# SSVEP-Based Brain-Computer Interface for Cursor Control

A User-Focused Graphical Interface for Real-Time SSVEP Cursor Control

EE3L11: Bachelor Graduation Project Electrical Engineering 2024/25

Abdoulaye Demba Jalloh & Karim Tageldin

Delft University of Technology

# SSVEP-Based Brain-Computer Interface for Cursor Control

## A User-Focused Graphical Interface for Real-Time SSVEP Cursor Control

by


Abdoulaye Demba Jalloh & Karim Tageldin

| Student Name | Student Number |
|---|---|
| Abdoulaye Demba Jalloh | 5727553 |
| Karim Tageldin | 5737397 |


| | |
|---|---|
| Supervisors: | Tiago Costa and Dante Muratore |
| Project Duration: | April, 2025 - June, 2025 |
| Faculty: | Faculty Electrical Engineering, Mathematics and Computer Science, Delft |

TU Delft

# Abstract

This thesis presents the design and implementation of a graphical user interface (GUI) for a real-time brain-computer interface (BCI) system based on steady-state visual evoked potentials (SSVEP). Developed as part of a larger collaborative project alongside Data Acquisition and Signal Processing subgroups, the GUI enables users to control a computer cursor using brain activity. Core functionalities include real-time EEG visualization, configurable data recording trials, visual stimulus presentation, and a cursor control interface that provides feedback based on live classification results. The system architecture allows the GUI to connect to the EEG headset via the Lab Streaming Layer (LSL) protocol, stream EEG data to the back-end pipeline, and translate classification output into interactive visual feedback. The integrated system successfully demonstrated real-time SSVEP-based control, validating both functional and technical requirements of the GUI subsystem. This work contributes to the development of more intuitive and accessible GUI designs for future BCI applications.

# Preface

This paper marks the completion of our Bachelor's graduation project, conducted as part of a collaborative effort to develop a real-time Brain-Computer Interface (BCI) system. The project was carried out by three dedicated groups: Graphical User Interface (GUI), Signal Acquisition, and Signal Processing. Our group was responsible for the GUI, focusing on developing an interactive front-end that supports real-time cursor control through brain activity.

Working on this project has been both challenging and rewarding. It provided us with the opportunity to apply and expand our knowledge in designing and building a GUI, while also learning how to collaborate effectively within a multidisciplinary team. The experience gave us valuable insights into the complexity and potential of BCI technologies.

We would like to sincerely thank our supervisors, Tiago Costa and Dante Muratore, for their ongoing support, guidance, and constructive feedback throughout the course of the project. Their expertise and encouragement played a crucial role in helping us stay motivated and on track.

We hope that this work will contribute to further developments in the field of BCI and are excited to see how future students and researchers will build on it.

*Abdoulaye Demba Jalloh & Karim Tageldin*
*Delft, June 2025*

# Contents

# 1
# Introduction

## 1.1 State-of-the-Art Analysis

Brain-Computer Interfaces (BCIs) are systems that enable direct communication between the brain and an external device by translating neural activity into actionable commands, bypassing peripheral nerves or muscular control [1]. Non-invasive BCI systems, particularly those based on electroencephalography (EEG), are very attractive due to their safety, portability, and relatively low cost [2]. BCIs have shown promise in restoring communication or control abilities to individuals with severe motor impairments. However, in recent years, BCI applications have expanded into broader domains, including entertainment, neurofeedback training, and assistive technologies for the general population [3], [4].

Among various BCI paradigms, the Steady-State Visual Evoked Potential (SSVEP) approach stands out for its robustness, minimal training requirements, and high information transfer rates. SSVEPs are brain responses elicited when a user focuses on a visual stimulus flickering at a constant frequency, making this paradigm particularly suitable for real-time applications [5]. Thanks to advances in signal processing and machine learning, modern SSVEP-based BCIs now achieve significantly improved accuracy and responsiveness, bringing real-time control applications closer to practical deployment [6]. However, achieving seamless integration between signal acquisition, real-time processing, and graphical interface design remains a key technical challenge [7].

## 1.2 Problem Definition

Although significant progress has been made in BCI signal processing and classification, many systems still lack intuitive and user-friendly graphical interfaces that connect neural input with effective system control [8], [9]. This gap is especially critical in real-time applications, where users must receive immediate and clear feedback to interact reliably with the system. Without it, control becomes unintuitive and prone to user error. Moreover, existing interfaces often overlook human-centered design principles, leading to visual fatigue and increased cognitive load, particularly during tasks requiring sustained attention to visual stimuli, such as SSVEP-based systems [10]. These issues reduce user engagement and system effectiveness.

The goal of this thesis is to bridge this gap by designing and implementing a GUI subsystem that enables reliable, real-time SSVEP-based cursor control. The interface must present visual stimuli while minimizing fatigue, handle live EEG data, communicate with the classification back-end, and deliver immediate and intuitive feedback, serving as both control environment and feedback platform.

The scope of the project is bounded by practical and academic constraints, including limited development time, the need to ensure compatibility with the Unicorn Hybrid Black EEG headset – which has a limited number of channels and restricted access to proprietary APIs – as well as ongoing back-end changes from collaborating subgroups, all of which influenced system design and integration decisions.

