Control Systems

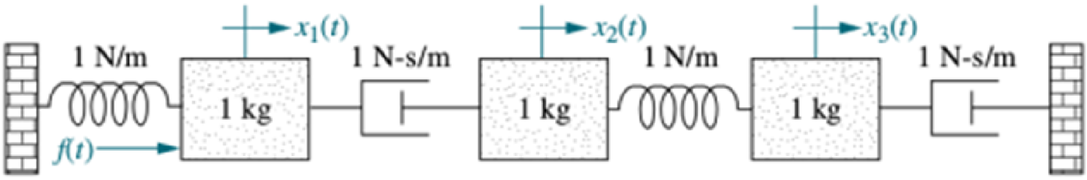
Lab Assessment #2

Date of experiment – 25/08/2023

Group Members: Adarsh Santoria (B21176), Vanshaj Nathani (B20237), Amit Meena (B21180)

Objective:

To obtain the state-space representation of a given system and find the transfer function using MATLAB. To use the transfer function so obtained to find the response of the system to standard test signals like impulse and step functions.



Experiment Design:

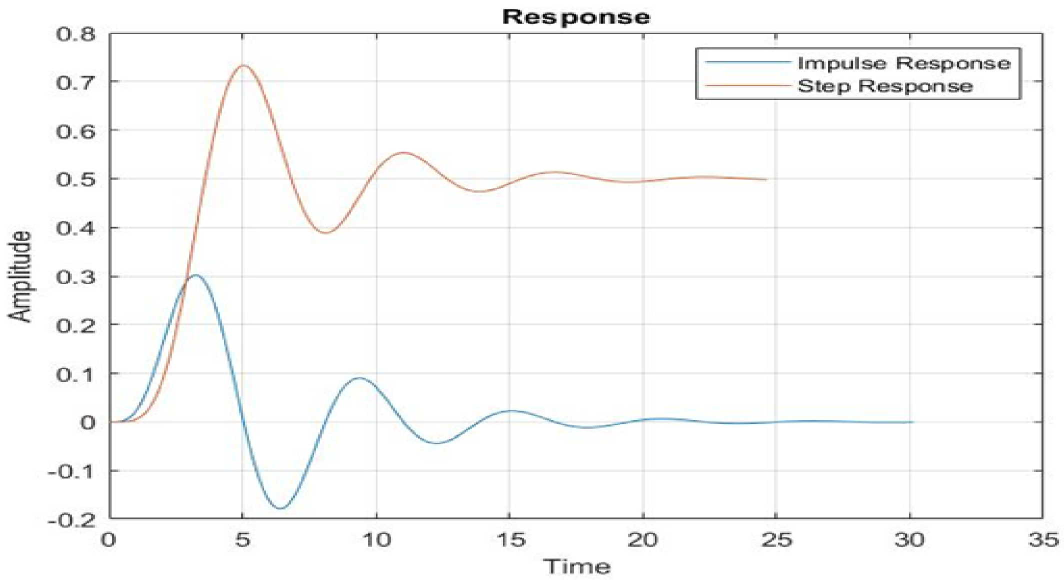
1. Responses of the system from TF obtained by ss2tf:

Code:

```
a = [0 1 0 0 0 0; -1 -1 0 1 0 0; 0 0 0 1 0 0; 0 1 -1 -1 1 0; 0 0 0 0 0 1; 0 0 1 0 -1 -1];
b = [0 1 0 0 0 0]';
c = [0 0 0 0 1 0];
d = 0;

[num,den] = ss2tf(a,b,c,d);
num = round(num,1);
den = round(den,1);
trans_func = tf(num,den);

% Impulse response
[y,t] = impulse(tf(num,den));
plot(t, y);
hold on;
%step(tf(num,den))
[z,t] = step(tf(num,den));
plot(t,z);
grid on;
legend("Impulse Response", "Step Response");
xlabel("Time");
ylabel("Amplitude");
title("Response");

[z, p, k] = tf2zp(num, den);
zpk(z,p,k)
```

By using the above code, we can generate the plot containing the required impulse and step responses.

Plot:



The plot contains the impulse and step responses.

Observations:

- In both the cases we have the decaying oscillations in the similar manner.
- Step response settles at 1.
- Unit response settles at 0.
- Unit response is negative in the transition at some points but the step response is always non negative.

