Characterisation of Simulated Spring Mass Damper System Using Step and Frequency Response Tests

Bonga Njamela[†]
EEE3094S 2023
University of Cape Town
South Africa
[‡]NJMLUN002

Abstract*—This paper describes the implementation of a step test and frequency response test in identifying the transfer function of a simulated spring mass damper system. Transfer functions mathematically model the behaviour of a system in the s-plane for a given input and provide a robust description of the stability and speed of system based on the location of a system's poles and zeros. Additionally, transfer functions offer a more simplistic overview of an arbitrary system compared to description of the system using differential equations. The transfer function obtained from the step test in the time domain was confirmed from the results of the frequency response test which describes how the system output responds to sinusoidal inputs of different frequencies. Assuming zero initial conditions and that all parameters were scaled to the mass m of the system, the experiment recorded values of the displacement of the mass to characterise the response of the system with particular interest in its steady state behaviour. The time domain response was analysed to determine the gain A, natural frequency ω_n, and damping coefficient ζ. The resulting damping coefficient and natural frequency were used to determine the damping frequency ω_d. The implementation of the step test showed that the gain of the system was 0.631, while the natural frequency was $2.209\,\mathrm{rad\,s^{-1}}$. Subsequently, the damping coefficient b and spring constant k were determined to be 2.709 and 1.487, respectively. These results were confirmed by the frequency response test whose Bode diagram was compared to a simulated version of the transfer function in MATLAB.

I. Introduction

The following paper describes the study of system identification conducted using two tests, namely the step test and frequency response test. System identification is a technique for constructing mathematical models of a dynamic system modelled by differential equations. This process of characterising the behaviour is done by measuring values of the input and output signals of the system in the time or frequency domain to determine the behaviour of the system. In this study, the system identification procedure was employed to characterise the altitude of a jet in response to an applied force. The system was modelled by a simulated spring-mass damper as illustrated by Fig 1.

An application of Newton's Second Law of motion to the physical system representing the spring-mass damper produces the second-order differential equation

$$m\frac{d^2x(t)}{dt^2} + b\frac{dx(t)}{dt} + kx(t) = F(t) \tag{1}$$

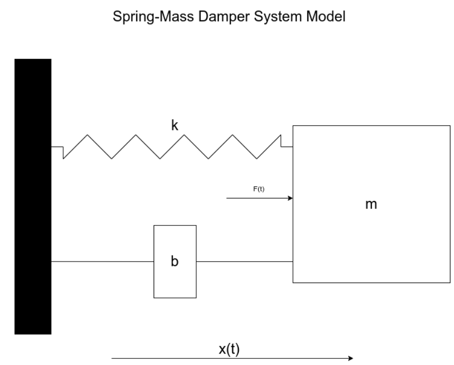

Fig. 1: The simulated spring-mass damper model is graphically illustrated by a moving mass m in response to the force $F(t)$. The damping coefficient b is modelled by the block labelled 'b' while the stiffness constant k is represented by the spring labeled 'k'.

where m is the mass, k is the spring constant, and b is the damping coefficient. This model is useful in describing the systems behaviour with respect to position, velocity and acceleration, which adds an extra layer of complexity to the analysis of the behaviour. Another shortcoming of using model to describe continuous time linear time-independent systems based on differential equations is that they are limited to describing the system in the time domain. A more useful generic model, which was implemented in this study, derives the behaviour of the system in the frequency domain based on the ratio between the input to the system and its output. This ratio is known as the transfer function G and can be derived directly from the unit step response. The transfer function provides useful information about the stability of the system from the location of the poles in the s-plane.

Assuming zero initial conditions and that the mass of the system is 1, measurements of the displacement of the mass were recorded for a given input voltage considered to be directly proportional to the applied force. In the step test, incremental levels of a step input were applied to the system and the response was recorded to analysis the output with particular interest in the steady-state where the influence of transient effects is reduced. The transfer function obtained from the step test was compared to the results obtained from the frequency model test which performs the system identification procedure in the frequency domain.

Frequency response refers to the response of the system to sinusoidal inputs with an amplitude of 1 at different frequencies. The value of the output displacement was tracked continuously and the resulting amplitude ratio between the input and the output was used to construct the magnitude plot of the Bode diagram. The phase difference between the input and output was used to determine the change in phase as the frequency changed. This Bode diagram obtained from the experiment was compared to the Bode diagram obtained using the equation of the transfer function from the step test.

This paper includes the methodology implemented in both experimental tests for determine the damping coefficient and spring constant which characterise the behaviour of the simulated spring-mass damper system. Following the description of the system identification procedure and estimation method verification, the results are discussed and a conclusion is drawn on the behaviour of the system and the effectiveness of the identification models.

II. METHODOLOGY

A. Step Test to Estimate Gain and Natural Frequency in the Transfer Function of the System

A spring-mass damper system as described by equation (1), was simulated with zero initial conditions and a mass $m = 1$. A step input was introduced to the system and the value of the response was measured over a suitable time, allowing for the system to reach steady-state. The maximum value of the step function was increased to from 1 to 5, and the results for each input were compared to determine the precision of the method. The aim of the step test was to determine the gain of the system, define the poles and zeros of the system as well as to determine the corner frequency of the system. This information was used to estimate the damping coefficient b and spring constant k from the transfer function $G(s)$ which characterised the response of the system to any input force $F(t)$.

