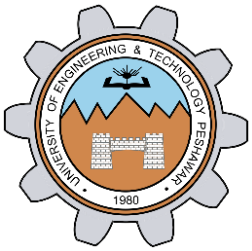
Assignment # 2



Fall 2023

CSE-402 Digital Signal Processing

Submitted by: **Ali Asghar**

Registration No.: **21PWCSE2059**

Class Section: **C**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Submitted to:

Engr. Ihsan Ul Haq

Date:

7th November 2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Code:

Editor - D:\Uni\DSP\Assignment 2\Smoothing.m

Task1.m | Smoothing.m | rand_randn.m

```
fs = 1000;
t = 0:1/fs:1;
A = 5;
f=5;
x = A * sin(2*pi*f*t);

meanNoise1 = 0;
std_dev_noise1 = 1;

meanNoise2 = 1;
std_dev_noise2 = 2;

meanNoise3 = 2;
std_dev_noise3 = -1;

noise1 = meanNoise1 + std_dev_noise1 * randn(size(t));
noise2 = meanNoise2 + std_dev_noise2 * randn(size(t));
noise3 = meanNoise3 + std_dev_noise3 * randn(size(t));

noisySig1 = x + noise1;
noisySig2 = x + noise2;
noisySig3 = x + noise3;

% Create plots for the clean and noisy signals
figure;
```

```
figure;

% Plot the clean sine wave
subplot(2,2,1);
plot(t, x);
title(['Sine Wave with A = ' num2str(A) ' And f = ' num2str(f)]);
xlabel('Time (s)');
ylabel('Amplitude');

% Plot the noisy signals with different standard deviations

subplot(2,2,2);
plot(t, noisySig1);
title(['Noisy Signal 1 (Std Dev = ' num2str(std_dev_noise1), ' Mean = ' num2str(meanNoise1) ' )']);
xlabel('Time (s)');
ylabel('Amplitude');


subplot(2,2,3);
plot(t, noisySig2);
title(['Noisy Signal 2 (Std Dev = ' num2str(std_dev_noise2), ' Mean = ' num2str(meanNoise2) ' )']);
xlabel('Time (s)');
ylabel('Amplitude');

subplot(2,2,4);
```

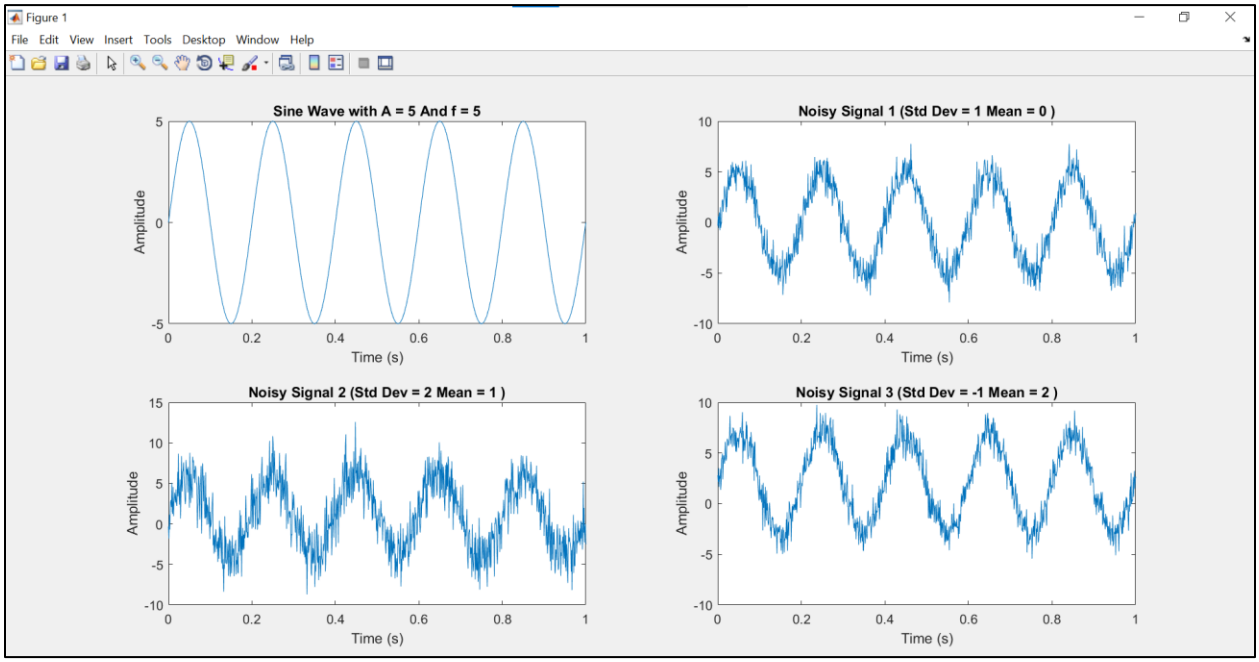
```
    subplot(2,2,4);
    plot(t, noisySig3);
    title(['Noisy Signal 3 (Std Dev = ' num2str(std_dev_noise3), ' Mean = ' num2str(meanNoise3) ' )']);
    xlabel('Time (s)');
    ylabel('Amplitude');


    %Filtering Process

    window_size1 = 3;  % Size of the moving average window
    window_size2 = 5;  % Size of the moving average window
    window_size3 = 7;  % Size of the moving average window

    % Filter the noisy signals using movmean
    filteredSig1_1 = movmean(noisySig1, window_size1);
    filteredSig1_2 = movmean(noisySig1, window_size2);
    filteredSig1_3 = movmean(noisySig1, window_size3);


    % Filter the noisy signals using movmean
    filteredSig2_1 = movmean(noisySig2, window_size1);
    filteredSig2_2 = movmean(noisySig2, window_size2);
    filteredSig2_3 = movmean(noisySig2, window_size3);

