DIGITAL SIGNAL PROCESSING

LAB REPORT # 4

SUBMITTED TO:

SIR HAMZA SHAMI

SUBMITTED BY:

AMINA QADEER

DE-42

SYNDICATE - A

DATE:

12/03/23

TASK # 01:

CODE:

Function

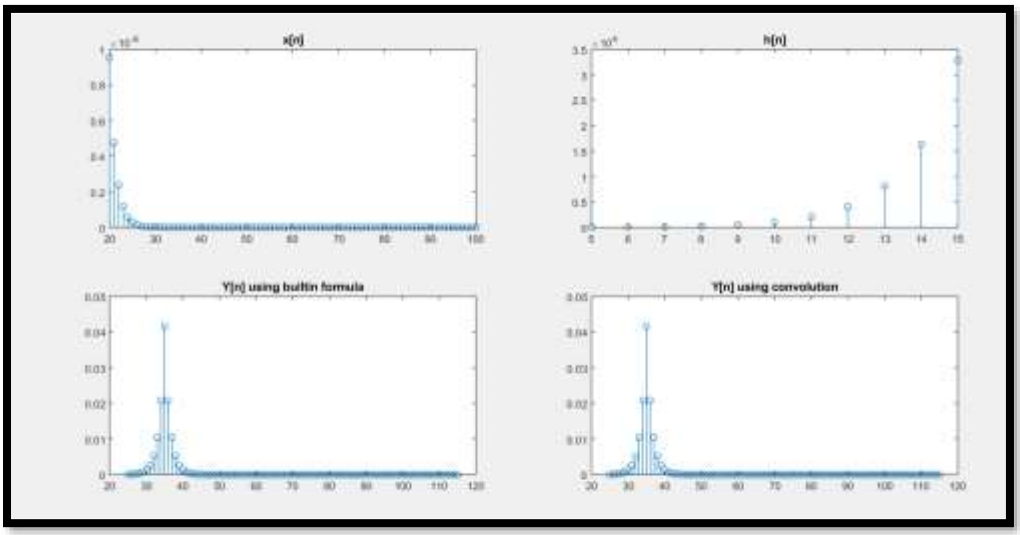
```
function [y] = Convolution (x,n1,h,n2)
x1 = [zeros(1,length(h)-1) x zeros(1,length(h)-1)];
h1 = [fliplr(h) zeros(1,length(x)+length(h)-2)];

n_start = n1(1)+n2(1);
n_end = n1(end) + n2(end);
nn=n_start:n_end;
count = 1;
for d = n_start:n_end
    y(count) = sum(x1.*h1);
    h1 = circshift(h1,[0,1]);
    count = count + 1;
end
```

Script

```
n1 = 20:100;
n2 = 5:15;
a = 0.5;
x = a.^n1;
h = (1/a).^n2;
nn = 25:115;
yy = Convolution(x,n1,h,n2);
subplot(2,2,1)
stem(n1,x);
title('x[n]')
subplot(2,2,2)
stem(n2,h);
title('h[n]')
subplot(2,2,3)
y = conv(x,h);
stem(nn,y);
title('Y[n] using builtin formula')
subplot(2,2,4)
stem (nn,yy)
title('Y[n] using convolution')
```

OUTPUT:



TASK # 02:

CODE:

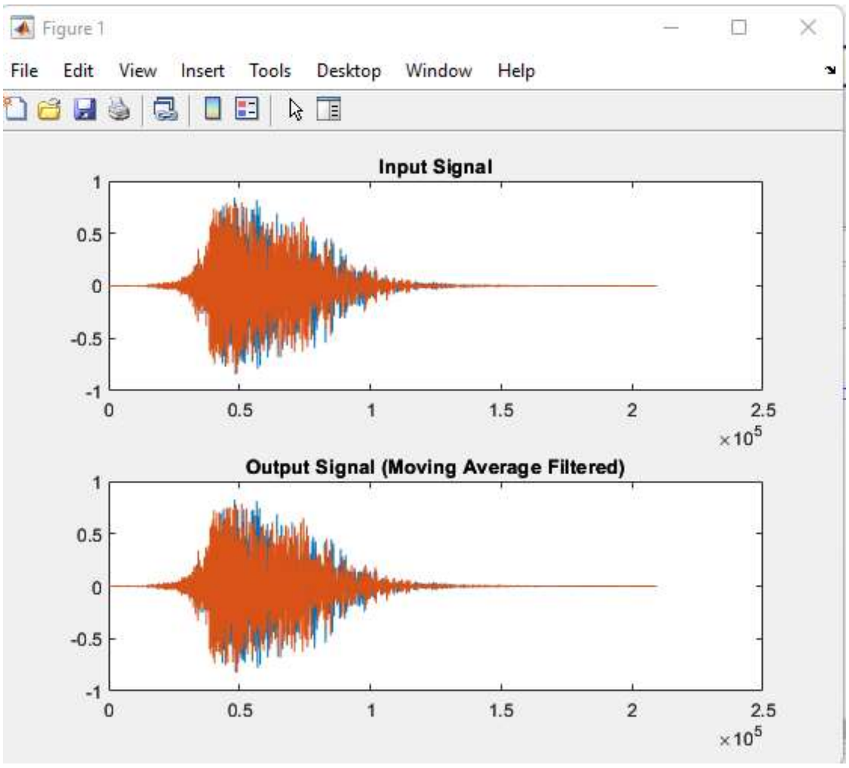
```
% Parameters
window_size = 5; % size of moving average window
input_file = 'C:\Users\Amina Qadeer\OneDrive - National University of Sciences &
Technology\Documents\audio.wav'; % path to input audio file
output_file = 'filtered_signal.wav'; % path to output filtered audio file

% Load input signal from file
[input_signal, sample_rate] = audioread("audio.wav");

% Apply moving average filter
output_signal = filter(ones(1, window_size)/window_size, 1, input_signal);

% Write output signal to file
audiowrite(output_file, output_signal, sample_rate);

% Plot input and output signals
figure;
subplot(2,1,1);
plot(input_signal);
title('Input Signal');
subplot(2,1,2);
plot(output_signal);
title('Output Signal (Moving Average Filtered)');
```



In this code, the filter function is used to apply the moving average filter to the input signal. The first argument is the filter coefficients, which in this case are an array of 1s divided by the size of the window. The second argument is 1, indicating that the filter is a digital filter. The third argument is the input signal.

The filtered output signal is then written to a WAV file using the audiowrite function. Finally, the input and output signals are plotted for comparison.

TASK # 02:

B) CODE:

```
% Prompt user to speak
fs = 16000; % sample rate
duration = 5; % duration in seconds
recObj = audiorecorder(fs, 16, 1);
disp('Speak into microphone...');
recordblocking(recObj, duration);
disp('Done.');
y = getaudiodata(recObj);

% Set filter parameters
N = 5; % filter length
kernel = ones(N, 1) / N; % averaging filter kernel

% Apply filter using custom convolution function
y_filtered_custom = my_conv(y, kernel);
```

```
% Apply filter using MATLAB's built-in convolution function
y_filtered_matlab = conv(y, kernel, 'same');

% Plot original and filtered signals
t = (0:length(y)-1) / fs;
subplot(3, 1, 1);
plot(t, y);
xlabel('Time (s)');
ylabel('Amplitude');
title('Original Signal');
subplot(3, 1, 2);
plot(t, y_filtered_custom);
xlabel('Time (s)');
ylabel('Amplitude');
title(sprintf('Moving Average Filtered Signal (Custom Function, N=%d)', N));
subplot(3, 1, 3);
plot(t, y_filtered_matlab);
xlabel('Time (s)');
ylabel('Amplitude');
title(sprintf('Moving Average Filtered Signal (MATLAB Function, N=%d)', N));

% Compare filtered signals
figure;
subplot(2, 1, 1);
plot(t, y_filtered_custom);
xlabel('Time (s)');
ylabel('Amplitude');
title(sprintf('Custom Convolution Filtered Signal (N=%d)', N));
subplot(2, 1, 2);
plot(t, y_filtered_matlab);
xlabel('Time (s)');
ylabel('Amplitude');
title(sprintf('MATLAB Convolution Filtered Signal (N=%d)', N));


script:
function y_out = my_conv(x, kernel)
    % Reshape input signal and kernel to be column vectors
    x = x(:);
    kernel = kernel(:);
    % Pad input signal with zeros
    x_padded = [zeros(length(kernel)-1,1); x; zeros(length(kernel)-1,1)];
    % Initialize output signal
    y_out = zeros(size(x));
    % Convolve kernel with input signal
    for i = 1:length(x)
        y_out(i) = kernel' * x_padded(i:i+length(kernel)-1);
    end
end
```

