Practice IV. Decimation and interpolation of 1D signals.

1 Objectives

- Understand the decimation and interpolation process in the temporal domain.
- Characterize decimated/interpolated 1D signals in the frequency domain.

Bring Headphones to the laboratory.

2 Previous Study

2.1 Fourier analysis of decimated sequences

In this section, we review some fundamental ideas of the sampling theory[1]. Let us start with an analog signal x(t) that is sampled taking f_s samples per second. After sampling, we obtain the discrete sequence $x[n]=x(n/f_s)$, whose Fourier transform is given by the following expression:

$$X_d(F)|_{F=f/f_s} = f_s \sum_{k=-\infty}^{\infty} X_a(f - kf_s) \qquad \textit{(Eq. 1)}$$

where $F=f/f_s$ stands for the discrete (normalized) frequency and the sub-indexes 'a' and 'd' are introduced to distinguish between analog and discrete Fourier transforms.

Let us consider now that the sequence x[n] is decimated by a factor M to obtain the sequence:

$$y[n] = x[nM] = x\left(\frac{nM}{f_s}\right)$$

As it is evident in the last expression, y[n] is the sequence that we would have obtained if the original analog signal was sampled with a sampling rate of f_s/M samples per second. Consequently, the Fourier transform of y[n] is just given by

$$Y_d(F)|_{F=Mf/f_s} = \frac{f_s}{M} \sum_{k=-\infty}^{\infty} X_a\left(f - k\frac{f_s}{M}\right) \qquad \textit{(Eq. 2a)}$$

where the (normalized) discrete frequency is redefined as $F=Mf/f_s$ because the sampling frequency has changed. If Eq. 2a is compared with Eq. 1, one concludes that the Fourier transforms $X_d(F)$ and $Y_d(F)$ are related as follows:

$$Y_d(F) = \frac{1}{M} \sum_{k=0}^{M-1} X_d\left(\frac{F}{M} - k\frac{1}{M}\right) \qquad \textit{(Eq. 2b)}$$

From Eq. 2a we can state that, if $X_a(f)$ is not band-limited to ${f_s}/{2M}$, the decimation by M will produce aliasing and the sequence y[n] will not preserve the original spectral composition of x(t). In equivalent terms, Eq. 2b shows that aliasing appears when $X_d(F)$ has spectral components higher than ${1}/{2M}$. Observe that ${1}/{2M}$ is the discrete frequency that corresponds to the analog frequency ${f_s}/{2M}$. Eq. 2b also shows that the original transform $X_d(F)$ overlaps with M-1 different versions of itself; as a consequence, the frequency spectrum of $Y_d(F)$ is originated by the superposition of M different replicas of $X_d(F)$.

[1] For more information, you are referred to slides 4.61 and 4.62 of chapter 4 ("Muestreo, diezmado e interpolación").

To analyze that circumstance, consider the case M=4 and, as an example, a signal x(t) whose Fourier transform $X_a(f)$ is shown in Figure 2.1 (this signal is band limited to ${f_s}/{2}$, as corresponds to a signal sampled without aliasing at a rate f_s; however, as it happens in most cases, the signal is not band limited to ${f_s}/{2M} = {f_s}/{8}$). This figure points out the 4 ***sub-bands*** of the spectrum that, according to Eq. 1, will overlap within the band $\left(-{f_s}/{2M}, {f_s}/{2M}\right)$ when x[n] is decimated to obtain y[n]. That band is shown in Fig. 22 and it corresponds to the interval of discrete frequencies (-0.5, 0.5) of the Fourier transform $Y_d(F)$.

