Group Members Name	Urwa Maryam, Umar Hayyat
Reg. #	2019-EE-352, 2019-EE-360
Marks	

Lab # 04

Modeling of Physical Systems using SIMULINK

Objective:

- The objective of this lab is to understand and implement the graphical user interface diagrams to model the physical systems for the purpose of design and analysis of control systems.
- To understand how to use SIMULINK in control systems.

Introduction:

This lab introduces powerful graphical user interface, Simulink of MATLAB. This software is used for solving the modeling equations and obtaining the response of a system to different inputs. Both linear and nonlinear differential equations can be solved numerically with high precision and speed, allowing system responses to be calculated and displayed for many input functions. To provide an interface between a system's modeling equations and the digital computer, block diagrams drawn from the system's differential equations are used.

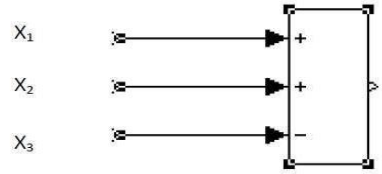
SIMULINK:

Simulink provides access to an extensive set of blocks that accomplish a wide range of functions useful for the simulation and analysis of dynamic systems. The blocks are grouped into libraries, generally there are following classes of functions:

- Mathematical functions such as summers and gains are in the Math library.
- Integrators are in the Continuous library.
- Constants, common input functions, and clock can all be found in the Sources library.
- Scope, To Workspace blocks can be found in the Sinks library.

Simulink is a graphical interface that allows the user to create programs that are actually run in MATLAB. Simulink uses blocks to write a program. Blocks are arranged in various libraries according to their functions. Properties of the blocks and the values can be changed in the associated dialog boxes. Some examples of blocks are given below:

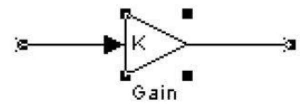
SUM (Math library)



A dialog box obtained by double-clicking on the SUM block performs the configuration of the SUM block, allowing any number of inputs and the sign of each. The sum block can be represented in two ways in Simulink, by a circle or by a rectangle.

GAIN (Math library)

A gain block is shown by a triangular symbol, with the gain expression written inside if it will fit. If not, the symbol - k - is used. The value used in each gain block is established in a dialog box that appears if the user double-clicks on its block.



INTEGRATOR (Continuous library)

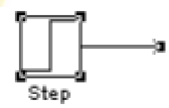
The block for an integrator as shown below looks unusual. The quantity 1/s comes from the Laplace transform expression for integration. When double-clicked on the symbol for an integrator, a dialog box appears allowing the initial condition for that integrator to be specified. It may be implicit, and not shown on the block, as in Figure (a). Alternatively, a second input to the block can be displayed to supply the initial condition explicitly, as in part (b) of Figure. Initial conditions may be specific numerical values, literal variables, or algebraic expressions.

Figure3: Two forms of the Simulink block for an integrator.

(a) Implicit initial condition. (b) Explicit initial condition.

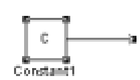
STEP (Source library)

Constants are created by the Constant block, which closely resembles Figure 4. Double clicking on the symbol opens a dialog box to establish the constant's value. It can be a number or an algebraic expression using constants whose values are defined in the workspace and are therefore known to MATLAB.



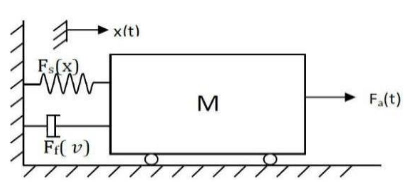
CONSTANTS (Source library)

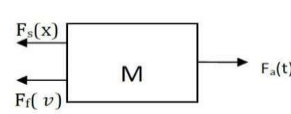
A Simulink block is provided for a Step input, a signal that changes (usually from zero) to a specified new, constant level at a specified time. These levels and time can be specified through the dialog box, obtained by double-clicking on the Step block.



Exercise 1:

Construct a Simulink diagram to calculate the response of the Mass Spring system. The input force





increases from 0 to 8 N at t = 1 s. The parameter values are M = 2 kg, K= 16 N/m, and B =4 N.s/m.

