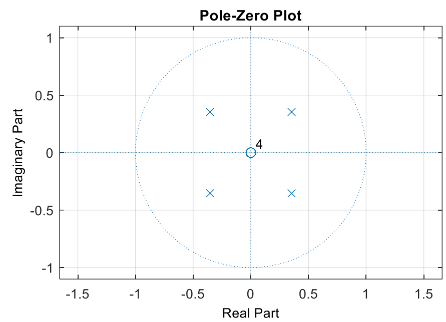
MultimediaSignal Processing 1st Module and
Fundamentals of Multimedia Signal Processing

date: 18/02/2020

Ex.1 (Pt.15)

A filter $H(z)$ has the following zero-pole plot:



Where the distance of every pole from the origin is $\frac{1}{2}$.

[**Pts. 2**] Which kind of filter is it (FIR or IIR)?
Is it stable or unstable? Is it causal or not?

[**Pts. 2**] Provide its z transform : $H(z) = \ldots$

[**Pts. 2**] Provide its difference equation: $y(n) = \ldots$

[**Pts. 3**] Draw a schematic for the filter implementation.

[**Pts. 3**] Provide the first 9 outputs of the filter for the impulse response $x(n) = \{1,0,0,\ldots\}$

[**Pts. 3**] Provide the first 9 outputs of the filter for the step function response $x(n) = \{1,1,1,\ldots\}$

Ex.2 (Pt.7)

A signal $y(t) = 4\cos(2\pi 10t) + 6\cos(2\pi 15t) + 8\cos(2\pi 20t)$ is sampled at 50Hz.

[**Pts. 1**] Represent its spectrum in normalized frequencies in the range $\{0..2\pi\}$

[**Pts. 2**] Represent its spectrum in normalized frequencies in the range $\{0..2\pi\}$ when the signal is downsampled of an order of 2 **without** any antialiasing filter.

[**Pts. 2**] Represent its spectrum in normalized frequencies in the range $\{0..2\pi\}$ when the signal is downsampled of an order of 2 with an ideal antialiasing filter.

[Pts. 2] Represent its spectrum in normalized frequencies in the range $\{0..2\pi\}$ when the signal is upsampled of an order of 2 **without** any smoothing filter.

Ex.3 (Pt. 11 – MATLAB code)

1. [**4 pt**] Define a sinusoidal signal x with amplitude 1.3, frequency 50 Hz, duration 0.205 seconds, sampled every 0.5 msec.
   - Define the period of the signal, expressed in time and in number of samples.
   - Select a number of samples corresponding to the highest possible multiple of the period, defining the signal x_per as the signal x evaluated in these samples (hint: you can use the function “floor” to round a value to the nearest integer less than or equal to that value).
   - Zero-pad the signal x until reaching 1500 samples, defining the signal x_pad.
   - For each of the three signals, compute the DFT over the exact number of signal samples and plot the “stem” of the DFT magnitude as a function of frequency. Is there any difference between the three DFTs? If yes, why? Comment on what you expect.
2. [**4 pt**] Consider a signal x1 with normalized frequency 0.3, same duration and amplitude of x. Define the signal y = x + x1. Exploiting a FIR filter with order 64,
   - Downsample the signal y by a factor M = 2, defining the signal y_down.
   - Decimate the signal y by a factor M = 2, defining the signal y_dec.
3. [**3 pt**] Compute and plot the magnitude of the DFTs of y, y_down and y_dec using N = 2048 samples, as a function of normalized frequency in [0, 1).
   - Which is the difference between the DFTs of y, y_down and y_dec?

Solutions

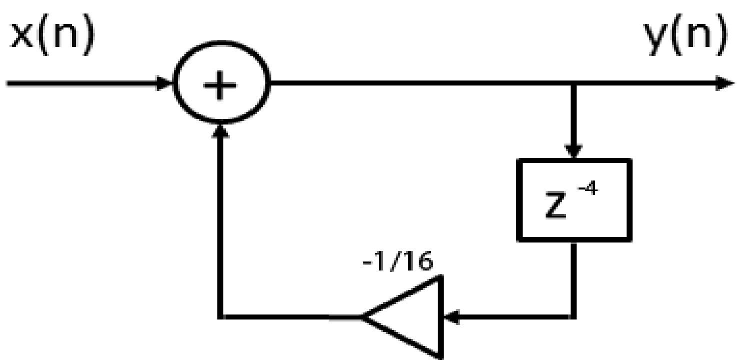
Ex.1

The filter is an IIR (pure IIR), stable and causal (since the number of zeros, to be causal, cannot be greater than the number of poles).

