Computer Aided Engineering MEE20007 (SIMULINK- Dynamic Systems) Lab4

In the previous tutorial on Simulink, you have learnt to build the basic models for solving mathematical and engineering problems. In this tutorial, you will acquire the knowledge about two important features:

(1) Integrating the computing power of MATLAB to Simulink
(2) Modelling and solving dynamic systems in Simulink

Exercise 1:

In this exercise you will be using a MATLAB function block in the Simulink modelling environment. Then you will create a function in MATLAB which will be used to drive a Simulink model.

1. Open a new blank model in Simulink.
2. From the "User Defined Functions" library bring the "MATLAB Function" block into the modelling window.
3. Place a "Constant" block at the input and three Display blocks near the output of this function block.
4. At this juncture, do not connect or wire the blocks together.
5. Double click on the MATLAB function block. The MATLAB Editor window for creating a new function will open.
6. Create this function:

```
function [mean,stdev,maximum] = statsCalc(vals)
% Calculates a statistical mean and a standard
% deviation for the values in vals.

len = length(vals);
mean = sum(vals)/len;
stdev = sqrt(sum(((vals-mean).^2))/len);
maximum=max(vals);
```

7. Save this function as "statsCalc".
8. Now click on "Run Model" button in the function editor. You will see three output ports created in the MATLAB function block in Simulink model.
9. Connect the blocks as shown in Fig 1 while assigning appropriate names to every block used for display.

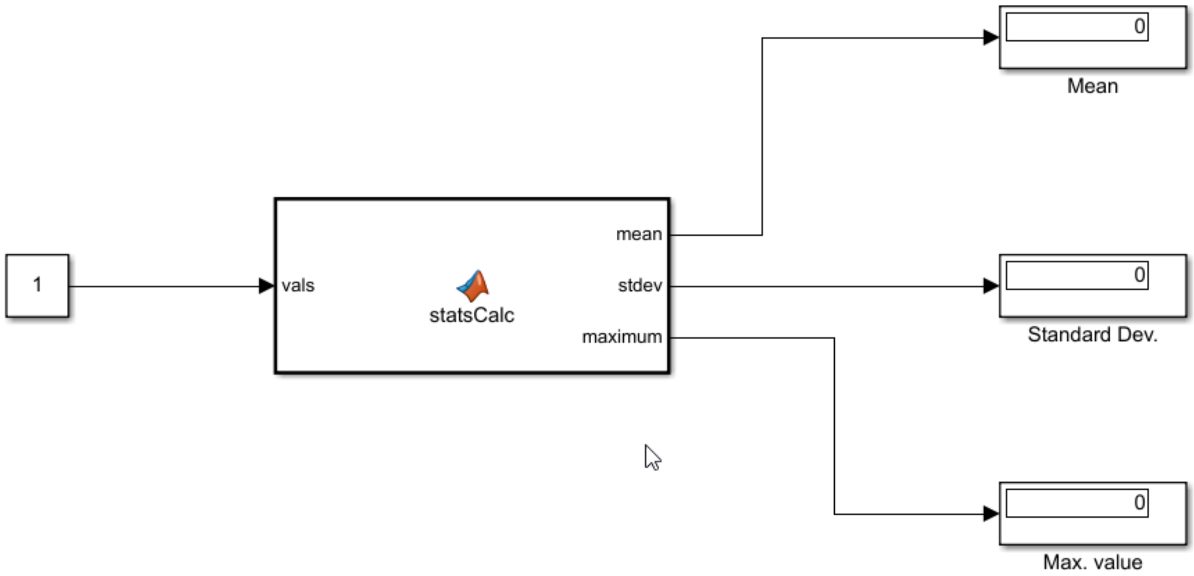


Figure 1: Connecting input and output blocks to MATLAB function block

10. Now change the value of constant block to [2 4.5 6 9 12]. Please note that the data is a vector and not a single value.

