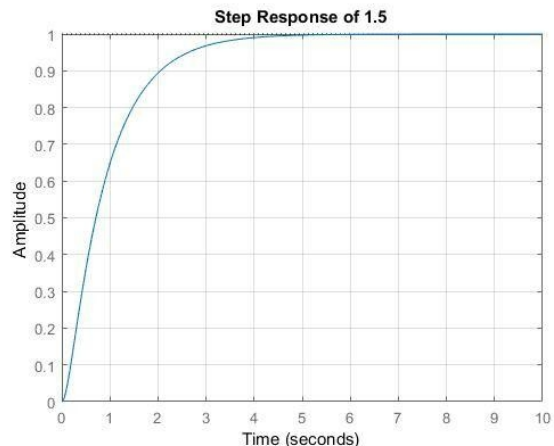
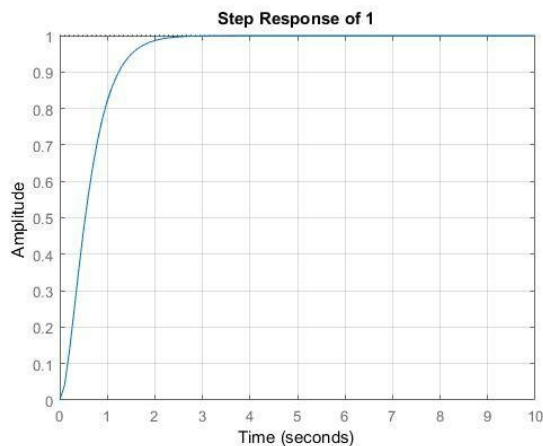
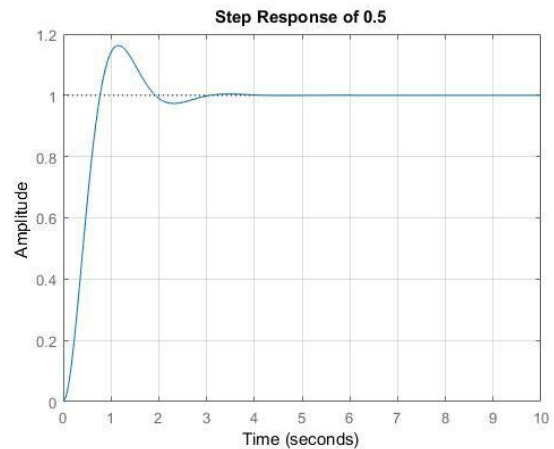
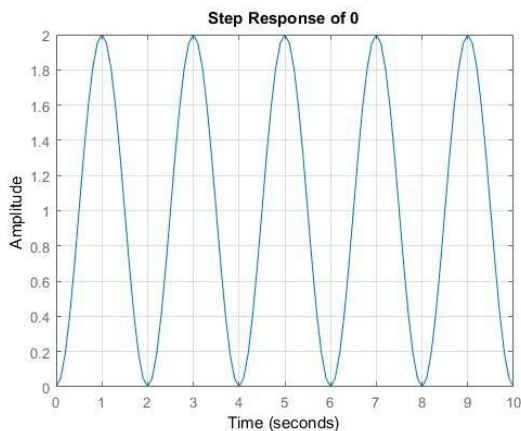


ECSE 307
Lab Report 2
Group 6
Ali Shobeiri
Ismail Faruk

Introduction

The purpose of this lab was to learn how to identify a second-order LTI system from its time and frequency responses. To learn this, we simulated a DC motor using a Matlab class and identified the model parameters of the DC motor using the properties of its time response.



Question 1:

In each case, the rise time, maximum overshoot, 5% settling time can be calculated using builtin Matlab function `stepinfo(H, 'SettlingTimeThreshold', 0.05)`. Where H is the transfer function defined for a specific value of ζ . This can also be done easily by using the plots and finding the maximum value for the peak time and overshoot, as well as using the same process for the undershoot. The settling time can be measured by finding the point where the amplitude of the response is limited to a 5% bandwidth, and can be found by visual inspection by drawing a horizontal line at 1.05 and 0.95. The values shown below are from `stepinfo` and were used to provide greater accuracy.