The generic form of the transfer function of the second-order system was derived from the differential equation (1) using the Laplace transformation

$$\begin{aligned} &m[s^2X(s) - sx(0^-) - \dot{v}(0^-)] \\ &+b[sX(s) - x(0^-)] + kX(s) = F(s) \end{aligned}$$

Since the simulation supposes zero initial conditions, the terms related to the initial speed and position were neglected, leaving

$$ms^2X(s) + bsX(s) + kX(s) = F(s) \tag{2}$$

which corresponds to the forced response of a second-order system, i.e. the response of the system due to the input.
The generic form of the second-order response transfer function in the s-plane is given by

$$G(s) = \frac{X(s)}{F(s)} = \frac{A\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{3}$$

where ω_n is the natural frequency, ζ is the damping ratio, and A is the gain. The Laplace transform in equation (2) was expressed in the same form as equation (3) by writing

$$\omega_n = \sqrt{\frac{k}{m}} \tag{4}$$

$$\zeta = \frac{b}{2m\omega_n} \tag{5}$$

$$U(s) = F(s)/k \tag{6}$$

Application of the Final Value Theorem showed that the gain A could be obtained in the limit as $s \to 0$

$$A = \lim_{s\to 0} s \cdot G(s) \cdot U(S) \tag{7}$$

corresponding to the value of the output displacement in the steady-state. The experiment estimated the gain graphically from the value of the output in the steady state minus the initial value.

The value of the damping ratio was also obtained graphically from the plot of the output displacement over a period of time. Given that the peak output displacement x_p can be expressed with respect to the gain A as,

$$x_p = A\left(1 + \exp\left\{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right\}\right) \tag{8}$$

the damping ratio was determined by substituting the gain obtained from the steady-state displacement. The damping frequency ω_d is related to the natural frequency ω_n and the damping frequency ζ by

$$\omega_d = \omega_n\sqrt{1-\zeta^2} \tag{9}$$

This damped frequency is related to the time t_p where the displacement reaches its peak value by

$$\omega_d = \frac{\pi}{t_p} \tag{10}$$

The experiment used the graphical peak time t_p to calculate the damped frequency ω_d using equation (10), followed by a computation of the natural frequency ω_n using equation (9). The experimental values of the natural frequency ω_n and the damping coefficient ζ were used to find the spring constant k and damping coefficient b.

B. Result Verification Using Bode Diagram

A Bode diagram was constructed using a frequency response test which performed a stepped-sine sweep where measurements of the output displacement were first made at lower frequencies followed by a small increments for a suitable frequency range. When the input force $F(t)$ is a sinusoid with amplitude equal to 1, its Laplace transform is

$$\mathcal{L}\{u(t)\} = U(s) = \mathcal{L}\{\sin(\omega t)\} = \frac{\omega}{s^2+\omega^2}$$

The output response is then

$$X(s) = \frac{\omega G(s)}{s^2+\omega^2} \tag{11}$$

A combination of algebraic manipulations and an inverse Laplace transform yields the response

$$x(t) = \rho \sin(\omega t + \theta) \tag{12}$$

implying that the output displacement in response to a sinusoidal input is also a sinusoid, albeit with a phase shift θ and scaled by the transfer coefficient ρ. Hence, the frequency response test was used to construct a Bode frequency response model of the system using the magnitude of the transfer function $\rho(j\omega)$ at a given frequency ω and the phase difference between the input force and the output displacement

$$\rho(j\omega) = \frac{\text{output displacement at } \omega}{\text{input force at } \omega} \tag{13}$$

$$\theta(j\omega) = \arg G(j\omega) \tag{14}$$

The experiment used a logarithmic scale to find the magnitude of the transfer function, or gain, at different frequencies. The shape of the Bode plots constructed from the results of the frequency test was compared to the shape obtained from theoretical computations.

C. Expected Response Characteristics

The shape of the response was expected to depend on the position of the poles in the complex s-plane. The position of the poles also indicates the stability of the system. Typically, systems are designed to have poles in the left-half of the s-plane as shown in Fig 2.

The location of the poles was given by

$$p_1, p_2 = -\zeta\omega_n \pm j\omega_d \tag{15}$$

It was also noted that shape of the response depended on the position of the poles. If the system was underdamped ($0 < \zeta < 1$), it was expected to have the shape of the blue plot in Fig 3. Similarly, an overdamped system ($\zeta > 1$) was expected to have a shape following the red curve in the diagram.

Simulations of the experimental transfer function in MATLAB were used to compare the graphical model

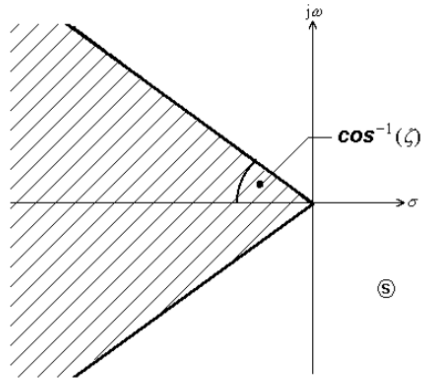

Fig. 2: The shaded region shows the region of suitable pole locations for a stable second-order system.

