Homework 5
CPE381

Canvas: `hw05`

Due: 2nd November 2024, 11:59 PM
100 points

You are allowed to use a generative model-based AI tool for your assignment. However, you must submit an accompanied reflection report on how you use the AI tool, what was the query to the tool, and how it improved your understanding of the subject. You must also add your thoughts on how you would tackle the assignment if there was no such tool available. Failure to provide a reflection report for every single assignment where an AI tool was used may result in a penalty and subsequent actions will be taken in line with plagiarism policy.

Submission instruction:

Upload a .pdf on Canvas with the format {firstname.lastname}_cpe381_hw05.pdf. If there is a programming assignment, then you should include your source code along with your PDF files in a zip file firstname.lastname}_cpe381_hw05.zip. If a plot is being asked, your PDF file must also contain plots generated by your MATLAB code. Your submission must contain your name, and UAH Charger ID or the UAH email address. Please number your pages as well.

---

1 Fourier Transform (20 points)

The rectangular pulse is defined by

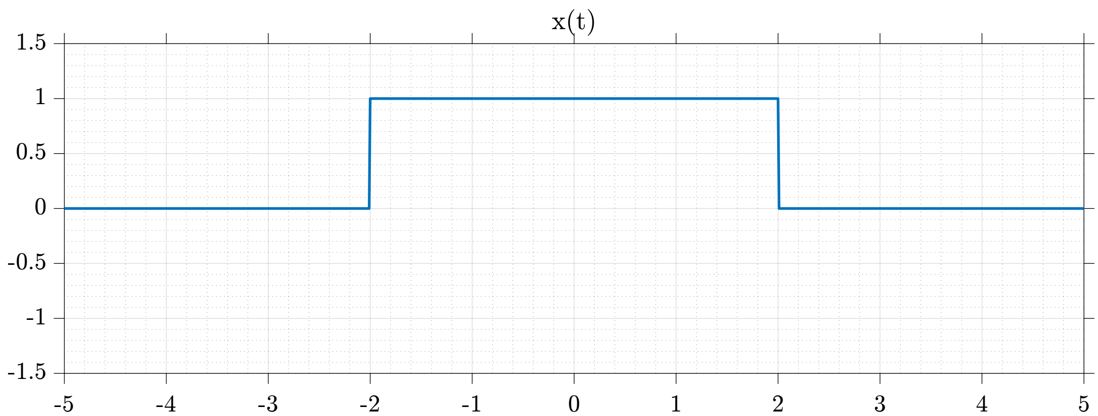
$$x(t) = \begin{cases} 1, & |t| \leq |a| \\ 0, & |t| > |a| \end{cases}$$

1. Write a MATLAB program to plot the signal $x(t)$. **(5 points)**.
2. Find the Fourier transform $X(\Omega)$ analytically (i.e. mathematically) of $x(t)$. **(10 points)**.
3. Write a MATLAB program to plot the magnitude spectrum $|X(\Omega)|$ vs Ω. **(5 points)**.

Solution

1.
```
a = 2.0; % You can choose any other value
t = -5:0.01:5;
x = zeros(size(t));  % Initialize 'x' with zeros
```

```
% Apply the piecewise function
x(abs(t) <= abs(a)) = 1;
f= figure;
f.Position = [554, 456, 1300, 420];
plot(t,x, 'LineWidth',2);
grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';
title('x(t)', 'Interpreter','latex');
ylim([-1.5, 1.5]);
set(gca, 'FontSize', 18);
```



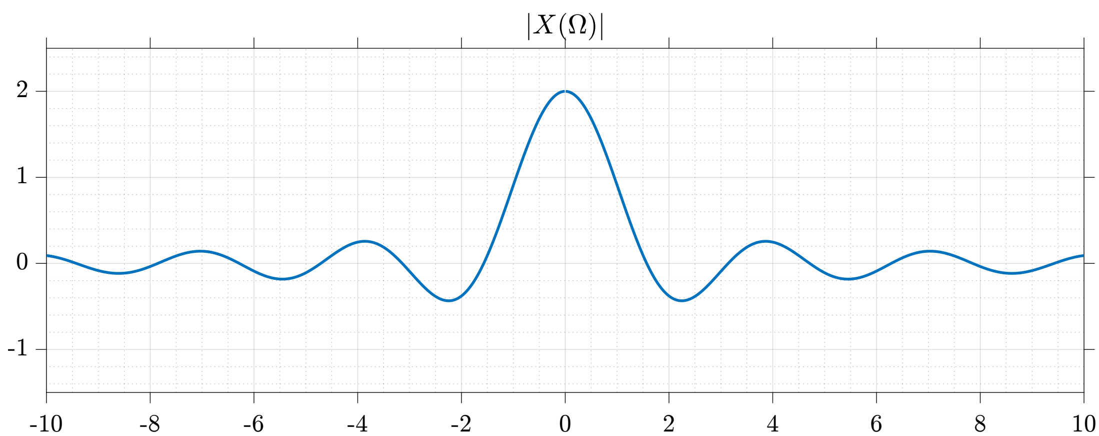
2.

