Mechatronic Systems Identification

Lab 5 - Non-parametric Identification

Khaldoun Fayad - 409597

Witold Surdej - 407100

16.04.2024

Description

The aim of this laboratory is for us to learn the non-parametric system identification methods.

Task 1

This task aims to prepare a mathematical model describing the RLC circuit showcased below

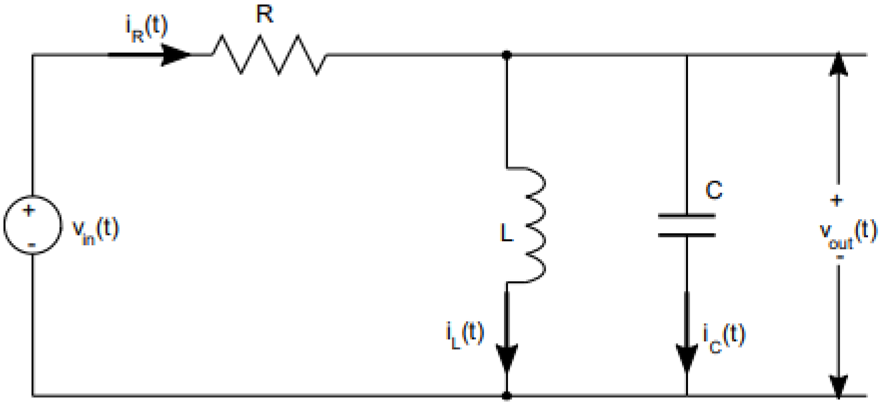

fig 1: parallel RLC circuit

by determining the relationship between $v(t)_{out}$ and $v(t)_{in}$ using the properties of electrical circuits. Then, to implement said model in the Simulink environment.

Task 1.1

The initial equations were derived on paper. By applying Kirchhoff's 1st and 2nd laws as well as the properties of the parallel connection of L and C we obtain the following analytical equation of the relationship between $v(t)_{out}$ and $v(t)_{in}$:

$$v_{out}(t) = v_C(t) = v_L(t)$$

$$v_{in}(t) = (\frac{1}{L}\int_0^t v_L(\tau)d\tau + C\dot{v}_c(t))R + v_{out}(t)$$ where

by taking the Laplace transform of the equation we obtain the transmittance

$$V_{in}(s) = (\frac{R}{Ls} + CRs + 1)V_{out}(s)$$

$$T(\omega) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{\frac{R}{Ls}+CRs+1} = \frac{s}{CRs^2+s+\frac{R}{L}}$$

Using the derived equations the following Simulink model was implemented.

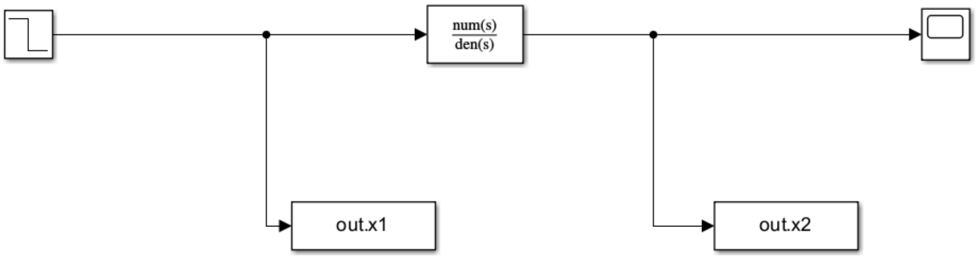

fig 2: Simulink model of the system.

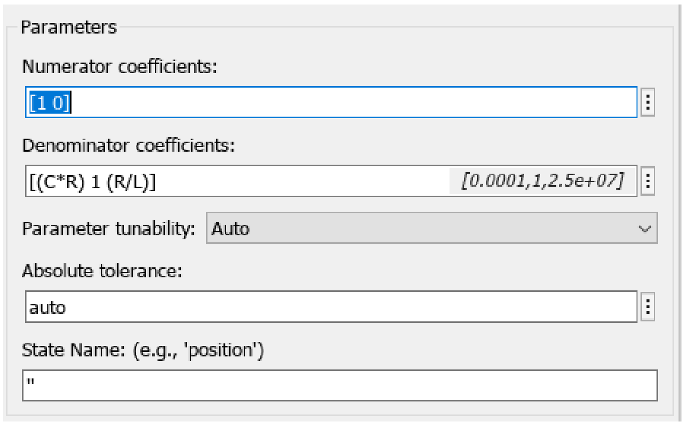

fig 3: Parameters of the transfer function block

```
dt=10^-7; %time step

R=10; %Ohm

C=1*10^-6; %milifarad

L=4*10^-6; %millihenry

out=sim('sim1.slx');
```

Finally, the impulse response of the system was simulated and the following results were obtained

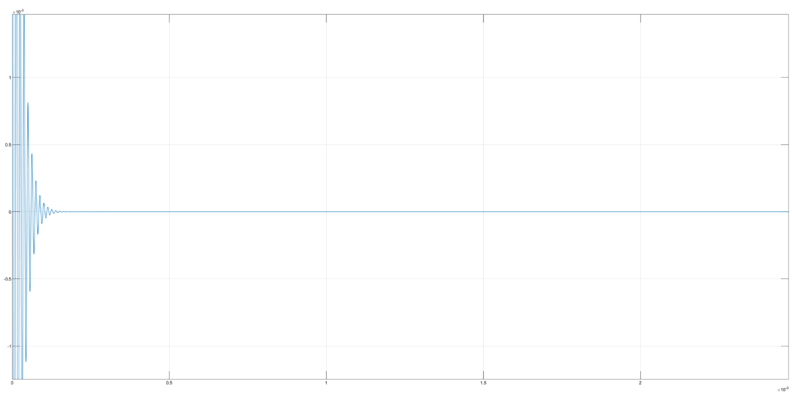

fig 4: Impulse response of the system.

Conclusions:

The transmittance $T(\omega)$ of a system, in the context of any linear system, is defined as the ratio of the output voltage $v_{out}(t)$ to the input voltage $v_{in}(t)$ at a given frequency ω.

In terms of circuit analysis, this transmittance ratio helps describe how the system responds to different frequencies of the input signal. By examining this ratio, we can understand how the circuit attenuates or amplifies certain frequencies relative to the input.

