Multimedia Signal Processing 1st Module

MATLAB part [12 pts]

June 21st, 2022

Text:

1) [2 pt] The signal z(t) is the sum of two cosinusoidal signals u(t) and v(t): u(t) has period 0.5 milliseconds; v(t) has period 1.2 milliseconds. Both signals have amplitude 1. The sampling period is 0.1 milliseconds and the signals are acquired for 50 milliseconds. Define the discrete-time signal z(n).
2) [5 pt] We want to filter out the high frequency components of the signal. Design an FIR filter (81 samples) to maintain almost unaltered the spectral content until the lowest frequency of the signal z(n).
   - Plot the impulse response of the filter h(n) with the function stem.
   - Plot the frequency response of the filter (only the absolute value) with the function 'freqz' over 2048 samples versus the frequency spectrum in Hz (use the 'whole' flag).
   - Filter the signal z(n), defining the signal z_f(n) and plot the signal z_f(n) as a function of the time samples.
     i) The signal z_f(n) should resemble one of the two original signals (u(n) or v(n)): which is the most similar signal? Why?
     ii) Plot the most similar signal (u(n) or v(n)) in the same figure.
     iii) Compare the two signals: which are the differences between them? Hint: they should differ both in time-shift and amplitude. Motivate both two differences. Which is the reason of the specific value of amplitude obtained? Which is the reason of the specific time delay obtained?
     iv) By following the previous considerations, define a new cosinusoidal signal z_th(n) (with a specific amplitude and time delay) that theoretically matches the signal z_f(n), apart from some initial artifacts.
     v) Plot the signal z_th(n) in the same figure of z_f(n) and check that the two signals match.
3) [2 pt] Instead of filtering the signal z(n), compute the circular convolution over the total number of signal samples between z(n) and h(n), exploiting the DFT properties. Define the result as z_c(n).
   - Are there any differences in the signal z_c(n) with respect to the signal z_f(n)? Plot the squared difference between the two signals as a function of time samples. Use the function 'semilogy'. Motivate the reason of these differences and in which samples you find them.
   - Which is the easiest way to remove the artifacts introduced in the cyclic convolution?
4) [3 pt]
   - Plot the absolute value of DFT of z_f(n) vs the frequency spectrum in Hz.
   - Apply a window to the signal z_f(n) such that its DFT corresponds to two exact peaks. Choose the proper window and number of samples by selecting among these possible functions: rectwin(); blackman(); hann(). (hint: be careful in choosing the temporal samples to remove the filtering artifacts). Define the windowed signal as z_w(n).
   - In a new figure, plot the absolute value of DFT of z_w(n) vs the frequency spectrum in Hz. Compute this DFT in two ways:
     i) by using the same number of samples used for z_f(n).
     ii) by using the number of samples of z_w(n).
     iii) In which case the DTF presents two peaks? What is happening in the other scenario? Motivate your answers.