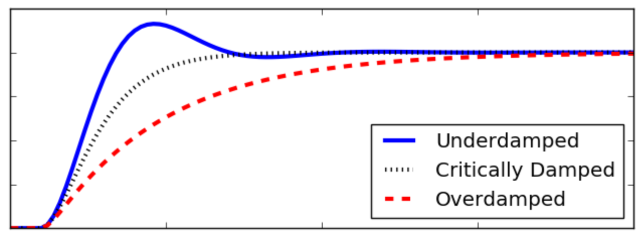

Fig. 3: Whether the system is underdamped, overdamped, or critically damped depends on the value of the damping coefficient ζ.

constructed from measurements of the output displacement to the expected model from theoretical interpolation.

III. Results

A. Transfer Function from Step Response Test

The system response to an input step is illustrated in figure 4. When the value of the input step was 1 V, the overshoot x_p was equal to 0.686 and the steady-state peak value produced a gain A of 0.631. Equation (8) was used to determine the value of the damping coefficient ζ as 0.613 following

$$0.686 = 0.631\left(1 + \left\{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right\}\right)$$
$$\implies \zeta = 0.613$$

The overshoot x_p was reached after time t_p equal to $1.8\,\mathrm{s}$. Substitution of time t_p into equation (10) produced a damped frequency ω_d of $1.745\,\mathrm{rad\,s^{-1}}$. Subsequently, the natural frequency ω_n was determined from equation (9) to be $2.21\,\mathrm{rad\,s^{-1}}$. The values resulting from the response to increments of the input step were found to be the same, showing satisfactory simulation precision. The gain increased proportionally with an increase input step. For example, a look at figure 4 shows that when the step was increased by 2 units from a level of 1 to 3, the output displacement in the steady-state was equal to $1.262\,\mathrm{m}$, corresponding to double the gain obtained when the input step was at a level of 1.

Additionally, values of the damping coefficient ζ were used to verify that the system was underdamped as indicated by the shape of the response. This was confirmed by the fact that ζ was less than 1.

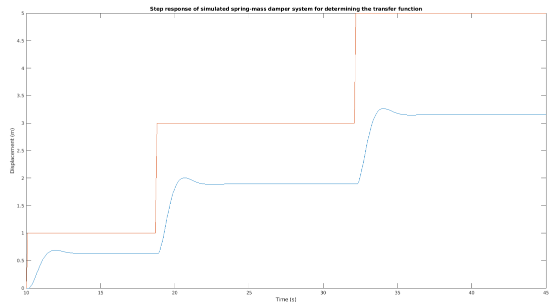

Fig. 4: Step test with voltage changing from 1 to 3, then 5V. A change in the input voltage corresponded to a proportional change in the applied force. The gain A, overshoot (or peak value) x_p and peak time t_p were obtained from the response shown in blue.

1) Transfer Function Modelling Spring-Mass Damper System: The values obtained in the step test produced a transfer function $G(s)$ given by

$$G(s) = \frac{3.08}{s^2 + 2.71s + 4.88} \tag{16}$$

The shape of the step response modelled by the experimentally derived transfer function was verified using a simulation in MATLAB. Using the software's `step` function produced a step response as shown in Figure 5. The correlation between the experimental step response and the simulated step response was 0.999, showing a satisfactory estimation of the system parameters. Since all parameters were assumed to be scaled

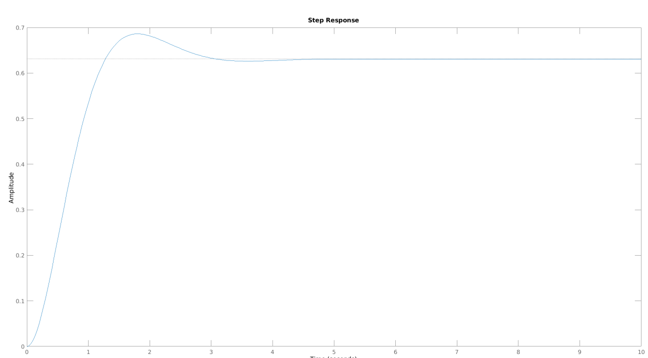

Fig. 5: MATLAB simulated step response using transfer function in equation (16). The shape of the response, the steady-state peak value, peak time, as well as the maximum overshoot correspond to the experimental results.

to the mass of the system, i.e. the mass m was taken as unity, calculations of the spring constant k and damping ratio b were easily evaluated by substituting the natural frequency ω_n and damping coefficients ζ in equation (4) and (5). The resulting system parameters in the differential equation modelling the motion of the spring-mass were recorded in Table II. Table I summarises the results obtained directly from the experimental step response.

Parameter	**Description**	**Experimental Value**
A	Gain	0.631
ω_n	Natural Frequency in $\mathrm{rad\,s^{-1}}$	2.21
ζ	Damping Coefficient	0.613
x_p	Response Amplitude Overshoot	0.686
t_p	Peak Time in s	1.8
ω_d	Damped Frequency in $\mathrm{rad\,s^{-1}}$	1.745
t_r	Rise Time in s	1.28

TABLE I: Summarises results obtained directly from the measured values of the output displacement in the position of the simulated spring-mass damper system after performing the step response test.

Parameter	**Description**	**Experimental Value**
b	Damping Coefficient	2.709
k	Spring Constant	1.487
m	Mass	1

TABLE II: Showing the coefficients of the differential equation modelling the spring-mass damper system described in equation (1).

