MultimediaSignal Processing 1^{st} Module and
Fundamentals of Multimedia Signal Processing

date: 7/11/2018

Ex.1 (Pt.13)

An analog signal $x(t)=2\cos(2\pi 20t)+3\sin(2\pi 60t)+4\cos(2\pi 80t)$ is sampled at 160 samples/s and filtered with an IIR filter with the following finite differences equation:

$$y[n]=x[n]+\sqrt{2}x[n-1]+x[n-2]-0.9\sqrt{2}y[n-1]-0.81y[n-2]$$

1. [2pts] Provide the z-transform of the filter.
2. [3pts] Provide the zeros-poles plot of the filter.
3. [3pts] Represent an approximate behavior of the magnitude and phase of the filter in the range (0 - π).
4. [5pts] What will be the discrete output signal when the input is the sampled version of $x(t)$?

Ex.2 (Pt.9)

A signal $x[n]=\{1,2,2,1,0,0,1,2,2\}$ has to be downsampled of an order of 3. In order to reduce aliasing a low pass filter $h[n]=\{1,2,3,3,2,1\}$ is adopted before downsampling in a polyphase manner.

1. **[3 pts.]** provide the schematics of the whole process in traditional and polyphase version.
2. **[6 pts.]** provide the output of every polyphase filter and the whole output of the filter.

Ex.3 (Pt. 11 – MATLAB code)

Given the filter h(n) = [1, 0.75, 0.5, 0.25, 0.5, 0.75, 1] with n starting from 0 and x(t)= A cos(2 pi f t);

1. **[3 pts.]** Create the signal x(tn) as x(t) from 0 to 0.5 seconds sampled at Fs=1000Hz, A=0.8 and f=50 Hz;
2. **[4 pts.]** Compute y_t and y_f as x filtered with h in the time and in the frequency domain, respectively
3. **[3 pts.]** Plot x, y_t and y_f in three subplots in the time domain (in seconds)
4. **[2 pts.]** BONUS: plot only the first 0.1 seconds of them. Hint: What does A(B>c) mean?

Solutions

Ex.1

The sampled signal will be:

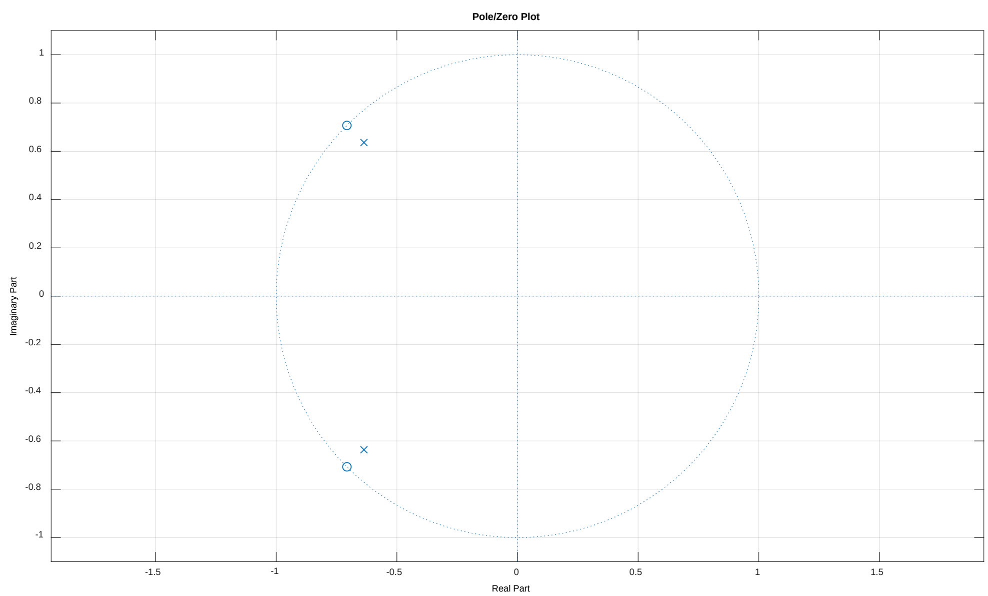
$$x[n] = 2\sin\left(\frac{\pi}{4}n\right) + 3\cos\left(\frac{3\pi}{4}n\right) + 4\sin(\pi n)$$

The z transform of the filter will be:

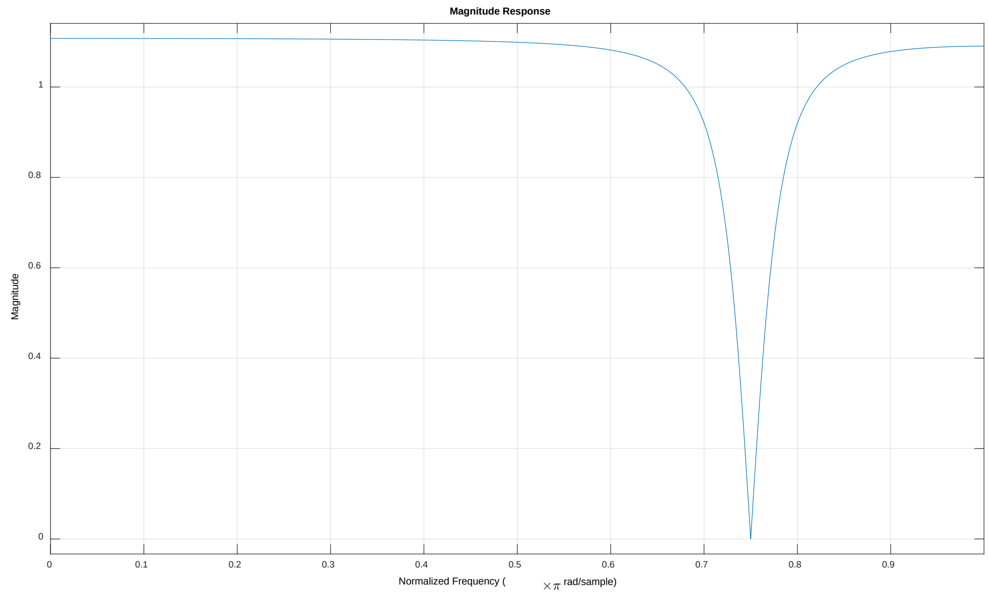
$$H[z] = \frac{1+\sqrt{2}z^{-1}+z^{-2}}{1+0.9\sqrt{2}z^{-1}+0.81z^{-2}}$$

The filter has two zeros in $z_{zeros} = e^{\pm j\frac{3}{4}\pi}$ and two poles in $z_{poles} = 0.9e^{\pm j\frac{3}{4}\pi}$

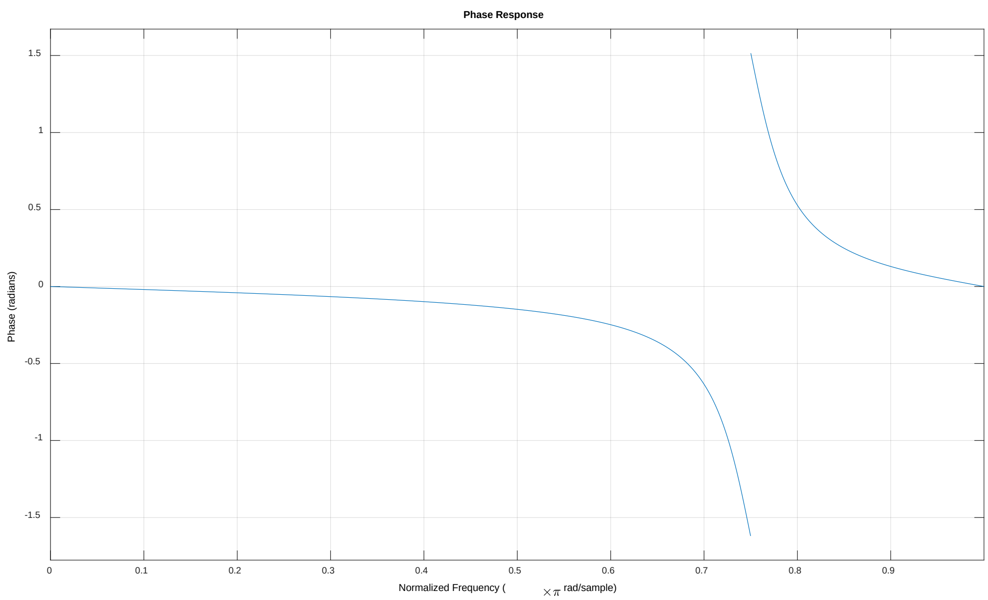
The zeros-poles plot is



The magnitude will be:



And the phase:



The output signal will be without the component at 60Hz (3/4 π in normalized pulsations) and for the other two components we have:

$$\left|H\left[z=e^{j\pi/4}\right]\right| = \left| \frac{1+\sqrt{2}\left(\frac{\sqrt{2}}{2}-\frac{\sqrt{2}}{2}j\right)-j}{1+0.9\sqrt{2}\left(\frac{\sqrt{2}}{2}-\frac{\sqrt{2}}{2}j\right)-0.81j} \right| = \left|\frac{2-2j}{1.9-1.71j}\right| = \frac{|7.22-0.38j|}{6.5341} = 1.1065$$

$$\angle H\left[z=e^{j\pi/4}\right] = \angle\left(7.22-0.38j\right) = \tan^{-1}\left(\frac{-0.38}{7.22}\right) = -0.053rad$$

$$\left|H\left[z=e^{j\pi}\right]\right| = \left|\frac{1-\sqrt{2}+1}{1-0.9\sqrt{2}+0.81}\right| = 1.09$$

$$\angle H\left[z=e^{j\pi}\right] = 0$$

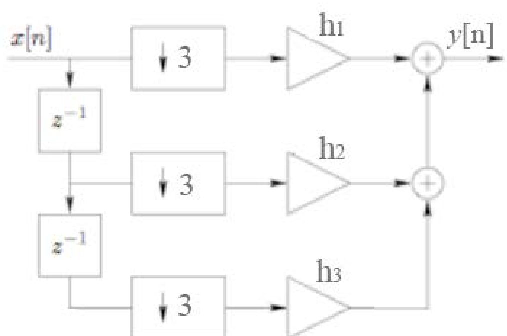
The output will be:

$$y[n] = 2\cdot 1.1065\cos\left(\frac{\pi}{4}n - 0.053\right) + 4\cdot 1.09\cos(\pi n)$$

Ex.2

Without the polyphase implementation the output signal will be obtained convolving $x[n]$ with $h[n]$ and downsampling (M=3) $y[n] = x[n] * h[n] = \{1,4,9,14,16,14,10, 8,10,13,14,11,6,2\}$ and after the downsampling: $y_d[n] = \{1,14,10,13,6\}$

The polyphase implementation would be:



The $h_0 = \{1,3\}$ filter will receive as input the sequence $\{1,1,1\}$, the $h_1 = \{2,2\}$ filter will receive as input the sequence $\{0,2,0,2\}$, the $h_2 = \{3,1\}$ filter will receive as input the sequence $\{0,2,0,2\}$

The three outputs from the 3 polyphase filters will then be

$y_0 = \{1,4,4,3\}$

$y_1 = \{0,4,4,4,4\}$ [Note: the first value is zero since the signal values before the first one are assumed to be equal to zero, this is the initialization value for the memory buffer].

