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

date: February 6th, 2025

Ex.1 (Pt.12)

From the following signal $x[n]=\{-2,1,0,-2,0,1\}$, we need to remove completely the spurious components at the continuous frequency and at the Nyquist frequency.

1. [3pts.] Working only in the Frequency domain, provide the $\mathbf{W}$ matrix in order to get the DFT of the signal.
2. [5pts.] Find the DFT of the signal, define and apply the proper filter to remove just the spurious components preserving the other ones.
3. [4 pts.] Find the final output signal $y[n]$ in the time domain using the iDFT.

Ex.2 (Pt.9)

We need to extract the Nyquist component working on blocks of 4 samples of the following signal $x[n]$ working in the <u>time</u> domain: $x[n]=\{-2,1,0,-2,0,1\}$.

We want to obtain the result using the Overlap and Add (OLA) technique working on 4 samples.

1. [3 pts.] define the filter $h[n]$
2. [6 pts] apply the filter $h[n]$ using OLA applying zero-padding where necessary.

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

1) [2 pt] The signal x(n) contains two sinusoidal contributions (with the same amplitude = 1) at the normalized frequencies 0.1 and 0.25. The period of x(n) is 1.25 [ms] and the duration is 100 [ms]. Define the signal x(n).
2) [3.5 pt] Define the filter H(z) as H(z) = (0.9025 + z^{-2})·(1 - 1.8cos(π/5)z^{-1}+ 0.81z^{-2}) / (1 + 0.9025 z^{-2}).
   - Plot the behaviour of the filter in the frequency domain.
   - Filter the signal x(n) with H(z), defining y(n).
   - Which is the value of y(n = 0)? Define it in MATLAB, but specify also the numerical value that you expect.
3) [4 pt] Compute the all-pass minimum-phase decomposition of the filter H(z), defining H_ap(z) and H_min(z) as the two components. (Hint: no computations are needed!)
   - Filter the signal x(n) with H_ap(z) and H_min(z), defining y_ap(n) and y_min(z).
   - Define the signal w(n) as the arithmetic mean between x(n) and y_ap(n).
   - Find the filter H_w(z) such that W(z) = H_w(z) · X(z).
4) [2.5 pt] Compute the DFTs of the signals x(n), y(n), y_min(n), y_ap(n), w(n) and plot their absolute values as a function of the normalized frequency axis, starting from frequency -0.5. Comment on the position/amplitude of the peaks you expect to see for every signal.

Solutions

Ex.1

$$w_6 = \frac{2\pi}{6} = \frac{\pi}{3}$$

$$\mathbf{W} = \begin{bmatrix} w_6^{0\cdot0} & w_6^{0\cdot1} & w_6^{0\cdot2} & w_6^{0\cdot3} & w_6^{0\cdot4} & w_6^{0\cdot5} \\ w_6^{1\cdot0} & w_6^{1\cdot1} & w_6^{1\cdot2} & w_6^{1\cdot3} & w_6^{1\cdot4} & w_6^{1\cdot5} \\ w_6^{2\cdot0} & w_6^{2\cdot1} & w_6^{2\cdot2} & w_6^{2\cdot3} & w_6^{2\cdot4} & w_6^{2\cdot5} \\ w_6^{3\cdot0} & w_6^{3\cdot1} & w_6^{3\cdot2} & w_6^{3\cdot3} & w_6^{3\cdot4} & w_6^{3\cdot5} \\ w_6^{4\cdot0} & w_6^{4\cdot1} & w_6^{4\cdot2} & w_6^{4\cdot3} & w_6^{4\cdot4} & w_6^{4\cdot5} \\ w_6^{5\cdot0} & w_6^{5\cdot1} & w_6^{5\cdot2} & w_6^{5\cdot3} & w_6^{5\cdot4} & w_6^{5\cdot5} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & e^{-j\frac{\pi}{3}} & e^{-j\frac{2\pi}{3}} & -1 & e^{-j\frac{4\pi}{3}} & e^{-j\frac{5\pi}{3}} \\ 1 & e^{-j\frac{2\pi}{3}} & e^{-j\frac{4\pi}{3}} & 1 & e^{-j\frac{2\pi}{3}} & e^{-j\frac{4\pi}{3}} \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & e^{j\frac{2\pi}{3}} & e^{j\frac{4\pi}{3}} & 1 & e^{j\frac{2\pi}{3}} & e^{j\frac{4\pi}{3}} \\ 1 & e^{j\frac{\pi}{3}} & e^{j\frac{2\pi}{3}} & -1 & e^{j\frac{4\pi}{3}} & e^{j\frac{5\pi}{3}} \end{bmatrix}$$