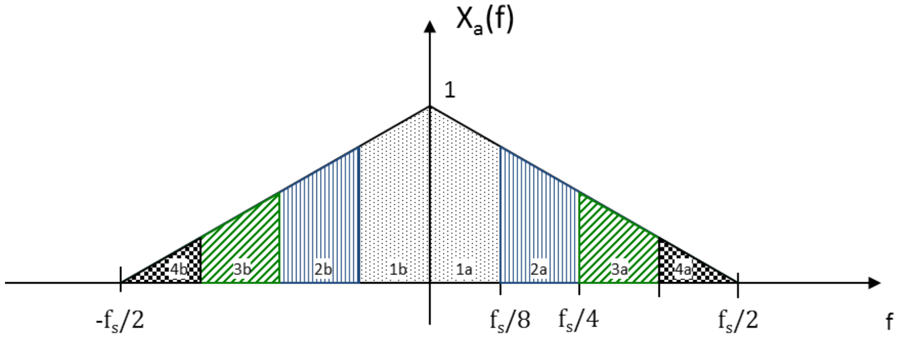


Figure 2.1: Fourier transform of the analog signal x(t). The spectrum is divided into 4 bands (1a, 2a, 3a and 4a) and the corresponding Hermitian bands (1b, 2b, 3b and 4b).

1. Following Eq. 2, represent in different drawings the Fourier transform $X_a(f)$ given in Figure 2.1 and its M-1 (M=4) alias that overlap the band of frequencies $\left(-{f_s}/{2M}, {f_s}/{2M}\right)$ and, in that way, contribute to produce the band $\left(-{f_s}/{2M}, {f_s}/{2M}\right)$ of $Y_d(F)|_{F=Mf/f_s}$. Obtain: a) the sub-bands of $X_a(f)$ that overlap the sub-band 1a, and b) the sub-bands of $X_a(f)$ that overlap the sub-band 1b.

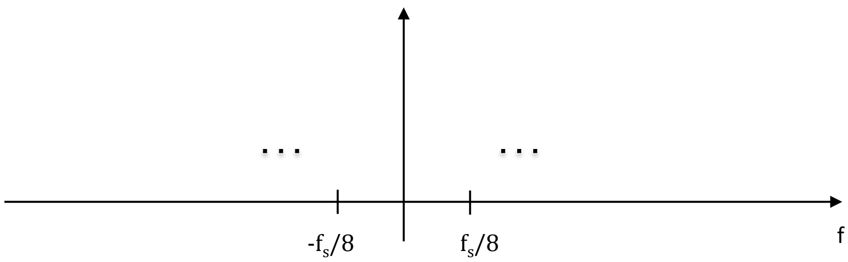


Figure 2.2: Interval of analog frequencies that corresponds to the interval of discrete frequencies -0.5≤F<0.5 of $Y_d(F)$.

2. The answer to question 1 allows us to obtain the four frequency components $\pm f_1$, $\pm f_2$, $\pm f_3$ and $\pm f_4$ of the original spectrum $X_a(f)$ that overlap the frequency component $\pm f_0$ ($0 \le f_0 \le f_s/8$) in the spectrum of the decimated sequence spectrum $Y_d(F)|_{F=\frac{4f}{f_s}}$. If we consider that f_0, f_1, f_2, f_3 and f_4 are positive numbers and that the pair $\pm f_i$ is in band i (the positive frequency in part 'a' and the negative one in part 'b') of Fig. 2.1, the following relations can be written:

   $f_1 = f_0$

   $f_2 = f_s/4 - f_0$

   Find f_3 and f_4.

Aliasing is generally not tolerated because, as we have seen, high frequency components of x(t) become new low frequency components that were not present in the original signal. To avoid aliasing, the solution is to introduce an anti-aliasing filter before decimation, that is:

$$y[n] = (x[k] * h_a[k])|_{k=nM}$$

where $h_a[n]$ is the impulse response of the anti-aliasing filter.

3. Give the frequency response of the ideal anti-aliasing filter for the present case (M=4). Draw the Fourier transform $Y_d(F)|_{F=\frac{4f}{fs}}$ of the decimated sequence when an ideal anti-aliasing filter is applied to x[n] before decimation (Hint: consider the Fourier Transform given in Figure 2.1).

