(Ch. 5) Alternative Model Forms: Transfer Functions, Block Diagrams, State Space Models

- In addition to writing standard Equations of Motion, there are other forms in which we can express the governing equations of the system, i.e. alternative model forms. These alternative forms provide useful ways to analyze the system.

Transfer Functions

Recall from Ch.2, we can analyze the forced response by taking the initial conditions to be zero temporarily + writing the transfer function.

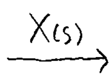
$$T(s) = \frac{X(s)}{F(s)}, \quad ICs = 0$$

(5.1) Block Diagrams

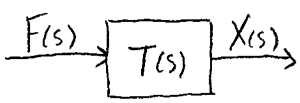
Transfer Function can be used to create a visual representation of the system, called a Block Diagram. Block Diagrams show cause + effect relations between components.

Block Diagram $\Longleftrightarrow$ Transfer Function

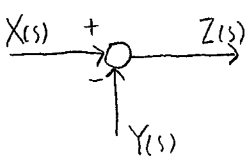
Four Basic Symbols:



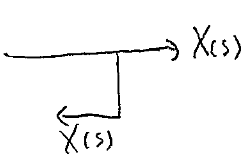
Arrow: represents variable, shows direction of cause-effect relation



Block: represents input-output relation of a Transfer Function

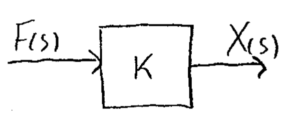


Circle or Summer: addition and/or subtraction depending on sign



Takeoff Point: splits off variable without changing value

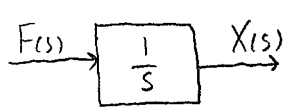
Block Multipliers:



Gain or Multiplier Block

$X(s) = K F(s)$

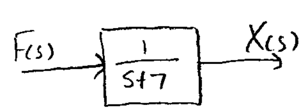
$x(t) = K F(t)$



Integrator Block

$X(s) = \frac{F(s)}{s}$

$x(t) = \int F(t)\,dt$



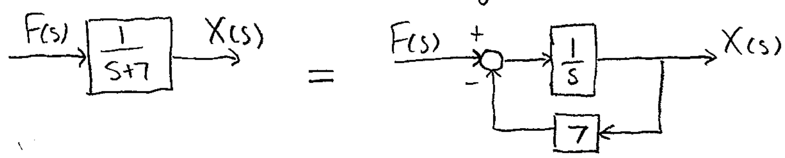
Transfer Function Block

$X(s) = \frac{F(s)}{s+7}$

$X(s)(s+7) = F(s) \Rightarrow \dot{x} + 7x = F(t)$

Equivalent Block Diagrams:

(Negative Feedback)



$$\frac{F(s) - 7X(s)}{s} = X(s) \quad \Rightarrow \quad F(s) = sX(s) + 7X(s)$$

$$F(s) = (s+7)X(s)$$

Series Elements:

$B(s) = F(s)T_1(s) \qquad X(s) = F(s)T_1(s)T_2(s) \qquad\qquad X(s) = F(s)\,T_1(s)\,T_2(s)$

Feedback Loops:

$A(s) = F(s) - B(s)$

$B(s) = X(s)H(s) \Rightarrow X(s) = G(s)\left[F(s) - X(s)H(s)\right]$

$X(s) = G(s)A(s) \qquad X(s) = \dfrac{G(s)F(s)}{1 + G(s)H(s)}$

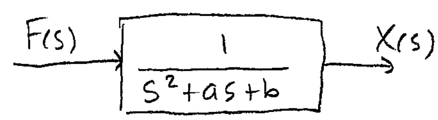
★ Note: This is a useful formula to remember for simplifying a negative feedback loop.

Rearranging Block Diagrams.

