Multimedia Signal Processing 1st Module

Date: July 3, 2026

Ex.1 (Pt.12)

A signal in the time domain has the following expression:

$$x[n] = \left(\frac{5}{4}\right)^n \cos\left(\frac{\pi}{2}n\right) + \left(\frac{1}{2}\right)^n \sin\left(\frac{3\pi}{4}n\right) + 1$$

Using either an IIR or a FIR filter $H(z)$ - choose one and justify your choice - we want to completely remove just the unstable components preserving as much as possible the stable ones.

- Define the z transform of the filter (motivating the answer)
- Plot the zeros-poles plot of the filter
- Provide the expression $H(\omega)$ of the filter response at different normalized frequencies ω and represent an approximated behavior of its magnitude $|H(\omega)|$

Ex.2 (Pt.9)

A signal with the highest frequency component at 50kHz is sampled at 400kHz and then downsampled of an order of 4.

- Define the ideal pipeline to perform this digital processing discussing if aliasing or other artifacts are present in this case (justifying the answer).
- Represent the spectrum in <u>normalized</u> frequencies (reporting the frequency values in the abscissas):
    1. before, the downsampling,
    2. after downsampling.
    3. Describe the ideal filter to be adopted.

Ex.3 on the back

Ex.3 (Pt. 11 – MATLAB code)

1. [2 pt] x and y are cosinusoidal signals, sampled with Fs = 5KHz and having a total number of samples = 1500. Their normalized frequencies are, respectively, 0.02 and 0.2. Define the signal z = x + y.
2. [3.5 pt] The filter H(z) has a denominator A(z) = (1 - 0.99*exp(1i*2*pi/5))(1 - 0.99*exp(-1i*2*pi/5)). The numerator B(z) = (1-exp(phase_b_1))(1-exp(phase_b_2)).
    - Which are the values "phase_b_1" and "phase_b_2" in order to have a filter, with real coefficients, that filters out from z the sinusoid with higher frequency?
    - In doing so, which filter do we obtain?
    - Define the filter (find its numerator and denominator in closed form) and plot its zeroes and poles in the complex plane.
3. [2 pt] Filter the signal z with H(z), obtaining z_f, and plot it as a function of its samples.
    - Among x and y, which one of the two is similar to z_f? Why? Motivate your answer.
    - In the same figure, plot the chosen signal (x or y) in order to verify your choice.
4. [3 pt] Decimate the signal z_f by a factor 10, defining the signal z_dec. If needed, exploit an FIR filter with 65 samples.
    - Plot the magnitude of the frequency response of the decimated signal as a function of normalized frequencies.
5. [1.5 pt] Plot z_dec as a function of its samples.
    - Among x and y, which one of the two is similar to z_dec? Why? Motivate your answers.
    - In the same figure, plot the chosen signal (x or y) in order to verify your choice. Plot the chosen signal using the same number of samples of z_dec.
    - If you plot the two signals, there is still one small difference between z_dec and the other signal. Apart from their amplitudes, the two signals are shifted in time. Which is the reason behind this?

<u>Motivate your answers.</u>

Solutions

Ex.1

The first term is unstable so we need to remove it: a two-zeros FIR filter can remove it.

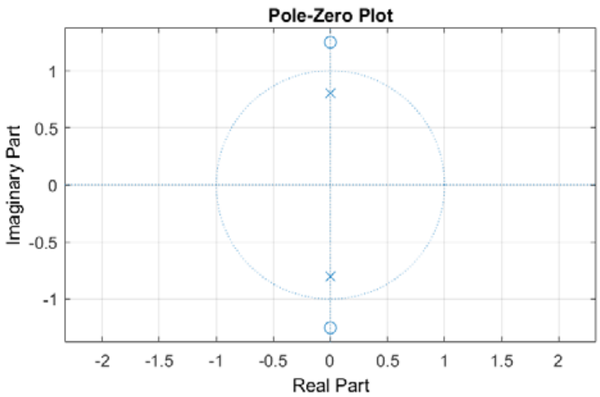
The position of the two zeros has to overlap the unstable frequencies of the signal: The normalized pulsations are $\omega = \pm\frac{\pi}{2}$ and $\rho = \frac{5}{4}$

Then: $H(z) = 1 - 2\rho\cos(\omega)z^{-1} + \rho^2 z^{-2} = 1 + \frac{25}{16}z^{-2}$

To preserve as much as possible of the original signal (at least from the magnitude part) a possible solution could be to configure it as an all-pass IIR Filter; this requires the adoption of a couple of poles in reciprocal conjugate position with respect to the zeros:

$$H(z) = \frac{16}{25} \cdot \frac{1 + \frac{25}{16}z^{-2}}{1 + \frac{16}{25}z^{-2}}$$

The amplitude response will be flat with amplitude: 1



Ex.2

The ideal digital processing chain is:

x(t) -> sampling at 400 kHz -> x[n] -> ideal anti-aliasing low-pass filter -> v[n] -> downsample by 4 -> y[m]

Before decimation, a digital anti-aliasing low-pass filter must be applied. For decimation by M = 4, the allowed digital bandwidth before downsampling is $\frac{f_s}{2M} = 50kHz$

Since the original signal already has $f_{max} = 50kHz$, it exactly reaches the final Nyquist frequency after downsampling. Therefore, in the ideal mathematical case there is no aliasing, but the case is critical because there is no guard band. In a real implementation, a cutoff slightly below 50 kHz would be safer to allow a finite transition band.