11. Run the model from Simulation window. You should see the output as shown in Fig 2.

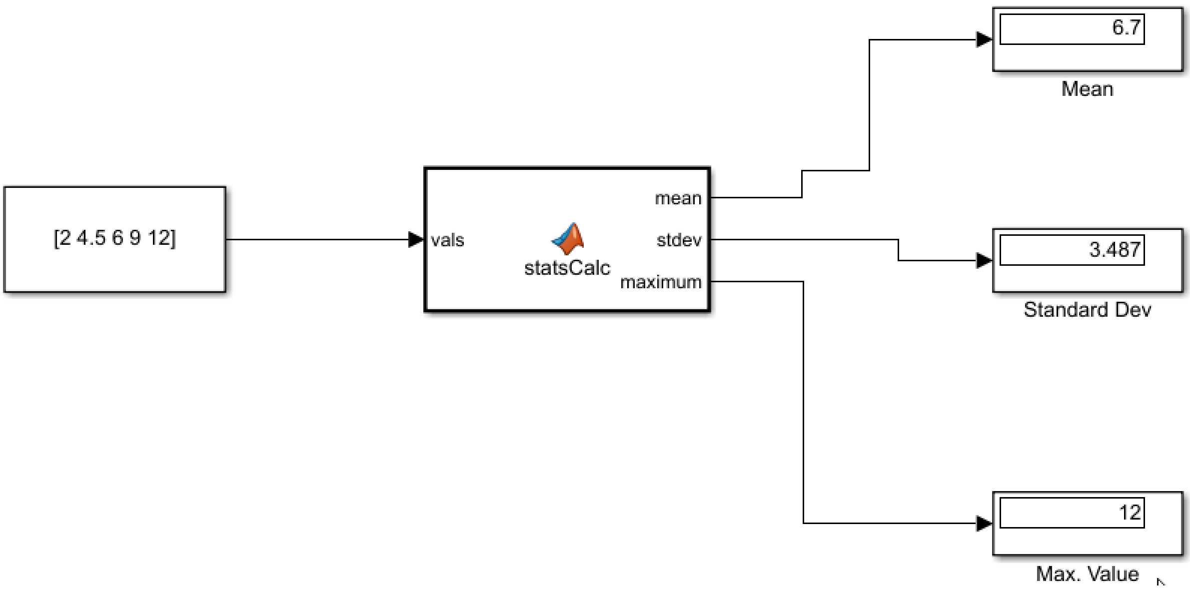


Figure 2: Results of statsCalc function in the Display blocks.

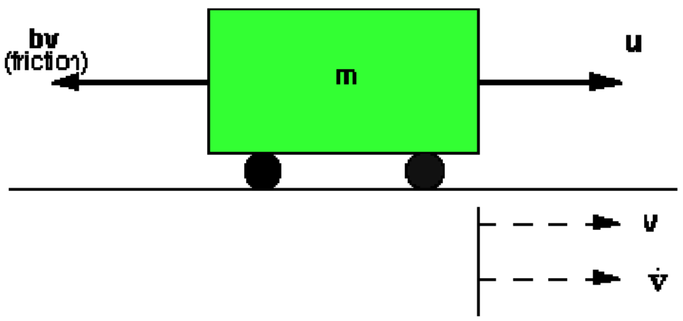
You can figure out that the statistical values are evaluated in the MATLAB while extracting output from the Simulink 'Constant' block and then displayed into the appropriate display panels in the Simulink model.

Exercise 2: Modelling a Cruise Control System in Simulink [1]

In this exercise you will model and solve in Simulink for the cruise system of a vehicle when it is excited by a given engine force. The system is solved by finding out the change of velocity with respect to time. The main interest is in finding out that after how much time the value of velocity will become steady and what value it will attain against the given friction and damping effects.

Physical setup and system equations

The model of the cruise control system is relatively simple. If the inertia of the wheels is neglected, and it is assumed that friction (which is proportional to the car's speed) is the main agent opposing the motion of the car, then the problem is reduced to the first order mass and damper system shown below.



Using Newton's law, modeling equations for this system becomes:

$$m\frac{dv}{dt} = u - bv \qquad (1)$$

where u is the force from the engine. For this example, let's assume that

Mass of the car = m = 1000kg
Damping or friction co-efficient = b = 50Nsec/m
Engine force = u = 500N

Building the Model

This system will be modeled by summing the forces acting on the mass and integrating the acceleration to give the velocity. Open Simulink and open a new model window. First, we will model the integral of acceleration

$$\int \frac{\mathbf{dv}}{\mathbf{dt}} = \mathbf{v}$$

- Insert an Integrator Block (from the Continuous block library) and draw lines to and from its input and output terminals.
- Label the input line "vdot" and the output line "v" as shown below. To add such a label, double click in the empty space just above the line.

