NeuroPi: A Portable Steady-State Visually Evoked Potential-based Brain-Computer Interface

Vitor M. Barbosa, Sarah N. Carvalho, Harlei M. A. Leite

***Abstract*—The NeuroPi system is a Brain-Computer Interface (BCI) based on Steady-State Visually Evoked Potentials (SSVEP). This system employs brain activity responses induced by oscillatory visual stimuli to enable intuitive and efficient control of external devices. The system was designed to non-invasively gather brain signals using electroencephalography. A low-cost biosignal amplifier was employed to capture, filter, and digitize these signals. Python scripts were utilized for signal processing in the stages of preprocessing, feature extraction, feature selection, and classification, ensuring seamless integration, customization, and straightforward adaptability. NeuroPi was designed with a focus on simplicity and user-friendliness, enabling the integration and control of electronic devices using brain signals. Additionally, the system is portable, cost-effective, and efficient, making it suitable for various real-world applications. Performance tests validate NeuroPi's effectiveness, highlighting its potential to contribute to the popularization of SSVEP-based BCI systems. The NeuroPi system is available to be freely used for research and development purposes.**

***Index Terms*—Brain-Computer Interface, Steady-State Visually Evoked Potential, Embedded System.**

I. Introduction

BRAIN-Computer Interface (BCI) is a closed-loop system that establishes a communication channel between a brain and electronic devices through the interpretation of cerebral signals [1]. Due to its independence from muscular movements, its utilization holds appeal in the advancement of assistive technologies [2]. The development of an efficient BCI system requires an extensive understanding of neuroscience and engineering and its state-of-the-art is in the phase of use in a controlled environment with experts [3].

Fig. 1 shows a typical architecture of an electroencephalography (EEG)-based BCI system. Initially, the signal is acquired through electrodes and then digitized. It subsequently undergoes pre-processing to eliminate unwanted artifacts and enhance the signal-to-noise ratio. Following, features are extracted and selected to feed the classifier. The classifier maps these features into actionable commands for a defined application. This iterative process continues until the user no longer wishes to control the device.

Vitor M. Barbosa (ORCID: 0000-0002-1540-5639) is with the NTT Data, São Paulo, Brazil. Sarah N. Carvalho (ORCID: 0000-0003-2316-7244) and Harlei M. A. Leite (ORCID: 0000-0001-9467-2120) are with the Technological Institute of Aeronautics (ITA), São José dos Campos, Brazil. The authors thank CNPq for the financial support and the Federal University of Ouro Preto for the institutional support.

NeuroPi is available at https://github.com/vitor-martinsb/NeuroPi. Please ensure to properly cite this manuscript when incorporating its content into your work. Contact e-mails: {harlei, sarahnc}@ita.br.

Submission: 2023-11-08, First decision: 2023-12-14, Acceptance: 2024-07-30, Publication: 2024-08-06.
Digital Object Identifier: 10.14209/jcis.2024.12

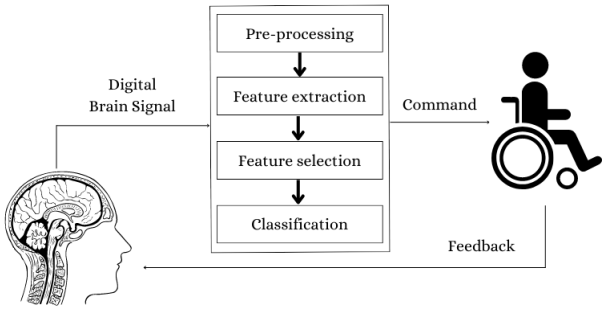

Fig. 1. Standard architecture of a BCI system. User interaction with the application occurs through the interpretation of brain signals. The signal is acquired through EEG, subjected to a signal processing module comprising pre-processing, feature extraction, feature selection, and classification stages, which map the features into control signals. The NeuroPi follows this architecture.

The Steady-State Visually Evoked Potential (SSVEP) is a widely used technique in the field of BCI. It involves the presentation of visual stimuli that flicker at specific frequencies. When the user is exposed to these visual stimuli, it generates electrical responses at the same frequencies as stimuli. These responses can be detected using EEG. By analyzing the EEG signals, it is possible to identify which specific frequencies the user is focusing on. Thus, to control an application, each visual stimulus is associated with a specific command to an external device.

SSVEP-based BCIs are known for their high accuracy and relatively fast response times compared to other BCI paradigms. This makes them suitable for various applications, including assistive technology for individuals with disabilities [4].

The development of SSVEP-based BCI systems presents several challenges that involve:

- Signal quality: One of the primary challenges is obtaining clean and reliable SSVEP signals. The EEG signals can be affected by various noises and artifacts, including muscle activity and environmental interference, which can degrade signal quality [5].
- Safety: SSVEP responses vary among the people. SSVEP-based BCI systems pose a potential risk for individuals with a history of epilepsy. Implementations that differentiate high frequencies can make this technology safer for these subjects [6].
- Fatigue and adaptation: Prolonged use of SSVEP-based BCIs can lead to user fatigue and neural adaptation, potentially reducing the BCI's effectiveness over time [7].
- Accuracy of the system: Implementation of robust signal processing techniques capable of handling noise and the

low amplitude of the acquired brain signal, accurately identifying the information related to the user's desired command [8].

