National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering and Computer Science

Faculty Member: Ms. Neelma Naz Dated: 28/02/2023

Semester: 6th Section: BEE 12C

EE-371: Linear Control Systems

Lab 5: State Space, Response of Systems to Various Inputs, and Interconnections of Systems

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 State Space, Response of Systems to Various Inputs, and Interconnections of Systems

2.1 Objectives

The objectives of this lab are:

- Learn how to create state space models in MATLAB, Simulink, and LabVIEW.
- Learn how to find time response of a system for various inputs in MATLAB, Simulink, and LabVIEW.
- Interconnection of systems in MATLAB.

2.2 Introduction

State space modeling is a powerful tool for analyzing dynamic systems and has a wide range of applications in various fields, including control systems, robotics, and aerospace engineering. The objective of this lab is to provide a comprehensive understanding of state space modeling and its practical applications. Specifically, the lab focuses on creating state space models and analyzing the time response of systems for various inputs.

The lab is divided into three parts, each focusing on a different software platform, namely MATLAB, Simulink, and LabVIEW. In each part, we will learn how to create state space models and simulate the response of the system for various inputs. We will also learn how to analyze the stability and performance of the system by evaluating its time response.

2.3 Software

LabVIEW is a graphical programming environment that uses icons to represent instructions and commands. Its graphical approach makes it easy to understand and manipulate data. It allows the user to tailor data inputs and outputs to meet their specific needs. The Control and Simulation module in LabVIEW allows the user to simulate and analyze complex systems, such as those found in robotics, mechatronics, and autonomous systems.

3 Lab Procedure

3.1 Exercise: Constructing a State Space Model

3.1.1 In MATLAB

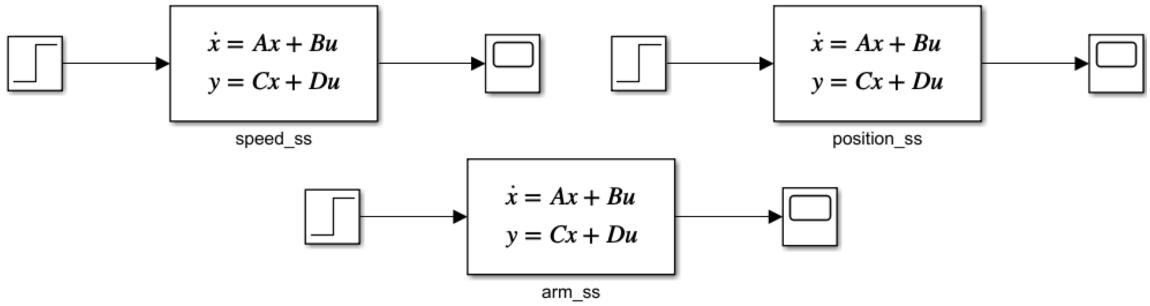
Use the transfer functions of motor speed, motor position and pendulum arm angle to find their state space representation.

```
%% Exercise 1
% Variables
J_eq = 1.843e-6;
m = 0.0270;
r = 0.0826;
K_g = 1;
J_m = 1.80e-4;
n_g = 1;
n_m = 0.69;
L = 0.0955;
g = 9.9;
K_t = 0.0334;
R = 8.7;

a = J_eq + m * r ^ 2 + n_g * K_g ^ 2 * J_m;
b = m * L * r;
c = (4/3) * m * L ^ 2;
d = m * g * L;
e = 2.7183;
f = (n_m * n_g * K_t * K_g) / R;

% Motor speed
num = [-c 0 f * d 0];
den = [(b ^ 2 - a * c) -e * c a * d e * d 0];
speed_tf = tf(num, den);

% Motor position
num = [-c 0 f * d];
den = [(b ^ 2 - a * c) -e * c a * d e * d 0];
position_tf = tf(num, den);

% Pendulum arm angle
num = [-b * f 0 0];
den = [(a * c - b ^ 2) e * c -a * d e * d 0];
arm_tf = tf(num, den);

speed_ss = ss(speed_tf);
position_ss = ss(position_tf);
arm_ss = ss(arm_tf);

                                            Output
speed_ss =
A =
x1 x2 x3 x4
x1 -1.193e+04 1.951 113.2 0
x2 64 0 0 0
x3 0 128 0 0
```