Table 1: Response characteristics of second order system for different ζ values

	$\zeta = 0$	$\zeta = 0.5$	$\zeta = 1.0$	$\zeta = 1.5$
Rise Time (s)	NaN	0.522	1.069	1.865
Settling Time (s)	NaN	1.683	1.510	2.628
Overshoot	NaN	0.163	0	0
Undershoot	NaN	0	0	0
Peak	NaN	1.163	0.9997	0.9999
Peak Time (s)	NaN	1.143	3.390	8.2755

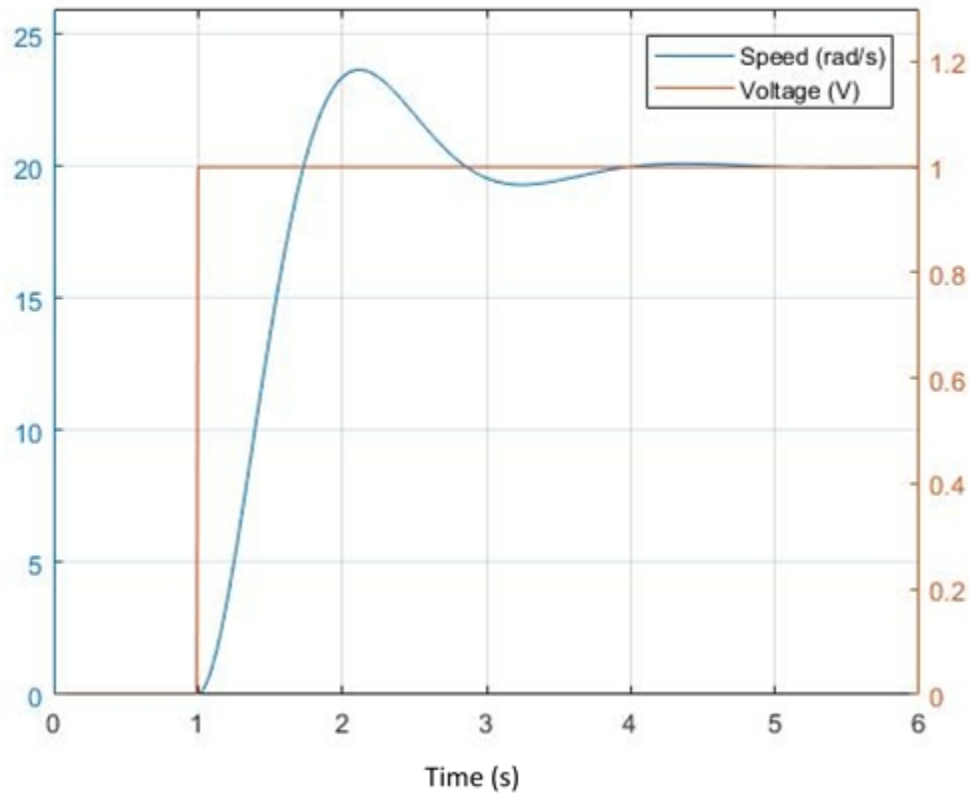
When $\zeta = 0$, the system is undamped. That can be identified by its lack of Settling Time mainly, which means the system never settles and keeps rising and falling, which would also explain why there is no Rise Time.

When $\zeta = 0.5$, the system is underdamped. That can be identified due to the presence of Overshoot.

When $\zeta = 1.0$, the system is critically damped. That can be identified due to lack of Overshoot and the least Settling Time.

When $\zeta = 1.5$, the system is overdamped. The system has no Overshoot just like a critically damped system but its Settling Time and Peak Time is more than when $\zeta = 1.0$.

Question 2



From the graph it can be seen that the DC Gain (steady state value) of the Motor is 20 rad/s. Since the system is underdamped we can infer that the damping ratio is $0 < \zeta < 1$.

By measuring the rise time of system, which is determined to be 0.5024 s we can calculate the natural frequency of the system by using the relationship: $t_r = 1.8/\omega_n$. The natural frequency of the system can be calculated to be 3.583 rad/s.

