Practical works – n^{o}4

Discrete Fourier Transform

The reasons to convert signals into frequency domain is to get the idea of different frequency components it is composed of. This might help to detect the required signals and classify them, for examples, speak or speech recognition from a sound wave, noise removal from the imag, or finding recurring patterns within images.

The Discrete Fourier Transform (DFT) converts discrete data from a time domain into frequency domain. The DFT analyzes the finite segment in one period of an infinitely extended periodic signal, which is defined as:

$$X[n] = \sum_{k=0}^{N-1} x[k]e^{-j(2\pi/N)kn}$$

Note: The Fast Fourier Transform (FFT) is simply an algorithm to compute the DFT in a faster way!

- **Exercice 1** – *DFT*

1.1 The DFT of a 5 Hz sin wave sampled with the sampling of $f_s = 50$ Hz over 1000 (N = 1000) samples is computed as follows:

```
f = 5; fs = 50;
t = 0:  1/fs :  1-1/fs;
x_n = sin(2*π*f*t);
N = length(x_n);
fr = (-N/2 :  N/2-1)* fs/N;
x_f = fftshift(fft(x_n));
subplot(221); plot(t, x_n); title('Signal'); xlabel('Time(sec)'); ylabel('Amplitude');
subplot(222); plot(fr, abs(x_n)); title('Magnitude'); xlabel('Frequency'); ylabel('|X(f)|');
subplot(223); plot(t, real(x_n)); title('Real'); xlabel('Frequency'); ylabel('Re(X(f))');
subplot(224); plot(t, imag(x_n)); title('Imaginary'); xlabel('Frequency'); ylabel('Im(X(f))');
```

1.2 Compute the DFT of a cosine wave, how that differs from DFT of a sine wave ?

1.3 Use square wave using the same frequency and sampling frequency as the sin and cosine wave. ("signal.square" and "square" in python and matlab, respectively)

1.4 Use Gaussian noise with 10000, samples.

- **Exercice 2** – *Sampling*

2.1 Generate and display the following signals of 1 sec duration.

$$x[n] = 3cos(2\pi\frac{f_1}{f_s}n) + 4sin(2\pi\frac{f_2}{f_s}n)$$

for $f_1 = 5$ Hz and $f_2 = 20$ Hz, sampled with the sampling frequencies of $f_s = [10, 20, 25, 40, 50, 100, 150]$

2.2 Plot $x[n]$ for different sampling frequency in time domain.

2.3 Discuss the aliasing effects in the time domain.

2.4 Compute the FFT of the above signals and display their centered frequency components. Discuss your observations.

- **Exercice 3** – *1D DFT for image classification*

3.1 1D DFT can be applied to analyze the frequency components along the one dimensional profile of an image. Here we want to use these information to classify the images into two different classes ("barcode" and "non-barcode" classes)

To do so we first

Load the images from 1D-DFT folder

3.2 you have to normalize and resize the images to the smalles size

3.3 Take 1D profile as shown in the example and compute its DFT.

3.4 Seprate the images to two different groups of "barcode" and "non-barcode" based on their frequency spectrum of their profile

Some information regarding the images:

There are 54 images while Images [1, 2, 6, 44:54] are barcode images and the others are similar patterns.

3.5 Discuss your results.