A REPORT ON THE DEVELOPMENT OF BCI EXPERTISE USING BRAIN COMPUTER CONTROLLED HAND ORHTOSIS DEVELOPMENT AS A CASE STUDY

STATUS: Ongoing

TERMS: CSP (Common Spatial Patterns), FBCSP (Filter Bank Common Spatial Patterns), PCA (Principal Component Analysis), LSL (Lab Streaming Layer)

SUMMARY:

So as to develop an understanding of brain computer interface as applied to the purpose of this project, literatures on brain computer interface applied to motor-imagery were reviewed. Also chapters of interest were reviewed in the text, Brain Computer Interface by Rajesh P.N. Rao

The openBCI hardware was used for practical implementation alongside its software. Other software used were Neuropype, Pipeline-designer, Lab-Recorder and the Python software. The documentation of Neuropype was consulted during use. Lab-Recorder and Pipeline-designer are softwares that come along with Neuropype package installation. Academic license for the Neuropype package was requested for and granted from the Neuropype website.

Signals were acquired, at first using openBCI hardware and software alone, of which the data was stored in BDF+ format. On second trial of data acquisition, data was acquired using openBCI hardware, openBCI software, Lab-Streaming-Layer and Lab Recorder of which Lab-Recorder was used to attach four event markers on the acquired data namely, imagine-grasp, grasp, imagine-release, release of which he data format was XDF. All subjects from which data was acquired filled a consent form online before data-acquisition.

The already acquired signals in XDF format were preprocessed using the neuropype software piepeline-designer. Lab-Recorder which was used in attaching event-markers to acquired data is a sub-component of the neuropype software just like the pipeline-designer.

Feature extraction using CSP, FBCSP were also performed on he acquired data using the neuropype pipeline-designer. PCA was performed on the acquired data using neuropype, LSL and a python script. Data acquired from CSP were applied to a machine learning classifier.

TABLE OF CONTENTS

|  |  |
| --- | --- |
| LITERATURE REVIEW |  |
| SIGNAL ACQUISITION |  |
| SIGNAL-PREPROCESSING & FEATURE-EXTRACTION |  |
| MACHINE-LEARNING |  |
| PLATES |  |

LITERATURE REVIEW

The following literature were reviewed towards developing an understanding of BCI and motor-imagery.