$$
\begin{aligned}
X(\Omega) &= \int_{-\infty}^{\infty} x(t)e^{-j\Omega t}dt \\
&= \int_{-a}^{0} 1 \cdot e^{-j\Omega t}dt + \int_{0}^{a} e^{-j\Omega t}dt \\
&= \frac{e^{-j\Omega t}}{-j\Omega}|_{-a}^{0} + \frac{e^{-j\Omega t}}{-j\Omega}|_{0}^{a} \\
&= \frac{1 - e^{j\Omega a}}{-j\Omega} + \frac{e^{-j\Omega a} - 1}{-j\Omega} \\
&= \frac{- e^{j\Omega a} + e^{-j\Omega a}}{-j\Omega} = \frac{e^{j\Omega a} - e^{-j\Omega a}}{j\Omega} \\
&= \frac{\sin(\Omega a)}{\Omega} = a\text{sinc}(\Omega a)
\end{aligned}
$$

3.

```
w = -10:0.01:10;
X = sin(w.*a)./w;
```

```
f= figure;
f.Position = [554, 456, 1300, 420];
plot(w,X, 'LineWidth',2);
grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';
title('$|X(\Omega)|$', 'Interpreter','latex');
ylim([-1.5, 2.5]);
set(gca, 'FontSize', 18);
exportgraphics(f, '../figures/CPE381_FA24_HW5_Q1_mag_spectrum.pdf');
```



Note: Some students might plot abs(X) to give the correct magnitude spectrum when X is a complex number. However, when X is a real number, then the abs function will make the negative part positive.

2 Duality (10 points)

Use the duality property to find the Fourier transform of the sinc signal:

$$x(t) = 20\frac{\sin(0.5t)}{0.5t} = 20\text{sinc}(0.5t), \quad -\infty < t < \infty.$$

Solution

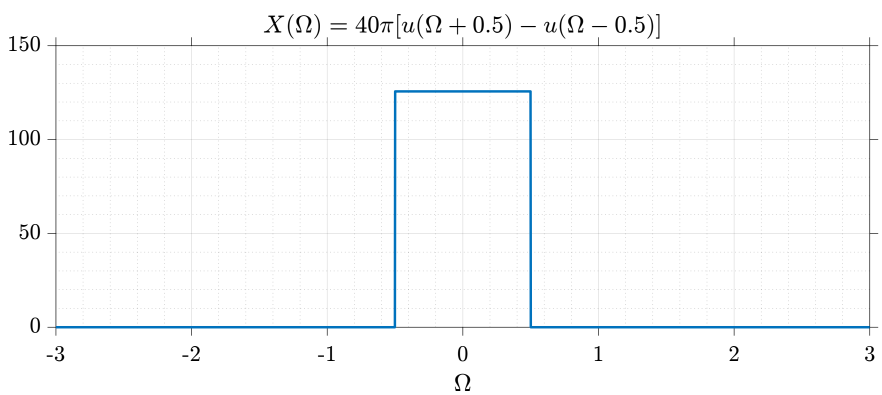
Its Fourier transform is calculated to be

$$
\begin{aligned}
P(\Omega) &= 20\int_{0.5}^{0} .5e^{-j\Omega t}dt \\
&= 20\frac{e^{-j\Omega t}}{-j\Omega t}\Big|_{-0.5}^{0.5} \\
&= 20\frac{e^{-j\Omega t}-e^{j\Omega t}}{-j\Omega 0.5} \\
&= 20\frac{e^{j\Omega t}-e^{-j\Omega t}}{j\Omega 0.5} \\
P(\Omega) &= \frac{20\sin(0.5\Omega)}{0.5\Omega}
\end{aligned}
$$

Then, by duality,

$$
\begin{aligned}
x(t) = P(t) &= \frac{20\sin(0.5t)}{0.5t} \\
\Rightarrow X(\Omega) = 2\pi p(-\Omega) &= 2\pi p(\Omega) = 2\pi \times p = 40\pi[u(\Omega+0.5) - u(\Omega-0.5)]
\end{aligned}
$$

Because $p(\ )$ is an even function.
Note that this signal is bandlimited, as we can see from its Fourier transform that its Ω_{max} is 0.5 rad/unit.



3 Filter Design (35 points)

Consider a sine wave as $x(t) = \sin(2\pi t)$.

1. What's the fundamental period of $x(t)$? **(2 points)**.
2. Write a MATLAB function to plot $x(t)$. Using the sampling frequency `fs` as 100 Hz. **(2 points)**.
3. Compute its Fourier Transform numerically using MATLAB using `fft` function. Do not use symbolic computation. Check the lecture slides, and course GitHub repo for an example. Plot the magnitude spectrum as a function frequency. **(10 points)**

4. Create a sine wave $\eta(t)$ of a frequency ten times that of $x(t)$ with an amplitude 10% of the amplitude of $x(t)$. Add $\eta(t)$ to the original sine wave so that $y(t) = \eta(t) + x(t)$. Plot $y(t)$. **(2 points).**

5. Compute the Fourier transform of $y(t)$ using `fft` function and plot its Magnitude spectrum $|Y(\Omega)|$ as a function of Ω. Explain in two sentences how the magnitude spectrum of $y(t)$ differs from that of $x(t)$. **(4 points)**.

6. Now, we will design a filter to remove high-frequency noise from $y(t)$. We will design a low-pass filter using `butter` function (that implements Butterworth filter). Choose the filter order of 4 and a cutoff frequency of $5Hz$.

   An example syntax for `butter` is:

   ```
   [b, a] = butter(order, cutoff_freq / (fs/2), 'low');
   ```

   `fs` is the sampling frequency.

