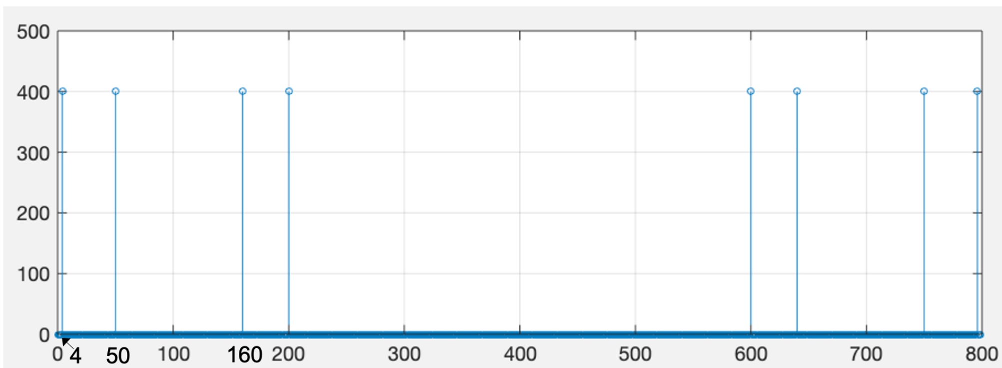
MultimediaSignal Processing 1st Module

MATLAB part [12 pts]

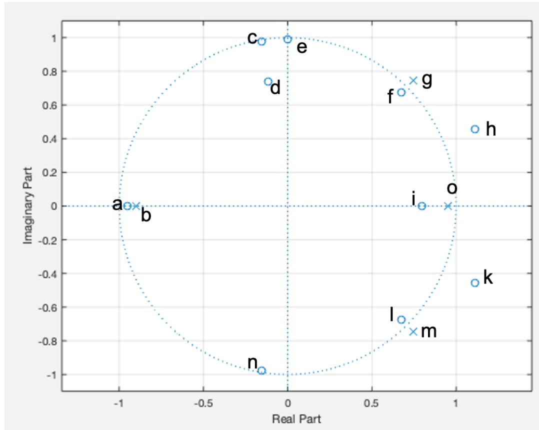
July 26th, 2021

Text:

1. [3 pt] You are given the following plot of a real DFT as a function of frequency samples:



   - Define the discrete-time signal x related to this DFT. (hint: the amplitude of each signal component is equal to 1, and the DFT has been computed using the total amount of signal samples (800), without zero-padding).
   - The period of x is 0.1 seconds. Find the sampling rate used to sample the signal.
2. [2.5 pt] Downsample the signal x with a factor M = 4, obtaining x_downsampled.
   - Which is the period of x_downsampled (in number of samples)?
   - Plot the absolute value of DFTs of x and of x_downsampled in the same figure (using the function stem), as a function of the normalized frequency axis starting from 0. In particular, for each signal, compute its DFT on a number of samples exactly equal to its period.
   - Comment on the position of all the peaks in both the two signals. Is there any frequency alias introduced by the downsampling process? If yes/no, where and why?
3. [3 pt] You have to project a low pass filter in order to filter out (if present) the frequency alias introduced by the downsampling process. You have two options: (1) generating a low-pass filter using the windowing method; (2) generating a low-pass filter using poles-zeroes.
   - Which is the maximum possible cutoff frequency to select in order to avoid aliasing?
   - Define the low-pass filter h_w using the windowing method having the maximum possible cutoff frequency, with order 64.
   - From the following zeros-poles diagram, select only the zeroes and poles such that you end up with a stable low-pass filter, minimum phase, with real coefficients, defined as h_zp. In particular, a = -0.95; b = -0.9; c, n = 0.99*exp(+-2*pi*j*0.275); d = 0.75*exp(2*pi*j*0.275); e = 0.99j; f, l = 0.95*exp(+-2*pi*j*0.125); g, m = 1.05*exp(+-2*pi*j*0.125); i = 0.8; h, k = 1.2* exp(+-2*pi*j*0.0625); o = 0.95.



- Find the numerator and denominator of the selected filter H_zp such that the filter gain in frequency = 0 is equal to 1.

4. [3.5] Filter the signal x with h_w and h_zp, defining x_w and x_zp.
   - Plot the 3 signals (x, x_w, x_zp) in the same figure.
   - Looking at the temporal behaviour of the signals, which is, among x_w and x_zp, the signal with the most accentuated low-pass behaviour?
   - Investigating the three signals also in the frequency domain, which is, in your opinion, the best filter of the two? Motivate your answer.
   - Select this filter and decimate the original signal x.
   - Compute the DFT of the decimated signal and check if frequency alias is maintained or deleted (compare the DFTs of the decimated and of the downsampled signals).