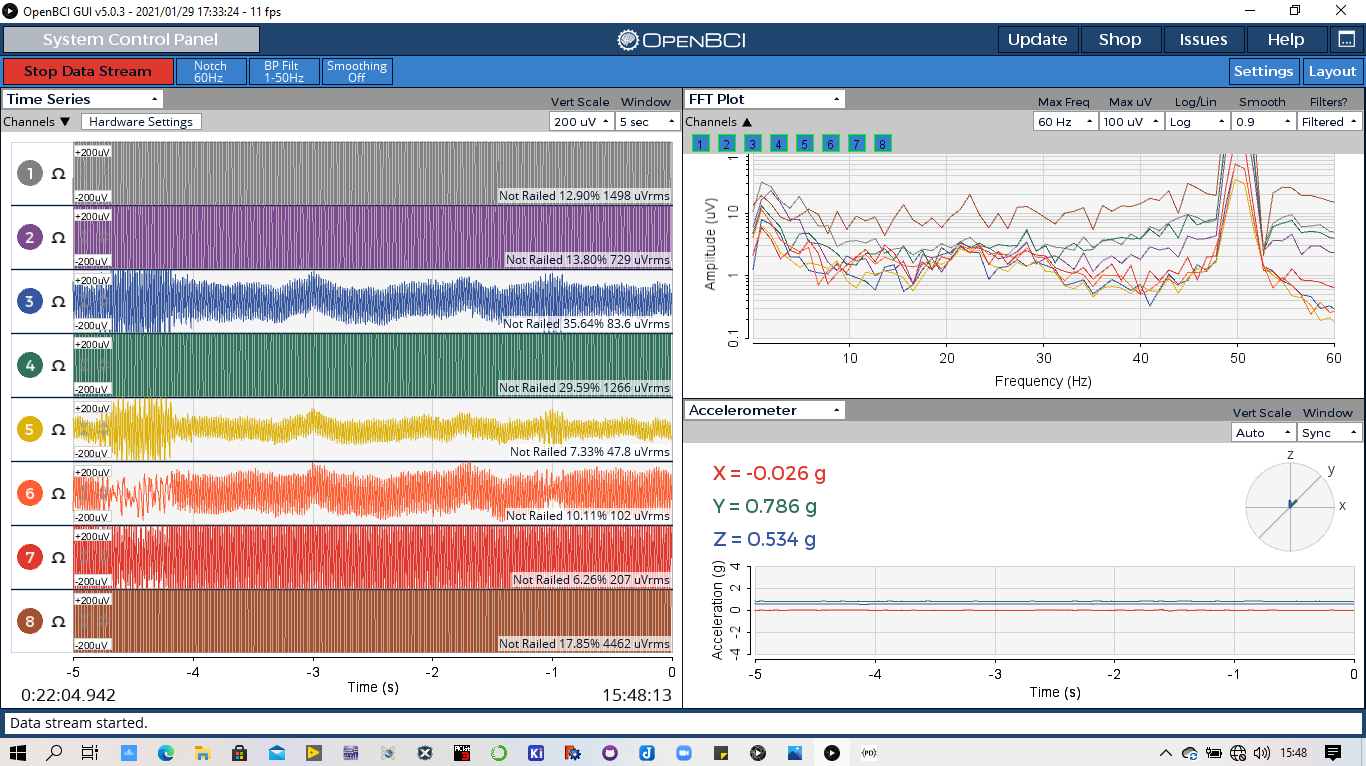
1. [Classification of Motor Functions from EEG by integrating CSP and CWT](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6891287/)
2. [EEG-Based Brain-Computer Interfaces Using Motor-Imagery: Techniques and Challenges](https://pubmed.ncbi.nlm.nih.gov/30909489/)
3. [Mu rhythm (de)synchronization and EEG single-trial classification of different motor imagery tasks](https://www.sciencedirect.com/science/article/abs/pii/S1053811905025140)
4. [comparative study of wavelet and CSP in EEG based motor-imagery](https://ieeexplore.ieee.org/document/8448212)
5. [Classification of Motor Functions from EEG by integrating CSP and CWT](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6891287/)
6. [MI classification using Mu/beta rhythms based on uncorrelating transforms CCSP](https://www.hindawi.com/journals/cin/2016/1489692/)

SIGNAL ACQUISITION

All subjects from which EEG data were acquired filled a form online giving their consent to be test-subjects. Only data acquired from subjects with low level of hair and no history of mental-ailment were used in the course of this project.

On first trial of signal acquisition, the ultracortex-markIV headset and cyton-board from openBCI were used alongside the openBCI software. The data acquired was stored in BDF+ format. Eight electrode channels were used namely, Fp1, Fp2, C3, C4, P7, P8, O1, O2 at a sampling frequency of 250Hz. Each electrode position was not railed on the openBCI software indicating the resistance between the electrodes and the head surface of the subject was within appropriate range, but the voltage levels were unexpectedly high. The data format for this set of data was BDF+.

.



*Figure1: OpenBCI interface while acquiring signal on first trial*

The second set of data-acquisition was more defined and another software, Lab-Recorder was involved. The Lab-Recorder via LSL, received EEG data from the openBCI software and attached event-markers to the data while giving event instructions graphically to the test-subjects. The python script controlling the Lab-Recorder software was edited to suite our event-sequence and timing. Our event-sequence was: imagine-grasp, grasp, imagine-release, release. A bottle was placed in front of the subjects to be held and then released for the grasp and release events respectively. For the imagine-grasp and imagine-release events, subjects were asked and expected to imagine themselves holding and releasing the bottle. This is the process of motor-imagery. The data format for this set of data was XDF.