2.2 Fourier analysis of interpolated sequences

Interpolation is the reverse of decimation and allows increasing the sampling rate by a factor of N. A convenient way to perform in practice interpolation is as follows:

1) Insert N-1 zeros between every pair of samples of x[n]:

$$v[n] = \begin{cases} x[n/N] & n \text{ multiple of } N \\ 0 & \text{otherwise} \end{cases}$$

2) Use an interpolation filter $h[n]$:

$$y[n] = v[n] * h[n]$$

It is known that the ideal interpolation filter is an ideal low-pass filter having the following infinite-length impulse response:

$$h[n] = sinc(n/N) = \frac{sin(\pi n/N)}{\pi n/N}$$

In order to avoid the complexity of the ideal interpolation filter, it is customary to use simpler finite-length interpolators. For example, if N=2, we can obtain y[n] in the following way:

$$y[n] = \begin{cases} v[n] = x[n/2] & n \text{ even} \\ \dfrac{v[n-1] + v[n+1]}{2} & n \text{ odd} \end{cases}$$

The last equation corresponds to the first-order (linear) interpolator. The name comes from the fact that a first-order polynomial (line) is used to connect the neighbouring non-zero samples of v[n], as illustrated in Fig. 2.3a. The impulse response of this interpolator results to be:

$$h[n] = \{1/2\,, \underline{1}, 1/2\}$$

In most applications, the distortion of linear interpolation gives bad performance and more sophisticated interpolators are required. One of the most popular interpolators in practice is the third-order (cubic) interpolator, which works as follows:

$$y[n] = \begin{cases} v[n] = x[n/2] & n \text{ even} \\ \dfrac{-v[n-3] + 9v[n-1] + 9v[n+1] - v[n+3]}{16} & n \text{ odd} \end{cases}$$

In that case, a third-order (cubic) polynomial is used to connect the neighbouring non-zero samples of v[n], as shown in Fig. 2.3b.

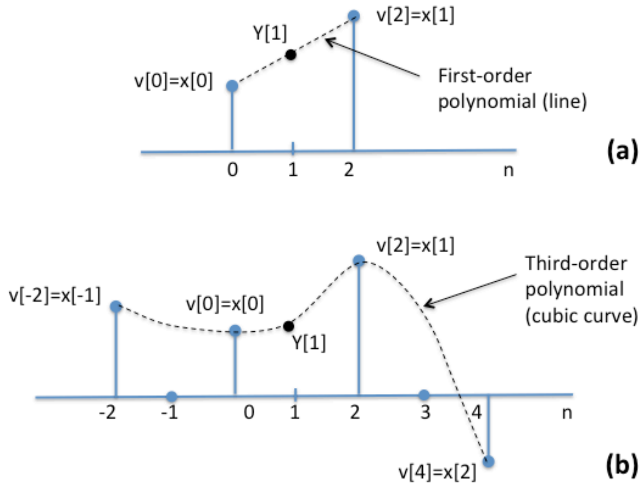


Figure 2.3: (a) Linear interpolation example for N=2; (b) Cubic interpolation example for N=2. In both cases, it is shown how y[1] is computed from the available neighbouring samples.

4. Determine for N=2 the impulse response h[n] of the cubic interpolator that, filtering v[n], gives y[n]. Note: h[n] will be required during the laboratory session.

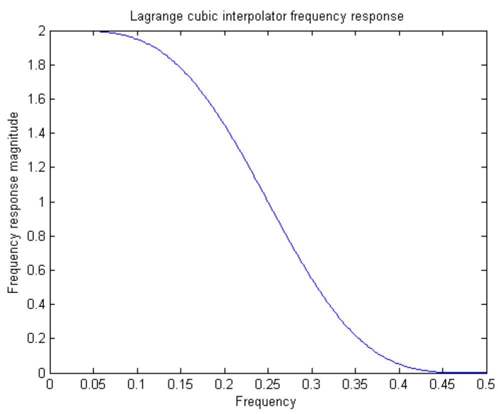


Figure 2.4: Frequency response of the cubic interpolator filter.

5. Observe the frequency response of the cubic interpolator filter shown in Figure 2.4. What features of the frequency response agree with those of an ideal interpolator by N=2? What are the differences?