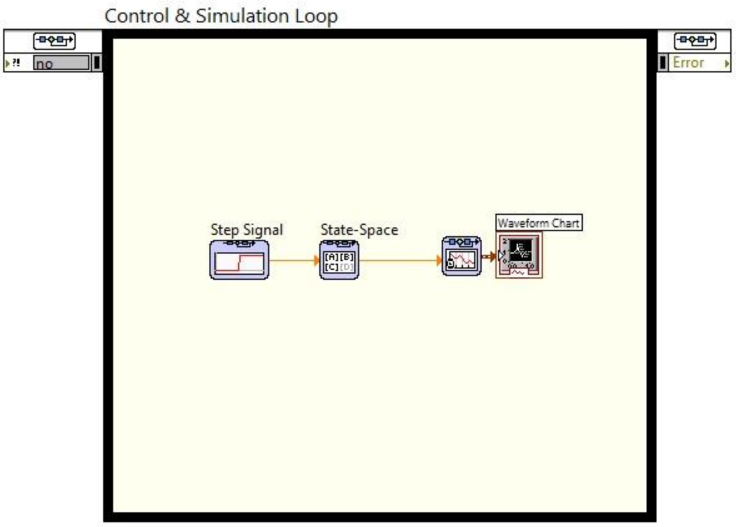
```
x4 0 0 1 0

B =
u1
x1 64
x2 0
x3 0
x4 0

C =
x1 x2 x3 x4
y1 68.56 0 -0.001724 0

D =
u1
y1 0

Continuous - time state - space model.

position_ss =
A =
x1 x2 x3 x4
x1 -1.193e+04 1.951 113.2 0
x2 64 0 0 0
x3 0 128 0 0
x4 0 0 1 0

B =
u1
x1 8
x2 0
x3 0
x4 0

C =
x1 x2 x3 x4
y1 0 8.57 0 -0.01379

D =
u1
y1 0

Continuous - time state - space model.

arm_ss =
A =
x1 x2 x3 x4
x1 -1.193e+04 1.951 -113.2 0
x2 64 0 0 0
x3 0 128 0 0
x4 0 0 1 0

B =
u1
x1 0.25
x2 0
x3 0
x4 0
```

```
C =
x1 x2 x3 x4
y1 0 -0.4713 0 0

D =
u1
y1 0

Continuous - time state - space model.
```