## 1.3 Thesis Synopsis

The thesis is structured as follows. Chapter 2 introduces a high-level system overview, describing the overall architecture and the role of the GUI. Chapter 3 presents the program of requirements for this project. Chapter 4 explores the literature foundations that informed design choices. Chapter 5 details the design and implementation of each GUI module. Chapter 6 describes how the GUI was integrated with the data acquisition and signal processing subsystems and evaluates the real-time cursor control. Chapter 7 presents the evaluation results, reflects on requirement fulfilment and suggests future improvements. Finally, Chapter 8 concludes the thesis with key takeaways.

# 2

# System Overview

This section presents a high-level overview of the SSVEP-based BCI system developed in this project. It describes the overall system architecture, including its core subsystems and their interactions, and clarifies the role of the Graphical User Interface (GUI) within the larger system. The objective is to establish a clear conceptual understanding of the system's structure and data flow before addressing implementation details and integration strategies in later chapters.

## 2.1 Overall System Architecture

The BCI system is composed of three main subsystems: Graphical User Interface (GUI), Data Acquisition, and Signal Processing. These components form an integrated loop that enables both various EEG-driven functionalities and real-time cursor control.

As illustrated in Figure 2.1, the pipeline begins with the GUI, which initiates a connection to the EEG headset using the Lab Streaming Layer (LSL) protocol. LSL is an open-source framework designed for real-time collection, transmission, and synchronization of EEG data from devices such as the Unicorn Hybrid Black headset and other biosignal devices. Once the stream is active, the headset continuously transmits raw EEG data. This data is utilized by the GUI in multiple contexts, including live EEG visualization and configurable recording trials for data collection.

In parallel, the streamed EEG data is forwarded to the back-end pipeline responsible for real-time control. This begins with the Data Acquisition subsystem, which applies pre-processing techniques including filtering and artifact removal. The cleaned EEG data is then passed to the Signal Processing subsystem, where classification is performed to determine the user's intended action by detecting SSVEP responses associated with specific visual stimuli. The resulting classification output, consisting of the predicted frequency and a confidence score, is returned to the GUI, which translates this output into interactive feedback, including cursor movement and on-screen visual indicators.

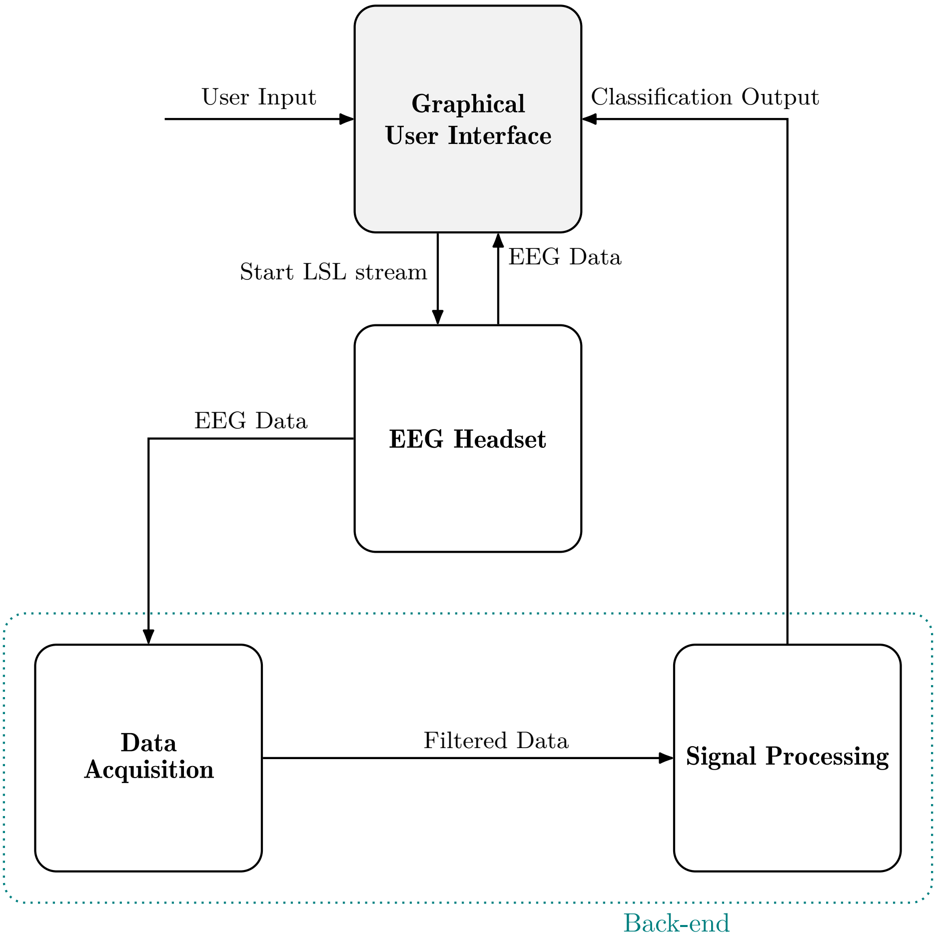



**Figure 2.1:** Overall System Architecture

## 2.2 Role of the GUI Subsystem

Within the system architecture, the GUI plays a central and dual role: it serves as both the entry point for initiating EEG data flow and the interface through which feedback is presented to the user.