$\ddot{x} + a\dot{x} + bx = f(t)$

$X(s)(s^2 + as + b) = F(s)$

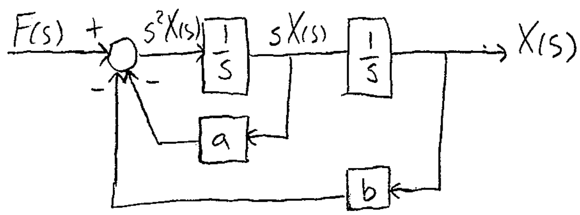
$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + as + b}$$



$s^2X(s) + asX(s) + bX(s) = F(s)$

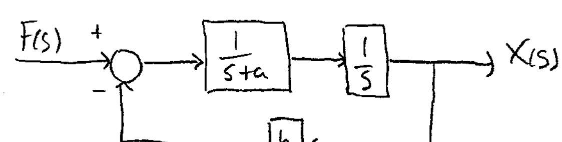
$s^2X(s) = F(s) - asX(s) - bX(s)$

$X(s) = \frac{1}{s}\left(\frac{1}{s}\left[F(s) - asX(s) - bX(s)\right]\right)$



Eliminate inner feedback loop:

Formula: $\dfrac{1/s}{1 + \frac{1}{s}a} = \dfrac{1}{s+a}$



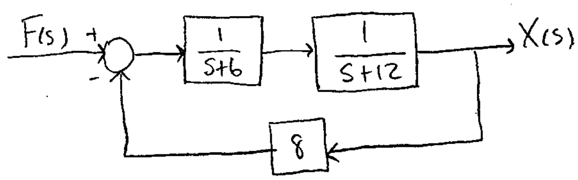
$(s+a)sX(s) + bX(s) = F(s)$

$X(s) = \left[F(s) - bX(s)\right]\frac{1}{s}\,\frac{1}{s+a}$

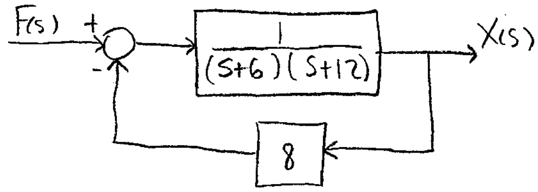
Block Diagrams ⇒ Transfer Functions

Determine TFs from Block Diagrams below

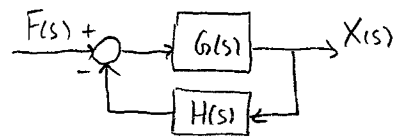
a)



Top two blocks can be combined:

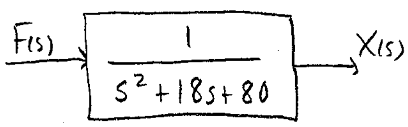


Use feedback loop simplification:



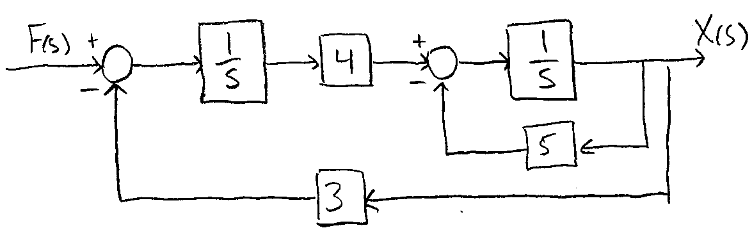
$$X(s) = \frac{G(s)}{1 + G(s)H(s)} F(s)$$

$$X(s) = \frac{\frac{1}{(s+6)(s+12)}}{1 + \frac{8}{(s+6)(s+12)}} F(s) = \frac{1}{(s+6)(s+12) + 8} F(s) = \frac{1}{s^2 + 18s + 80} F(s)$$

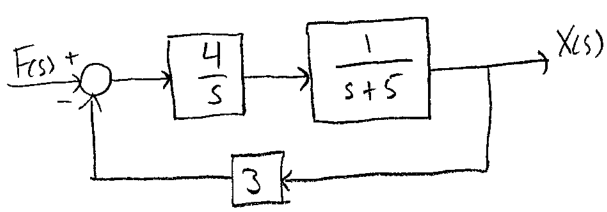


So the TF is: $\boxed{T(s) = \frac{X(s)}{F(s)} = \frac{1}{s^2+18s+80}}$

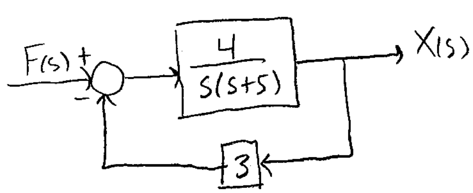
b)



