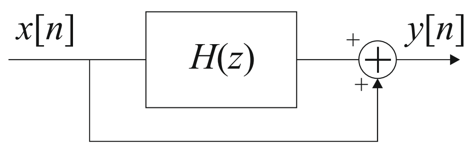
Multimedia Signal Processing 1st Module and Fundamentals of Multimedia Signal Processing

date: November 13th, 2024

Ex.1 (Pt.12)

An analog signal, $x(t) = 3\cos(2\pi f_1 t) + 5\sin(2\pi f_2 t)$, where $f_1 = 2kHz$ and $f_2 = 4kHz$ is sampled at 12kHz. We need to **completely** remove the component at f_2 using the configuration reported below



The filter $H(z)$ is an **All Pass** stable and causal filter with 2 zeros and 2 poles.

[4 pts] Design the filter $H(z)$ so that in the output signal $y[n]$ the component at f_2 in the original signal $x[n]$ is completely removed. Provide the difference equation of $H(z)$.

[2 pts] Provide the zeros-poles plot of $H(z)$.

[3 pts] Provide the zeros-poles plot of the whole filter $F(z)$ (considering also the loop) that defines the relation between $y[n]$ and $x[n]$.

[3 pts] After describing how to evaluate the amplitude response of the **total** filter $F(z)$ at f_1 (the exact result is not requested), provide its approximate magnitude response at all frequencies.

Ex.2 (Pt.10)

We need to filter in real-time the signal $x[n] = \{3, -1, -1, 0, 2, -3, -2, 1\}$ with the filter $h[n] = \{1, 2, 1\}$.

[3 pts.] Working with blocks of 5 samples, describe how to apply and provide the output using the Overlap and Add approach.

[3 pts.] Working with blocks of 5 samples, describe how to apply and provide the output using the Overlap and Save approach working in the time domain.

[4 pts.] Describe how the previous result can be obtained working in the frequency domain, in particular provide the associated W matrix associated to the Discrete Fourier Transform.

CONTINUES ON THE BACK

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

1) [2 pt] The signal x(n) is the sum of two sinusoidal contributions at frequencies 1.6KHz and 4KHz, respectively, and it evolves in time such that x(n = 0) = x(n = 20), with 20 samples corresponding to 1.25 ms. Define the signal, which has a duration of 58 ms.
2) [3.5 pt] Plot the absolute value of the DFT of the signal vs normalized frequencies between –0.5 and 0.5.
   - Can you see the peaks corresponding to the sinusoidal contributions? Are they located ***exactly*** in the expected normalized frequencies? Why?
   - Propose a solution to be able to see these peaks in the correct positions (hint: you have multiple options...Choose the one you prefer).
   - According to the chosen solution, define a new signal w(n) as "modified" version of x(n).
3) [4 pt] You are given a set of zeros and poles as follows:
   - `zeros_set = [0.99*exp(-pi*j/5); 0.99*exp(pi*j/5); 1.11*j; -1.11*j; -0.9]`
   - `poles_set = [1.01*exp(pi*j/5); 1.01*exp(-pi*j/5); 1/1.11*j; -1/1.11*j; 0.85]`
   - Select the zeroes and the poles to build a causal and stable all-pass filter H(z) with real coefficients. Define the filter H(z). Be careful to the final magnitude value of the filter.
   - Filter the signal w(n) through the filter H(z), defining the signal z(n).
   - Which are the expected amplitude values of the signal z(n) with respect to those of w(n)?
4) [2.5 pt] Define a new system H1(z) = 1 + H(z).
   - Filter the signal w(n) through the system, defining the output signal as y(n).
   - Compute the DFT of the signal y(n) and plot its absolute value vs normalized frequencies in [-0.5, 0.5].

Which are the frequency components that remain visible in the DFT of y(n)? Motivate your answer (hint: which are the zeroes and poles of H1(z)?).

Solutions

Ex.1

The sampled signal will be:

$$x[n]=3\cos\left(\frac{2\pi\cdot 2kHz}{12kHz}n\right)+5\sin\left(\frac{2\pi\cdot 4kHz}{12kHz}n\right)=3\cos\left(\frac{\pi}{3}n\right)+5\sin\left(\frac{2\pi}{3}n\right)$$

The all-pass filter has to introduce a phase rotation of π rads at $\omega_2=\frac{2\pi}{3}$ in order to be able to subtract completely that component from the input in the adder.