Since the acceleration (dv/dt) is equal to the sum of the forces divided by mass, we will divide the incoming signal by the mass.

- Insert a Gain block connected to the integrators input line and draw a line leading to the input of the gain.
- Edit the gain block by double-clicking on it and change its value to "1/m".
- Change the label of the Gain block to "inertia" by clicking on the word "Gain" underneath the block.

Now, we will add in the forces which are represented in Equation (1). First, we will add in the damping force.

- Attach a Sum block to the line leading to the inertia gain.
- Change the signs of this block to "+-".
- Insert a gain block below the inertia block, select it by single-clicking on it, and select Flip from the Format menu (or type Ctrl-I) to flip it left-to-right.
- Set the gain value to "b" and rename this block to "damping".
- Tap a line (hold Ctrl while drawing) off the integrator's output and connect it to the input of the damping gain block.
- Draw a line from the damping gain output to the negative input of the Sum Block.

The force which brings the vehicle from rest into motion is the force ‘u’ applied by the engine of the car. Since this force is applied in a very short interval of time, therefore, it is simulated by a step input. Therefore, insert a step block and set the Step time to 0.2 and Final value to ‘u’ as shown in the Fig 3.

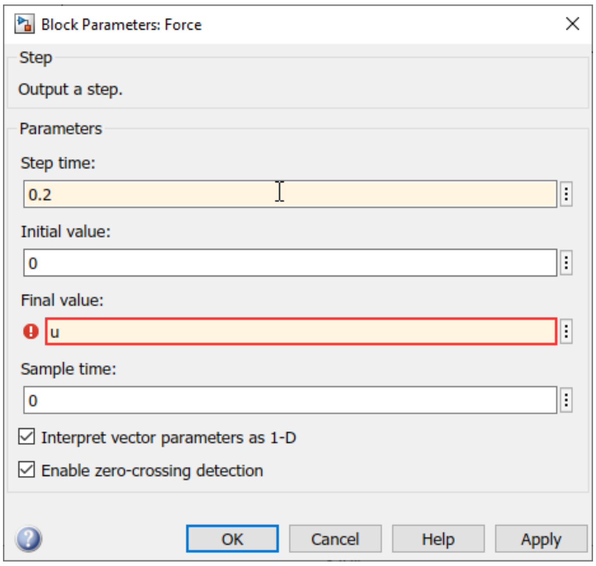


Figure 3: Parameters of Step input block

For viewing the variation of (output) vehicle velocity insert a Scope at the output of integrator block. The assembled model will like the diagram shown in Fig 4.

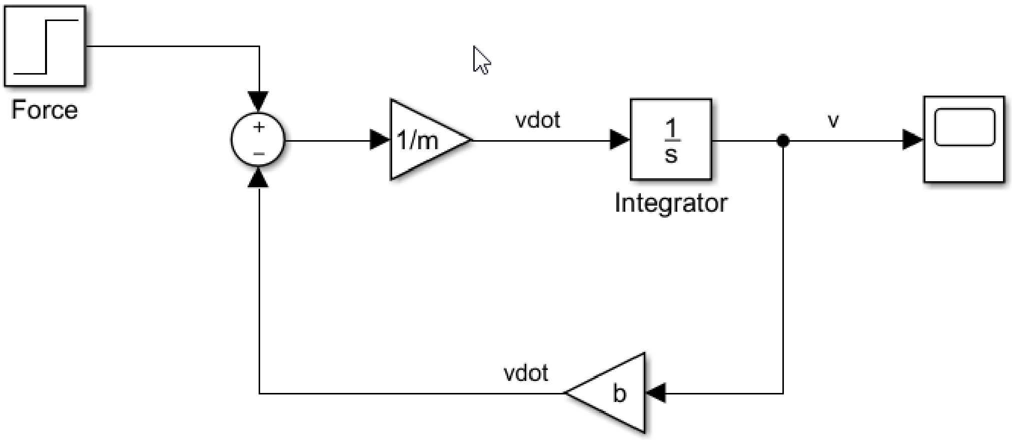


Figure 4: Finished model of the vehicle cruising system

For simulating the system, we anticipate that the velocity will reach a steady state value in 2 minutes. Thus, set the simulation time to 120 (secs).

