Abstract

In the era of brain computer interfaces, the importance of P300 speller is evident. However, the Speller technology can have a wide foot print only if its cost is reduced. Electroencephalogram (EEG) device is the major contributing factor to the cost of the speller. Our effort is directed towards using a low cost EEG device and to study if it can be used for the speller. We have taken the Farwell and Donchin's speller model as our base. This model uses a 6x6 key matrix. Previous studies around this model, use 8 or more sensors to correctly identify the target key. However since our low cost EEG device (Muse 2) has 4 sensors, we thought of beginning with a smaller (3x3) key matrix. Reducing the number of keys increases the probability of identifying the target key correctly. This study aims towards doing a comparative study of "accuracy" and "speed" of different sizes of BCI spellers using low cost, fewer sensors EEG device. This study is conducted on OpenViBE platform. It uses different classifiers (LDA, MLP and CNN) for processing the EEG signals for target identification. This study also identifies the amount of training data required for speller and it's consequent efficient classifier.

Introduction

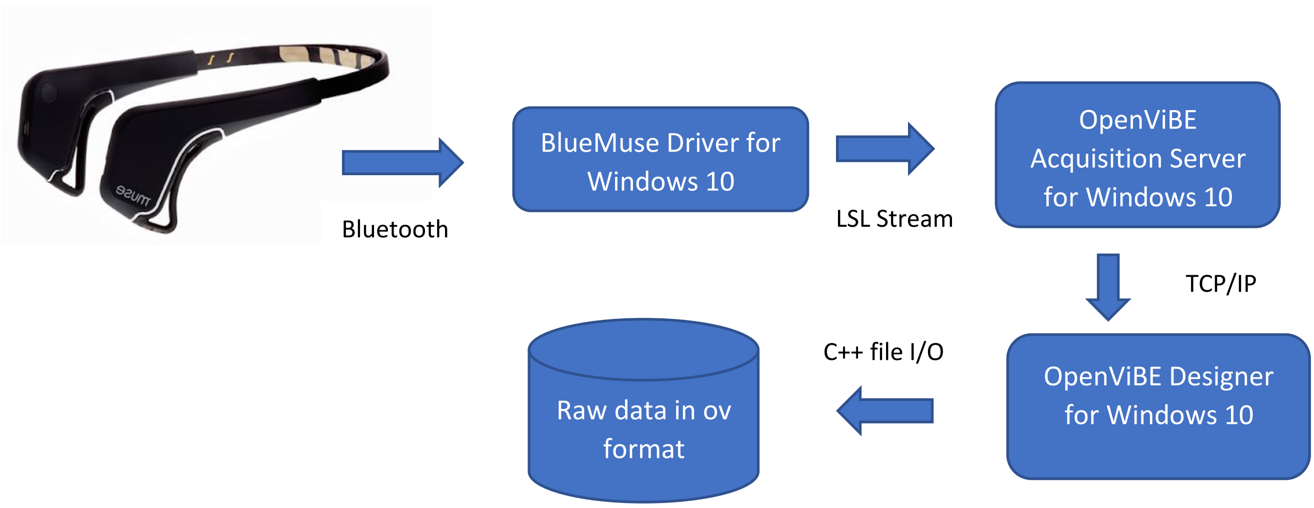
The Farwell and Donchin P300 speller is one of the important Brain Computer Interfaces used mainly for patients with Locked-In syndrome, who possess good cognitive ability. The speller is based on Oddball paradigm where the occipital lobe (the visual cortex) of the brain produces a P300 visual evoked potential, on encountering an infrequent deviant (target) stimulus among a sequence of similar repetitive stimuli. The name P300 refers to the fact that the peak of the signal appears after 300 msec on presenting the deviant stimulus to the subject. We are using Muse 2 device for our experiments. This device has two electrodes (AF7 and AF8) placed on the forehead and (TP9 and TP10) placed behind the ears as per 10-20 electrode positioning system. Muse 2, has no electrode placed on the occipital lobe. However, there are a couple of studies showing the presence of this visual evoked potential on Temporal and Frontal electrodes. Hence we decided to use this device for speller. We used OpenViBE platform for signal processing functionality. In order to take advantage of full range of TensorFlow's machine learning libraries, we extended the OpenViBE's python interfacing API and used TensorFlow's LDA, MLP and Conv1D classifiers and predictors. OpenViBE's speller user interface is used in order to generate the Oddball stimuli. The built-in speller user interface contains 6x6 matrix of characters and numbers. However, it is customized to contain 3x3 matrix of numbers. This paper will present a comparative analysis of 3x3 and 6x6 matrix performance with Muse device.