Task 1.2

This task demonstrates how to plot a bode plot of the RLC circuit. We will utilize Matlab's inbuilt function as well as our self-built procedure. We will compare the results from both and estimate the damping of the system using two graphical methods.

```
omega = 2 * pi * logspace(-3,3,173);

num=[1 0];

den=[(C*R) 1 (R/L)];

SYS=tf(num,den);

bode(SYS), grid
```

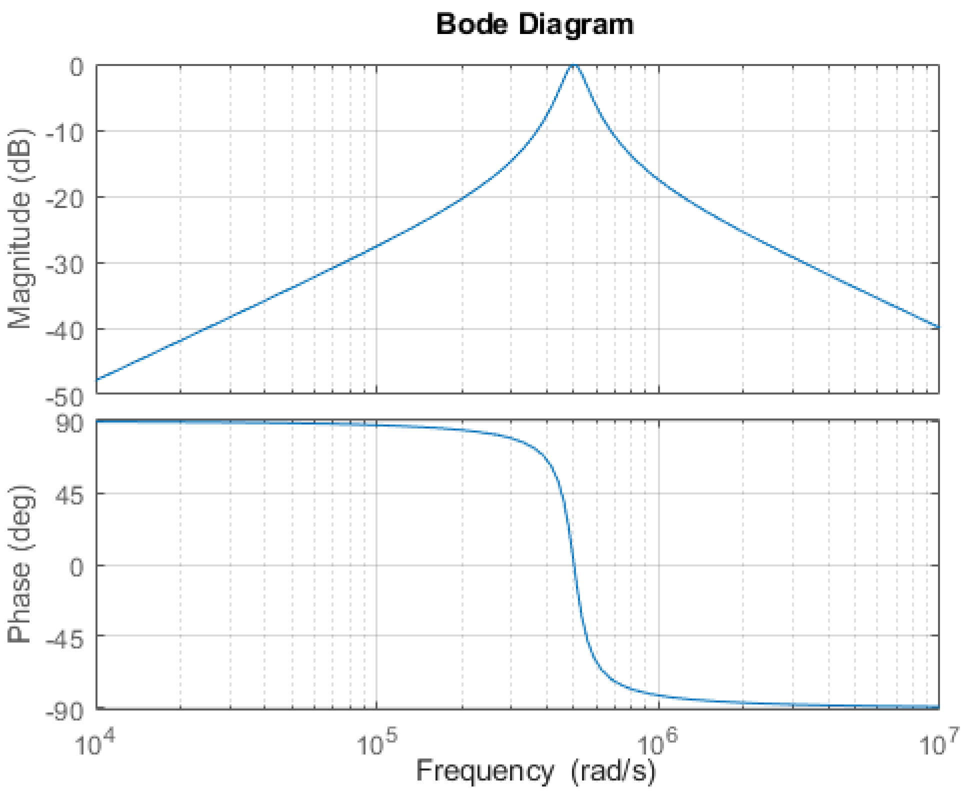

fig 5: Bode plot of the original system obtained using Matlab's function

```
x1=out.x1;
y1=out.x2;
X1=fft(x1);
Y1=fft(y1);
H=Y1./X1;
N=length(H);
f = 2*pi*1/dt*(0:(N-1))/N;
mag_sys=abs(H);
phase_deg=rad2deg(angle(H));
figure()
subplot(2,1,1);
semilogx(f,20*log10(mag_sys)), grid on
xlim([10^4 10^7])
xlabel('Frequency [rad/s]')
ylabel('Magnitude [dB]')
subplot(2,1,2);
semilogx(f,phase_deg), grid on
ylim([-100 100])
xlim([10^4 10^7])
xlabel('Frequency [rad/s]')
ylabel('Phase [deg]')
%calculating damping using 2 methods
%logarithmic damping decrement:
x0 = 5.424*exp(-3);
x1 = 2.885*exp(-3);
epsilon1 = log(x0/x1)/(2*pi)
%Half powe method
omega2 = 552354;
omega1 = 452640;
omegar = 500015;
epsilon11 = (omega2 - omega1)/(2*omegar)
```

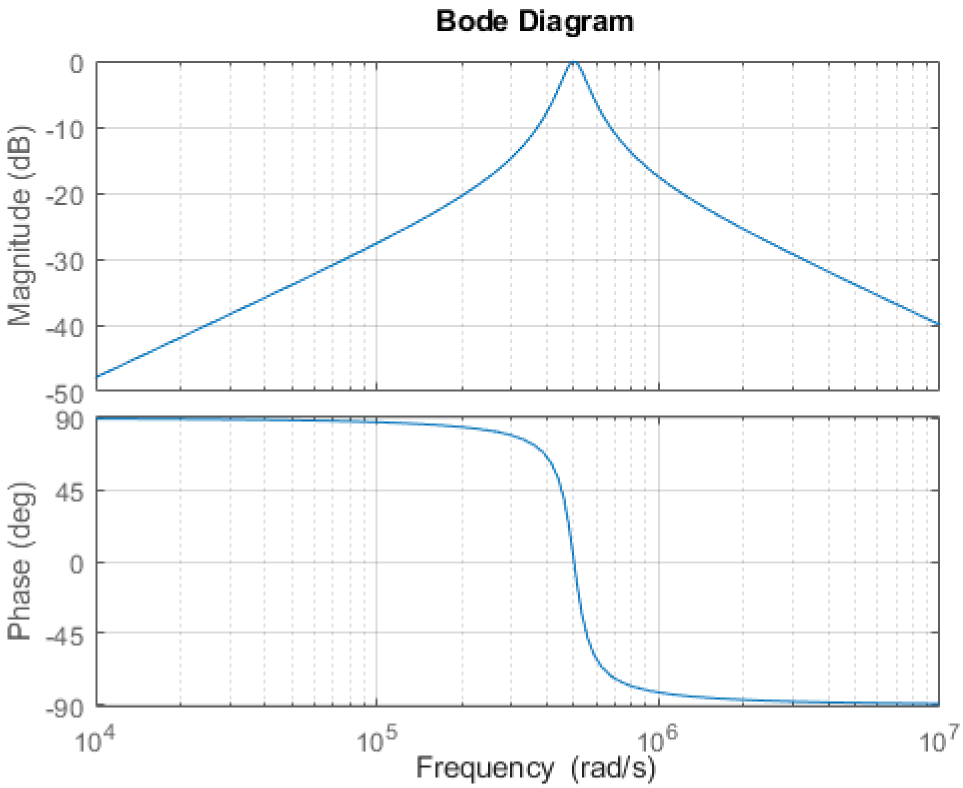

fig 6: Bode plot manually calculated

The damping ratio of the system was calculated using the logarithmic damping decrement, and half power method, the results are as follows:

logarithmic damping decrement :

$$\xi = \frac{\ln(\frac{x_0}{x_1})}{2\pi}$$

Half power method:

$$\xi = \frac{\omega_2 - \omega_1}{2\omega_r}$$

fig 7: formulas for calculating damping using the two methods

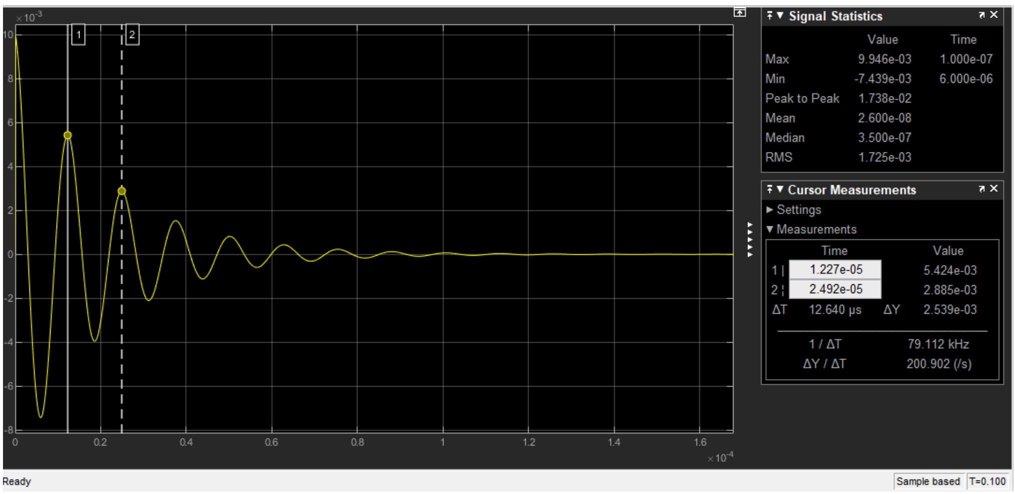

fig 8: impulse response of the system, with the x0 and x1 points marked.

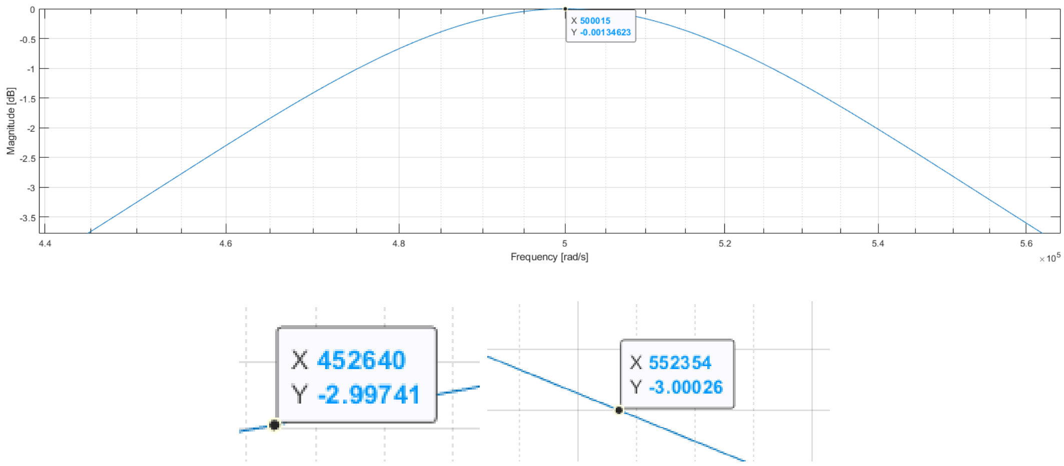

fig 9: Determining the half power points of interest omega 1, omega 2, and omega r

```
%logarithmic damping decrement:
x0 = 5.424*exp(-3);
x1 = 2.885*exp(-3);

epsilon1 = log(x0/x1)/(2*pi)
epsilon1 = 0.1005
```

```
%Half powe method
omega2 = 552354;
omega1 = 452640;
omegar = 500015;
epsilon11=(omega2 - omega1)/(2*omegar)
epsilon11 = 0.0997
```

Conclusions: The plots obtained directly from the Matlab function and from the Simulink are identical, after performing proper scaling and unit conversions. The system in question is damped which we can observe on the plots above since the amplitude is decreasing over time.

When it comes to the estimation of the damping of the system, both graphical methods performed similarly. The slight discrepancy is due to inherent inaccuracies due to the manual selection of the points of interest, but they still present a fairly accurate representation of the true damping of the system.

Task 1.3

In this task we will be performing the same procedure as in task 1.2, but for two different resistances. We aim to determine when our system becomes critically damped, and what's the influence of the resistance on the filter bandwidth and attenuation.

The code is exactly the same (with the exception of the values of R which are now 5 and 100 Ohm's) therefore the code will be omitted.

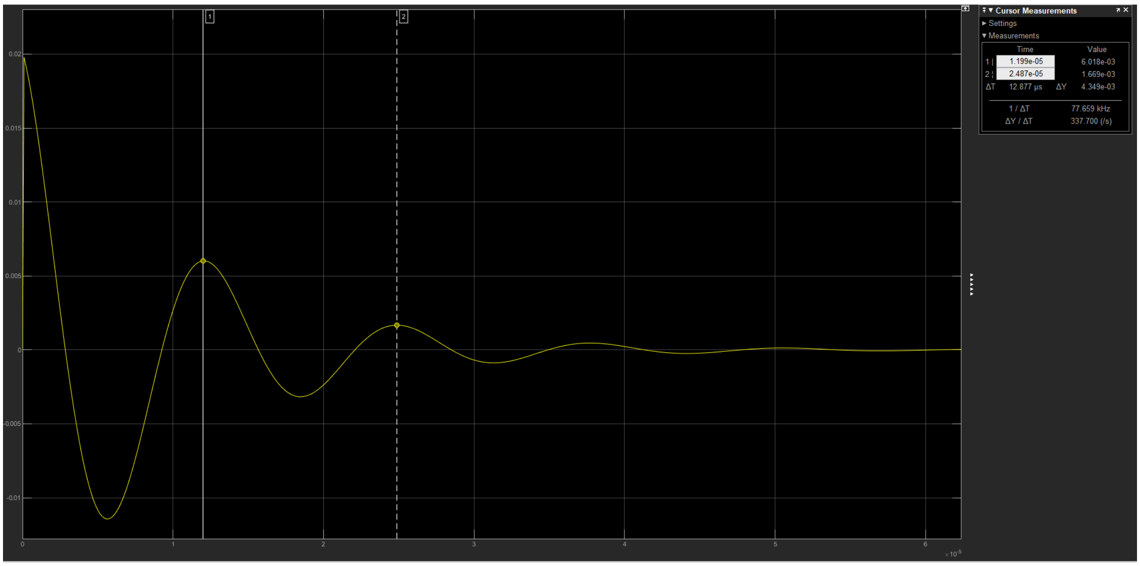

fig 10: Impulse response of the system with R=5Ohm.