$M_p = e^{-(\pi \cdot \zeta) / (\sqrt{1 - \zeta^2})}$ using this relationship and measuring from the graph that the maximum overshoot is 3.6631, one can calculate the value of zeta to be = 0.6963.

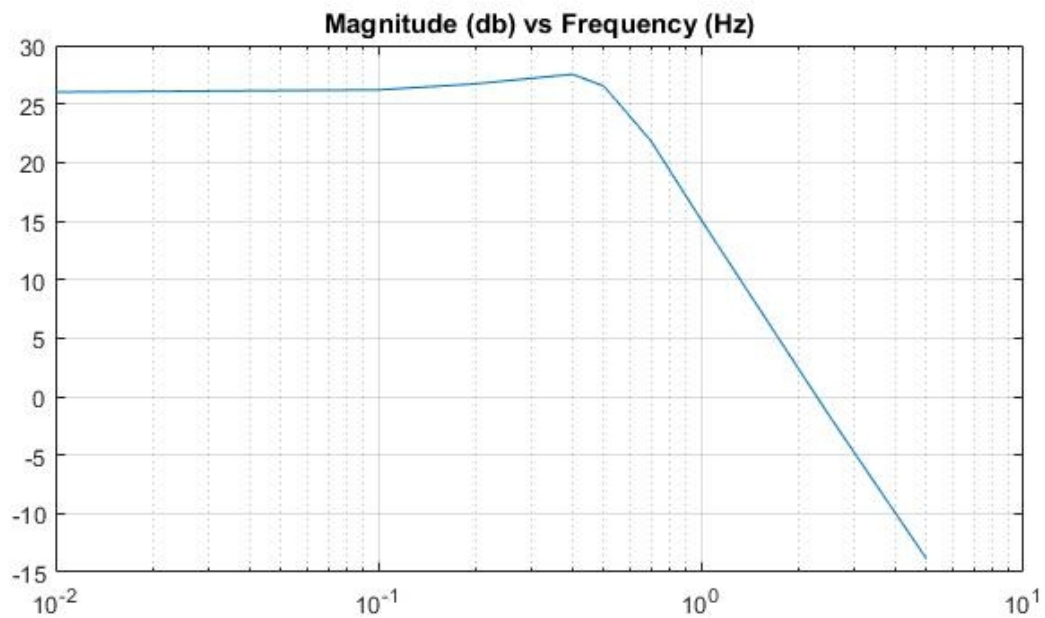
With the gain K being equal to one, and the relationship given by: $K = \frac{K_t}{bR + K_t K_c}$ we can calculate B to be equal to 1%.

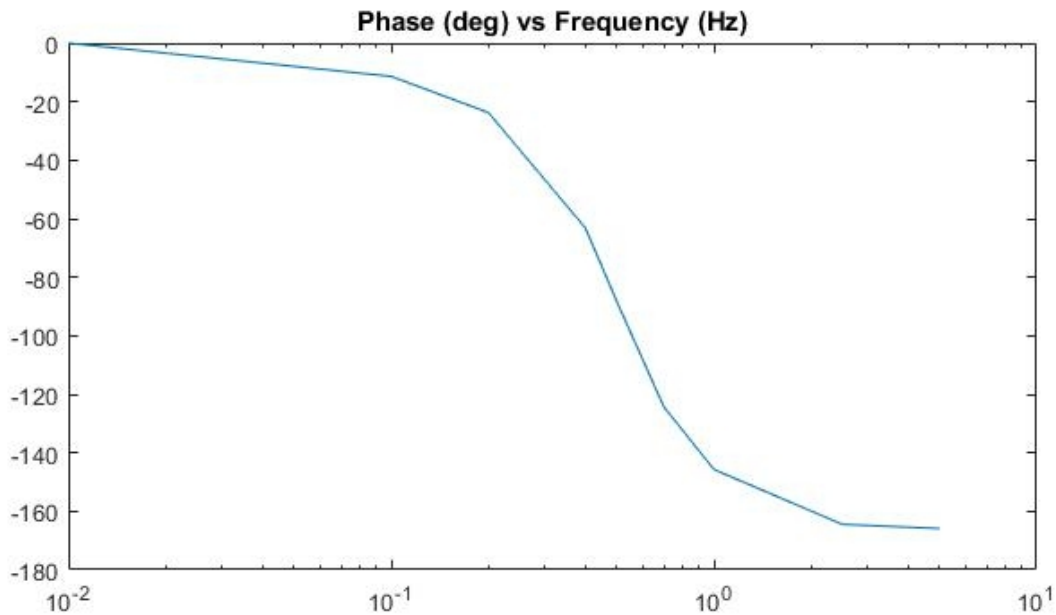
Question 3:

Part 1

Frequency (Hz)	A _y	Y _u = 1	Magnitude (db)	Phase (deg)
0.01	20	20	-26.0206	0
0.1	20.42	20.02	-26.2011	11.35929
0.2	21.66	19.83	-26.7132	23.72139
0.4	23.81	10.73	-27.5352	63.21454
0.5	21.2	0.6421	-26.5267	88.26438
0.7	12.37	-6.981	-21.8474	124.3571
1	5.709	-4.716	-15.1312	145.6965
2.5	0.8269	-0.7967	1.65094	164.4674
5	0.2026	-0.1965	13.86721	165.9046

Part 2





Part 3

From the Bode plot, it can be observed that the system order is 2.

As can be seen from the bode plot, the amplitude is decreasing by more than 20dB per decade and closer to 40dB per decade. From this it can be inferred that the system order is two.

Taking a look at the -180 phase shift, this would support the assumption that the system has an order of 2.

Part 4

The DC gain is approximately 26.02.

The DC gain of the system is easily observed to be 26.02. This can be seen by observing the system amplitude at a frequency close to 0.

Part 5

The cut off frequency is at 0.5 Hz

The cut off frequency is measured as the value at which the amplitude bode plot begins to decline. From the bode plot this can be seen to be 0.5Hz.

Conclusion

Through this lab, we learned how to use Matlab to identify a second-order LTI system from its time and frequency responses, identify the model parameters of the DC motor using the properties of its time responses and understood its mathematical representation. We also learned how to model a continuous time system in discrete time using Matlab, identify the transfer function of a dynamical system in time domain and in frequency domain.

Appendix

Code for Q1

```
k = 1
w = pi
s = tf('s')

for zeta = [0,0.5,1,1.5]
    H = (k*w^2)/(s^2 + 2*zeta*w*s + w^2)
    step(H, 10)
    title('Step Response of ' + zeta)
    grid on
    stepinfo(H, 'SettlingTimeThreshold', 0.05)
    figure
end
```

Code for Q2

```
addpath('Interface');

Parameters.Resistance = 1.5;
Parameters.Inertia = 5e-4;

Motor = SimDCMotor(Parameters);

%% Initialization
% Clear all input and output from the Command Window
clc

% Many helper functions are provided in the `Interface` sub-directory.
% We start by adding that directory to Matlab's search path.
addpath('Interface');

% Set parameters for DC motor
Parameters.Resistance = 1.5; % Terminal resistance (Ohm)
Parameters.Inertia = 5e-4; % Rotor inertia (kg m2)

% Create an object handle for SimDCMotor with specified parameters
Motor = SimDCMotor(Parameters);
%% System Identification : Step Response
```



```
% The first method for system identification is to observe the step
% response of the system. To do so, we create a delayed step signal, input
% it as the voltage to the motor, and observe its angular velocity, and
% plot the output signal (in time).
```

```
Motor.reset(); % reset the motor internal variable (including time)
```

```
voltage = 1;
delay = 1;
duration = 5;
```

```
% Drive Motor with a voltage for a duration starting from delay
Motor.drive(voltage, delay, duration);
```

```
%=== Plot example ===%
t = Motor.time;
y = Motor.velocity;
u = Motor.voltage;
figure(1)
clf;
yyaxis left
plot(t, y)
ylim([0 1.3*y(end)])
yyaxis right
plot(t, u)
ylim([0 1.3*u(end)])
stepinfo(y, t, 'SettlingTimeThreshold', 0.05)
legend(['Speed (' Motor.Units '/s)'], 'Voltage (V)')
grid on
```

