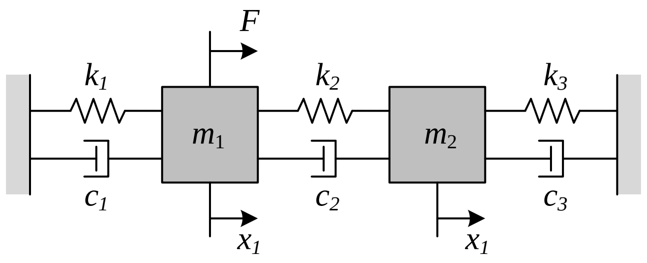
State-Space Models in MATLAB and Simulink (see Sec. 5.3)

ME 3050 – Dynamic Modeling and Controls

Consider the 2DOF system shown. We wish to place this system into state-space form and then solve for the response in MATLAB. The equations of motion are given by:



$$m_1\ddot{x}_1 + (c_1 + c_2)\dot{x}_1 - c_2\dot{x}_2 + (k_1 + k_2)x_1 - k_2 x_2 = f$$
$$m_2\ddot{x}_2 - c_2\dot{x}_1 + (c_2 + c_3)\dot{x}_2 - k_2 x_1 + (k_2 + k_3)x_2 = 0$$

The system can be placed in state-space form by first defining the following four state variables:

$$z_1 = x_1$$
$$z_2 = x_2$$
$$z_3 = \dot{x}_1 = \dot{z}_1$$
$$z_4 = \dot{x}_2 = \dot{z}_2$$

Substituting the state variables into the equations of motion eventually leads to the state –space form of the model given as:

$$\dot{z}_1 = z_3$$
$$\dot{z}_2 = z_4$$
$$\dot{z}_3 = \frac{1}{m_1}\left(-(k_1 + k_2)z_1 + k_2 z_2 - (c_1 + c_2)z_3 + c_2 z_4 + F\right)$$
$$\dot{z}_4 = \frac{1}{m_2}\left(k_2 z_1 - (k_2 + k_3)z_2 + c_2 z_3 - (c_1 + c_2)z_4\right)$$

In matrix form, this can be written as:

$$\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\\ \dot{z}_3\\ \dot{z}_4\end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-(k_1+k_2)}{m_1} & \frac{k_2}{m_1} & \frac{-(c_1+c_2)}{m_1} & \frac{c_2}{m_1} \\ \frac{k_2}{m_2} & \frac{-(k_2+k_3)}{m_2} & \frac{c_2}{m_2} & \frac{-(c_2+c_3)}{m_2}\end{bmatrix}\begin{bmatrix} z_1\\ z_2\\ z_3\\ z_4\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \frac{1}{m_1}\\ 0\end{bmatrix} f(t)$$

Let's suppose we are interested in analyzing the displacement of mass 1, the acceleration of mass 2, and the damping force exerted by c_1. The output equation becomes:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{k_2}{m_2} & \frac{-(k_2+k_3)}{m_2} & \frac{c_2}{m_2} & \frac{-(c_2+c_3)}{m_2} \\ 0 & 0 & c_1 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + 0 f(t)$$

Now that the system is in standard state-space form, we can enter it into MATLAB to perform our analysis. For this analysis assume:

$$m_1 = m_2 = 5kg, k_1 = k_2 = k_3 = 2N/m, c_1 = c_2 = c_3 = 1Ns/m$$

To begin, we should give our script a description and also clear all variables, graphs, and the command window:

```
%This script contains a comprehensive example of MATLAB commands for
%manipulating and solving state-space models.

clc
clear all
close all
```

Next, we can define the variables used in our script. In this case, define the masses, spring constants, and damping coefficients.

```
%Set the values of m,c,k
m1=5;
m2=5;
k1=2;
k2=2;
k3=2;
c1=1;
c2=1;
c3=1;
```

Now, we can place the system in state-space form by defining the **A**, **B**, **C**, and **D**, matrices and using the 'ss' function as follows:

```
%Define the system in state-space form
A=[0, 0, 1, 0; 0, 0, 0, 1; -(k1+k2)/m1, k2/m1, -(c1+c2)/m1, c2/m1; k2/m2, -
(k2+k3)/m2, c2/m2, -(c2+c3)/m2];
B=[0; 0; 1/m1; 0];
C=[1, 0, 0, 0; k2/m2, -(k2+k3)/m2, c2/m2, -(c2+c3)/m2; 0, 0, c1, 0];
D=0;
sys1=ss(A,B,C,D);
```

