Multimedia Signal Processing 1st Module and Fundamentals of Multimedia Signal Processing

Date: June 19th, 2023

Ex.1 (Pt.13)

A digital filter has the following difference equation:

$$y[n]=\frac{1}{9}\left(6\sqrt{2}y[n-1]-4y[n-2]+4x[n]-6\sqrt{2}x[n-1]+9x[n-2]\right)$$

[4 pts] What is the transfer function $H(z)$ of such a filter?

Draw its pole-zero plot. What kind of filter is it?

[4 pts] Draw its approximated amplitude and phase response in normalized frequencies.

[4 pts] A continuous signal $x(t)=5\cos(2\pi 1000t)+7\cos(2\pi 4000t)$ is sampled at 8kHz and then filtered with the previously defined filter. What will be output discrete signal $y[n]$?

Ex.2 (Pt.8)

The continuous signal $x(t)=6\sin(\omega_1 t)-6\sin(\omega_2 t)$, where $\omega_1=2\pi\cdot 75kHz$ and $\omega_2=2\pi\cdot 125kHz$, is sampled at $600kHz$.

[2 pts] Analyze the signal in the frequency domain and depict the signal in the $0-2\pi$ pulsations range.

The signal is then downsampled without any lowpass filter of an order of M=3.

[3 pts] what will be the output signal? Describe the reason for such an output.

[3 pts] provide a suitable discrete filter to remove/attenuate any undesired effect ad depict the output signal.

CONTINUES ON THE BACK

Ex.3 (Pt.11)

1) [3 pt] Define the sinusoidal signal x(t), which is composed of two contributions at f1 = 2 KHz and f2 = 1.25 KHz. The signal is sampled such that it repeats periodically every 40 samples, and it is defined over 400 samples. Plot the signal as a function of the time [seconds].
2) [3 pt] We want to enhance the contribution at f1 and to lower down that of f2. To do so, we are given an LTI system H_1(z), causal and stable, with real coefficients and minimum phase, with this finite-difference equation:

$$y(n) = a*x(n) + b*x(n-1) + c*x(n-2) + d*y(n-1) + e*y(n-2).$$

   - Choose the values of parameters a, b, c, d, e such that h_1(n=0) = 1.5.
   - Filter the signal x(n) with H_1(z), defining the signal y(n).
3) [2.5 pt] The signal y(n) is summed to a periodic sequence = [1, 0, -1, 0, 1, 0, -1, 0…]. Define the output signal as z(n). Design the filter H_2(z), which is defined by the same finite-difference equation of H_1(z) but with different coefficients, to maintain only the signal y(n) from the signal z(n). Filter the signal z(n), defining the signal w(n).
4) [2.5 pt] Compute the DFTs of the signals x(n), y(n), z(n), w(n) and plot their absolute values as a function of the normalized frequency axis. Comment on the position/amplitude of the peaks you expect to see for every signal.
   - Which are the differences between the DFTs of y(n) and z(n)?
   - Which are the differences between the DFTs of y(n) and w(n)?

   Motivate your answers.

Solutions

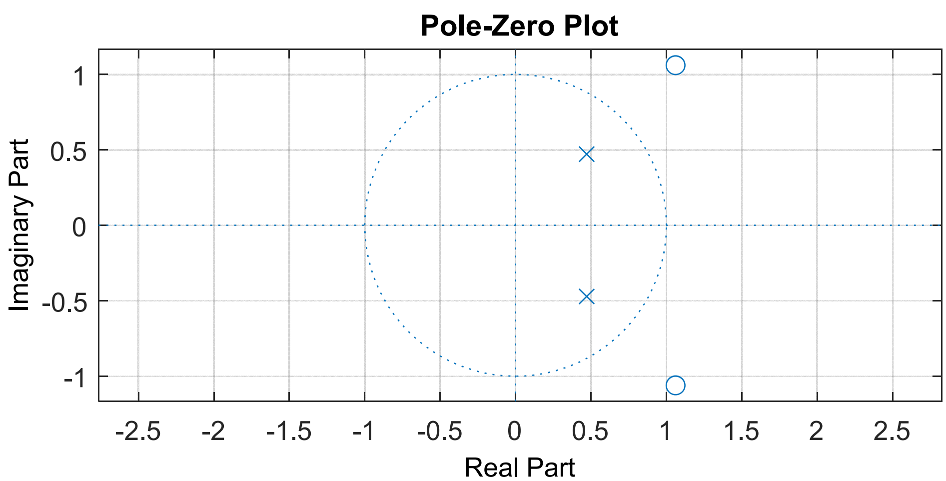
Ex.1

The filter has the following z transform associated to the difference equation:

$$H(z)=\frac{4-6\sqrt{2}z^{-1}+9z^{-2}}{9-6\sqrt{2}z^{-1}+4z^{-2}}$$

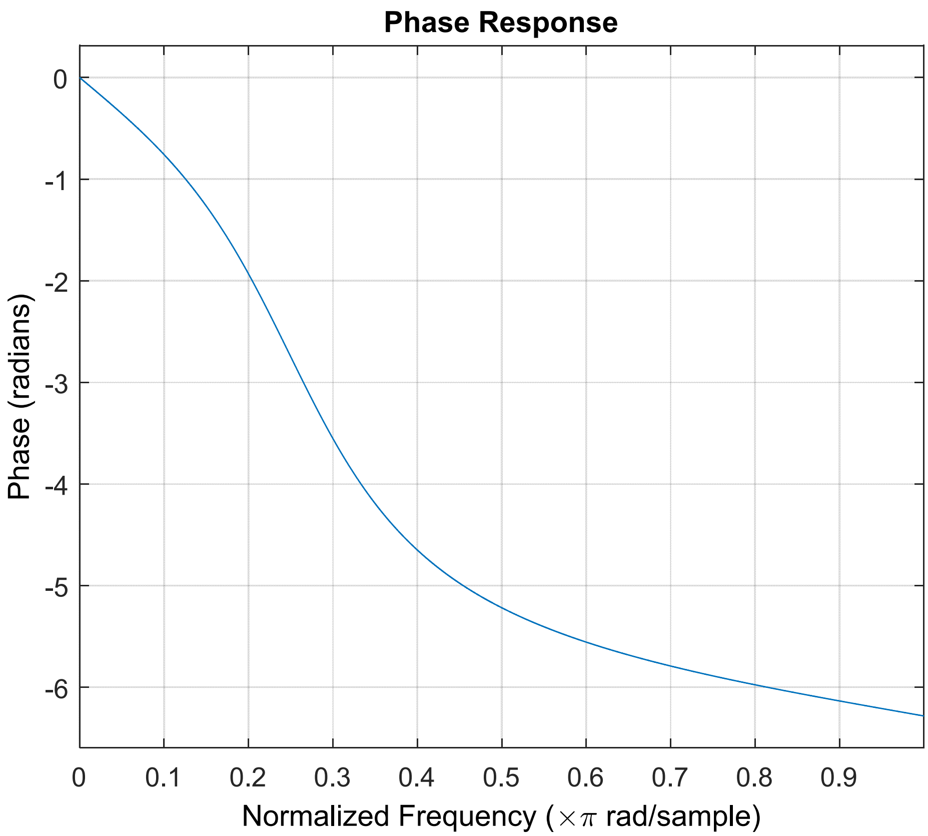
It is an all-pass filter, it is also a causal, stable and maximum phase filter.

Its zero-pole plot is the following:



Since it is an all-pass filter its amplitude is constant across all frequencies. In this case the amplitude is always 1.

The phase has the following behavior:



By construction we can say that the couples of conjugate zeros and conjugate poles are at the normalized frequency $\frac{\pi}{4}$ indicating that at that frequency we will have a phase rotation of π rads.