Since the filter $H(z)$ has two zeros and two poles in reciprocal conjugate position (property of the all-pass filters) we can place the poles close the unit circle ($\rho=0.9$) and the zeros in their reciprocal conjugate position at $\omega_2=\frac{2\pi}{3}$:

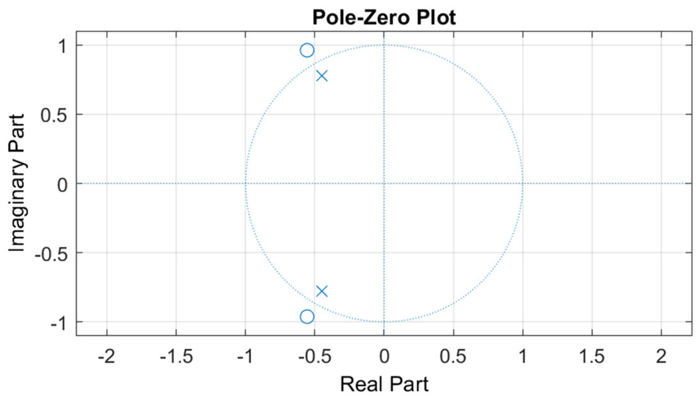
$$H(z)=\frac{\rho^2-2\rho\cos(\omega_2)z^{-1}+z^{-2}}{1-2\rho\cos(\omega_2)z^{-1}+\rho^2z^{-2}}=\frac{0.81-2\cdot 0.9\cdot(-1/2)z^{-1}+z^{-2}}{1-2\cdot 0.9\cdot(-1/2)z^{-1}+0.81z^{-2}}=$$
$$=\frac{0.81+0.9z^{-1}+z^{-2}}{1+0.9z^{-1}+0.81z^{-2}}$$

[the coefficients of the numerator will be in reverse order with respect to the ones of the denominator to satisfy the reciprocal conjugate position of each zero with respect to the poles]

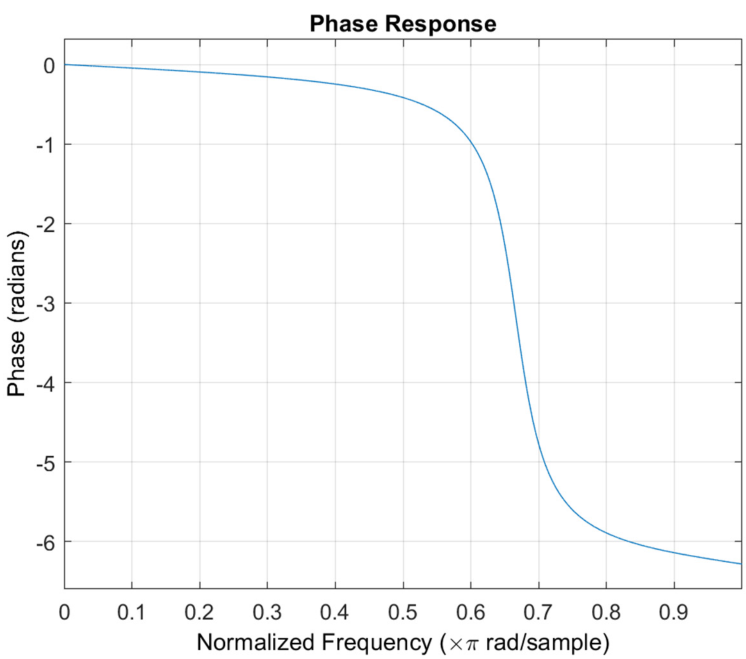
The difference equation is: $y[n]=0.81x[n]+0.9x[n-1]+x[n-2]-0.9y[n-1]-0.81y[n-2]$

The total filter will be: $F(z)=1+\frac{0.81+0.9z^{-1}+z^{-2}}{1+0.9z^{-1}+0.81z^{-2}}=\frac{1.81+1.8z^{-1}+1.81z^{-2}}{1+0.9z^{-1}+0.81z^{-2}}$