The GUI's responsibilities include:

- **Stream Management:** Establishing the LSL connection with the EEG headset and managing live data flow.
- **Live EEG Visualization and Data collection:** Displaying real-time EEG signals to help assess signal quality and launching configurable EEG recording trials for offline analysis.
- **Visual Stimuli Presentation:** Rendering flickering targets to evoke SSVEP responses from the user.
- **Back-end Coordination:** Extracting EEG segments at regular intervals and sending them to the back-end pipeline for real-time classification.
- **Feedback Translation:** Receiving classification results and converting them into responsive visual feedback such as cursor movements and prediction indicators.

Further details of the implementation of the GUI modules are presented in chapter 5, while the integration process and the real-time interaction with the back-end are discussed in chapter 6.

# 3

# Programme of Requirements

This chapter outlines the system requirements, spanning high-level goals to GUI-specific specifications.

## 3.1 General Requirements

The following general requirements apply to the overall BCI system and define its intended context of use:

- **G1.** The BCI system must allow users to control a cursor using SSVEP-based EEG signals.
- **G2.** The system must operate in real time, with an end-to-end latency (stimulus to feedback) ideally below 1 second.
- **G3.** The BCI must support at least two distinct flicker targets for multi-class classification.
- **G4.** The system must be intuitive and accessible for first-time users without prior BCI experience.

## 3.2 GUI Subgroup Requirements

The following requirements are specific to the GUI subsystem and address its functional scope, performance expectations, and integration constraints.

### 3.2.1 Functional Requirements

- **F1.** The GUI must render flickering visual stimuli at predefined frequencies to evoke SSVEP responses.
- **F2.** The GUI must support configurable EEG recording trials for offline analysis and data collection.
- **F3.** The GUI must allow users to smoothly navigate through modules via intuitive control buttons.
- **F4.** The GUI must support live EEG visualization across all channels to help assess signal quality.
- **F5.** The GUI must provide real-time feedback upon classification, including visual indicators.
- **F6.** The interface must function independently of other subsystems to enable standalone validation and testing using mock data or simulated streams.

### 3.2.2 Performance Requirements

- **P1.** Flickering stimuli must be rendered with frame-precise timing based on the screen refresh rate.
- **P2.** The GUI must run smoothly on standard consumer-grade hardware without noticeable performance degradation.
- **P3.** The real-time EEG plot must have a latency of less than 1 second.

- **P4.** Cursor control classification must be triggered at fixed intervals, ensuring the GUI remains responsive during processing.

### 3.2.3 Implementation and Integration Requirements

- **I1.** The GUI must be compatible with the Unicorn Hybrid Black EEG headset using LSL for data streaming.
- **I2.** The GUI must be integrated with the data acquisition and signal processing subsystems.
- **I3.** The GUI must trigger EEG data acquisition and pass EEG segments to the back-end pipeline for real-time classification.
- **I4.** The GUI must handle classification output and convert it into real-time visual feedback.

# 4

# Literature Foundations for GUI Design

This chapter presents only the essential background needed to understand and justify the GUI design choices made in this project. Rather than offering a broad or general literature review, it focuses on design-relevant insights directly related to SSVEP-based BCIs. The chapter begins by briefly comparing common BCI paradigms and explaining the rationale for selecting SSVEP. It then explores key interface design considerations and concludes by linking specific design decisions in this project to key findings from the literature.

## 4.1 BCI Paradigms and Motivation for SSVEP

Noninvasive EEG-based BCIs use different paradigms to interpret user intent. The most common ones include motor imagery (MI), P300 event-related potentials, and steady-state visual evoked potentials (SSVEPs). Each has unique strengths and limitations. MI requires users to imagine specific body movements, often needing extensive training and offering slower feedback. P300 paradigms rely on detecting responses to rare “oddball” stimuli, offering moderate performance, but their cognitive demands make them less practical for fast interaction [11].

SSVEP-based systems, on the other hand, are driven by visual stimuli flickering at distinct frequencies. When a user gazes at a flickering target, the corresponding frequency becomes prominent in their EEG signal, enabling target identification. This paradigm is known for its high information transfer rates (ITRs), intuitive interaction, and minimal user training requirements. For example, early systems enabled real-time 2D cursor control by assigning directional movement to distinct flickering targets [12]. More recent designs using up to 40 visual targets have achieved ITRs of 267 bits/min, demonstrating strong performance potential [13].

Importantly, SSVEP offers a favorable balance between speed, accuracy, and ease of use in real-time applications. These characteristics make it particularly suited for the objectives of this project. Although it requires thoughtful visual design to manage potential fatigue, SSVEP remains a strong fit for developing intuitive and practical BCI interfaces [14], [15].

## 4.2 Interface Design and User-Centered Considerations

Designing an effective SSVEP-based BCI interface requires balancing technical performance with user comfort and clarity. This section discusses key principles that shaped our GUI, including flicker frequency selection, visual layout, and feedback mechanisms. All these factors influence how intuitive and sustainable the interface is during real-time use.

### 4.2.1 Visual Stimulus Design

SSVEP systems rely on flickering visual targets to trigger distinct EEG responses. Choosing the right flicker frequencies is important. Technically, the display's refresh rate limits which frequencies can be shown smoothly. For instance, a 60 Hz screen reliably supports frequencies like 10 Hz or 15 Hz, which are clean divisors [16].

From a perceptual standpoint, very low frequencies ($< 6$ Hz) tend to produce intense flickering that is uncomfortable for users, while very high frequencies ($> 30$ Hz) often evoke weaker EEG responses. Studies have shown that the 8–18 Hz range provides the best compromise between signal strength and visual comfort [17].

