

MultimediaSignal Processing 1st Module and Fundamentals of Multimedia Signal Processing

date: 22nd July 2019

Ex.1 (Pt.13)

A signal $x(t)$ is sampled at 10kHz, and we want to find the relative magnitude of its components at 1kHz with respect to the components at 4kHz but we cannot use the FFT.

Therefore we decide to filter the signal at 1kHz and 4kHz multiplying it with a real sinusoid at 1kHz and at 4kHz respectively.

1. [3 pts] Depict the pole zero plots for the two sinusoids.
2. [6 pts] Define the difference equations both for the first and the second sinusoid.
3. [3 pts] Describe how we shall apply the previously defined filters in order to measure the relative intensity of the input signal at 1kHz with respect to 4kHz.
4. [1 pts] What is a weakness of the proposed system with respect to the Fourier Transform?

Ex.2 (Pt.9)

An audio signal is sampled at 30kHz and we want to change its sample rate without losses at 40kHz.

1. [3 pts] Describe the procedure in order to increase the signal sample rate.
2. [6 pts] Provide a description of the ideal filter that you would use in order to filter the signal and describe a real filter that could approximate the ideal one.

Ex.3 (Pt. 11 – MATLAB code)

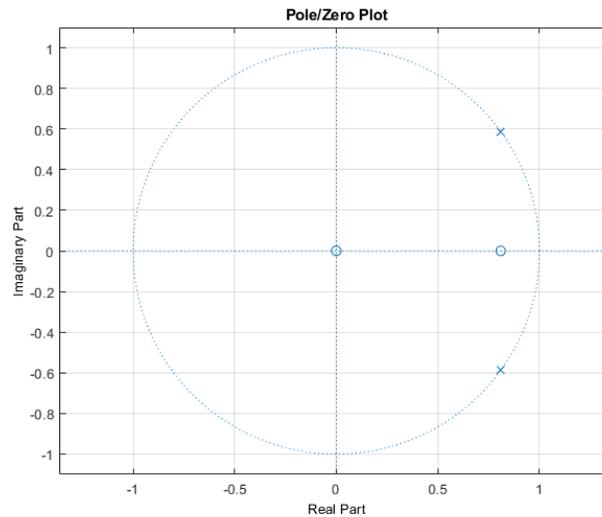
1. [1pt.] Load the wave file 'TomsDiner.wav' into x
2. [2pt.] Compute y by filtering x with the maximum phase filter given by
 $b=[1, -1.98, 1.77, -0.17, 0.21, 0.34];$
 $a=[1, 0.08, 0.40, 0.27];$
3. [4 pt.] compute the allpass-minimum-phase decomposition of $B(z)/A(z)$
4. [2 pt.] Restore $x_{\hat{}}$ by filtering y with the inverse filter exploiting the minimum-phase
5. [2 pt.] Plot and compare x, y and $x_{\hat{}}$ in the time domain, as three subplots (time domain in seconds)

Solutions

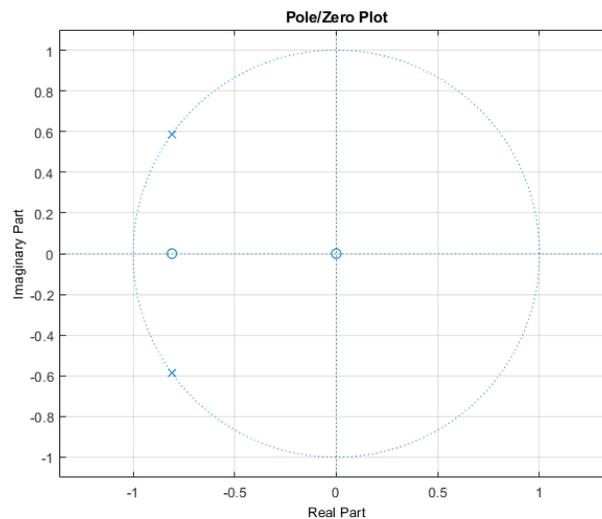
Ex.1

In order to implement the first sinusoid we have to set two conjugate poles at $\theta_1 = \pm \frac{1kHz}{10kHz} \cdot 2\pi$ while

for the second one the $\theta_2 = \pm \frac{4kHz}{10kHz} \cdot 2\pi$.