   Write down filter coefficients b, and a. **(5 points)**.

7. Now, we will use the filter to clean the noisy sine wave $y(t)$. The syntax is as follows

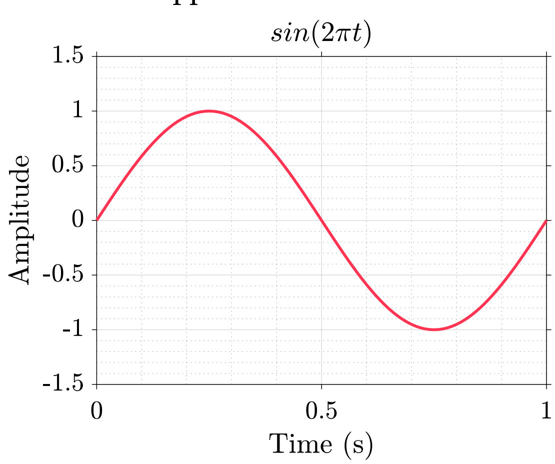
   ```
   filtered_sine = filter(b, a, noisy_sine);
   ```

   Plot the filtered sine wave. **(5 points)**.

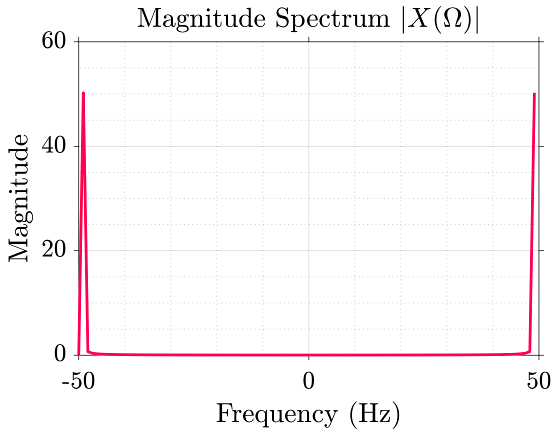
8. Report the mean squared error between the original signal $x(t)$ and the filtered sine wave. **(5 points)**.

Solution

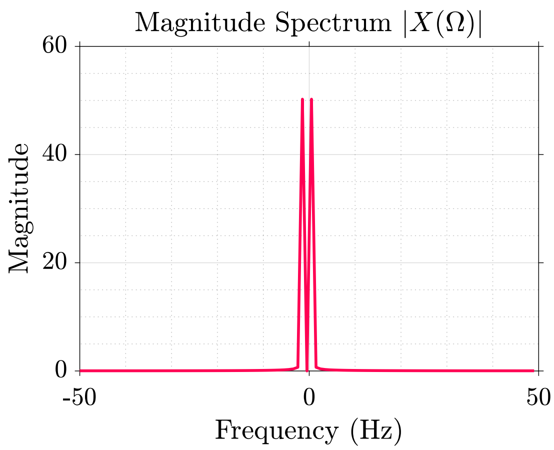
1. Fundamental period is $T_0 = 2\pi/2\pi = 1$
2. See code-snippet.



3. See code-snippet.



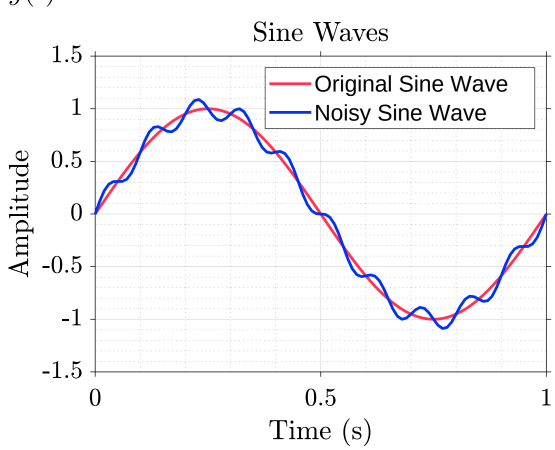
Note: some students might have used `fftshift(fft(x))`. In such case, the plot would look like as follows:

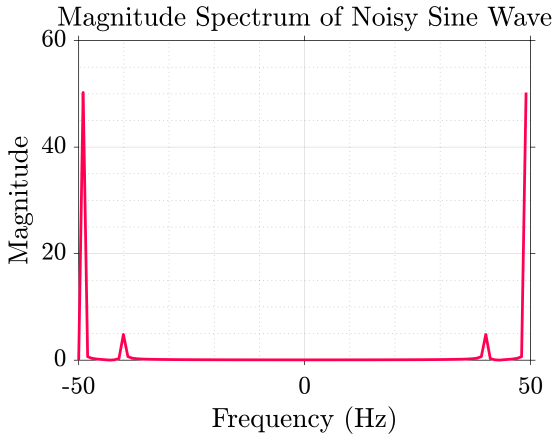


4. See code-snippet.

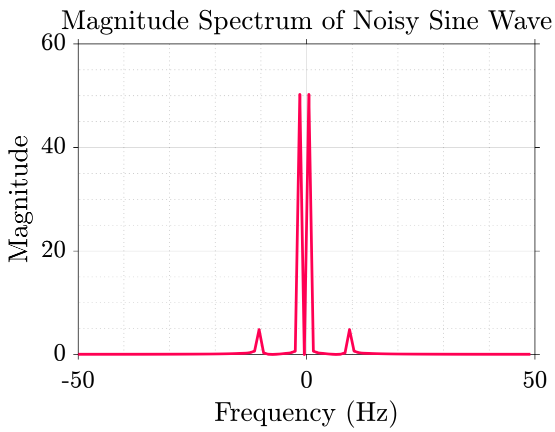
   Note: If you took 20*log10 of Fourier transform magnitude. In that case, the shape will look different, but that's also a correct answer.