If you run the simulation now, you will get an error message because you have not defined the values of ‘m’, ‘b’ and ‘u’.

You can simply type these values at the MATLAB command prompt and press Enter. Now come back to Simulink and run the model. The parameter values will be read from the MATLAB and fed into the Simulink model. After clicking on the Scope you will get the following output. Change the Display to highlight the title and Y-axis display.

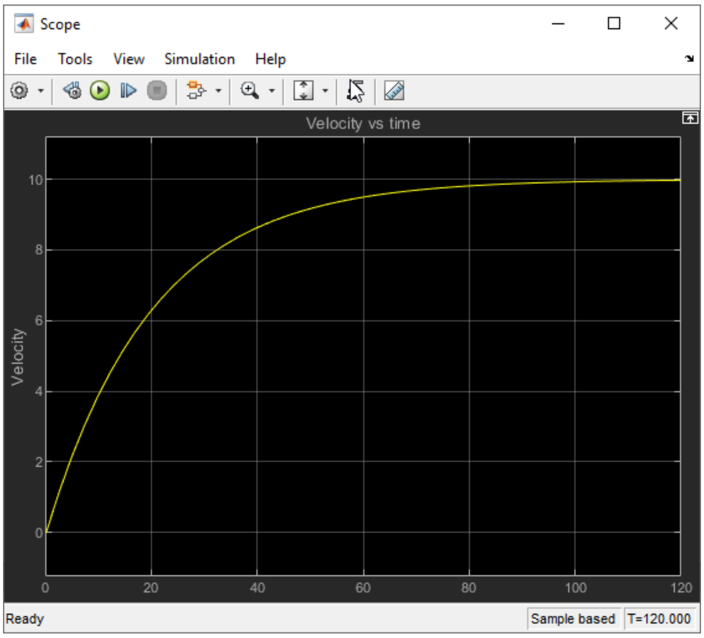


Figure 5: Final plot of velocity vs time

In how much time, the velocity becomes steady and what is the value in km/hr?

References:

Control Tutorials for Matlab and Simulink, Carnegie Melon University.
http://www.library.cmu.edu/ctms/ctms/simulink/examples/cruise/ccsim.htm

Exercise 3: (To be completed in the Class Room with the help of Tutor)

The water level in a tank is modelled by the first order differential equation as:

$$A\,\frac{dh}{dt} = bh - a\sqrt{2gh}$$

Where: A = 0.25 m^2; a = 7.5 x 10^{-6} m^2 and b = 5 x 10^{-5} m^2/s ; with g = 9.805 m/s^2

Solve the differential equation to find out the change in level of tank for 5 minutes of flow starting with h = 300 mm. (Ans: 318 mm)

Exercise 4:

Solving a second order system -- Mass-Spring-Dashpot System Simulation

In contrast to Exercise 2, if you want to find out the variation of displacement of any point on the chassis of the vehicle, then the system will become a second order differential system. The classic and the most well-known example of a second order dynamic system is the spring-mass-damper system shown in Fig 6. This system involves a mass 'm' connected simultaneously to a spring of stiffness 'k' and a damper with damping ratio 'c'. The differential equation that mathematically models this system is also based on Newton’s law of motion and described in equation (2):

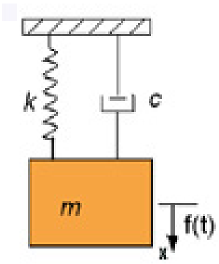


Figure 6

$m\,\ddot{x} + c\,\dot{x} + k\,x = f(t)$ ------------------ (2)

where *f(t)* is the excitation forcing function.

In this exercise we shall study the use of Simulink to simulate the response of this system to "unit step input" i.e. a step force with a final value of 1.

By re-arranging the above differential equation, an expression is written for the output variable ‘x’. The dot above ‘x’ represents (d/dt). It is evident that the expression needs to be integrated twice to evaluate the variation of x vs t.

$$\ddot{x} = \frac{1}{m}\left(f(t) - c\,\dot{x} - kx\right) \qquad (3)$$

Based on Eqn. 3, connect the blocks to finish the model diagram as shown in Figure 7. Use the options to flip (Ctrl + I) and rotate the blocks as necessary for appropriate positioning within the model. Like in the previous exercise, you need to begin from the integrator output which is 'Displacement' or 'x', but be careful that in this solution, 'x' will be obtained after double integration or after the second Integrator block.