- Portability and user-friendly design: A portable BCI system should be easy to set up and use, allowing users to quickly get started without extensive preparation. Furthermore, intuitive interfaces and user-friendly controls enhance the portability and popularization of the system [9].

This study addressed this last problem, with the development of an embedded SSVEP-based BCI system named NeuroPi [10]. The primary feature of the NeuroPi is its operation on a Raspberry Pi 3 model B [11], a compact single-board computer equipped with various types of communication interfaces and easily integrated with electronic devices, as shown in Fig. 2, facilitating its integration into different types of applications. For the acquisition of EEG signals, an OpenBCI was used [12], it is low-cost with a configurable hardware platform and open-source tools. The embedded BCI was designed to be plug-and-play, eliminating the need for technical expertise in its setup, and making it possible for operation by users without technical knowledge, after the device has been configured for the first time.

The primary contribution of this work lies in the creation of an accessible and user-friendly BCI system. By leveraging the affordability and versatility of the Raspberry Pi 3 Model B and the OpenBCI platform, the NeuroPi system significantly lowers the barrier to entry for BCI technology. Its plug-and-play design ensures that users without technical expertise can operate the system effortlessly after the initial setup. This innovation democratizes access to BCI technology, potentially broadening its applications in various fields such as assistive technology, communication, and neurofeedback.

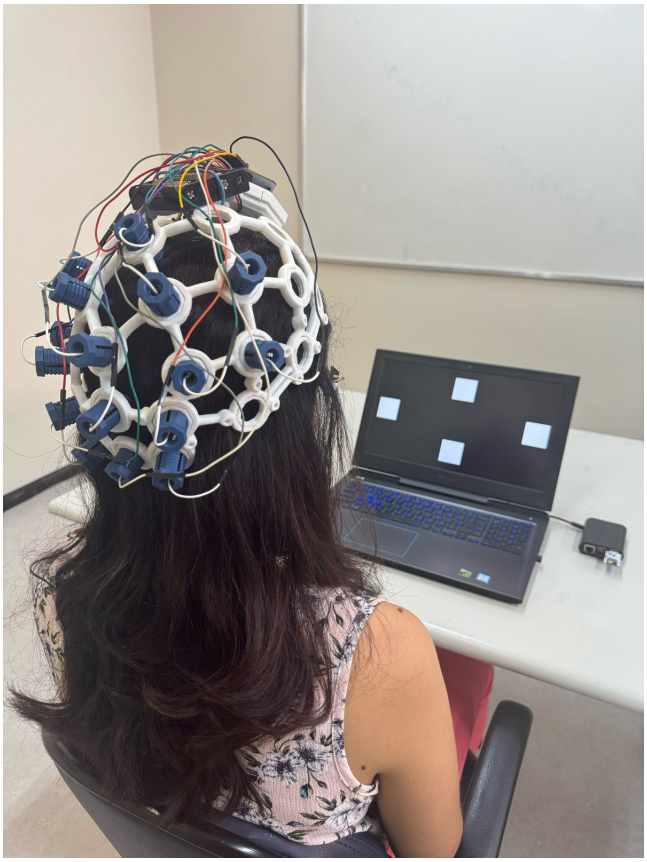

Fig. 2. User operating the NeuroPi. The setup features a laptop displaying four visual stimuli, with the Raspberry Pi positioned on the right side of the table, receiving data sent by the OpenBCI. The user is wearing a cap with 16 dry electrodes distributed across the scalp. Each electrode is seamlessly connected to the OpenBCI board, conveniently positioned at the front of the cap.

II. NeuroPi system design

NeuroPi was designed according to the classic architecture of BCI systems (Fig. 1) with five modules: (i) acquisition, (ii) pre-processing, (iii) feature extraction, (iv) feature selection, and (v) classification. We present the implementation details of each module in the following.

A. Acquisition module

The acquisition module serves as the interface between the BCI system and the OpenBCI EEG signal acquisition equipment. In this study, our system was based on the Cyton board, enabling the simultaneous capture of data from 8 channels, each sampled at 250 Hz, with 24-bit channel data resolution. To accommodate more than 8 electrodes, the Daisy expansion board could be employed, connecting to Cyton and expanding the channel count to 16, sampled at 125 Hz. This extension allows for comprehensive spatial coverage of the scalp [12]. Communication between OpenBCI and the NeuroPi occurred via a Bluetooth 4.0 Low Energy (BLE) RF Module, facilitated by a custom code running on a Raspberry Pi, as shown in Fig. 3.

We employed a 3D-printed Ultracortex Mark IV cap [13] to ensure precise electrode positioning on the scalp, following the established 10-10 standard pattern [14]. The electrode placement process on the cap is a one-time task for each user, with occasional adjustments to ensure proper contact with the scalp. The electrodes used are dry, eliminating the requirement for any type of fixing gel. To check electrode scalp contact, an impedance test is performed.

The code for the acquisition module has been developed in Python, and the OpenBCI libraries were adapted to ensure compatibility with the ARM (Advanced RISC Machines) architecture. For BCI system designers considering the use of the NeuroPi, the acquisition code has been parameterized, simplifying the process of selecting the desired number of channels. This flexibility enables it to function with either the Cyton board alone or in conjunction with the Daisy expansion board. Furthermore, the acquisition module can be adapted to integrate other EEG equipment with the NeuroPi, providing versatility for different EEG system configurations.

