CIE-442: Digital Signal Processing

Project 1

Submission Deadline: Week 10 [3rd December 2022]

Objectives:

- Understanding Spectrogram and implementing it from scratch using STFT.
- Investigating the effect of window size, type, overlapping on spectrogram representation and the relation between time & frequency resolutions.
- Understanding the process of filter design.
- Handling, manipulating, and converting raw digital signals into useful representation.

Introduction:

The aim of this project is to build up an analytical system based on the spectrogram using Short-Time Fourier Transform (STFT) to assist neurologists in monitoring and diagnosing epileptic electroencephalography (EEG) signals. Epilepsy is a disorder in the electrical activity of the brain that causes repetitive and unprovoked seizures. In general, the epileptic EEG signal is characterized by two main states: interictal and ictal. The interictal state is considered the period between seizures while the ictal state is the period of a seizure which is characterized by its large spikes. Other states (i.e., preictal and postictal) are out of our scope in this project. CHB-MIT [1] is the source of the EEG records that will be used for this project and found in this link:
https://drive.google.com/drive/folders/129n48qq7R7Y-QE68dciBESN9YCW8PCf1?usp=sharing

Project tasks:

I. Pre-processing

1. At the beginning, you should import and read the attached raw EEG records found in the provided link above. For the same patient, the record “chb12_32_data.mat” contains no seizures (only interictal state) while “chb12_29_data.mat” contains 6 seizures (6 different ictal states). You should extract only the **ictal states** from “chb12_29_data.mat” using “chb12-summary.txt” to own a seizures-only record & a seizure-free record. (**1 point**)

2. Then, designing an IIR Notch Filter is needed using MATLAB Filter Designer UI App for **interference removal from both records**. The provided EEG records were acquired using non-invasive electrodes placed on different locations on the scalp to pick up the brain electrical signals. One of the common artifacts (i.e., picked up signals from other electrical sources not by brain) in the raw EEG signals are the notably power line harmonics which is a **60 Hz** power signal in CHB-MIT [1] (can be 50 Hz depending on the country). Thus, the objective of this step is to null out this undesired interference using your own **IIR Notch filter** to clean up EEG data. (**2 points**)

3. The outcome of the filter design process should be the designed filter coefficients, the filter frequency response, and a plot for the signal after filtration. (**1 point)**

4. The given data are Multi-Channel. Every channel of the data corresponds to a unique electrode position which describes more the signal originated from this specific part of the brain and thus, channel selection is an important feature of the system. You are required to provide the option for the user (using text box or check box) to select certain channels, then averaging all the selected channels collection into a single one. (**1 point**)

II. Spectrogram

Knowing that all the EEG signal records of CHB-MIT [1] were sampled at **256 HZ**, you should convert the EEG data into spectrogram representation segmented into “n” seconds window frames using STFT. You can use MATLAB built-in function ***stft()*** for the STFT; however, you are required to construct the spectrogram from scratch. The **pre-processed data** should be segmented and the spectrogram plots can be generated by:

5. Choosing different values for frequency (f) and time (t) resolutions. Time resolution represents the number of samples of the time domain signal passed to the STFT. Frequency resolution is the number of FT samples. Explain the trade-off between both and the effect of changing each throughout the chosen values. (**2 points**)

6. Putting an option for overlapping. The time domain window frames can have overlapped samples from segment to segment. The ratio between the number of overlapped samples and the total window samples is called the overlapping ratio. Show the effect of applying different overlapping ratios. (**2 points**)

7. Choosing the window size and investigating the effect of changing it. (**2 points**)
8. Applying windowing using different window types and investigating the effect of changing their parameters. (**2 points**)
   You are required to investigate the effect of **four** different windows:

   a. Rectangular Window (no-window)

   b. Triangular Window

   c. Hamming Window: $w[i] = 0.54 - 0.46\cos(2\pi i/M)$

   d. Blackman Window:
   $w[i] = 0.42 - 0.5\cos(2\pi i/M) + 0.08\cos(4\pi i/M)$

➔ Where, filters run from $i = 0$ to $i = M$ for the total of $M + 1$ points. (see chapter 16 in The Scientist and Engineer's Guide to Digital Signal Processing)

III. Plotting

9. Plot the normal record versus the abnormal record including all the selected channels and after averaging in time domain. (**1 point**)
10. Display the corresponding spectrogram plots of all the content of the previous point. (**1 point**)

IV. Bonus

You have an option to download more records from CHB-MIT [1], then convert them into .MAT format using the attached "EdfToMat.m" (both "ReadEDF.mat" and "SaveEDF.mat" should be included in the workspace) to generate more spectrograms using your implemented function and do any further work based on deep learning (DL) or traditional machine learning (ML) methods. For example, the generated spectrogram plots can be saved as images to be used as inputs for supervised learning.

The bonus will be extra (**5 points**) given as follows:

- Manipulating the data and feeding it to a ML or DL model to be trained without coding errors. (**2 points out of 5**)
- Manipulating the data, training model(s), tuning for different hyperparameters, tuning for different spectrogram parameters (*f* & *t* resolution, window size, window type) and documenting the different runs. (**4 points out of 5**)

- Manipulating the data, training model(s), tuning for different hyperparameters, tuning for different spectrogram parameters (f & t resolution, window size, window type) and documenting the different runs and **reporting high metrics** (e.g., accuracy). (**5 points out of 5**)

Project submission requirements and criteria:

- Team members maximum **3**.
- Include in your submission, all the implemented or used code scripts/modules for all the required project tasks to be ready for running.
- Documentation is mandatory along with the codes using MATLAB Live Editor. Any project without documentation **will not be accepted** for grading.
- Documentation should explain all your implemented tasks and their relation with the used code modules. Also, all the relevant outputs should be displayed in your Live Editor according to the assigned tasks.
- All the variables within the codes should be readable and with meaningful names. For each module, all the variables including the input(s) and output(s) should be explained carefully in the documentation .
- Any plagiarized codes or documentations, either fully or partially, will receive **zero** as final project grade.

Project Grading Rubric:

(Total grade of this project is worth 20% of the course grade)

→ The **20** points are distributed as follows:

5 points: for the individual discussion.

15 points: distributed among the required tasks as detailed above. Note that the grade per task includes the documentation of the task. The task grading will be on a level from 0-3.

- **Level 0**: Task not done
- **Level 1**: Inadequate
- **Level 2**: Needs improvement
- **Level 3**: Meets expectation

[1] https://physionet.org/content/chbmit/1.0.0/