The value of mass is 2.0; damping coefficient is 0.65 and stiffness being 0.6. Excitation force f(t) is modelled by the step input with step size as unity. You can set the parameters to symbolic values in the blocks and type their values on MATLAB command prompt for reading into Simulink.

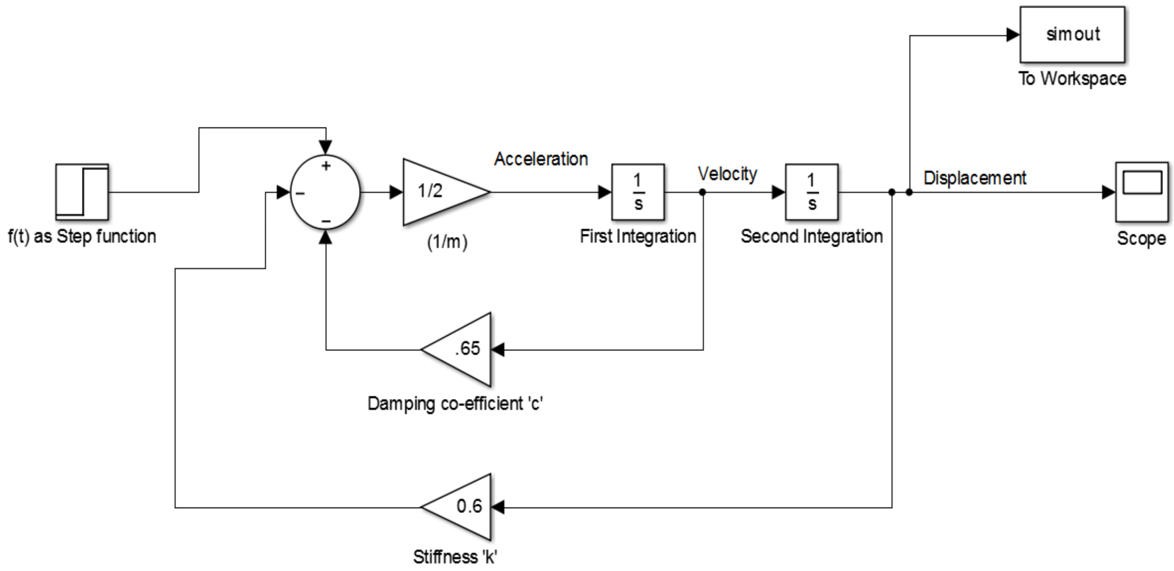


Figure 7

Run the simulation for 30 seconds by clicking on the 'Run' button (alternately you may use keyboard command CTRL+T). The output that will be visible in the Scope block should be similar to what is show in Figure 8.

