Department of Electrical Engineering and Computer Science

Faculty Member: Ms. Neelma Naz Dated: 02/02/2023

Semester: 6th Section: BEE 12C

EE-371: Linear Control Systems

Lab 1: Laplace Transform, Transfer Functions and Control System Toolbox in MATLAB

Lab Instructor: Yasir Rizwan

Group Members

Name	Reg. No	Lab Report Marks	Viva Marks	Total
		10 Marks	5 Marks	15 Marks
Danial Ahmad	331388			
Muhammad Umer	345834			
Tariq Umar	334943			

1 Table of Contents

2 Laplace Transform, Transfer Function and Control System Toolbox

2.1 Objectives

The objectives of this lab are:

- An introduction to MATLAB Control systems toolbox, which is one of the most popular software tools used by control engineers
- An introduction to MATLAB Simulink, another very popular software tool
- Learn how to find Laplace and inverse Laplace transforms in MATLAB
- Find how to represent transfer functions (or models) in MATLAB and Simulink

2.2 Introduction

In this lab report, we will be utilizing MATLAB to explore the concepts of Laplace transform, transfer function, and control systems. The Laplace transform is a powerful tool allowing us to study the frequency domain representation of time-domain signals. The transfer function represents the relationship between the input and output of a linear, time-invariant system in the frequency domain, and it can be used to study the behavior and stability of control systems. The Control System Toolbox in MATLAB provides a range of tools and functions for analyzing and designing control systems.

In this lab, we will use MATLAB to perform Laplace transformations, calculate transfer functions, and design control systems. The aim of this lab is to provide hands-on experience with the Laplace transform, transfer function, and Control System Toolbox in MATLAB, and to demonstrate their usefulness in the analysis and design of control systems.

2.3 Software

MATLAB is a high-level programming language and numerical computing environment. Developed by MathWorks, it provides an interactive environment for numerical computation, visualization, and programming. MATLAB is widely used in various fields, including engineering, science, and finance, due to its capabilities for matrix and vector operations, implementation of algorithms, and creation of graphical representations of data.

2.3.1 New Commands

Following are the new commands learned in this lab:

- laplace(f(t)); takes Laplace transform of a function f(t)
- ilaplace(F(s)); takes inverse Laplace transform of a function F(s)
- pole(F(s)); returns poles of F(s)
- zero(F(s)); returns zeros of F(s)
- tf(num, den); where num represents numerator and den represents denominator
- zpk(p, z, k); where p represents poles, z represents zeros, and k represents gain

3 Lab Procedure

3.1 Exercise 1

Find the Laplace transform of the following time domain functions:

i. $G1(s) = tsin(2t) + e^{-2t}$
ii. $G2(s) = \sin(2t)\cos(2t)$
iii. $G3(s) = e^{-t}\cos(3t)$
iv. $G4(s) = -e^{-t} + 9te^{-t}5e^{-2t} + t - 2$
v. $G5(s) = 5t^2\cos(3t + 45)$

Find the inverse Laplace transform of the following frequency domain functions:

i. $G1(s) = \frac{1}{s(s+2)(s+3)}$
ii. $G2(s) = \frac{10}{(s+1)^2(s+3)}$
iii. $G3(s) = \frac{2(s+1)}{s(s^2+s+2)}$
iv. $G4(s) = \frac{s+1}{s(s+2)(s^2+s+2)}$

Code

```
syms t;

g1 = (t * sin(2 * t) + exp(2 * t));
g2 = sin(2 * t) * cos(2 * t);
g3 = exp(-t) * cos(3 * t);
g4 = -exp(-t) + 9 * t * exp(-t) + 5 * exp(-2 * t) + (t - 2);
g5 = 5 * (t ^ 2) * cos(3 * t + pi/4);

fprintf('Laplace of g1(t) = %s\n', laplace(g1))
fprintf('Laplace of g2(t) = %s\n', laplace(g2))
fprintf('Laplace of g3(t) = %s\n', laplace(g3))
fprintf('Laplace of g4(t) = %s\n', laplace(g4))
fprintf('Laplace of g5(t) = %s\n', laplace(g5))
```