Any change to the physical parameters given in table II would results in a different behaviour of the system all together.

B. System Stability Analysis From Pole Position

A stable, underdamped second-order spring-mass damper system given in the form of equation (3) has two poles located in the left-half s-plane as described by figure 2. The resulting poles from the experiment were found to be located in the left-half of the s-plane as shown in the pole plot in figure 6. This angles subtended by the straight line vectors from the origin to the poles p_1 and p_2 and the real axis were within the desirable region as shown in figure 2.

This was expected from the shape of the response which indicated stability by a bounded steady-state maximum and that the system had complex poles due to the presence of overshoot.

C. Result Validation using Frequency Response Test

The aim of the frequency response test was to validate the transfer function obtained in the previous section by constructing Bode diagrams from the sinusoidal response of the system. The frequency of a sinusoid with an amplitude of 1 was increased from $0.2\,\mathrm{rad\,s^{-1}}$ to $100\,\mathrm{rad\,s^{-1}}$ in suitable increments highlighting the region of the Bode magnitude plot near the natural frequency. The Bode diagram was simulated

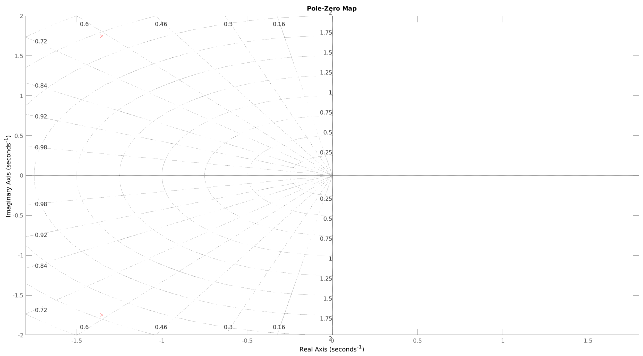

Fig. 6: Pole plot showing the location of the poles (red crosses) in the left-half of the s-plane, indicating that the system was stable. Note that the system has no zeros.

in MATLAB using the result expressed in equation (13) and compared to the Bode diagram obtained from experimentation with different frequencies. This theoretical Bode diagram is shown in figure 7 and compared to the actual Bode diagram depicted in figure 11 and **??**. Figure 8 and 9 show the

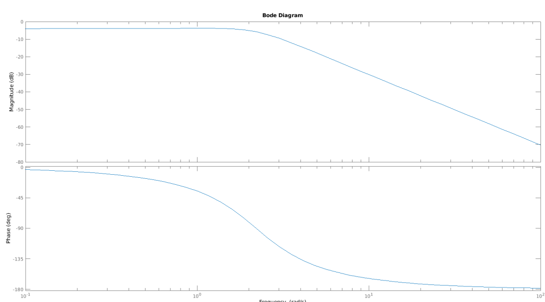

Fig. 7: A vertical line is drawn at the natural frequency $2.21\,\mathrm{rad\,s^{-1}}$, where the phase is $-90°$ and the roll-off becomes $-40\,\mathrm{dB}$/decade. This expected Bode diagram was compared to the experimental Bode plot to verify the transfer function obtained in equation (13) by way of a frequency response test.

response of the system to an input sinusoid with frequency of $0.220\,\mathrm{rad\,s^{-1}}$ and $20.0\,\mathrm{rad\,s^{-1}}$, respectively.

Near the natural frequency ω_n at $10^{0.334}$, the phase of the response was $-90°$ as shown in figure 10.

The Bode diagram was constructed from the response of the system at different frequencies by evaluating the logarithm of the amplitude ratio and the inverse tangent of the measured phase difference. The experimental Bode diagram shown in figures 11 and 12. Errors in approximation can be accounted for by difficulty in obtaining accurate gain and phase difference values for inputs that oscillate rapidly.

The shape of the Bode diagram approximated the expected shape, showing that the system identification method applied was satisfactory in modelling the behaviour of the spring-mass damper system.

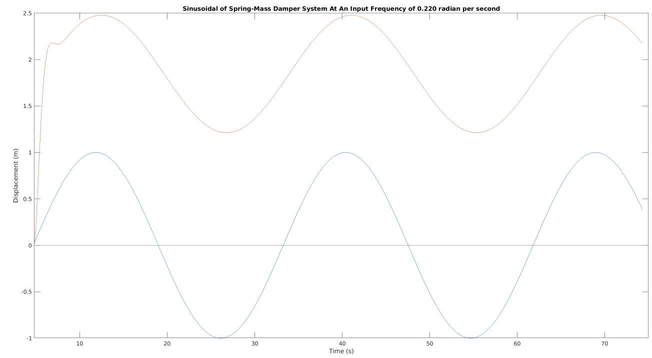

Fig. 8: At lower frequencies, the output displacement oscillated with a phase that was similar to the phase of the input sine wave. This frequency response corresponds to measurements of the output displacement when the input sine wave frequency was $0.22\,\mathrm{rad\,s^{-1}}$.