Experimental Setup

The experiment begins with interfacing Muse – 2 with OpenViBE using BlueMuse driver and collecting data from various subjects. The data collection process involves the following steps:

1. The BlueMuse Driver communicates with Muse -2 device over Bluetooth.
2. BlueMuse connects with OpenViBE's Acquisition Server over Lab Streaming Layer (LSL).
3. The Acquisition Server then sends the data to OpenViBE designer over TCP/IP.
4. OpenViBE designer captures the raw data in .ov format

Please look at the following figure to have a pictorial view of the above steps.



Below, we have shown the configuration of all the following components required for the project:

1. BlueMuse Driver
2. Acquisition Server
3. OpenViBE designer

Let's begin with BlueMuse Driver configuration.

Configuration of BlueMuse Driver

1. When the bluetooth connection with your Muse-2 device and the BlueMuse driver is alive, the following screen (Figure 1 Streaming from Muse 2Figure 1) appears when you open the BlueMuse driver user interface.

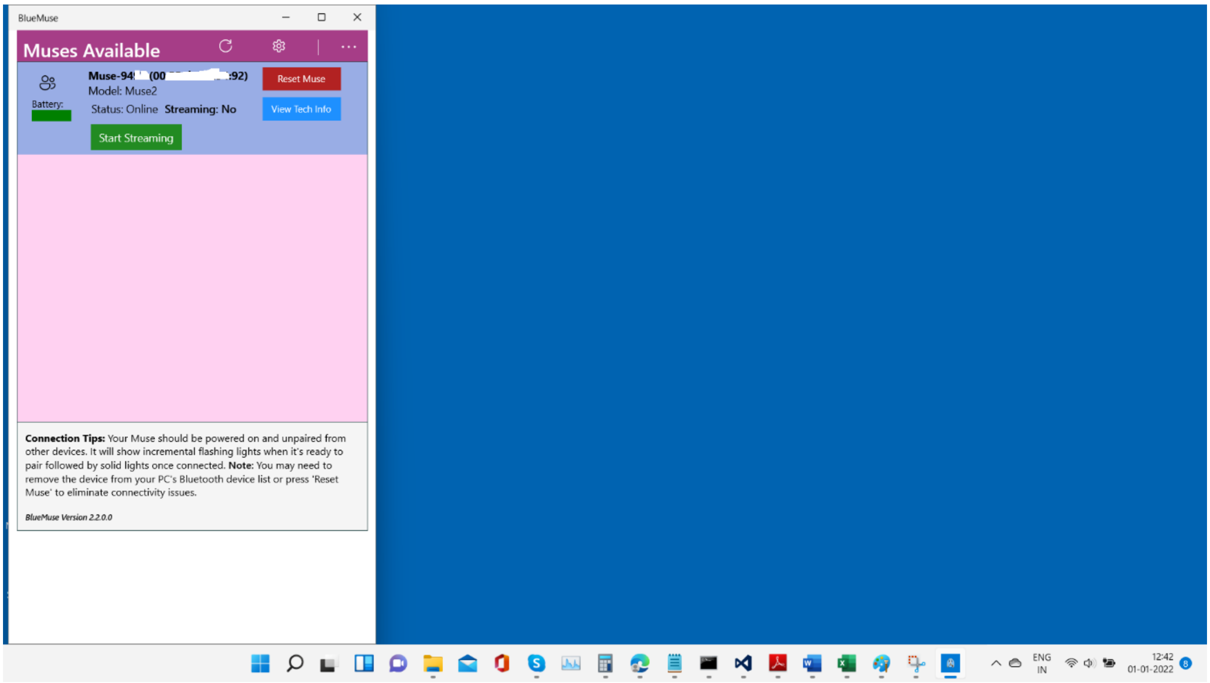


Figure 1 Streaming from Muse 2

2. On clicking the **Gear Icon** in the top menu bar, the following screen appears, where we have kept all default setting intact, except we have unchecked all the checkboxes except the **EEG Enabled** checkbox.

