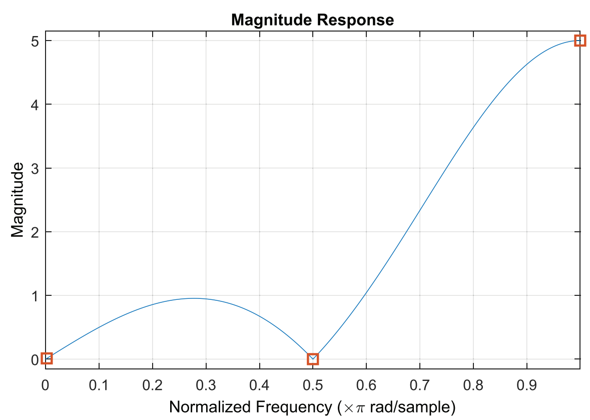
Multimedia Signal Processing 1st Module and Fundamentals of Multimedia Signal Processing

date: July 22nd, 2024

Ex.1 (Pt.12)

We need to design a digital filter with the response (Magnitude) represented in the following picture. In particular, we must honor the filter values at the following normalized frequencies represented by the red squares: $\left|H(\omega=0)\right|=0$, $\left|H\left(\omega=\frac{\pi}{2}\right)\right|=0$ e $\left|H(\omega=\pi)\right|=5$.



[5 pts.] Provide a possible z-transform of the filter and its pole-zero plot.

[3 pts.] Provide the block diagram for the implementation of the proposed digital filter.

[4 pts.] Depict an approximate representation of the phase of the proposed filter and find the filter output with correct amplitude and phase when the input is a sinusoid at the Nyquist frequency: $x[n]=\cos(\pi n)$.

Ex.2 (Pt.10)

The signal $x(n)$, $x(n)=\{\ldots,0,0,\underline{3},3,1,-1,2,0,0,\ldots\}$, where $\underline{1}$ represents $x(0)$, is applied as the input to the following system:

$$x(n) \longrightarrow \boxed{\uparrow 2} \longrightarrow \boxed{H(z)} \longrightarrow \boxed{\downarrow 3} \longrightarrow y(n)$$

If the impulse response $h(n)$ is given by $h(n)=\{\underline{1},2,1\}$, the what is the output signal $y(n)$?

Using the same impulse response, what will be the output in this case?

$$x(n) \longrightarrow \boxed{\uparrow 2} \longrightarrow \boxed{H(z^2)} \longrightarrow y(n)$$

CONTINUES ON THE BACK

Ex.3 (Pt.12) To be solved writing the MATLAB code on the sheet.

1) [2 pt] Define a sinusoidal signal s(n) = cos(2*pi*f0*n), n = [0, N-1], N = 500, such that s(n=0) = s(n=10).
2) [3 pt] You are given a filter H(z) = B(z) / A(z), where:
   - B(z) = (1 - z_0 z^-1)*(1 - a*cos(pi/5)z^-1 + b*z^-2)
   - A(z) = (1 - c*cos(theta)z^-1 + d*z^-2).
   - Choose the values of z_0, a, b, c, theta, d such to define a notch filter, which should be causal and stable with real-coefficients.
   - Plot the absolute value of the frequency response of the filter vs the normalized frequency axis, considering 1024 samples.
3) [3 pt]
   - Compute the linear convolution between B(z) and ten periods of s(n), considering only the first samples of the result equal to ten periods of s(n).
   - Compute also the circular convolution (exploiting the DFT properties) between s(n) and B(z), considering a number of samples equal to ten periods of s(n).
   - Plot the two convolution results in the same figure using the function 'stem', together with the first samples of the signal s(n) (consider the same length of the filtered signals).
   - What do you expect to find as resulting signals? Are there any differences between the results? If yes, in which samples? Motivate your answer.
4) [4 pt] We want to define the all-pass transfer function version of the filter H(z), namely Hap(z), which should be causal and stable with real coefficients.
   - By directly apply the all-pass conversion to the filter H(z), are we defining a stable filter? Why?
   - Modify the values of the roots (you choose the values) such that the all-pass filter satisfies the requirements and define Hap(z).
   - Filter the signal s(n) with Hap(z), defining s_f(n).
   - Plot the amplitudes of the DFT of s(n) and of the DFT of s_f(n) in the same figure. What do you expect to find, apart some small deviations?