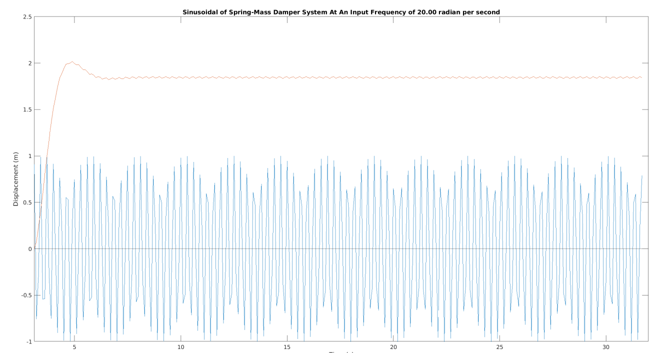

Fig. 9: At higher frequencies, the system response approximated the shape of the step response as high frequencies are attenuated. This frequency response corresponds to measurements of the output displacement when the input sine wave frequency was $20.0\,\mathrm{rad\,s^{-1}}$.

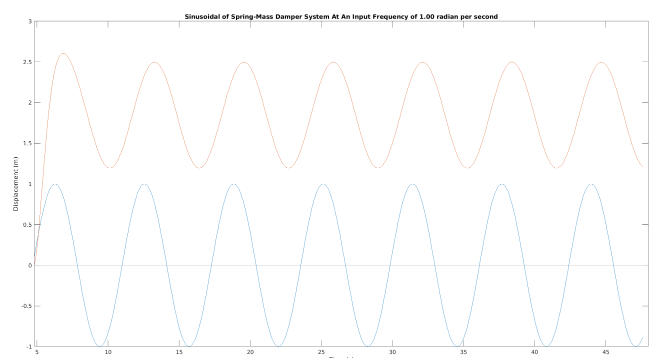

Fig. 10: When the input sine wave frequency was $1.0\,\mathrm{rad\,s^{-1}}$, the response and the input were almost completely out of phase. The figure also shows that the response lags behind the input.

IV. Conclusion

In conclusion, the experiment demonstrated the successful implementation of both a step test and a frequency

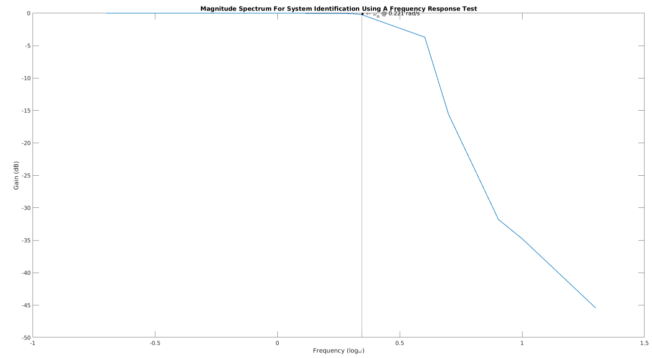

Fig. 11: Bode magnitude plot obtained from the ratio of the response to a manual sinusoidal frequency sweep. The shape of the diagram approximates the expected shape as shown if figure 7.

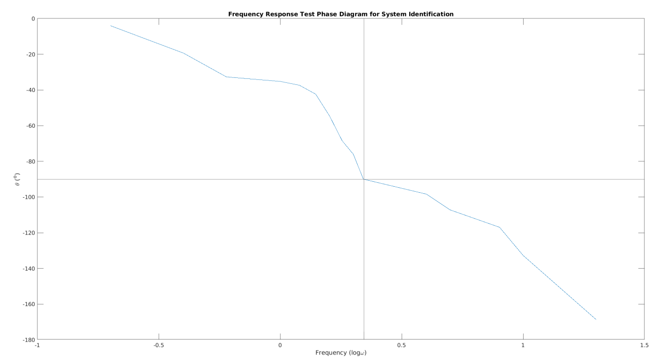

Fig. 12: Bode phase plot obtained from the phase difference between the output and the input at steady-state. The shape tends to the expected curve with few outliers.

response test for identifying the transfer function of a simulated spring-mass-damper system. Transfer functions offer a powerful mathematical framework for modelling system behaviour in the complex s-plane, providing insights into stability and response speed through the analysis of poles and zeros. Moreover, they offer a more accessible and concise representation of system dynamics compared to differential equations.

The results obtained from the step test in the time domain were consistent with those derived from the frequency response test, ensuring the reliability of the identified transfer function. The experiments, conducted with zero initial conditions and parameter scaling to the system's mass, focused on characterizing the system's response, particularly its steady-state behaviour. The poles of the system were located suitably in the left-half plane of the system. Locations also corresponded to the shape of the response, pointing to an adequate implementation of the step test in system identification.

Through the analysis of the time domain response, key system parameters, including the gain A, natural frequency ω_n, and damping coefficient ζ, were determined. These values enabled the calculation of the damping frequency ω_d. In particular, the system's gain was found to be 0.631, while the natural frequency was measured at $2.209\,\mathrm{rad\,s^{-1}}$. Subsequently, the damping coefficient b and spring constant k were determined to be 2.709 and 1.487, respectively.

The accuracy of these findings was evaluated thorough a comparison of the frequency response test results, presented in a Bode diagram, with a simulated version of the transfer function implemented in MATLAB. This comprehensive approach to system identification highlighted the effectiveness of the step test and frequency response test in accurately characterizing the dynamics of the spring-mass-damper system.