Code for Q3

```
%% System Identification : Frequency Analysis
```

```
% [0.01, 0.1, 0.2, 0.4, 0.5, 0.7, 1.0, 2.5, 5.0]
```

```
Motor.reset();
```

```
dt = 0.01; % Choose an appropriate sampling time (in seconds)
T = 6; % Choose an appropriate total duration of simulation (in seconds)
time = 0:dt:T; % A vector containing all time samples
```

```

Omega = 5*2*pi; %this is for 5 Hz frequency
for t = time
    % Generate a cosine wave input at current time
    u = cos(Omega*t);

    % Drive motor for a duration of dt
    Motor.drive(u, t, dt);
end

```

```

%=== Plot example ===%

```

```

t = Motor.time;
y = Motor.velocity;
u = Motor.voltage;
delay = 5; % wait for output to reach steady-state.
figure(2)
clf;
hold on;
plot(t(t > delay), y(t > delay))
plot(t(t > delay), u(t > delay))
xlim([5,T])
legend(['Speed (' Motor.Units '/s)'], 'Voltage (V)')
grid on
% ginput(4)

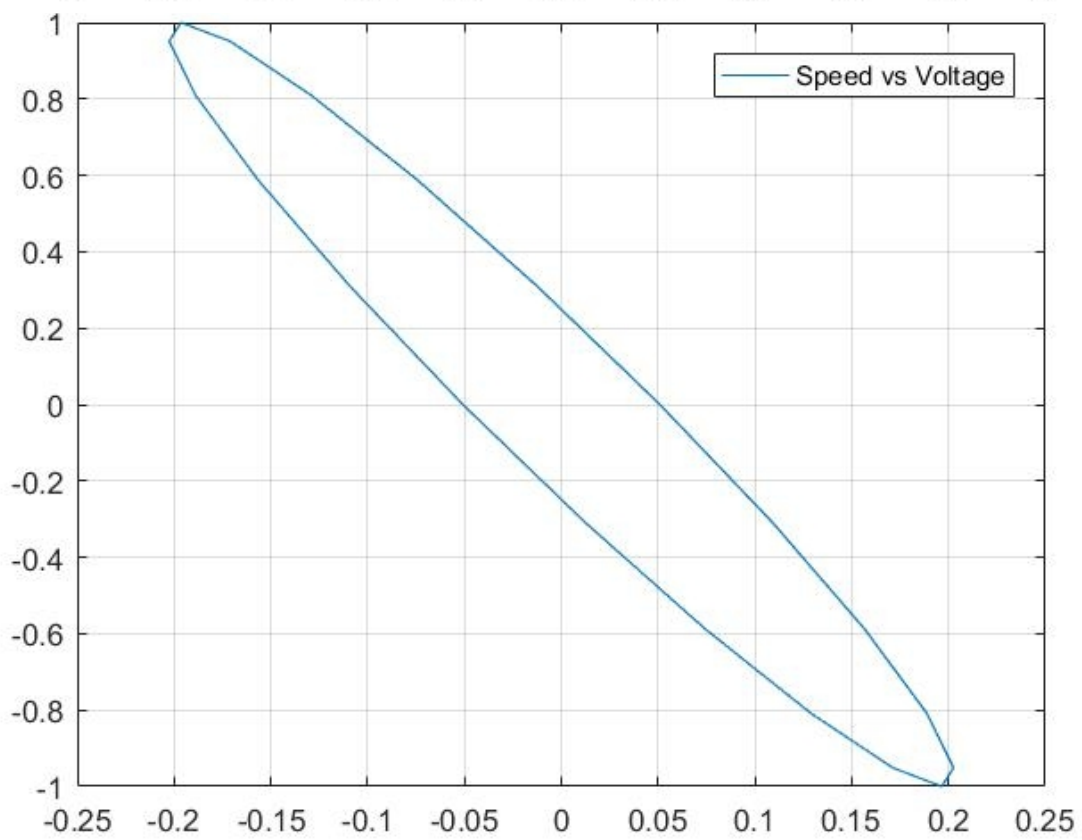
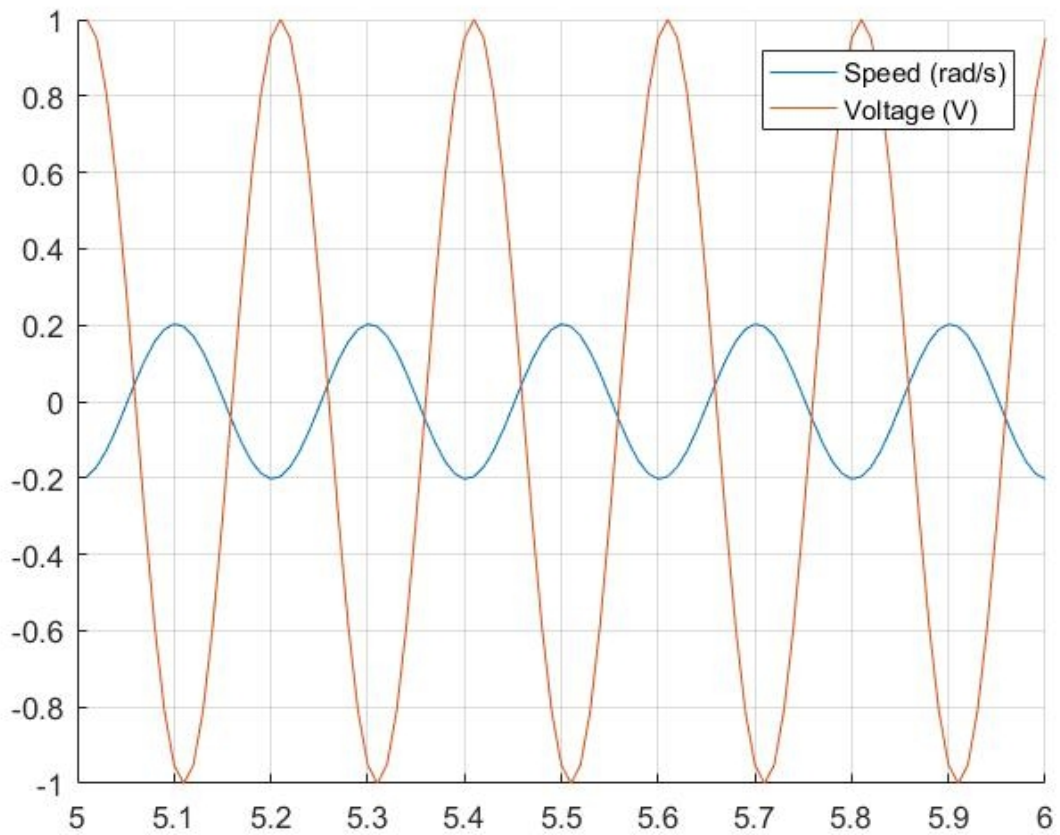
```

```

figure(3) % Lissajous method
clf;
plot(y(t > delay), u(t > delay))
legend('Speed vs Voltage')
grid on
% ginput(2)

```

Sample Output for 5 Hz



%%To get Bode Plot

```

Motor = SimDCMotor(Parameters);
%%
freq = [0.01
0.1
0.2
0.4
0.5
0.7
1.0
2.5
5.0
];

b = [20
20.02
19.83
10.73
0.6421
-6.981
-4.716
-0.7967
-0.1965
];

Ay = [20
20.42
21.66
23.81
21.2
12.37
5.709
0.8269
0.2026
];

figure(10)
subplot(2,1,1)

% Magnitude Bode Plot
semilogx(freq, 20*log10(Ay));
title('Magnitude (db) vs Frequency (Hz)')

```

```
subplot(2,1,2)
```

```
% Phase Bode Plot
```

```
semilogx(freq, -acos(b./Ay)*180/pi);
```

```
title('Phase (deg) vs Frequency (Hz)')
```