*Figure 2: t5est5[-subject5 during third set of data-acquisition, lab-recorder shown giving instructions on PC screen*

[Link to Edited Lab-Recorder code, for event-marker attachment to signal](https://drive.google.com/drive/u/1/folders/1Vi5RJ8HZMTyLa58kJLKK3PbBvlo2taWv)

Subsequent data-acquisition followed the same process of the second set of data-acquisition. All data were acquired at a sampling rate of 250Hz and no form of filter was applied alongside data-acquisition. All forms of filtering happened in the signal-processing stage to be discussed in upcoming sections of this report.

SIGNAL-PREPROCESSING AND FEATURE-EXTRACTION

3.1 SIGNAL-PREPROCESSING

The acquired EEG signals were pre-processed by filtering to reduce power-line interference noise and also to reduce biological artifacts such as ocular, electromyography and electrocardiograph artifacts found within 4Hz, above 30Hz and not greater than 1.2Hz respectively. As the µ and β frequency range which are required for motor-imagery lie between 8Hz-30Hz, there will be little loss of information if any. A band-pass FIR filter of low and high cut-off frequencies, 8Hz and 30Hz respectively and with stop band attenuation frequencies of 5Hz and 32Hz was used to achieve this. Power-line interference noise is about 50Hz, so the band-pass filter works well for reducing power-line interference noise, whilst reducing artifacts. The signal-preprocessing was done using Neuropype pipeline-designer

3.2 FEATURE-EXTRACTION

Up till date, four methods have been applied for feature extraction namely:

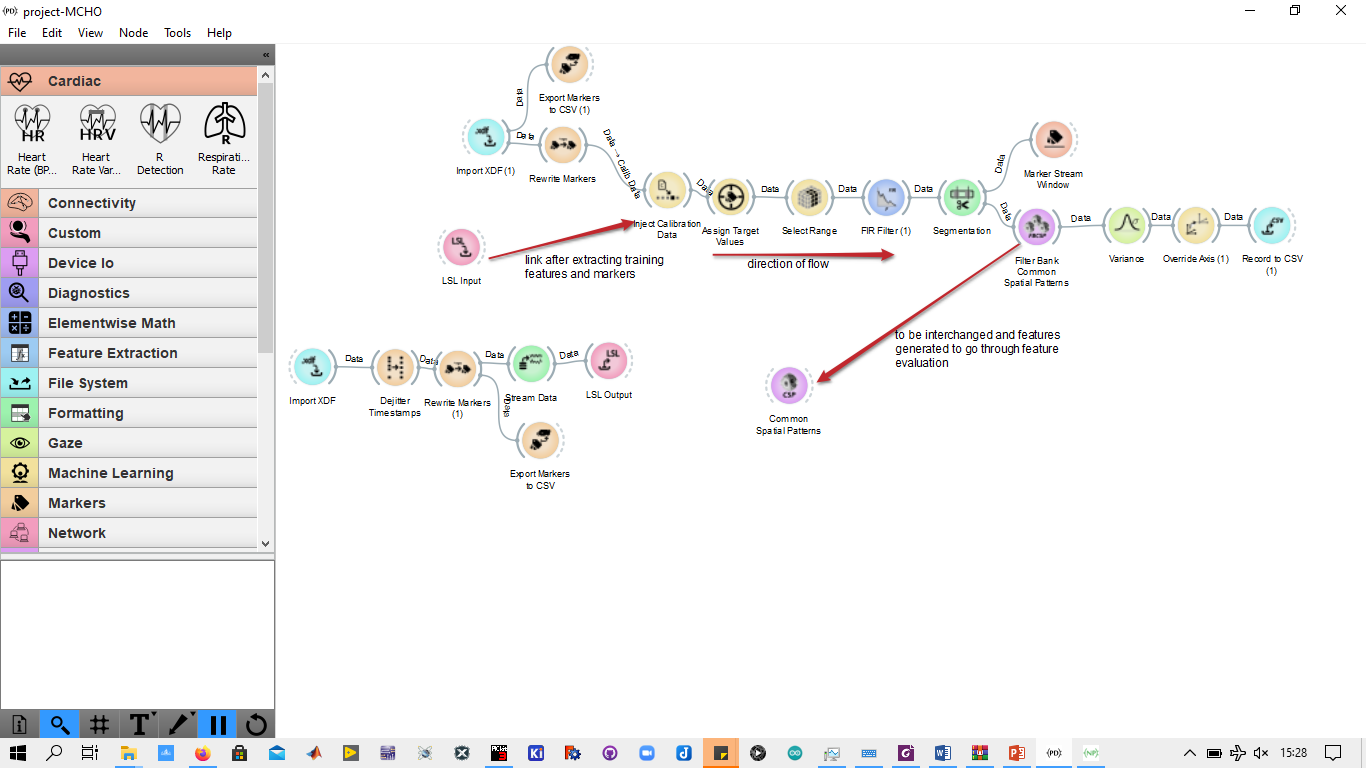
1. Common Spatial Patterns (CSP)
2. Filter Bank Common Spatial Patterns (FBCSP)
3. Principal Component Analysis (PCA)
4. Wavelet Transform