APPENDIX

Appendix A: MATLAB Code for Data Processing

Listing 1: MATLAB Script implemented in analysing the results from the step and frequency response tests.

```
%Set up the Import Options and import the data
opts = delimitedTextImportOptions("NumVariables", 3);

% Specify range and delimiter
opts.DataLines = [2, Inf];
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["Times", "Input", "Output_Displacement"];
opts.VariableTypes = ["double", "double", "double"];

% Specify file level properties
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Import the data
step_resp = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/controllab1rawdata/F34BSuspensionTestData09_STEP1-3-5.CSV", opts);
sin_002 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_02.CSV", opts);
sin_004 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_04.CSV", opts);
sin_006 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_06.CSV", opts);
sin_008 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_08.CSV", opts);
sin_010 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_10.CSV", opts);
sin_012 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_12.CSV", opts);
sin_014 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_14.CSV", opts);
sin_016 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_16.CSV", opts);
sin_018 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_18.CSV", opts);
sin_020 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_20.CSV", opts);
sin_022 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_22.CSV", opts);
sin_040 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_40.CSV", opts);
sin_100 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_100.CSV", opts);
sin_500 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_500.CSV", opts);
sin_1000 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_1000.CSV", opts);
sin_2000 = readtable("/home/bonga/Documents/EEE3094S/EEE3094S_Lab_01/recontrollab1rawdata/F34BSuspensionTestData_2000.CSV", opts);

Clear temporary variables
clear opts


Step Response Test Analysis
step_time = step_resp.Times(13:end);
step_input = step_resp.Input(13:end);
step_output = step_resp.Output_Displacement(13:end);
step_output = step_output - 5;

plot(step_time, step_output);
xlabel("Time (s)");
ylabel("Displacement (m)");
xlim([10 45]);
title("Step response of simulated spring-mass damper system for determining the transfer function");
hold on
plot(step_time, step_input);
hold off

Data processing of Frequency Response Test results performed on Simulated Spring-Mass Damper System
% Input sine wave frequency = 0.020 radian per second
time_002 = sin_002.Times;
input_002 = sin_002.Input;
output_002 = sin_002.Output_Displacement;

% Plot response of system to input 0.020 radian per second sine wave
time_002 = time_002(32:end); %start of test
input_002 = input_002(32:end);
output_002 = output_002(32:end) - 5;

plot(time_002,input_002);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 0.020 radian per second");
yline(0);
xlim([26.1 300]);
hold on
plot(time_002, output_002);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 0.020 rad per second

gain_002 = output_002./input_002;
gain_002_db = 20*log10(abs(gain_002));

sin_002_gain = table(time_002, input_002, gain_002_db);
plot(input_002, gain_002_db)

% Input sine wave frequency = 0.040 radian per second
time_004 = sin_004.Times;
input_004 = sin_004.Input;
output_004 = sin_004.Output_Displacement;

% Plot response of system to input 0.040 radian per second sine wave
time_004 = time_004(43:end); %start of test
input_004 = input_004(43:end);
output_004 = output_004(43:end) - 3.155;

plot(time_004,input_004);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 0.040 radian per second");
yline(0);
xlim([29.1 225]);
%xline(107.4);
hold on
```

```
plot(time_004, output_004);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 0.040 rad per second

gain_004 = output_004./input_004;
gain_004_db = 20*log10(abs(gain_004));

sin_004_gain = table(time_004, input_004, gain_004_db);
plot(input_004, gain_004_db)

% Input sine wave frequency = 0.060 radian per second
time_006 = sin_006.Times;
input_006 = sin_006.Input;
output_006 = sin_006.Output_Displacement;

% Plot response of system to input 0.060 radian per second sine wave
time_006 = time_006(52:end); %start of test
input_006 = input_006(52:end);
output_006 = output_006(52:end) - 3.155;

plot(time_006,input_006);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 0.060 radian per second");
yline(0);
xlim([8.7 150]);
%xline(60.95);%t_d=0.85
hold on
plot(time_006, output_006);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 0.060 rad per second

gain_006 = output_006./input_006;
gain_006_db = 20*log10(abs(gain_006));

sin_006_gain = table(time_006, input_006, gain_006_db);
plot(input_006, gain_006_db)

% Input sine wave frequency = 0.080 radian per second
time_008 = sin_008.Times;
input_008 = sin_008.Input;
output_008 = sin_008.Output_Displacement;

% Plot response of system to input 0.080 radian per second sine wave
time_008 = time_008(39:end); %start of test
input_008 = input_008(39:end);
output_008 = output_008(39:end) - 3.155;

plot(time_008,input_008);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 0.080 radian per second");
yline(0);
%xline(105.7);
xlim([27.4 140]);
hold on
plot(time_008, output_008);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 0.080 rad per second