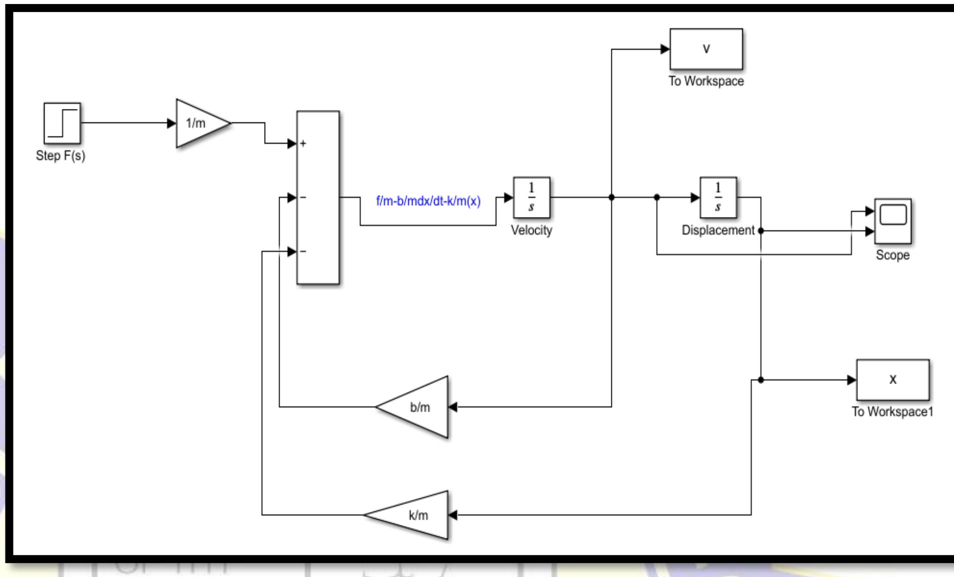
1. Make Simulink model
2. Plot velocity and displacement

Part 1:

$$M\frac{d^2x}{dt} + B\frac{dx}{dt} + Kx(t) = F$$

$$F = Kx(t) + Bv(t) + Ma(t)$$

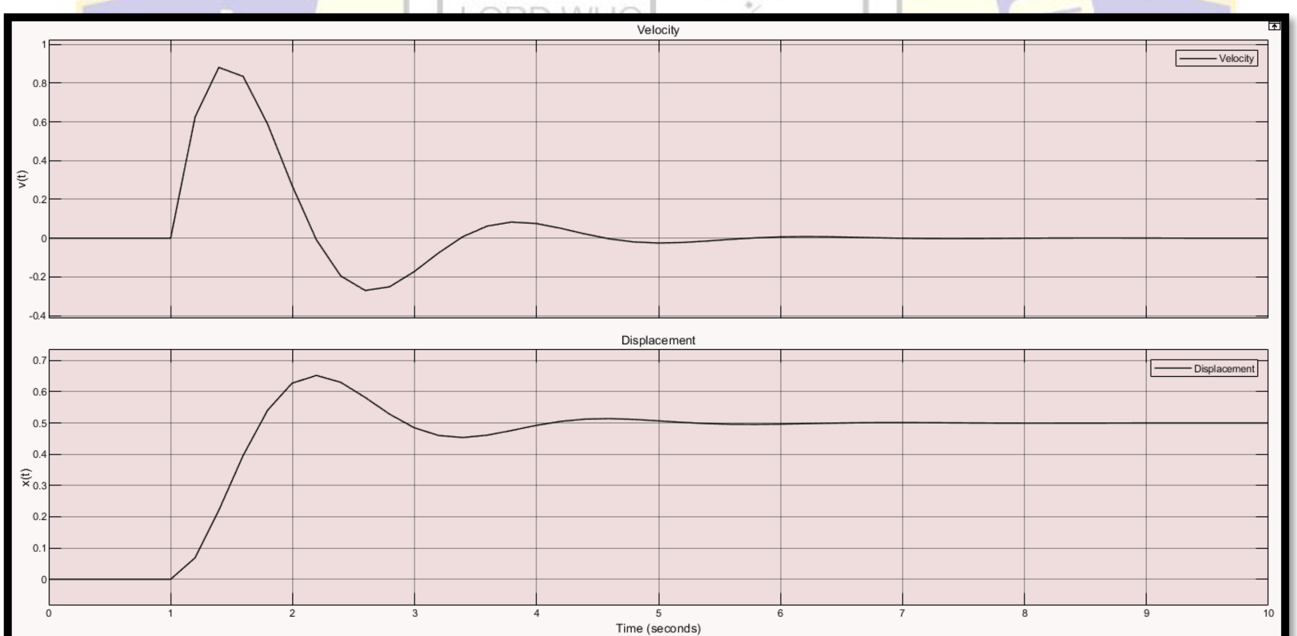
$$\frac{F}{M} - \frac{B}{M}v(t) - \frac{K}{M}x(t) = a(t)$$