3.1.2 In Simulink



3.1.3 In LabVIEW

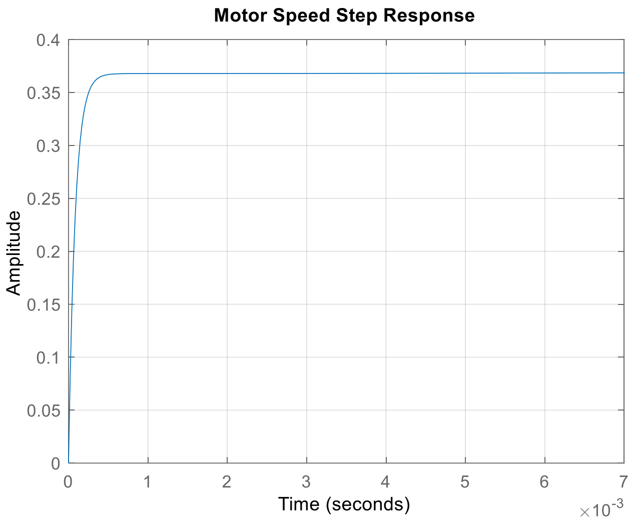


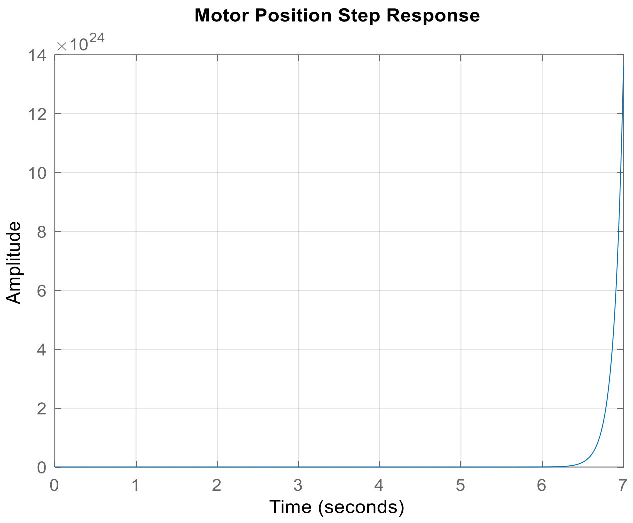
3.2 Exercise: Step Response in MATLAB

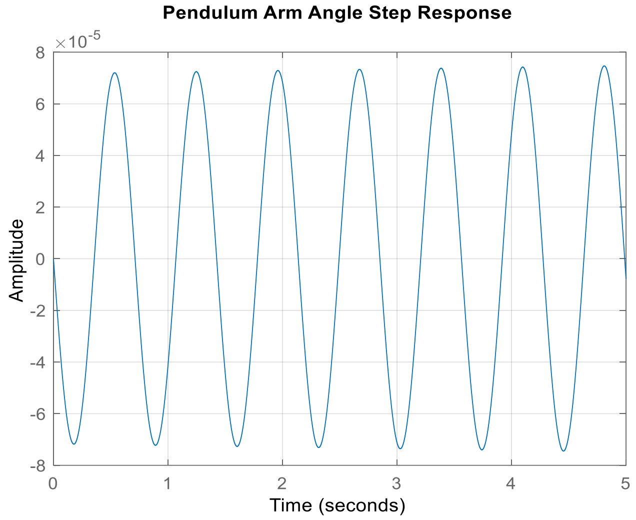
Using the models of motor and pendulum find the unit step response for motor speed, motor position and pendulum arm angle. You can use any model representation you like. Comment on the response that you get for all the systems.

```
%% Exercise 2
t = 0:0.001:5;

figure
step(speed_tf)
```

```
grid
title('Motor Speed Step Response')

figure
step(position_tf)
grid
title('Motor Position Step Response')

figure
step(arm_tf, t)
grid
title('Pendulum Arm Angle Step Response')
```





