**DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING**

**COLLEGE OF E&ME, NUST, RAWALPINDI**

**Subject Name**

**Digital Signal Processing**

**Lab Number**

**6**

**SUBMITTED TO:**

**LE Sundas Ashraf**

**SUBMITTED BY:**

**Student Name**

1. Wahaaj Nasir

**Reg#413238**

**DE- 44 Dept C&SE**

**Objectives:**

Processing in MATLab

**Related Topic/Chapter in theory class:**

Basics Of Digital Signal Processing

**Hardware/Software required:**

Hardware: PC

Software Tool: MATLab

**Task 1:**

**Write a MATLAB function which takes two signals x[n] and h[n] as parameters and perform the convolution of the two signals.**

**Solution:**

%%

function y = man\_conv(x, h)

y = zeros(length(x) + length(h) -1);

for i = 1:length(x)

for j = 1:length(h)

y(i+j-1) = y(i+j-1) + x(i) \* h(j);

end

end

end

x\_n = [1 2 3 4];

h\_n = [1 1 1];

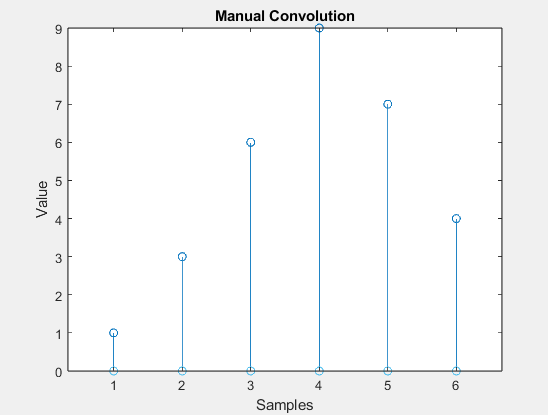
y\_man = man\_conv(x\_n, h\_n);

stem(y\_man)

xlabel("Samples")

ylabel("Value")

title("Manual Convolution")

****

**Task 2:**

**Use MATLAB built-in function of convolution to perform the convolution of two same signals and compare the results with Lab Task # 01. The result of convolution should be same. MATLAB function for convolution is conv.**

**Solution**

%% Task 2

y\_mat = conv(x\_n, h\_n);

subplot(2, 1, 1)

stem(y\_man)

xlabel("Samples")

ylabel("Value")

title("Manual Convolution")

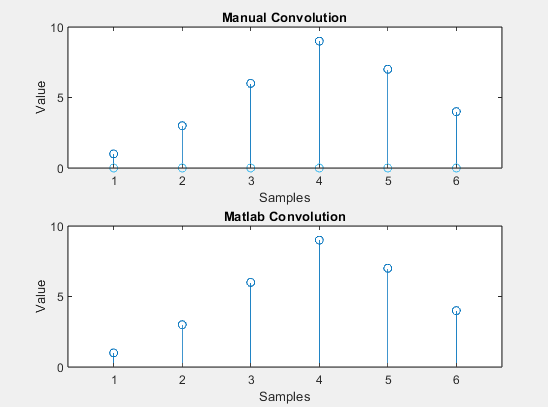
subplot(2, 1, 2)

stem(y\_mat)

xlabel("Samples")

ylabel("Value")

title("Matlab Convolution")



**Task 3:**

**Generate a clean sinusoidal signal with a duration of 2 seconds and a sampling frequency of 1000 Hz. Then, create Gaussian noise with an amplitude of 0.5 and add it to the clean sinusoidal signal to produce a noisy signal**

**Solution**

fs = 1000;

T = 2;

t = 0:1/fs:T;

clean\_signal = sin(2\*pi\*10\*t);

noise = 0.5 \* randn(size(t));

noisy\_signal = clean\_signal + noise;

subplot(2, 1, 1)

plot(clean\_signal)

xlabel("Time")

ylabel("Amplitude")

title("Original Signal")

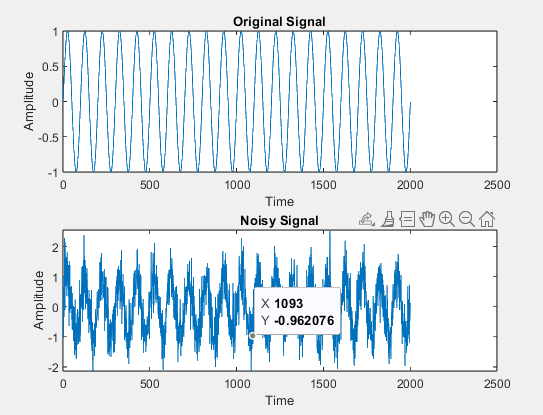
subplot(2, 1, 2)

plot(noisy\_signal)

xlabel("Time")

ylabel("Amplitude")

title("Noisy Signal")



**Task 4:**

**Create a MATLAB function to implement a moving average filter for the noisy signal generated in Task 2. Apply the moving average filter using different window sizes (e.g., 3, 5, 10) and compare the filtered signals. Analyze the trade-off between the level of smoothing achieved and the potential distortion of the original signal.**

**Solution**

%% Task 4

function y = av\_filter(x, M)

y = zeros(length(x));

for i = 1:length(x)

start = max(1, i - floor(M/2));

edge = min(length(x), i+floor(M/2));

y(i) = 1/M \* mean(x(start:edge));

end

end

y\_filtered\_3 = av\_filter(noisy\_signal, 3);

y\_filtered\_5 = av\_filter(noisy\_signal, 5);

y\_filtered\_10 = av\_filter(noisy\_signal, 10);

subplot(5, 1, 1)

plot(clean\_signal)

xlabel("Time")

ylabel("Amplitude")

title("Original Signal")

subplot(5, 1, 2)

plot(noisy\_signal)

xlabel("Time")

ylabel("Amplitude")

title("Noisy Signal")

subplot(5, 1, 3)

plot(y\_filtered\_3)

xlabel("Time")

ylabel("Amplitude")

title("Filtered Signal by 3")

subplot(5, 1, 4)

plot(y\_filtered\_5)

xlabel("Time")

ylabel("Amplitude")

title("Filtered Signal by 5")

subplot(5, 1, 5)

plot(y\_filtered\_10)

xlabel("Time")

ylabel("Amplitude")

title("Filtered Signal by 10")