Spacing between targets also matters. If visual elements are placed too close together, the brain signals they trigger can interfere with each other. Moderate spacing, around a few degrees of visual angle, helps prevent this and keeps the interface easy to use.

### 4.2.2 Visual Comfort and Cognitive Load

Staring at flickering targets for too long can be tiring. To reduce this strain, designers often use visual patterns that are easier on the eyes, such as QR textures or checkerboard layouts [15]. These patterns keep performance high while improving comfort.

Cognitive load is also an issue. When the interface is too crowded or too flashy, it becomes harder to focus. Using simple icons, consistent layouts, and only showing what's necessary helps users stay focused and reduces mental effort [18]. A clean and predictable design goes a long way in making the system easier to use.

### 4.2.3 Feedback Methods in SSVEP Interfaces

Effective feedback is vital in SSVEP interfaces to guide user attention and confirm system responses, especially since control relies on gaze without physical input. One common approach is to apply external modifications to the selected target, such as adding a border or overlaying a highlight. In cursor-based applications, feedback is often presented through a moving cursor, offering continuous confirmation of directional output and helping users stay oriented during control tasks [18].

Alternatively, rather than adding external feedback elements, some systems integrate feedback directly into the stimulus itself by modifying its contrast, brightness, or size in real time. This method reinforces user selections without crowding the interface with extra elements. Subtle changes like these have been shown to improve the user experience while keeping the interface clear and focused [19]. The key is to find a balance between clarity and comfort, ensuring the feedback supports the user without increasing visual or mental fatigue.

## 4.3 Design Insights Applied in This Work

The literature review provided several practical insights that directly influenced the design of the GUI system. One of the key takeaways was the importance of selecting appropriate flicker frequencies. A range between 8–18 Hz was chosen, as this offered a strong balance between signal clarity and user comfort. Frequencies were selected to align with the screen refresh rate and to avoid harmonics or overlaps that could reduce classification performance.

Equally important was the visual layout. Rather than placing targets too close together, sufficient spacing was ensured to reduce neural interference and visual fatigue. A clean, grid-based design was adopted to make it easy for users to locate and focus on individual targets without feeling overwhelmed.

In terms of feedback, simple but effective visual cues such as a moving cursor and subtle highlighting of selected targets were implemented. To support long-term usability, interface contrast was kept moderate, consistent target sizes were used, and a minimalistic design was prioritized to reduce distractions and help users stay focused during control tasks.

# 5

# GUI Development and Implementation

This chapter presents how the Graphical User Interface (GUI) was designed and implemented to support real-time interaction in the SSVEP-based BCI system. The focus lies on practical aspects that directly influenced development, such as the measurement context in which the GUI operates, key technical decisions around framework and architecture, and a detailed walkthrough of the implemented interface modules. While previous chapters discussed high-level design considerations and system architecture, this chapter provides a closer look at how user-facing components were translated into a working application.

## 5.1 Measurement Setup

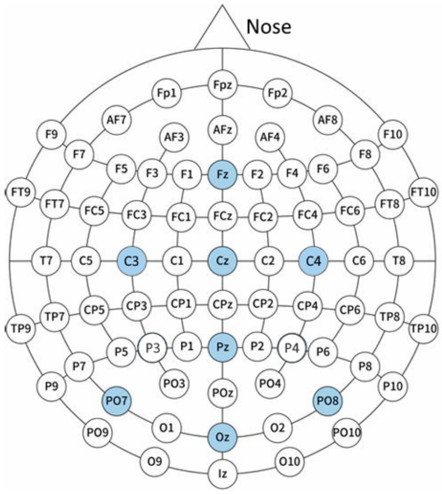
To understand how the Graphical User Interface (GUI) integrates into the broader BCI workflow, it is important to first describe the typical measurement environment in which it is used. Although EEG data acquisition is handled by a separate subsystem, the GUI was developed and tested in close coordination with GUI-driven EEG sessions. This subsection outlines the physical setup and conditions under which participants interacted with the system.

The headset used throughout this project is the g.tec Unicorn Hybrid Black (see Figure 5.1a), a dry-electrode, wireless EEG system offering 8 channels with a sampling frequency of 250 Hz. Its dry electrodes are convenient for rapid setup and eliminate the discomfort associated with gel-based systems, making it ideal for non-clinical testing environments. During sessions, the headset is positioned according to the international 10-20 system [20], with electrodes placed primarily over the occipital and parietal regions to target areas sensitive to SSVEP signals (see Figure 5.1b). Research indicates that the strongest SSVEP responses are typically recorded at electrodes PO7, Oz, and PO8, making them particularly reliable for detecting visual-evoked signals [21].

During each session, participants were seated comfortably at a distance of approximately 60–70 cm from a computer screen displaying the visual interface. To minimize movement artifacts and enhance signal quality, users were instructed to remain as still as possible, maintain a relaxed posture, and avoid sudden head or eye movements. Sessions were conducted in a quiet, distraction-free environment with dim lighting to reduce visual strain and enhance focus on the flickering targets.

By standardizing the recording conditions in this way, the team ensured consistency across trials and provided a realistic context for testing the GUI under conditions that reflect its intended real-time use. For more detailed information about the EEG acquisition procedures and signal recording setup, the reader is referred to the work of the Data Acquisition subgroup [22].