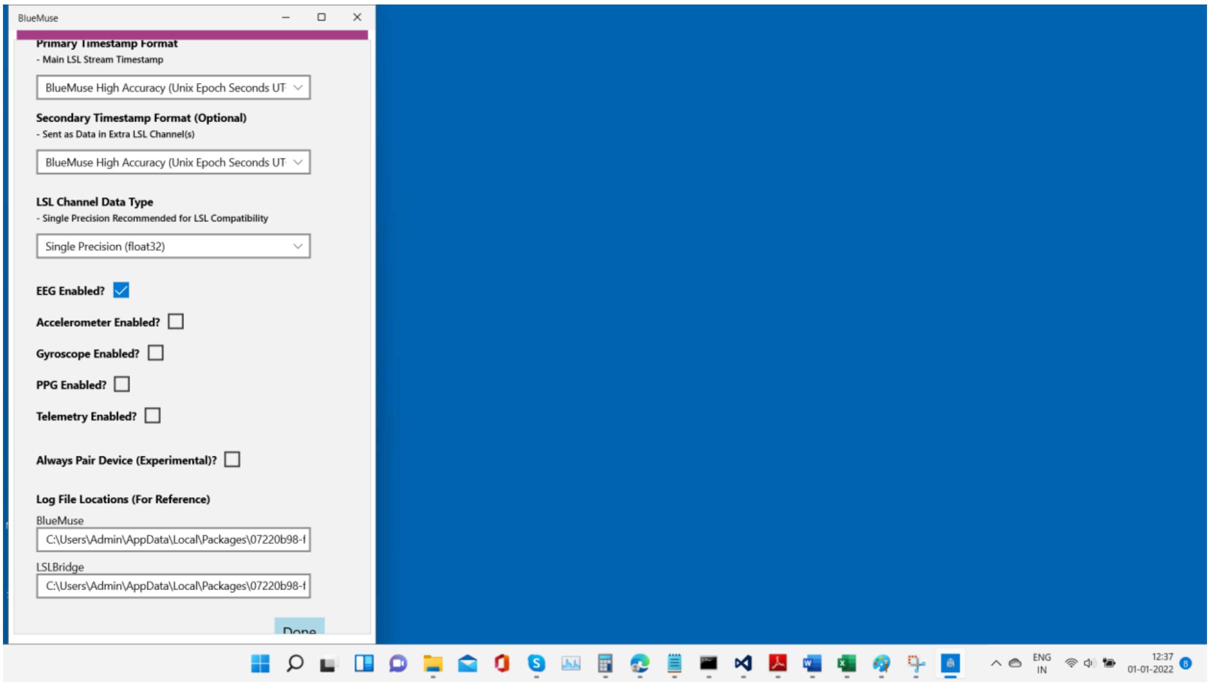


Figure 2 Muse 2 Driver Configuration for Streaming EEG

3. Once the configuration in Figure 2 is done, we can start streaming EEG to OpenViBE acquisition server.

Configuration of OpenViBE Acquisition Server

1. Run the OpenViBE acquisition server to get its user interface as shown below (in Figure 3). Keep the default settings, except choosing **the LabStreamingLayer** as the communication medium between BlueMuse and Acquisition server.

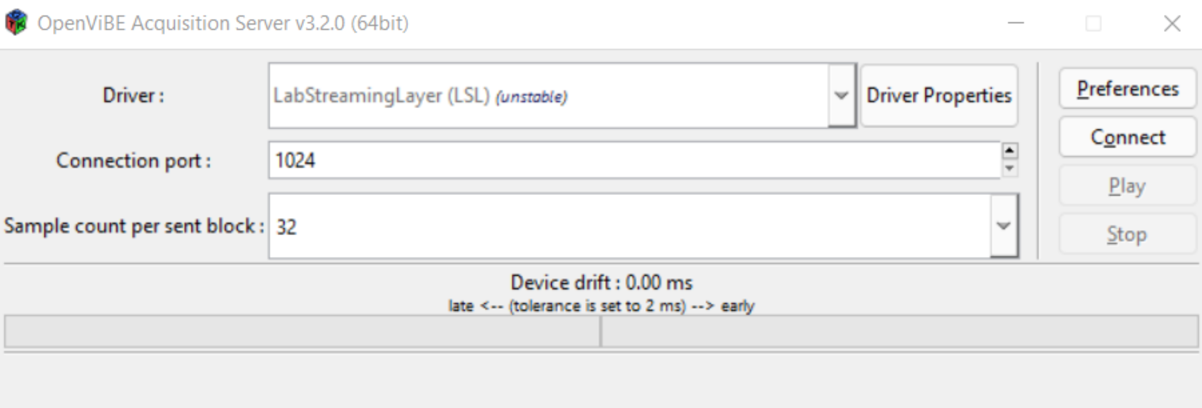


Figure 3 Configuration of OpenViBE acquisition server

2. Click on the **Driver Properties**. If things go smooth so far, **EEG** option will be shown in the signal stream box along with the device identity as shown in Figure 4. Click on Apply.