and



The z transforms of the two filters will be:

$$H_1(z) = \frac{1}{2} \left(\frac{1}{1 - e^{-j\pi/5} z^{-1}} + \frac{1}{1 - e^{j\pi/5} z^{-1}} \right) = \frac{1 - \cos(\pi/5) z^{-1}}{1 - 2 \cos\left(\frac{1}{5}\pi\right) z^{-1} + z^{-2}} \text{ and}$$

$$H_2(z) = \frac{1}{2} \left(\frac{1}{1 - e^{-j4\pi/5} z^{-1}} + \frac{1}{1 - e^{j4\pi/5} z^{-1}} \right) = \frac{1 - \cos(4\pi/5) z^{-1}}{1 - 2 \cos\left(\frac{4}{5}\pi\right) z^{-1} + z^{-2}}.$$

The difference equations will then become:

$$y_1(n) = x(n) - \cos(\pi/5)x(n-1) + 2\cos(\pi/5)y_1(n-1) - y_1(n-2) \quad \text{and}$$

$$y_2(n) = x(n) - \cos\left(\frac{4}{5}\pi\right)x(n-1) + 2\cos\left(\frac{4}{5}\pi\right)y_2(n-1) - y_2(n-2)$$

In order to apply the filters we have to take a portion of the input signal and process it with the two filters (the difference equations above).

To avoid polarizations in the measure we should take a set of samples multiple of the wavelengths (in term of samples) of both the sinusoids.

The ratio of the two magnitudes of the outputs will give the result.

The weakness with respect to the Fourier Transform is related to the fact that we are not using the phase as in the Discrete Fourier Transform, so, signal components with the same amplitude and frequency but different phases will give different results.

Ex.2

In order to change the sample rate according to the requests, a possible solution is to upsample of an order of 4 and downsample of an order 3.

In order to remove the spurious spectral replicas an ideal low pass filter is requested between upsampling and downsampling with a cut-off frequency at $\frac{\pi}{4}$. Further detail can be found in the textbook.

Ex.3

```

%% 1.[1pt.] Load the wave file 'TomsDiner.wav' into x
[x,Fs]=audioread('TomsDiner.wav');
%% 2.[2pt.] Compute y by filtering x with the maximum phase
filter given by
b=[1, -1.98, 1.77, -0.17, 0.21, 0.34];
a=[1, 0.08, 0.40 ,0.27];

y=filter(b,a,x);
%% 3. [3 pt.] compute the allpass-minimum phase decomposition of
B(z)/A(z)

z=roots(b);
p=roots(a);

z_allpass=z(abs(z)>=1);% maximum phase zeros outside from the
    % unit circle
p_allpass=1./conj(z_allpass);% poles of the allpass filter
    % derived from maximum phase zeros of the original filter
z_mp=[z(abs(z)<1); p_allpass];% minimum phase version of the
    % zeros of the original filter
p_mp=p;

b_mp=poly(z_mp);
b_ap=poly(z_ap);
a_mp=poly(p_mp);
a_ap=poly(p_ap);

```

```

%% 4 [2 pt.] Restore x_hat by filtering y with the inverse
filter exploiting the minimum-phase
... maximum phase decomposition

x_hat=filter(a_mp,b_mp,y);% zeros and poles are reversed in
      % order to obtain the reconstruction filter

%% 5. [2pt.] Plot and compare x, y and x_hat in the time domain,
as three subplots (time domain in seconds)
t=(0:length(x)-1)/Fs;
figure;
subplot(3,1,1);
plot(t,x)
xlabel('Time [s]')
ylabel('x(n)')
subplot(3,1,2);
plot(t,y)
xlabel('Time [s]')
ylabel('y(n)')
subplot(3,1,3);
plot(t,x_hat)
xlabel('Time [s]')
ylabel('x_{hat}(n)')

```