Add "$\frac{1}{s}$" + "4" blocks and simplify the inner feedback loop:



Combine "$\frac{4}{s}$" + "$\frac{1}{s+5}$" blocks:



Simplify the feedback loop:

$$X(s) = \frac{\frac{4}{s(s+5)}}{1+\frac{12}{s(s+5)}}F(s) = \frac{4}{s(s+5)+12}F(s) = \frac{4}{s^2+5s+12}F(s)$$

$$F(s) \longrightarrow \boxed{\frac{4}{s^2+5s+12}} \longrightarrow X(s)$$

So the TF is: $\boxed{T(s) = \frac{X(s)}{F(s)} = \frac{4}{s^2+5s+12}}$

(5.2) State-Space Models (State-Variable Models)

The purpose of state-space models is to replace a single higher-order differential equation with a set of multiple coupled first order equations. The main advantages include the ability to write and analyze the system in matrix form, and the fact that many software packages (i.e. MATLAB) have powerful analysis tools for state-space models.

In order to write a system in state-space form, it must be expressed as a set of first order equations (i.e. single dot: $\dot{x}, \dot{y}$, etc.)

Example: $m\ddot{x} + c\dot{x} + kx = f$

If we define two new variables: $x_1 = x$, $x_2 = \dot{x}$ (these are the <u>state variables</u>)

These imply $\dot{x}_1 = x_2$ (this is the first of our coupled first-order equations)

Then we can substitute x_1 + x_2 into the EOM:

$$m\dot{x}_2 + c x_2 + k x_1 = f$$

(could also pick $m\dot{x}_2 + c\dot{x}_1 + kx_1 = f$, but then we would have 2 derivatives, so this won't work)

Solve for $\dot{x}_2$

$$\dot{x}_2 = \frac{1}{m}(f - kx_1 - cx_2)$$

(this is the 2nd coupled first-order equation)

So our state-space form is as follows

$$\dot{x}_1 = x_2$$

$$\underbrace{\dot{x}_2}_{\text{first derivatives of our state variables}} = \underbrace{\frac{1}{m}(f - kx_1 - cx_2)}_{\text{equations in terms of our state variables (no derivatives)}}$$

Furthermore, we can express the state-space method in matrix form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f(t)$$

In compact form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B} f(t)$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{K}{m} & -\frac{c}{m} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ \dot{x}_1 \end{bmatrix}$$

Standard Form of the State Equation

For a system with "n" state variables + "m" inputs, the standard form is:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

where $\mathbf{x}$ + $\mathbf{u}$ are column vectors of the state variables and inputs, respectively.

Common nomenclature:

- $\mathbf{x}$ is the <u>state vector</u> (dimensions $n \times 1$) (rows x columns)
- $\mathbf{A}$ is the <u>state matrix</u> (dimensions $n \times n$) → note, this is a square matrix
- $\mathbf{B}$ is the <u>input matrix</u> (dimensions $n \times m$)
- $\mathbf{u}$ is the <u>input vector</u> (dimensions $m \times 1$)

We must also define an output equation, which contains information about the variables we wish to solve for. The output equation for "p" outputs has the form:

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

where:

- $\mathbf{y}$ is the <u>output vector</u> (dimensions $p \times 1$)
- $\mathbf{C}$ is the <u>state output matrix</u> (dimensions $p \times n$)
- $\mathbf{D}$ is the <u>control output matrix</u> (dimensions $p \times m$)

Let's consider the previous example and find the output equation if we are simply interested in finding the displacement of the mass, which is equal to X_1.

Our state equation is

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f(t)$$

$$\dot{\mathbf{x}} = \mathbf{A} \quad \mathbf{x} + \mathbf{B} \mathbf{u}$$

Our state variables are:

$X_1 = X$

$X_2 = \dot{X}_1 = \dot{X}$

So, we wish to solve for the displacement, x i.e. X_1. Since we only want one output, $p=1$. We can define the output equation as:

$$X_1 = [1 \ 0] \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + 0 \cdot f(t)$$

$$\mathbf{y} = \mathbf{C} \quad \mathbf{x} + \mathbf{D} \mathbf{u}$$

Now, what if we wanted to solve for the spring force and acceleration of the mass?

