Multimedia Signal Processing 1st Module

Fundamentals of Multimedia Signal Processing

23/1/2018

Ex.1 (Pt.14)

A signal in the time domain has the following expression:

$$x[n]=\left(\frac{3}{2}\right)^n \cos\left(\frac{\pi}{4}n\right)+\left(\frac{1}{2}\right)^n \sin\left(\frac{\pi}{2}n\right)$$

Using a FIR filter $h(n)$ we want to completely remove just the unstable components preserving the stable ones.

Define the z transform of the filter [5pts].
Plot the zeros-poles plot of the filter [4pts].
Provide the expression $H(\omega)$ of the filter response at different normalized frequencies ω and the expression of its magnitude $|H(\omega)|$ [5pts].

Ex.2 (Pt.9)

A sinusoid at 10 KHz is sampled at $40000\frac{samples}{s}$ and then upsampled of an order of 4 (three zeros added every sample) .

1. Represent the spectrum in normalized frequencies before and after upsampling.
2. Define an ideal low pass filter (normalized and real cut frequency) to remove artifacts in the upsampled signal.

Ex.3 (Pt. 11 – MATLAB code)

Given the signal x(t)= A_1 cos(2 pi f_1 t) + A_2 cos(2 pi f_2 t)

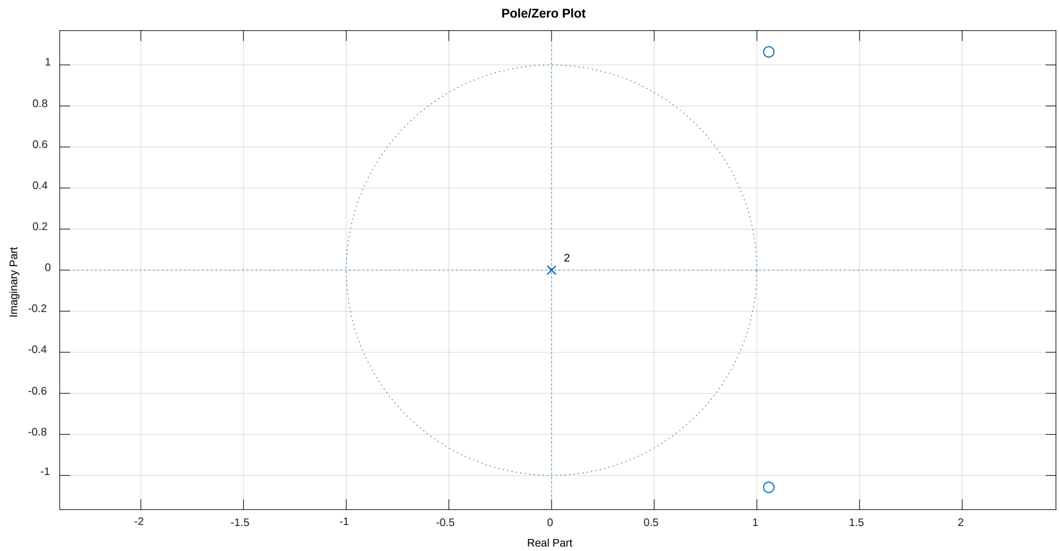
1. Create the signal x(n) as x(t) with t from 0 to 0.5 seconds sampled at Fs=8000 Hz. Use A_1=0.7, A_2=0.5, f_1=1800 Hz, f_2=3600 Hz.
2. Create the signal y(n) by re-sampling x(n) to 6000 Hz, without using the MATLAB functions for automatic re-sampling
3. Plot the magnitude of the 2048-point DFTs of the original and resampled signal (same plot, normalized frequency)

Solutions

Ex.1

The Z transform of the signal will be:

$$H(z)=\left(1-\frac{3}{2}\cdot e^{j\pi/4}z^{-1}\right)\left(1-\frac{3}{2}\cdot e^{-j\pi/4}z^{-1}\right)=1-3\frac{\sqrt{2}}{2}z^{-1}+\frac{9}{4}z^{-2}$$