gain_008 = output_008./input_008;
gain_008_db = 20*log10(abs(gain_008));

sin_008_gain = table(time_008, input_008, gain_008_db);
plot(input_008, gain_008_db)

% Input sine wave frequency = 0.100 radian per second
time_010 = sin_010.Times;
input_010 = sin_010.Input;
output_010 = sin_010.Output_Displacement;

% Plot response of system to input 0.100 radian per second sine wave
time_010 = time_010(60:end); %start of test
input_010 = input_010(60:end);
output_010 = output_010(60:end) - 3.155;

plot(time_010,input_010);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 0.100 radian per second");
yline(0);
%xline(106.8);
xlim([44.1 132]);
hold on
plot(time_010, output_010);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 0.100 rad per second

gain_010 = output_010./input_010;
gain_010_db = 20*log10(abs(gain_010));

sin_010_gain = table(time_010, input_010, gain_010_db);
plot(input_010, gain_010_db)

% Input sine wave frequency = 0.120 radian per second
time_012 = sin_012.Times;
input_012 = sin_012.Input;
output_012 = sin_012.Output_Displacement;

% Plot response of system to input 0.120 radian per second sine wave
time_012 = time_012(47:end); %start of test
input_012 = input_012(47:end);
output_012 = output_012(47:end) - 3.155;
```

```
plot(time_012,input_012);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 0.120 radian per second");
yline(0);
%xline(63.99);
xlim([11.7 83]);
hold on
plot(time_012, output_012);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 0.120 rad per second

gain_012 = output_012./input_012;
gain_012_db = 20*log10(abs(gain_012));

sin_012_gain = table(time_012, input_012, gain_012_db);
plot(input_012, gain_012_db)

% Input sine wave frequency = 0.140 radian per second
time_014 = sin_014.Times;
input_014 = sin_014.Input;
output_014 = sin_014.Output_Displacement;

% Plot response of system to input 0.140 radian per second sine wave
time_014 = time_014(67:end); %start of test
input_014 = input_014(67:end);
output_014 = output_014(67:end) - 3.155;

plot(time_014,input_014);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 0.140 radian per second");
yline(0);
xlim([11.2 122]);
%xline(78.4);
hold on
plot(time_014, output_014);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 0.140 rad per second

gain_014 = output_014./input_014;
gain_014_db = 20*log10(abs(gain_014));

sin_014_gain = table(time_014, input_014, gain_014_db);
plot(input_014, gain_014_db)

% Input sine wave frequency = 0.140 radian per second
time_016 = sin_016.Times;
input_016 = sin_016.Input;
output_016 = sin_016.Output_Displacement;

% Plot response of system to input 0.160 radian per second sine wave
time_016 = time_016(68:end); %start of test
input_016 = input_016(68:end);
output_016 = output_016(68:end) - 3.155;

plot(time_016,input_016);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 0.160 radian per second");
yline(0);
xlim([8.8 112]);
%xline(87.4);
hold on
plot(time_016, output_016);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 0.160 rad per second

gain_016 = output_016./input_016;
gain_016_db = 20*log10(abs(gain_016));

sin_016_gain = table(time_016, input_016, gain_016_db);
plot(input_016, gain_016_db)

% Input sine wave frequency = 0.180 radian per second
time_018 = sin_018.Times;
input_018 = sin_018.Input;
output_018 = sin_018.Output_Displacement;

% Plot response of system to input 0.180 radian per second sine wave
time_018 = time_018(62:end); %start of test
input_018 = input_018(62:end);
output_018 = output_018(62:end) - 3.155;

plot(time_018,input_018);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 0.180 radian per second");
yline(0);
xlim([9.4 135]);
%xline(96.55);
hold on
plot(time_018, output_018);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 0.180 rad per second

gain_018 = output_018./input_018;
gain_018_db = 20*log10(abs(gain_018));

sin_018_gain = table(time_018, input_018, gain_018_db);
plot(input_018, gain_018_db)
```

```
% Input sine wave frequency = 0.200 radian per second
time_020 = sin_020.Times;
input_020 = sin_020.Input;
output_020 = sin_020.Output_Displacement;
% Plot response of system to input 0.200 radian per second sine wave
time_020 = time_020(51:end); %start of test
input_020 = input_020(51:end);
output_020 = output_020(51:end) - 3.155;

plot(time_020,input_020);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 0.200 radian per second");
yline(0);
%xline(68.6);
xlim([5.9 105]);
hold on
plot(time_020, output_020);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 0.200 rad per second

gain_020 = output_020./input_020;
gain_020_db = 20*log10(abs(gain_020));

sin_020_gain = table(time_020, input_020, gain_020_db);
plot(input_020, gain_020_db)

% Input sine wave frequency = 0.220 radian per second
time_022 = sin_022.Times;
input_022 = sin_022.Input;
output_022 = sin_022.Output_Displacement;

% Plot response of system to input 0.220 radian per second sine wave
time_022 = time_022(41:end); %start of test
input_022 = input_022(41:end);
output_022 = output_022(41:end) - 3.155;

plot(time_022,input_022);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 0.220 radian per second");
yline(0);
%xline(61.8);
xlim([4.8 75]);
hold on
plot(time_022, output_022);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 0.220 rad per second

gain_022 = output_022./input_022;
gain_022_db = 20*log10(abs(gain_022));

sin_022_gain = table(time_022, input_022, gain_022_db);
plot(input_022, gain_022_db)