Code:

```
k=16;
b=4;
m=2;
```

Part 2:



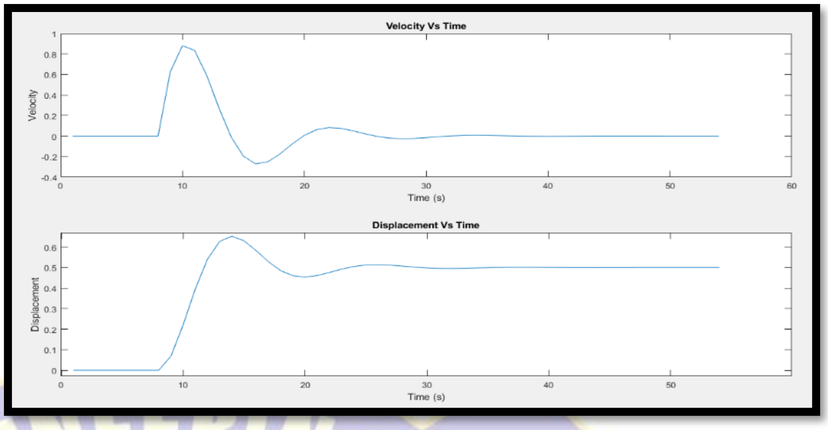
Second Solution:

Part 1:

```
k=16;
b=4;
m=2;
sim('lab4simulink')
subplot(2,1,1)
```

```
plot(v)
title('Velocity Vs Time')
xlabel('Time (s)')
ylabel('Velocity')
subplot(2,1,2)
plot(x)
title('Displacement Vs Time')
xlabel('Time (s)')
ylabel('Displacement')
```

Part 2:

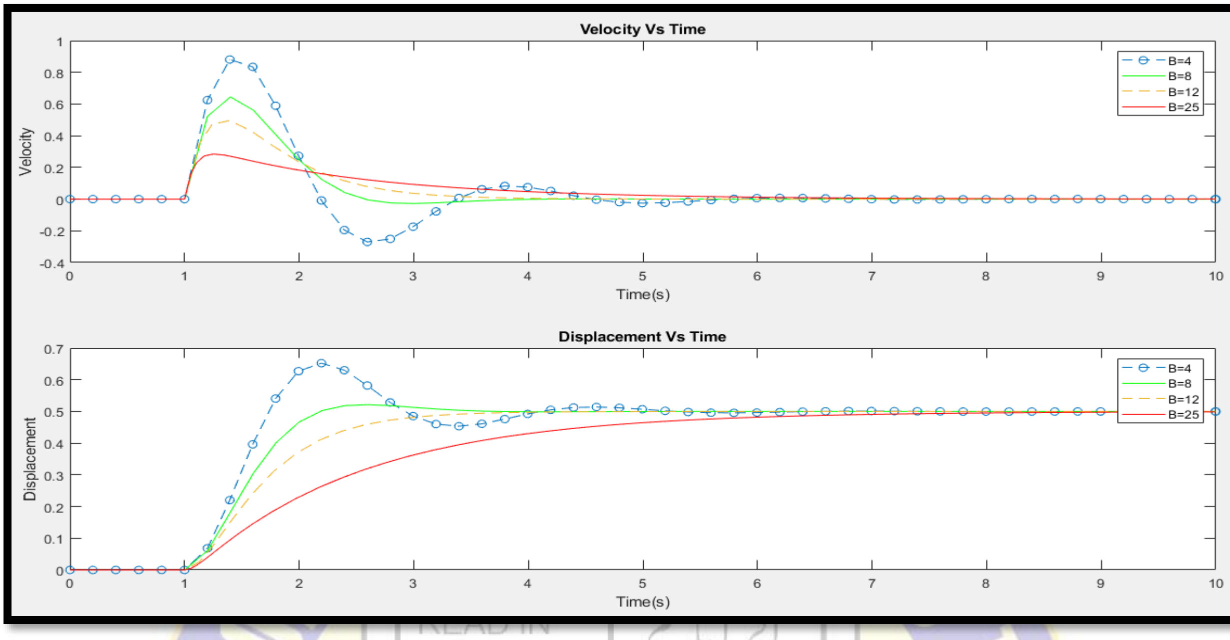


Exercise 2:

Simulate and compare the results of the variations in B in exercise 1. Take values of B = 4, 8, 12, 25 N-s/m.

Code:

```
close all;
clear all; clc;
m=2;
k=16;
b=4;
sim('lab4simulink')
subplot(2,1,1)
plot(t,v,'--o')
hold on;
subplot(2,1,2)
plot(t,x,'--o')
hold on;
b=8;
sim('lab4simulink')
subplot(2,1,1)
plot(t,v,'g')
hold on;
subplot(2,1,2)
plot(t,x,'g')
hold on;
b=12;
sim('lab4simulink')
subplot(2,1,1)
plot(t,v,'--')
hold on;
subplot(2,1,2)
plot(t,x,'--')
hold on;
b=25;
sim('lab4simulink')
subplot(2,1,1)
plot(t,v,'r')
title('Velocity Vs Time')
xlabel('Time(s)')
ylabel('Velocity')
hold on;
legend('B=4','B=8','B=12','B=25')
subplot(2,1,2)
plot(t,x,'r')
title('Displacement Vs Time')
xlabel('Time(s)')
ylabel('Displacement')
hold on;
legend('B=4','B=8','B=12','B=25')
```



Comment:
The effect of changing B is to alter the amount of overshoot or undershoot. These are related to a term called the damping ratio.

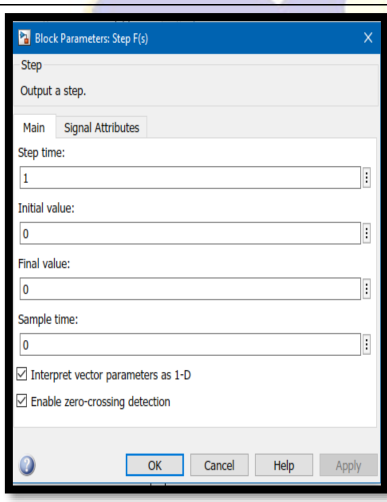
Exercise 3:

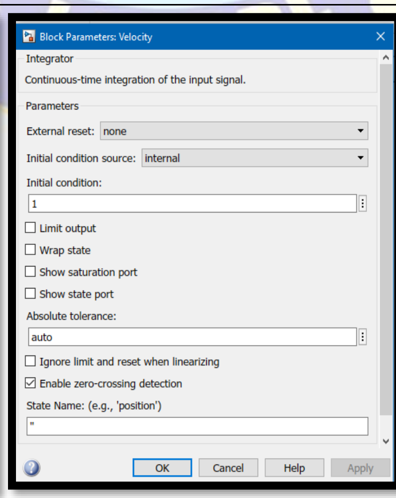
Find the response of the above system when there is no input for t ≥0, but when the initial value of the displacement x(0) is zero and the initial velocity v(0) is 1 m/s.