```

```
    % Filter the noisy signals using movmean
    filteredSig3_1 = movmean(noisySig3, window_size1);
    filteredSig3_2 = movmean(noisySig3, window_size2);
    filteredSig3_3 = movmean(noisySig3, window_size3);

    % Create plots for the filtered signals
    figure;

    subplot(3,1,1);
    plot(t, filteredSig1_1);
    title(['Filtered Sine Wave (Std Dev = ' num2str(std_dev_noise1), ' Mean = ' num2str(meanNoise1) ') & window size = ' num2str(window_size1)]);
    xlabel('Time (s)');
    ylabel('Amplitude');

    subplot(3,1,2);
    plot(t, filteredSig1_2);
    title(['Filtered Sine Wave (Std Dev = ' num2str(std_dev_noise1), ' Mean = ' num2str(meanNoise1) ') & window size = ' num2str(window_size2)]);
    xlabel('Time (s)');
    ylabel('Amplitude');

    subplot(3,1,3);
    plot(t, filteredSig1_2);
    title(['Filtered Sine Wave (Std Dev = ' num2str(std_dev_noise1), ' Mean = ' num2str(meanNoise1) ') & window size = ' num2str(window_size3)]);
    xlabel('Time (s)');
```

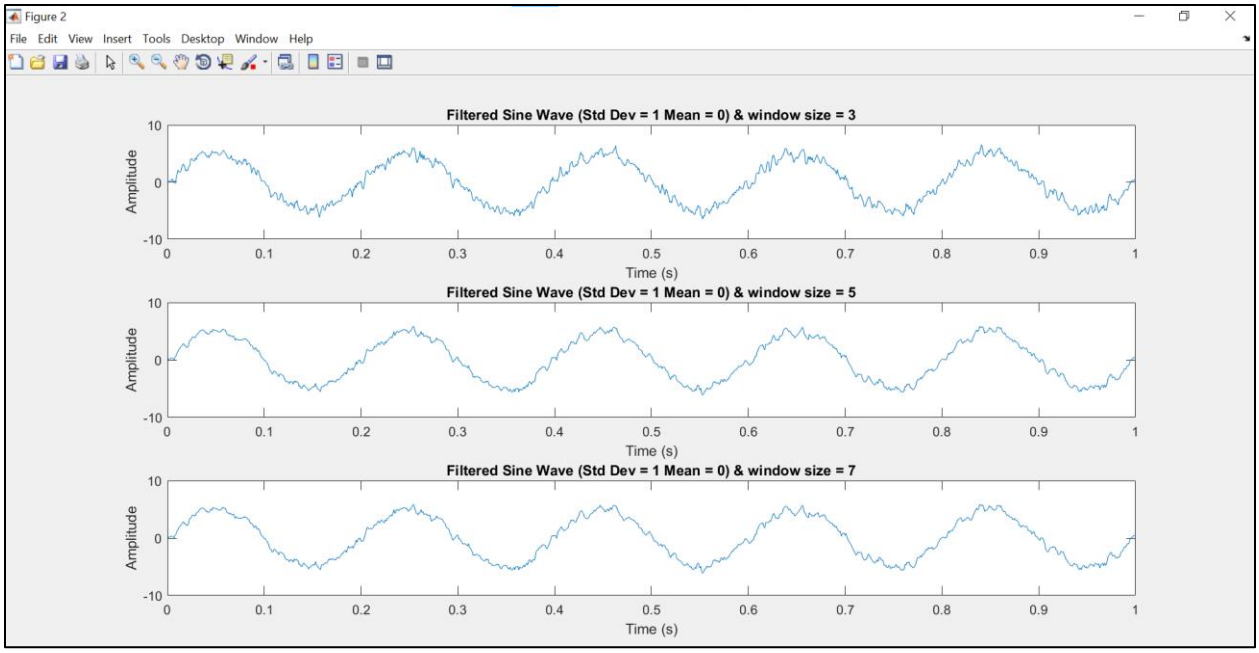
```
    xlabel('Time (s)');
    ylabel('Amplitude');


    % Create plots for the filtered signals
    figure;

    subplot(3,1,1);
    plot(t, filteredSig2_1);
    title(['Filtered Sine Wave (Std Dev = ' num2str(std_dev_noise2), ' Mean = ' num2str(meanNoise2) ') & window size = ' num2str(window_size1)]);
    xlabel('Time (s)');
    ylabel('Amplitude');

    subplot(3,1,2);
    plot(t, filteredSig2_2);
    title(['Filtered Sine Wave (Std Dev = ' num2str(std_dev_noise2), ' Mean = ' num2str(meanNoise2) ') & window size = ' num2str(window_size2)]);
    xlabel('Time (s)');
    ylabel('Amplitude');

    subplot(3,1,3);
    plot(t, filteredSig2_3);
    title(['Filtered Sine Wave (Std Dev = ' num2str(std_dev_noise2), ' Mean = ' num2str(meanNoise2) ') & window size = ' num2str(window_size3)]);
    xlabel('Time (s)');
    ylabel('Amplitude');

