b)

This graph (a) does not appear to show the velocity approaching a steady-state value, as expected for the first-order response to a step input. This result is due to the settling time of the system being greater than the 10 seconds the simulation was run. To observe the system reaching steady-state, click **Simulation, Parameters** in the model window, and change the Stop Time to 150 seconds. Now, re-running the simulation will result in the velocity graph as shown in (b).

From this graph, we observe that the system has a steady-state velocity of about 10 m/s, and a time constant of about 25 seconds. Let's check these results with our original equation. For a step input of F = 400 N, the system equation is:

Setting dv/dt = 0 gives a steady-state velocity of 10 m/s, a result which agrees with the velocity graph above.To find the time constant of the system the characteristic equation as shown below can be used. 1000s + 40 = 0

Solving this gives the characteristic root, s = -0.04, and thus the time constant is indeed 25 seconds (τ = -1/s), as in the above the graph.

**SIMULINK diagram from State-space model**

**Example-5**

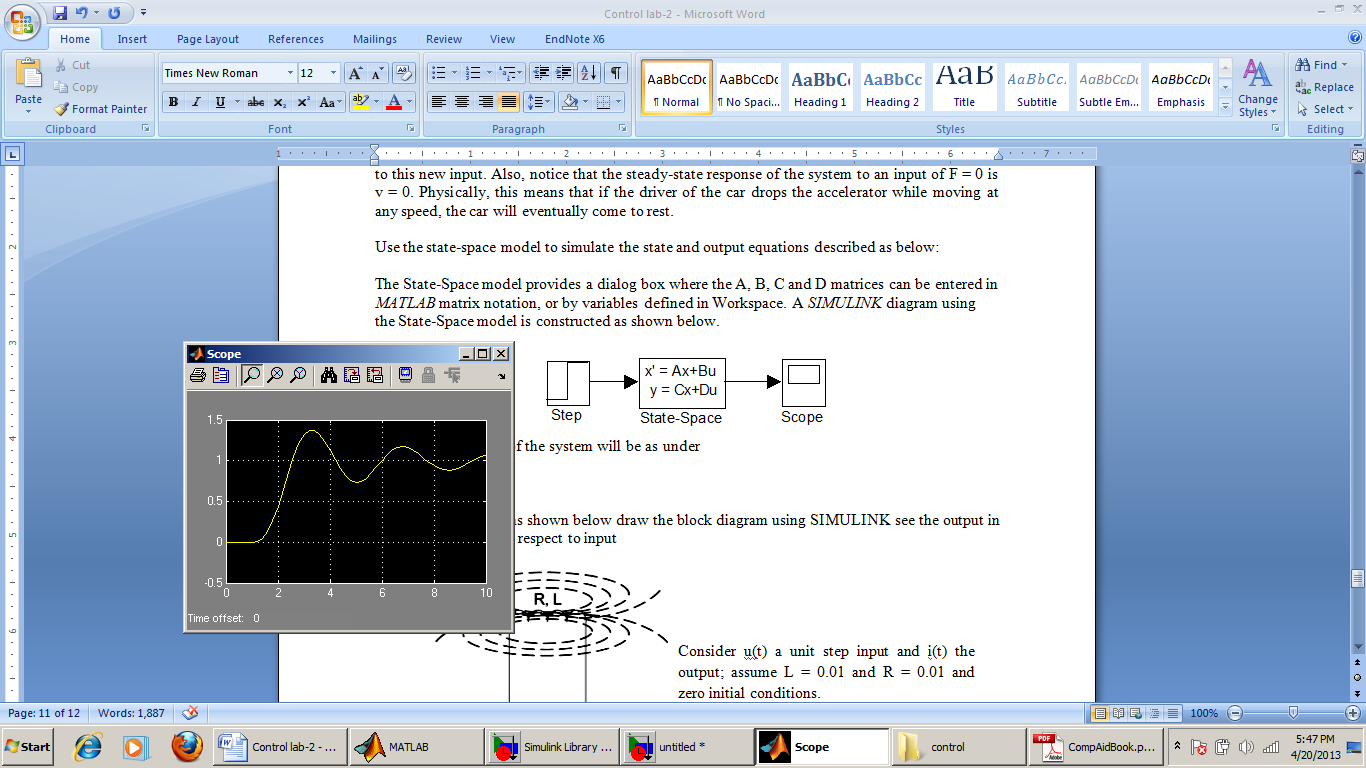
Use the state-space model to simulate the state and output equations described as below:

And

The State-Space model provides a dialog box where the A, B, C and D matrices can be entered in *MATLAB* matrix notation, or by variables defined in Workspace. A *SIMULINK* diagram using the State-Space model is constructed as shown below.