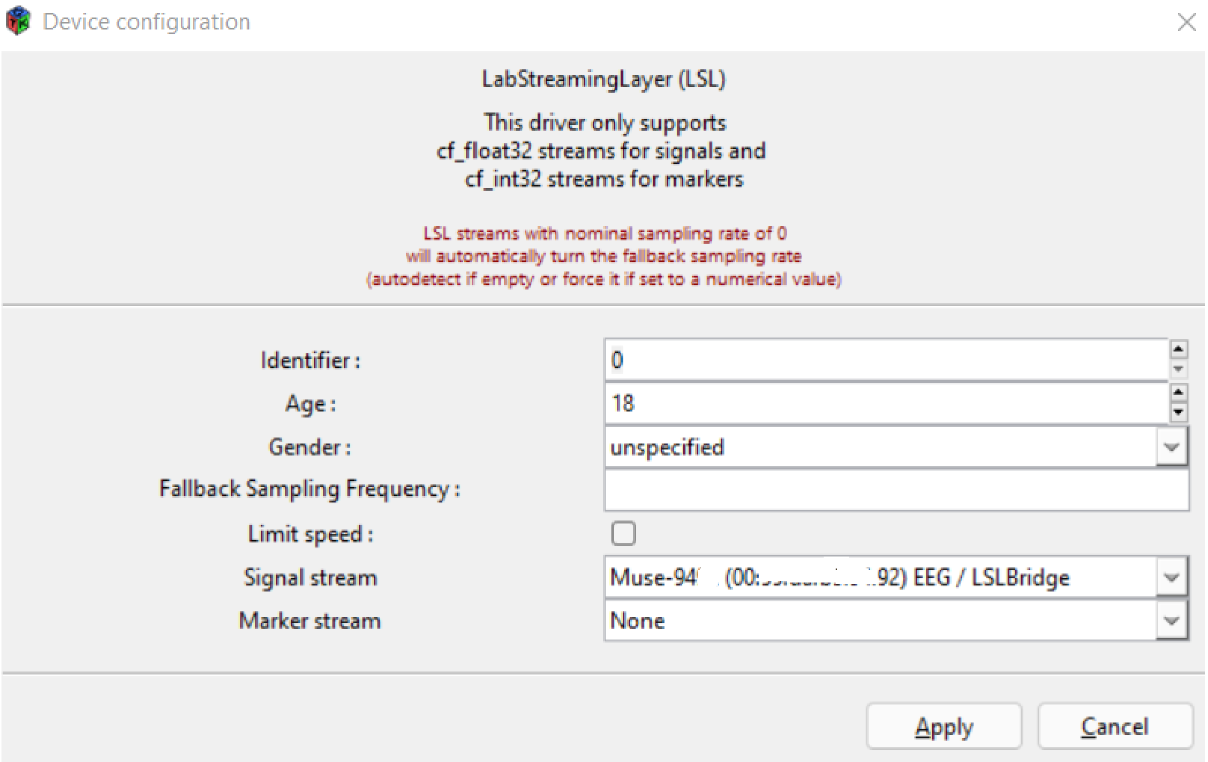


Figure 4 OpenViBE Acquisition Server Configuration - LSL

3. Click on **Preferences** button in Figure 3 and the user interface will appear as shown in Figure 5. Select the options/values as shown in the figure. Click on Apply to go back to Figure 3.