We need to define these two quantites in terms of the state variables and inputs.

- Spring force: $f = KX = K \underset{\text{state variable}}{X_1}$
- Acceleration: $a = \ddot{X} = \dot{X}_2$ derivative of state variable, but we need to solve in terms of state variables, not derivatives of them.

So, let's go back to the original equation

$$m\ddot{X} + c\dot{X} + KX = f$$

Solve for acceleration ($\ddot{x}$)

$$\ddot{X} = \frac{1}{m}(f - KX_1 - cX_2)$$

state variables

now this is in terms of the state variables

So our two outputs can be written as

$$y_1 = KX_1$$
$$y_2 = -\frac{K}{m}X_1 - \frac{c}{m}X_2 + \frac{f}{m}$$

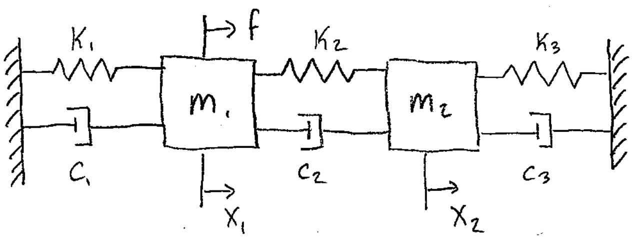
Finally, we can write our output equation

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} K & 0 \\ -\frac{K}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f(t)$$

$$\mathbf{y} = \mathbf{C} \quad \mathbf{x} + \mathbf{D} \mathbf{u}$$

(5.3) State-Space with MATLAB

Consider the following 2DOF Mass-Spring-Damper system. We will write the system in state-space and then analyze in MATLAB.



We solved this previously, the EOMs are:

$$m_1 \ddot{X}_1 + (C_1 + C_2)\dot{X}_1 - C_2 \dot{X}_2 + (K_1 + K_2) X_1 - K_2 X_2 = f$$

$$m_2 \ddot{X}_2 - C_2 \dot{X}_1 + (C_2 + C_3)\dot{X}_2 - K_2 X_1 + (K_2 + K_3) X_2 = 0$$

Let's define the four state variables (recall, we need to define two per each 2^{nd} order equation)

$$Z_1 = X_1$$
$$Z_2 = X_2$$
$$Z_3 = \dot{X}_1 = \dot{Z}_1 \qquad \Rightarrow \qquad \left.\begin{aligned} \dot{Z}_1 &= Z_3 \\ \dot{Z}_2 &= Z_4 \end{aligned}\right\} \text{These are our first two state equations}$$
$$Z_4 = \dot{X}_2 = \dot{Z}_2$$

Now let's substitute the state variables into the EOMs (being careful to only include one derivative, i.e. $\dot{z}_3$ + $\dot{z}_4$)

$$m_1 \dot{Z}_3 + (C_1 + C_2) Z_3 - C_2 Z_4 + (K_1 + K_2) Z_1 - K_2 Z_2 = f$$

$$m_2 \dot{Z}_4 - C_2 Z_3 + (C_2 + C_3) Z_4 - K_2 Z_1 + (K_2 + K_3) Z_2 = 0$$

Solve for $\dot{Z}_3$ + $\dot{Z}_4$

$$\dot{Z}_3 = \frac{1}{m_1}\left(-(K_1 + K_2) Z_1 + K_2 Z_2 - (C_1 + C_2) Z_3 + C_2 Z_4 + f\right)$$

$$\dot{Z}_4 = \frac{1}{m_2}\left(K_2 Z_1 - (K_2 + K_3) Z_2 + C_2 Z_3 - (C_1 + C_2) Z_4\right)$$

We can express the state-space system in Matrix form:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-(K_1+K_2)}{m_1} & \frac{K_2}{m_1} & \frac{-(C_1+C_2)}{m_1} & \frac{C_2}{m_1} \\ \frac{K_2}{m_2} & \frac{-(K_2+K_3)}{m_2} & \frac{C_2}{m_2} & \frac{-(C_2+C_3)}{m_2} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_1} \\ 0 \end{bmatrix} f(t)$$

Let's suppose we are only interested in the displacements of the masses. Then our two outputs are

$$y_1 = x_1 = z_1$$
$$y_2 = x_2 = z_2$$

So our output equation is

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + 0 \cdot f(t)$$

Now, what if we want to analyze the displacement of mass 1, the acceleration of mass 2, and the damping force exerted by C_1.