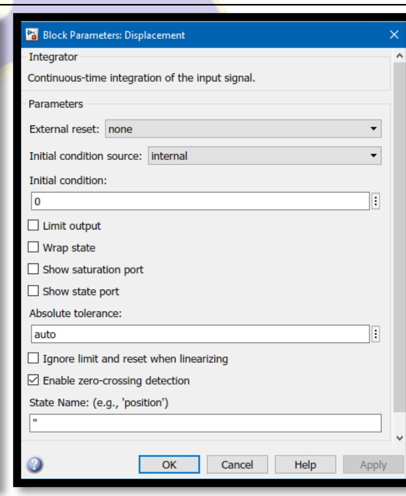
Steps:
In the previous program
- Set the size of the input step to zero
- Set the initial condition on Integrator for velocity to 1.0.
- Plot the results by running m-files.

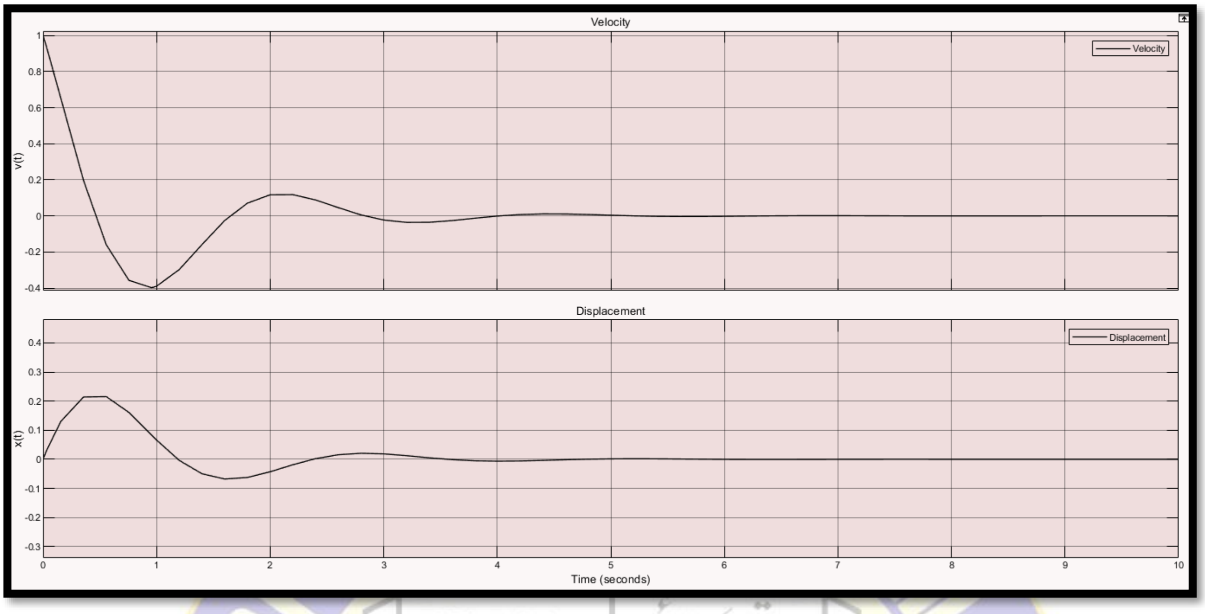
Code:

```
k=16;
b=4;
m=2;
```









Exercise 4:

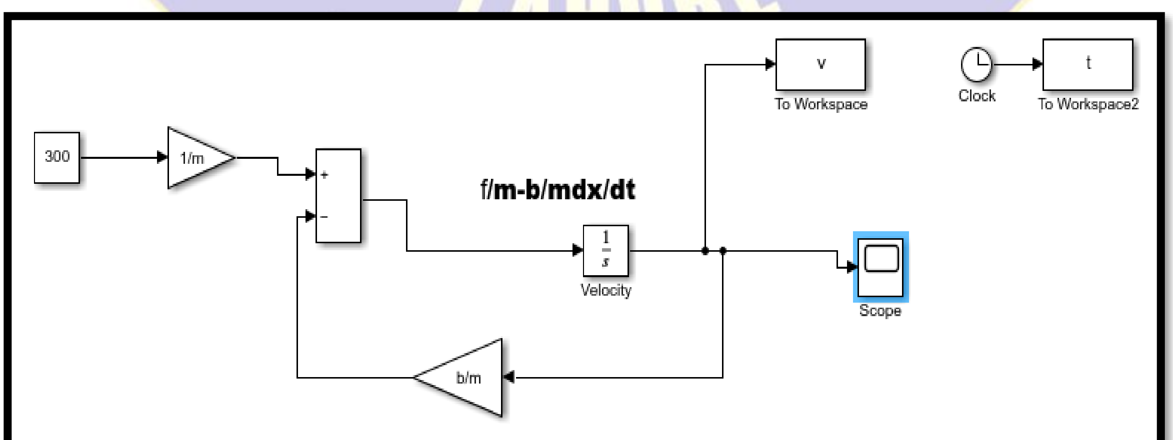
As we know in the cruise system, the spring force F(x)=0 which means that K=0 so we get:

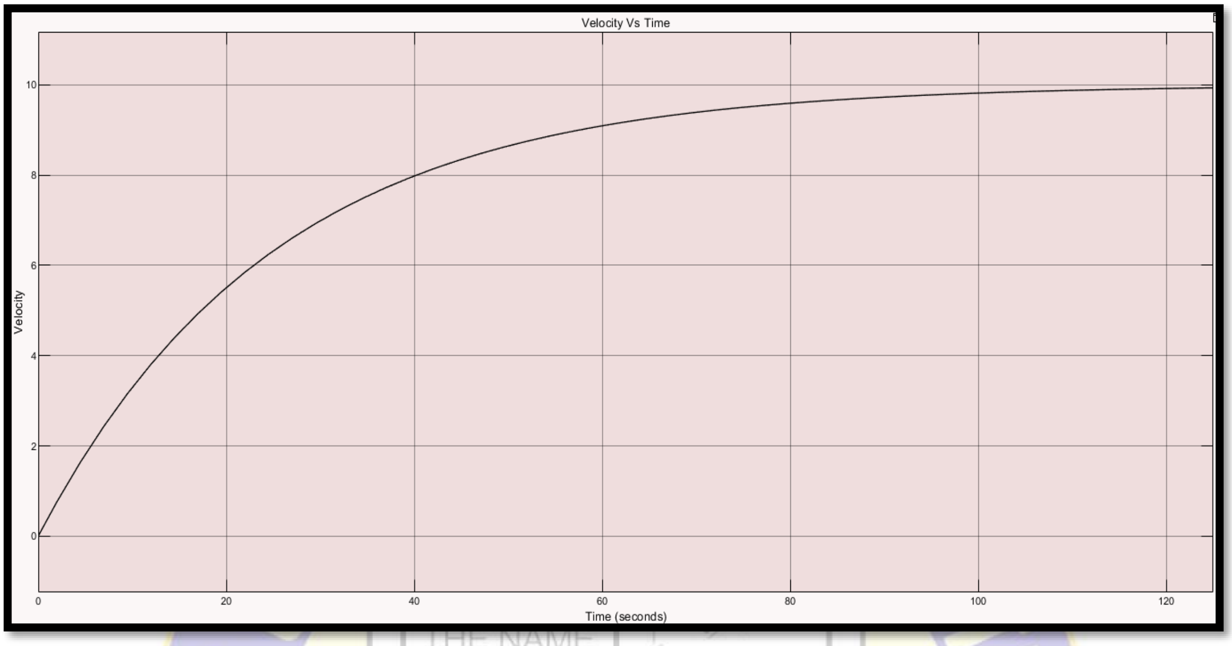
$$M\frac{dv(t)}{dt} + Bv = F_a(t)$$

Find the velocity response of the above system by constructing a Simulink block diagram and calling the block diagram from MATLAB m-file. Use M=750, B=30 and a constant force Fa = 300. Plot the response of the system such that it runs for 125 seconds.

Code:

```
clc;
clear all;
close all;
b=30;
m=750;
```





Conclusion:

In this lab we learned the modeling of physical mass spring system on Simulink. We also study the damping ratio increases when percentage overshoot decreases and response comes to steady state value. We observe the response of system by changing the initial values of velocity, displacement and force. At the end we observe the behavior of cruise system in spring force means K is zero.