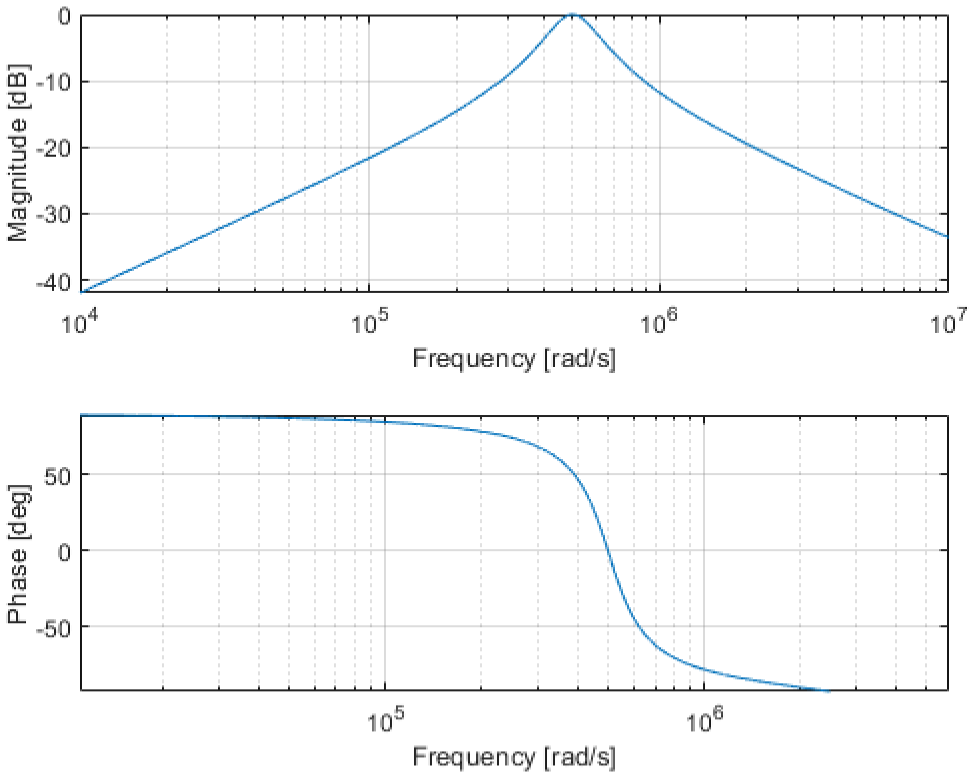

fig 10: Bode plot of the system with R=5Ohm.

```
%logarithmic damping decrement:

x02 = 6.018*exp(-3);
x12 = 1.669*exp(-3);

epsilon2 = log(x02/x12)/(2*pi)
```

```
epsilon2 = 0.2041
```

fig 11: Damping of the system with R=5Ohm.

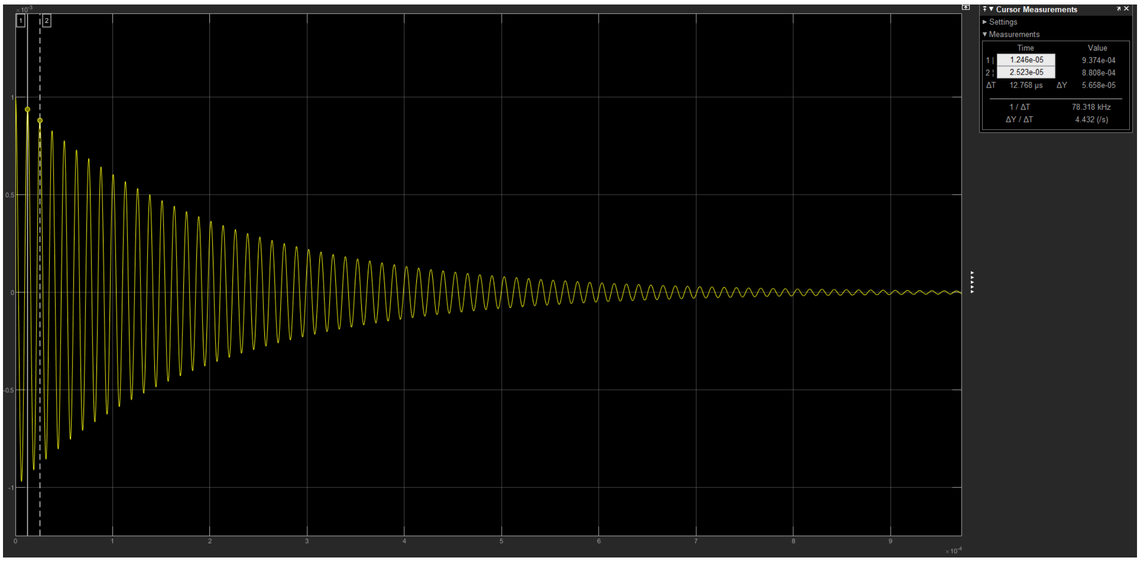

fig 12: Impulse response of the system with R=100 Ohm.

