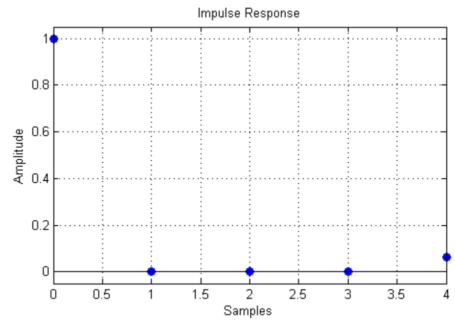
**Multimedia Signal Processing 1st Module**

12 /2/2015 – Exam

## Ex.1 (Pt.13)

An audio signal is sampled at 8kHz, the transmission channel presents an echo and its impulse response is show below: (the values are [1,0,0,0,1/16)

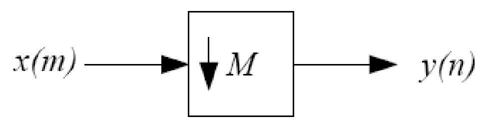



1. Define the z-transform of the transmission channel
2. Plot the zero poles diagram of the filter.
3. Find the exact position of zeros.
4. Find the exact gain/attenuation for a sinusoid at 1kHz
5. Find the exact gain/attenuation for a sinusoid at 2kHz
6. Define an IIR filter to completely remove the echo providing its difference equation.

## Ex.2 (Pt.9)

Describe the downsampling of an order of 4 of a signal: (where M=4).

1. If the spectrum of original signal extends from $-\pi/3$ to $\pi/3$ (in normalized frequencies) draw the final spectrum after downsampling (quoting both axes and indicating central frequencies for all the replicas)
2. If aliasing is present in the downsampled signal, suggest a way to avoid it (keeping the same downsampling rate).



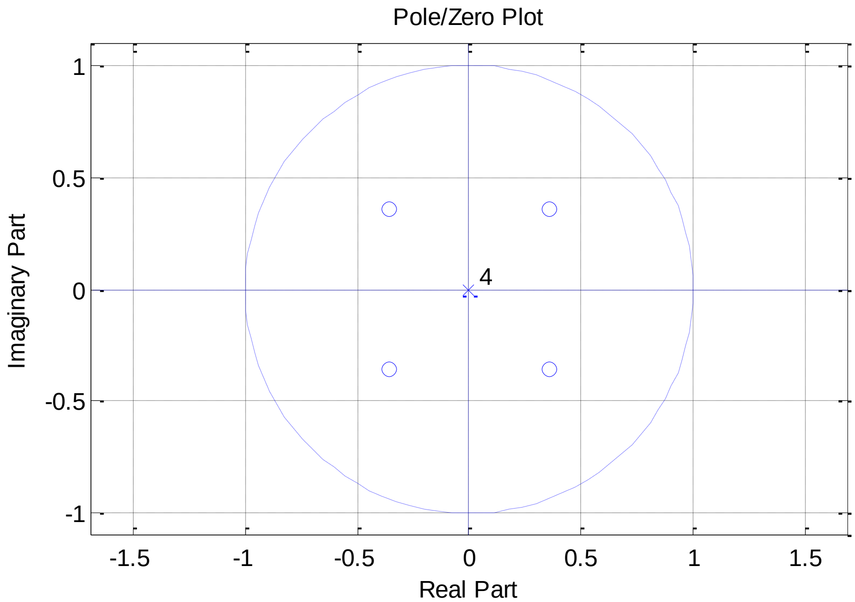

## Ex.3 (Pt. 11 – MATLAB code)

```
Given the filter h=[0,2,1] and the signal x=[4 3 2 6 4 3];
with time indices starting both from zero
1) compute y as the linear convolution of h and x in the time domain
   (you can use MATLAB functions)
2) compute y2 as the linear convolution of h and x in the frequency domain
   (you can use MATLAB functions)
3) Plot the squared error between y and y2
4) Plot the two signals y and y2 in the same plot
```

# Solutions

## Ex.1

The z-transform will be: $H(z)=1+\frac{1}{16}z^{-4}$




Since the signal is sampled at 8 kHz the sinusoid at 1kHz will correspond to the $\omega_1 = 1kHz\cdot\frac{2\pi}{8kHz}=\frac{\pi}{4}$ and the signal at 2kHz will correspond to the $\omega_2 = 2kHz\cdot\frac{2\pi}{8kHz}=\frac{\pi}{2}$

$$\left|H\left(z=e^{j\omega_1}=\frac{\sqrt{2}}{2}(1+j)\right)\right|=\left|1+\frac{1}{16}\left(\frac{\sqrt{2}}{2}(1+j)\right)^{-4}\right|=0.9375$$

$$\left|H\left(z=e^{j\omega_2}=j\right)\right|=\left|1+\frac{1}{16}(j)^{-4}\right|=1.0625$$

The IIR filter is a filter with a pole placed over every zero of H: $H_{IIR}(z)=\frac{1}{1+\frac{1}{16}z^{-4}}$

The difference equation will be: $y(n)=x(n)-\frac{1}{16}y(n-4)$

## Ex.2

The final spectrum will extend from $-M\frac{\pi}{3}\leftrightarrow M\frac{\pi}{3}$ i.e. $-4\frac{\pi}{3}\leftrightarrow 4\frac{\pi}{3}$, since it extends above π, aliasing will be present and it can be avoided applying a low pass filter to cut frequencies higher than $\frac{\pi}{4}$

## Ex.3

```
clc
clear all
close all

% Given the filter h=[0,2,1] and the signal x=[4 3 2 6 4 3];
% with time indices starting both from zero

%1) compute y as the linear convolution of h and x in the time domain
% (you can use MATLAB functions)
h=[0,2,1];
 x=[4 3 2 6 4 3];
y=conv(x,h);

%2) compute y2 as the linear convolution of h and x in the frequency domain
% (you can use MATLAB functions)

Nfft=length(h)+length(x)-1;

X=fft(x,Nfft);
H=fft(h,Nfft);
Y2=X.*H;
y2=ifft(Y2,Nfft);

% 3) Plot the squared error between y and y2

figure;
plot(0:Nfft-1,(y-y2).^2);
title('Squared error');
xlabel('n');
ylabel('|y-y_2|^2')
% 4) Plot the two signals y and y2 in the same plot

figure;
plot(0:Nfft-1,y);
hold on
plot(0:Nfft-1,y2,'r--');
hold off
title('Time vs Frequency domain linear convolution');
xlabel('n');
ylabel('y(n), y_2(n)')
legend('Time-domain convolution','Frequency-domain convolution');
```