The output response of the system will be as under

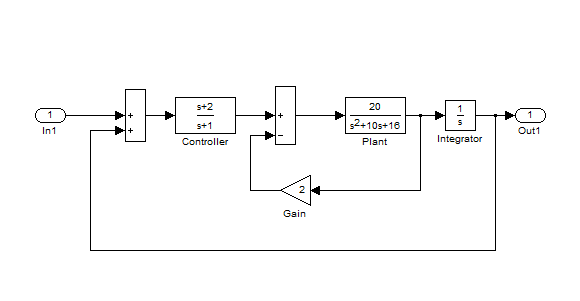


**State-space model from SIMULINK diagram**

SIMULINK provides the **linmod**, and **dlinmod** functions to extract linear models from the block diagram model in the form of the state-space matrices A, B, C, and D. The input and outputs of the SIMULINK diagram must be defined using **Inport** and **Outport** blocks in place of the **Source** and **Sink** blocks.

**Example-6**

Obtain the state-space model for the system represented by the block diagram shown below



The model is saved with a filename **ex6**. Run the simulation and to extract the linear model of this *SIMULINK* system, in the Command Window, enter the command

[A,B,C,D]=linmod('ex6')

The result is

|  |  |  |  |
| --- | --- | --- | --- |
| A =  0 0 0 20  1 -1 0 0  1 1 -10 -56  0 0 1 0 | B =  0  1  1  0 | C =  1 0 0 0 | D =  0 |

In order to obtains the transfer function of the system from the state-space model, we use the command

[num,den]=ss2tf(A,B,C,D)

The result is

num =

0 -0.0000 -0.0000 20.0000 40.0000

den =

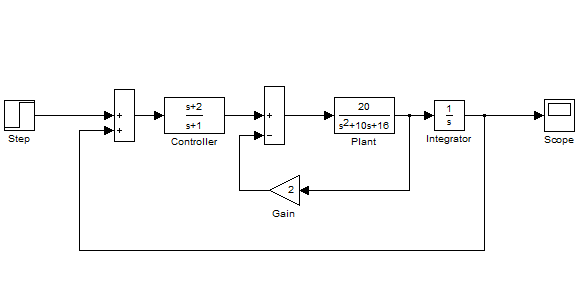
1.0000 11.0000 66.0000 36.0000 -40.0000

**SUBSYSTEMS**

*SIMULINK* subsystems provide a capability within *SIMULINK* similar to subprograms in traditional programming languages.

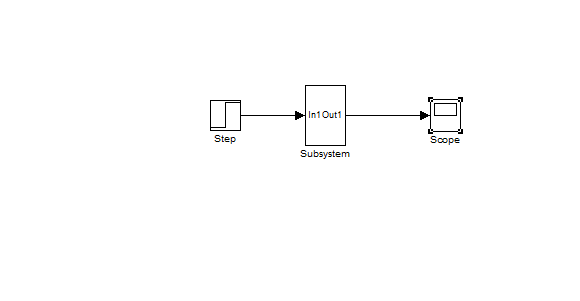
**Example-7**

To encapsulate a portion of an existing *SIMULINK* model into a subsystem, consider the *SIMULINK* model shown below and proceed as follows:



1. Select all the blocks and signal lines to be included in the subsystem with the bounding box as shown.

2. Choose Edit and select Create Subsystem from the model window menu bar. *SIMULINK* will replace the select blocks with a subsystem block that has an input port for each signal entering the new subsystem and an output port for each signal leaving the new subsystem. *SIMULINK* will assign default names to the input and output ports.



**Exercise:**

1. For the systems shown below draw the block diagram using SIMULINK see the output in the scope with respect to input



Consider u(t) a unit step input and i(t) the output; assume L = 0.01 and R = 0.01 and zero initial conditions.



Consider F(t) a step input and x(t) the output; assume m = 2Kg, K= 32 and B = 2 N-s/m and zero initial conditions.

1. For the system defined by the equation

Draw the SIMULINK block diagram and plot the output response y(t) with respect to u(t).