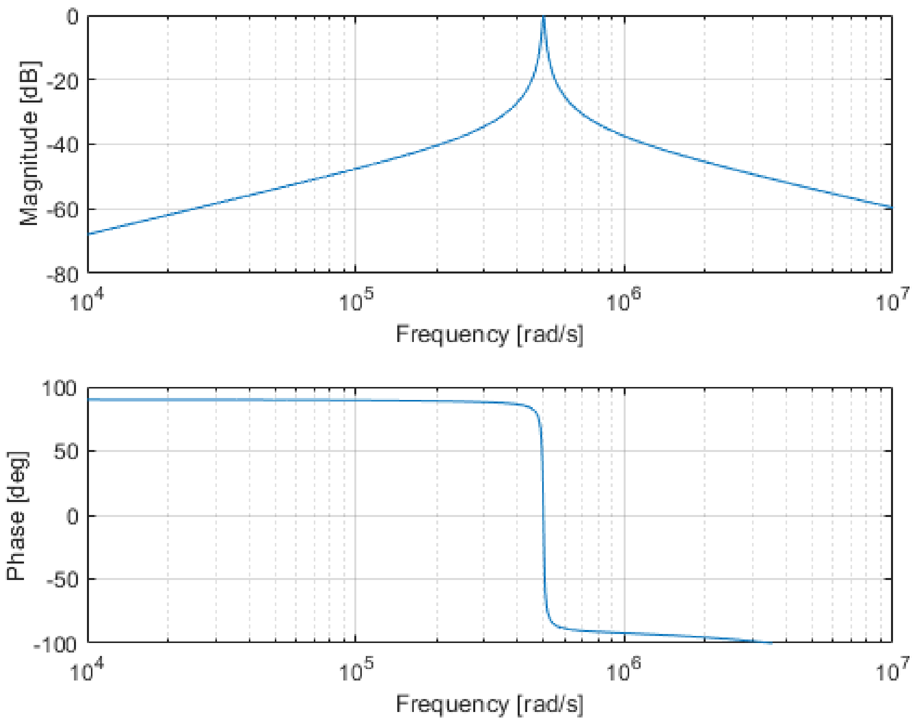

fig 13: Bode plot of the system with R=100 Ohm.

```
%logarithmic damping decrement:

x03 = 9.374*exp(-3);
x13 = 8.808*exp(-3);

epsilon3 = log(x03/x13)/(2*pi)
```

```
epsilon3 = 0.0099
```

fig 14: Damping of the system with R=100 Ohm.

Conclusions:

The damping of the system with resistance R=5 Ohm is 0.2, while the system with R=100 Ohm is 0.01.

We can draw the following conclusions regarding the influence of the resistance on the system response:

To analyze the effects of resistance R on the filter bandwidth, attenuation, and critical damping in an RLC circuit with parallel inductor L and capacitor C, we need to consider the damping ratio of the circuit. The damping ratio determines whether the system is underdamped, critically damped, or overdamped.

- Bandwidth: The bandwidth of the RLC circuit, which determines the range of frequencies over which the circuit effectively operates, is influenced by the resistance R. Higher values of R tend to increase the damping effect, narrowing the bandwidth of the circuit.
- Attenuation: Attenuation refers to the decrease in amplitude of the output signal relative to the input signal. Higher values of R generally lead to increased attenuation, particularly at higher frequencies, due to increased damping.

When it comes to the critically damped system, We can observe it in the case when resistance is increased to 100 Ohm’s.

Generally, critical damping occurs when the damping ratio equals 1. This condition ensures the fastest response to a step input without oscillation, but an alternative definition can be drowned. A system is critically damped when we can observe an exponential decay without oscillations.

Task 2

This task aims at determining allowing us to understand how different excitation waves could be similar in terms of determining the properties of the output signal.

Task 2.1

The task involves simulating the response of a system to different excitation frequencies and analyzing the system's amplitude and phase response. Conclusions from such an experiment can provide insightful information about the dynamic characteristics of the system and validate its behavior against theoretical predictions, such as those presented in Bode plots.

```
%% Task 2
%Defining the variables
dt=10^-7; %time step
R=10; %Ohm
C=1*10^-6; %milifarad
L=4*10^-6; %millihenry
%Ex. 1.1
f1 = 70 * 6283.18531;
f2 = 80 * 6283.18531;
f3 = 90 * 6283.18531;
out=sim('sim2.slx');
% First Frequency
figure()
plot(out.tout,out.x1)
hold on
plot(out.tout,out.x2)
xlabel('Time [s]')
ylabel('Amplitude [-]')
ylim([-1.5 1.5])
xlim([0 (0.4*10^-3)])
hold off
legend ('Input Signal','Output Signal')
```

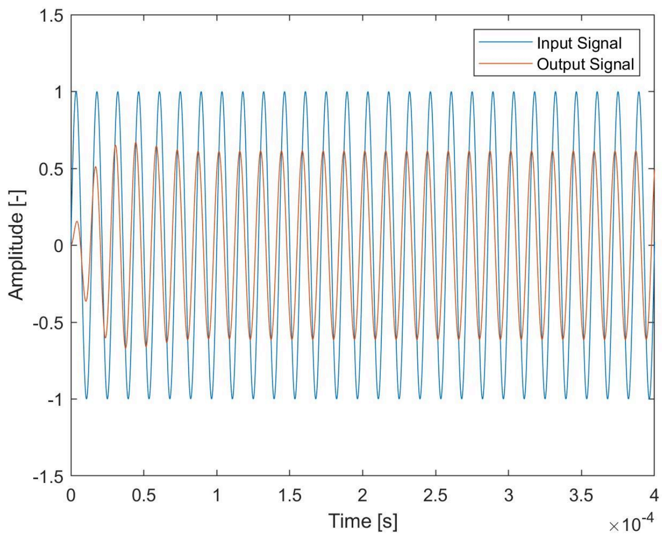

fig 15: The plot of the input and output signals of wave with f = 70kHz.

```
%Bode Calculation and plotting
x1=out.x1;
y1=out.x2;
X1=fft(x1);
Y1=fft(y1);
H=Y1./X1;
N=length(H);
f = 2*pi*1/dt*(0:(N-1))/N;
mag_sys=abs(H);
phase_deg=rad2deg(angle(H));
figure()
subplot(2,1,1);
semilogx(f,20*log10(mag_sys)), grid on
xlim([10^4 10^7])
xlabel('Frequency [rad/s]')
ylabel('Magnitude [dB]')
subplot(2,1,2);
semilogx(f,phase_deg), grid on
ylim([-100 100])
xlim([10^4 10^7])
xlabel('Frequency [rad/s]')
ylabel('Phase [deg]')

T = 0.0001587 - 0.0001444;
amp = 0.61;
dtime = 0.0001465-0.0001444;
dphi = (360*dtime)/T
```

```
dphi =

    52.8671
```

fig 16: First Phase calculation result.