B. Pre-processing module

During the preprocessing stage of the NeuroPi system, we employed two filters: an 8th-order Butterworth filter with a passband ranging from 5 to 50 Hz, and a 2nd-order notch filter at 60 Hz. Subsequently, a Common Average Reference (CAR)

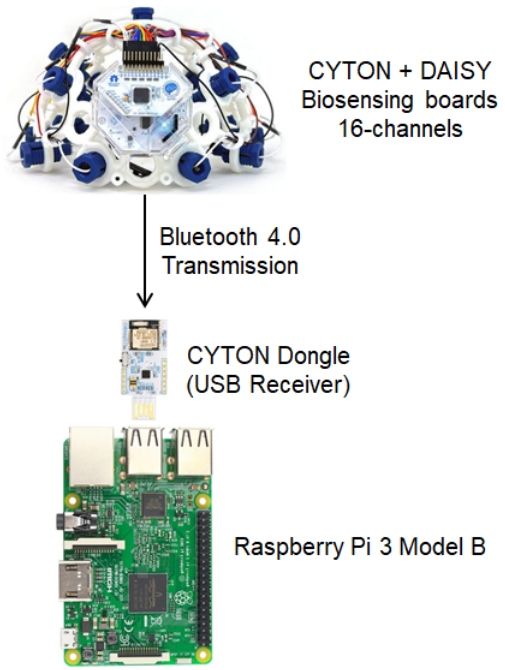

Fig. 3. Communication between OpenBCI and Raspberry Pi via Bluetooth.

was applied to remove common components across all electrodes [15], as they are not related to the brain activity under evaluation. The goal of preprocessing is to enhance the signal-to-noise ratio and facilitate the detection of relevant patterns of brain activity. These techniques can be activated/deactivated by the user, and the system is designed to allow the integration of additional preprocessing options.

In mathematical terms, the filtered signal V_i^{CAR} at the i-th sample is computed as the potential measured V_i minus the average signal across all N channels, as follows:

$$V_i^{CAR} = V_i - \frac{1}{N}\sum_{j=1}^{N} V_j \tag{1}$$

Despite its mathematical and computational simplicity, CAR can adjust the electrode potential reference and mitigate some of the artifacts present in the signal. This effectiveness is attributed to the fact that most artifacts that interfere with the SSVEP response typically appear simultaneously and with comparable intensity across all electrodes. In contrast, the signal of interest exhibits greater intensity in only a select few electrodes, particularly those located in the occipital zone [16].

C. Feature extraction module

The feature extraction module consists of representing the signal compactly, allowing the discrimination of stimulus classes through their characteristics [17]. This step prepares the signal so that the classifier can operate directly in the feature space.

NeuroPi conducts feature extraction in the frequency domain by computing the magnitude of the Fast Fourier Transform (FFT) at the frequencies of the visual stimuli and their harmonics, as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi \frac{kn}{N}} \tag{2}$$

where, $X[k]$ represents the component at frequency k, $x[n]$ is the input signal in the time domain and N is the total number of samples in the signal.

D. Feature selection module

In the context of BCI systems, the feature selection algorithm plays an important role in identifying the channels that contain the most pertinent information for the classification process. Its objective is to enhance performance, minimize signal processing time, and enable the system to achieve a better representation [18]. There are numerous feature selection techniques commonly used in BCIs, and all of them can be integrated into the system.

NeuroPi utilizes a filter-based feature selection technique known as the Pearson Correlation Coefficient [19]. Pearson's correlation coefficient assumes linearity in the relationship between variables, normal distribution of data, and the absence of outliers. It can be sensitive to outliers and may not capture nonlinear relationships. Equation 3 shows how the correlation R_c is calculated, considering an input vector x_c associated with class y, representing the visual stimuli to be discriminated.

$$R_c = \frac{cov(x_c, y)}{\sqrt{var(x_c)var(y)}} \tag{3}$$

The correlation coefficient R_c for the c-th feature (channel) varies within the range $[-1, +1]$. When there is a strong correlation between the feature vector x_c and the label y, it results in R_c being close to $|1|$. This indicates that the feature vector makes a valuable contribution to a classification system. Conversely, as R_c approaches zero, the correlation between the feature and the label becomes negligible, suggesting that the feature vector can be eliminated.

Therefore, feature selection involves ranking the absolute value of the correlation coefficient, with the top k ranked channels being integrated into the feature matrix destined for the classifier. The parameter k can be determined through cross-validation or set, for example, as half of the total number of channels used. The maximum value for k could indeed be set to match the number of EEG channels being read, but with this configuration, no feature selection would take place.

E. Classification module

A classification system maps the input signals to the possible output classes. In the context of BCI systems, the objective is to determine which of the visual stimuli the subject was concentrating on, using the recorded brain signal.

The classification process consists of three stages: (i) training, (ii) validation, and (iii) operation [20]. In the training stage, a labeled database is used to "teach" the system,

essentially identifying the hyperplanes or surfaces that distinguish each of the classes. Subsequently, a separate dataset is employed to validate the effectiveness of the generated discrimination system and to assess the classifier's performance. Once validated, if the performance is satisfactory, the classifier can be deployed in operational mode.

NeuroPi utilizes a low computational cost algorithm for the classifier. The Least Squares linear classifier aims to find hyperplanes that best separate different classes of data by minimizing the sum of squared errors. The linear model for a multi-class problem is given by:

$$\hat{y} = X^T w \tag{4}$$

where $\hat{y}$ is the predicted output, X is the feature matrix and w represents the weight vector, which determines the direction and slope of the decision boundary.