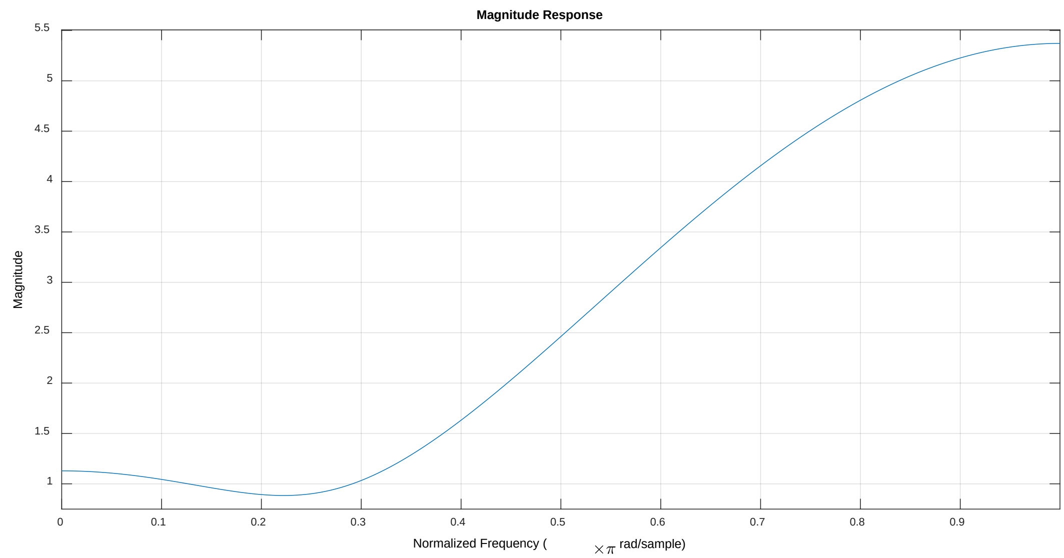


$$H(\omega)=1-3\frac{\sqrt{2}}{2}e^{-j\omega}+\frac{9}{4}e^{-j2\omega}$$

$$\left|H(\omega)\right|=\sqrt{\left(1-3\frac{\sqrt{2}}{2}e^{-j\omega}+\frac{9}{4}e^{-j2\omega}\right)\left(1-3\frac{\sqrt{2}}{2}e^{j\omega}+\frac{9}{4}e^{j2\omega}\right)}=$$

$$=\sqrt{1+\frac{18}{4}+\frac{81}{16}-\left(\frac{3}{2}\sqrt{2}+\frac{27}{8}\sqrt{2}\right)\cdot 2\cos(\omega)+\frac{9}{4}\cdot 2\cos(2\omega)}=$$

$$\sqrt{\frac{169}{16}-\frac{39}{4}\sqrt{2}\cos(\omega)+\frac{9}{2}\cos(2\omega)}$$



Ex.2

In normalized frequencies, sampling the sinusoid there will be an impulse at $\pm\frac{10KHz}{40KHz}2\pi = \pm\frac{\pi}{2}$

After upsampling it will move to $\pm\frac{1}{4}\cdot\frac{\pi}{2}$ and 3 further impulse will appear in normalized frequencies within the $-\pi \ldots \pi$ range: $\pm\frac{1}{4}\cdot\frac{3}{2}\pi$, $\pm\frac{1}{4}\cdot\frac{5}{2}\pi$ and $\pm\frac{1}{4}\cdot\frac{7}{2}\pi$

To remove artifacts due to upsampling an ideal low pass filter should have a cut-off normalized frequency of $\frac{\pi}{4}$.

Ex.3

```
clear all
close all
clc

% Given the signal x=A1*cos(f1*2*pi*t)+A2*cos(f2*2*pi*t);

% 1. Create the signal x with A1=0.7, f1= 1800Hz, A2=0.5,
f2=3600Hz
%    and t from 0 to 0.5 seconds sampled at Fs=8000Hz
Fs=8000;
t=0:1/Fs:0.5;
A1=0.7; f1=1800; A2=0.5; f2=3600;
x=A1*cos(2*pi*f1*t)+A2*cos(2*pi*f2*t);

% 2. Create the signal y by re-sampling x to 6000 Hz
%    (without using the MATLAB functions for automatic re-
sampling)
```

```
q=6000/Fs;
[L, M]=rat(q);

y_int=zeros(1,length(x)*L);
y_int(1:L:end)=x;
y_int=filter(fir1(63,1/L),1,y_int);
y_dec=filter(M*fir1(63,1/M),1,y_int);
y=y_dec(1:M:end);

% 3. Plot the magnitude of the 2048-point DFTs of the
original and the resampled
% signal (same plot, normalized frequency);
Nfft=2048;
X=fft(x, Nfft);
Y=fft(y, Nfft);
w_norm=linspace(0,2,Nfft);
figure;
plot(w_norm, abs(X)); hold on
plot(w_norm, abs(Y)); hold off
xlabel('normalized \omega');
ylabel('|X(k)|, |Y(k)|');
title('Magnitude of the DFT');
legend('Original Signal', 'Resampled signal');
```