HOMEWORK 8
CPE381

Canvas: `hw08`

Due: 2nd December 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_hw08.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_hw08.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 From Transfer Function to Discret-time Domain(20 points)

Consider a transfer function

$$H(z) = \frac{z-5}{z+3} \tag{1}$$

1. Write $H(z)$ in the inverse power of z, i.e., z^{-1}. **(5 points)**
2. Assume all initial conditions as zero, the system is causal, the input sequence is $x[n]$, and the output sequence is $y[n]$.

   Use the time-shifting property of z-transform and write down the difference-equation corresponding to the transfer function given in Equation (1). **(15 points)**

Solution

1.
$$H(z) = \frac{1-5z^{-1}}{1+3z^{-1}} \tag{2}$$

2.

$$\begin{aligned}\frac{Y(z)}{X(z)} &= \frac{1-5z^{-1}}{1+3z^{-1}} \\ Y(z)+3z^{-1}Y(z) &= X(z)-5z^{-1}X(z) \\ Y(z) &= X(z)-5z^{-1}X(z)-3z^{-1}Y(z)\end{aligned} \tag{3}$$

Taking inverse z-transform and applying time-shift property

$$y[n] = x[n] - 5x[n-1] - 3y[n-1] \tag{4}$$

which is the required difference-equation.

2 Implementing Discrete-Filter in MATLAB from Scratch

A generalized transfer function of a discrete-time filter is given by

$$H(z) = \frac{\sum_{i=0}^{N} b_i z^{-i}}{\sum_{j=0}^{M} a_j z^{-j}} \tag{5}$$

where the numerator is polynomial of order N in the power of z^{-1} and the denominator is polynomial order M in the power of z^{-1}.

Here, coefficients are normalized, i.e. $a_0 = 1$. If $a_0 \neq 1$, then we divide all the coefficients a_j and b_i by a_0. a_j and b_i are called filter coefficients. We can further write them as vector $a = [a_0, a_1, \cdots, a_M]$ where of course $a_0 = 1$, and $b = [b_0, b_1, \cdots, b_N]$.

1. Assume all initial conditions as zero, the system is causal, the input sequence is $x[n]$, and the output sequence is $y[n]$.

   Use the time-shifting property of z-transform and write down the difference-equation corresponding to the generalized transfer function given in Equation (5). Rearrange the difference equation so that the left-hand side of the equation will be $y[n]$ and the rest of the terms will be on the right-hand side. **(15 points)**

2. Write a ***function*** `cpedigitalfilter` called in MATLAB that uses simple for loop to calculate $y[n]$ given the coefficients of the transfer function $H(z)$ and the input $x[n]$. The function should take three arguments

   (a) The input vector $x[n]$
   (b) Filter coefficient b from the numerator polynomial of the transfer function $H(z)$
   (c) Filter coefficient a from the numerator polynomial of the transfer function $H(z)$

   (20 points)

   An additional **5 points** for checking if $a[0] = 1$, and if not, you must normalize the coefficients by dividing all the coefficients with $a[0]$.

3. Use the filter coefficient you obtained in Homework 5, Question 3, Part 6 from the `butter` function to filter out the same noisy sine wave you filtered in Homework 5, Q3. As a part

of this problem, you must use `cpedigitalfilter` to filter the noisy sine wave, and not the `filter` function available in the MATLAB library. Report the value of the mean-squared error as a difference between the original clean sine wave and the filtered sine wave. **(20 points)**

4. You also obtained filter coefficients in Part 3 of Homework 5, Question 4. Along with the acceleration data provided in Homework 5, and the filter coefficients, smoothen out the original acceleration data using `cpedigitalfilter` function. Report the mean-squared error between the original acceleration, and the filtered acceleration. **(20 points)**

Solution

1.

$$\begin{aligned}
H(z) &= \frac{\sum_{i=0}^{N} b_i z^{-i}}{\sum_{j=0}^{M} a_j z^{-j}} \\
\frac{Y(z)}{X(z)} &= \frac{\sum_{i=0}^{N} b_i z^{-i}}{\sum_{j=0}^{M} a_j z^{-j}} \\
Y(z)\sum_{j=0}^{M} a_j z^{-j} &= X(z)\sum_{i=0}^{N} b_i z^{-i} \\
Y(z)[a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M}] &= X(z)[b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-N}]
\end{aligned} \tag{6}$$