3.2.1 CSP & FBCSP:

Using the same pipeline on Neuropype Pipeline-designer, CSP and FBCSP were used interchangeably both for four pattern pairs.

CSP having four pattern pairs produced an output with a dimension of 8X1(space, time), where two space axes correspond to one pattern pair.

FBCSP on the other hand produced data output with dimension of 16X1(space, time). This is because the FBCSP method splits the EEG signals into two frequency bands namely, the µ-band and β-band with frequency bands 8Hz to 12Hz and 13Hz to 30Hz respectively. Using four pattern pairs just as in CSP, rather one pattern having two axes, one pattern had four axes, one pair each for the µ-band and β-band.



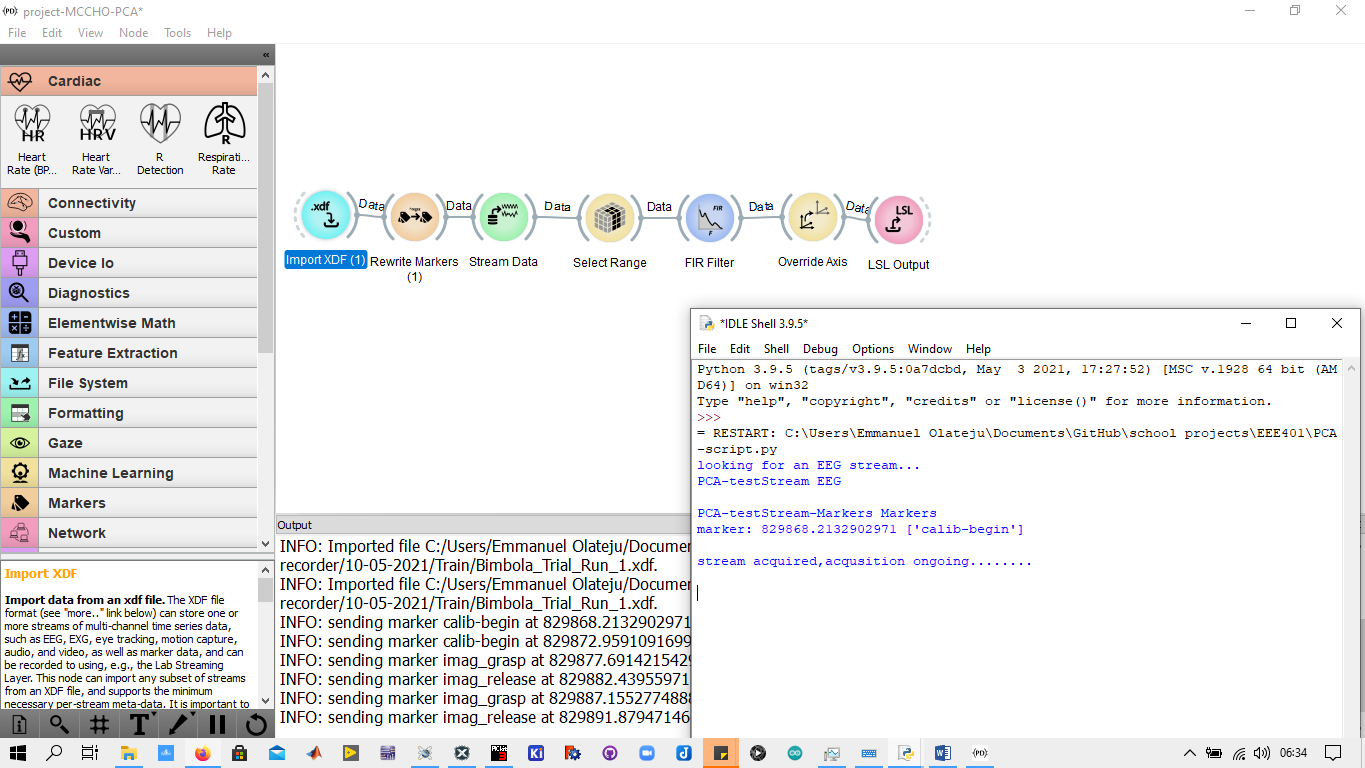
*Figure 3: Neuropy6pe pipeline used for CSP and FBCSP feat5ures-extraction method from recorded data*