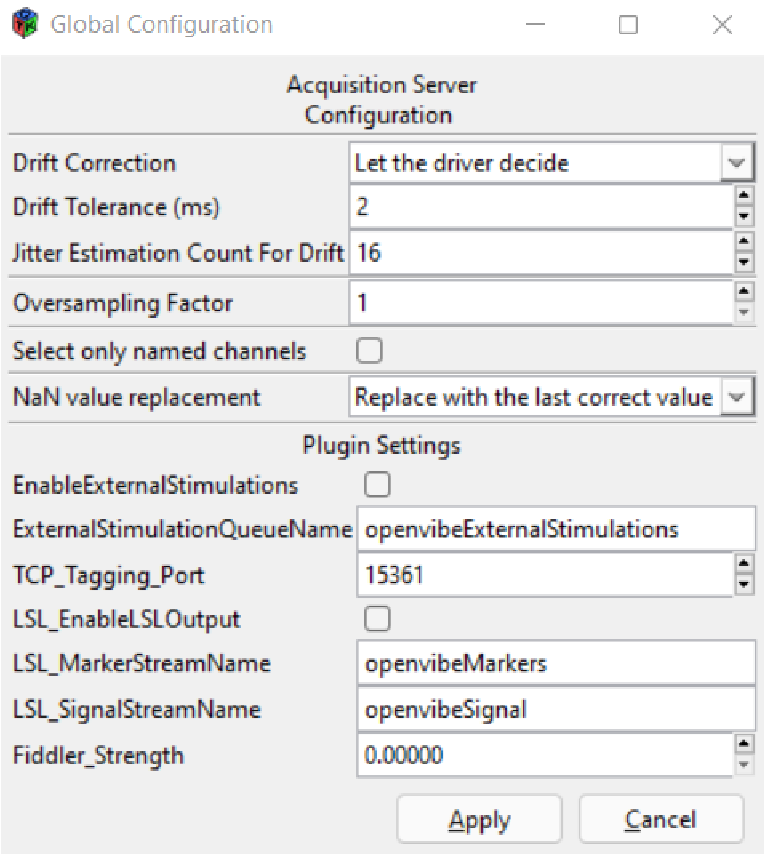


Figure 5 Acquisition Server Preferences

4. In Figure 3 click on **Connect** and **Play** to get the OpenViBE acquisition server running with LSL driver.

Configuration of OpenViBE Designer For Data Collection

Open the OpenViBE designer UI and open the **p300-speller-1-acquisition.xml**. Configure the **P300 Speller Stimulator** box as shown in the below figure (Figure 6). We have configured the flash duration to be 170 msec and non-flash duration to be 30 msec. This makes the stimuli repetition interval to be 200 msec. The number of rows in the speller UI are configured to be 3. Number of columns of the speller are also 3. This gives a 3 x 3 matrix of keys for the speller. We have configured each session to have 10 characters in the training session. When we run this scenario, a 3 x 3 speller looks as in Figure 7. Each of the training characters (target character) is highlighted in BLUE colour. Once the character gets highlighted, the subject has to focus on the character till it completes 24 flashes. The number 24, attributes to **12 repetitions x (1 row flash + 1 column flash = 2 flashes)**. After completing 24 flashes on the target, another character is presented to the subject in BLUE colour. Once all 10 characters are completed, the Speller User Interface closes and a data file is created in **.ov** format. A sample storage format is **p300-online-[2021.12.31-10.33.34].ov**. Rename this file such that files for different subjects can be identified correctly, for example: **Subject-1-p300-online-[2021.12.31-10.33.34].ov**. This file further needs to be processed to extract the epoch data for each of the stimulation.