   $y(t)$



5. 

Note: some students might have used `fftshift(fft(y))`. In such case, the plot would look like as follows:

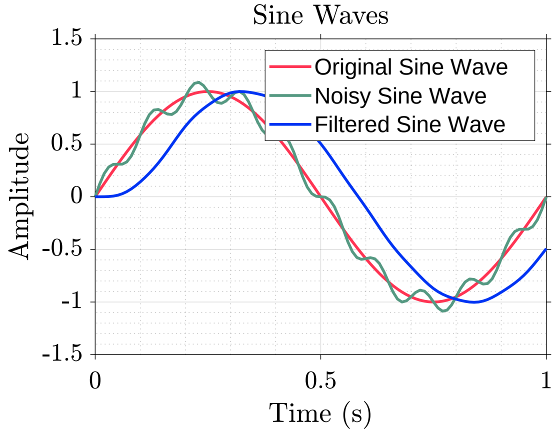


With an added high-frequency noise, we will see another secondary peak in the magnitude spectrum

6. See code-snippet.

   Filter coefficients:

   `b = [0.0004166 0.0016664 0.0024996 0.0016664 0.0004166], a = [1 -3.1806 3.8612 -2.1122 0.43827]`

7. 

8. Mean Squared Error: 0.11736

Code-snippet:

```
%% Define the time period
T = 1;

% Define the sampling frequency
fs = 100; % Hz
% Generate the time array
t = 0:1/fs:T;
% Generate the sine wave
x = sin(2*pi/T*t);
% Plot the sine wave
fig = figure;
plot(t, x,'LineWidth',2,'color', '#ff3453')
grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';
ylim([-1.5, 1.5]);

xlabel('Time (s)', 'Interpreter','latex');
ylabel('Amplitude', 'Interpreter','latex');
title('$sin(2\pi t)$', 'Interpreter','latex');
set(gca, 'FontSize', 18);
exportgraphics(fig, ...
    '../figures/CPE381_FA24_HW5_Q3_Sin.pdf');

%%
```

```matlab
%X = fft(x);
X = fftshift(fft(x));
% Plot the magnitude spectrum
N = length(X); % number of points in fft
f = (-N/2:N/2-1)*(fs/N); % frequency vector

fig = figure;
plot(f, abs(X) ,'LineWidth',2, 'color', '#ff0053')
xlabel('Frequency (Hz)', 'Interpreter','latex');

ylabel('Magnitude', 'Interpreter','latex');
title('Magnitude Spectrum $|X(\Omega)|$',  'Interpreter','latex');

grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';
set(gca, 'FontSize', 18);
exportgraphics(fig, ...
    '../figures/CPE381_FA24_HW5_Q3_Sin_Fourier_Magnitude_FFTShift.pdf');


fig = figure;
plot(f, angle(X) ,'LineWidth',2, 'color', '#ff0053')
xlabel('Frequency (Hz)', 'Interpreter','latex');

ylabel('Phase', 'Interpreter','latex');
title('Phase Spectrum $|X(\Omega)|$',  'Interpreter','latex');

grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';
set(gca, 'FontSize', 18);
exportgraphics(fig, ...
    '../figures/CPE381_FA24_HW5_Q3_Sin_Fourier_Phase_FFTShift.pdf');


%%
```

```
% Define the noise frequency and amplitude
noise_freq = 10;  % 10 times the sine wave frequency
noise_amp = 0.1;  % 10% of the sine wave amplitude

% Add high frequency noise to the sine wave
noisy_sine = add_high_freq_noise(t, x, noise_freq, noise_amp);


fig = figure;
% Plot the original and noisy sine waves
plot(t, x,'LineWidth',2,'color', '#ff3453');
hold on;
plot(t, noisy_sine,'LineWidth',2,'color', '#0034f3')
legend('Original Sine Wave', 'Noisy Sine Wave');
grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';
ylim([-1.5, 1.5]);

xlabel('Time (s)', 'Interpreter','latex');
ylabel('Amplitude', 'Interpreter','latex');
title('Sine Waves', 'Interpreter','latex');
set(gca, 'FontSize', 18);
exportgraphics(fig, '../figures/CPE381_FA24_HW5_Q3_NoisySin.pdf');

%%
%Y = fft(noisy_sine);
Y = fftshift(fft(noisy_sine));
% Plot the magnitude spectrum
N = length(Y); % number of points in fft
f = (-N/2:N/2-1)*(fs/N); % frequency vector

fig = figure;
plot(f, abs(Y) ,'LineWidth',2, 'color', '#ff0053')
xlabel('Frequency (Hz)', 'Interpreter','latex');

ylabel('Magnitude', 'Interpreter','latex');
title('Magnitude Spectrum of Noisy Sine Wave',  'Interpreter','latex');

grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
```

```
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';
set(gca, 'FontSize', 18);
exportgraphics(fig, ...
    '../figures/CPE381_FA24_HW5_Q3_NoisySin_Fourier_Magnitude_FFTShift.pdf');


fig = figure;
plot(f, angle(X) ,'LineWidth',2, 'color', '#ff0053')
xlabel('Frequency (Hz)', 'Interpreter','latex');

ylabel('Phase', 'Interpreter','latex');
title('Phase Spectrum of Noisy Sine Wave',  'Interpreter','latex');

grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';
set(gca, 'FontSize', 18);
exportgraphics(fig, ...
    '../figures/CPE381_FA24_HW5_Q3_NoisySin_Fourier_Phase_FFTShift.pdf');


%%

% Define the filter order and cutoff frequency
order = 4;
cutoff_freq = 5;  % Hz

% Design the low-pass filter
[b, a] = butter(order, cutoff_freq / (fs/2), 'low');

% Print the filter coefficients
fprintf('Filter coefficients: b = [%s], a = [%s]\n', num2str(b), num2str(a));

filtered_sine = filter(b, a, noisy_sine);

% Plot the original and filtered sine waves
fig = figure;
% Plot the original and noisy sine waves
plot(t, x,'LineWidth',2,'color', '#ff3453');
```

```
hold on;
plot(t, noisy_sine,'LineWidth',2,'color', '#559982');
plot(t, filtered_sine,'LineWidth',2,'color', '#0034f3')
legend('Original Sine Wave', 'Noisy Sine Wave', 'Filtered Sine Wave');
grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';
ylim([-1.5, 1.5]);

xlabel('Time (s)', 'Interpreter','latex');
ylabel('Amplitude', 'Interpreter','latex');
title('Sine Waves', 'Interpreter','latex');
set(gca, 'FontSize', 18);
exportgraphics(fig, ...
    '../figures/CPE381_FA24_HW5_Q3_FilteredSin.pdf');

% MSE
MSE = mean((x - filtered_sine).^2);
fprintf('Mean Squared Error: %.5f\n', MSE);

function noisy_sine = add_high_freq_noise(t, y, noise_freq, noise_amp)
    % Generate high frequency noise
    noise = noise_amp * sin(2*pi*noise_freq*t);

    % Add noise to the sine wave
    noisy_sine = y + noise;
end
```