3.2.2 PCA:

This dimensionality reduction technique was used in revealing directions of maximum statistical variability in the raw EEG data. Each output column represents a vector and the output of PCA is a linear-combination of vector directions of maximum variance with the first vector giving the direction of highest maximum variance.

Neuropy6pe pipeline-designer alongside LSL and an external python script was used for this method. EEG data acquired from this method is y6et to be applied to other feature extraction nodes (as PCA simply makes variant features more visible by unveiling directions of maximum variance), neit5her has it5 been applied to any machine-learning classifier.

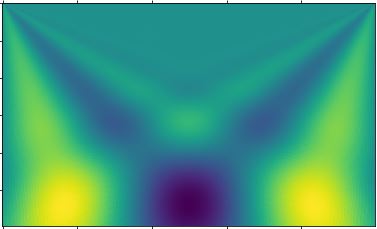
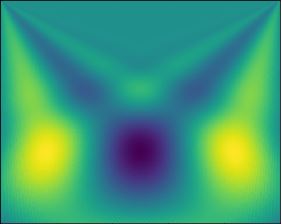
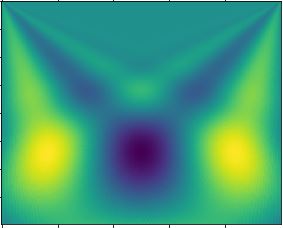
A python-script was used as the PCA node on Neuropype’s pipeline-designer gave a dimensionless data as its output making it difficult to interpret, hence the need for writing a python-script to receive data via and LSL and then perform PCA.



*Figure 4: Neuropype pipeline used for streaming data t5o python script through LSL. Python IDLE shell shown running, receiving LSL st5ream and apply6ing PCA*

3.2.3WAVELET TRANSFORM:

Wavelet transform was employed in developing scalograms to be sent as features to machine-learning classifiers. The mother wavelet adopted up till date is the morlet wavelet. Wavelet transform was employed because of its balance between temporal and frequency resolution. The choosing of the morlet wavelet was arbitrary and jus for the purpose of testing the wavelet transform method. The best mother wavelet is still to be chosen based on comparison of vanishing-moment and other parameters that determine the quality of a mother-wavelet.



*Figure 5: from top left: (a) scalogram of release event (b) scalogram of anot5her inst5ance of release event (c)scalogram of grasp event*