- Displacement of m_1: $y_1 = x_1 = z_1$
- Acceleration of m_2: $y_2 = \ddot{x}_2 = \dot{z}_4$ ← <u>derivative</u> of state variable, but we need the outputs in terms of the state variables + outputs.

  Going back to the EOM for mass 2 and solving for acceleration

  $$y_2 = \ddot{x}_2 = \frac{1}{m_2}\left(K_2 z_1 - (K_2+K_3)z_2 + C_2 z_3 - (C_2+C_3)z_4\right)$$

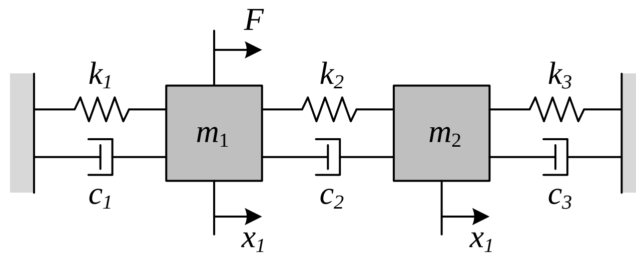
- Damping force of C_1: $y_3 = C_1 \dot{x}_1 = C_1 z_3$

So, our output equation is now

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{K_2}{m_2} & \frac{-(K_2+K_3)}{m_2} & \frac{C_2}{m_2} & \frac{-(C_2+C_3)}{m_2} \\ 0 & 0 & C_1 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + 0 \cdot f(t)$$

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:

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

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);
```

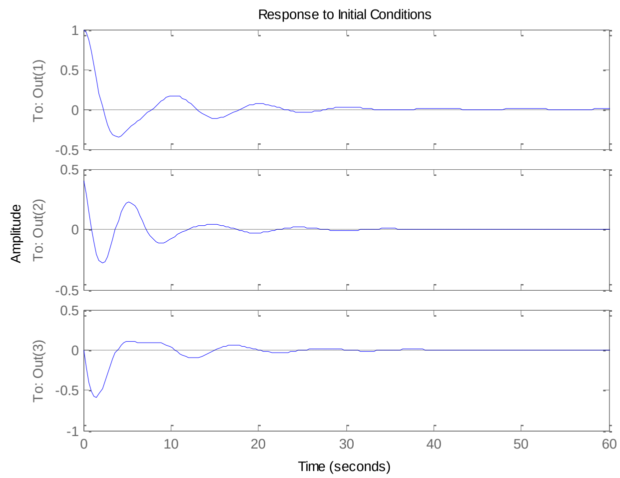
With the system placed in state-space form in MATLAB, we can now use the built-in analysis functions to analyze the response of the system to initial conditions, an impulse (with zero initial conditions), a step input (with zero initial conditions), or an arbitrary forcing function with non-zero initial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

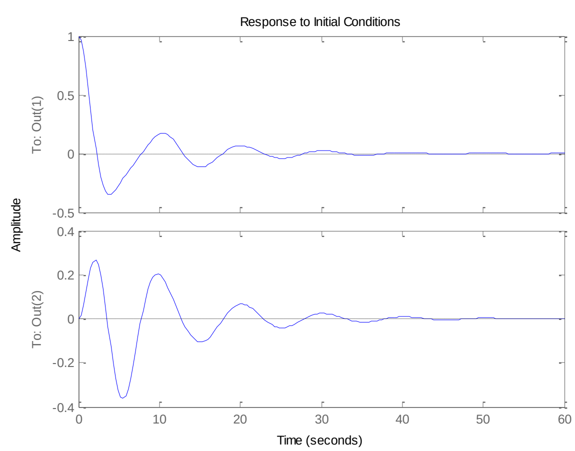


In order to more clearly visualize the system, let's just focus on analyzing the displacement of the two masses. In order to do so, we must redefine our output equation, i.e. the **C**, and **D** matrices and "rebuild" our system in MATLAB as follows:

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

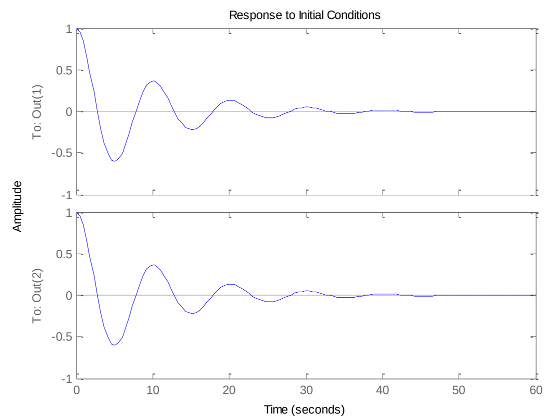
Now, let's look at the response to an initial displacement on mass 1 again:

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



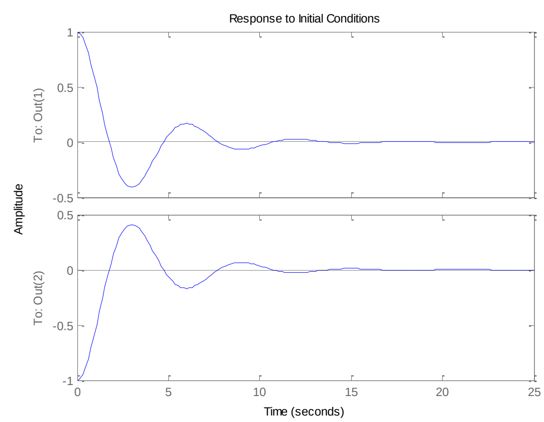
Now, what if the two masses are displaced equally and let go together at the same time?

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



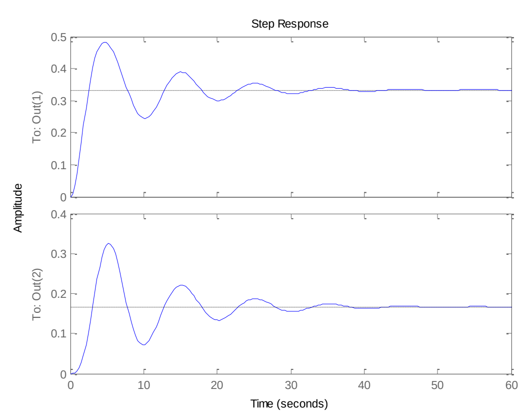
Finally, what if both masses are displaced the same amount, but in opposite directions?

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



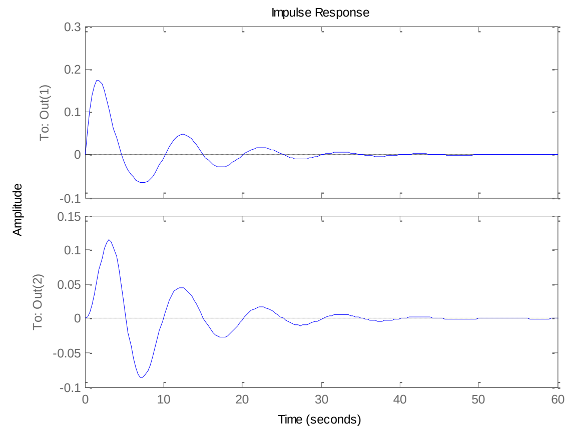
Now, what if we wish to investigate the response of the system to a step input? This is accomplished by simply entering:

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



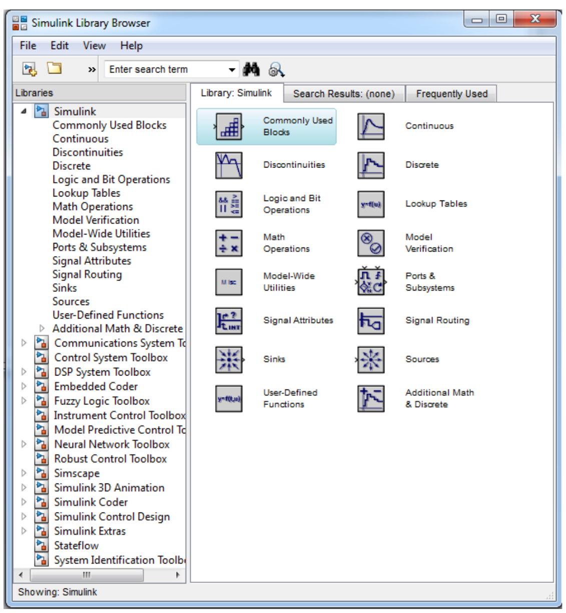