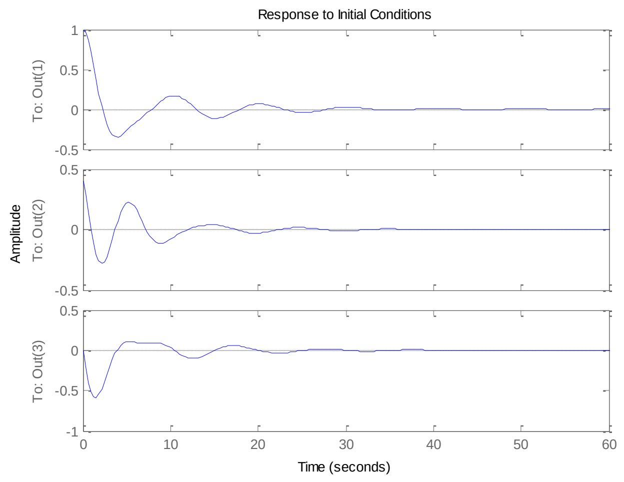
With the system placed in state-space form in MATLAB, we can now use the built-in analysis functions to analyze the response of the system to initial conditions, an impulse (with zero initial conditions), a step input (with zero initial conditions), or an arbitrary forcing function with non-zero initial conditions.

Let's investigate the response to an initial displacement on mass one by entering the following:

```
%Examine the initial condition response using the "initial" function.
```

```
%Assume the initial conditions are x1(0)=1, x2(0)=0, x1_dot(0)=0,
%x2_dot(0)=0;
figure;initial(sys1, [1, 0, 0, 0])
```

MATLAB plots the following response, where the top graph is the displacement of mass 1, the middle graph is the acceleration of mass 2, and the bottom graph is the damping force exerted by c_1:

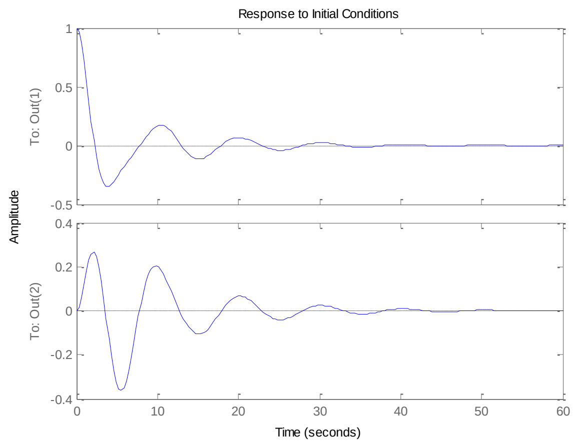


In order to more clearly visualize the system, let's just focus on analyzing the displacement of the two masses. In order to do so, we must redefine our output equation, i.e. the **C**, and **D** matrices and "rebuild" our system in MATLAB as follows:

```
C1=[1, 0, 0, 0; 0, 1, 0, 0];
sys2=ss(A,B,C1,D);
```

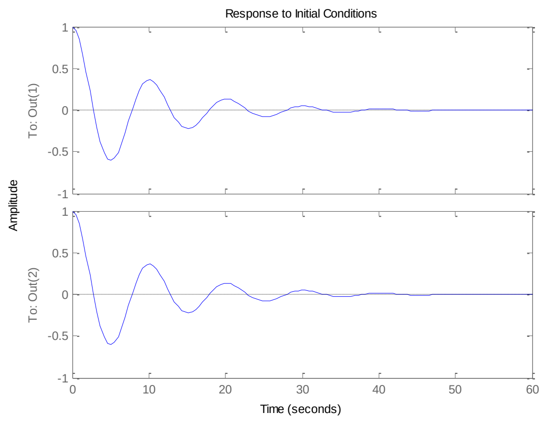
Now, let's look at the response to an initial displacement on mass 1 again:

```
figure;initial(sys2, [1, 0, 0, 0])
```