xlabel("Applied Force");
ylabel("Gain");
title("Plot of gain vs applied force to determine gain at maximum force.");

% Input sine wave frequency = 0.400 radian per second
time_040 = sin_040.Times;
input_040 = sin_040.Input;
output_040 = sin_040.Output_Displacement;

% Plot response of system to input 0.400 radian per second sine wave
time_040 = time_040(33:end); %start of test
input_040 = input_040(33:end);
output_040 = output_040(33:end) - 3.155;

plot(time_040,input_040);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 0.400 radian per second");
yline(0);
xlim([4.5 51]);
%xline(35.8);
hold on
plot(time_040, output_040);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 0.400 rad per second

gain_040 = output_040./input_040;
gain_040_db = 20*log10(abs(gain_040));

sin_040_gain = table(time_040, input_040, gain_040_db);
plot(input_040, gain_040_db)

% Input sine wave frequency = 1.00 radian per second
time_100 = sin_100.Times;
input_100 = sin_100.Input;
output_100 = sin_100.Output_Displacement;

% Plot response of system to input 1.00 radian per second sine wave
time_100 = time_100(28:end); %start of test
input_100 = input_100(28:end);
output_100 = output_100(28:end) - 3.155;

plot(time_100,input_100);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 1.00 radian per second");
yline(0);
%xline(17.3);
%xline(23.6);
xlim([4.8 48]);
```

```
hold on
plot(time_100, output_100);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 1.00 rad per second

gain_100 = output_100./input_100;
gain_100_db = 20*log10(abs(gain_100));

sin_100_gain = table(time_100, input_100, gain_100_db);
plot(input_100, gain_100_db)

% Input sine wave frequency = 5.00 radian per second
time_500 = sin_500.Times;
input_500 = sin_500.Input;
output_500 = sin_500.Output_Displacement;

% Plot response of system to input 5.00 radian per second sine wave
time_500 = time_500(20:end); %start of test
input_500 = input_500(20:end);
output_500 = output_500(20:end) - 3.155;

plot(time_500,input_500);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 5.00 radian per second");
yline(0);
%xline(49.15);
xlim([28.5 60]);
hold on
plot(time_500, output_500);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 5.00 rad per second

gain_500 = output_500./input_500;
gain_500_db = 20*log10(abs(gain_500));

sin_500_gain = table(time_500, input_500, gain_500_db);
plot(input_500, gain_500_db)

% Input sine wave frequency = 10.000 radian per second
time_1000 = sin_1000.Times;
input_1000 = sin_1000.Input;
output_1000 = sin_1000.Output_Displacement;

% Plot response of system to input 10.00 radian per second sine wave
time_1000 = time_1000(91:end); %start of test
input_1000 = input_1000(91:end);
output_1000 = output_1000(91:end) - 3.155;

plot(time_1000,input_1000);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 10.0 radian per second");
yline(0);
xlim([10.4 38]);
hold on
plot(time_1000, output_1000);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 10.00 rad per second

gain_1000 = output_1000./input_1000;
gain_1000_db = 20*log10(abs(gain_1000));

sin_1000_gain = table(time_1000, input_1000, gain_1000_db);
plot(input_1000, gain_1000_db);

% Input sine wave frequency = 20.000 radian per second
time_2000 = sin_2000.Times;
input_2000 = sin_2000.Input;
output_2000 = sin_2000.Output_Displacement;

% Plot response of system to input 20.00 radian per second sine wave
time_2000 = time_2000(21:end); %start of test
input_2000 = input_2000(21:end);
output_2000 = output_2000(21:end) - 3.155;

plot(time_2000,input_2000);
xlabel("Time (s)");
ylabel("Displacement (m)");
title("Sinusoidal of Spring-Mass Damper System At An Input Frequency of 20.00 radian per second");
yline(0);
xlim([3.1 32]);
hold on
plot(time_2000, output_2000);
hold off

% Calculate gain in decibel from the ratio of the output to the input for
% frequency of 20.00 rad per second

gain_2000 = output_2000./input_2000;
gain_2000_db = 20*log10(abs(gain_2000));

sin_2000_gain = table(time_2000, input_2000, gain_2000_db);
plot(input_2000, gain_2000_db);

Plot Bode Diagram
% Tabulate results of gain vs frequency to plot magnitude spectrum
Frequency = [0.2; 0.4; 0.6; 0.8; 1.00; 1.20; 1.40; 1.60; 1.80; 2.00; 2.20; 4.00; 5.00; 8.00; 10.00; 20.00];
Frequency = log10(Frequency);
Gain = [7.875;7.875;7.875;7.875;7.875;7.875;7.871;7.871;7.871;7.868;7.854;7.505;6.318;4.695;4.399;3.33];
Gain = (Gain - 7.875)*10;
%Gain = [19.780;19.084;16.258;13.760;11.821;10.238;8.899;7.739;6.716;5.801;4.973;-0.220;-8.135;-20.654;-21.329;-19.74];
%Gain = Gain - 19.780;
magnitude_table = table(Frequency, Gain);
```

```
% Plot magnitude spectrum
magnitude_plot = plot(Frequency, Gain, 'LineWidth',1);
xlabel("Frequency (log\omega)");
ylabel("Gain (dB)");
xline(log10(2.21));
title("Magnitude Spectrum For System Identification Using A Frequency Response Test");

% Add arrow pointing to cut-off frequency in Bode magnitude spectrum
txt = '\bullet␣\leftarrow␣\omega_n␣@␣0.221␣rad/s';
text(Frequency(11),Gain(11),txt);

% Create column for phase change correponding to change in frequency
Phase = [-4;-19.38;-32.66;-34.13;-35.22;-37.37;-42.36;-54.8;-68.51;-76.02;-90;-98.38;-107.19;-116.96;-132.72;
magnitude_table.Phase = Phase;

% Plot Phase Spectrum using log\omega values
plot(Frequency, Phase);
xlabel("Frequency (log\omega)");
ylabel("\theta (^o)");
xline(log10(2.21));
yline(-90);
%ylim([1 -95]);
title("Frequency Response Test Phase Diagram for System Identification");
```