$$X[k] = \mathbf{W} \cdot \begin{bmatrix} -2 \\ 1 \\ 0 \\ -2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \\ -5 \\ -2 \\ -5 \\ 1 \end{bmatrix}, H[k] = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}; \qquad Y[k] = X[k] \cdot H[k] = \begin{bmatrix} 0 \\ 1 \\ -5 \\ 0 \\ -5 \\ 1 \end{bmatrix}$$

$$y[n] = \mathbf{W}^{-1} Y[k] = \frac{1}{6} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & e^{j\frac{\pi}{3}} & e^{j\frac{2\pi}{3}} & -1 & e^{j\frac{4\pi}{3}} & e^{j\frac{5\pi}{3}} \\ 1 & e^{j\frac{2\pi}{3}} & e^{j\frac{4\pi}{3}} & 1 & e^{j\frac{2\pi}{3}} & e^{j\frac{4\pi}{3}} \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & e^{-j\frac{2\pi}{3}} & e^{-j\frac{4\pi}{3}} & 1 & e^{-j\frac{2\pi}{3}} & e^{-j\frac{4\pi}{3}} \\ 1 & e^{-j\frac{\pi}{3}} & e^{-j\frac{2\pi}{3}} & -1 & e^{-j\frac{4\pi}{3}} & e^{-j\frac{5\pi}{3}} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ -5 \\ 0 \\ -5 \\ 1 \end{bmatrix} = \begin{bmatrix} -4/3 \\ 1 \\ 2/3 \\ -2 \\ 2/3 \\ 1 \end{bmatrix}$$

Ex.2

$h[n] = \{1, -1, 1, -1\}$ to analyze the Nyquist frequency on blocks of 4 samples.

Assuming that the samples before and after our input signal are zero (zero padding) working on blocks of 4 samples and keeping the reminder of 3 samples at each step we will get:

$$y[n] = x[n] * h[n] = \{\text{-2} \quad 3 \quad \text{-3} \quad 1 \quad 1 \quad \text{-1} \quad 1 \quad 1 \quad \text{-1}\}$$