Taking inverse z-transform and applying time-shift properties,

$$\begin{aligned}
a_0 y[n] + a_1 y[n-1] + a_2 y[n-2] + \cdots + a_M y[n-M] \\
= b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_N x[n-N] \\
a_0 y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_N x[n-N] \\
- a_1 y[n-1] - a_2 y[n-2] - \cdots - a_M y[n-M] \\
y[n] = \frac{b_0}{a_0} x[n] + \frac{b_1}{a_0} x[n-1] + \frac{b_2}{a_0} x[n-2] + \cdots + \frac{b_N}{a_0} x[n-N] \\
- \frac{a_1}{a_0} y[n-1] - \frac{a_2}{a_0} y[n-2] - \cdots - \frac{a_M}{a_0} y[n-M]
\end{aligned} \tag{7}$$

2.
```
function y = cpedigitalfilter(x, b, a)
    N = length(b);  % order of c
    M = length(a);      % order of d
    % Initialize y vector with zeros
    y = zeros(size(x));

    % Calculate y[n]
    for n = 1:length(x)
        sum_b = 0;
        sum_a = 0;

        % Calculate sum of b_i * x[n-i]
```

```
        for i = 1:N
            if n - i > 0
                % n-i+1
                sum_b = sum_b + b(i) * x(n-i+1);
            end
        end

        % Calculate sum of a_j * y[n-j]
        for j = 1:M
             if n - j > 0
                sum_a = sum_a + a(j) * y(n-j+1);
             end
        end

        % Calculate y[n]
        y(n) = sum_b - sum_a;
    end
end
```

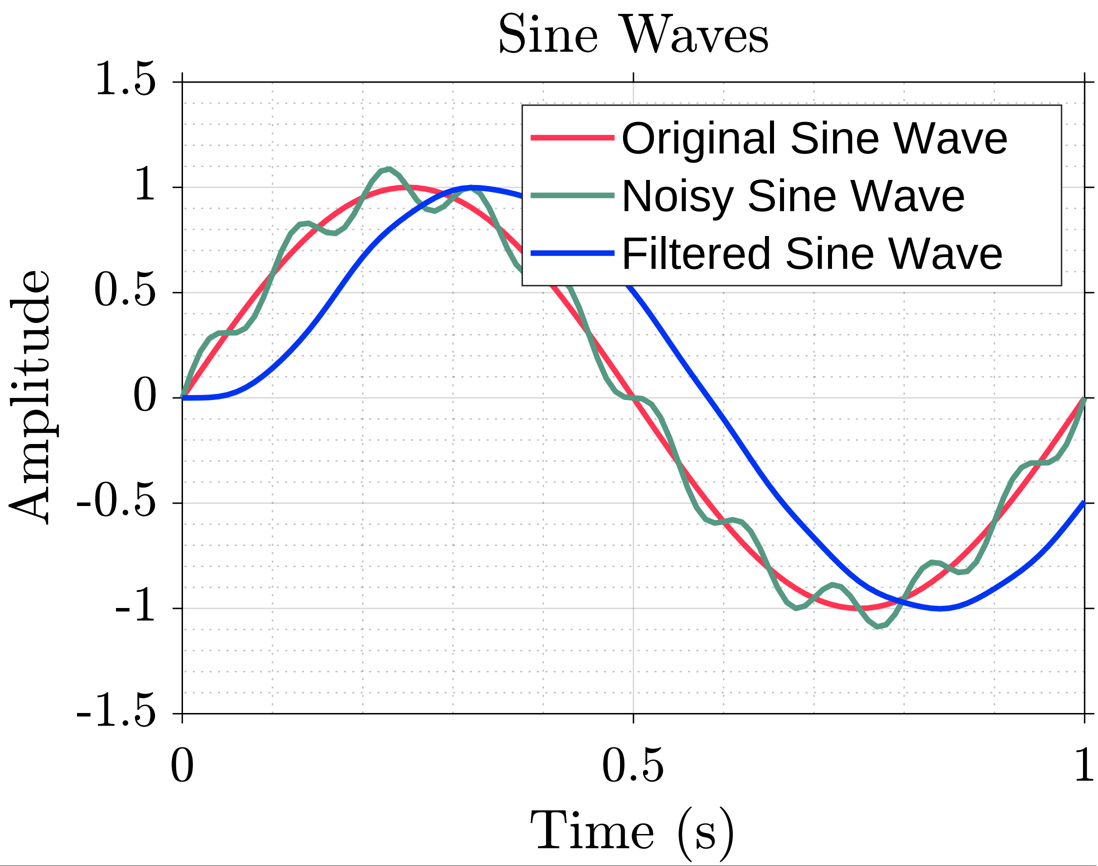
3. Mean Squared Error: 0.12065

   Code:

```
% 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

%

% 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);
noisy_sine = noisy_sine';

% From Homework 5

% b_i coefficients
```

```
b = [0.0004166 0.0016664 0.0024996 0.0016664 0.0004166];
% a_j coefficients
a = [1 -3.1806 3.8612 -2.1122 0.43827];


filtered_sine = cpedigitalfilter(noisy_sine, b, a);
% 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);


mse = mean((noisy_sine - filtered_sine).^2);

fprintf('Mean Squared Error: %.5f\n', mse);


% Function to read x[n] from file
function x = read_x_from_file(filename, column)
    data = csvread(filename);
    x = data(:, column);
end

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
```