Output

```
Laplace of g1(t) = 1/(s - 2) + (4*s)/(s^2 + 4)^2
Laplace of g2(t) = 2/(s^2 + 16)
Laplace of g3(t) = (s + 1)/((s + 1)^2 + 9)
Laplace of g4(t) = 9/(s + 1)^2 - 1/(s + 1) + 5/(s + 2) - (2*s - 1)/s^2
Laplace of g5(t) = (5*2^(1/2)*(6/(s^2 + 9)^2 - (24*s^2)/(s^2 + 9)^3))/2 -
(5*2^(1/2)*((6*s)/(s^2 + 9)^2 - (8*s^3)/(s^2 + 9)^3))/2
```

```
Code
syms s;

G1 = 1 / (s * (s + 1) * (s + 3));
G2 = 10 / ((s + 1) ^ 2 * (s + 3));
G3 = 2 * (s + 1) / (s * (s ^ 2 + s + 2));
G4 = (s + 1) / (s * (s + 2) * (s ^ 2 + 2 * s + 2));

fprintf('Inverse Laplace of G1(s) = %s\n', ilaplace(G1))
fprintf('Inverse Laplace of G2(s) = %s\n', ilaplace(G2))
fprintf('Inverse Laplace of G3(s) = %s\n', ilaplace(G3))
fprintf('Inverse Laplace of G4(s) = %s\n', ilaplace(G4))

Output
Inverse Laplace of G1(s) = exp(-3*t)/6 - exp(-t)/2 + 1/3
Inverse Laplace of G2(s) = (5*exp(-3*t))/2 - (5*exp(-t))/2 + 5*t*exp(-t)
Inverse Laplace of G3(s) = 1 - exp(-t/2)*(cos((7^(1/2)*t)/2) -
(3*7^(1/2)*sin((7^(1/2)*t)/2))/7)
Inverse Laplace of G4(s) = exp(-2*t)/4 - (exp(-t)*cos(t))/2 + 1/4
```

3.2 Exercise 2

Find the poles and zeros of the transfer function given below. Remember the zeros are the roots of the numerator polynomial and poles are the roots of the denominator polynomial.

$$\frac{5s + 10}{s^2 + 7s + 12}$$

```
Code
num = [5 10];
den = [1 7 12];

G1 = tf(num, den);
fprintf('Zeroes(s): %d\n', zero(G1))
fprintf('Pole(s): %d\n', pole(G1))

Output
Zeroes(s): -2
Pole(s): -4
         -3
```

3.3 Exercise 3

Create the following transfer functions in MATLAB and convert them to the other form. Give each transfer function a different name, e.g., G1, G2 etc.

$$\frac{30s - 180}{(s^3 + 4s^2 + 13s + 7)} \qquad \frac{(s^3 + 11s^2 + 35s + 250)}{s^5 + 4s^4 + 39s^3 + 108s^2}$$

$$\frac{s^3 + s + 1}{s^3 + s^2 + 6} \qquad \frac{s^2 + 5s + 6}{(s^5 + 4s^3 + 15s + 35)}$$

$$\frac{(s + 4)(s + 2 - 4i)(s + 2 + 4i)}{(s - 2)(s + 4)(s + 5i)(s - 5i)} \qquad \frac{s^2 - 1}{s^4 + 4s^2 + 6s + 4}$$

Code

```
%% Task 3.1
num = [30 -180];
den = [1 4 13 7];

G1 = tf(num, den);
display(G1)
zpk(G1)

%% Task 3.2
num = [1 0 1 1];
den = [1 1 0 6];

G2 = tf(num, den);
display(G2)
zpk(G2)

%% Task 3.3
p = [-4 -2 + 4i -2 - 4i];
z = [2 -4 -5i 5i];

G3 = zpk(p, z, 1);
display(G3)
tf(G3)

%% Task 3.4
num = [1 11 35 250];
den = [1 4 39 108];

G4 = tf(num, den);
display(G4)
zpk(G4)

%% Task 3.5
num = [1 5 6];
den = [1 4 15 35];

G5 = tf(num, den);
```

```
display(G5)
zpk(G5)

%% Task 3.6
num = [1 0 -1];
den = [1 0 4 6 4];

G6 = tf(num, den);
display(G6)
zpk(G6)
```

