% Carlos Lazo

% ECE 503

% Homework #10

% Due: 04/05/10

%% 1) Zero-Phase Linear Filtering

clear all; close all; clc;

% Define all frequencies, filter order, and time vector

% based on the sampling rate.

Fc = 25;

Fs = 100;

N = 4;

t = 0.0 : 1/Fs : 1.0;

% Create 4th order (5 coefficient) lowpass filter.

b = fir1(N,(Fc/(Fs/2)),'low');

% Create 5 sine waves at the given frequencies and filter:

f1 = 5; f2 = 10; f3 = 15; f4 = 20; f5 = 25;

s1 = sin(2\*pi\*f1.\*t); y1 = filter(b,1,s1);

s2 = sin(2\*pi\*f2.\*t); y2 = filter(b,1,s2);

s3 = sin(2\*pi\*f3.\*t); y3 = filter(b,1,s3);

s4 = sin(2\*pi\*f4.\*t); y4 = filter(b,1,s4);

s5 = sin(2\*pi\*f5.\*t); y5 = filter(b,1,s5);

figure;

subplot(2,1,1)

plot(t, s1);

title('Original Sine Wave, 5Hz');

grid on;

subplot(2,1,2);

plot(t, y1);

title('Sine Wave After Causal Filter, 5Hz');

grid on

fprintf('\nPaused -----\n'); pause

close all;

figure;

subplot(2,1,1)

plot(t, s2);

title('Original Sine Wave, 10Hz');

grid on;

subplot(2,1,2);

plot(t, y2);

title('Sine Wave After Causal Filter, 10Hz');

grid on;

fprintf('\nPaused -----\n'); pause

close all;

figure;

subplot(2,1,1)

plot(t, s3);

title('Original Sine Wave, 15Hz');

grid on;

subplot(2,1,2);

plot(t, y3);

title('Sine Wave After Causal Filter, 15Hz');

grid on;

fprintf('\nPaused -----\n'); pause

close all;

figure;

subplot(2,1,1)

plot(t, s4);

title('Original Sine Wave, 20Hz');

grid on;

subplot(2,1,2);

plot(t, y4);

title('Sine Wave After Causal Filter, 20Hz');

grid on;

fprintf('\nPaused -----\n'); pause

close all;

figure;

subplot(2,1,1)

plot(t, s5);

title('Original Sine Wave, 25Hz');

grid on;

subplot(2,1,2);

plot(t, y5);

title('Sine Wave After Causal Filter, 25Hz');

grid on;

fprintf('\nPaused -----\n'); pause

close all;

% Now use filtfilt to see the new results:

y1\_ff = filtfilt(b,1,s1);

y2\_ff = filtfilt(b,1,s2);

y3\_ff = filtfilt(b,1,s3);

y4\_ff = filtfilt(b,1,s4);

y5\_ff = filtfilt(b,1,s5);

figure;

subplot(2,1,1)

plot(t, s1);

title('Original Sine Wave, 5Hz');

grid on;

subplot(2,1,2);

plot(t, y1\_ff);

title('Sine Wave After Using filtfilt(), 5Hz');

grid on

fprintf('\nPaused -----\n'); pause

close all;

figure;

subplot(2,1,1)

plot(t, s2);

title('Original Sine Wave, 10Hz');

grid on;

subplot(2,1,2);

plot(t, y2\_ff);

title('Sine Wave After Using filtfilt(), 10Hz');

grid on;

fprintf('\nPaused -----\n'); pause

close all;

figure;

subplot(2,1,1)

plot(t, s3);

title('Original Sine Wave, 15Hz');

grid on;

subplot(2,1,2);

plot(t, y3\_ff);

title('Sine Wave After Using filtfilt(), 15Hz');

grid on;

fprintf('\nPaused -----\n'); pause

close all;

figure;

subplot(2,1,1)

plot(t, s4);

title('Original Sine Wave, 20Hz');

grid on;

subplot(2,1,2);

plot(t, y4\_ff);

title('Sine Wave After Using filtfilt(), 20Hz');

grid on;

fprintf('\nPaused -----\n'); pause

close all;

figure;

subplot(2,1,1)

plot(t, s5);

title('Original Sine Wave, 25Hz');

grid on;

subplot(2,1,2);

plot(t, y5\_ff);

title('Sine Wave After Using filtfilt(), 25Hz');

grid on;

fprintf('\nPaused -----\n'); pause

close all;

% Carlos Lazo

% ECE 503

% Homework #10

% Due: 04/05/10

%% 2 - Sample Rate Conversion

function out\_v = HW10\_upsample(in\_v, I)

% Start by creating a vector to use for zero-insertion:

zero\_v = zeros(1, I \* length(in\_v));

% Insert values of the input vector into the zero vector:

for i = 1 : length(in\_v), zero\_v((i\*I)-(I-1)) = in\_v(i); end

% Create a 10th-order butterworth filter for this model.

% Lowpass filter at pi/I.

N = 10;

[b, a] = butter(N, 1/I);

% Since we want a non-causual zero-phase filter, use filtfilt.

