Date: 07-08-2019

Experiment 2

Title of the Experiment:

- MATLAB Task
  - Linear System Representation
  - Signal Processing Toolbox (tf2zpk and zpk2tf)
  - Time domain analysis – Linear Analysis (Basic Functions)
  - Model Interconnection (Basic Functions)
    - Create Transfer Function using control system toolbox
    - Obtain poles, zeros and gain of Transfer Function
    - Recreate the Transfer Function using the poles, zeros and gain obtained above
    - Plot the pole zero map
    - Consider the transfer function to be $G(s) = \frac{s+2}{s^2+s+2}$
- SIMULINK Task
  - Construct a Simulink model to calculate response of mass spring system. The input forces increase from $0\ to\ 5N\ at\ t\ =\ 1\ sec. m\ =\ 2\ Kg, k\ =\ 16\ N/m\ and\ B\ =\ 4NS/m.$

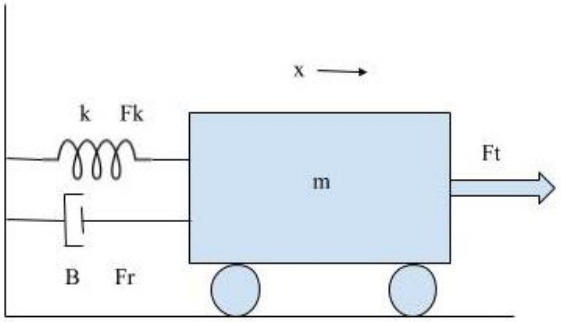

Fig 2(a): Spring Mass System

Objective of the Experiment: To study various MATLAB functions for finding zeros, poles and gains of transfer function and simulate the same system using SIMULINK.

Software: MATLAB 2018

Equations/Formula:

Given transfer function of a system is $G(s) = \frac{s+2}{s^2+s+2}$

Theory:

- **Transfer Function:** a transfer function (also known as system function or network function) of an electronic or control system component is a mathematical function which theoretically models the device's output for each possible input in its simplest form, this function is a two-dimensional graph of an independent scalar input versus the dependent scalar output, called a transfer curve or characteristic curve. Transfer functions for components are used to design and analyse systems assembled from components, particularly using the block diagram technique, in electronics and control theory.

If we have an input function of X(s), and an output function Y(s), we define the transfer function G(s) to be

$$G(s) = \frac{Y(s)}{X(s)}$$

- **tf**: Creates a transfer function from the numerator and the denominator
  Syntax: `sys = tf ([1 2 1], [1 3 1])`

- **tf2zpk**: Find the zeros z poles p and gains k of the given system with numerator and denominator.
  Syntax: `[z,p,k] = tf2zpk([1 2 1],[1 3 1])`

- **zpk2tf**: finds the transfer function given zeros z poles p and gains k of the system
  Syntax: `[b,a] = zpk2tf ([1 1], [1 3 1],1)`

- **pzmap**: Creates a Pole-Zero plot of a dynamic system. The sys keyword in the syntax implies a tf object created using the tf command.
  Syntax: `pzmap(sys)`

- **impulse**: Impulse response plot of dynamic system
  Syntax: `impulse(sys,T)`

- **step**: Step response of dynamic system
  Syntax: `step(sys,T)`

- **pzmap**: Pole-zero plot of dynamic system
  Syntax: `pzmap(sys)`

MATLAB Code and Results:

1. **Creating Transfer Function:**

```
clc;
clear all;
close all;
% making direct function
s = tf('s');
H = (s+2)/(s^2+s+2);
% using coefficients
num = [0 1 2];
den = [1 1 2];
h = tf(num,den);
```

2. **Obtain poles, zeros and gain of a Transfer Function:**

```
% poles, zeros and gain
[z,p,k] = tf2zpk(num,den);
```

3. **Recreate the Transfer Function using the poles, zeros and gain obtained above:**

```
% regenrate transfer function
[b,a] = zp2tf(z,p,k);
```

4. **Plot the pole zero map:**

```
%plotting pole-zeros, step response and Impulse response
pzmap(h);
step(h);
impulseplot(h);
```

Result:

```
Command Window
>> h

h =

     s + 2
  -----------
  s^2 + s + 2

Continuous-time transfer function.

>> H

H =

     s + 2
  -----------
  s^2 + s + 2

Continuous-time transfer function.
```