4 Filtering a Real Signal (35 points)

1. Download an acceleration data from https://github.com/rahulbhadani/CPE381_FA24/blob/master/Data/accel.csv, and read into MATLAB. The unit of the acceleration is m/s^2. Plot the signal as a function of time. **(5 points)**

2. Perform Fourier transform on the data using `fft` function. **(5 points)**. Note that the signal is not uniformly sampled. Hence, you need to determine the average sampling frequency from the timestamp provided in the data file. **(10 points)**

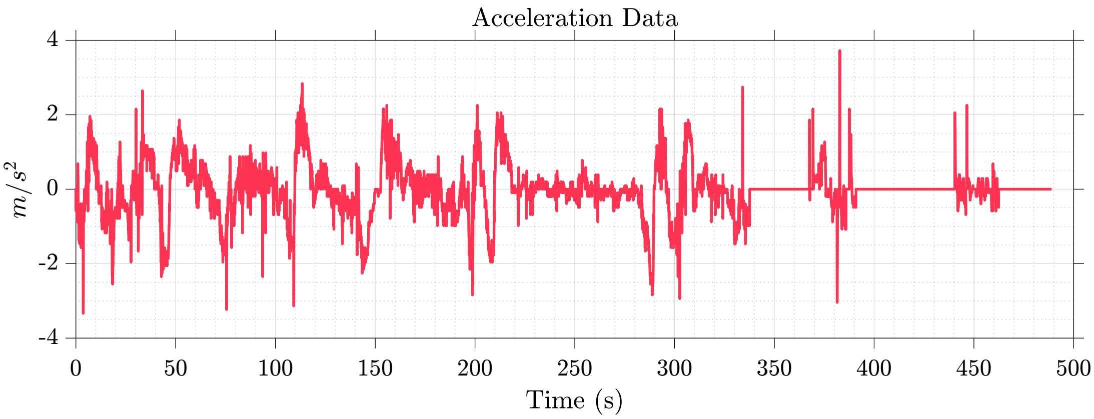
3. Design a low pass filter using `butter` function with cutoff frequency $2Hz$ and choose the order as 4. Write down filter coefficients b, and a. **(5 points)**.

4. Use the filter coefficients with `filter` function to smoothened-out the acceleration data. Plot the acceleration data along with the smoothened-out acceleration data on the same plot, and also create another plot between 130 - 133 seconds to zoom out and visualize the smoothened-out acceleration signal. **(5 points)**

5. Plot the Magnitude and Phase spectrum of the Fourier Transform of the origin acceleration data and smoothened-out acceleration on the same plot for the comparison. (Magnitude and Phase spectrum will be on separate plots) **(5 points)**