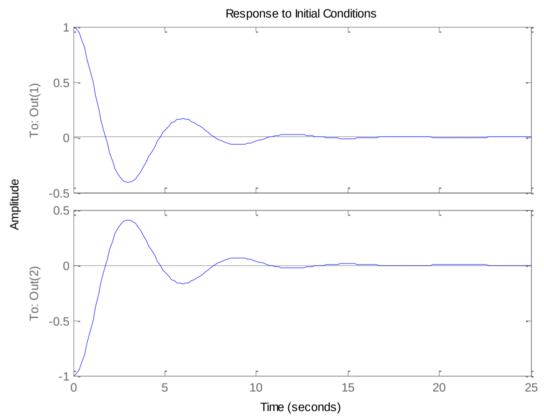
Now, what if the two masses are displaced equally and let go together at the same time?

```
figure;initial(sys2, [1, 1, 0, 0])
```



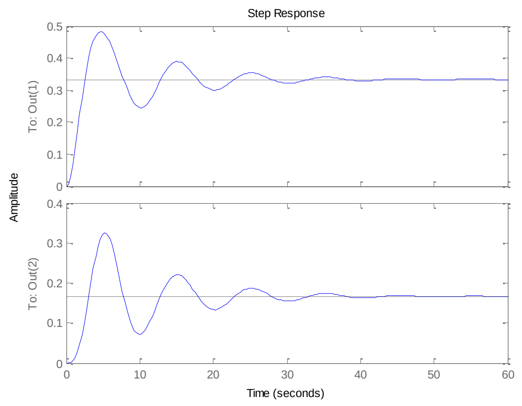
Finally, what if both masses are displaced the same amount, but in opposite directions?

```
figure;initial(sys2, [1, -1, 0, 0])
```



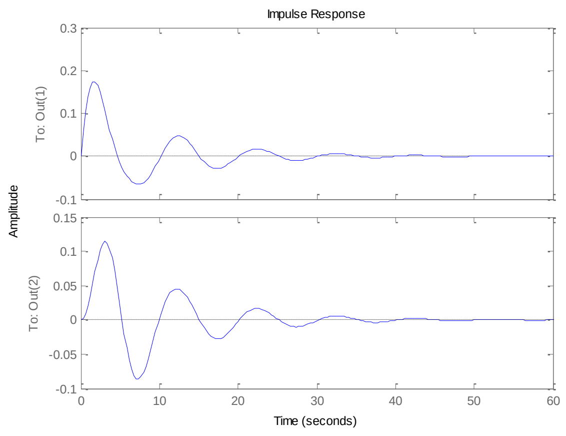
Now, what if we wish to investigate the response of the system to a step input? This is accomplished by simply entering:

```
figure;step(sys2)
```



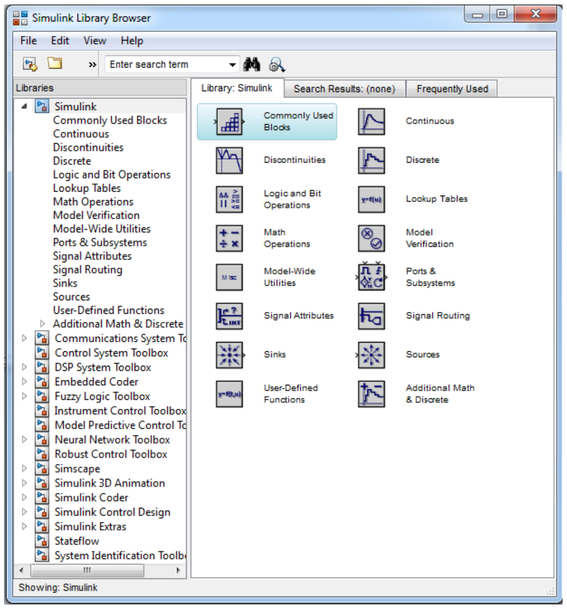
Finally, we can find the impulse response using the “impulse” function as follows:

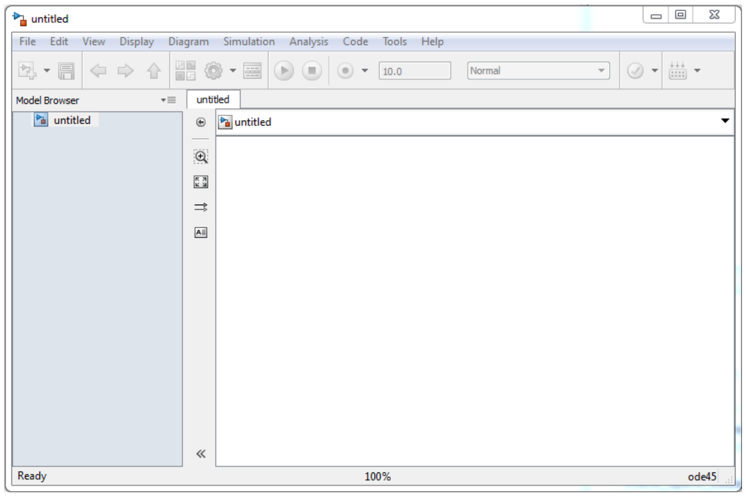
```
figure;impulse(sys2)
```



Simulink

In addition to using the built-in MATLAB functions to analyze this system, one can also use Simulink, which is an add-on analysis tool to MATLAB. To launch Simulink, simply enter 'simulink' in the command window. Once launched, create a new file and you should have the following:

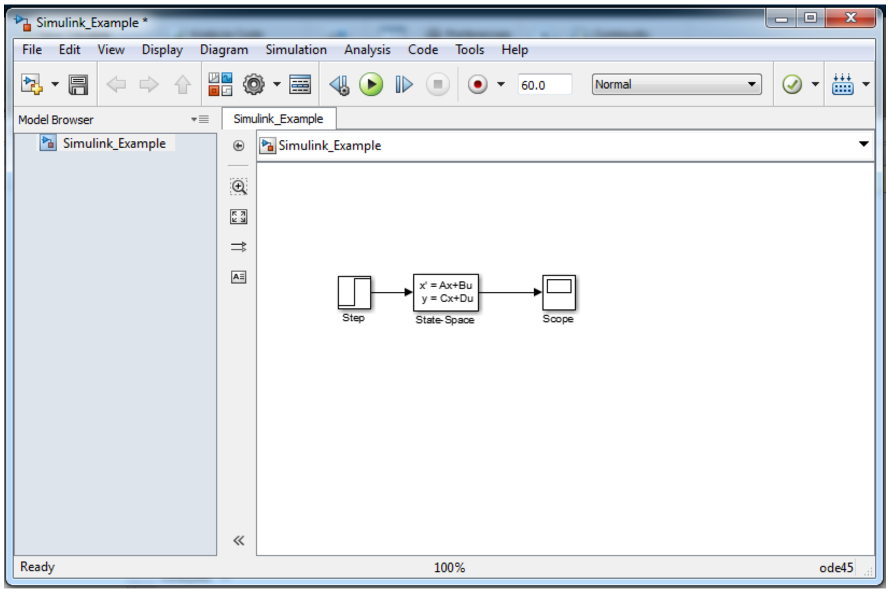


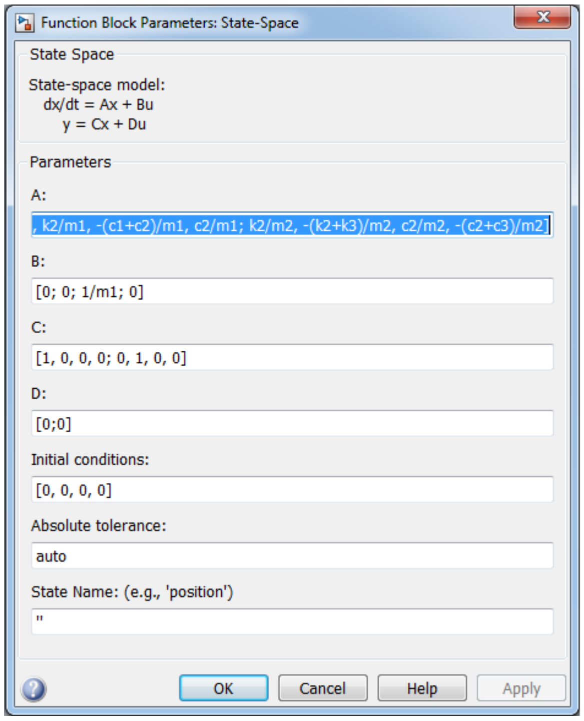

untitled
File Edit View Display Diagram Simulation Analysis Code Tools Help
10.0
Normal
Model Browser
untitled
Ready
100%
ode45

Now, we can build the model by:

1. Selecting a step input from the "sources" category and setting the step time to 0, initial value to 0, the final value to 1 and the sample time to 0.
2. Inserting a state-space model from the "continuous" category and entering the **A**, **B**, **C**, and **D** matrices and the initial conditions (note, that the variables must exist in the MATLAB workspace in order to pull them into Simulink).
3. Inserting a scope from the "sinks" category in order to view the data.
4. Changing the simulation time to 60 seconds.
5. Pressing the "play" button.