Solution:

```
close all
clearvars
clc

%% 1.

% [3 pt] You are given the following real DFT as a function of frequency
samples:
% (see the exam text).
% Define the time-discrete signal x related to this DFT.
% (hint: the amplitude of each portion of the signal is equal to 1,
% and the DFT has been computed using the total amount of signal samples,
% without zero-padding).
% The period of x is 0.1 seconds. Find the sampling rate used to sample
the signal.

N = 800;
% the signal is composed by 4 cosinusoids, as the DFT is real and we find
8
% peaks (4 peaks before N/2, and other 4 peaks mirrored after N/2).
% the normalized frequencies correspond to the peak positions divided by N
= 800.
f_vec = [4; 50; 160; 200]./N;
f0 = f_vec(1);
f1 = f_vec(2);
f2 = f_vec(3);
f3 = f_vec(4);

% define the signal x:
n = 0:N-1;
x = sum(cos(2*pi*f_vec.*n), 1);
% otherwise:
%x = cos(2*pi*f0*n) + cos(2*pi*f1*n) + cos(2*pi*f2*n) + cos(2*pi*f3*n);

% The period of x is 0.1 seconds. Find the sampling rate used to sample
the signal.

% periods of the sinusoids:
P0 = 1/f0; % --> 200
P1 = 1/f1; % --> 16
P2 = 1/f2; % --> 5
P3 = 1/f3; % --> 4
% lcm between (200, 16, 5, 4) --> 400.

px_samples = 400;
px_sec = 0.1;
```

```
Fs = px_samples/px_sec;

%% 2.

% [2.5 pt] Downsample the signal x with a factor M = 4, obtaining
% x_downsampled.
% Which is the period of x_downsampled (in number of samples)?
% Plot the DFT of x and of x_downsampled in the same figure (using the
function stem),
% as a function of the normalized frequency axis starting from 0.
% In particular, compute for each signal its DFT on a number of samples
% exactly equal to its period.
% Comment on the position of all the peaks in both the two signals.
% Is there any frequency alias introduced by the downsampling process? If
% yes/no, where and why?

% downsample the signal
M = 4;
x_downsampled = x(1:M:end);

% the period is equal to the period of x divided by M.
p_downsampled = px_samples/M;

%compute for each signal its DFT on a number of samples
% exactly equal to its period.

norm_freq_axis_x = 0:1/px_samples:1 - 1/px_samples;
X = fft(x, px_samples);

norm_freq_axis_x_downsampled = 0:1/p_downsampled:1 - 1/p_downsampled;
X_downsampled = fft(x_downsampled, p_downsampled);

figure(1);
leg = {};
% original signal
stem(norm_freq_axis_x, abs(X));
leg{1} = 'DFT of the Original signal';
grid;
hold on;
% downsampled signal
stem(norm_freq_axis_x_downsampled, abs(X_downsampled));
leg{2} = 'DFT of the Downsampled signal';
legend(leg);

% Comment on the position of all the peaks in both the two signals.
% Is there any frequency alias introduced by the downsampling process? If
% yes/no, where and why?
% in the original signal, we have peaks in f0, f1, f2, f3, 1-f0, 1-f1,
% 1-f2, 1-f3.
```

```
% in the downsampled signal, we have frequency alias, because the signal
% frequency content is not limited to 1/(2*M), which is the cut-off
% frequency to be used to avoid alias.
% in X_downsampled, we find peaks shifted in 4 times the original peak
% positions. f0=0.005 will be shifted in 0.02, 1-0.02 (no alias);
f1=0.0625 will be
% shifted in 0.25, 1-0.25 (no alias); f2=0.2 will be shifted in 0.8, 1-
0.8=0.2,
% causing aliasing; f3=0.25 will be shifted in 1, 1-1 = 0, causing
% aliasing.

%% 3.

% [3 pt] You have to project a low pass filter in order to filter out (if
present)
% the frequency alias introduced by the downsampling process.
% you have two options:
% (i): generating a low-pass filter using the windowing method.
% (ii): generating a low-pass filter using poles-zeroes.
% Which is the maximum cutoff frequency to select in order to avoid
aliasing?
% Define the low-pass filter h_w using the windowing method having the
maximum
% possible frequency, with order 64.
% From the following zeros-poles diagram, select only the zeroes and poles
% such that you end up with a stable low-pass filter, minimum phase, with
