MultimediaSignal Processing 1st Module

MATLAB part [12 pts]

January 15th, 2021

Text:

1. [2 pt] The signal z is the sum of two cosinusoidal signals, x and y. y has an angular frequency = 1000*pi [rad/second] and x repeats periodically every 5 entire cycles of y. The signal durations correspond to 10 periods of the signal z.
2. [2.5 pt] You have to design a multirate system to pass from a sampling rate of 4 KHz to 3 KHz. In particular, we want a low pass filter that should pass unaltered all the sinusoidal signals with period greater or equal to 32 samples.
    - Which is the minimum possible cut-off frequency of this filter?
    - Given the above constraint on the low pass filter, which is the maximum downsampling factor M which we can use to guarantee the absence of alias after the conversion process?
    - In this case, select a value of M equal to half the maximum M.
    - Select the low pass filter with the maximum possible cutoff frequency.
    - Choose the value of the upsampling factor L according to the selected M value.
3. [1.5 pt] Convert the sampling rate from 4KHz to 3 KHz. Use 64 samples for the low pass filter. Define the final signal as z_multirate1, and define all the signals derived at the main steps of the multirate conversion process (you may need to define, for example, the filtered signal, the downsampled and/or the upsampled signal, etc..)
4. [3 pt] Compute the DFTs of the signal z, of the signals defined at the intermediate steps of the multirate conversion, and of the signal z_multirate1. In all cases, use N = 2048 samples to compute the DFTs.
    - Plot the DFTs versus normalized frequencies, centering the DFTs around frequency 0.
    - Comment the position of the DFT peaks for each plotted signal.
5. [2 pt] Repeat what done in point 3, inverting the position of the downsampling with the upsampling, while maintaining the position of the filtering process. Define all the signals involved in the process and define the final output signal as z_multirate2.
    - Compute the DFTs of all the signals over N samples and plot the DFT amplitudes of the signals versus the same frequency axis defined before.
    - Is the DFT amplitude of z_multirate2 equal to that of z_multirate1? If yes/no, why?
    - Focus on the peaks of the DFT amplitude of z_multirate2: are they due to signal x or y? Are these peaks originated by alias or not? Why?
6. [1 pt] Choose the minimum possible value of M (integer) to make the multirate conversion process.

- Perform the multirate conversion, defining the signal z_multirate3 (hint: be careful in defining the cutoff frequency of the filter).
- Check (in a new figure) if the peaks of the absolute value of DFT of z_multirate3 match with those of DFT of z_multirate1