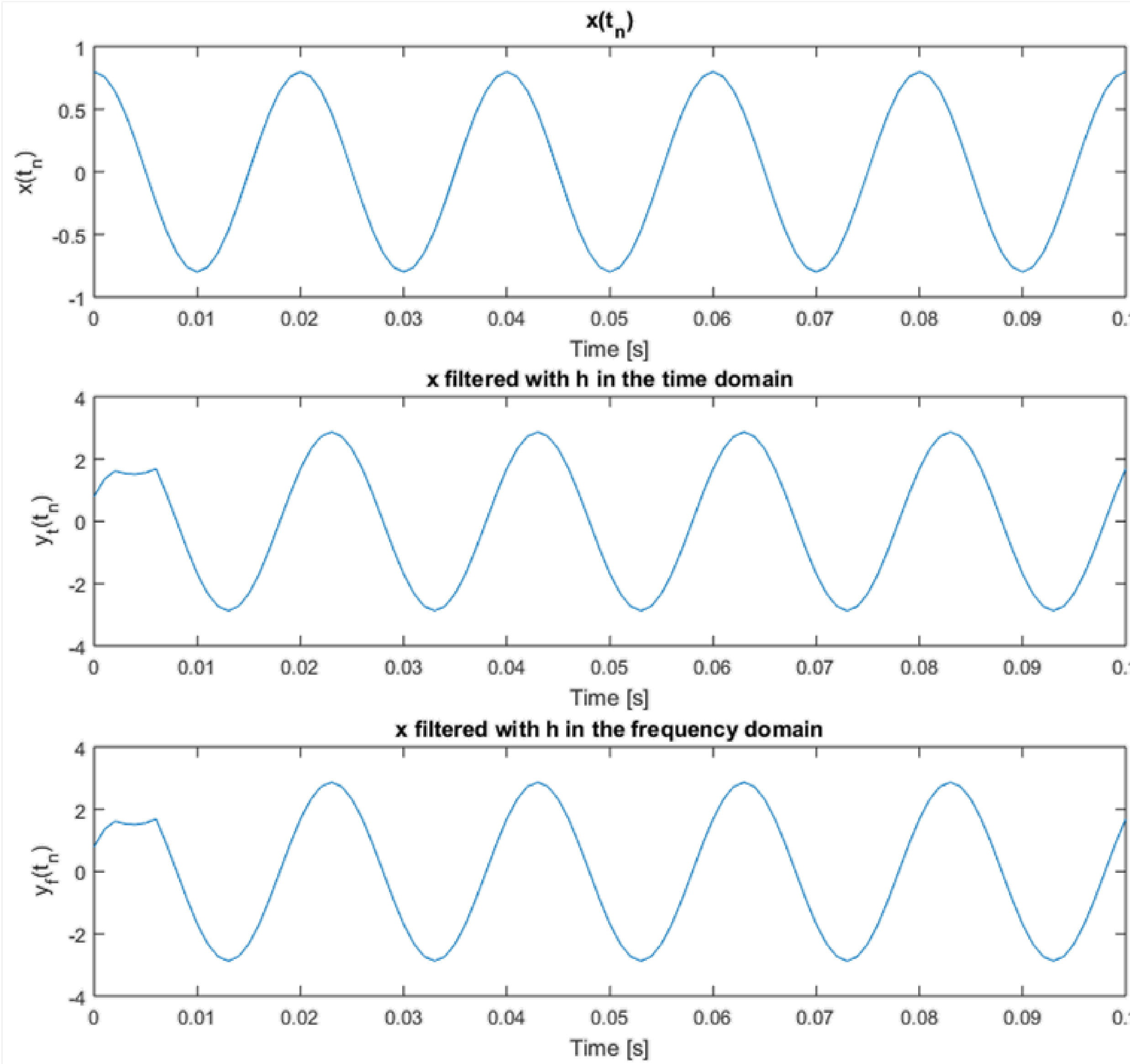
$y_2 = \{0,6,2,6,2\}$ [see previous note for the initial zero value].

And the total output will be the sum columwise of these outputs: $y[n] = \{1,14,10,13,6\}$

Ex.3

```
clear all
close all
clc
```

```
% Given the filter h(n) = [1 .75 .5 .25 .5 .75 1] with
n_h
% starting from 0 and x(t)= A* cos(2*pi*f*t);

h=[1 .75 .5 .25 .5 .75 1];

% 1) create the signal x(t_n) as x(t) from 0 to 0.5
seconds sampled at
% Fs=1000Hz, A=0.8 and f=50 Hz;
A=0.8; f=50;
Fs=1000;
t_n=0:1/Fs:0.5;
x=A*cos(2*pi*f*t_n);

% 2) compute y_t and y_f as x filtered with h in the
time and in the
% frequency domain, respectively

y_t=conv(x,h);
Nfft=2^ceil(log2(length(x)+length(h)-1));
X=fft(x,Nfft); H=fft(h, Nfft);
Y=X.*H; y=ifft(Y);
y_f=y(1:length(x)+length(h)-1);

% 3) Plot x, y_t and y_f in three subplots
% in the time domain (in seconds)
% bonus: try to plot only the first 0.1 seconds of them
% hint: what does A(B>c) with A, B vectors and c scalar
do?
t_y=[0:length(y_t)-1]/Fs;
figure;
subplot(3,1,1);
plot(t_n(t_n<=0.1),x(t_n<=0.1)); xlabel('Time [s]');
ylabel('x(t_n)');
title('x(t_n)')
subplot(3,1,2);
plot(t_y(t_y<=0.1),y_t(t_y<=0.1)); xlabel('Time [s]');
ylabel('y_t(t_n)');
title('x filtered with h in the time domain')
subplot(3,1,3);
plot(t_y(t_y<=0.1),y_f(t_y<=0.1)); xlabel('Time [s]');
ylabel('y_f(t_n)');
title('x filtered with h in the frequency domain')
```


x(t_n)
x(t_n)
1
0.5
0
-0.5
-1
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
Time [s]
x filtered with h in the time domain
y_t(t_n)
4
2
0
-2
-4
Time [s]
x filtered with h in the frequency domain
y_f(t_n)
Time [s]