% real coefficients, defined as h_zp. (see exam text).
% Find the numerator and denominator of the selected filter H_zp such that
% the filter gain in frequency = 0 is equal to 1.

% the maximum cutoff frequency is given by 1/(2M) = 0.125.
f_cutoff = 1/(2*M);

% low-pass filter with the windowing method:
cutoff_filter = 2*f_cutoff;
h_w = fir1(64, cutoff_filter);

% Select only the zeroes and poles such that you end up with a
% stable low-pass filter, minimum phase, with real coefficients.
% stable --> all poles must be inside the unit circle.
% low-pass --> in frequency = 0, we do not want zeroes, but only poles. In
% frequency = 0.5, we do not want poles, but only zeroes.
% minimum phase --> all zeroes must be inside the unit circle.
% real coefficients --> discard complex zeroes which do not occur in
% complex conjugate pairs.

zeroes = [0.95*exp(2*pi*1i*0.125); 0.95*exp(-2*pi*1i*0.125); -0.95; ...
    0.99*exp(2*pi*1i*0.275);0.99*exp(-2*pi*1i*0.275)];
poles = [0.95];
```

```
A_zp = poly(poles);
B_zp = poly(zeroes);

% Find the numerator and denominator of the selected filter H_zp such that
% the filter gain in frequency = 0 is equal to 1.
% In order to have gain = 1 in f = 0, we should impose that:
% k*B_zp(z=1)/A_zp(z=1) = 1 --> k = sum(A_zp)/sum(B_zp).
k = sum(A_zp) / sum(B_zp);
% multiply the numerator by k
B_zp = k * B_zp;

%% 4.

% [3.5 pt] Filter the signal x with h_w and h_zp, defining x_w and x_zp.
% Plot the 3 signals (x, x_w, x_zp) in the same figure.
% Looking at the temporal behaviour of the signals, which is, among x_w
and
% x_zp, the signal with the most accentuated low-pass behaviour?
% Investigating the three signals also in the frequency domain, which is,
in your opinion, the
% best filter of the two? Motivate your answer.
% Select this filter and decimate the original signal x.
% Compute the DFT of the decimated signal and check if frequency alias is
% maintained or deleted.
% (compare the DFTs of the decimated and of the downsampled signals).

% Filter the signal x with h_w and h_zp, defining x_w and x_zp.

x_w = filter(h_w, 1, x);
x_zp = filter(B_zp, A_zp, x);

% Plot the 3 signals (x, x_w, x_zp) in the same figure.
% Looking at the temporal behaviour of the signals, which is, among x_w
and
% x_zp, the signal with the most accentuated low-pass behaviour?

leg = {};
figure;
plot(x);
leg{1} = 'Original signal';
hold on;
grid;
plot(x_w);
leg{2} = 'Signal filtered with h_w';
plot(x_zp);
leg{3} = 'Signal filtered with h_{zp}';
legend(leg);
```

```
% The signal with the most low-pass behaviour is that filtered with h_zp.

% Investigating also the frequency domain, which is, in your opinion, the
% best filter of the two?
% Compute the signals dft
X = fft(x);
X_w = fft(x_w);
X_zp = fft(x_zp);

norm_freq_axis = 0: 1/N:1 - 1/N;
figure;
leg = {};
% original signal
stem(norm_freq_axis, abs(X));
leg{1} = 'DFT of the Original signal';
hold on;
% fir signal
stem(norm_freq_axis, abs(X_w));
leg{2} = 'DFT of the Original signal filtered with h_w';
% iir signal
stem(norm_freq_axis, abs(X_zp));
leg{3} = 'DFT of the Original signal filtered with h_{zp}';

% One good option could be to choose h_w. It filters out the frequency
% components after 1/(2M), at the same time not attenuating too much
signal
% content at lower frequencies.

% Select this filter and decimate the original signal x.
x_dec = x_w(1:M:end);

% Compute the DFT of the decimated signal and check if frequency alias is
% maintained or deleted with respect to the downsampling scenario.
% (compare the DFTs of the decimated and of the downsampled signals).

X_dec = fft(x_dec);
X_downsampled = fft(x_downsampled);
figure;
leg = {};
plot(0:1/length(x_dec):1-1/length(x_dec), abs(X_dec));
leg{1} = 'DFT of the Decimated signal';
hold on;
plot(0:1/length(x_downsampled):1-1/length(x_downsampled),
abs(X_downsampled), '--');
leg{2} = 'DFT of the Downsampled signal';
legend(leg);

% The frequency alias is no more present.
```