```

```
    ylabel('Amplitude');


    % Create plots for the filtered signals
    figure;

    subplot(3,1,1);
    plot(t, filteredSig3_1);
    title(['Filtered Sine Wave (Std Dev = ' num2str(std_dev_noise3), ' Mean = ' num2str(meanNoise3) ') & window size = ' num2str(window_size1)]);
    xlabel('Time (s)');
    ylabel('Amplitude');

    subplot(3,1,2);
    plot(t, filteredSig3_2);
    title(['Filtered Sine Wave (Std Dev = ' num2str(std_dev_noise3), ' Mean = ' num2str(meanNoise3) ') & window size = ' num2str(window_size2)]);
    xlabel('Time (s)');
    ylabel('Amplitude');

    subplot(3,1,3);
    plot(t, filteredSig3_3);
    title(['Filtered Sine Wave (Std Dev = ' num2str(std_dev_noise3), ' Mean = ' num2str(meanNoise3) ') & window size = ' num2str(window_size3)]);
    xlabel('Time (s)');
    ylabel('Amplitude');


```

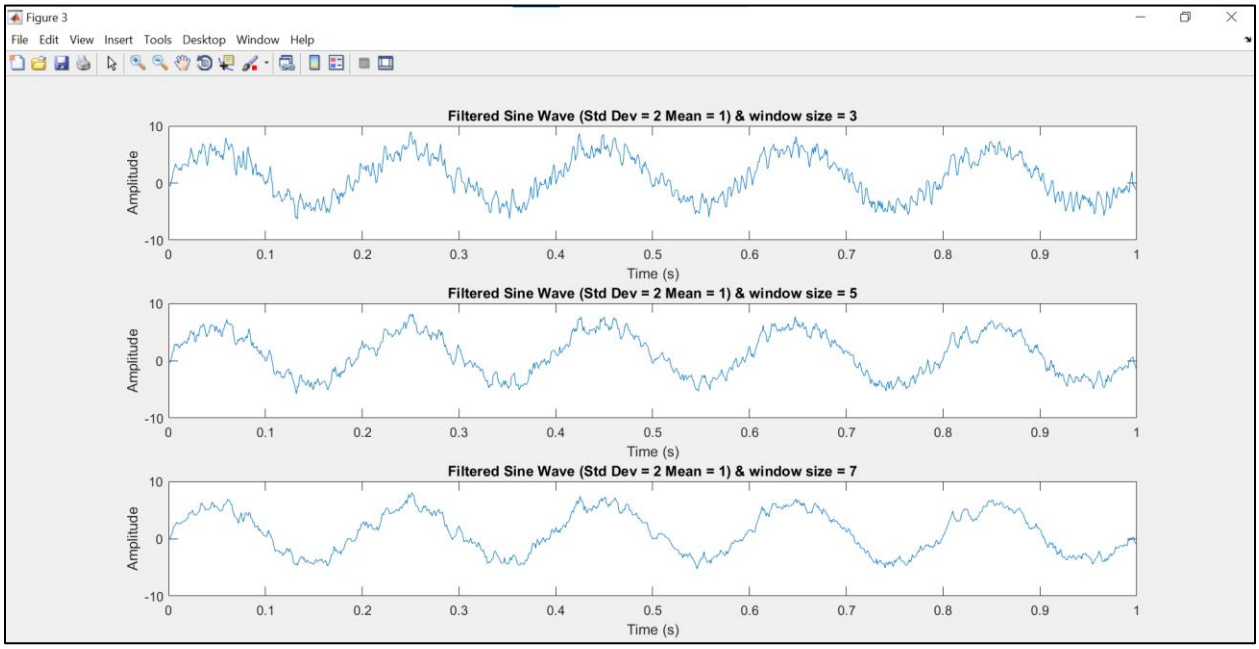
Output:



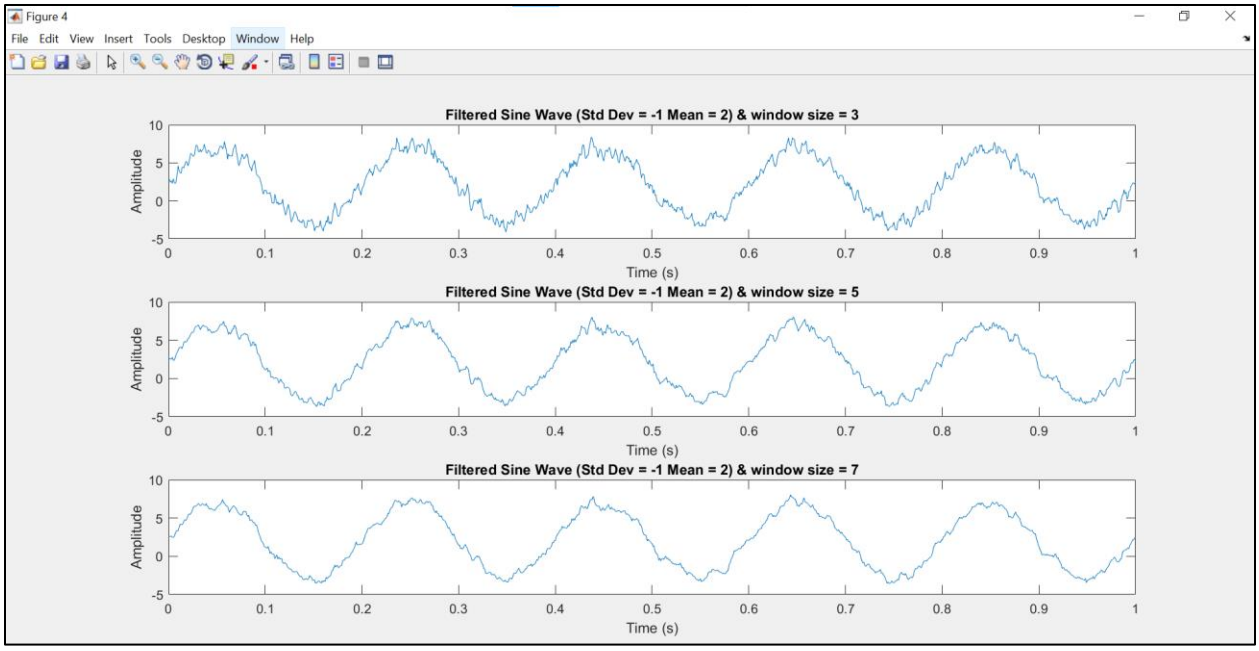
Generation of 3 Noisy Gaussian Signals for different mean and standard deviation



Filtering of Noisy Signal 1 for different window sizes



Filtering of Noisy Signal 2 for different window sizes



Filtering of Noisy Signal 3 for different window sizes

Remarks on Output:

With the help of movemean() function, I was able to filter the noisy signal. The choice of window size is of interest in here. For window size 3 and 5, the output isn't very smooth. For window size 7, the output waveform looks smoothen and denoised.

movmean() is a MATLAB function for moving average filtering, which smooths time series data by calculating the mean within a sliding window of a specified size (k). It's commonly used to reduce noise and emphasize underlying trends or patterns in data.

In summary, the code above showcases a practical approach to signal processing, including noise generation, smoothing, and analysis. It allows for a clear visualization and assessment of the effects of smoothing on noisy signals, which is a common task in various signal processing applications.