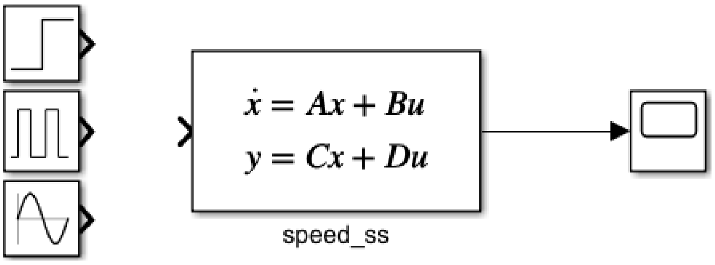


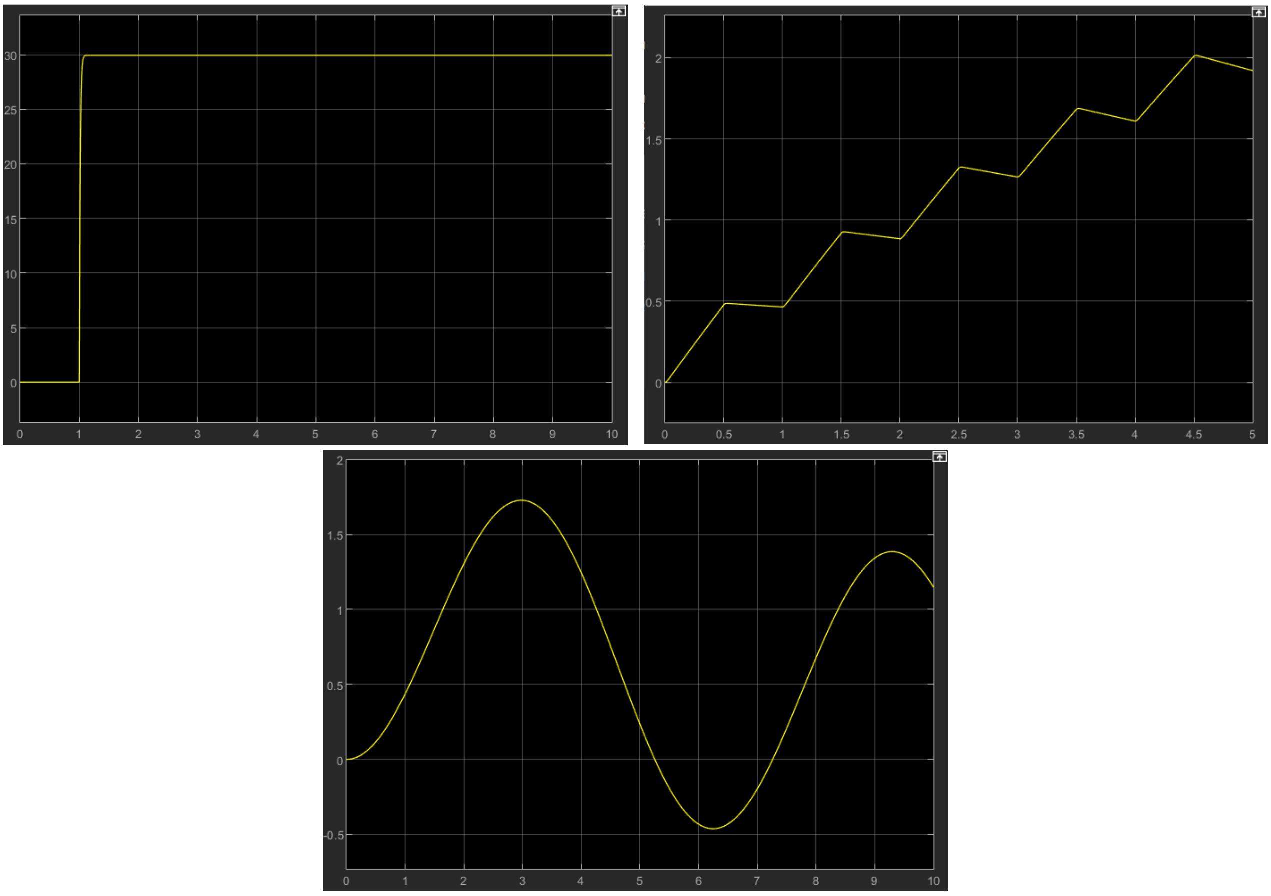
Comments: The motor speed corresponding to the input step voltage accelerates until reaching a steady state, as one would expect of a motor. The motor position response observes the "ringing" effect, and it overshoots the desired value before settling down. The sinusoidal response of the pendulum arm angle is caused by the gravitational force on the pendulum, which makes it accelerate towards the equilibrium position and overshoot before oscillating around it due to momentum.

