Signals and Systems

Lab experience 3

Report deadline : 7th January 2025

Politecnico di Torino

Exercise 1: DTF and FFT

- Load a music file of at least 20 seconds (you are free to choose the file) and determine the sampling frequency F_s
  - To load the file you can use
    - The function `audioread` in Matlab
    - The function `scipy.io.wavfile.read()` in Python (or other equivalent functions depending on the file format)
  - Read carefully the manual to understand how these functions work
- Estimate the spectrum of energy over temporal windows, using two approaches:
  - Implementing yourself the DFT (Discrete Fourier Transform)
  - Using the built-in FFT (Fast Fourier Transform) function
- Compare the obtained results

Exercise 1: DTF and FFT

- The music file should be divided into K temporal windows of length M seconds
  - Knowing the sampling frequency F_s you can determine how many samples N are contained in each window
  - Try different values for M
- For each window, plot the resulting spectrum of energy
  - It is convenient to plot the spectrum in dB
  - Pay particular attention to the frequency axis that should be quoted in kHz
  - You may need to use the `fftshift()` function: read what it is and what it does

Exercise 1: DTF and FFT

- Required outputs:
  - Plot of the spectra of energy for different windows and considering different M values
    - There is no need to put all the figures in the report, but only a subset of them which are more significant
  - Comparison of the computational time required by your implementation and the built-in FFT function
    - Use `tic()` and `toc()` functions

Discrete Fourier Transform (DFT)

- Given a signal $x(n)$ of N samples, the **discrete Fourier transform** is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi n\frac{k}{N}}, \quad \forall k = 0,1,2,\ldots,N-1$$

- $X(k)$ an be interpreted as the DTFT $X(e^{j2\pi f})$ evaluated at the N equi-spaced frequencies:

$$f_k = \frac{k}{N}, \quad \forall k = 0,1,2,\ldots,N-1 \qquad\qquad X(k) = \mathrm{DFT}[x(n)] = X\left(e^{j2\pi f_k}\right)$$

- The **spectrum of energy** is defined as: $S_x = |X(k)|^2$

Exercise 2: DFT and circular convolution

- Consider a portion of the music file of exercise 1 (e.g. 10 seconds) and filter it using a given FIR filter
    - The impulse response $h[n]$ of the filter is contained in the file «FIR_impluse_response.mat»
    - It is a raised-cosine with bandwidth equal to the 10% of the total bandwidth

- Perform the filtering in two ways:
    - In time domain using the convolution
    - Using the circular convolution, i.e. applying the DFT

Exercise 2: DFT and circular convolution

- Required outputs:
    - Compare the filtered signals in time domain
    - Plot of the spectra of energy before and after the filtering for both implementations

- Suggestion: to play the music and listen to the impact of filtering, in Matlab you can use the function `soundsc`

DFT and circular convolution

- Given a non-periodic signal $x[n]$ and a filter describing an LTI system with impulse response $h[n]$, the filtered signal at the output of the system can be computed as:

$$y[n] = h[n] * x[n]$$
$$Y(k) = H(k)X(k)$$

- Where $x[n]$ and $h[n]$ have extensions $[0, N_x - 1]$ and $[0, N_h - 1]$, respectively, with $N_x \neq N_h$, and $x[n]$ has extension $[0, N_y - 1]$ with $N_y = N_x$+ N_h-1
- $X(k) = DFT\{x[n]\}$ and $Y(k) = DFT\{y[n]\}$

DFT and circular convolution

- To perform filtering in frequency domain using the DFT

$$Y(k) = H(k)X(k)$$

- $X(k)$ and $H(k)$ should have the same size N
- For this reason **zero-padding** can be performed, consisting in adding a sufficient number of zeros such that

$$N \geq N_x + N_h - 1$$

- If you use the FFT, N can be chosen equal to 2^m, where $m = log_2(N_x + N_h - 1)$