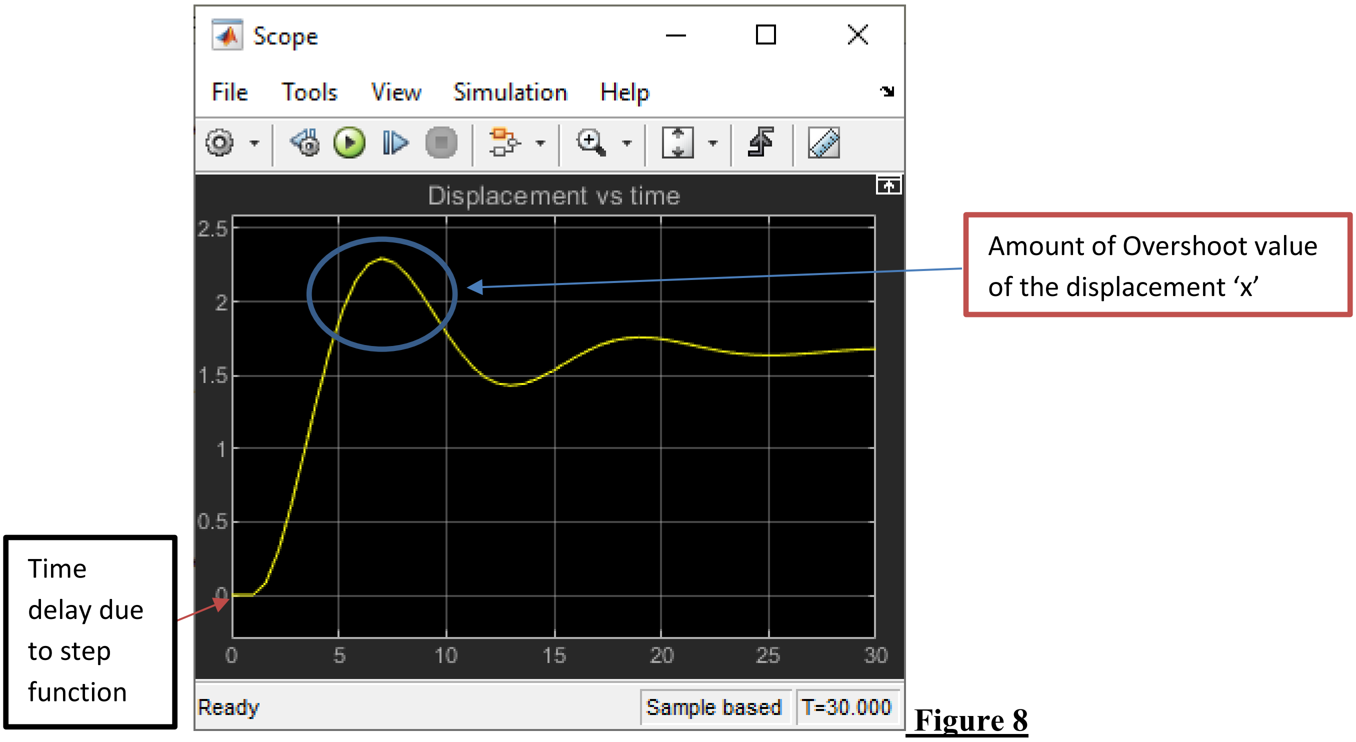


Figure 8

That's it! You have successfully modelled and simulated a **second-order under-damped dynamic system.**

Question: *After how much time the oscillations die down and the output becomes stable?*

Re-run the simulation with the values of the parameters set as: m = 2.0; c=0.9; k=1. What changes do you observe in the system's response?

To show that you may obtain different form of output, the model includes another block (in addition to the scope block) called **"simout".** The output from this block is sent into the MATLAB workspace. To illustrate how this block works, select a name for the output for example "Displacement" as the variable name in the block's parameter setting (double click on the "simout" block to bring up the parameter dialog window). You will see the appropriate variables created in MATLAB workspace which you can use for plotting and data analysis.

Exercise 5 (To be completed in the class with the help of Tutor)

For the spring-mass-damper system shown in Fig 9, find out the displacement 'y' of the mass after 50 seconds in response to an excitation force which changes from 0 to 5.2 N in 0.5 secs.

The given values are; mass = 20 kg; stiffness of spring= 2 N/m and damping coefficient= 4 N.s/m. **(Ans: 2.61 m)**

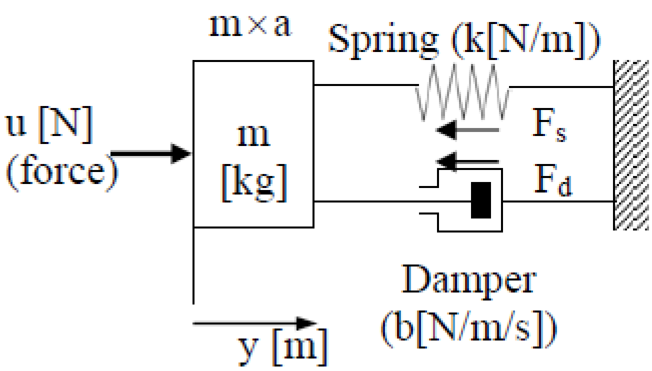


Figure 9: Second order system with its components

Exercise to be submitted on CANVAS:

A system is modelled by the second order differential equation and is modelled by the following equation:

$$\frac{d^2x}{dt^2} = 5\cos(2t) - 3\frac{dx}{dt} - 4x$$

Create a Simulink model to find out the variation of the variable 'x' with respect tot time and estimate the time at which the value becomes steady and the amount of overshoot.