(a) Unicorn Hybrid Black EEG headset[23]




(b) Electrode placement (indicated in blue). Modified image from[20]

**Figure 5.1:** Figures of EEG headset and electrode placement

## 5.2 Design Choices and Justifications

The graphical user interface (GUI) for this EEG-based brain-computer interface (BCI) was developed with a strong focus on ease of use, responsiveness, real-time EEG visualization, and seamless integration with a Python back-end. Key frameworks—PyQt, PyQtGraph, and Lab Streaming Layer (LSL)—were selected for their ability to deliver an intuitive, robust, and efficient GUI:

- **PyQt**: Primarily motivated by its seamless integration with Python, PyQt simplifies the connection with back-end processing algorithms. Unlike JavaScript-based alternatives, which require WebSockets or other bridging methods, PyQt directly interfaces with Python, eliminating integration barriers and streamlining the development process. PyQt excels in rapid prototyping, facilitating quicker iterations and faster development cycles, while maintaining smooth performance crucial for responsive user interactions.
- **PyQtGraph**: Selected as the logical extension following the choice of PyQt. PyQtGraph's native compatibility with PyQt and Python reduces integration complexity and potential compatibility issues. It consistently demonstrated superior performance in real-time EEG visualization compared to alternatives like Matplotlib or JavaScript plotting tools, with significantly lower latency and higher frame rates essential for smooth and responsive EEG signal visualization.
- **Lab Streaming Layer (LSL)**: Chosen for EEG data streaming due to its low-latency, synchronized data transmission capabilities, critical for accurate real-time feedback. Additionally, LSL was freely available within the Unicorn suite software, offering a cost-effective solution for the project.

While JavaScript and WebSockets offer flexibility and rapid deployment capabilities, their higher initial setup complexity compared to the selected Python-based tools does not outweigh the straightforward integration, superior real-time performance, and enhanced prototyping capabilities provided by PyQt, PyQtGraph, and LSL.

## 5.3 GUI Modules and Functional Implementation

The Graphical User Interface (GUI) consists of four interactive modules that structure the user experience across different stages of the BCI workflow: Main Menu, Training, Calibration, and Cursor Control. Each module is implemented using a modular architecture supported by four internal components: the Data Handler for managing EEG input and recording, the Visualization Engine for rendering real-time plots and visual feedback, the Interface Controller for controlling timing and logic flow, and the User Interaction Layer for handling visual elements and user interaction.

The following subsections describe each module in terms of its core functionality, visual design, and conceptual implementation, highlighting how the internal components support their operation without going into detailed code.

### 5.3.1 Main Menu

The main menu serves as the entry point of the system, where users are prompted to select their name to begin a new session (see Figure 5.2). The interface is minimal by design, presenting a short welcome message and two clearly labeled buttons for user selection. This focused layout supports ease of use and sets a consistent starting point for all interactions.

After a user is selected, the system checks whether the EEG headset is properly connected and streaming via the Lab Streaming Layer (LSL). This validation step is handled by the Interface Controller and ensures that the user cannot proceed unless the device is active. A status message is shown during the check. If the stream is found, the user receives a confirmation and advances to the next screen. If not, an error message prompts the user to resolve the connection before continuing. This step adds robustness by preventing access to modules that rely on live EEG data.

Upon successful connection, the user is directed to the Mindstream Hub (see Figure 5.3), a central screen that presents the three core modules: Training, Calibration, and Cursor Control. Each module is displayed as a card with a short description, allowing the user to select their next step with clarity. This hub-based design offers a smooth transition into the BCI workflow and presents a clear overview of available functions.

User interactions and transitions are managed by the User Interaction Layer, while the Interface Controller oversees logic flow and connection handling. Together, these components ensure a smooth and reliable start to the user experience.

