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

date: January 17, 2023

Ex.1 (Pt.12)

A digital filter $h[n]$ satisfies the following equation:

$$y[n] = x[n] + x[n-2] - 0.81y[n-2].$$

It is applied to a continuous signal $\overline{x}(t) = 3\sin(2\pi 100t) + \cos(2\pi 150t)$ sampled at 400 sps (samples per seconds).

[3 pts] Draw the poles-zeros plot.

[4 pts] Depict an approximate amplitude response of the filter $|H(z)|$.

[5 pts] Calculate the exact output signal $\overline{y}[n]$ for the input signal $\overline{x}[n]$ properly evaluating the filter effect on the signal components.

Ex.2 (Pt.12)

The previous signal $\overline{x}(t) = 3\sin(2\pi 100t) + \cos(2\pi 150t)$ sampled at 400 sps (samples per seconds) has to be upsampled to 1kHz.

[2 pts] Describe the procedure in order to change the signal samples rate detailing the upsampling and/or the downsampling steps and the properties of the adopted filters.

[3 pts] Draw the signal representation in the frequency domain (in the range of normalized frequencies between 0 and 1/2).

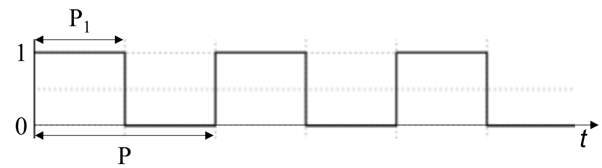
[3 pts] In case an ideal low pass filter $H(f)$ should be available define its cut-off frequency to be used in this case and draw the signal after the downsampling stage.

[4 pts] In case a real filter should be used a linear interpolator will be adopted: define the filter in the time domain and represent the first 10 samples of the output resampled signal.

CONTINUES ON THE BACK

Ex.3 (Pt.12)

1) [2 pt] In a system working at sampling rate 1KHz, define a square wave s(t) like the one in the figure, with period P = 80 samples, duty cycle (= P1/P) = 50% and duration = 1 sec.



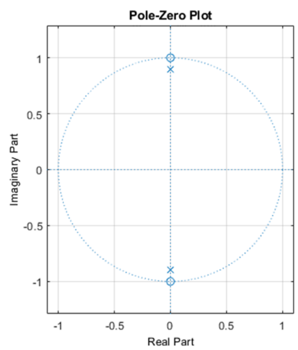
2) [2 pt] The signal s(t) enters an LTI system with an impulse response equal to the first P1 samples (i.e., the duty cycle) of the signal itself.
   - Find the output signal s_out(t) to the system exploiting the function *overlap_and_save.m* defined in this way:
     output_signal = overlap_and_save(input_signal, LTI_filter, block_length, overlap), where the output_signal has the same duration of the input_signal. You choose the block length and the number of samples for the overlap. NOTE: you don't have to write the function code, give it for granted.
   - What do you expect to see as output signal? Which is the period of the output signal?
3) [2 pt] Compute the cyclic convolution between the signal and the impulse response of the system, over a number of samples equal to the period of s(t).
   - What do you expect to see as output signal?
   - Which is the number of samples to use in the cyclic convolution in order to obtain a constant signal as output?
4) [5 pt] The signal s(t) is multiplied by the signal x(t), defined as x(t) = cos(80*pi*t) + cos(100*pi*t), obtaining the signal y(t).
   - Plot the absolute value of the DFT of the signals x(t) and y(t) as a function of the frequency in Hz.
   - What do you expect from X(f)? And from Y(f)? Do we see the exact theoretical spectrum of the sum of two cosinusoidal signals? Explain and motivate your answers.
   - Select the lowest possible duty cycle of s(t) and the maximum possible number of signal samples such that the behaviour of Y(f) coincides with that of X(f).

Solutions

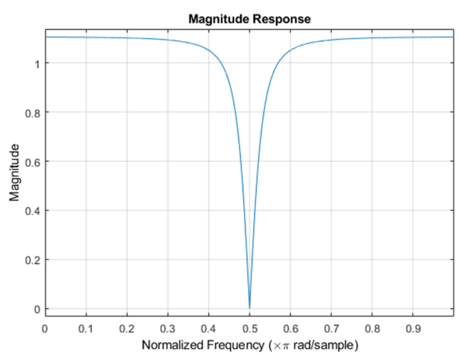
Ex.1

$$H(z)=\frac{1+z^{-2}}{1+0.81z^{-2}}$$

The poles-zeros plot is the following.