$$H(z)=\frac{1}{\left(1-\frac{1}{2}\cdot 2\cdot\frac{\sqrt{2}}{2}z^{-1}+\frac{1}{4}z^{-2}\right)\cdot\left(1+\frac{1}{2}\cdot 2\cdot\frac{\sqrt{2}}{2}z^{-1}+\frac{1}{4}z^{-2}\right)}=$$

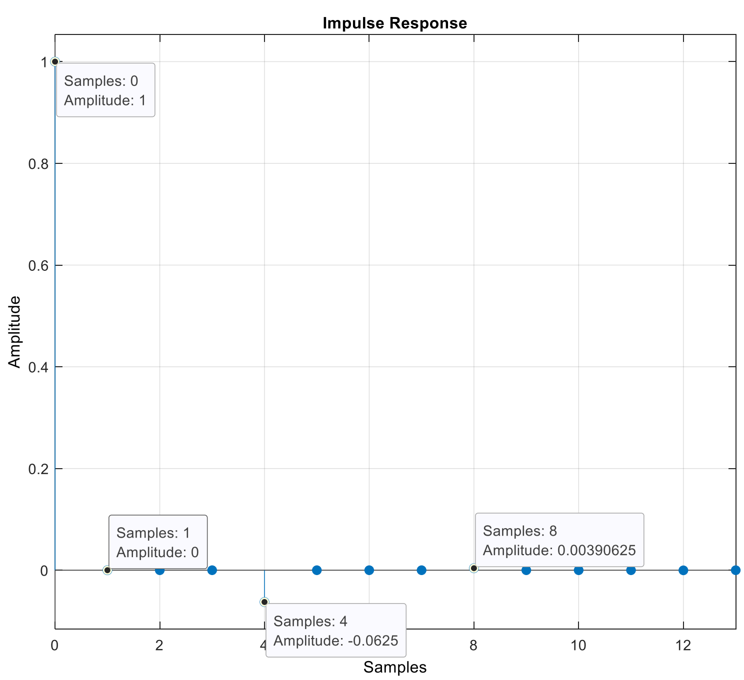
$$=\frac{1}{1+\frac{1}{16}z^{-4}}$$

$$y(n)=x(n)-\frac{1}{16}y(n-4)$$



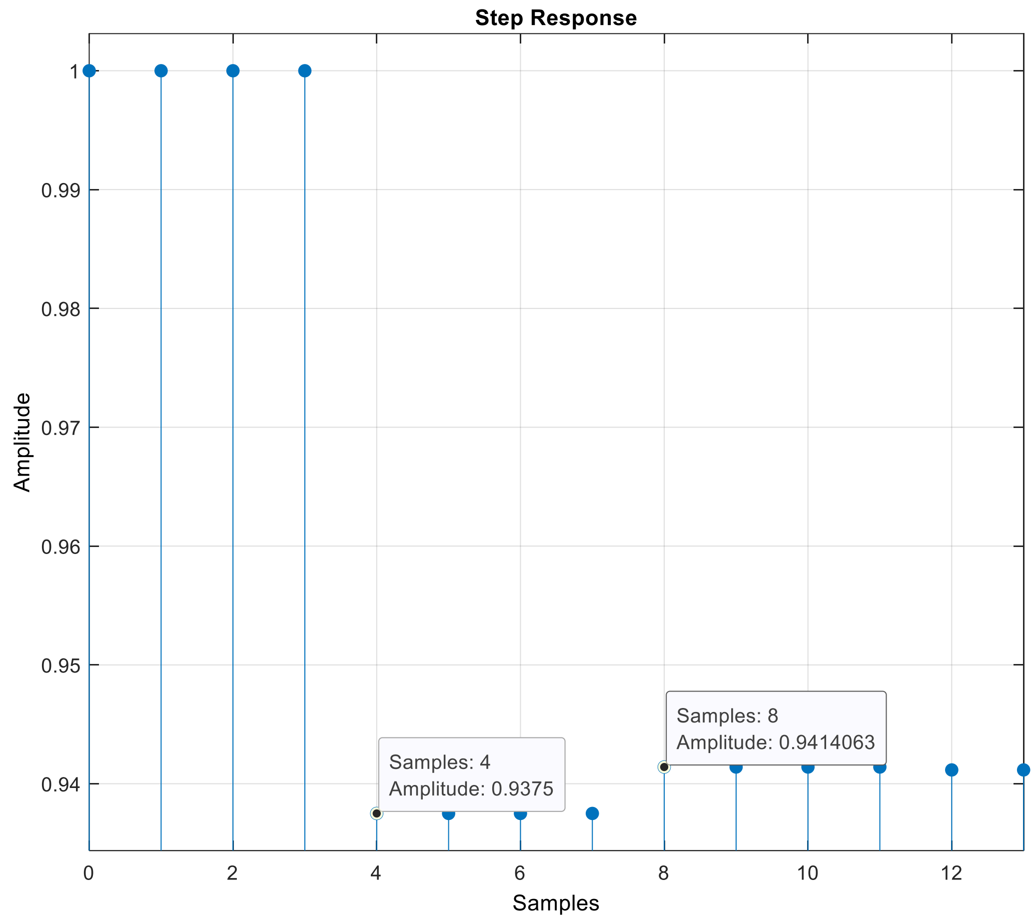
Impulse response will be:

$$y(n)=\left\{1,0,0,0,-\frac{1}{16},0,0,0,\left(-\frac{1}{16}\right)^2,0,0,0,\ldots\right\}$$