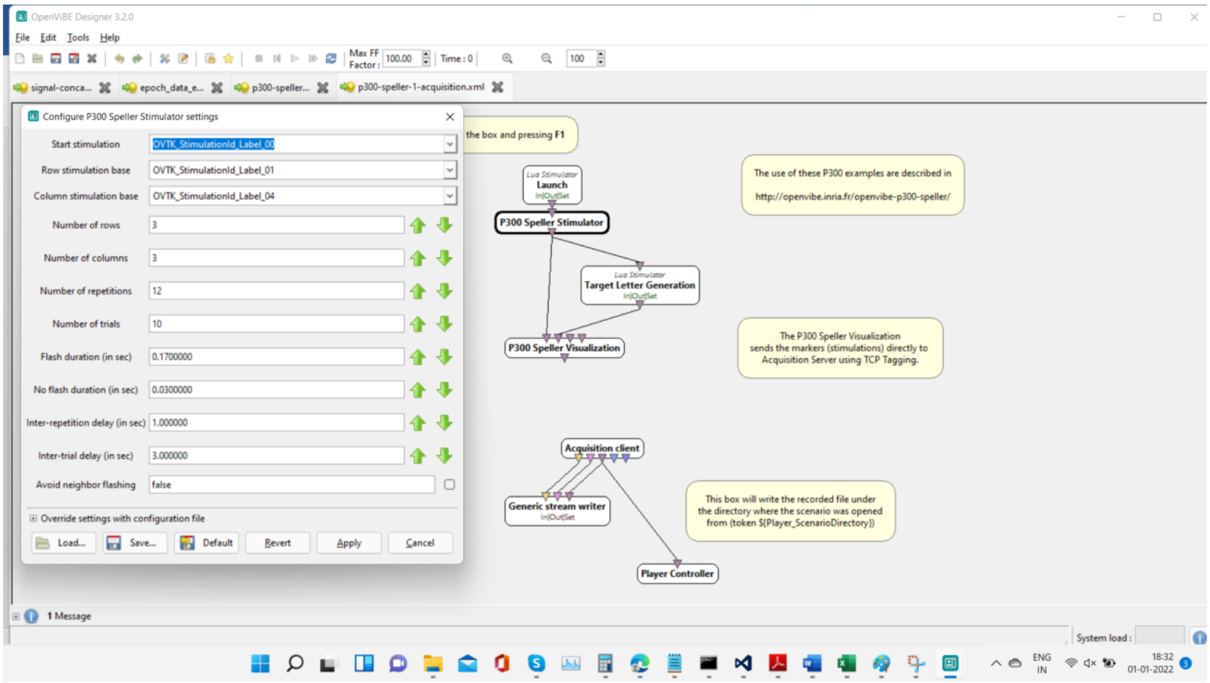


Figure 6 Data Acquisition Scenario of OpenViBE Designer

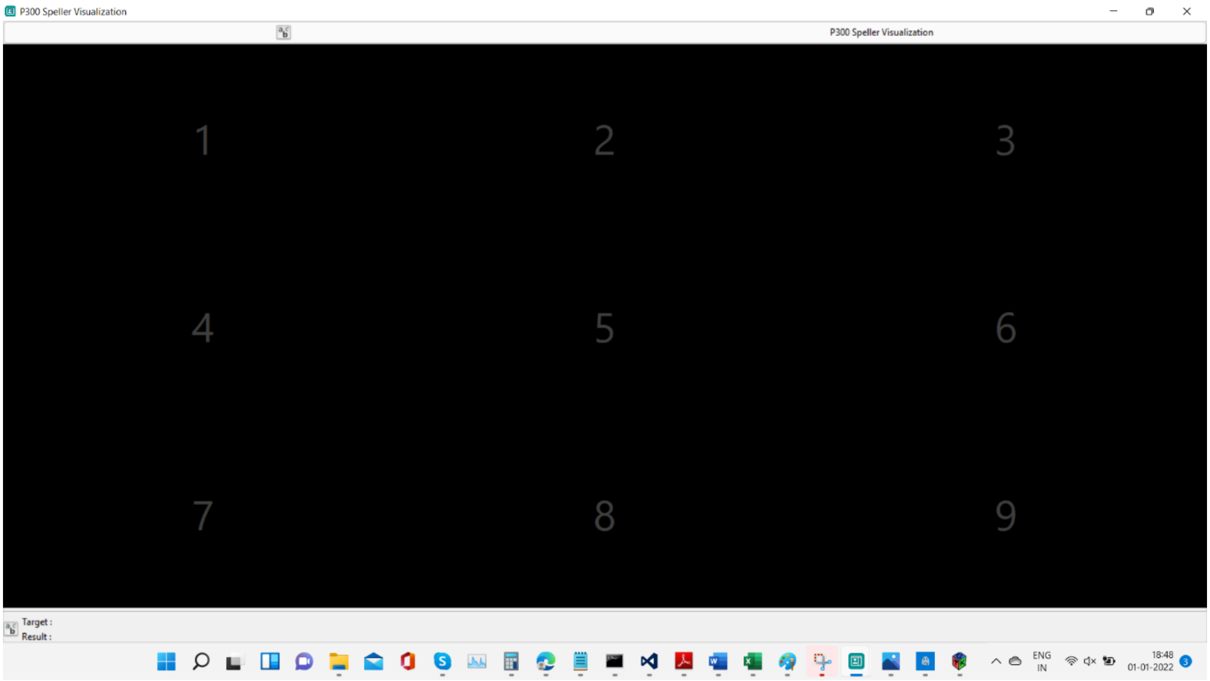


Figure 7 P300 Speller of size 3 x 3

Epoch Data Extraction

In the OpenViBE designer UI, open the scenario file **epoch_data_extraction.xml** (Figure 8). Configure the EEG channels (Figure 9) from which to extract the data, using the channel selector. In our condition Muse-2 provides four channels TP9, AF7, AF8 and TP10. Then configure the Temporal Filter for extracting the P300 signal as shown in Figure 10. Most of the literature specifies a frequency band of 1 Hz to 12 Hz for extracting the P300 signal. Hence we have extracted the same band from the raw data using 4th order Butterworth Bandpass Filter. The ripple is set to its default configuration by OpenViBE. The signal is then decimated by 8 using Signal Decimation Box (Figure 11). The epochs of 600 msec are extracted posterior to each stimulation. This duration of epoch is chosen based on the literature for extracting P300.