Output

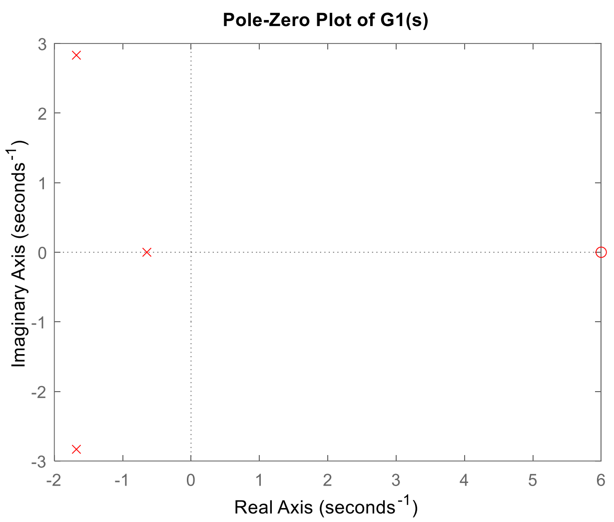
```
G1 =

30 s - 180
----------------------
s ^ 3 + 4 s ^ 2 + 13 s + 7

Continuous - time transfer function.

ans =

30 (s - 6)
- --------------------------------
(s + 0.6462) (s ^ 2 + 3.354s + 10.83)

Continuous - time zero / pole / gain model.

G2 =

s ^ 3 + s + 1
- ------------
s ^ 3 + s ^ 2 + 6

Continuous - time transfer function.

ans =

(s + 0.6823) (s ^ 2 - 0.6823s + 1.466)
----------------------------------
(s + 2.219) (s ^ 2 - 1.219s + 2.704)

Continuous - time zero / pole / gain model.

G3 =

(s + 4) (s ^ 2 + 4s + 20)
```

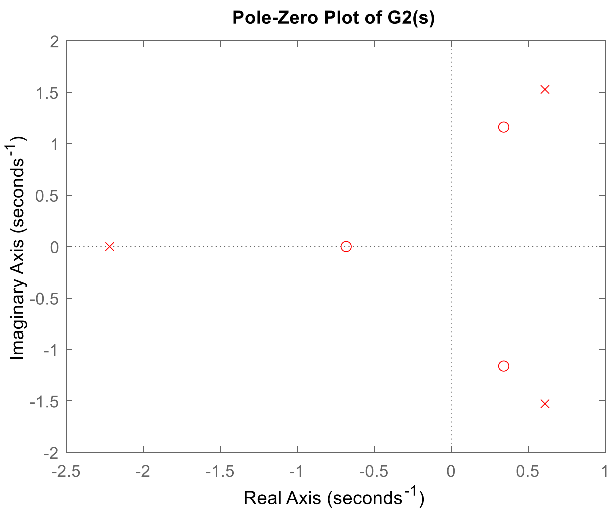
```
----------------------
(s - 2) (s + 4) (s ^ 2 + 25)

Continuous - time zero / pole / gain model.

ans =

s ^ 3 + 8 s ^ 2 + 36 s + 80
- ---------------------------------
s ^ 4 + 2 s ^ 3 + 17 s ^ 2 + 50 s - 200

Continuous - time transfer function.

G4 =

s ^ 3 + 11 s ^ 2 + 35 s + 250
- ------------------------
s ^ 3 + 4 s ^ 2 + 39 s + 108

Continuous - time transfer function.

ans =

(s + 10) (s ^ 2 + s + 25)
- --------------------
(s + 3) (s ^ 2 + s + 36)

Continuous - time zero / pole / gain model.

G5 =

s ^ 2 + 5 s + 6
- ----------------------
s ^ 3 + 4 s ^ 2 + 15 s + 35

Continuous - time transfer function.

ans =

(s + 3) (s + 2)
--------------------------------
(s + 2.944) (s ^ 2 + 1.056s + 11.89)

Continuous - time zero / pole / gain model.

G6 =
```