The goal of the Least Squares linear classifier is to minimize the sum of squared errors between the predicted outputs and the actual class labels for the training data, given by:

$$J(w) = \sum_{i=1}^{N} (y_i - X_i^T w)^2 \tag{5}$$

where N is the number of training samples, y_i is the actual class label for the i-th sample, X_i is the feature matrix corresponding to the i-th sample, and w is the parameter to be estimated. To ensure the hyperplane's degrees of freedom, the classifier bias was integrated into the feature matrix, as elaborated further in Fig. 5.

We can use the Monroe Penrose's approach to find the optimal w:

$$w = (X^T X)^{-1} X^T y \tag{6}$$

where $\hat{y} \geq 0$ indicates that the data belongs to the class, and $\hat{y} < 0$ indicates that it does not belong to the class [18]. The classifier was designed using a one-versus-all approach. Thus, after projecting w for each possible class, $\hat{y}$ is computed for each one, with the maximum value indicating the class to which the treated sample belongs.

F. Implementation details

The modules for pre-processing, feature extraction, feature selection, and classification vary according to the operating mode of NeuroPi. In the offline mode, the primary objective is to effectively train the classifier system, while in the online mode, the user harnesses the BCI to operate a device. The following subsections provide a comprehensive overview of the implementation specifics for these two operational modes.

1) Offline mode: During the offline operating mode, the NeuroPi classifier is trained using the user's brain signals. To achieve this, the user is exposed to visual stimuli flickering at different frequencies while their EEG signals are recorded. The number and the selection of these visual stimulus frequencies are parameters that can be customized.

We have established a standardized acquisition process for the training phase in NeuroPi with 8 trials, each lasting 12 s for every stimulus. After, the 12 s signals are windowed. The window size determines the user's interaction speed with the device. For instance, if the window size is set to 3 s, it means the user must focus their attention for 3 s to issue a command to a device. The offline classifier is trained using signals of the same length as expected in the online mode. All training parameters, including the number of stimuli, stimulus frequencies, trial counts, and window sizes are easily adjusted. These adjustments directly impact the Information Transfer Rate (ITR) of the BCI system, as shown in Fig. 4, which presents the ITR for a BCI with a windowing of 3 s with 2, 4, 8, 16, and 32 possible choices.

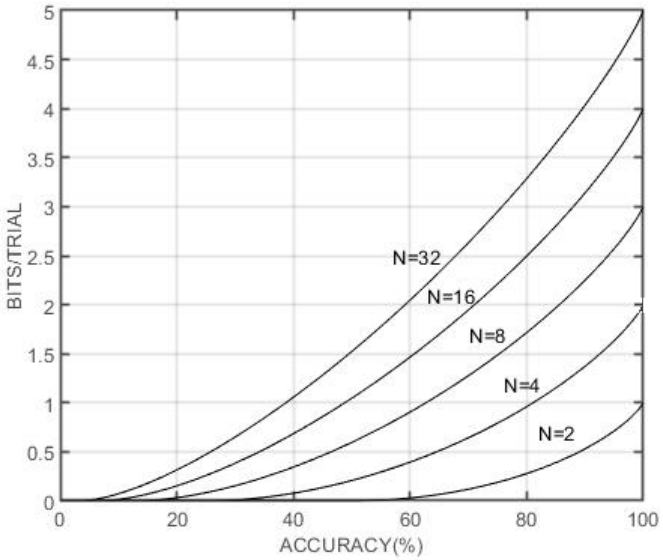

Fig. 4. Information transfer rate in bits/trial (bits/selection) and in bits/min (for 20 trials/min) when the number of possible choices (i.e. N) is 2, 4, 8, 16, and 32.

The feature extraction is applied to each windowed signal, resulting in a single matrix with dimensions of $M \times F.N$, where M is the number of samples, and $F.N$ is the product of the number of stimuli F and the number of channels N. For the linear classifier, a bias term is introduced to account for translation in the hyperplane within the feature space. This addition involves appending a column of 1 s to the feature matrix. Fig. 5 shows the feature matrix used for training a linear classifier. This particular example considers 4 visual stimuli blinking at frequencies of 6, 10, 12, and 15 Hz, using 8 channels and a 3-second window size. The dataset includes 8 trials, each lasting 12 s. As a result, there are $M = 128$ rows (4 stimuli x 4 windows/trial x 8 trials) and 33 columns (4 stimuli x 8 channels + 1). Each element in the feature matrix represents the magnitude of FFT estimated at the fundamental frequency of the stimulus for each channel.

The feature matrix is divided into 75% of the samples to train and the remaining 25% to validate the classifier, maintaining the proportion of samples for each stimulus. Each sample in the feature matrix corresponds to an entry in the associated label matrix, which indicates the class to which each sample belongs. In our approach, we use the values 1 and -1 to include or exclude from the class, respectively. Fig. 6 shows an example of a label matrix, considering four

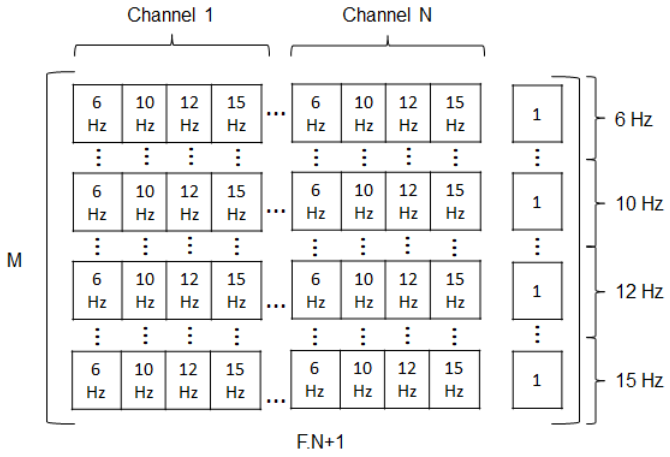

Fig. 5. Feature matrix. In this example, there are $M = 128$ samples, $F = 4$ visual stimuli, and $N = 8$ or 16 channels.

different visual stimuli.