Solution:

```
close all
clearvars
clc

%% 1.

% [2 pt] The signal z is composed by the sum of two cosinusoidal signals,
x and y.
% In particular, y has an angular frequency = 1000*pi [rad/s]
% and x repeats periodically every 5 entire cycles of y.
% The signal duration is equal to 10 periods of the signal z.

Fs = 4000; % defined in the next point
period_y = 1/500; % = 1000pi/(2pi)
period_x = 5 * period_y;
period_z = period_x; % least common multiple between (period_x, period_y)
= period_x
t = 0:1/Fs:10*period_z - 1/Fs;
y = cos(1000*pi*t);
x = cos(2*pi/period_x *t);
z = x + y;

%% 2.

% [2.5 pt] You have to design a multirate system to pass from a sampling
rate of 4 KHz to 3 KHz.

% In particular, we want a low pass filter that should pass unaltered all
% the sinusoidal signals with period greater or equal to 32 samples.
% Which is the minimum possible cut-off frequency of this filter?
period_min = 32;
% f_cutoff >= 1/period_min
f_cutoff_min = 1/period_min;

% Since we have to convert the sampling rate from 4KHz to 3Khz, the ratio
% L/M should be 3/4 (or a multiple of it) --> the most constraining factor
% for the cutoff frequency is M, not L. So, we have to impose also that
% f_cutoff <= 1/(2M)

% Given the above constraint on the low pass filter,
% which is the maximum downsampling factor M which we can use to guarantee
% the absence of alias after the downsampling process?
% f_cutoff <= 1/2M --> M<= 1/(2*f_cutoff) --> max(M) = 1/(2*f_cutoff_min)
M_max = 1/(2*f_cutoff_min);
```

```
% In this case, select a value of M equal to half the maximum M, then
% select the low pass filter with the maximum possible cutoff frequency.
% Choose the value of L according to the selected M value.
M = M_max /2;
f_cutoff = 1/(2*M); % f_cutoff <= 1/2M --> max(f_cutoff) = 1/(2M)
L = 3/4*M;

%% 3.

% [1.5 pt] Convert the sampling rate from 4KHz to 3 KHz. Use 64 samples
for the low pass filter.
% Define the final signal as z_multirate1, and define all the signals
% derived at the main steps of the multirate conversion process (you
% may need to define, for instance, the filtered signal, the
% downsampled and/or the upsampled signal, etc..)

% upsampling
z_up = zeros(1, length(z) * L);
z_up(1:L:end) = z;

% filtering
cutoff_filter = 2*f_cutoff;
h = L*fir1(63, cutoff_filter);
z_f = filter(h, 1, z_up);

% downsampling
z_multirate1 = z_f(1:M:end);

%% 4.

% [3 pt] Compute the DFTs of the signal z, of the signals defined at the
% intermediate steps of the multirate conversion, and of the signal
% z_multirate1. In all cases, use N = 2048 samples to compute the DFTs.

N = 2048;
Z = fft(z, N);
Z_up = fft(z_up, N);
Z_f = fft(z_f, N);
Z_multirate1 = fft(z_multirate1, N);

% plot the DFTs versus normalized frequencies, centering the DFTs around
frequency 0

norm_freq_axis = [0:1/N:1 - 1/N] -0.5;

figure(1);
leg = {};
% original signal
plot(norm_freq_axis, fftshift(abs(Z)));
```

```
leg{1} = 'Original signal';
grid;
hold on;
% upsampled signal
plot(norm_freq_axis, fftshift(abs(Z_up)));
leg{2} = 'Upsampled signal';
% filtered signal
plot(norm_freq_axis, fftshift(abs(Z_f)));
leg{3} = 'Filtered signal';
% final multirate
plot(norm_freq_axis, fftshift(abs(Z_multirate1)));
leg{4} = 'Multirate1 converted signal';
legend(leg);

% Comment the position of the DFT peaks for each plotted signal.
% The signal z has 4 peaks in fx = 0.025, -fx, fy = 0.125, -fy
% The signal z_up has the spectrum compressed by a factor L = 6, and the
% replicas repeated every 1/L.
% The signal z_f contains only the spectrum of z_up for f <= f_cutoff, so
% the replicas are deleted.
% The signal z_multirate1 is expanded by a factor M = 8.

%% 5.

% [2 pt] Repeate point 3, inverting the position of downsampling with the
% upsampling, while maintaining the position of the filtering process.
% Define all the signals involved in the process and the final output
% signal as z_multirate2.
% Compute again the DFTs of all the signals over N samples. Plot the
% DFT amplitudes of the signals versus the same frequency axis defined
before.

% downsampling
z_down = z(1:M:end);

% filtering
z_f = filter(h, 1, z_down);

% upsampling
z_multirate2 = zeros(1, length(z_f) * L);
z_multirate2(1:L:end) = z_f;

Z_down = fft(z_down, N);
Z_f = fft(z_f, N);
Z_multirate2 = fft(z_multirate2, N);

figure(2);
leg = {};
% original signal
```

```
plot(norm_freq_axis, fftshift(abs(Z)));
leg{1} = 'Original signal';
grid;
hold on;
% downsampled signal
plot(norm_freq_axis, fftshift(abs(Z_down)));
leg{2} = 'Downsampled signal';
% filtered signal
plot(norm_freq_axis, fftshift(abs(Z_f)));
leg{3} = 'Filtered signal';
% final multirate
plot(norm_freq_axis, fftshift(abs(Z_multirate2)));
leg{4} = 'Multirate2 converted signal';
legend(leg);

% Is the DFT amplitude of z_multirate2 equal to that of z_multirate1?
% If yes/no, why?
% They are not equal, as inverting the downsampling with the upsampling
% does not return the same result. Considering the downsampling block as
first
% operation inevitably causes a loss of information on the signal.

% Focus on the peaks of the DFT amplitude of z_multirate2:
% to which of the two initial signals (x or y) are they due?
% Are these peaks originated by alias or not? Why?
% The peaks are due to the signal y. Indeed, the downsampling operation
causes an
% expansion of the spectrum by a factor M=8. The peaks of the signal x
will
% be in f=+- 0.2, while those of signal y will be in f=+- 1. Since the DFT
% is periodic with period = 1 in normalized frequency domain, the peaks
% related to y will be reported also in f=0, creating aliasing components.
% Then, the low pass filter will pass only the frequency components <=
% 1/2M, and so only the aliasing peak in f=0 remains.
% Then, with the upsampling, the spectrum is compressed by a factor L=6,
so
% we will see peaks repeated every 1/L. These peaks are all due to
aliasing
% components introduced by the signal y.

%% 6.

% [1 pt] Choose the minimum possible value of M (integer) to make the
multirate conversion process.
% Perform the multirate conversion, defining the signal z_multirate3
(hint: be
% careful in defining the cutoff frequency of the filter).
% Check (in a new figure) if the peaks of the absolute value of
% DFT of z_multirate3 match with those of DFT of z_multirate1.
```

```
M = 4; % minimum value of M (integer) to maintain the same multirate
conversion
L = 3; % L (integer) = 3/4*M
f_cutoff = 1/(2*M); % the cutoff frequency depends on M!

% upsampling
z_up = zeros(1, length(z) * L);
z_up(1:L:end) = z;

% filtering
cutoff_filter = 2*f_cutoff;
h = L*fir1(63, cutoff_filter);
z_f = filter(h, 1, z_up);

% downsampling
z_multirate3 = z_f(1:M:end);

% compute the DFT of z_multirate over N = 2048 samples
Z_multirate3 = fft(z_multirate3, N);

% plot the DFTs versus normalized frequencies
figure(3);
leg = {};
plot(norm_freq_axis, fftshift(abs(Z_multirate1)));
leg{1} = 'Multirate1 converted signal';
hold on;
% Z_multirate3
plot(norm_freq_axis, fftshift(abs(Z_multirate3)), '--');
leg{2} = 'Multirate3 converted signal';
legend(leg);
leg = {};

% The peaks perfectly match. Indeed, the operation is the same, even if in
% case (1) we used L = 6 and M = 8, while in case (3) we used L = 3 and M
=
% 4. The ratio L/M remains the same.
```