Cairo University- Faculty of Engineering
Electronics and Communications Engineering Department
Analysis of Continuous-Time Signals (ELC 2030) – Fall 2024

Signals Project

It is required to use Matlab or any other programming language to perform the following task:

Heart rate monitoring using ECG signals

In this task, we will explore different ideas for heart rate monitoring from an electrocardiogram (ECG) signal. We will obtain the heart rate from the time domain and from the frequency domain. We will first consider a segment of the signal with constant heart rate, then we will monitor its variation, and detect abnormally high or low rates.

Background:

The spectrum of a discrete-time signal (the DTFT) is generally a continuous function of frequency. Therefore, it is not suitable for numerical computations. Instead, we will use the **Discrete Fourier Transform (DFT)**, which is a sampled version of the DTFT [Project lecture].

When a continuous-time signal is sampled at a rate f_s Hz, its spectrum is repeated at multiples of f_s Hz. The N-point DFT of a DT signal $x[n]$ gives N samples of one period of its spectrum. Since we have N samples in a range of frequencies from 0 to f_s Hz, the spacing between the samples is f_s/N. The N-point DFT X[k] is given in the range of $k = 0\ to\ N-1$ and corresponds to frequencies $f = 0\ to\ (N-1)f_s/N$. Note that N must be greater than or equal to L, the length of x[n].

We say that $X[k]$ is the DFT of $x[n]$ and $x[n]$ is the inverse DFT (IDFT) of $X[k]$. These two operations are implemented in Matlab using ‘fft’ and ‘ifft’ functions, respectively. Fast Fourier transform (FFT) is an efficient algorithm for computing the DFT.

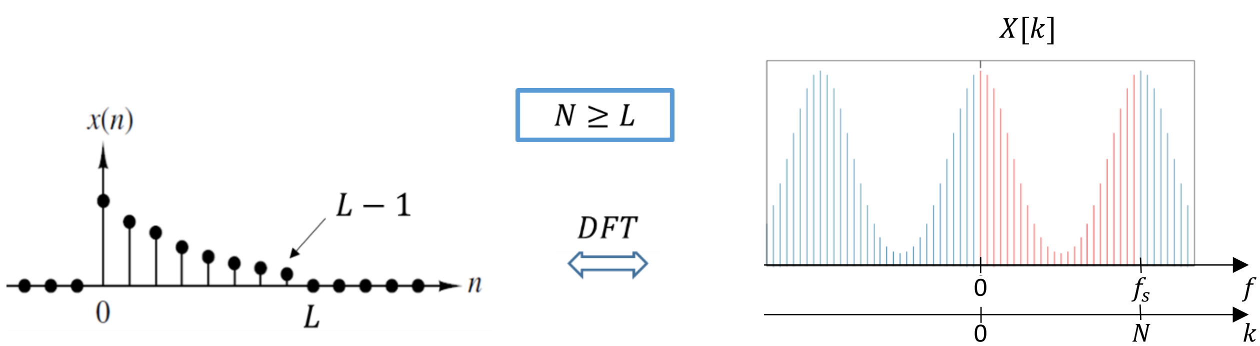

Fig. 1: DFT

Task:

a. In Matlab load the given file ‘ecg_data.mat’. It contains a vector ‘ecg_signal’ of an ECG signal, sampled at a rate Fs=360 Hz.
   Plot the first 10 seconds of the signal in time domain, against time in seconds.
   Hints:
   - you have to make use of Fs.
   - The heart rate is constant in the above segment.

b. From the above 10-second segment, suggest and implement programmatically a method to obtain the average heart rate in beats per minute (BPM).
   Hint: pay careful attention to the conversion of units in this and the following parts.

c. Obtain and plot the magnitude spectrum of the above 10-second segment against frequency in Hz.

d. Design and apply a filter to remove the out-of-band noise. Apply it to the signal used in the following parts. Plot the signal and its magnitude spectrum after applying the filter. Describe and justify your selection of filter.

e. From the spectrum, suggest and implement programmatically a method to obtain the heart rate in BPM, using frequency domain. Compare your result to the result obtained from time domain.

   Now we will consider the **whole ECG signal** given in the file. You will notice that the heart rate varies with time. It is required to plot a graph showing how the heart rate varies with time throughout the whole signal. It is required to draw plots in which the x-axis represents time and the y-axis represents the heart rate in BPM.

f. Using the whole time-domain signal, suggest and implement programmatically a method to obtain and plot the heart rate in BPM against time, displaying any variations.

g. Now we will repeat using frequency domain. If we obtain the Fourier transform of the whole signal, it will show the overall frequency content of the signal. However, in this case the spectral content of the signal varies with time, and we are interested in monitoring such variation to track the heart rate, therefore we will use short-time Fourier transform (STFT). In STFT, we divide the signal into short segments (overlapping or non-overlapping), and apply Fourier transform to each. This is done by multiplying the file by a moving window function, and then obtaining the FT of the product signal for each shift. Different window functions can be used (rectangular, triangular, Hamming, Hann, …).
   - Suggest and implement programmatically a method to obtain and plot the heart rate in BPM against time, using STFT. Experiment with different window functions. Pick your selection of window function and justify it.
   Compare the results that you obtained from time and frequency domains, and justify any differences.
   Hint: Window size should be chosen to capture more than one period; you may select it to capture 3 complete periods. You may take 50% overlap.

h. If the normal human heart rate is from 60 to 100 BPM, programmatically highlight on the graph the intervals with higher or lower rate than normal. Also display the intervals of time for each case, in seconds.

   Notes:
   1. The methods used in time domain and frequency domain to obtain the heart rate should be **clearly explained**, together with the **motivation** of each.
   2. You may use the Matlab filter designer tool “filterDesigner” to design the filter and export it as an object. Then use the “filter” function to filter the audio using the designed filter. If used, add a screenshot of the design page to the report. FilterDesigner can also be used to generate the response plots of the filter.

Deliverables:
1. An **uncompressed pdf** project report containing:
   a. Explanation of your work in detail
   b. All the required results and answers to questions.
   c. All the required figures. Label your figures properly.
   d. All the codes, with proper comments, included at the end.
2. All the code files in a compressed zip file.

Instructions:
- You can work in teams of up to 2 members per team.
- Any copied answers, results, or codes will result in zero grade.
- The code given in the report should be given as text, not as screenshots.
- It is not allowed to use any form of Artificial Intelligence in this project.
- You should cite any references or webpages that you use.

Due date: December 20, 2024, at 11:59 PM.