3 Laboratory

3.1 Decimation of 1D signals

In this section we are going to gain some insights about the decimation and interpolation processing using a music signal. Next, we read the file containing the music signal, define some parameters for the experiments that follow and play the music:

```
FileName='music.wav';                    % Define the filename
[y,Fx] = audioread(FileName);            % Read a stereo wav file
x=(y(:,1)+y(:,2))/2;
sound(x,Fx); pause;                      % Listen to audio

% Experiment parameters
T=10;                                    % Signal length
M=4;                                     % Decimation factor
mt=1;                                    % FT analysis time
window=480;                              % FT analysis window length
NDFT=512;                                % Number of DFT samples
f=[0:1/NDFT:0.5];
NF=[1:NDFT/2+1];
```

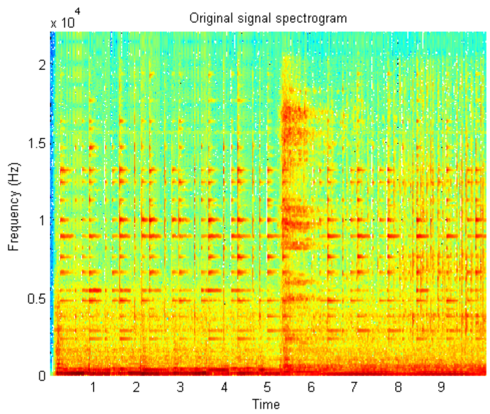
First of all, by using the Short Time Fourier Transform (STFT) we analyze the spectral content of the music and its evolution in time without overlapping successive windows:

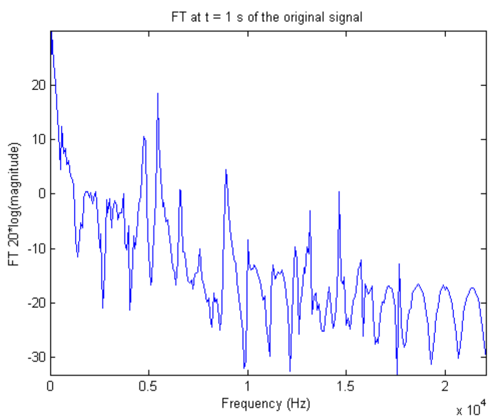
```
[S,g,t,p] = spectrogram(x(1:T*Fx),window,0,NDFT,Fx,'yaxis');
figure(1),surf(t,f*Fx,20*log10(abs(S)),'EdgeColor','none');
title 'Original signal spectrogram'; axis xy; axis tight; colormap(jet); view(0,90);
xlabel('Time'); ylabel('Frequency (Hz)');
```

and compute the Fourier transform (FT) at a specific time (t=1s):

```
sx=x(mt*Fx:mt*Fx+window-1);
Tsx=fft(sx,NDFT);
figure(2),plot(f*Fx,20*log10(abs(Tsx(NF)))),axis tight;
title(['FT at t = ',num2str(mt),' s of the original signal']);
xlabel('Frequency (Hz)'); ylabel('FT 20*log(magnitude)');
```

Figure 3.1 and Figure 3.2 show the magnitude of both transforms.





Figures 3.1 and 3.2: Spectrogram of the music signal (left) and Fourier transform at t=1s (right).

1. Listen to the music again and relate the musical events with the content of the STFT (Fig. 3.1). Describe the event that produces the STFT change at approximately 5.5s?
2. In Figure 3.2, identify the strongest frequency components of the music at t=1s. Give their frequencies. Relate these components with the information given in Figure 3.1.