Ex.3

```
clearvars

close all

clc


%% 1 [2 pt]


% x and y are cosinusoidal signals, sampled with Fs = 5KHz and having a

% total number of samples = 1500. Their normalized frequencies are,

% respectively, 0.02 and 0.2. Define the signal z = x + y.


f_x = 0.02;

f_y = 0.2;

N = 1500;

Fs = 5000;

time = 0:1/Fs:(N-1)/Fs;

x = cos(2*pi*f_x*Fs*time);

y = cos(2*pi*f_y*Fs*time);


z = x + y;


%% 2 [3.5 pt]


% The filter H(z) has a denominator A(z) = (1 - 0.99*exp(1i*2*pi/5))(1 -

% 0.99*exp(-1i*2*pi/5)). The numerator B(z) =

% (1-exp(phase_b_1))(1-exp(phase_b_2)).

% Which are the values "phase_b_1" and "phase_b_2" in order to have a filter,
```

```
% with real coefficients, that filters out from z the sinusoid with higher frequency?
% In doing so, which filter do we obtain?
% Define the filter (find its numerator and denominator in closed form)
% and plot its zeroes and poles in the complex plane.

% phase_b_1 = - phase_b_2 in order to have a filter with real coefficient.
% phase_b_1 = 1i*2*pi*f_y in order to delete this sinusoidal component.
phase_b_1 = 1i*2*pi*f_y;
phase_b_2 = - phase_b_1;

% The filter is a notch

% let's exploit the convolution property
A_1 = [1, -0.99 * exp(1i*2*pi/5)];
A_2 = [1, -0.99 * exp(-1i*2*pi/5)];
A = conv(A_1, A_2);

B_1 = [1, -exp(phase_b_1)];
B_2 = [1, -exp(phase_b_2)];
B = conv(B_1, B_2);

figure;
zplane(B, A);
legend('roots of H(z)');

%% 3 [2 pt]

% Filter the signal z with H(z), obtaining z_f, and plot it as a function of its samples.
% Among x and y, which one of the two is similar to z_f? Why? Motivate your answer.
% In the same figure, plot the chosen signal (x or y) in order to verify your choice.



z_f = filter(B, A, z);
```

```
% the correct answer is x. Indeed, the filter H(z) should filter out the
% y-component from signal z.

figure;
plot(z_f);
hold on;
plot(x);
grid;
legend('z_f', 'x');

%% 4 [3 pt]

% Decimate the signal z_f by a factor 10, defining the signal z_dec.
% If needed, exploit an FIR filter with 65 samples.
% Plot the magnitude of the frequency response of the decimated signal
% as a function of normalized frequencies.

M = 10;
lpf = fir1(64, 1/M);

% low-pass filtered signal
z_lp = filter(lpf, 1, z_f);

% decimate the signal
z_dec = z_lp(1:M:end);

% FFT
F_dec = fft(z_dec);
N_dec = length(F_dec);

figure;
plot(0:1/N_dec:(N_dec-1)/N_dec, abs(F_dec));
title('Z_{dec}(f)');
```

```
%% 5 [1.5 pt]

% Plot z_dec as a function of its samples.
% Among x and y, which one of the two is similar to z_dec? Why? Motivate your answers.
% In the same figure, plot the chosen signal (x or y) in order to verify your choice.
% Plot the chosen signal using the same number of samples of z_dec.
% If you plot the two signals, there is still one small difference
% between z_dec and the other signal. Apart from their amplitudes,
% the two signals are shifted in time. Which is the reason behind this?

% the correct answer is y. Indeed, the decimated signal has now
% frequency components in normalized frequencies = 0.2 and 0.8, exactly
% like the signal y.
% there is still one difference between the two signals: z_decimated has a
% delay of 3 samples with respect to y, and this is due to the filtering
% process. Indeed, the filter "lpf" is causal and introduces a delay.
% Since the filter has 65 taps, the peak is in sample # 33. after the
% downsampling, the delay is of round(33/M) taps = 3 taps.

figure;
plot(z_dec);
hold on;
grid;
plot(y(1:length(z_dec)));
legend('z_{dec}', 'y');
```