% Need to multiply filter by gain I.

out\_v = I \* filtfilt(b, a, zero\_v);

% Carlos Lazo

% ECE 503

% Homework #10

% Due: 04/05/10

%% 2 - Sample Rate Conversion

clear all; close all; clc;

% Define all frequencies and the given sine wave:

F = 1;

Fs = 20;

t = 0.0 : 1/Fs : 1.0;

x = sin(2\*pi\*F.\*t);

% Upsample original signal:

I = 3;

x\_up = HW10\_upsample(x,I);

% Plot both the original signal and the upsampled signal:

figure;

subplot(2,1,1);

stem(x);

xlabel('Sample');

ylabel('Magnitude');

title('Original Signal');

grid on;

subplot(2,1,2);

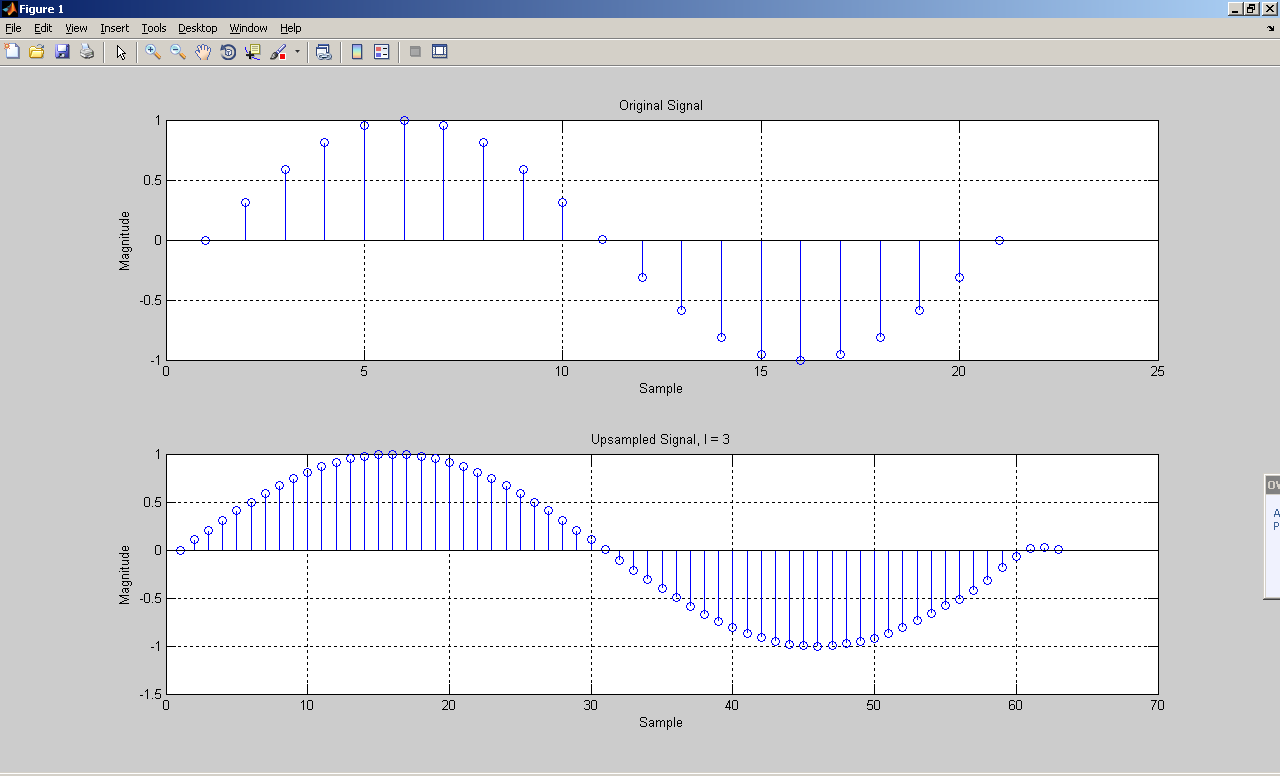
stem(x\_up);

xlabel('Sample');

ylabel('Magnitude');

title('Upsampled Signal, I = 3');

grid on;



% Carlos Lazo

% ECE 503

% Homework #10

% Due: 04/05/10

%% 3) Hilbert Transformers

clear all; close all; clc;

% Define the filder order.

N = 20;

% Create the different Hilbert Transformers,

% and model the different cutoff frequencies.

b1 = firpm(N, [.1 .90], [1 1], 'Hilbert');

b2 = firpm(N, [.1 .93], [1 1], 'Hilbert');

b3 = firpm(N, [.1 .96], [1 1], 'Hilbert');

b4 = firpm(N, [.1 .99], [1 1], 'Hilbert');

[H1, w1] = freqz(b1, 1, 1024);

[H2, w2] = freqz(b2, 1, 1024);

[H3, w3] = freqz(b3, 1, 1024);

[H4, w4] = freqz(b4, 1, 1024);

% Plot all magnitude responses:

figure ('Name', 'Homework 10 - Hilbert Transformers');

subplot(2,2,1);

plot(w1,abs(H1));

xlabel('Frequency in radians');

ylabel('Magnitude');

title('Hilbert Transformer - Passband cutoff @ w = 0.90\*pi');

grid on;

subplot(2,2,2);

plot(w2,abs(H2));

xlabel('Frequency in radians');

ylabel('Magnitude');

title('Hilbert Transformer - Passband cutoff @ w = 0.93\*pi');

grid on;

subplot(2,2,3);

plot(w3,abs(H3));

xlabel('Frequency in radians');

ylabel('Magnitude');

title('Hilbert Transformer - Passband cutoff @ w = 0.96\*pi');

grid on;

subplot(2,2,4);

plot(w4,abs(H4));

xlabel('Frequency in radians');

ylabel('Magnitude');

title('Hilbert Transformer - Passband cutoff @ w = 0.99\*pi');

grid on;