The music is low-pass filtered and decimated by a ratio of M=4. Let's listen to the result and plot its Fourier transform at the instant t=1s:

```
y=decimate(x(1:T*Fx),M,'FIR');
Fy=Fx/M;
sound(y,Fy); pause;

sy=y(mt*Fy:mt*Fy+window/M-1);
Tsy=fft(sy,NDFT);
figure(3),plot(f*Fy,20*log10(abs(Tsy(NF)))),axis tight;
title(['FT at t = ',num2str(mt),' s of the filtered and decimate signal']);
xlabel('Frequency (Hz)'); ylabel('FT 20*log(magnitude)');
```

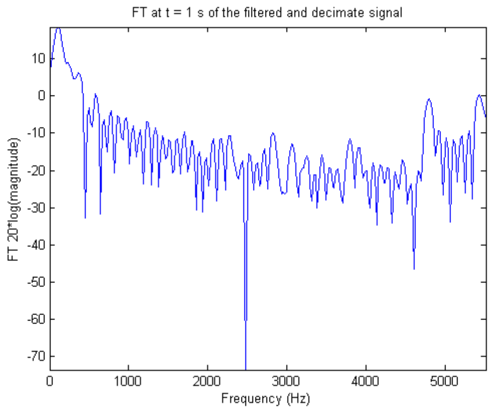


Figure 3.3: Fourier transform at t=1s of the music filtered and decimated.

3. Which is the sampling frequency of the decimated signal?
4. Which should the cut-off frequency of the anti-aliasing filter be?
5. Relate the spectral content of this new signal to the original signal spectral composition at t=1s (compare Figure 3.2 and Figure 3.3). Identify those components of the original signal that are preserved in the decimated version and those that are filtered out.
6. The length of the window used to compute the Fourier transform of the decimated signal is the length used with the original signal divided by M. Could you give a reason for this change? Why and how much the magnitude of the Fourier Transform has decreased?

The original music is decimated again by a ratio of M=4, but now **no low-pass filter** is applied. Let's listen to the new signal and plot the magnitude of its Fourier transform at t=1s.

```
z=downsample(x(1:T*Fx),M);
Fz=Fx/M;
sound(z,Fz); pause;

sz=z(mt*Fz:mt*Fz+window/M-1);
Tsz=fft(sz,NDFT);
figure(4),plot(f*Fz,20*log10(abs(Tsz(NF)))),axis tight;
title(['FT at t = ',num2str(mt),' s of the decimated signal']);
xlabel('Frequency (Hz)'); ylabel('FT 20*log(magnitude)');
```

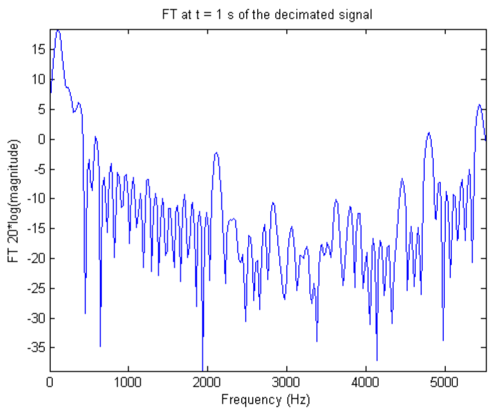


Figure 3.4: Fourier transform at t=1s of the music signal decimated without previous anti-aliasing filtering.

7. Do you perceive a significant difference between both versions of the decimated music? Could you describe this difference?

8. As it can be seen in Figure 3.4, the Fourier transform of the new signal shows two spectral components at t=1s that are not present at the Fourier transform of the pre-filtered decimated signal (Figure 3.3). Give their frequencies.

These new components have been produced by aliasing. Let us analyze this aliasing in more detail:

9. Identify the frequency components of the original signal at t=1s in Figure 3.2 that more likely can produce aliasing when the signal is decimated by M=4 with no antialiasing filter.
10. Give the frequency of the spectral components of the original music that produce the two aliasing components present in Figure 3.4 (Hint: take into account your answer to the question 2 of the previous study).