We should end up with:




Function Block Parameters: State-Space

State Space

State-space model:
dx/dt = Ax + Bu
y = Cx + Du

Parameters

A:
, k2/m1, -(c1+c2)/m1, c2/m1; k2/m2, -(k2+k3)/m2, c2/m2, -(c2+c3)/m2]

B:
[0; 0; 1/m1; 0]

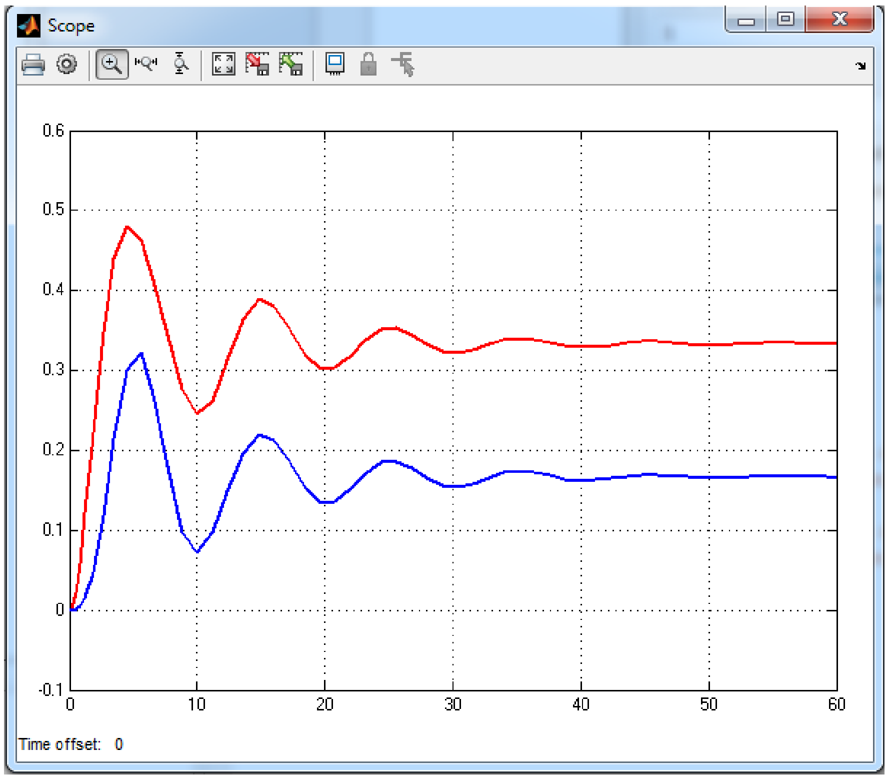
C:
[1, 0, 0, 0; 0, 1, 0, 0]

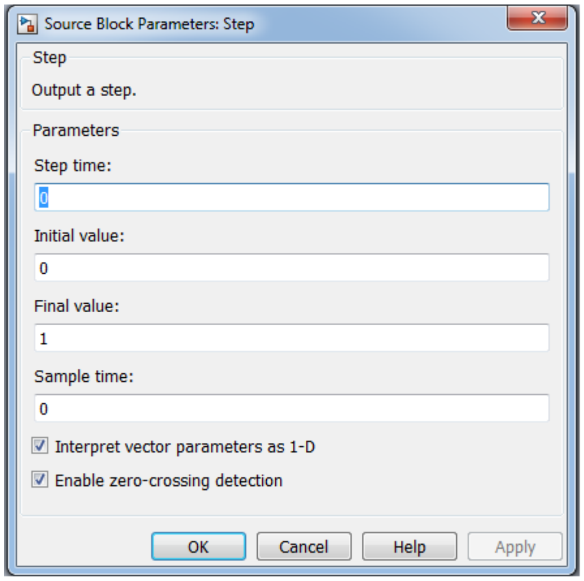
D:
[0;0]

Initial conditions:
[0, 0, 0, 0]

Absolute tolerance:
auto

State Name: (e.g., 'position')
''




Source Block Parameters: Step

Step

Output a step.

Parameters

Step time:
0

Initial value:
0

Final value:
1

Sample time:
0

☑ Interpret vector parameters as 1-D

☑ Enable zero-crossing detection

Finally, we can observe that the results obtained here are identical to those obtained when finding the step response in MATLAB using the “step” function.