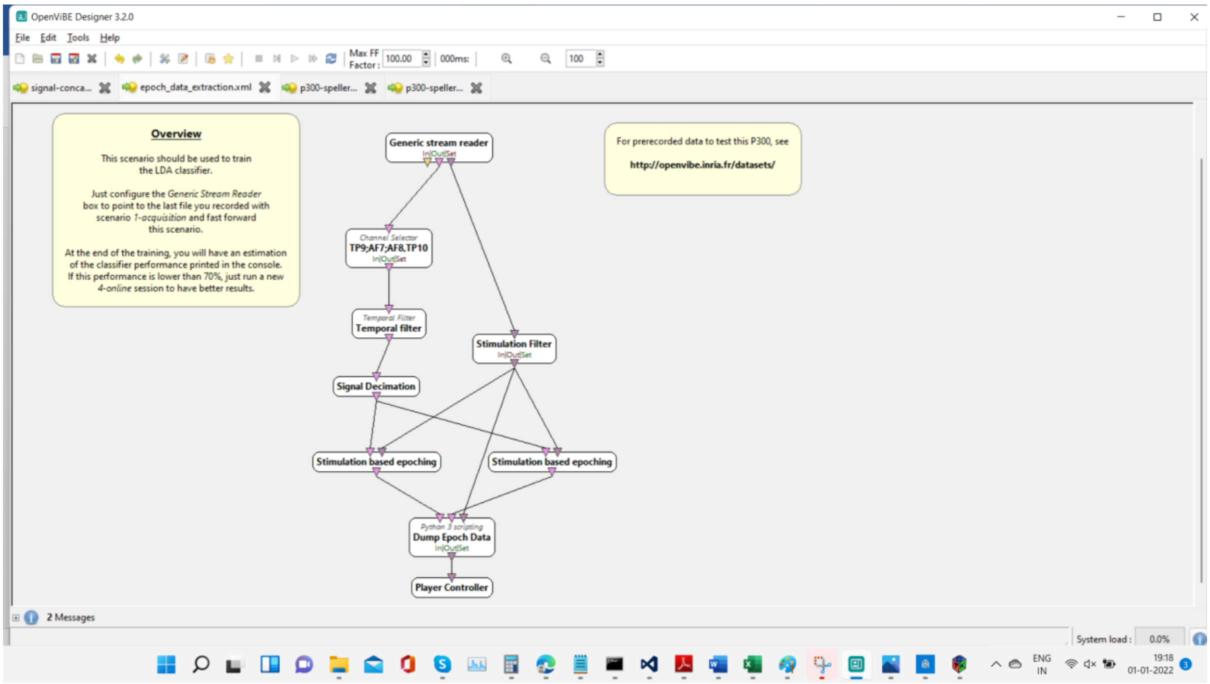


Figure 8 Epoch Data Extraction Scenario

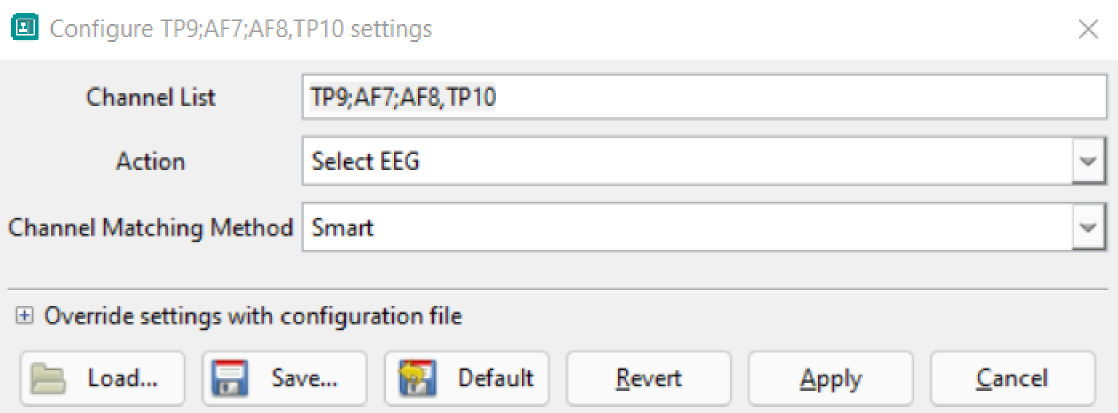


Figure 9 Configuration of Channel Selector Box

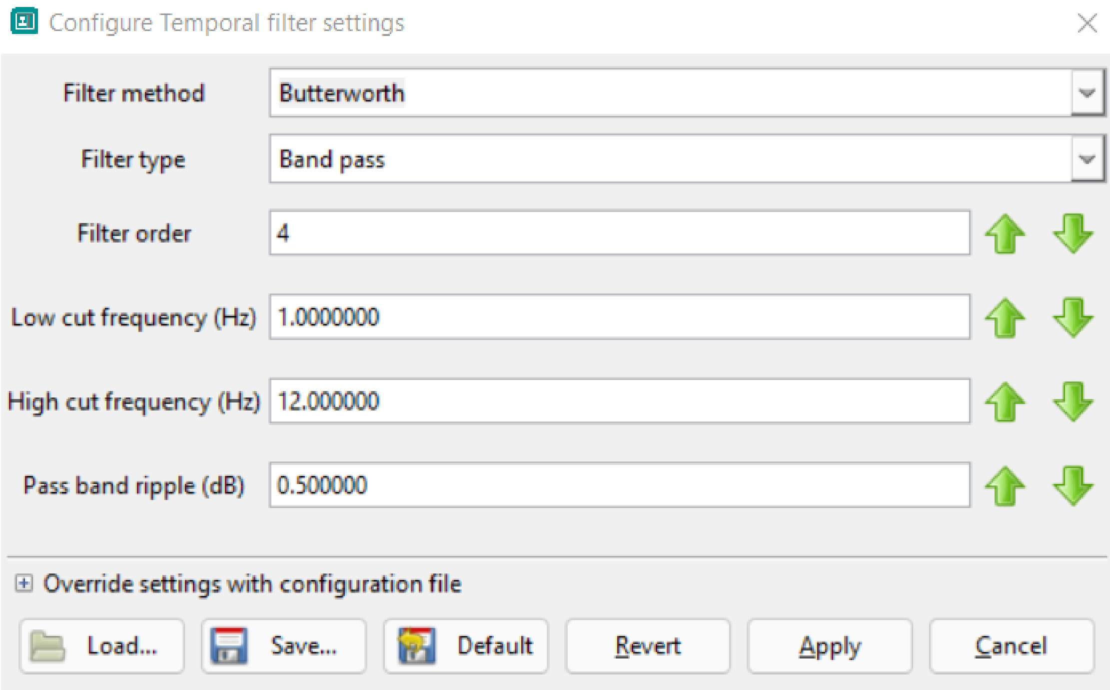