(1)

```
Command Window
>> z

z =

    -2

>> p

p =

  -0.5000 + 1.3229i
  -0.5000 - 1.3229i

>> k

k =

     1
```

(2)

```
Command Window
>> b

b =

     0     1     2

>> a

a =

    1.0000    1.0000    2.0000
```

(3)

Fig 2(b): 1 – shows generated Transfer Function
2 – shows the generated values of zeros, poles and gain of a TF
3 – shows the generated TF from zeros, poles and gain

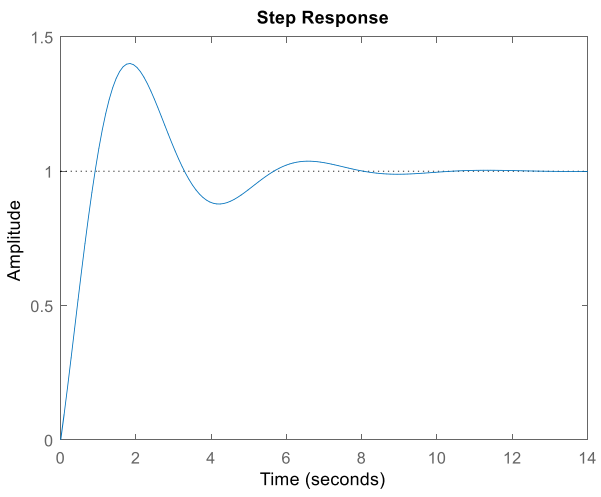

Fig 2(c): Step response of the system

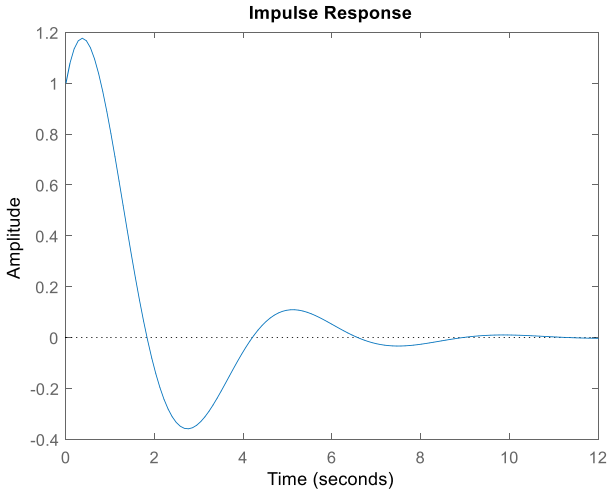

Fig 2(d): Impulse response of the system

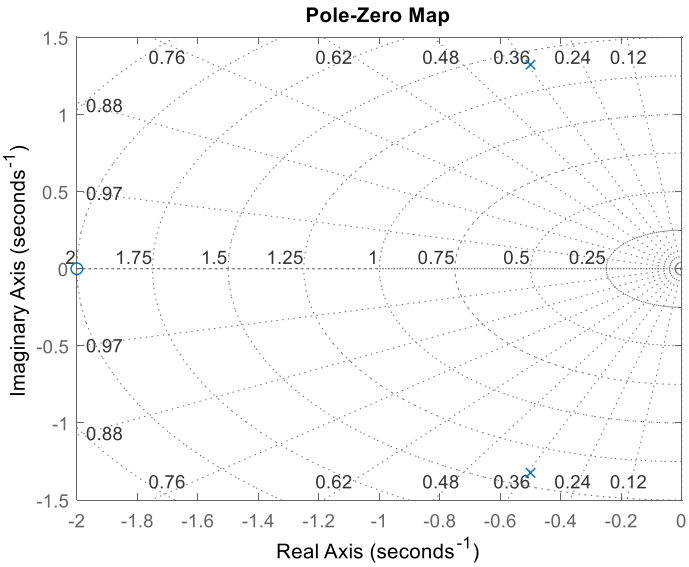

Fig 2(e): Pole Zero map

Simulink:

Balancing the force equations we get $F(t) = m\frac{d^2x(t)}{dt^2} + B\frac{dx(t)}{dt} + kx(t)$. After substituting values and taking Laplace transform on both sides we get:

$$G(s) = \frac{1}{2s^2 + 4s + 4}$$

Simulink Diagram:

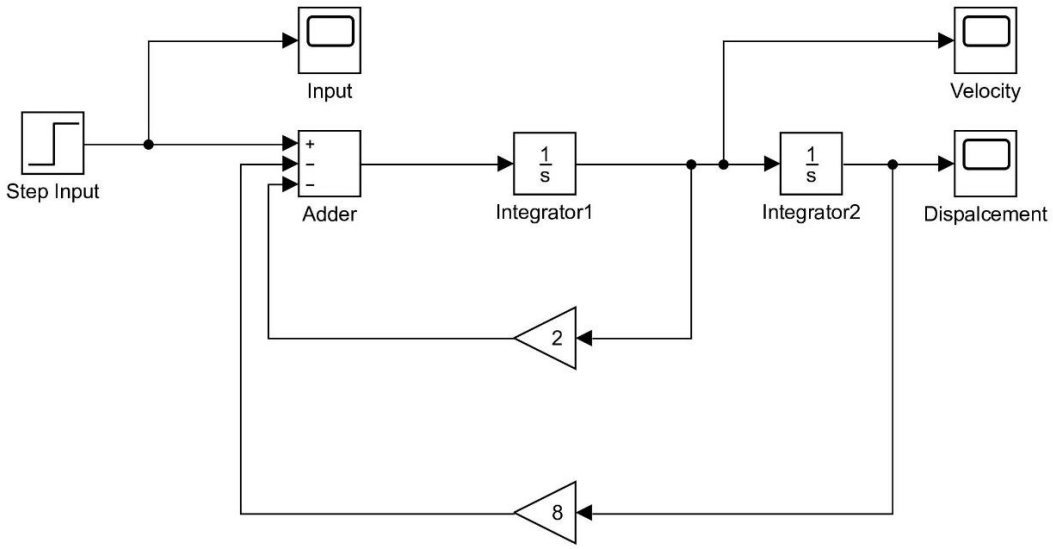

Fig 2(f): Simulink Diagram representing the spring mass system

Output: The following figures show the response of system on displacement and velocity of spring mass system.

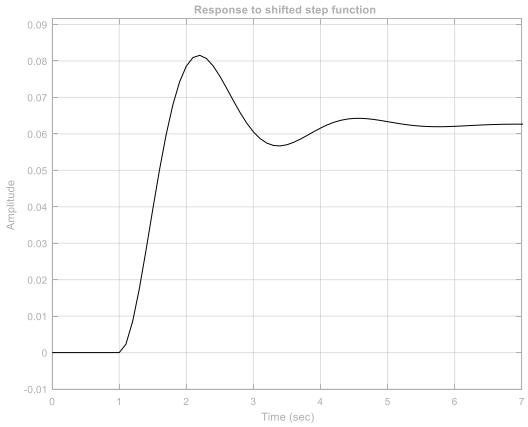

Fig 2(g): Displacement response of the system

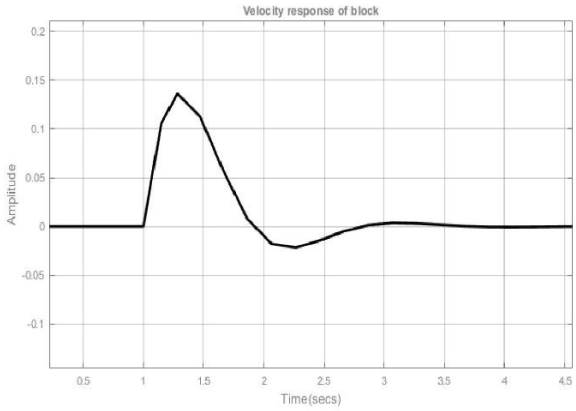

Fig 2(g): Velocity response of the system

Conclusion:

- MATLAB functions were used to generate the transfer function, poles, zeros and gain for a given transfer function, Impulse and step response were plotted for the same transfer function.
- Transfer function was successfully generated using the Control System Toolbox and poles, zeros and gain was analyzed.
- The damped spring mass system was modelled and analyzed using SIMULINK.
- It was observed that initially the effect of the spring force is very less compared to the applied step input, hence the mass accelerates to increase velocity and displacement, eventually due to high displacement and hence high spring force the mass is pulled back and forth by the counter acting spring and applied force, resulting in oscillations. These oscillations of the system are damped out and hence the mass attains a constant displacement after a few time constants.