3.2 Interpolation of 1D signals

Finally, we interpolate by a ratio of N=2 the filtered and decimated music. You are expected to write the impulse response of the Lagrange cubic interpolator that has been obtained in the previous study (question 4) and, afterwards, plot its frequency response (Figure 3.5):

```
h=[ ];                                  % Impulse response of the Lagrange interpolator
N=2;                                    % Interpolation factor
Th=fft(h,NDFT);
figure(5),plot(f,abs(Th(NF))),axis tight,title 'Lagrange cubic interpolator frequency
response';
xlabel('Frequency'); ylabel('Frequency response magnitude');
```

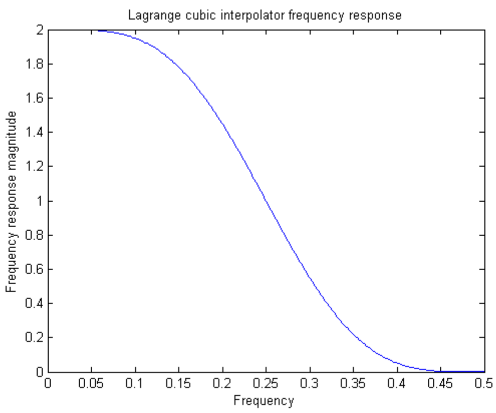


Figure 3.5: Frequency response of the cubic interpolator.

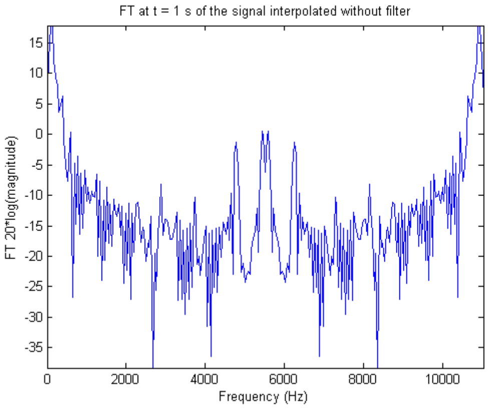
We generate the signal with one null sample between every two samples of the signal to be interpolated:

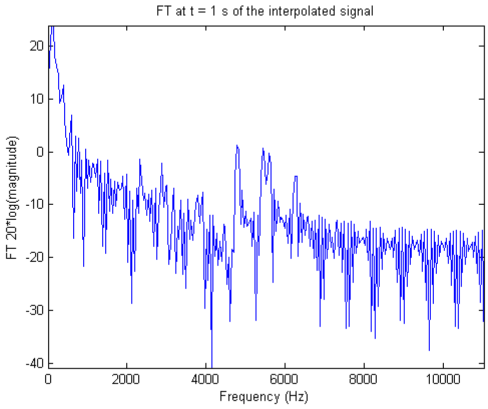
```
Fv=Fy*N;
v=upsample(y(1:T*Fy),N);
sound(v,Fv); pause;

sv=v(mt*Fv:mt*Fv+window/M*N-1);
Tsv=fft(sv,NDFT);
figure(6),plot(f*Fv,20*log10(abs(Tsv(NF)))),axis tight;
title(['FT at t = ',num2str(mt),' s of the signal interpolated without filter']);
xlabel('Frequency (Hz)'); ylabel('FT 20*log(magnitude)');
```

And we filter that new signal using the cubic interpolator filter:

```
Fw=Fy*N;
w=conv(v,h,'valid');
sound(w,Fw); pause;

sw=w(mt*Fw:mt*Fw+window/M*N-1);
Tsw=fft(sw,NDFT);
figure(7),plot(f*Fw,20*log10(abs(Tsw(NF)))),axis tight;
title(['FT at t = ',num2str(mt),' s of the interpolated signal']);
xlabel('Frequency (Hz)'); ylabel('FT 20*log(magnitude)');
```

The Fourier transform of both signals at t=1s are shown in Figure 3.6 and Figure 3.7, respectively.





Figures 3.6 and 3.7: Fourier transform at t=1s of the music signal with zero-valued samples in between the original samples before (left) and after (right) being filtered.

11. Compare the Fourier transforms in Figure 3.3, Figure 3.6 and Figure 3.7. Point out the main differences between the Fourier transforms in Figure 3.3 and Figure 3.6; repeat this analysis between Figure 3.6 and Figure 3.7. In both cases, explain the causes of the differences you have found out.