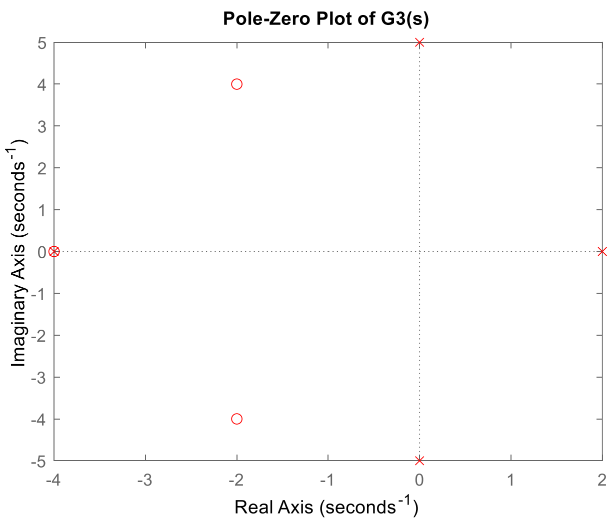
```
s ^ 2 - 1
- --------------------
s ^ 4 + 4 s ^ 2 + 6 s + 4

Continuous - time transfer function.

ans =

(s + 1) (s - 1)
----------------------------------------------
(s ^ 2 + 1.381s + 0.7803) (s ^ 2 - 1.381s + 5.126)
```

3.4 Exercise 4

Plot the poles and zeros of any three transfer functions that you have already created in the MATLAB.







3.5 Exercise 5

Create a model for the following transfer functions in Simulink:

$$\frac{30s-180}{(s^3+4s^2+13s+7)} \qquad \frac{s^3+s+1}{s^3+s^2+6}$$

```
num = [30 -180]
den = [1 4 13 7]
```

```
num = [1 0 1 1]
den = [1 1 0 6]
```

$$\boxed{\frac{30s-180}{s^3+4s^2+13s+7}} \qquad \boxed{\frac{s^3+s+1}{s^3+s^2+6}}$$

3.6 Exercise 6

The transfer function of a pendulum on a cart is given by:

$$\frac{\frac{mL}{q}s^2}{s^4+\frac{b(I+mL^2)}{q}s^3-\frac{(M+m)mgL}{q}s^2-\frac{mgL}{q}s}$$

where

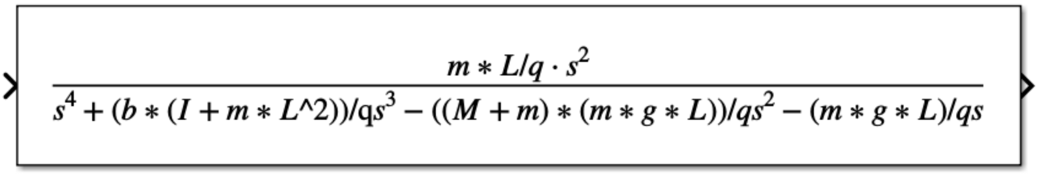
- Mass of cart = M = 0.5 kg
- Mass of pendulum = m = 0.5 kg
- Friction of cart =b= 0.1 N/m/sec
- Length of pendulum = L=0.3m
- Inertia of pendulum = I = 0.006 kg m2
- $q = [(M+m)(I+mL^2)-(mL)^2]$

Create a model for this transfer function in Simulink. Also create an m file to create this transfer function in MATLAB.

Code

```
%% Task 6
g = 9.81;
M = 0.5;
m = 0.5;
b = 0.1;
L = 0.3;
I = 0.006;
q = ((M + m) * (I + m * L ^ 2)) - (m * L) ^ 2;

num = [m * L / q 0 0];
den = [1 (b * (I + m * L ^ 2)) / q -((M + m) * (m * g * L)) / q ...
         -(m * g * L) / q 0];
```

```
G7 = tf(num, den);
display(G7)

                                        Output
G7 =

                5.263 s^2
  --------------------------------------
  s^4 + 0.1789 s^3 - 51.63 s^2 - 51.63 s
```



4 Conclusion

In this introductory laboratory session, we were introduced to the fundamental concepts of transfer functions and the techniques for their implementation in MATLAB. The session covered various basic functions such as laplace(), ilaplace(), pole(), zero(), tf(), and zpk(). We learned how to construct transfer functions in MATLAB using two methods; one method involves passing the numerator and denominator values to the tf() function, and the other method involves passing the values of poles, zeros, and gain to the zpk() function. Additionally, we were also informed that it is possible to convert from one form to another by simply passing the given function to the other form.

Finally, we were instructed on the creation of transfer function blocks in Simulink. In this section, we learned two different methods; one method involves directly writing the coefficients of the numerator and denominator, while the other method entails representing the actual value using a variable and then assigning values to the variables in the settings.