The sampled signal will have the following discrete representation:

$$x[n] = 5\cos\left(\frac{\pi}{4}n\right) + 7\cos(\pi n)$$

The first component at 1kHz will then undergo a phase rotation of 180° while the second component, at the Nyquist frequency, will preserve its phase. The output will then be:

$$y[n] = -5\cos\left(\frac{\pi}{4}n\right) + 7\cos(\pi n).$$

Ex.2

The sampled signal will present impulses at the following normalized frequencies:

$\bar{\omega}_1 = \frac{2\pi 75kHz}{600kHz} = \frac{\pi}{4}$ and $\bar{\omega}_2 = \frac{2\pi 125kHz}{600kHz} = \frac{5\pi}{12}$, in the frequency domain it can be represented as:

$$X(\omega) = 3j \cdot \delta(\omega - \bar{\omega}_1) - 3j \cdot \delta\left(\omega - (2\pi - \bar{\omega}_1)\right) - 3j \cdot \delta(\omega - \bar{\omega}_2) + 3j \cdot \delta\left(\omega - (2\pi - \bar{\omega}_2)\right)$$

With the simple downsampling I will obtain:

$\bar{\bar{\omega}}_1 = 3\bar{\omega}_1 = \frac{3}{4}\pi$ and $\bar{\bar{\omega}}_2 = 3\bar{\omega}_2 = \frac{5}{4}\pi$

So: $X_d(\omega) = \frac{1}{3}\left(3j\cdot\delta\left(\omega-\bar{\bar{\omega}}_1\right) - 3j\cdot\delta\left(\omega-\left(2\pi-\bar{\bar{\omega}}_1\right)\right) - 3j\cdot\delta\left(\omega-\bar{\bar{\omega}}_2\right) + 3j\cdot\delta\left(\omega-\left(2\pi-\bar{\bar{\omega}}_2\right)\right)\right)$

But $2\pi-\bar{\bar{\omega}}_1 = 2\pi-\frac{3}{4}\pi = \frac{5}{4}\pi = \bar{\bar{\omega}}_2$ and $2\pi-\bar{\bar{\omega}}_2 = 2\pi-\frac{5}{4}\pi = \frac{3}{4}\pi = \bar{\bar{\omega}}_1$ so, due to the downsampling without antialising filter, the two sinusoids, due to the opposite sign, will cancel each other out.

Introducing a low pass antialising filter with a cut-off frequency at $\omega_{cut-off} = \frac{\pi}{3}$ the component at $\bar{\omega}_2 = \frac{5\pi}{12}$ will be removed (or, at least, attenuated by a non ideal filter).