3.3 Exercise: Step Response in Simulink

Using the models of motor speed, find the response to the following inputs in Simulink

- Unit step
- Square wave
- Sine wave

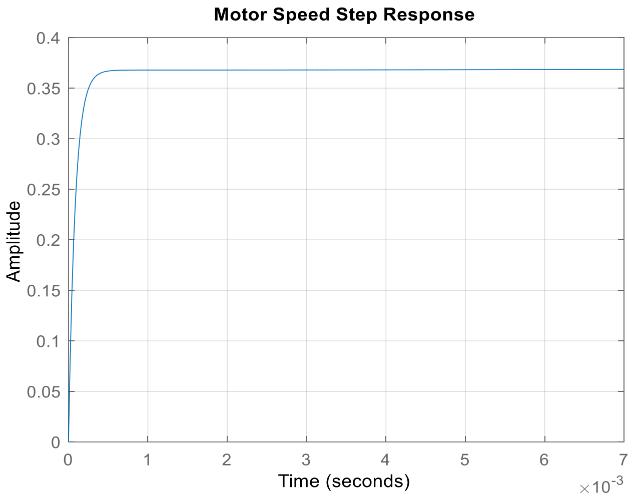


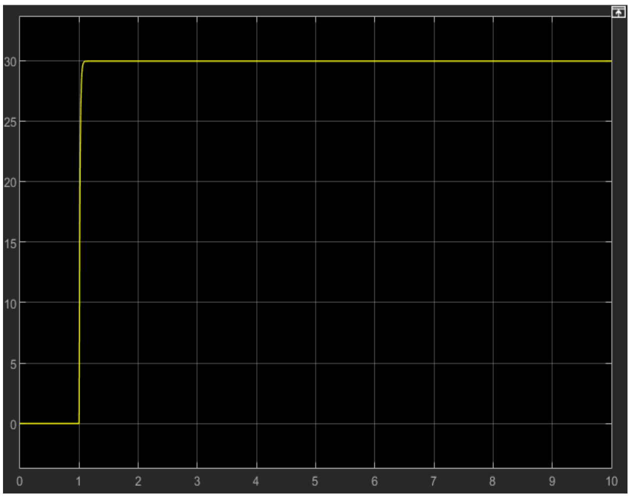


Response of motor speed to different varying inputs. A) Top left: Unit step response, B) Top right: Square wave response, C) Bottom center: Sine wave response

3.4 Exercise: Comparison

Compare the unit step response for motor speed obtained from MATLAB, LabVIEW, and Simulink. Are they similar? Are they expected to be similar?