2. Better way of representing a TF:

Code:

```
syms s;

a = [0 1 0 0 0 0; -1 -1 0 1 0 0; 0 0 0 1 0 0; 0 1 -1 -1 1 0; 0 0 0 0 0 1; 0 0 1 0 -1 -1];
b = [0 1 0 0 0 0]';
c = [0 0 0 0 1 0];
d = 0;
i = eye(6);

tf = c*inv(s*i-a)*b+d;

```

- The zero pole gain method can be obtained by using the following code

```
a = [0 1 0 0 0 0; -1 -1 0 1 0 0; 0 0 0 1 0 0; 0 1 -1 -1 1 0; 0 0 0 0 0 1; 0 0 1 0 -1 -1];
b = [0 1 0 0 0 0]';
c = [0 0 0 0 1 0];
d = 0;

[num,den] = ss2tf(a,b,c,d);

num = round(num,1);
den = round(den,1);


[z, p, k] = tf2zp(num, den);
zpk(z,p,k)
```

Results:

- *Representation1 (using the relation between A,B,C,D)*

```
>> tf

tf =

1/(s^5 + 3*s^4 + 5*s^3 + 6*s^2 + 4*s + 2)
```

- *Representation2 (using the ss2tf and zpk)*

```
ans =

                              s
  -------------------------------------------------
  s (s+1.544) (s^2 + s + 1) (s^2 + 0.4563s + 1.296)

Continuous-time zero/pole/gain model.
```

Observations:

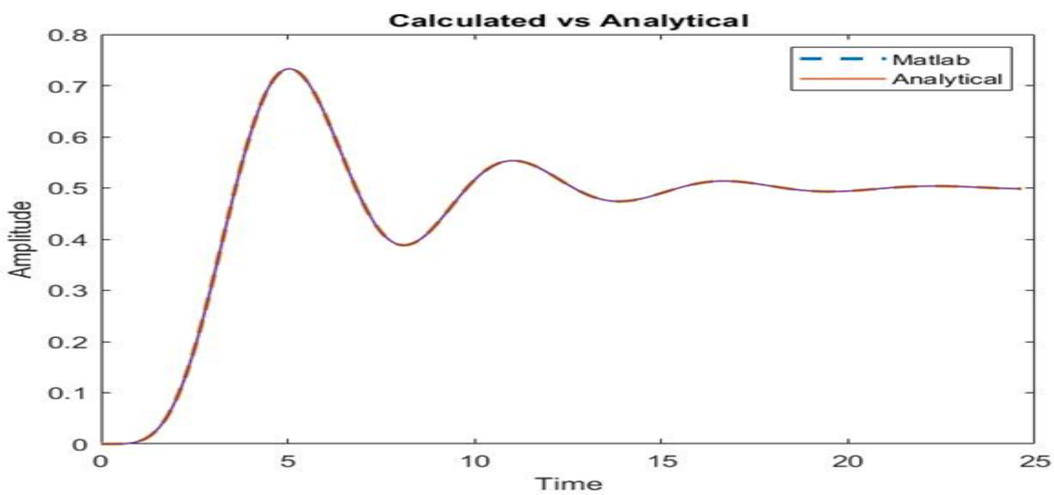
- The representation1 is the simplified version of the tf.
- The representation2 is the version of tf where we have the complete idea of the zeroes and poles.
- While studying any behaviour of the system, representation2 can be useful.
- Also while solving numericals, we have the factored version in representation2 (zpk model) which eases the work.

3. MATLAB vs Analytical:

Code:

```
a = [0 1 0 0 0 0; -1 -1 0 1 0 0; 0 0 0 1 0 0; 0 1 -1 -1 1 0; 0 0 0 0 0 1; 0 0 1 0 -1 -1];
b = [0 1 0 0 0 0]';
c = [0 0 0 0 1 0];
d = 0;

[num,den] = ss2tf(a,b,c,d);

num = round(num,1);
den = round(den,1);

trans_func = tf(num,den);
% Using the ss2tf
[y1,t1] = step(trans_func);
%Analytical
[y2,t2] = step(tf([0 0 0 0 0 1 0],[1 3 5 6 4 2 0]));

%Plotting the results
plot(t1, y1, 'LineWidth',2, 'LineStyle','--');
hold on;
plot(t2, y2, 'LineWidth',1);
grid on;
legend("Matlab", "Analytical");
xlabel("Time");
ylabel("Amplitude");
title("Response");xlabel("Time");
ylabel("Amplitude");
title("Calculated vs Analytical");
```

Results:



Observations:

- We can clearly observe that the analytical and MATLAB calculated responses overlapped each other.

- This verifies our calculation.

Inference:

- By seeing the step response of the system, we can notice that it is similar to the undamped case in 2nd order systems.

- Whenever we give the unit input (step), we will be overshooting initially.

- After we can notice the decaying oscillations.

- Finally the output settles to the desired value after some duration.

Improvements:

1. Wisely tuning or selecting the values of the spring constant (stiffness constant) and the dashpot coefficients will decrease the overshoot.
2. We can use the mechanical constraint in this system to suppress the excess overshoot in the output (i.e. restricting the block M3).