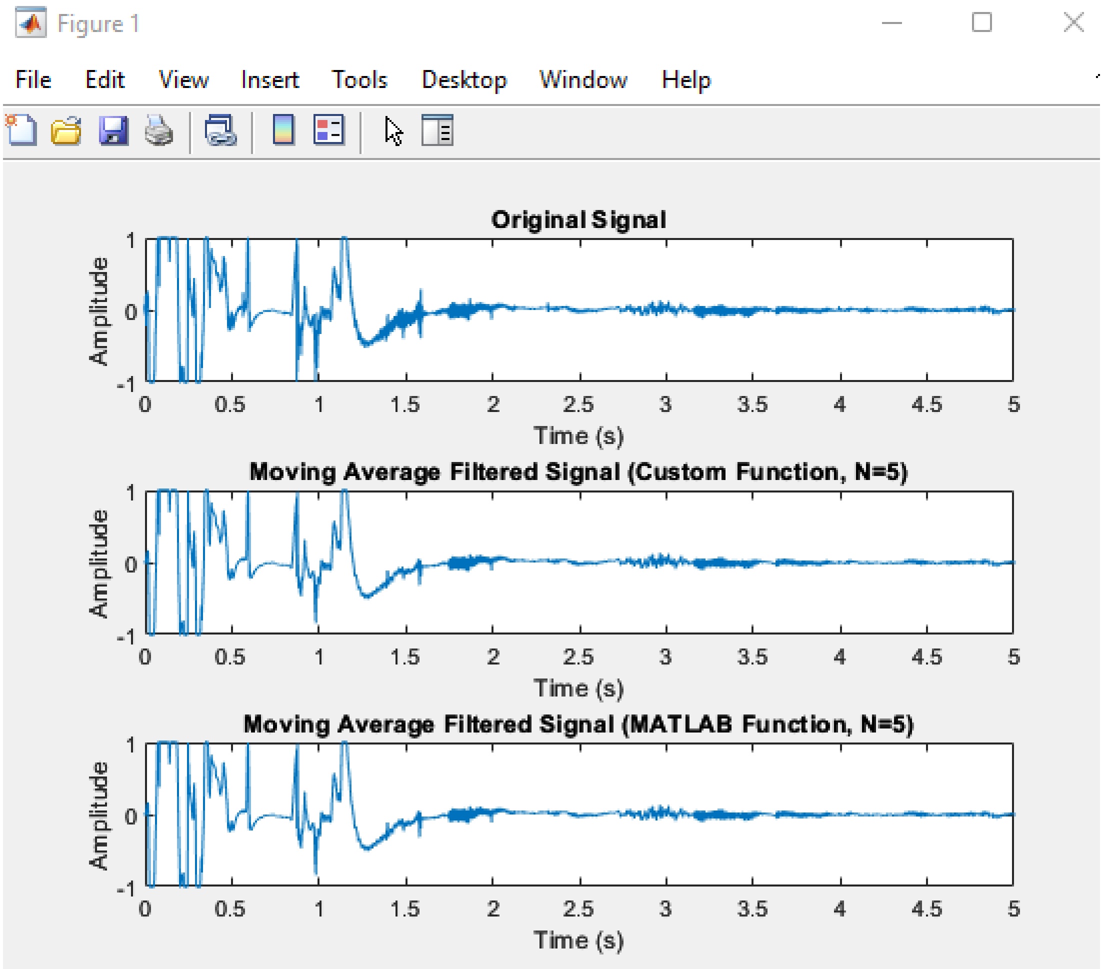


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Original Signal
Amplitude
Time (s)
Moving Average Filtered Signal (Custom Function, N=5)
Amplitude
Time (s)
Moving Average Filtered Signal (MATLAB Function, N=5)
Amplitude
Time (s)