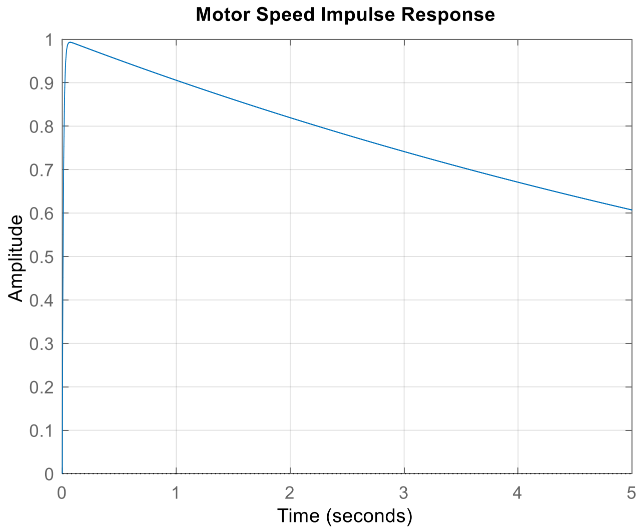


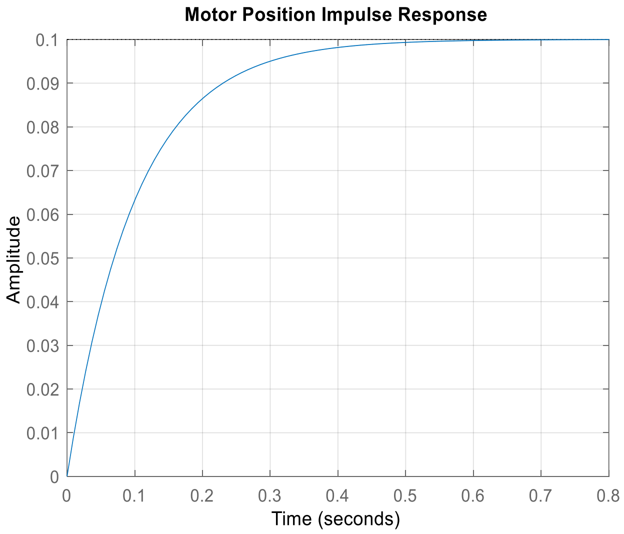
Answer: The response obtained from Simulink and LabVIEW is similar as we define the custom input ourselves, however, the MATLAB function step() adapts to the design-under-test and holds variable step time and steady state internally, due to which the settling time and amplitude of the response may change.

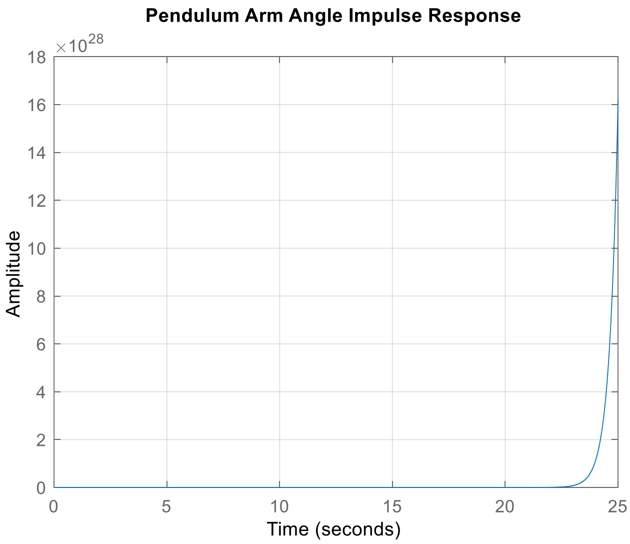
3.5 Exercise: Impulse Response in MATALB

Find the impulse response of motor speed, motor position and pendulum arm angle.

```
impulse(speed_ss, t)
impulse(position_ss, t)
impulse(arm_ss, t)
```