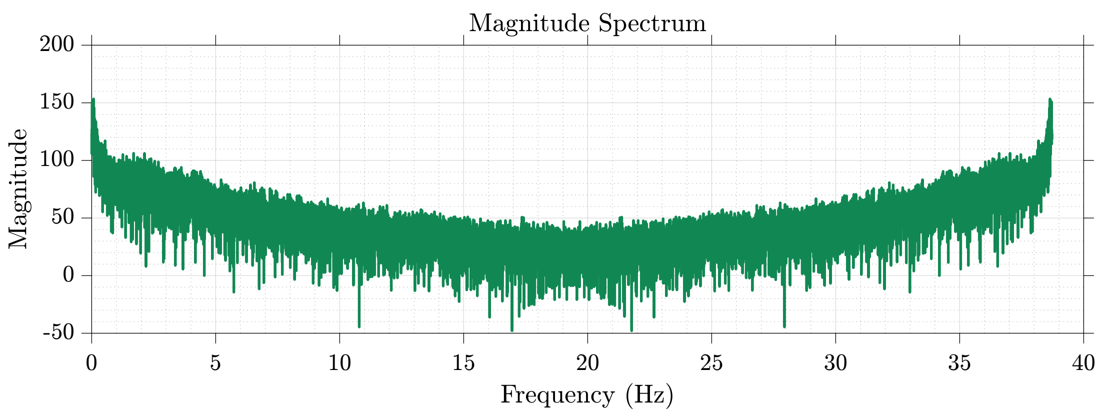
Solution

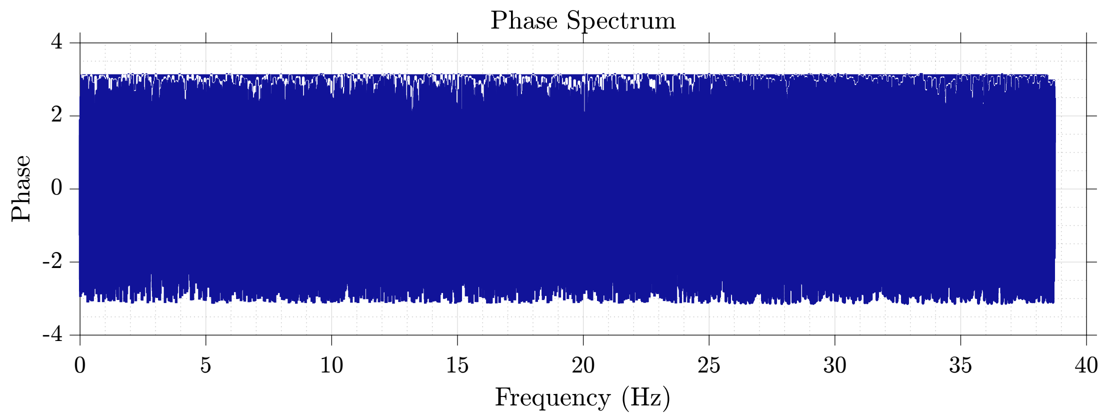
1. See code-snippet.



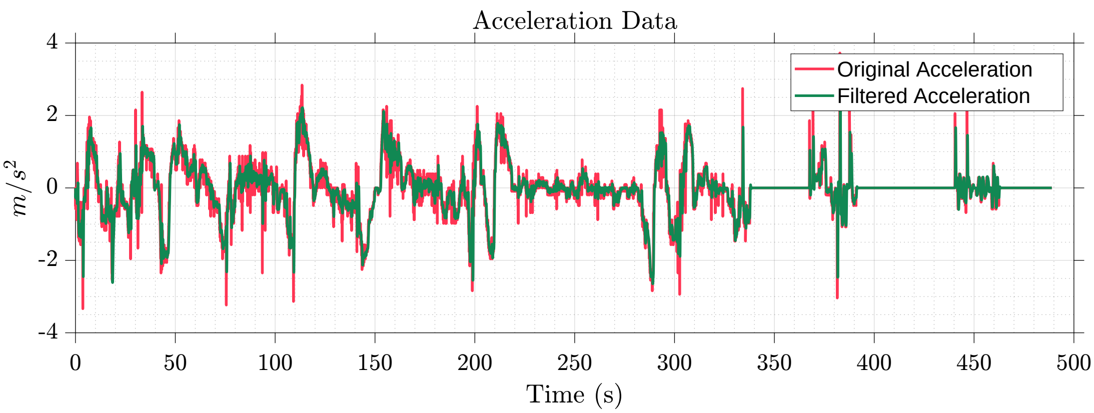
2. See code-snippet

   Optional (no points deducted for not showing these): plots:

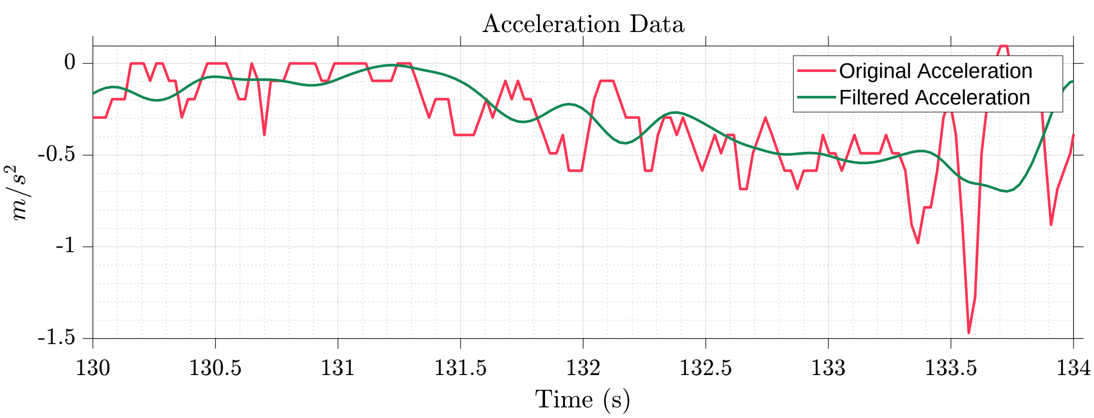




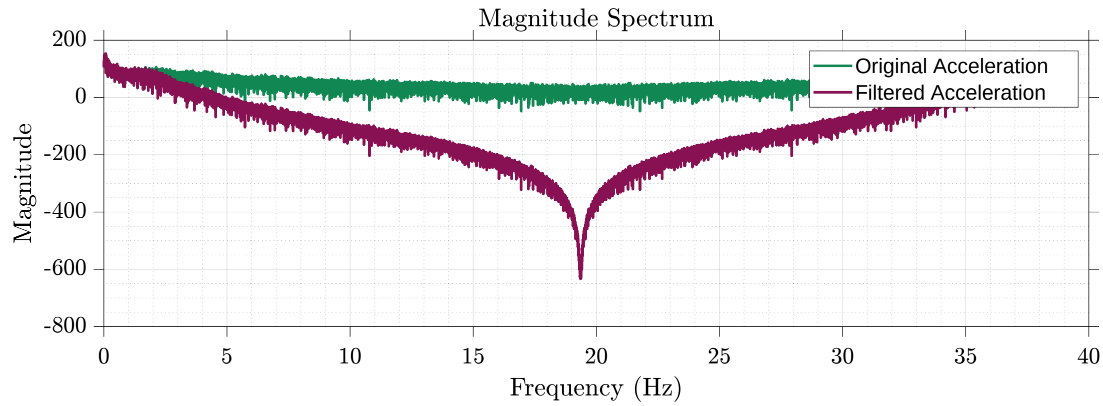
3. Filter coefficients: `b = [0.00046901 0.001876 0.002814 0.001876 0.00046901]`, `a = [1 -3.1537 3.8011 -2.0663 0.42636]`

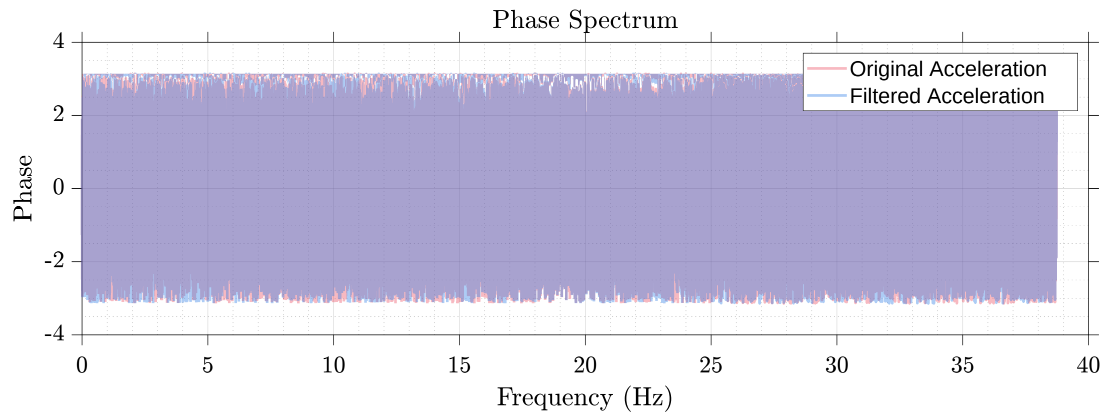


4.



5.