The amplitude response is the following:



The sampled signal will be:

$$\bar{x}[n]=3\sin\left(2\pi\frac{100}{400}n\right)+\cos\left(2\pi\frac{150}{400}n\right)=3\sin\left(\frac{\pi}{2}n\right)+\cos\left(\frac{3}{4}\pi n\right)$$

Due to the filter behavior the component at $f=\frac{1}{4}$ (at normalized pulsation $\omega=\frac{\pi}{2}$) is completely removed while the component at $f=\frac{3}{8}$ (at normalized pulsation $\omega=\frac{3}{4}\pi$) will have the following amplitude:

$$\left|H\left(z=e^{j\frac{3}{4}\pi}\right)\right|=\left|\frac{1+e^{-j\frac{3}{2}\pi}}{1+0.81e^{-j\frac{3}{2}\pi}}\right|=\frac{|1+j|}{|1+0.81j|}=\frac{\sqrt{2}}{\sqrt{1+0.81^2}}\cong 1.1$$

$$\phi = \measuredangle H\left(z = e^{j\frac{3}{4}\pi}\right) = \measuredangle(1+j) - \measuredangle(1+0.81j) = \frac{\pi}{4} - \tan^{-1}(0.81) \cong 0.105rad$$

$$\overline{y}[n] = 1.1\cos\left(\frac{3}{4}\pi n + \phi\right)$$

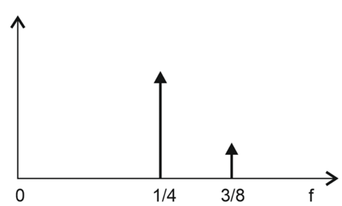
Ex.2

In order to get the signal with the new sample rate the procedure will be based on an upsampling (↑ L=5) a low pass (ideal) filter with cut-off frequency $f_c = \frac{1}{2}\cdot\frac{1}{5} = \frac{1}{10} \rightarrow \omega_c = \frac{\pi}{5}$ and a downsampling (↓ M=2).

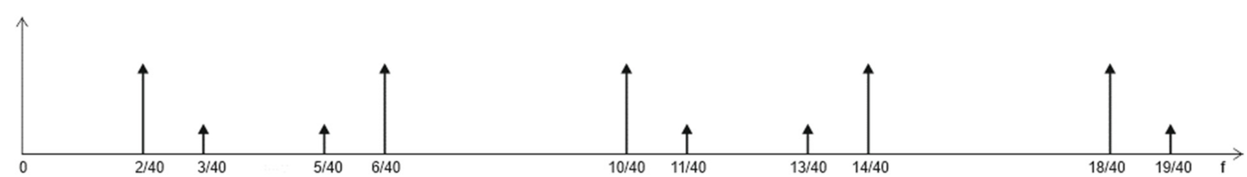
The initial signal can be written:

$$\overline{x}[n] = 3\sin\left(\frac{\pi}{2}n\right) + \cos\left(\frac{3}{4}\pi n\right) = \frac{3}{2j}\left(e^{j\frac{\pi}{2}n} - e^{-j\frac{\pi}{2}n}\right) + \frac{1}{2}\left(e^{j\frac{3}{4}\pi n} + e^{-j\frac{3}{4}\pi n}\right).$$

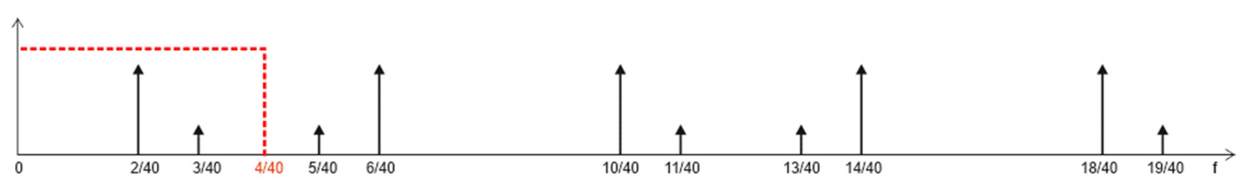
Its frequency representation will be:



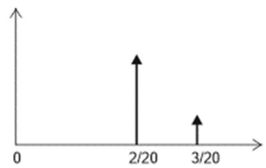
After adding 4 samples equal to ‘0’ every sample the frequency representation will be:



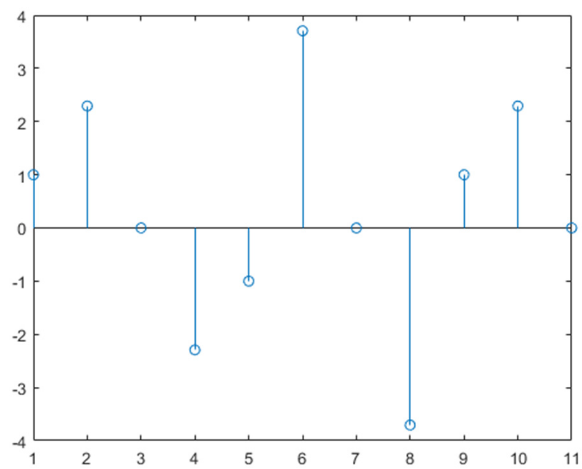
Applying the ideal Low Pass filter:



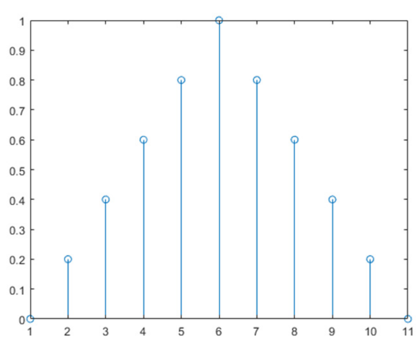
After downsampling:



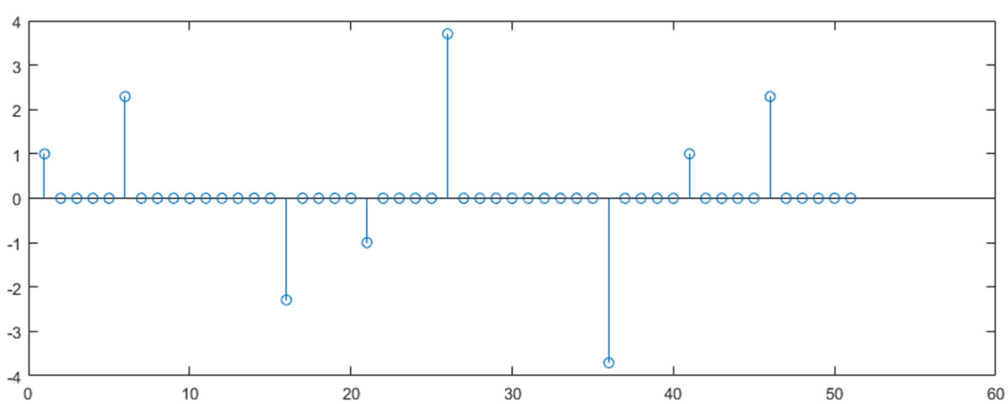
The original signal, for $n = 0:10$,is:



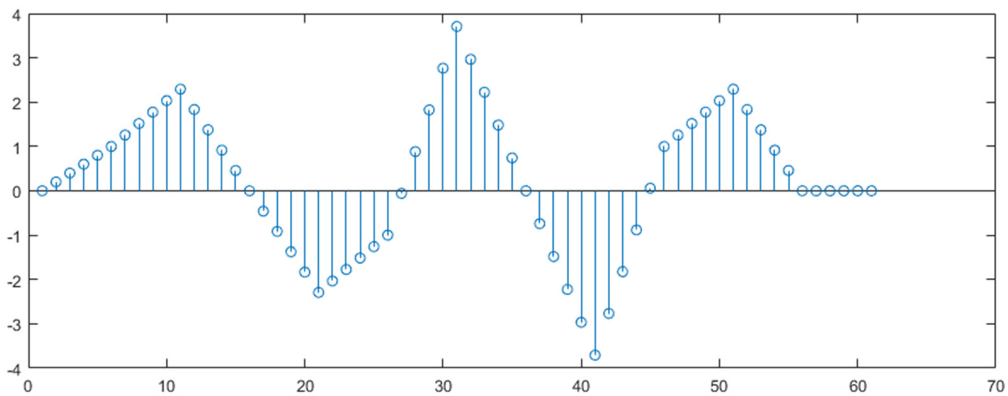
The interpolation low pass filter will be:



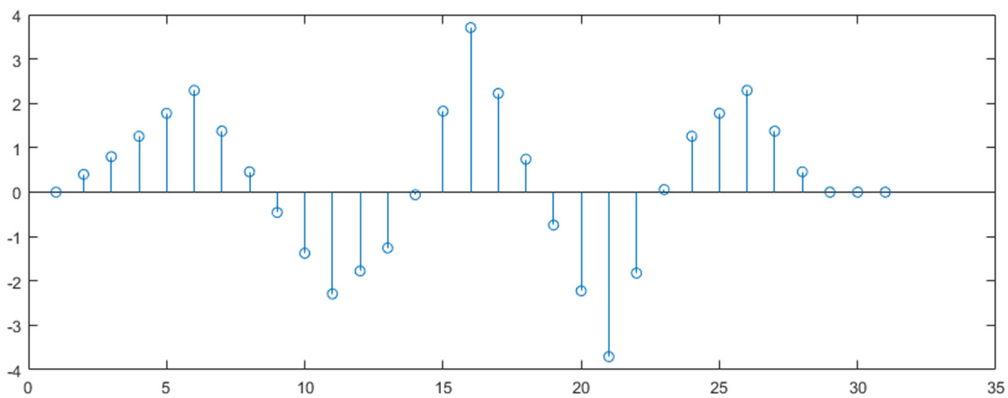
Adding zeros for upsampling the input signal will become:



After the linear interpolation it will be:



After downsampling:



Ex.3 (MATLAB CODE)

```
close all
clearvars
clc

%% 1. [2 pt]

% In a system working at sampling rate 1KHz, define a square wave s(t)
like
% the one in the figure, with period P = 80 samples,
% duty cycle (= P1/P) = 50% and duration = 1 sec.

Fs = 1000;
duration = 1;
% number of signal samples
N = round(Fs * duration);
P = 80;
duty_cycle = round(P/2);

% define a single period
s_singleperiod = [ones(1, duty_cycle), zeros(1, P - duty_cycle)];
```

```
% repeat for a finite number of times, until reaching N samples
N_periods = ceil(N / P);
s = repmat(s_singleperiod, 1, N_periods);
s = s(1:N);

% to better analyze the square wave (not required)
figure;
stem(s);
title('Square wave');
grid;
xlabel('time [samples]');

%% 2. [2 pt]

% The signal s(t) enters an LTI system with an impulse response equal
to the
% first P1 samples (i.e., the duty cycle) of the signal itself.
% Find the output signal s_out(t) to the system exploiting the
function
% overlap_and_save.m defined in this way:
% output_signal = overlap_and_save(input_signal, LTI_filter,
block_length, overlap),
% where the output_signal has the same duration of the input_signal.
% You choose the block_length and the number of samples for the
overlap.
% NOTE: you don,Äôt have to write the function, give it for granted.
% What do you expect to see as output signal?
% Which is the period of the output signal?

% define the impulse response as one duty_cycle of the signal
h = ones(1, duty_cycle);

% choose the block_length (it should be larger than length(h))
block_length = 60;
% the overlap is fixed according to h.
overlap = length(h) - 1;
s_out = overlap_and_save(s, h, block_length, overlap);

% The output signal should be a triangular wave with period P.
% In the part related to the first period, the wave is a rectangular
pulse
% --> the result of the linear convolution with a rectangular pulse is
a
% triangular pulse with length 2*duty_cycle - 1. The next period start
at
% sample = 2*duty_cycle, therefore the result related to the previous
period
% does not overlap with the next samples. Every square-wave period,
% convolved with the rectangular pulse, will result in a triangular
pulse,
% generating a triangular wave with period = P.

% to better analyze the result (not required)
figure;
stem(s_out);
```

```
title('Overlap and save result');
grid;
xlabel('time [samples]');

%% 3. [2 pt]

% Compute the cyclic convolution between the signal and the impulse
response
% of the system, over a number of samples equal to the period of s(t).
% What do you expect to see as output signal?
% Which is the number of samples to use in the cyclic convolution in
% order to obtain a constant signal as output?

s_out_cc = cconv(s, h, P);
% The output signal will be a triangular pulse with length P.
% Indeed, if we consider a number of samples equal to the period, we
find
% the cyclic convolution between two rectangular pulses with duration
=
% duty_cycle, padded with zeros until 2*duty_cycle -1.
% Thanks to the zero padding, we are not introducing artifacts in the
% cyclic convolution.

% to better analyze the result (not required)
figure;
stem(s_out_cc);
title('Cyclic convolution result (1)');
grid;
xlabel('time [samples]');

% If we want to obtain a constant output signal, we should compute the
cyclic
% convolution over a number of samples equal to the duty-cycle.
% This way, the two rectangular pulses are not zero-padded and are
constant
% signals. The cyclic convolution between two constant signals is
again a
% constant.

% to better analyze the result (not required)
s_out_cc_constant = cconv(s, h, duty_cycle);
figure;
stem(s_out_cc_constant);
title('Cyclic convolution result (2)');
grid;
xlabel('time [samples]');

%% 4. [5 pt]

% The signal s(t) is multiplied by the signal x(t), defined as
% x(t) = cos(80*pi*t) + cos(100*pi*t), obtaining the signal y(t).
% Plot the absolute value of the DFT of the signals x(t) and y(t)
% as a function of the frequency in Hz.
% What do you expect from X(f)? And from Y(f)?
```

```
% Do we see the exact theoretical spectrum of the sum of two
cosinusoidal signals?
% Explain and motivate your answers.
% Select the lowest possible duty cycle of s and the maximum possible
number
% of signal samples such that the behaviour of Y(f) coincides with
that of X(f).

% omega_1 = 2*pi*f_1 = 80*pi
f_1 = 40; % Hz
% omega_2 = 2*pi*f_2 = 100*pi
f_2 = 50; % Hz
fn_1 = f_1 / Fs;
fn_2 = f_2 / Fs;
% The signal x(t) must have the same duration of s(t), so that we can
% multiply them.
x = cos(2*pi*fn_1*[0:N-1]) + cos(2*pi*fn_2*[0:N-1]);
y = x.*s;

% to better analyze the result (not required)
figure;
stem(y);
title('Signal y');
grid;
xlabel('time [samples]');

% DFTs
X = fft(x);
Y = fft(y);
freq_axis = 0:Fs/N:Fs - Fs/N;
figure;
% X
stem(freq_axis, abs(X));
title('Absolute value of the DFT of the signal x(n)');
grid;
xlabel('f [Hz]');
% We expect to see 4 peaks in 40 Hz, 50 Hz and their symmetric.
% The spectrum coincides with the theoretical one because the number
of
% samples N is a multiple of the period of x(n).
% Indeed, in order to compute the period of x:
% T1 = 1/0.04 = 25 samples; T2 = 1/0.05 = 20 samples;
% Tx = lcm(25, 20) = 100 samples.

%Y
figure;
stem(freq_axis, abs(Y));
title('Absolute value of the DFT of the signal y(n)');
grid;
xlabel('f [Hz]');
% The signal y(n) contains the contribution of the square wave,
therefore
% the spectrum is not that of the sum of 2 cosine waves, but it is
affected
% by the DFT of the square wave (periodic sinc).
```

```
% To make the spectrum of y(n) coincide with that of x(n), we have to
% satisfy two conditions:
% 1) the number of signal samples should be an integer multiple of the
% period of x(n)
% 2) we should avoid zero padding, because it introduces artifacts in
the
% DFT.
% --> we have to increase the duty-cycle of s until reaching Tx
samples.
% --> to avoid zero padding, the number of signal samples should be
exactly
% equal to the duty-cycle.
% --> the signal y(n) coincides with one single period of x(n),
therefore X(f)
% = Y(f)

Tx = 100;
new_duty_cycle = Tx;
new_N = new_duty_cycle;

% to better analyze the result (not required)
new_P = 2*new_duty_cycle;
new_s = [ones(1, new_duty_cycle), zeros(1, new_P - new_duty_cycle)];
N_periods = ceil(new_N / new_P);
new_s = repmat(new_s, 1, N_periods);
new_s = new_s(1:new_N);
new_y = x(1:new_N).*new_s;

new_Y = fft(new_y);
freq_axis = 0:Fs/new_N:Fs - Fs/new_N;
figure;
stem(freq_axis, abs(new_Y));
title('Absolute value of the DFT of the new signal y(n)');
grid;
xlabel('f [Hz]');
```