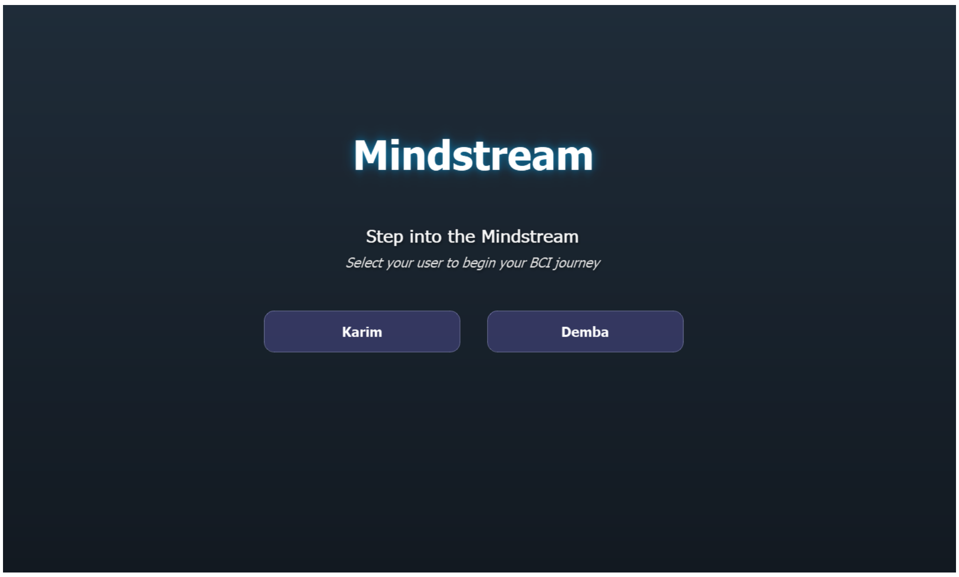



**Figure 5.2:** Main menu with user selection. Names shown are for demonstration purposes

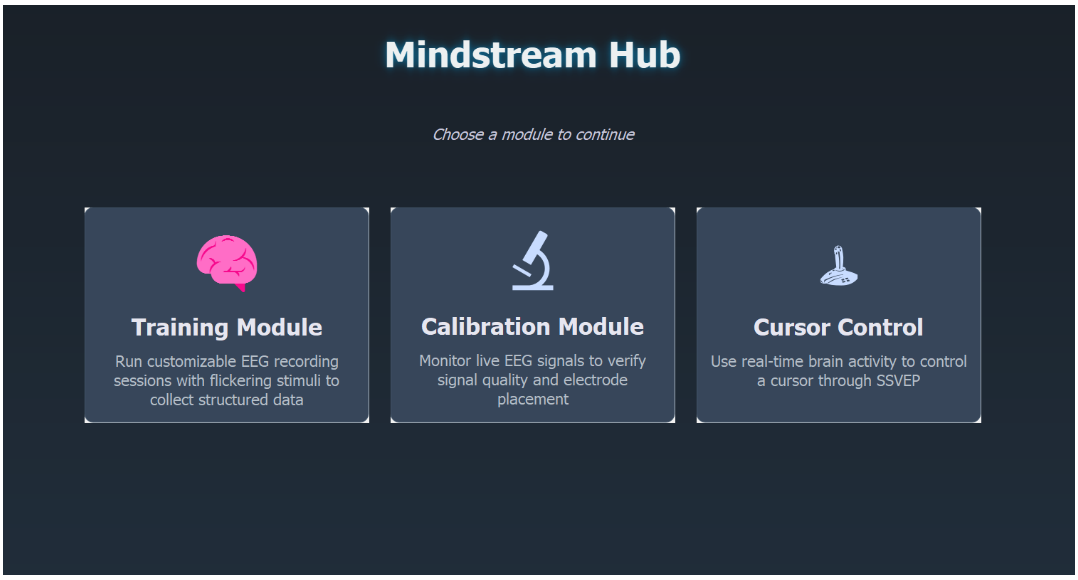



**Figure 5.3:** Mindstream Hub for navigating between core modules

### 5.3.2 Training Module

The Training Module serves as the foundation for structured EEG data collection within the GUI. Its primary purpose is to enable consistent signal recording during controlled visual stimulation. Although the project did not implement user-specific model training, this module is flexible enough to accommodate such functionality in future iterations. Throughout development, it was actively used by the Data Acquisition and Signal Processing teams to collect consistent datasets for testing and validating their subsystems.

After entering the module, users begin by launching a configurable trial setup (see Figure 5.4). The scrollable interface allows full control over the trial structure, including global settings (baseline duration and flicker color) and up to five independent stimulus cycles. Each cycle can be toggled on or off and includes parameters such as flicker duration, up to three simultaneous frequencies, the number of repetitions, and rest durations between flickers. When multiple cycles are enabled, additional rest intervals between them can also be defined.

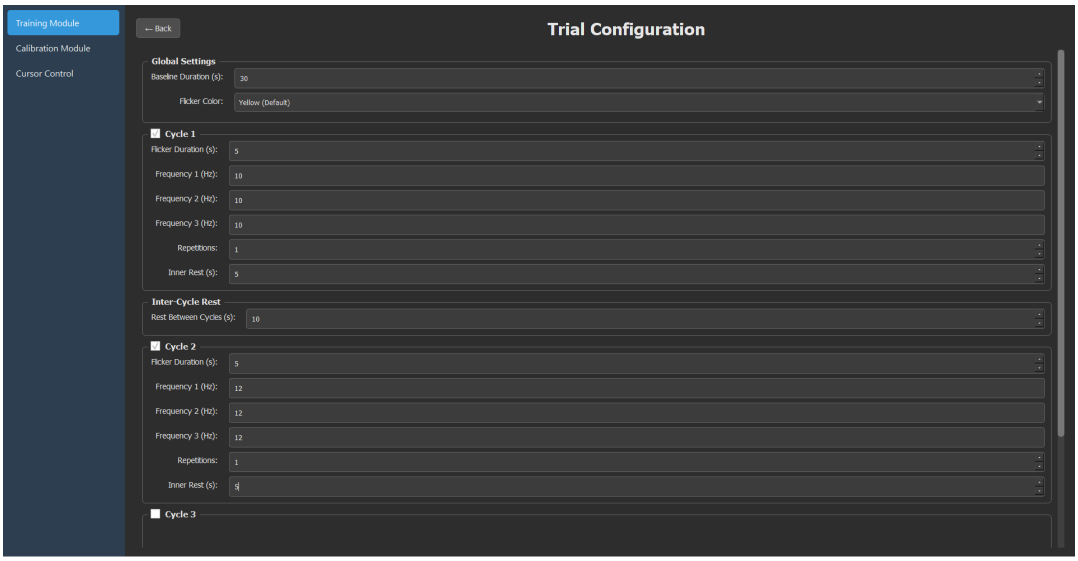



**Figure 5.4:** Training Module configuration screen

Once a trial begins and the EEG stream is detected, the interface transitions to an active trial view (see Figure 5.5). During visual stimulation, the screen displays the configured flicker color and provides real-time textual feedback about the current trial phase, frequency, and remaining time. The example shown in Figure 5.5 captures the flickering screen as seen during a stimulation period.