Solutions

Ex.1

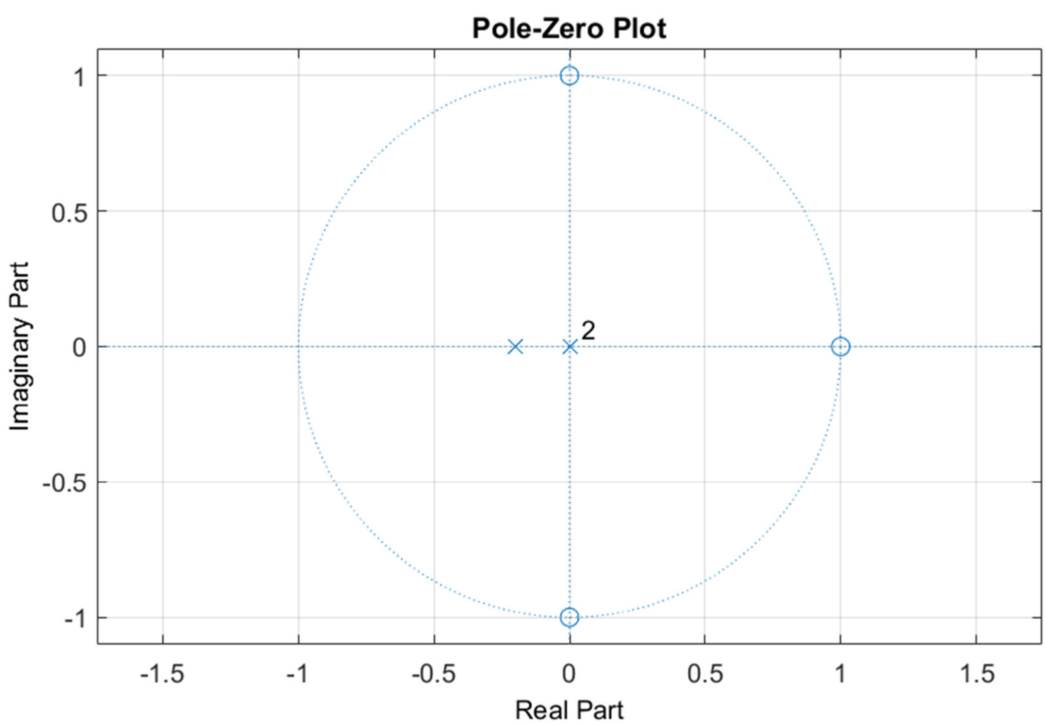
A possible implementation is a filter with 3 zeros at frequencies $\omega = 0, -\frac{\pi}{2}, \frac{\pi}{2}$ and a pole at $\omega = \pi$.

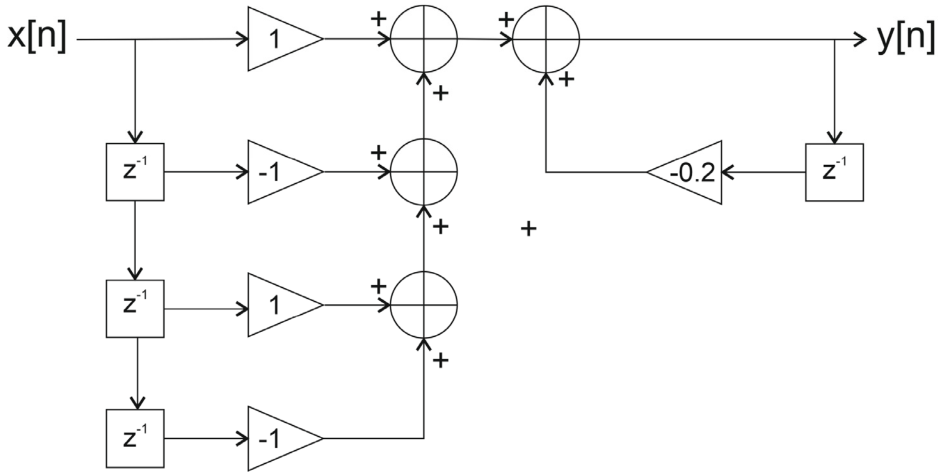
Since the output at the 3 frequencies $\omega = 0, -\frac{\pi}{2}, \frac{\pi}{2}$ must be 0, the the 3 zeros are exactly on the unit circle $|z| = 1$, while, the pole can be placed on the real axis between 0 and -1. The exact position of the pole can be determined in order to satisfy the constraints:

$$H(z) = \frac{(1 - z^{-1})(1 + z^{-2})}{1 + az^{-1}}$$

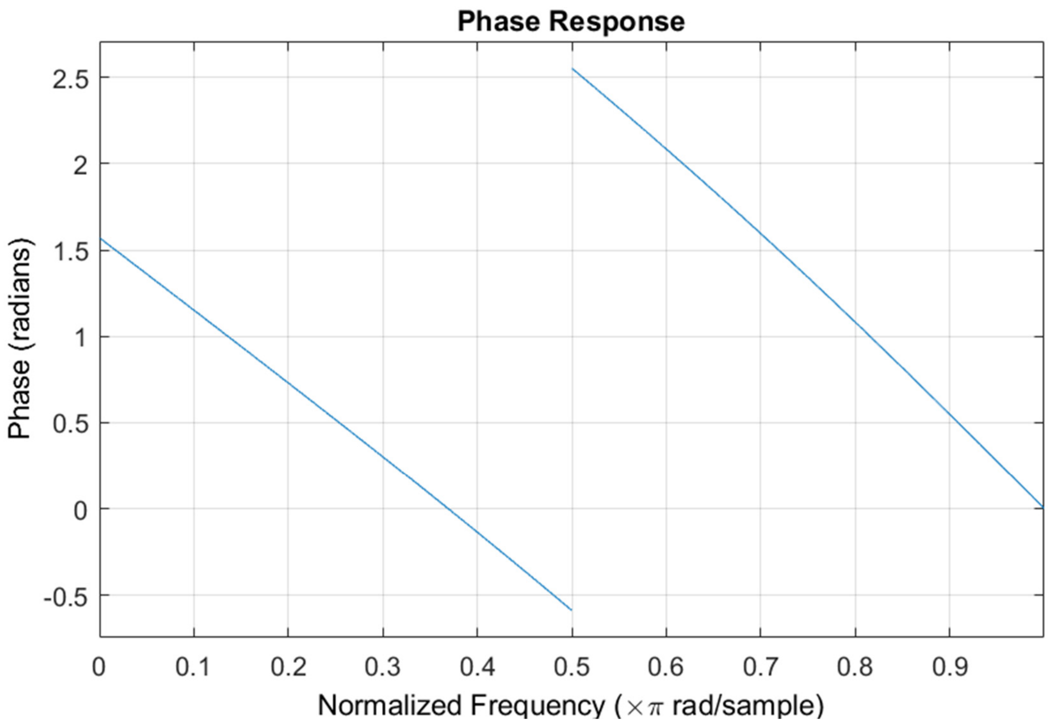
Forcing $|H(z = -1)| = 5$ means that $|H(z = -1)| = \frac{(1-(-1))(1+(1))}{1-a} = 5 \rightarrow a = \frac{1}{5}$.

Then $H(z) = \frac{1 - z^{-1} + z^{-2} - z^{-3}}{1 + 0.2z^{-1}}$





The phase will be:



Ex.2

The results for upsampling, filtering and, then, downsampling are

$$x_u[n] = \{3,0,3,0,1,0,-1,0,2,0\}$$
$$x_f[n] = x_u * \{1,2,1\} = \{3,6,6,6,4,2,0,-2,1,4,2,0\}$$
$$x_d[n] = \{3,6,0,4\}$$