Figure 10 Bandpass Filter to Extract P300

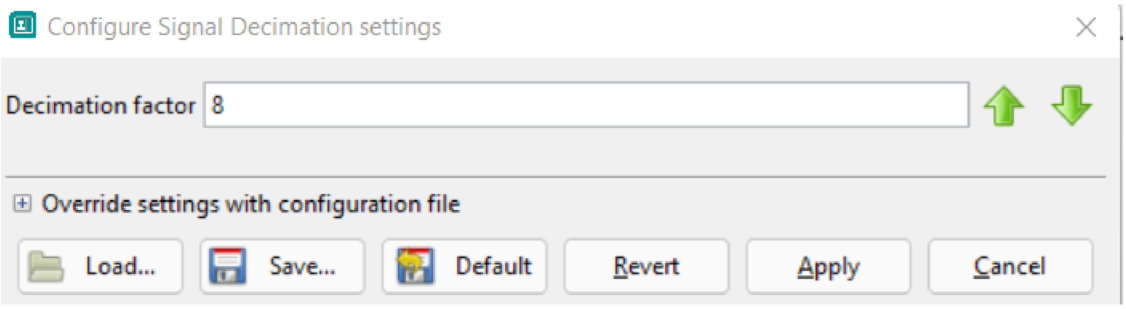


Figure 11 Decimation of Raw Data

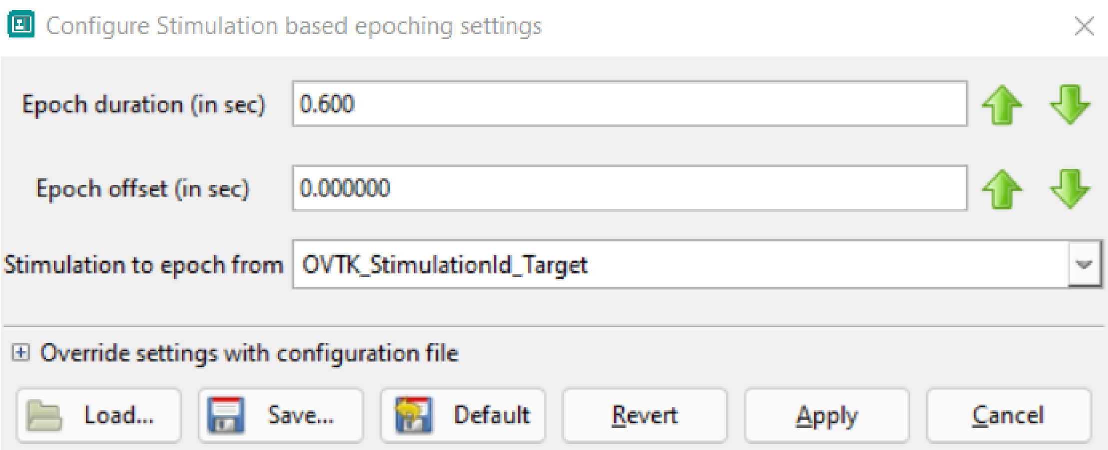


Figure 12 Set Epoch Duration

How to configure a 3 x 3 speller

References

OpenViBE (inria.fr)