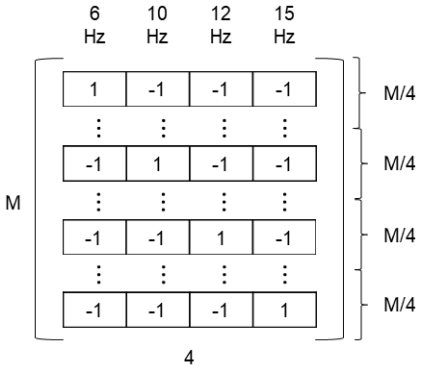

Fig. 6. Label matrix for the classification system: Each row represents a sample, and each column corresponds to a class label. Entries indicate whether a sample belongs to a particular class (1) or not (-1).

Once the feature matrix is partitioned, we can proceed with the feature selection stage. A common approach is to include the top-ranked channels determined by the Pearson correlation coefficient. Subsequently, the classifier is trained using the training data partition, and the classifier's accuracy is estimated using the validation partition. A widely used criterion for feature selection is to continue including channels until no further improvement in classifier performance is observed.

Concluding these stages, the BCI is ready to be used in the online mode, with its performance estimated based on the validation set. The k-fold cross-validation technique can be employed to enhance the BCI accuracy estimation.

2) Online mode: When the NeuroPi is powered on, the EEG signal is continuously captured. The first acquisition window is discarded to eliminate transient noise. Afterward, the signal is processed at each configured windowing interval, generating a control signal for the application. The processing time typically takes around 1 ms. Fig. 7 shows the online mode operation, using a 3-second windowing as an example.

The application controlled by NeuroPi receives commands through the GPIO pins of the Raspberry Pi Model B. When an associated frequency is activated, a logic value of 1 is

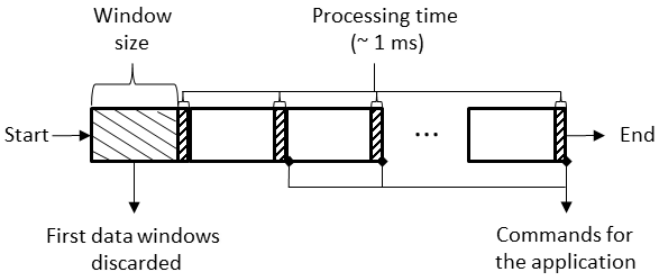

Fig. 7. Online mode operation. When the system is initiated, the first window is discarded to allow the user time to begin concentration. Between each acquisition window, there is an interval of approximately 1 ms for the execution of the processing and classification module, as the Raspberry Pi operates sequentially.

transmitted to a specific GPIO pin. NeuroPi was initially preconfigured to work with four frequencies: 6 Hz (GPIO20), 10 Hz (GPIO21), 12 Hz (GPIO22), and 15 Hz (GPIO16), but it has the flexibility to support additional frequencies by defining more GPIO pins on the board.

III. Methodology

To evaluate the NeuroPi system, two scenarios were considered: the first, involving 8 electrodes ($O1$, Oz, $O2$, $PO3$, POz, $PO4$, Pz, and Cz), and the second, involving 16 electrodes ($O1$, Oz, $O2$, $PO3$, POz, $PO4$, Pz, Cz, $C3$, $CP1$, $P3$, $CP5$, $C4$, $CP6$, $P4$, and $CP2$). The reference and ground electrodes were positioned on the mastoid processes. Dry EEG electrodes with a 5 mm dimension, developed by OpenBCI, were used and positioned on the user's head through a 3D-printed cap, as depicted in Fig. 2.

The NeuroPi system was configured on a Raspberry Pi 3 Model B and connected to the OpenBCI EEG acquisition system. The visual stimulus was developed in the Unity game engine and displayed on a laptop, taking the form of a square that flickers, alternating between black and white at frequencies of 6, 10, 12, and 15 Hz. There is no direct connection between the laptop and the Raspberry Pi, as the purpose of the stimulus is to generate an SSVEP signal that will be identified by the NeuroPi. Other formats and means of reproducing visual stimuli can be used, such as the use of LEDs and various images. NeuroPi does not provide a visual stimulation interface because each application requires a specific type of visual stimulus. The design of how a stimulus should be, as well as its construction, must be thought out in conjunction with the application, following the functional and non-functional requirements of the application.

A healthy male volunteer aged 23 participated in the experiment. The experiments were organized into two stages. In the first stage, the cap was placed on the user's head, and the training phase (offline) was initiated, involving 10 acquisitions of 12 s for each visual stimulus. In the second stage, a 30-second acquisition was performed for each visual stimulus in the scenario with 8 electrodes, and 21 s for each visual stimulus in the scenario with 16 electrodes, to carry out the system validation phase (online). The number of collections conducted is a configurable parameter in the NeuroPi. The

acquisition protocol was approved by the Ethics Committee of the University of Campinas (n. 791/2010).