In the second case:

$$x'_f[n] = x_u * \{1,0,2,0,1,0\} = \{3,0,9,0,10,0,4,0,1,0,3,0,2,0,0\}$$

Ex.3 (MATLAB CODE)

```
close all
clearvars
clc

%% 1.

% [2 pt] Define a sinusoidal signal s(n) = cos(2*pi*f0*n), n = [0, N-1],
% N = 500, such that s(n=0) = s(n=10).

period = 10;
f0 = 1/period;
N = 500;
n = 0:N-1;
s = cos(2*pi*f0*n);

%% 2.

% [3 pt] You are given a filter H(z) = B(z) / A(z), where:
% B(z) = (1 - z_0 z^-1)*(1 - a*cos(pi/5)z^-1 + b*z^-2).
% A(z) = (1 - c*cos(theta)z^-1 + d*z^-2).
% Choose the values of z_0, a, b, c, theta, d such to define a notch
filter,
% which should be causal and stable with real-coefficients.
% Plot the absolute value of the frequency response of the
% filter vs the normalized frequency axis, considering 1024 samples.

% there's no constraint on z_0, we can choose a generic real value
z_0 = 2;
% the filter should be a notch.
% the zeros are on the unit circle --> a = 2*rho_z, with rho_z = 1
a = 2;
b = 1;
% the poles must be inside the unit circle, but with the same phase of the
% zeros --> theta = pi/5
theta = pi/5;
c = 2*0.95;
d = 0.95^2;

B = conv([1, -z_0], [1, -a*cos(pi/5), b]);
A = [1, -c*cos(theta), d];

% frequency response of the filter
[H_ap, omega] = freqz(B, A, 1024, 'whole');
figure,
plot(omega./(2*pi), abs(H_ap));
title('|DTFT| of the filter H(z)');
grid;
xlabel('f [norm]');

%% 4.

% [3 pt] Compute the linear convolution between B(z) and ten periods of
```

```
% the signal s, considering only the first samples of the result exactly
equal
% to the value of 10 periods of s(n). Then, compute also the circular
convolution
% (exploiting the fft properties)
% between s(n) and B(z), considering an amount of samples equal to 10
% periods of s(n).
% Plot the two convolution results in the same figure using the function
% 'stem', together with the first samples of the signal s(n) (consider
% the same length of the filtered signals).
% What do you expect to find as resulting signals? Are there any
% differences between the results? If yes, in which samples? Motivate your
answer.

linear_conv = conv(s(1:10*period), B);
linear_conv = linear_conv(1:10*period);

S = fft(s(1:10*period));
F_B = fft(B, 10*period);

cyclic_conv = ifft(S.*F_B);

figure,
stem(s(1:10*period));
hold on;
stem(linear_conv, '--');
stem(cyclic_conv, '-.')
legend('s(n)', 'linear convolution result', 'cyclic convolution result');

% We expect to find a signal which is = 0, since the filter B(z) has zeros
% exactly in the frequency f0 of the signal.
% The two results differ only in the first 3 samples, which correspond to
% the length (filter B(z)) - 1. This is due to periodic artifacts
% of the cyclic convolution between the two signals. This
% consideration is the rationale behind the overlap and save method.

%% 3.

% [4 pt] We want to define the all-pass transfer function version of the
% filter H(z), namely Hap(z), which should be causal and stable with
% real coefficients. By directly apply the all-pass conversion to the
% filter H(z), are we defining a stable filter? Why?
% Modify the values of the roots (you choose the values) such that the
% all-pass filter satisfies the requirements.
% Filter the signal s(n) with Hap(z), defining s_f(n).
% Plot the amplitudes of the DFT of s(n) and of the DFT of s_f(n)
% in the same figure. What do you expect to find, apart some small
deviations?

% the zeros are on the unit circle --> to obtain poles that are inside the
% circle, let's move them out of the circle.
% you should make the same operation over the zero z_0 if you selected it
% to be inside the circle.

% let's define B(z) again with new values
a = 2*1.1;
b = 1.1^2;
B = conv([1, -z_0], [1, -a*cos(pi/5), b]);
```

```
% define the all-pass transfer function
A_tilde = fliplr(conj(A));
B_tilde = fliplr(conj(B));

Bap = conv(B, A_tilde);
Aap = conv(A, B_tilde);

% normalize everything by Aap(1)
Bap = Bap / Aap(1);
Aap = Aap / Aap(1);

% filter the signal
s_f = filter(Bap, Aap, s);

% compute ffts
S = fft(s);
S_f = fft(s_f);

figure; plot(abs(S));
hold on; plot(abs(S_f), '--');
grid;
legend('DFT of s(n)', 'DFT of s_f(n)');

% the two amplitudes should be one equal to the other (apart from small
deviations),
% since the signal s(n)
% has been filtered with an all-pass filter which does not modify the
% amplitude response of the filtered signal.
```