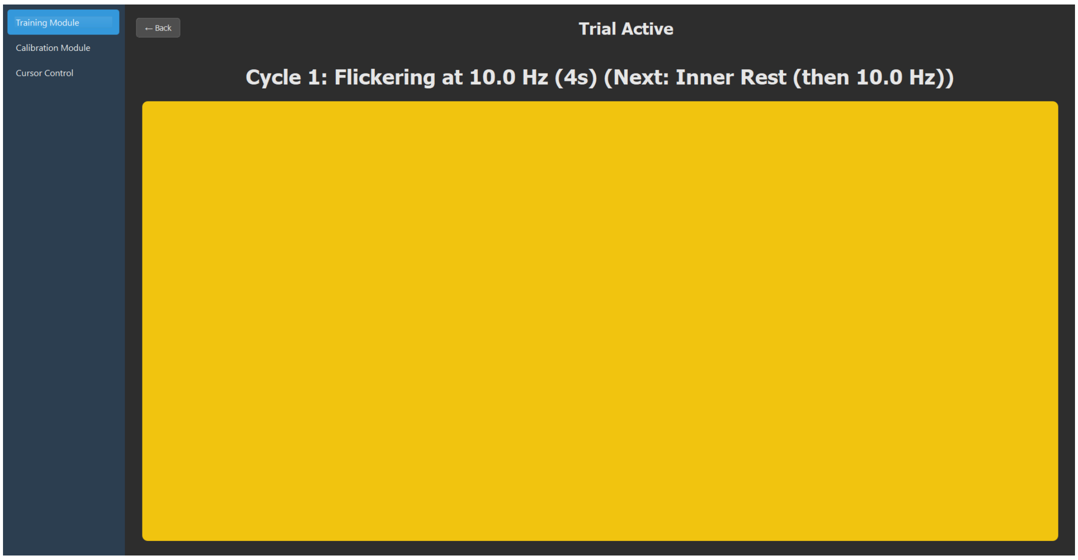



**Figure 5.5:** Training Module during an active trial

This module integrates all four internal GUI components. The User Interaction Layer manages interface logic, trial configuration inputs and screen transitions. The Interface Controller manages the sequence and timing of trial events such as baseline, flicker, and rest periods. The Data Handler connects to the EEG stream via LSL, records data in real time, and saves them to a CSV file that can be used by other subgroups for further processing and analysis. The Visualization Engine renders flicker stimuli and displays dynamic status updates, providing continuous visual feedback throughout the session. Together, these elements enable the Training Module to support reliable and configurable EEG trials while maintaining a responsive and user-friendly interface.

### 5.3.3 Calibration Module

The Calibration Module enables real-time visualization of EEG signals across all eight channels, allowing users to assess overall signal quality and verify electrode placement prior to data recording or control tasks. Upon activation, the module attempts to connect to the headset's EEG stream using the Lab Streaming Layer (LSL). If successful, it initiates continuous plotting of incoming signals using a 5-second rolling time window, as shown in Figure 5.6. Each channel is rendered in a dedicated subplot with color-coded traces, auto-scaling, labeled axes, and synchronized time alignment for clarity. These plots are arranged in a 2-column-by-4-row grid layout to optimize space usage and improve readability. In addition, users can click on an individual channel plot to zoom in for a detailed inspection, with the option to revert to the full grid view. The plot refreshes every 50 milliseconds, providing low-latency feedback that enables users to observe signal behavior in near real time without compromising performance.

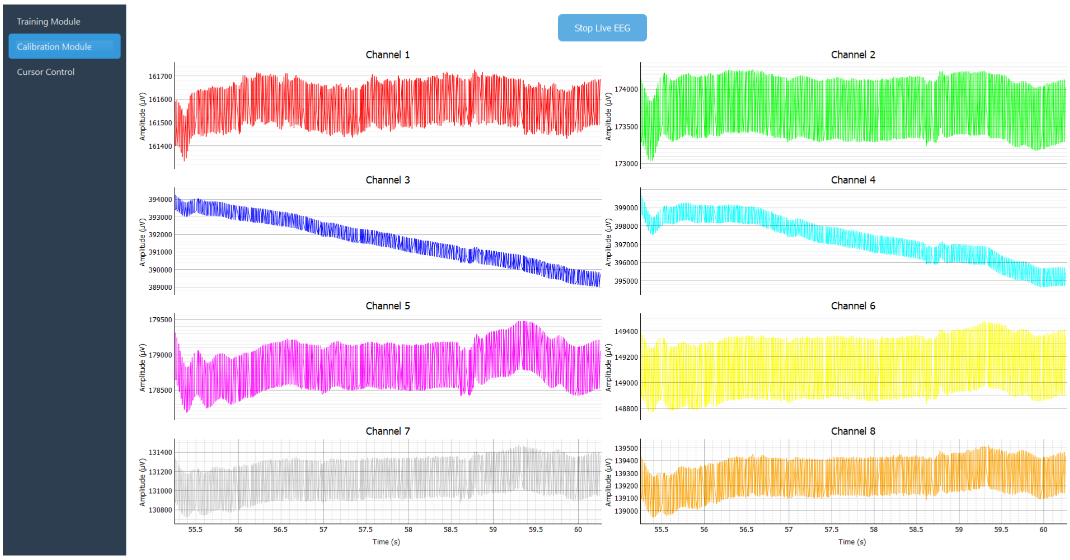



**Figure 5.6:** Live EEG visualization interface

Unlike the official Unicorn software, this module does not include real-time indicators of signal stability or electrode contact quality. These features depend on internal device diagnostics accessed via the proprietary Unicorn Python API, which was not available for this project. As a result, replicating hardware-level metrics such as impedance or contact status was not feasible, as this would require complex signal inference and would still lack accuracy. Despite this limitation, visual inspection proved to be a practical alternative. Through consistent use of the Unicorn software during earlier development phases, the team became familiar with the typical visual characteristics of clean versus problematic EEG signals. This enabled visual inspection to serve as a reliable indicator for verifying signal quality, particularly to identify flat-lined channels, sudden artifacts, or unusual fluctuations.