The data collected in the training and validation phases were windowed in 3-second intervals, allowing the BCI to issue a command every 3 s. The window size is related to the BCI's operating speed, where smaller windows enable faster interaction, and larger windows allow for slower interaction. A smaller window size makes the classification process more challenging, as it receives a smaller amount of data compared to larger windows. The window size is a configurable parameter in the NeuroPi.

The classifier results were evaluated using the mean accuracy and standard deviation. The objective of the experiments was to demonstrate that the NeuroPi is capable of executing all stages of a BCI system, encompassing acquisition, processing, and classification. This underscores its viability as a suitable option for BCI research and the creation of applications controlled by BCI systems.

IV. Evaluation and Validation

The NeuroPi employed the techniques traditionally used in SSVEP-based BCI systems. Our research group has previously presented performance analyses of these techniques in the context of BCI-SSVEP systems in papers [8] and [9].

In this section, we provide a preliminary evaluation of the recorded EEG signal quality and a performance analysis of the NeuroPi system using 8 and 16 electrodes.

A. Acquisition quality

The EEG signal recorded at Oz electrode is presented in Fig. 8 in the time domain (considering a 3-second window without overlap) and in the frequency domain (showing the useful information band from 5 to 20 Hz) for 10 Hz, and allows a comparison of the effects of the Butterworth, notch, and CAR filters on the quality of the acquisitions. In the time domain, a slower oscillatory pattern is visible, primarily due to the filtering of the power line frequency at 60 Hz. Additionally, there is an attenuation in the signal's amplitude due to the action of filters and the A/D conversion process used in the OpenBCI platform for each channel. In the frequency domain, it is noticeable that the peaks at the respective frequencies of interest (10 Hz) become more pronounced after filtering, enhancing the signal-to-noise ratio.

B. NeuroPi with 8 electrodes

Tab. I presents the accuracy rates of NeuroPi in both offline and online modes using 8 electrodes (O1, Oz, O2, PO3, POz, PO4, Pz, and Cz). Due to the limited number of channels, the feature selection was not operated. Results indicate better performance in the offline mode.

The decline in performance during the online mode can likely be attributed to the absence of an integrated real-world application for BCI use. Consequently, a longer recording duration requiring 30 seconds of sustained concentration, without the presence of inherent motivations or challenges typically associated with real-world applications, may have led to subject distraction. Moreover, since this stage followed the offline training, subject fatigue might have been a contributing factor.

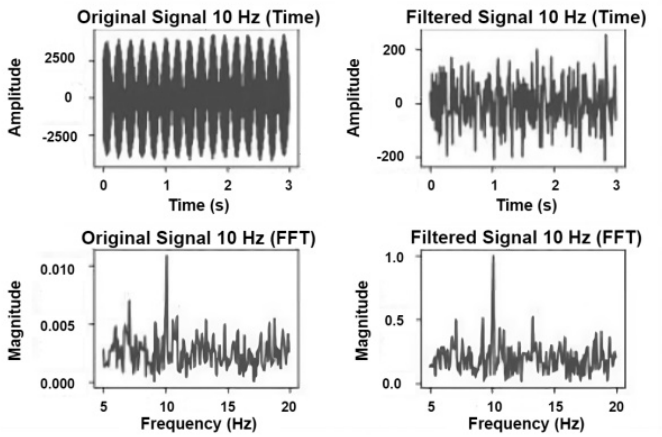

Fig. 8. EEG signal acquired through NeuroPi in the time and frequency domains. Signals acquired when the user was exposed to a stimulus at 10 Hz were recorded at Oz electrode. It is possible to observe in the frequency domain a higher amplitude at the specific frequency.

TABLE I
NeuroPi performance accuracy in offline and online modes using 8 electrodes for 4 frequencies: 6, 10, 12, and 15 Hz.

	Frequency (Hz)				
BCI	**6**	**10**	**12**	15	**Avg±Std**
Offline	78,75%	88,13%	60,63%	86,25%	78,44%±12,54
Online	60,00%	50,00%	60,00%	50,00%	55,00%±5,77

Currently, ongoing research is focused on improving the user interface to reduce fatigue resulting from stimuli and create a more user-friendly interaction. To encourage the widespread use of BCI systems, it is imperative to follow the recommendations established by the Human-Computer Interaction (HCI) field. Such compliance guarantees an enjoyable, secure, and effective user experience, particularly when these systems are applied in assistive technologies designed for users with physical limitations.

C. NeuroPi with 16 electrodes

Tab. II presents the performance of NeuroPi with 16 electrodes (O1, Oz, O2, PO3, POz, PO4, Pz, Cz, C3, CP1, P3, CP5, C4, CP6, P4, and CP2) in offline mode, both without and with feature selection. The BCI performance significantly improves when feature selection is applied, and a lower standard deviation is observed, indicating a more consistent system. In the best configuration, the following 12 channels were selected by the Pearson correlation coefficient method, in order of relevance: O2, PO4, O1, C4, C3, PO3, Cz, CP1, P4, CP2, CP6, and Oz.