Finally, we can find the impulse response using the “impulse” function as follows:

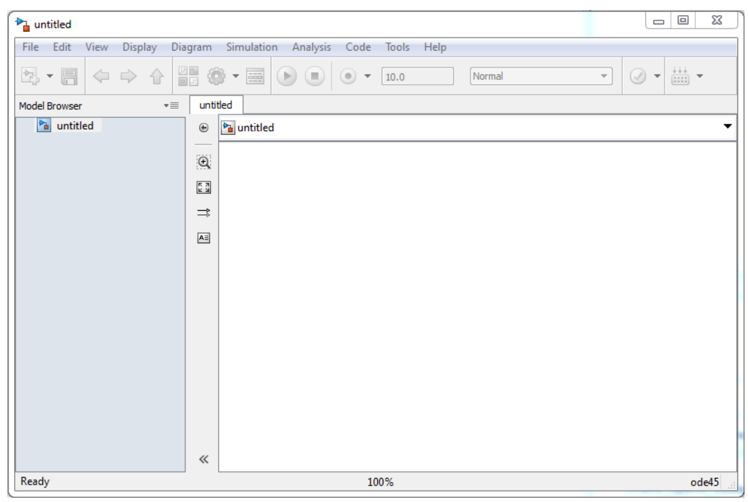
```
figure;impulse(sys2)
```



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:

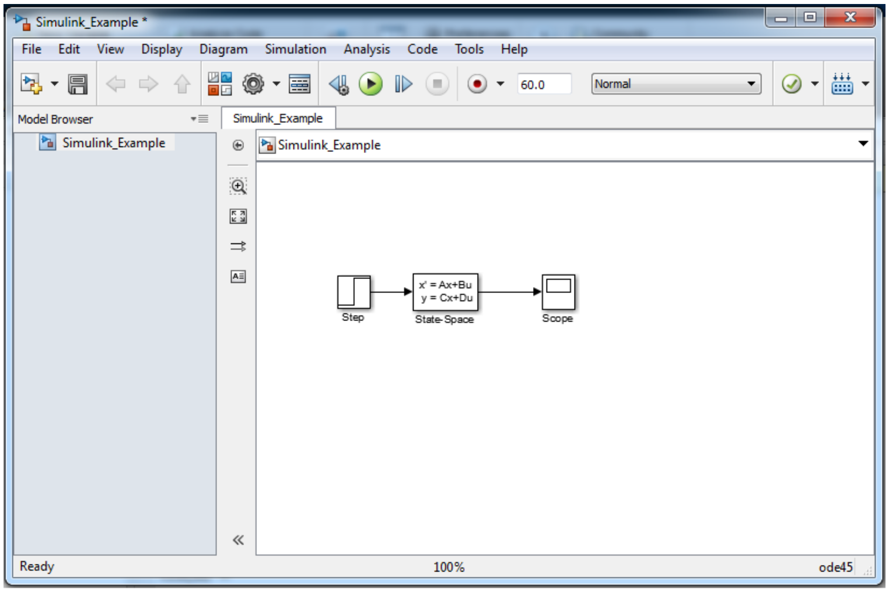


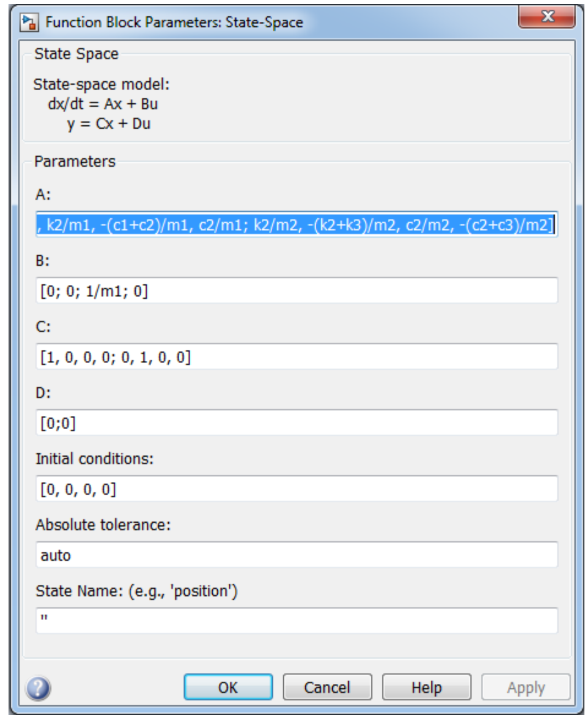


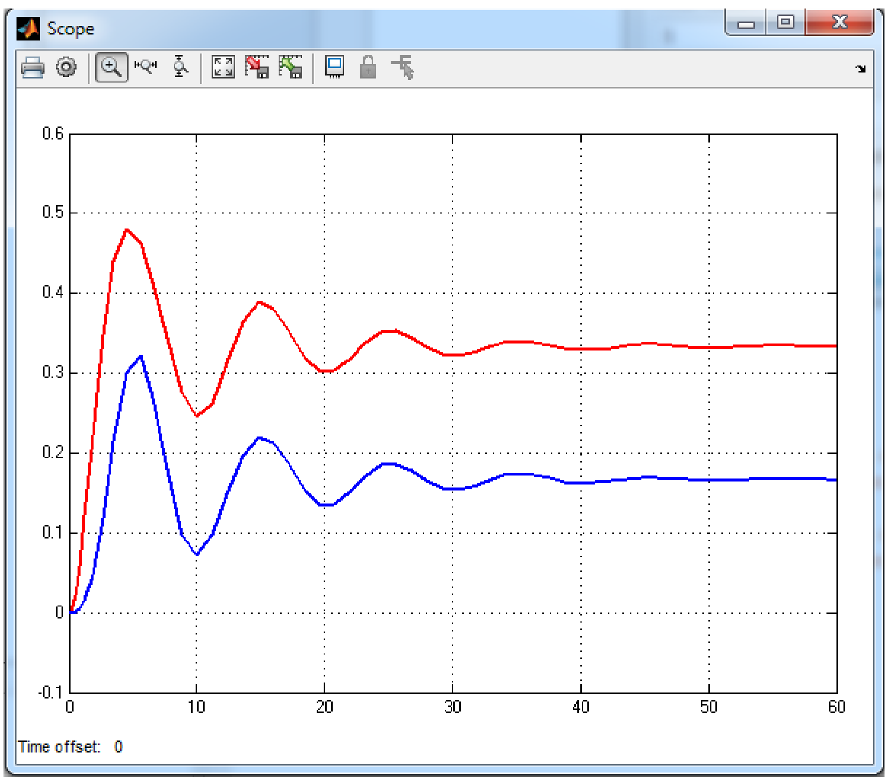
Now, we can build the model by:

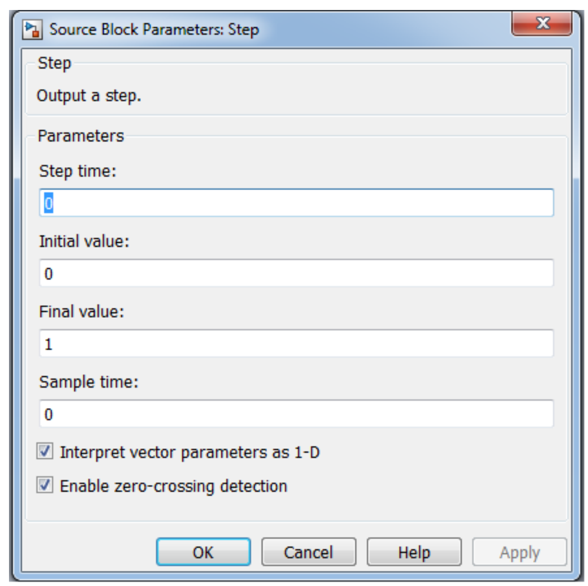
1. Selecting a step input from the “sources” category and setting the step time to 0, initial value to 0, the final value to 1 and the sample time to 0.
2. Inserting a state-space model from the “continuous” category and entering the **A**, **B**, **C**, and **D** matrices and the initial conditions (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:









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.