FSAB 1106 - meeting 10
Discrete Fourier Transform - Filtering

G. Janssens, S. Brousmiche, J. Giard, J. Olszewska

Matlab exercise

Consider a signal $x(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t)$, $f_1 = 150$ Hz, $f_2 = 200$ Hz.

1. Sample $x(t)$ in order to avoid aliasing. Truncate the signal $x[n]$ with a window enabling its spectrum to be as close as possible than the one from $x(t)$ by choosing correct sampling frequency and number of samples. Explain your results.
2. Low-pass filter design :
   a. Calculate the impulse response $h[n]$ of an ideal lowpass filter that removes the f_2 component. Use the inverse discrete Fourier transform of the filter described in figure 1.
   b. Then compute this impulse response with Matlab for a cut-off frequency $f_c = 175$ Hz. Discretize the ideal filter with a large number of points and apply the `ifft` method.
   c. Truncate $h[n]$. Compare the ideal filter to the spectrum of the acquired filter for different window lengths. Explain the effect of the truncature on the spectrum. What can you conclude about the filter quality ?
   d. Is this filter causal ? What does it imply for its practical use ?
   e. Apply this low-pass filter to the signal $x[n]$ by using the function `conv`. Compare the resulting signal in temporal and frequential spaces (for different window lengths). Verify that you obtain a signal close to $x[n] = \cos(2\pi F_1 n)$.
3. High-pass filter design :
   – Repeat these steps for a high-pass filter with the same cut-af frequency.
   – Compare the impulse response with the low-pass impulse response and give the relation between them.
4. Filtering of an audio signal[1] :
   – Open the `wav` file with function `wavread`. What is its sampling frequency ? Plot its pseudo-analogic spectrum (frequenties in Hz).
   – Adapt the filters to the new sampling frequency.
   – Apply the filters to the audio signal in order to separate bass from saxophone and record them as right and left channel of a new stereo stream (`wav` file). Listen to the resulting music (using `wavplay`). Plot spectrum of both channels.
   – Create an equalizer by applying different gain to the channels. Comment your results.
5. [**Advanced question (optional)**] Butterworth filters design.
   – Give the frequency response of the lowpass Butterworth filter of order 1 and 2.
   – Compare these frequency responses.

[1] Extract from *Indiana* (Hanley, MacDonald), by Don Byas (s) and Slam Stewart (b), June 1945

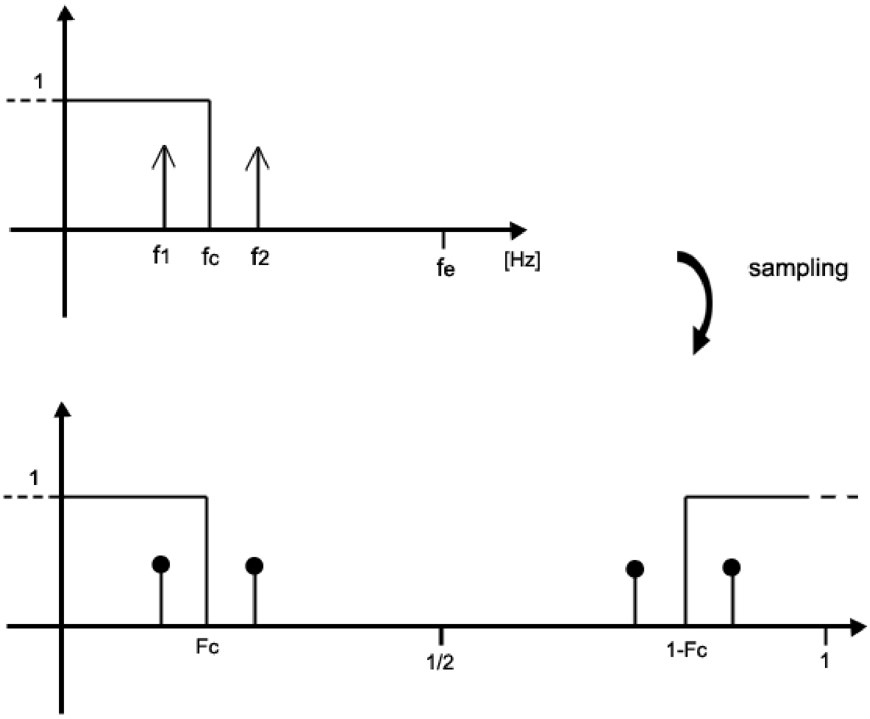

FIG. 1 – Low-pass filter (in continuous and discrete spaces).

- Compute $H(e^{j\Omega})$ using the `butter` function. This function gives the numerator and denominator of $H(e^{j\Omega})$. (Comment : The frequencies used in the MATLAB function are frequencies normalized by the half of the sampling frequency)
  `[num,den]=butter(order,cutoff_freq);`
  $$H(e^{j\Omega}) = \frac{num(n).e^{-j\Omega(n-1)} + ... + num(2).e^{-j\Omega} + num(1)}{den(m).e^{-j\Omega(m-1)} + ... + den(2).e^{-j\Omega} + den(1)} \quad (1)$$
- Give the recurrence equation enabling this filtering, that is express $Y[n]$ as a function of $X[n-i]$ and $Y[n-k]$ $\forall i, k$ (Reminder : the multiplication by $e^{j\Omega}$ in the frequency domain is the same as a shift in the time domain). Is this filter causal ? Why ?
- Apply these filters to $x[n]$ by using the `filter` function and compare the results you get.