V. Related Works

A BCI system platform should implement the steps of brain signal acquisition, pre-processing, feature extraction, selection, classification, and command output for the controlled application. Currently, various BCI system platforms are available,

TABLE II
IMPACT OF FEATURE SELECTION ON NEUROPI PERFORMANCE ACCURACY USING 16 ELECTRODES IN OFFLINE MODE FOR 4 FREQUENCIES: 6, 10, 12, AND 15 HZ.

	Frequency (Hz)				
Selection	**6**	**10**	**12**	**15**	**AVG±std**
None	25,00%	62,50%	62,50%	75,00%	56,25%±18,75
Pearson	75,00%	87,50%	75,00%	75,00%	78,13%±5,41

with notable options including BCI2000 [21], BioSig [22], BCI++ [23], OpenViBE [24] and PMW [25].

BCI2000 emerged in the early 2000s to address the demand for tools for BCI system development. Its primary aim was to provide a set of tools. Around the same time, BioSig was introduced as a tool for offline EEG data analysis using MATLAB and Octave. In 2004, it received an update to analyze online data, often referred to as real-time brain-computer interface (rtsBCI), in MATLAB/Simulink. Both BCI2000 and BioSig focus on supporting research in the BCI field.

BCI++ was developed with an emphasis on end-users and researchers. Its structure is designed to simplify interaction with external devices and includes a graphical user interface (GUI) known as AEnima. It was created in response to the lack of BCI platforms that emphasize Human-Computer Interface (HCI) aspects. Following a similar user-friendly approach through user interfaces, OpenViBE was launched in 2009, offering tools for BCI development and integration with virtual reality (VR) environments, primarily focused on research. By providing a graphical programming language within the platform, OpenViBE does not require programming knowledge to expand its functionalities.

With the growing interest in using Python in the neuroscience community, PMW was developed in this language, distinguishing itself from the earlier platforms predominantly written in C++. Due to Python's smoother learning curve compared to C++, it has become the preferred language in the neuroscience field. PMW was released in three releases between 2008 and 2015: The first release, named Pythonic feedback framework (Pyff), enabled the development of visual stimuli and feedback. The second release, called Mushu, handled the acquisition of EEG signals from multiple devices. Finally, the third release, named Wyrm, provided a toolkit for BCI system development.

Developing BCI is not just about advancing science and technology but also about enhancing people's everyday lives. While theoretical research is essential for an in-depth understanding of BCI systems, creating BCIs tailored for end-users is equally critical. These practical applications not only democratize access to innovative technologies but also have the potential to transform lives. In this regard, NeuroPi was developed to be an embedded SSVEP-based BCI, ready for use in controlling applications such as wheelchairs, prosthetics, and entertainment applications, among others. These requirements guided its development, making it portable, cost-effective, and easy to integrate into various applications.

VI. CONCLUSION

When we focus on creating BCIs for the general public, we are opening doors to a myriad of possibilities that can provide independence to subjects with severe motor disabilities, enabling them to control electronic devices such as wheelchairs or computers solely through their brain signals. Furthermore, they can revolutionize how we interact with technology, making interfaces more intuitive and accessible for everyone. This not only enhances people's quality of life but also establishes a positive cycle of learning, feedback, and continuous improvement.

For the popularization of BCI systems, it is essential to develop tools for both BCI research and application-oriented BCI systems to practically validate theoretical concepts. With current computational resources, BCI systems are approaching reality, and user-side testing is as crucial as the development of new techniques for acquiring, processing, and classifying brain signals.

In this research, we introduce NeuroPi, an embedded Brain-Computer Interface (BCI) solution, which is freely available for researchers and application developers [10]. Our presentation offers a comprehensive description of NeuroPi, outlining its features and capabilities for immediate use in research and application development. NeuroPi was designed with portability, customization, cost-efficiency, and ease of integration into several applications. Our goal is to promote the widespread adoption of BCI systems and contribute to an improved quality of life for people with disabilities.

For future work, it is suggested to incorporate new preprocessing, feature extraction, and classification techniques into NeuroPi, aiming to enhance its performance. Additionally, conducting tests involving high-frequency visual stimuli is essential to reduce visual fatigue in users and facilitate its use by individuals with a history of photoepilepsy.

REFERENCES

[1] J. R. Wolpaw, "Brain-computer interfaces (BCIs) for communication and control," in *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, 2007, pp. 1–2. DOI: 10.1145/1296843.1296845.

[2] B. Graimann, B. Allison, and G. Pfurtscheller, "Brain–computer interfaces: A gentle introduction," in *Brain-Computer Interfaces*, Springer, 2009, pp. 1–27. DOI: 10.1007/978-3-642-02091-9_1.

[3] S. K. Mudgal, S. K. Sharma, J. Chaturvedi, and A. Sharma, "Brain computer interface advancement in neurosciences: Applications and issues," *Interdisciplinary Neurosurgery*, vol. 20, p. 100 694, 2020. DOI: 10.1016/j.inat.2020.100694.

[4] M. Li, X. Chen, and H. Cui, "A high-frequency SSVEP-BCI system based on simultaneous modulation of luminance and motion using intermodulation frequencies," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2023. DOI: 10.1109/TNSRE.2023.3281416.

[5] G. Garcia-Molina and D. Zhu, "Optimal spatial filtering for the steady state visual evoked potential: BCI application," in *2011 5th International IEEE/EMBS Conference on Neural Engineering*, IEEE, 2011, pp. 156–160. DOI: 10.1109/NER.2011.5910512.