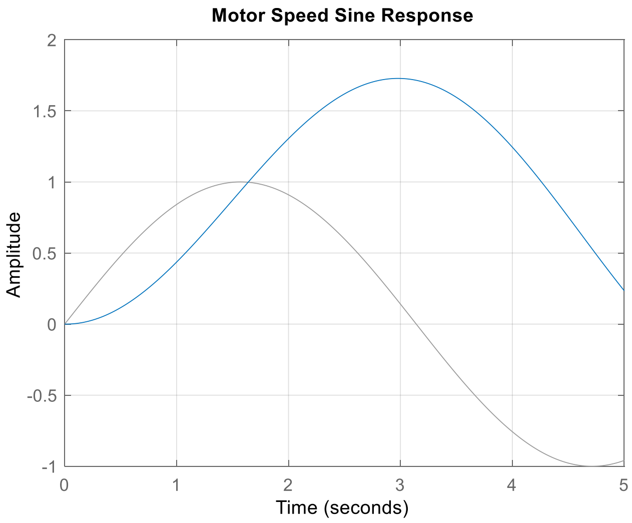


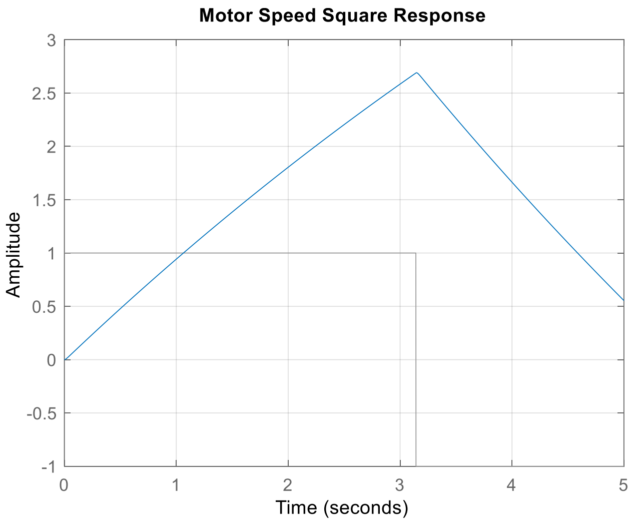


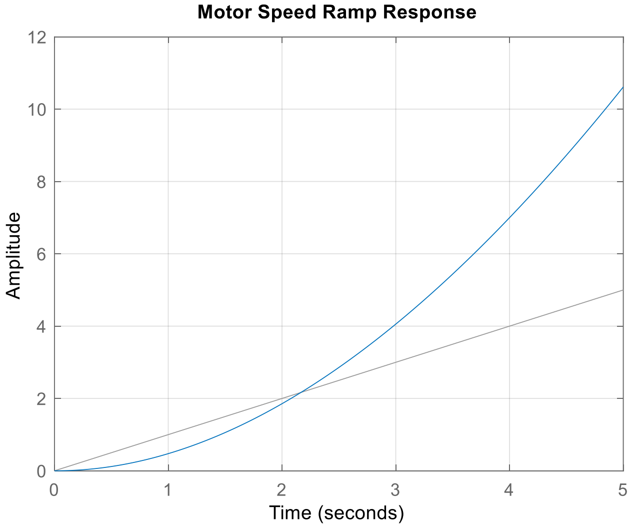
3.6 Exercise: Response of Arbitrary Inputs in MATLAB

Find the response of motor speed to a ramp, square wave, and sine wave inputs in MATLAB.

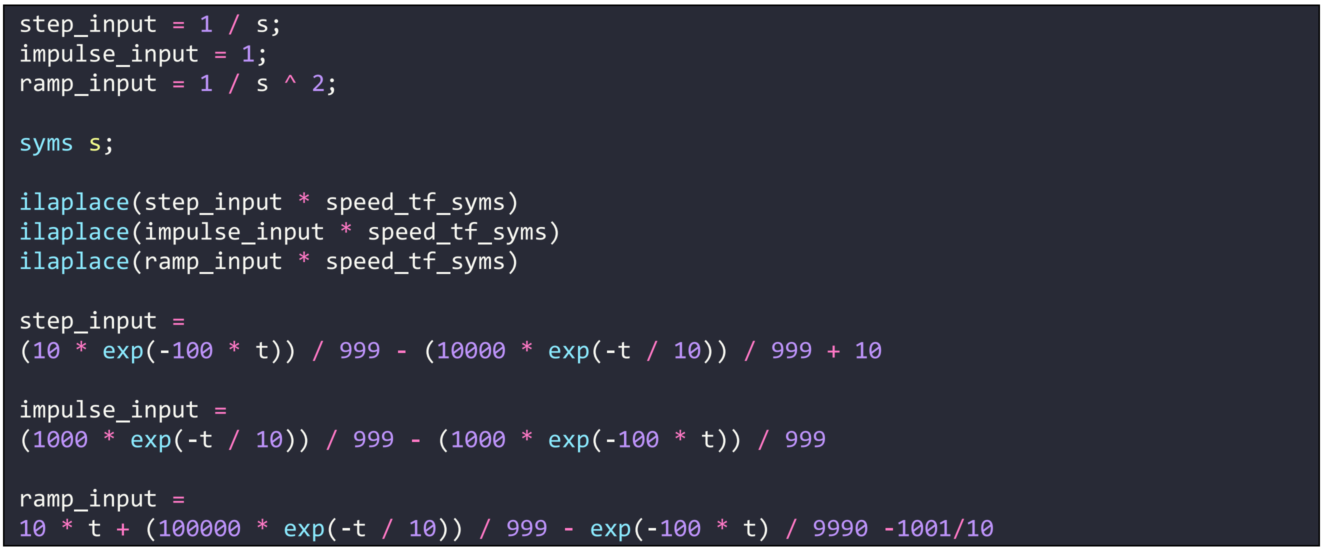
```
t = 0:0.001:5;
u_ramp = t; % ramp
u_square = square(t); % square
u_sine = sin(t); % sine

lsim(speed_ss, u_ramp, t)
grid
title('Motor Speed Ramp Response')

lsim(speed_ss, u_square, t)
grid
title('Motor Speed Square Response')

lsim(speed_ss, u_sine, t)
grid
title('Motor Speed Sine Response')
```