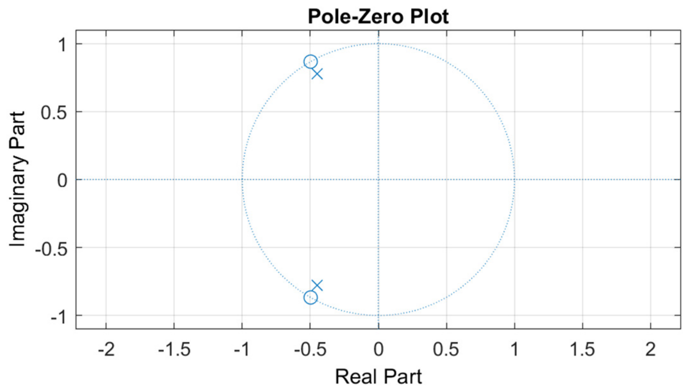
Zeros-poles plots of $H(z)$:



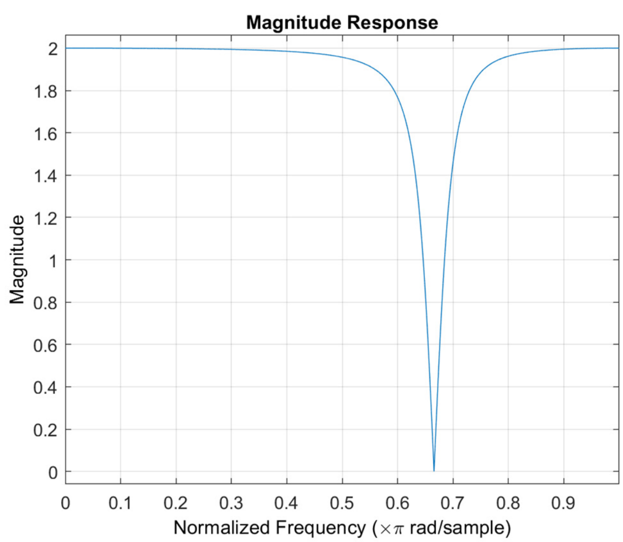
Phase response of $H(z)$:



Zeros-poles of $F(z)$:



Amplitude response of $F(z)$:



The amplitude response at f_1 depends from the chosen closeness of the two poles to the unit circle. To obtain its exact value we have to evaluate:

$$\left|F\left(z=e^{j2\pi f_1}=\frac{1}{2}+\frac{\sqrt{3}}{2}j\right)\right|=\left|\frac{1.81z^2+1.8z+1.81}{z^2+0.9z+0.81}\right|=\left|\frac{1.81\left(-\frac{1}{2}+\frac{\sqrt{3}}{2}j\right)+1.8\left(\frac{1}{2}+\frac{\sqrt{3}}{2}j\right)+1.81}{-\frac{1}{2}+\frac{\sqrt{3}}{2}j+0.9\left(\frac{1}{2}+\frac{\sqrt{3}}{2}j\right)+0.81}\right|$$

Ex.2

The Overlap and Add approach will give the following steps:

$[3 \quad -1 \quad -1 \quad 0 \quad 2]*[1 \quad 2 \quad 1]=[3 \quad 5 \quad 0 \quad -3 \quad 1 \quad \underline{4} \quad \underline{2}]$ where the last two samples will not be sent to the output but will be stored in order to be added to the beginning of the next block.

The second step will be $[-3 \quad -2 \quad 1 \quad 0 \quad 0]*[1 \quad 2 \quad 1]=[-3 \quad -8 \quad -6 \quad 0 \quad 1 \quad \underline{0} \quad \underline{0}]$ where we have to take into account (add to the output) the reminder from the previous step : $[4 \quad 2 \quad 0 \quad 0 \quad 0]$:

$$[-3 \quad -8 \quad -6 \quad 0 \quad 1]+[4 \quad 2 \quad 0 \quad 0 \quad 0]=[1 \quad -6 \quad -6 \quad 0 \quad 1]$$

The output will be $[3 \quad 5 \quad 0 \quad -3 \quad 1 \quad 1 \quad -6 \quad -6 \quad 0 \quad 1]$.

Overlap and Save requires the usage of circular convolution. Since the filter has a length of 3 There will be circular overlap of 2 samples, we can the process the signal in this way:

Add two zeros at the beginning of the signal to avoid the unwanted effects of the circular convolution and apply it:

$[0 \quad 0 \quad 3 \quad -1 \quad -1]\circledast[1 \quad 2 \quad 1]=[\underline{-3} \quad \underline{-1} \quad 3 \quad 5 \quad 0]$ then remove the first two underlined values that are just the side effect of the circular convolution.