[6] B. Allison, T. Luth, D. Valbuena, A. Teymourian, I. Volosyak, and A. Graser, "Bci demographics: How many (and what kinds of) people can use an SSVEP BCI?" *IEEE transactions on neural systems and rehabilitation engineering*, vol. 18, no. 2, pp. 107–116, 2010. DOI: 10.1109/TNSRE.2009.2039495.

[7] X. Chai, Z. Zhang, K. Guan, T. Zhang, J. Xu, and H. Niu, "Effects of fatigue on steady state motion visual evoked potentials: Optimised stimulus parameters for a zoom motion-based brain-computer interface," *Computer Methods and Programs in Biomedicine*, vol. 196, p. 105 650, 2020. DOI: 10.1016/j.cmpb.2020.105650.

[8] S. N. Carvalho, T. B. Costa, L. F. Uribe, *et al.*, "Comparative analysis of strategies for feature extraction and classification in SSVEP BCIs," *Biomedical Signal Processing and Control*, vol. 21, pp. 34–42, 2015. DOI: 10.1016/j.bspc.2015.05.008.

[9] H. M. A. Leite, S. N. Carvalho, T. B. S. Costa, R. Attux, H. H. Hornung, and D. S. Arantes, "Analysis of user interaction with a brain-computer interface based on steady-state visually evoked potentials: Case study of a game," *Computational intelligence and neuroscience*, vol. 2018, 2018. DOI: 10.1155/2018/4920132.

[10] "NeuroPI." (Accessed on July 29, 2024), [Online]. Available: https://github.com/vitor-martinsb/NeuroPi.

[11] "Raspberry pi," Raspberry Pi Foundation. (Accessed on July 29, 2024), [Online]. Available: raspberrypi.org.

[12] "OpenBCI." (Accessed on July 29, 2024), [Online]. Available: openbci.com.

[13] "Ultracortex Mark IV." (Accessed on July 29, 2024), [Online]. Available: https://docs.openbci.com/AddOns/Headwear/MarkIV/.

[14] H. H. Jasper, "Ten-twenty electrode system of the international federation," *Electroencephalography Clin Neurophysiol*, vol. 10, pp. 371–375, 1958. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/10590970/.

[15] O. Bertrand, F. Perrin, and J. Pernier, "A theoretical justification of the average reference in topographic evoked potential studies," *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, vol. 62, no. 6, pp. 462–464, 1985. DOI: 10.1016/0168-5597(85)90058-9.

[16] D. J. McFarland, L. M. McCane, S. V. David, and J. R. Wolpaw, "Spatial filter selection for EEG-based communication," *Electroencephalography and clinical Neurophysiology*, vol. 103, no. 3, pp. 386–394, 1997. DOI: 10.1016/s0013-4694(97)00022-2.

[17] G. Pfurtscheller, B. Z. Allison, G. Bauernfeind, *et al.*, "The hybrid BCI," *Frontiers in neuroscience*, vol. 4, p. 1283, 2010. DOI: 10.3389/fnpro.2010.00003.

[18] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995, ISBN: 9780198538646.

[19] S. Theodoridis and K. Koutroumbas, *Pattern recognition*. Elsevier, 2006, ISBN: 1597492728.

[20] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998, ISBN: 0132733501.

[21] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "BCI2000: A general-purpose brain-computer interface (BCI) system," *IEEE Transactions on biomedical engineering*, vol. 51, no. 6, pp. 1034–1043, 2004. DOI: 10.1109/TBME.2004.827072.

[22] A. Schlögl and C. Brunner, "Biosig: A free and open source software library for BCI research," *Computer*, vol. 41, no. 10, pp. 44–50, 2008. DOI: 10.1109/MC.2008.407.

[23] P. Perego, L. Maggi, S. Parini, and G. Andreoni, "BCI++: A new framework for brain computer interface application," in *SEDE*, Citeseer, 2009, pp. 37–41. [Online]. Available: https://api.semanticscholar.org/CorpusID:30374734.

[24] Y. Renard, F. Lotte, G. Gibert, *et al.*, "Openvibe: An open-source software platform to design, test, and use brain–computer interfaces in real and virtual environments," *Presence*, vol. 19, no. 1, pp. 35–53, 2010. DOI: 10.1162/pres.19.1.35.

[25] B. Venthur, "Design and implementation of a brain computer interface system," Ph.D. dissertation, Technischen Universität Berlin, 2015.

VII. Biography Section

Vitor M. Barbosa holds a Master's degree in Electrical Engineering from the University of São Paulo, Brazil. He is currently employed at NTT Data. His primary research interests include signal processing, brain-computer interfaces, and machine learning.

Sarah N. Carvalho holds dual Bachelor of Science degrees in Electrical Engineering from Politecnico di Torino, Italy (2010) and the University of Campinas, Brazil (2011). She earned her Master's degree in 2012 and Ph.D. in 2016, both in Electrical Engineering from the University of Campinas, Brazil. Currently, she is an assistant professor at the Institute of Aeronautics, focusing her research on brain-computer interfaces, biomedical signal processing, telecommunication systems, and machine learning.

Harlei M. A. Leite obtained his Ph.D. in Electrical Engineering from the University of Campinas, Brazil, in 2018. He is currently working as an assistant professor at the Institute of Aeronautics. His research interests encompass brain-computer interfaces, human-computer interfaces, machine learning, and signal processing.