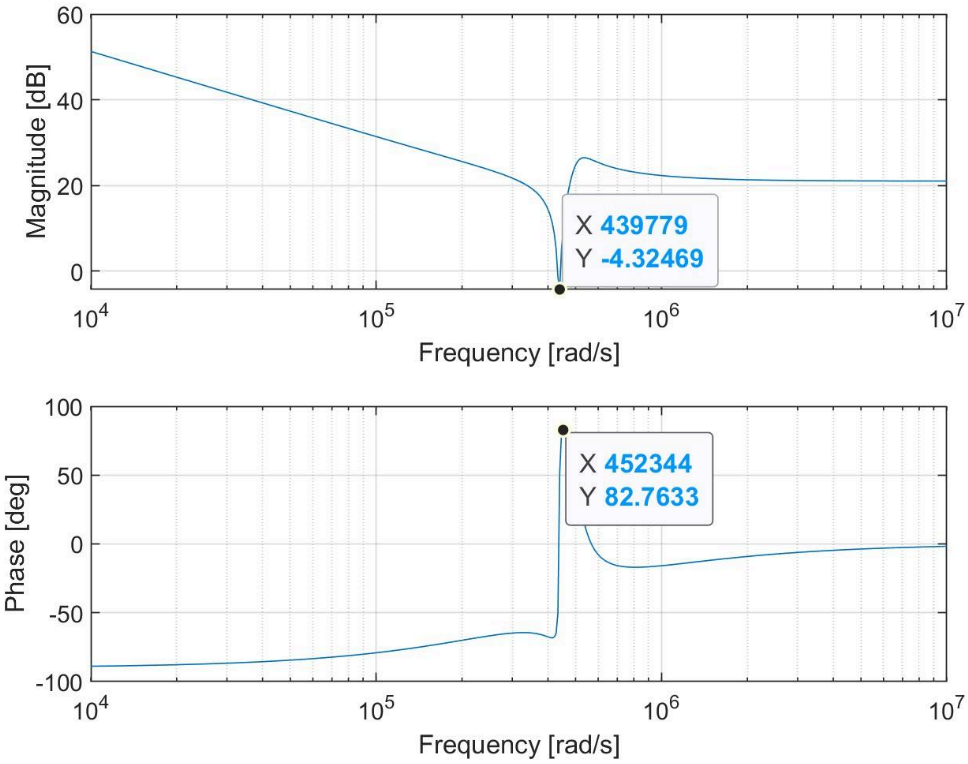

fig 17: Bode plots of the signals of wave with $f = 70$kHz.

```
% Second frequency
out=sim('sim2_80.slx');
figure()
plot(out.tout,out.x1_80)
hold on
plot(out.tout,out.x2_80)
xlabel('Time [s]')
ylabel('Amplitude [-]')
ylim([-1.5 1.5])
xlim([0 (0.4*10^-3)])
hold off
legend ('Input Signal','Output Signal')
```

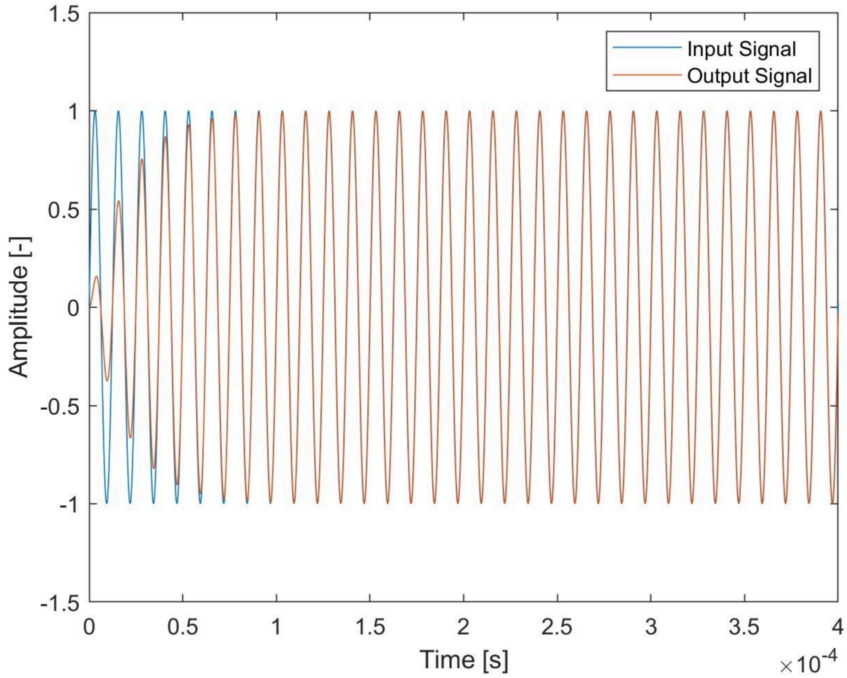

fig 18: The plot of the input and output signals of wave with f = 80kHz.

```
%Bode Calculation and plotting
x1=out.x1_80;
y1=out.x2_80;
X1=fft(x1);
Y1=fft(y1);
H=Y1./X1;
N=length(H);
f = 2*pi*1/dt*(0:(N-1))/N;
mag_sys=abs(H);
phase_deg=rad2deg(angle(H));
figure()
subplot(2,1,1);
semilogx(f,20*log10(mag_sys)), grid on
xlim([10^4 10^7])
xlabel('Frequency [rad/s]')
ylabel('Magnitude [dB]')
subplot(2,1,2);
semilogx(f,phase_deg), grid on
```

```
ylim([-200 0])
xlim([10^4 10^7])
xlabel('Frequency [rad/s]')
ylabel('Phase [deg]')
amp80 = 1;
dphi80 = 0
```

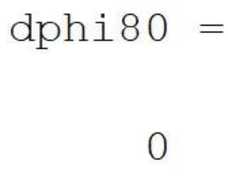

fig 19: The calculated phase result.