The next step will involve the last two samples of the previous block and the circular convolution:

$[-1 \quad -1 \quad 0 \quad 2 \quad -3]\circledast[1 \quad 2 \quad 1]=[\underline{-5} \quad \underline{-6} \quad -3 \quad 1 \quad 1]$, and so on:

$$[2 \quad -3 \quad -2 \quad 1 \quad 0]\circledast[1 \quad 2 \quad 1]=[\underline{3} \quad \underline{1} \quad -6 \quad -6 \quad 0].$$

To get the same result of the OLA we have also to consider the last input sample:

$$[1 \quad 0 \quad 0 \quad 0 \quad 0]\circledast[1 \quad 2 \quad 1]=[\underline{1} \quad \underline{2} \quad 1 \quad 0 \quad 0].$$

The $\mathbf{W}$ matrix, for blocks of 5 elements of the DFT, will be a 5x5 matrix of elements W_{rc}, where r Is the raw index (starting from 0) while c is the column index (starting from 0).

$$W_{rc}=e^{-j\frac{2\pi}{5}r\cdot c}$$

Ex.3 (MATLAB CODE)

```
close all

clearvars

clc


%% 1. [2 pt]
```

```
% The signal x(n) is the sum of two sinusoidal contributions at

% frequencies 1.6KHz and 4KHz, respectively, and it evolves in time

% such that x(n = 0) = x(n = 20), with 20 samples corresponding to 1.25 ms.

% Define the signal, which has a duration of 58 ms.


f0 = 1.6e3;

f1 = 4e3;

duration = 58e-3;


% We need to find the sampling rate to define the time-axis.

% To find Fs, we know that P_samples and P_seconds are related one with the

% other by P_samples = Fs * P_seconds.

% Compute Fs = P_samples/P_seconds.

P_samples = 20;

P_seconds = 1.25e-3;

Fs = P_samples / P_seconds;


time = 0:1/Fs:duration;


x = cos(2*pi*f0*time) + cos(2*pi*f1*time);


%% 2. [3.5 pts]


% Plot the absolute value of the DFT of the signal vs normalized

% frequencies between –0.5 and 0.5.

% Can you see the peaks corresponding to the sinusoidal contributions,

% located exactly in the expected normalized frequencies? Why?

% Propose a solution to be able to see these peaks in the correct positions

% (hint: you have multiple options...Choose the one you prefer).
```

```
% According to the chosen solution, define a new signal w(n) as
% “modified” version of x(n).


X = fft(x);
N = length(x);
freq_axis = (0:1/N:1 - 1/N) - 0.5;


figure;
stem(freq_axis, abs(fftshift(X)));
title('Absolute value of the DFT of the signal x(n)');
grid;
xlabel('f [norm]');


% The normalized frequencies are:
% f0_n = 1.6e3/Fs = 0.1
% f1_n = 4e3/Fs = 0.25.


% The number of samples of the signal is defined by round(duration*Fs) = 928
% This is not an integer multiple of the period, therefore we will not see
% the two peaks exactly at the normalized frequencies of the two cosines.


% Moreover, the delta between one frequency sample and the next one is
% dependent on 928, therefore delta-f = 1/928 = 0.0011.. f0_n and f1_n are not
% multiples of this. To be sure of seeing the two peaks in their exact
% location, the step size should be 0.05, or a divisor of it.
% If we impose that, in general, for k being a generic integer,
% 0.05/k = 1/N_samples --> N_samples = k/0.05 = k*20 = k*P_samples
% We come back to the previous consideration --> the bumber of samples
% should be an integer multiple of the period
```

```
% The closest multiple of the period to 928 is N = 940.

% Solution # 1: zero-padding
% we can zero-padd the signal until reaching an integer multiple of the
% period.
% by zero padding, we will see the two peaks, even if some artifacts will
% remain due to cyclic convolution with the sinc
N_final = 940;
w_1 = [x, zeros(1, N_final - length(x))];

% Solution # 2: acquire more measurements
% in this case, we will see 4 exact peaks with equal amplitudes