Ex.3

```
close all
clearvars
clc

%% 1

% [3 pt] Define the sinusoidal signal x, which is composed of
% two contributions at f1 = 2 KHz and f2 = 1.25 KHz.
% The signal is sampled such that it repeats periodically every 40
samples,
% and it is defined over 400 samples.
% Plot the signal as a function of the time [seconds].

N = 400;
P_samples = 40;
f1 = 2e3;
f2 = 1.25e3;

% The overall period is the least common multiple of the two periods -->
% P1 = 1/2e3 --> 5e-4 seconds
% P2 = 1/1.25e3 --> 8e-4 seconds
% P = 40e-4 seconds = 4 ms
% Knowing P, we can find Fs = P_samples / P [secs] = 40/4e-3 = 1e4 = 10KHz

Fs = 10e3;
time_axis = 0:1/Fs:(N-1)/Fs;

% define the signal
x = cos(2*pi*f1*time_axis) + cos(2*pi*f2*time_axis);

figure;
plot(time_axis, x);
grid;
```

```
title('x(t)');

%% 2

% [3 pt] We want to enhance the contribution at f1 and to lower down that
of f2.
% To do so, we are given an LTI system H_1(z), causal and stable,
% with real coefficients and minimum phase, with this finite-difference
equation:
% y(n) = a*x(n) + b*x(n-1) + c*x(n-2) + d*y(n-1) + e*y(n-2).
% Choose the values of parameters a, b, c, d, e such that h_1(n=0) = 1.5.
% Filter the signal x(n) with H_1(z), defining the signal y(n).

f1_n = f1/Fs; % --> 1/5
f2_n = f2/Fs; % --> 1/8

% The filter structure is the following one:
% B = [a, b, c];
% A = [1, -d, -e];

% Since the filter has real coefficients, both numerator and denominator
% of the filter should present the following structure:
% B(z) = gain * (1 - 2*rho_z*cos(theta_z)*z^{-1} + rho_z^2 * z^{-2}) -->
% numerator
% A(z) = 1 - 2*rho_p*cos(theta_p)*z^{-1} + rho_p^2 * z^{-2}) -->
denominator

% for enhancing f1, we need to introduce a pole with theta_p = 2*pi*f1_n =
% 2*pi*1/5
% for lowering f2, we need to introduce a zero with theta_z = 2*pi*f2_n =
% 2*pi*1/8 = pi/4

% we can choose the remaining parameters as we want, but we need to keep
% h(n=0) = 1.5 and build a stable causal filter with minimum phase -->
% gain = 1.5 such to have a = 1.5
% poles and zeros inside the circle.

% Possible solution:
rho_z = 0.9;
theta_z = pi/4;
rho_p = 0.9;
theta_p = 2*pi*1/5;
B_1 = 1.5*[1, -2*rho_z*cos(theta_z), rho_z^2];
A_1 = [1, -2*rho_p*cos(theta_p), rho_p^2];

% filter behaviour (not required)
[H, omega] = freqz(B_1, A_1, 2048, 'whole');
figure,
plot(omega./(2*pi), abs(H));
title('|DTFT| of the filter H_1(f)');
grid;

% filter the signal x
y = filter(B_1, A_1, x);

%% 3.
```

```
% [2.5 pt] The signal y(n) is summed to a periodic sequence = [1, 0, -1,
0, 1, 0, -1, 0,Ä¶].
% Define the output signal as z(n). Design the filter H_2(z),
% which is defined by the same finite-difference equation of H_1(z)
% but with different coefficients, to maintain only the signal y(n) from
% the signal z(n). Filter the signal z(n), defining the signal w(n).

% Notice that the sequence is a sinusoidal sequence with period = 4.
f3_n = 1/4;
z = y + cos(2*pi*f3_n*(0:N-1));

% We have to design a notch filter in theta = 2*pi*f3_n = pi/2
rho_z = 1;
rho_p = 0.95;
theta = 2*pi*f3_n;
B_2 = [1, -2*rho_z*cos(theta), rho_z^2];
A_2 = [1, -2*rho_p*cos(theta), rho_p^2];

% filter behaviour (not required)
[H, omega] = freqz(B_2, A_2, 2048, 'whole');
figure,
plot(omega./(2*pi), abs(H));
title('|DTFT| of the notch filter H_2(f)');
grid;

% filter the signal z
w = filter(B_2, A_2, z);

%% 4.

% [2.5 pt] Compute the DFTs of the signals x(n), y(n), z(n), w(n) and
% plot their absolute values as a function of the normalized frequency
axis.
% Comment on the position/amplitude of the peaks you expect to see for
% every signal.
% Which are the differences between the DFTs of y(n) and z(n)?
% Which are the differences between the DFTs of y(n) and w(n)?
% Motivate your answers.

X = fft(x);
Y = fft(y);
Z = fft(z);
W = fft(w);

freq_axis = 0:1/N:1 - 1/N;

figure;
stem(freq_axis, abs(X));
title('Absolute value of the DFT of the original signal x(n)');
grid;
% we expect to see 4 peaks related to the two cosinusoidal signals, with
% the same amplitude. peaks are centered in 1/5, 1/8, 4/5 and 7/8.

figure;
stem(freq_axis, abs(Y));
title('Absolute value of the DFT of the signal y(n)');
grid;
% The gain of the peaks has been modified by the filter H_1(z), which
```

```
% enhances the frequency f1 and attenuates f2.

figure;
stem(freq_axis, abs(Z));
title('Absolute value of the DFT of the signal z(n)');
grid;
% The signal z contains other two peaks in the frequencies 0.25 and 0.75.

figure;
stem(freq_axis, abs(W));
title('Absolute value of the DFT of the signal w(n)');
grid;
% The signal w has a DFT which resembles that of y, because the filter
% H_2(z) is a notch filter and removes the frequency component at 0.25.

% The differences of the DFT of y and z are due to the peaks in 0.25 and
% 0.75, which are present in z and not in y. The rest is the same.

% Apart from small errors, there are no differences in the DFTs of y and
w,
% because w is obtained by filtering z with a notch filter. The notch
% filter removes the frequency components at 0.25 and 0.75, leaving almost
% untouched the rest of the spectrum.
```