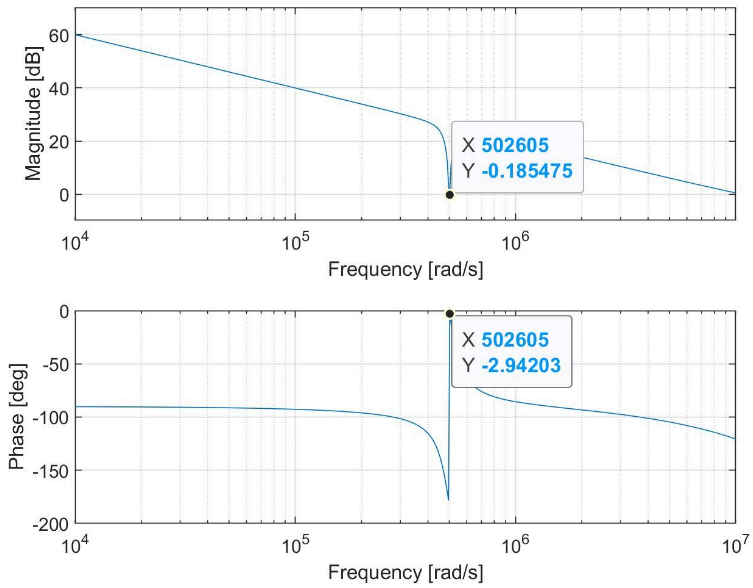

fig 20: Bode plots of the signals of wave with $f = 80$kHz.

```
% Third frequency
out=sim('sim2_90.slx');
figure()
plot(out.tout,out.x1_90)
hold on
plot(out.tout,out.x2_90)
xlabel('Time [s]')
ylabel('Amplitude [-]')
ylim([-1.5 1.5])
xlim([0 (0.4*10^-3)])
hold off
legend  ('Input  Signal','Output
Signal')
```

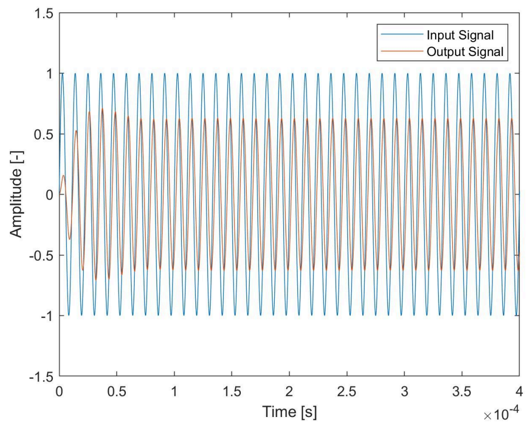

fig 21: The plot of the input and output signals of wave with $f = 90$kHz.

```
%Bode Calculation and plotting
x1=out.x1_90;
y1=out.x2_90;
X1=fft(x1);
Y1=fft(y1);
H=Y1./X1;
N=length(H);
f = 2*pi*1/dt*(0:(N-1))/N;
```

```
mag_sys=abs(H);
phase_deg=rad2deg(angle(H));
figure()
subplot(2,1,1);
semilogx(f,20*log10(mag_sys)), grid on
xlim([10^4 10^7])
xlabel('Frequency [rad/s]')
ylabel('Magnitude [dB]')
subplot(2,1,2);
semilogx(f,phase_deg), grid on
ylim([-200 200])
xlim([10^4 10^7])
xlabel('Frequency [rad/s]')
ylabel('Phase [deg]')
T90 = 0.0001139 - 0.0001028;
amp90 = 0.61;
dtime90 = 0.0001044 - 0.0001028;
dphi90 = (360*dtime90)/T90
```

```
dphi90 =

    51.8919
```

fig 22: The calculated phase result.

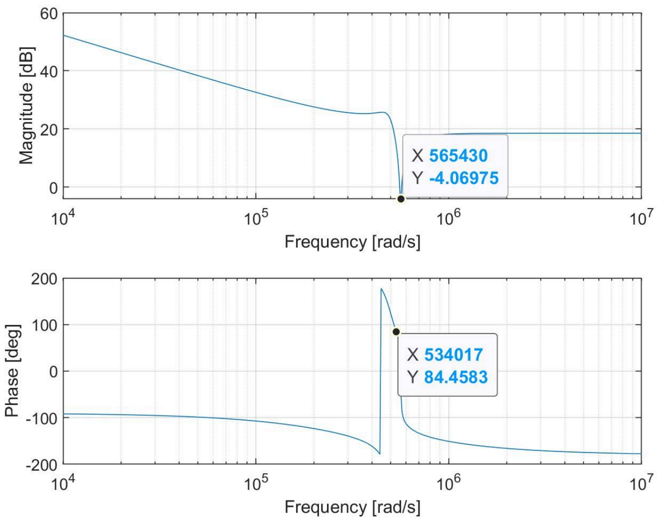

fig 23: Bode plots of the signals of wave with f = 90kHz.

Conclusions:

The phase shift between the input and output signals is calculated using the provided relationship, $d\phi = 360\, x \frac{dt}{T}$, where dt is the measured time shift and T is the period of the wave. By directly measuring the time shift in the simulations and converting this to phase shift, the experiment offers a practical demonstration of phase relationship analysis in frequency response studies. Comparing these calculated phase shifts with those predicted by the Bode plot provides a critical check on the phase accuracy of the system model.

At each excitation frequency, the observed amplitudes of the output signals can be assessed against expected values from the Bode plot. Consistency between the simulated amplitudes and those predicted by the Bode plot confirms the accuracy of the system model and its linear response characteristics over the tested frequency range. Any discrepancies might indicate nonlinear behaviors, model inaccuracies, or the influence of unmodeled dynamics.

Task 2.2

The objective of the task was to compare the response of a system to a specific input signal (a 70 kHz sine wave) obtained through two different methodologies: direct simulation and convolution with the system's impulse response. Analyzing the results of these methods provides insights into the system's behavior and validates the simulation techniques.

```
%Ex. 2.2
% Parameters
frequency = 70e3; % 70 kHz
amplitude = 1; % Amplitude of the sine wave
samplingFrequency = 200e3; % 200 kHz
duration = 0.001; % Duration in seconds (1 ms)
% Time vector
t = 0:1/samplingFrequency:duration;
% Sine wave
sineWave = amplitude * sin(2 * pi * frequency * out.tout);
% Plot the sine wave
figure()
plot(out.tout, sineWave);
ylim([-1.5 1.5])
xlim([0 (0.4*10^-3)])
xlabel('Time (seconds)');
ylabel('Amplitude');
title('70 kHz Sine Wave');
%Calling the Step signal
out=sim('sim1.slx');
con = conv(out.x2,sineWave);
figure()
subplot(2,1,1)
plot(con)
xlim([0 2000])
xlabel('Time [s]')
ylabel('Amplitude [-]')
title('Convoluted Signal')
f1 = 70 * 6283.18531;
out=sim('sim2.slx');
subplot(2,1,2)
plot(out.tout,out.x2)
xlabel('Time [s]')
ylabel('Amplitude [-]')
title('Signal Impulse Response')
ylim([-1.5 1.5])
xlim([0 (0.4*10^-3)])
hold off
```