N_final = 940;
% we need to redefine the time axis (easier to do this in samples)
time_samples = 0:N_final -1;
f0_n = f0/Fs;
f1_n = f1/Fs;
w_2 = cos(2*pi*f0_n*time_samples) + cos(2*pi*f1_n*time_samples);

% % check the signal ffts (not required)
% W_1 = fft(w_1);
% W_2 = fft(w_2);
% freq_axis = (0:1/N_final:1 - 1/N_final) - 0.5;
% figure;
% stem(freq_axis, abs(fftshift(W_1)));
% title('Absolute value of the DFT of the signal w_1(n) (zero-padding)');
% grid;
% xlabel('f [norm]');
```

```
% figure;
% stem(freq_axis, abs(fftshift(W_2)));
% title('Absolute value of the DFT of the signal w_2(n) (more measurements)');
% grid;
% xlabel('f [norm]');

%% 3. [4 pt]

% You are given a set of zeros and poles as follows:
% zeros_set = [0.99*exp(-pi*j/5); 0.99*exp(pi*j/5); 1.11*j; -1.11*j; -0.9]
% poles_set = [1.01*exp(pi*j/5); 1.01*exp(-pi*j/5); 1/1.11*j; -1/1.11*j; 0.85]
% Select the zeroes and the poles to build a causal and stable all-pass
% filter H(z) with real coefficients. Define the filter H(z).
% Be careful to the final magnitude value of the filter.
% Filter the signal w(n) through the filter H(z), defining the signal z(n).
% Which are the expected amplitude values of the signal z(n) with respect
% to those of w(n)?

zeroes_set = [0.99*exp(-pi*1i/5); 0.99*exp(pi*1i/5); 1.11*1i; -1.11*1i; -0.9];
poles_set = [1.01*exp(pi*1i/5); 1.01*exp(-pi*1i/5); 1/1.11*1i; -1/1.11*1i; 0.85];

% % to better analyze the situation (not required)
% figure;
% zplane(zeroes_set, poles_set);
% title('zeroes and poles available');
% grid;

% we must select zeroes and poles in conjugate reciprocal positions.
% to build a causal stable filter, we choose only zeroes outside the
```

```
% unit circle, corresponding to poles inside the circle.

zeroes = [1.11*1i; -1.11*1i];

poles = [1/1.11*1i; -1/1.11*1i];


% define the filter H(z)

B = poly(zeroes);

A = poly(poles);


% we need to adjust the gain:

c_0 = sum(A) / sum(B);


% multiply by c_0

B = c_0 * B;


% % check the behaviour of the filter (not required)

% [H, omega] = freqz(B, A, 1024, 'whole');

% figure,

% plot(omega./(2*pi), abs(H));

% title('|DTFT| of the filter H(z)');

% grid;

% xlabel('f [norm]');


% filter the signal (I'm taking w_2 but it's the same for w_1)

z = filter(B, A, w_2);


% z is an all-pass --> the amplitude values remains more or less the same

% as those of w(n)


%% 4. [2.5 pts]
```

```
% Define a new system H1(z) = 1 + H(z).

% Filter the signal w(n) through the system, defining the output signal as y(n).

% Compute the DFT of the signal y(n) and plot its absolute value vs

% normalized frequencies in [-0.5, 0.5).

% Which are the frequency components that remain visible in the DFT of y(n)?

% Motivate your answer (hint: which are the zeroes and poles of H1(z)?).


% We start from a filter H(z) = B(z) / A(z).

% The filter H1(z) = 1 + H(z) = (B(z) + A(z))/A(z).


B1 = B + A;

A1 = A;


y = filter(B1, A1, w_2);


Y = fft(y);

N = length(y);

freq_axis = (0:1/N:1 - 1/N) - 0.5;


figure;

stem(freq_axis, abs(fftshift(Y)));

title('Absolute value of the DFT of the signal y(n)');

grid;

xlabel('f [norm]');


% By performing hand-made computations, we find that the poles remain the

% same as in H(z), while the zeros have the same phase (corresponding to the

% frequency f1_n) but absolute value = 1.
```

```
% For this reason, H1(z) is a notch filter in f1.
% Therefore, apart from small deviations,
% the only components that remain are those due to f0.

% % zeroes-poles plot (not required)
% figure;
% zplane(B1, A1);
% title('zeroes and poles of H_1(z)');
% grid;
```