Solution:

```
close all
clearvars
clc

%% 1. [2 pt]

% The signal z(t) is the sum of two cosinusoidal signals u(t) and v(t):
% u(t) has period 0.5 milliseconds; v(t) has period 1.2 milliseconds.
% Both signals have amplitude 1. The sampling period is 0.1 milliseconds
% and the signals are acquired for 50 milliseconds.
% Define the discrete-time signal z(n).

period_u = 0.5e-3;
f_u = 1 / period_u;

period_v = 1.2e-3;
f_v = 1 / period_v;

% sampling period
Ts = 0.1e-3;
duration = 50e-3;

% time axis
time = 0:Ts:duration;

u = cos(2*pi*f_u*time);
v = cos(2*pi*f_v*time);
z = u + v;

%%  2. [5 pt]

% We want to filter out the high frequency components of the signal.
% Design an FIR filter (81 samples) to maintain almost unaltered the
% spectral content until the lowest frequency of the signal z(n).
% Plot the impulse response of the filter h(n) with the function stem.
% Plot the frequency response of the filter (only the absolute value) with
% the function 'freqz' over 2048 samples versus the frequency spectrum in
% Hz (use the 'whole' flag).
% Filter the signal z(n), defining the signal z_f(n) and plot the signal
% z_f(n) as a function of the time samples.
% The signal z_f(n) should resemble one of the two original signals
% (u(n) or v(n)): which is the most similar signal? Why?
% Plot the most similar signal (u(n) or v(n)) in the same figure.
% Compare the two signals: which are the differences between them?
% Hint: they should differ both in time-shift and amplitude.
% Motivate both two differences. Which is the reason of the specific value
% of amplitude obtained? Which is the reason of the specific time delay obtained?
% By following the previous considerations, define a new cosinusoidal
% signal z_th(n) (with a specific amplitude and time delay) that
% theoretically matches the signal z_f(n), apart from some initial artifacts.
% Plot the signal z_th(n) in the same figure of z_f(n) and check that the
% two signals match.
```

```
% We need the sampling rate to find the normalized frequency
Fs = 1/Ts;

% The lowest frequency f_v should correspond to the cutoff
cutoff = f_v/Fs;
cutoff_filter = 2*cutoff;
N = 81;
h = fir1(N-1, cutoff_filter);

% impulse response of the filter
stem(h);
title('h(n)');
xlabel('n');
grid;

% Frequency response of the filter
[H, omega] = freqz(h, 1, 2048, 'whole');
figure,
plot(omega./(2*pi).*Fs, abs(H));
title('|H(f)|');
xlabel('f[Hz]');
grid;

% Filter the signal z, defining the signal z_f.
z_f = filter(h, 1, z);

% Plot the signal z_f as a function of the time samples.
figure;
plot(z_f);
leg = {};
leg{1} = 'z_f(n)';
grid;

% The signal z_f should resemble the signal v, because we designed the
% filter to filter out the signal components at higher frequencies.
% Only the frequency component f_v remains.

% Plot the signal v(n)
hold on;
plot(v, '--');
leg{2} = 'v(n)';

% The amplitude is different because the cutoff frequency defines the -6dB
% point, so the resulting amplitude for the filtered signal at f_v is exactly 0.5.
% The time shift is different because the filtering process introduces some
% delay, and the exact delay corresponds to the temporal position of the
% maximum of the filter --> 41 samples.

amplitude_th = 0.5;
[~, filter_delay] = max(h);
time_shift_th = filter_delay;

% theoretical signal z_th
z_th = amplitude_th * cos(2*pi*f_v * (time - time(time_shift_th)));
```

```
% Plot the signal z_th
plot(z_th, '*--');
leg{3} = 'z_{th}(n)';
xlabel('n');
legend(leg);

%% 3. [2 pt]

% Instead of filtering the signal z(n), compute the circular convolution
% over the total number of signal samples between z(n) and h(n),
% exploiting the DFT properties. Define the result as z_c(n).
% Are there any differences in the signal z_c(n) with respect to the signal
z_f(n)?
% Plot the squared difference between the two signals as a function of time
% samples. Use the function 'semilogy'. Motivate the reason of
% these differences and in which samples you find them.
% Which is the easiest way to remove the artifacts introduced
% in the cyclic convolution?

% Compute the DFT of the filter and of the signal z over N_f samples
N_f = length(z);
H = fft(h, N_f);
Z = fft(z, N_f);

% Cyclic convolution
z_c = ifft(Z.*H);

figure;
semilogy((z_f - z_c).^2);
grid;
xlabel('n');
leg = {};
leg{1} = '(z_f(n) - z_c(n))^2';

% The two signals differ in the first 80 samples, which correspond to the
% length of the filter (=81 samples) - 1. This is due to periodic artifacts
% of the cyclic convolution between the two signals.

% To remove the artifacts, we can compute the cyclic convolution over a
% number of samples equal to length(z) + length(h) - 1.
N_f_c = length(z) + length(h) -1;
Z_c = fft(z, N_f_c);
H_c = fft(h, N_f_c);
z_c_c = ifft(Z_c.*H_c);

% We can plot again the squared difference
hold on;
% we have to limit the resulting signal
semilogy((z_f - z_c_c(1:length(z))).^2, '*-');
leg{2} = '(z_f(n) - z_c_c(n))^2';
legend(leg);

%% 4. [3 pt]
```

```
% Plot the absolute value of DFT of z_f(n) vs the frequency spectrum in Hz.
% Apply a window on the signal z_f(n) such that its DFT corresponds to two
% exact peaks. Choose the proper window and number of samples by selecting
% among these possible functions: rectwin(); blackman(); hann().
% (hint: be careful in choosing the temporal samples to remove the
% filtering artifacts). Define the windowed signal as z_w(n).
% In a new figure, plot the absolute value of DFT of z_w(n) vs the frequency
% spectrum in Hz. Compute this DFT in two ways:
% i)    by using the same number of samples used for z_f(n).
% ii)   by using the number of samples of z_w(n).
% In which case the DTF presents two peaks?
% What is happening in the other scenario? Motivate your answers.

% DFT of z_f(n)
Z_f = fft(z_f);
freq_axis = 0:(1/length(z_f)) * Fs:Fs - (1/length(z_f)) * Fs;

figure;
stem(freq_axis, abs(Z_f));
grid;
title('|Z_f(f)|');
xlabel('f[Hz]');

% We select the rectangular window over a number of samples exactly equal
% to one period of the signal v(n) (the one that remains after the
% filtering). In doing this, we can see two exact peaks in the spectrum.
P_zf_sample = period_v * Fs;
w_r = rectwin(round(P_zf_sample));

% We don't start selecting the signal samples from the beginning, because
% we have the artifacts left by the filtering process (see Figure 3).
% For instance, we can start from a number of samples equal to the
% length of the filter h(n).
z_w = z_f(length(h):length(h) + P_zf_sample - 1) .* w_r';

% DFT of z_w(n) by using the same number of samples used for z_f(n).
Z_w = fft(z_w, length(z_f));

figure;
leg = {};
stem(freq_axis, abs(Z_w));
leg{1} = '|Z_w(f)| over Z samples';
hold on;
grid;

% DFT of z_w(n) by using the number of samples of z_w(n).
Z_w = fft(z_w);

freq_axis = 0:(1/length(z_w)) * Fs:Fs - (1/length(z_w)) * Fs;
stem(freq_axis, abs(Z_w));
leg{2} = '|Z_w(f)| over the window samples';

% In the second scenario, we can see two perfect peaks: indeed, we selected
% a number of samples exactly equal to the period of the sinusoid.
% In the first scenario, we cannot see two perfect peaks, because we are
```

```
% zero-padding the signal. Hence, in the DFT, we see artifacts due to the
% convolution with the periodic sinc.
```