Ex.3

```
close all
clearvars
clc

%% 1. [2 pt]

% The signal x(n) contains two sinusoidal contributions (with the same
% amplitude = 1) at the normalized frequencies 0.1 and 0.25.
% The period of x(n) is 1.25 [ms] and the duration is 100 [ms].
% Define the signal x(n).

f0 = 1/10;
f1 = 1/4;
P_seconds = 1.25e-3;
duration = 0.1;

% 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.
% First, find P_samples, then compute Fs = P_samples/P_seconds.

P_samples = lcm(10, 4); % = lcm(1/f0, 1/f1)
Fs = P_samples/P_seconds;

time = 0:1/Fs:duration - 1/Fs;

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

%% 2. [3.5 pt]

% Define the filter H(z) as H(z) = (0.9025 + z^(-2))·(1 - 1.8cos(pi/5)z^(-1)+
% 0.81z^(-2)) / (1 + 0.9025z^(-2)).
% Plot the behaviour of the filter in the frequency domain.
% Filter the signal x(n) with H(z), defining y(n).
% Which is the value of y(n = 0)? Define it in Matlab, but specify also
% the numerical value that you expect.

% To define the filter, we can exploit the convolution property
B = conv([0.9025, 0, 1], [1, -1.8*cos(pi/5), 0.81]);
A = [1, 0, 0.9025];

% Behaviour of the filter in the frequency domain
[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]');

% % to better analyze the filter (not required)
% figure;
% zplane(B, A);
% title('Z-plane of filter H(z)');
% grid;

% filter the signal
y = filter(B, A, x);

y_0 = x(1)*B(1);
```

```
% we expect that y_0 = x(n=0) * h(n=0)
% x(n=0) = 2, as we have the sum of two cosinusoidal signals being 1 in
% n=0.
% h(n=0) is only due to the numerator coefficient in n = 0, therefore it
% will be 0.9025.
% y_0 = 2 * 0.9025 = 1.8050

%% 3. [4 pt]

% Compute the all-pass minimum-phase decomposition of the filter H(z),
% defining H_ap(z) and H_min(z) as the two components.
% (Hint: no computations are needed!)
% Filter the signal x(n) with H_ap(z) and H_min(z), defining y_ap(n) and y_min(z).
% Define the signal w(n) as the arithmetic mean between x(n) and y_ap(n).
% Find the filter H_w(z) such that W(z) = H_w(z) · X(z).

% The filter is already decomposed in an all-pass component and a
% minimum-phase component.
% The all-pass component H_ap(z) = (0.9025 + z^(-2))/ (1 + 0.9025z^(-2)).
% The minimum-phase component H_min(z) = (1 - 1.8cos(pi/5)z^(-1)+
% 0.81z^(-2)).
% Even if you want to follow the standard methodology, you will find that,
% after all the steps, you end up with these two exact components.

B_ap = [0.9025, 0, 1];
A_ap = [1, 0, 0.9025];

B_min = [1, -1.8*cos(pi/5), 0.81];
A_min = 1;

% % to better analyze the filters (not required)
% figure;
% zplane(B_ap, A_ap);
% title('Z-plane of filter H_{ap}(z)');
% grid;
% [H, omega] = freqz(B_ap, A_ap, 1024, 'whole');
% figure,
% plot(omega./(2*pi), abs(H));
% title('|DTFT| of the filter H_{ap}(z)');
% grid;
% xlabel('f [norm]');
%
% figure;
% zplane(B_min, A_min);
% title('Z-plane of filter H_{min}(z)');
% grid;
% [H, omega] = freqz(B_min, A_min, 1024, 'whole');
% figure,
% plot(omega./(2*pi), abs(H));
% title('|DTFT| of the filter H_{min}(z)');
% grid;
% xlabel('f [norm]');

% Filter the signal x
y_ap = filter(B_ap, A_ap, x);
y_min = filter(B_min, A_min, x);

% Define the signal w as the arithmetic mean between x(n) and y_ap(n)
w = 0.5*x + 0.5*y_ap;

% Find the filter H_w such that W(z) = X(z) * H_w(z)
% W(z) = (X(z) + X(z)*H_ap(z)) / 2 = X(z) (H_ap(z) + 1)/2
```

```
% --> H(z) = (H_ap(z) + 1)/2.
B_w = (B_ap + A_ap)*0.5; % B_w = [(1 + 0.9025), 0, (1 + 0.9025)]/2;
A_w = A_ap;
% --> This is a notch filter in f1. We can understand it by looking at the
% position of the zeros and poles: they have the same phase, but the zeros
% are on the unit circle.

% % to better analyze the filter (not required)
% figure;
% zplane(B_w, A_w);
% title('Z-plane of filter H_w(z)');
% grid;
% [H, omega] = freqz(B_w, A_w, 1024, 'whole');
% figure,
% plot(omega./(2*pi), abs(H));
% title('|DTFT| of the filter H_{w}(z)');
% grid;
% xlabel('f [norm]');

%% 4. [2.5pt]

% Compute the DFTs of the signals x(n), y(n), y_min(n), y_ap(n), w(n)
% and plot their absolute values as a function of the normalized frequency
% axis, starting from frequency 0.
% Comment on the position/amplitude of the peaks you expect to see
% for every signal.

X = fft(x);
Y = fft(y);
Y_min = fft(y_min);
Y_ap = fft(y_ap);
W = fft(w);

N = length(y);
freq_axis = 0:1/N:1 - 1/N;

figure;
stem(freq_axis, abs(X));
title('Absolute value of the DFT of the signal x(n)');
grid;
xlabel('f [norm]');
% We find four peaks in f0, f1, 1-f0, 1-f1. They have the same amplitude.

figure;
stem(freq_axis, abs(fftshift(Y)));
title('Absolute value of the DFT of the signal y(n)');
grid;
xlabel('f [norm]');
% We find four peaks in f0, f1, 1-f0, 1-f1.
% The peaks in frequency f0 are slightly attenuated
% with respect to those of x(n),
% because the filter has zeros in f0 with absolute value = 0.9.
% The peaks in frequency f1 are not attenuated by the filter.
% Their amplitude differs from that of x(n) because the filter introduces
% a gain in f = f1 that is different from 1.

figure;
stem(freq_axis, abs(fftshift(Y_min)));
title('Absolute value of the DFT of the signal y_{min}(n)');
grid;
xlabel('f [norm]');
% We find basically no differences with respect to y(n). H(z) and H_min(z)
```

```
% differ only for the all-pass component, which has no effect on the
% amplitude.

figure;
stem(freq_axis, abs(fftshift(Y_ap)));
title('Absolute value of the DFT of the signal y_{ap}(n)');
grid;
xlabel('f [norm]');
% The peaks are basically the same as x(n), because the filter is an all-pass

figure;
stem(freq_axis, abs(fftshift(W)));
title('Absolute value of the DFT of the signal w(n)');
grid;
xlabel('f [norm]');
% The filter is a notch in f = f1. Therefore, we find only the
% contributions at f0, with an amplitude which is similar to that of x(n).
```