Code-snippet:

```
csv_file = '../Data/accel.csv';


% Read the acceleration data from the CSV file
data = readtable(csv_file);
acceleration = data.Message;

% Plot the acceleration data
fig = figure;
fig.Position = [554, 456, 1300, 420];
plot(data.Time, acceleration, 'LineWidth',2,'color', '#ff3453');
title('Acceleration Data', 'Interpreter','latex');
grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
```

```
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';

xlabel('Time (s)', 'Interpreter','latex');
ylabel('$m/s^2$', 'Interpreter','latex');
set(gca, 'FontSize', 18);
exportgraphics(fig, ...
'../figures/CPE381_FA24_HW5_Q4_Acceleration.pdf');

% Compute the Fourier transform of the acceleration data
fs = 1 / mean(diff(data.Time));  % sampling frequency
acceleration_fft = fft(acceleration);


fig = figure;
fig.Position = [554, 456, 1300, 420];
plot((0:length(acceleration_fft)-1) * fs / length(acceleration_fft), ...
    20.*log(abs(acceleration_fft)), 'LineWidth',2,'color', '#118853');
title('Acceleration Data', 'Interpreter','latex');
grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';

xlabel('Frequency (Hz)', 'Interpreter','latex');
ylabel('Magnitude', 'Interpreter','latex');
title('Magnitude Spectrum',  'Interpreter','latex');
set(gca, 'FontSize', 18);
exportgraphics(fig, ...
'../figures/CPE381_FA24_HW5_Q4_Acceleration_Magnitude_Spectrum.pdf');


fig = figure;
fig.Position = [554, 456, 1300, 420];
plot((0:length(acceleration_fft)-1) * fs / length(acceleration_fft), ...
    angle(acceleration_fft), 'LineWidth',2,'color', '#111399');
title('Acceleration Data', 'Interpreter','latex');
grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
```

```
yaxis.TickLabelInterpreter = 'latex';

xlabel('Frequency (Hz)', 'Interpreter','latex');
ylabel('Phase', 'Interpreter','latex');
title('Phase Spectrum',  'Interpreter','latex');
set(gca, 'FontSize', 18);
exportgraphics(fig, ...
'../figures/CPE381_FA24_HW5_Q4_Acceleration_Phase_Spectrum.pdf');


acceleration = data.Message;

% Define the filter parameters
fs = 1 / mean(diff(data.Time));  % sampling frequency
cutoff_freq = 2;
order = 4;


% Design the low-pass filter
[b, a] = butter(order, cutoff_freq / (fs/2), 'low');

% Apply the filter to the acceleration data
acceleration_filtered = filter(b, a, acceleration);

 % Plot the acceleration data
fig = figure;
fig.Position = [554, 456, 1300, 420];
plot(data.Time, acceleration, 'LineWidth',2,'color', ...
    '#ff3453','DisplayName', 'Original Acceleration');
hold on;
plot(data.Time, acceleration_filtered, 'LineWidth',2, ...
    'color', '#118853', 'DisplayName','Filtered Acceleration');
title('Acceleration Data', 'Interpreter','latex');
grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';
legend;

xlabel('Time (s)', 'Interpreter','latex');
ylabel('$m/s^2$', 'Interpreter','latex');
set(gca, 'FontSize', 18);
exportgraphics(fig, ...
```

```
'../figures/CPE381_FA24_HW5_Q4_Filtered_Acceleration.pdf');


 % Plot the acceleration data
fig = figure;
fig.Position = [554, 456, 1300, 420];
plot(data.Time, acceleration, 'LineWidth',2,'color', ...
    '#ff3453','DisplayName', 'Original Acceleration');
hold on;
plot(data.Time, acceleration_filtered, 'LineWidth',2, ...
    'color', '#118853', 'DisplayName','Filtered Acceleration');
title('Acceleration Data', 'Interpreter','latex');
grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';
legend;
xlim([130, 134]);

xlabel('Time (s)', 'Interpreter','latex');
ylabel('$m/s^2$', 'Interpreter','latex');
set(gca, 'FontSize', 18);
exportgraphics(fig, ...
'../figures/CPE381_FA24_HW5_Q4_Filtered_Acceleration_zoomed.pdf');

% Compute the Fourier transform of the acceleration data
fs = 1 / mean(diff(data.Time));  % sampling frequency
filtered_acceleration_fft = fft(acceleration_filtered);


fig = figure;
fig.Position = [554, 456, 1300, 420];
plot((0:length(acceleration_fft)-1) * fs / length(acceleration_fft), ...
    20.*log(abs(acceleration_fft)), 'LineWidth',2,'color', '#118853', ...
    'DisplayName', 'Original Acceleration');
hold on;
plot((0:length(filtered_acceleration_fft)-1) * fs / length(filtered_acceleration_fft), ...
    20.*log(abs(filtered_acceleration_fft)), 'LineWidth',2,'color', '#881153', ...
    'DisplayName', 'Filtered Acceleration');
alpha(.5);
title('Acceleration Data (Unfiltered and Filtered)', 'Interpreter','latex');
grid on;
grid minor;
```

```
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';

xlabel('Frequency (Hz)', 'Interpreter','latex');
ylabel('Magnitude', 'Interpreter','latex');
title('Magnitude Spectrum',  'Interpreter','latex');
set(gca, 'FontSize', 18);
legend;
exportgraphics(fig, ...
'../figures/CPE381_FA24_HW5_Q4_FilteredAcceleration_Magnitude_Spectrum.pdf');


fig = figure;
fig.Position = [554, 456, 1300, 420];
plot((0:length(acceleration_fft)-1) * fs / length(acceleration_fft), ...
    angle(acceleration_fft), 'LineWidth',2,'Color',[0.9 0.1 0.2 0.3],...
    'DisplayName', 'Original Acceleration');
hold on;
plot((0:length(filtered_acceleration_fft)-1) * fs / length(filtered_acceleration_fft), ...
    angle(filtered_acceleration_fft), 'LineWidth',2,'Color',[0.2 0.5 0.9 0.4], ...
    'DisplayName', 'Filtered Acceleration');
title('Acceleration Data', 'Interpreter','latex');
grid on;
grid minor;
set(gca, 'XColor', [0, 0, 0], 'YColor', [0, 0, 0], 'TickDir', 'out');
xaxis = get(gca, 'XAxis');
xaxis.TickLabelInterpreter = 'latex';
yaxis = get(gca, 'YAxis');
yaxis.TickLabelInterpreter = 'latex';

xlabel('Frequency (Hz)', 'Interpreter','latex');
ylabel('Phase', 'Interpreter','latex');
title('Phase Spectrum',  'Interpreter','latex');
set(gca, 'FontSize', 18);
legend;
exportgraphics(fig, ...
'../figures/CPE381_FA24_HW5_Q4_Filtered Acceleration_Phase_Spectrum.pdf');


% Compute the Mean Squared Error (MSE)
% between original and filtered acceleration data
mse = mean((acceleration - acceleration_filtered).^2);
```

```
% Print the MSE
fprintf('Mean Squared Error (MSE): %f\n', mse);
```