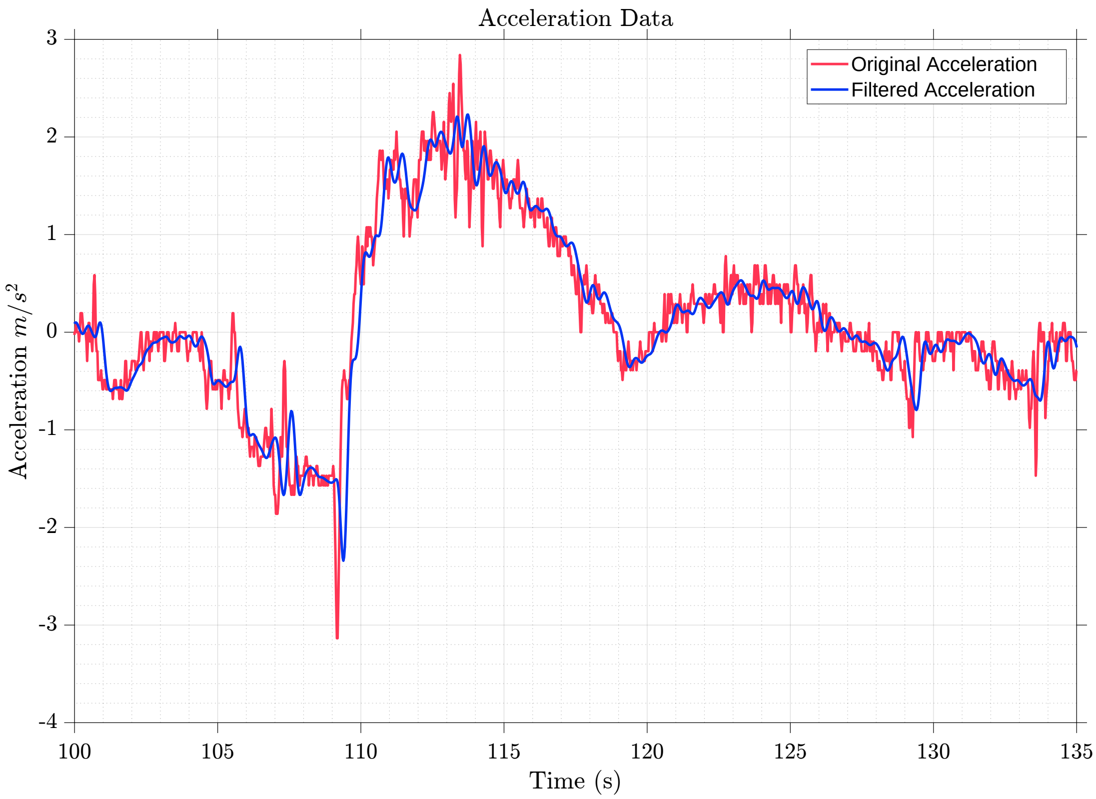


4. Mean Squared Error: 0.10034

   Code:

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

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

% From Homework 5

% b_i coefficients
b = [0.00046901 0.001876 0.002814 0.001876 0.00046901];
% a_j coefficients
a = [1 -3.1537 3.8011 -2.0663 0.42636];


filtered_accel = cpedigitalfilter(acceleration, b, a);
% Plot the original and filtered sine waves
```

```
fig = figure;
fig.Position = [995 331  1382  895];
% Plot the original and noisy sine waves
plot(data.Time, acceleration,'LineWidth',2,'color', '#ff3453');
hold on;
plot(data.Time, filtered_accel,'LineWidth',2,'color', '#0034f3')
legend('Original Acceleration', 'Filtered Acceleration');
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');
xlim([100, 135]);
ylabel('Amplitude', 'Interpreter','latex');
title('Sine Waves', 'Interpreter','latex');
set(gca, 'FontSize', 18);


mse = mean((acceleration - filtered_accel).^2);

fprintf('Mean Squared Error: %.5f\n', mse);
```


Acceleration Data
Original Acceleration
Filtered Acceleration
Acceleration m/s^2
Time (s)
3
2
1
0
-1
-2
-3
-4
100
105
110
115
120
125
130
135