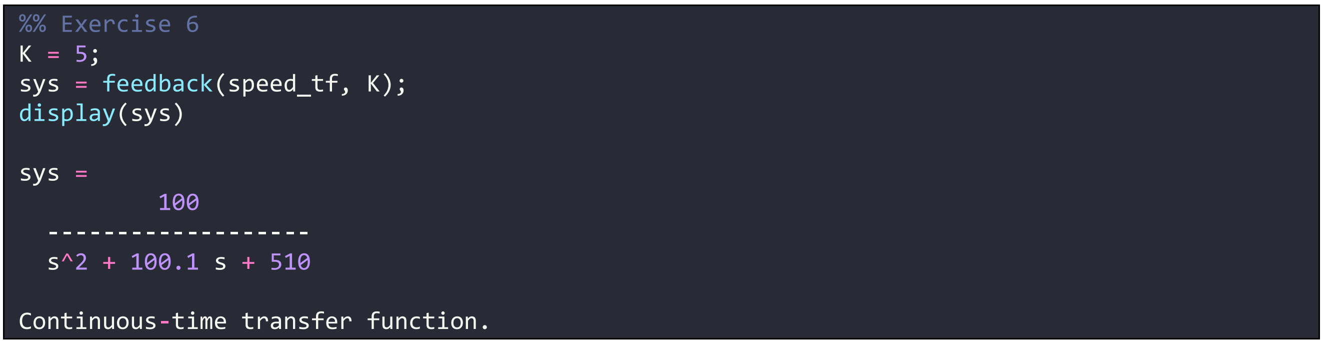


3.7 Exercise: Time Response using Inverse Laplace Transforms


```
step_input = 1 / s;
impulse_input = 1;
ramp_input = 1 / s ^ 2;

syms s;

ilaplace(step_input * speed_tf_syms)
ilaplace(impulse_input * speed_tf_syms)
ilaplace(ramp_input * speed_tf_syms)

step_input =
(10 * exp(-100 * t)) / 999 - (10000 * exp(-t / 10)) / 999 + 10

impulse_input =
(1000 * exp(-t / 10)) / 999 - (1000 * exp(-100 * t)) / 999

ramp_input =
10 * t + (100000 * exp(-t / 10)) / 999 - exp(-100 * t) / 9990 -1001/10
```

3.8 Exercise: Interconnection of Systems in MATLAB

Define a system which has a transfer function of just 5 i.e., a simple gain. Connect this system in negative feedback with the model of dc motor speed.


```
%% Exercise 6
K = 5;
sys = feedback(speed_tf, K);
display(sys)

sys =
          100
  -------------------
  s^2 + 100.1 s + 510

Continuous-time transfer function.
```

4 Conclusion

In conclusion, this lab provided a comprehensive understanding of state space modeling and analysis of systems using MATLAB, Simulink, and LabVIEW. The lab focused on creating state space models, simulating them, and analyzing their response for various inputs. Additionally, the lab explored the interconnection of systems in MATLAB. Through the lab, we learned how to create state space models in MATLAB, Simulink, and LabVIEW. We also gained a thorough understanding of how to find the time response of a system for various inputs. By analyzing the system response, we were able to evaluate the stability and performance of the system.