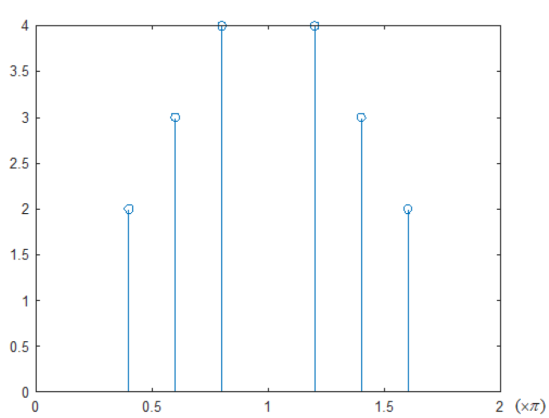
Step response will be:

$$y(n)=\left\{1,1,1,1,\frac{15}{16},\frac{15}{16},\frac{15}{16},\frac{15}{16},1-\frac{1}{16}\left(\frac{15}{16}\right),\frac{241}{256},\frac{241}{256},\frac{241}{256},\ldots\right\}$$

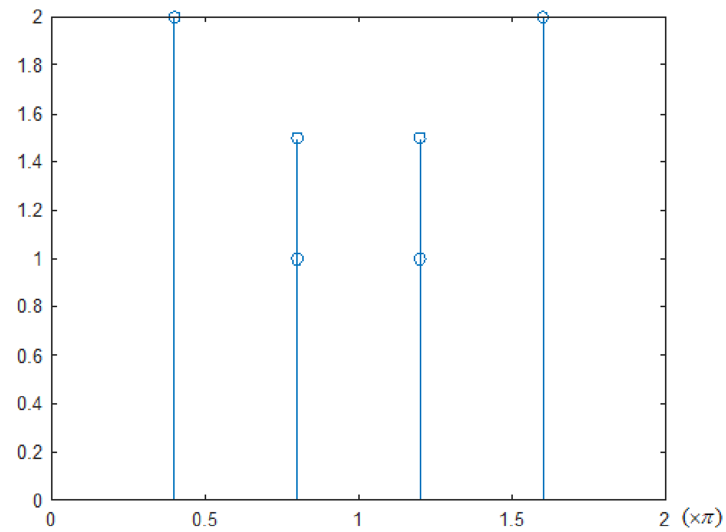


Ex.2

The spectrum of the sampled signal will be:

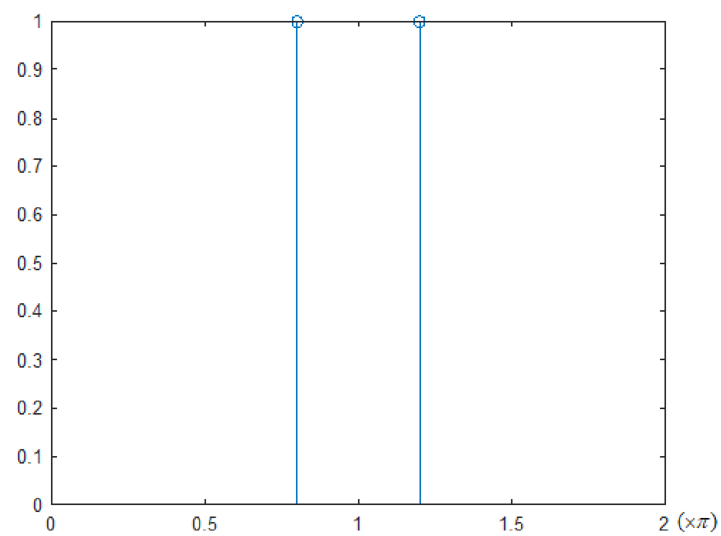


After downsampling there will be aliasing with a partial overlap of the spectra:

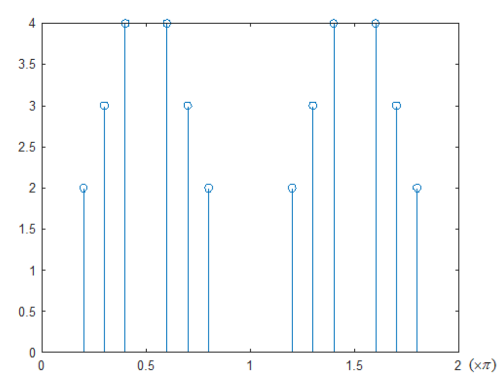


If the original signal is upsampled then the spectrum will be:

The ideal low pass antialiasing filter would cut at 12.5Hz with the consequence that only the sinusoid at 10Hz will pass through:



Upsampling of an order of 2 will give a squeezing of the spectrum:



Ex.3

```
%% 1.

% Define a sinusoidal signal x with amplitude 1.3, frequency 50 Hz,
% duration 0.205 seconds, sampled every 0.5 msec.

ampl = 1.3;
f0 = 50;
Ts = 0.5e-3;
Fs = 1 / Ts;
duration = 0.205;
time = 0:Ts:duration;

x = ampl * cos(2*pi*f0*time);

% Define the period of the signal, expressed in time and in number of
samples.

period_time = 1 / f0;
period_samples = Fs / f0;

% Select a number of samples corresponding to the highest possible multiple
% of the period, defining the signal x_per as the signal x evaluated
% in these samples.

n_per = floor(length(x) / period_samples);
max_multiple = n_per * period_samples;
x_per = x(1:max_multiple);

% Zero-pad the signal x until reaching 1500 samples, defining the signal
x_pad.

x_pad = zeros(1, 1500);
x_pad(1:length(x)) = x;

% For each of the three signals, compute the DFT over the exact number of
% signal samples and plot the "stem" of the DFT magnitude as a function of
frequency.
% Is there any difference between the three DFTs? If yes, why?
% Comment on what you expect.

X_f = fft(x);
N_x = length(X_f);
freq_axis = 0: Fs/N_x:Fs*(N_x- 1)/ N_x;
figure;
stem(freq_axis, abs(X_f));

X_f_per = fft(x_per);
N_x_per = length(X_f_per);
freq_axis = 0: Fs/N_x_per:Fs*(N_x_per- 1)/ N_x_per;
figure;
stem(freq_axis, abs(X_f_per));

X_f_pad = fft(x_pad);
N_x_pad = length(X_f_pad);
freq_axis = 0: Fs/N_x_pad:Fs*(N_x_pad- 1)/ N_x_pad;
figure;
stem(freq_axis, abs(X_f_pad));

% the signal x is a sinusoid, but it is not defined on a multiple of its
% period, therefore the DFT X_f will not show just two peaks related to
% the sinusoid, but other frequency components. Indeed, the actual
% peaks due to the sinusoid should be in f = 50 Hz and f = 1950 Hz, but
% we cannot see them in the precise position.
```

```
% In order to see only two peaks, we should evaluate the DFT on the complete
% period (or multiples of it) of the signal, and this is the case of x_per.
% The zero-padding operation can help identifying the actual sinusoid peak
% because it corresponds to interpolating X_f with a periodic sinc, but
% actually does not introduce new information about the signal.

%% 2.

% Consider a signal x1 with normalized frequency 0.3, same duration and
% amplitude of x. Define the signal y = x + x1.

f1_norm = 0.3;
x1 = ampl * cos(2*pi*f1_norm*Fs*time);

y = x + x1;

% Exploiting a FIR filter with order 64:
% Downsample the signal y by a factor M = 2, defining the signal y_down.

M = 2;

y_down = y(1:M:end);

% Decimate the signal y by a factor M = 2, defining the signal y_dec.

lpf = fir1(64, 1/M);
y_filtered = filter(lpf, 1, y);
y_dec = y_filtered(1:M:end);

%% 3.

% Compute and plot the magnitude of the DFTs of y,
% y_down and y_dec using N = 2048 samples, as a function of normalized
% frequency in [0, 1).

N = 2048;

Y_f = fft(y, N);
Y_f_down = fft(y_down, N);
Y_f_dec = fft(y_dec, N);

norm_freq_axis = 0:1/N:(N -1)/N;

figure;
plot(norm_freq_axis, abs(Y_f));
hold on
plot(norm_freq_axis, abs(Y_f_down));
hold on
plot(norm_freq_axis, abs(Y_f_dec));

% Which is the difference between the DFTs of y, y_down and y_dec?

% The DFT of y contains four peaks in normalized frequencies 0.025, 0.975,
% 0.3 and 0.7.
% The DFT of y_down is expanded by a factor 2 and repeated with period 1,
% therefore we will have peaks in 0.05, 0.6, 1.4, 1.95, periodically repeated
% with period 1. In the interval [0, 1) we will see peaks in 0.05, 0.6, 0.4
and 0.95.
% y_dec has been filtered to avoid frequency aliasing, therefore we will see
only
% frequency components < 1 / 2M. Only the sinusoid at 50 Hz survives the
filtering.
```