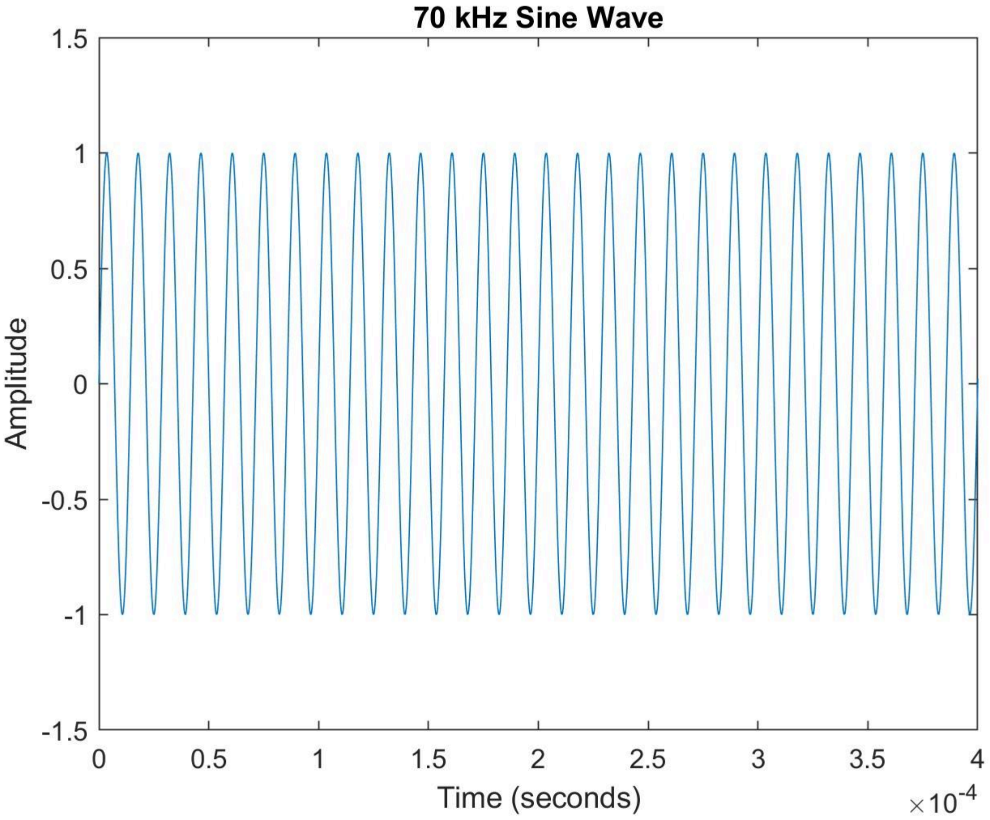

fig 24: The MATLAB generated Sine wave.

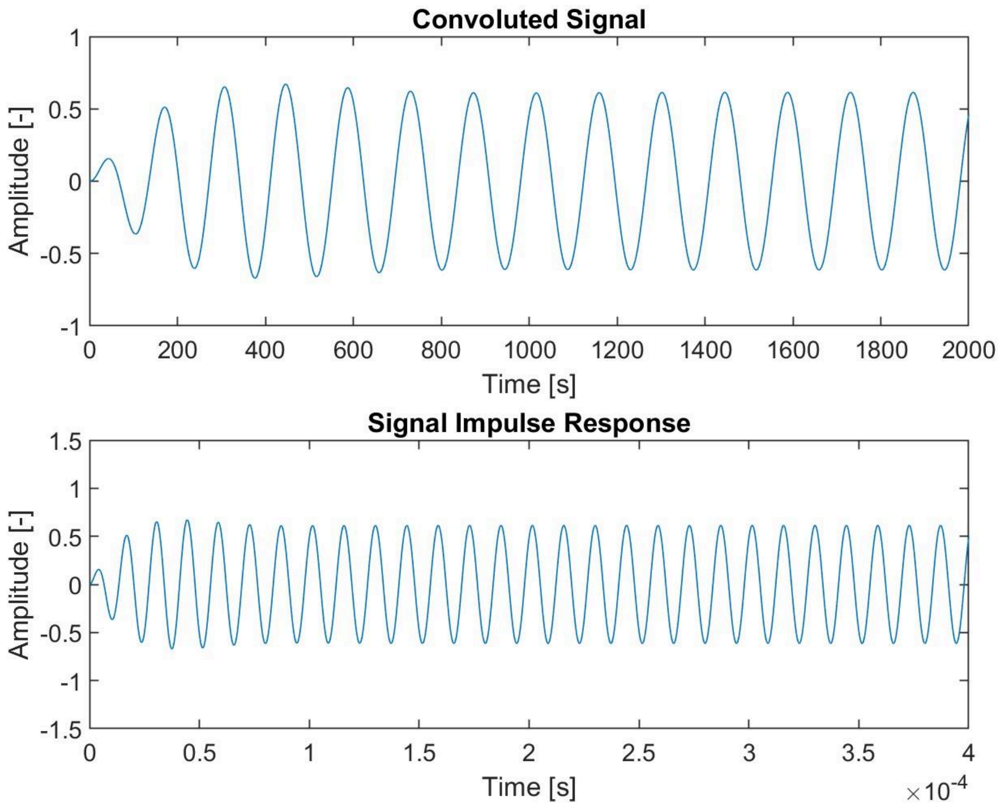

fig 25: Comparison of the Convoluted signal and the original impulse response..

Conclusions:

The experiment validates the use of convolution with an impulse response as an effective method for simulating system responses to input signals. By convolving the sine wave input with the impulse response of the system, the resultant output signal matched the output obtained directly from the system simulation under identical conditions. Comparing the amplitude and phase shift of the system's output signal obtained via convolution and direct simulation at 70 kHz provides a test of consistency between these methods. If the methods are accurately implemented, the results should be nearly identical, confirming the linearity and time-invariance of the system as modeled. Any significant discrepancies might indicate errors in the simulation setup, impulse response calculation, or the convolution process.