From an implementation perspective, the Visualization Engine handles dynamic rendering using PyQt-Graph, while the Data Handler manages LSL-based data streaming and short-term buffering for each channel. The Interface Controller manages timing and refresh cycles through a lightweight update loop, and the User Interaction Layer offers a simple toggle to start or stop the live view.

### 5.3.4 Cursor Control

The Cursor Control Module provides an interactive interface for real-time cursor movement based on SSVEP classification. Its layout consists of a central circular cursor and two peripheral flickering blocks, positioned on the left and right sides of the screen (see Figure 5.7). Each block flickers at a distinct frequency (8.57 Hz and 12 Hz), selected to evoke frequency-specific brain responses corresponding to left and right control commands, respectively. These frequencies were selected based on offline experiments conducted with the Signal Processing subgroup using the Training Module, where they consistently yielded the highest classification accuracy. For a complete explanation and evaluation of these results, the reader is referred to their thesis [24].

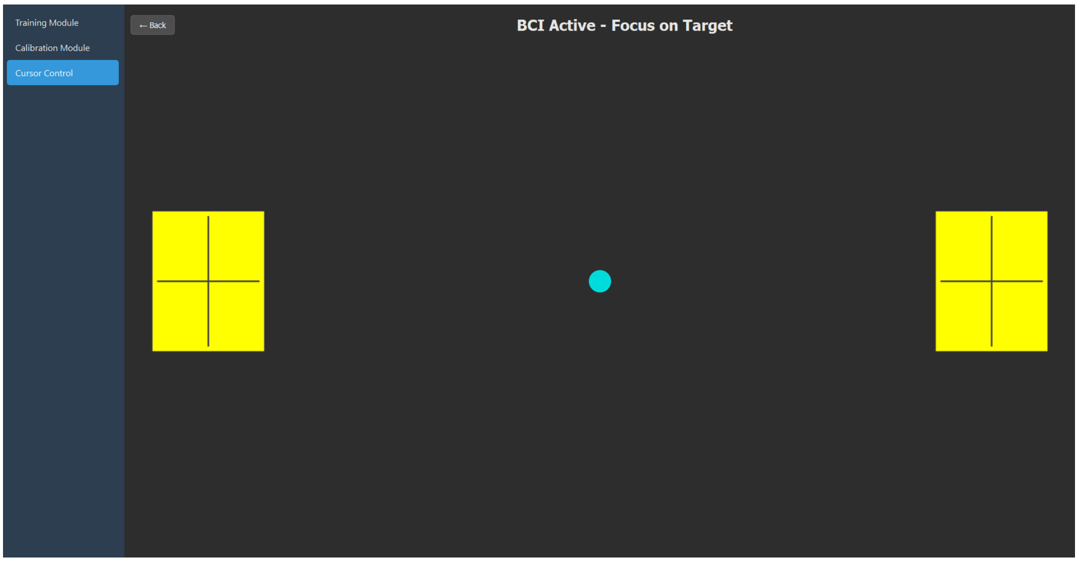



**Figure 5.7:** SSVEP-based Cursor Control Module interface

To guide user attention and reduce visual fatigue, each flickering block includes a centered plus-shaped crosshair. As discussed in Section 4.2.2, such fixation markers support gaze stability and reduce signal noise by helping users maintain consistent visual focus on the targets.

Initially, the module was tested offline using keyboard inputs (A/D keys) to simulate classification outcomes. This approach allowed for validation of the layout, flicker logic, and cursor movement independently of live EEG input. In the intended final implementation, the module connects to the headset's EEG stream via the Lab Streaming Layer (LSL). Once connected, classification runs periodically on recent EEG samples, and if a predicted frequency matches a target and exceeds a confidence threshold, the cursor is moved left or right accordingly. The transition to real-time functionality, including how the cursor control interface was further developed and integrated with other subgroups to support live BCI interaction, is discussed in more detail in Chapter 6.

The module integrates all four internal components. The User Interaction Layer manages input handling, button controls and feedback labels. The Interface Controller handles timing, flickering logic and cursor positioning. The Visualization Engine renders all interface elements using a custom paintEvent method, while the Data Handler retrieves real-time EEG data and communicates with the classification pipeline. Together, these components enable responsive and intuitive control based on SSVEP input.

# 6
# System Integration and Evaluation

This chapter outlines how the separate subsystem components—Data Acquisition, Signal Processing, and Graphical User Interface (GUI)—were integrated into a functioning real-time SSVEP-based BCI system. The goal was to transition from an offline pipeline—where EEG data was recorded, pre-processed, and classified in separate stages— to a real-time loop in which EEG data is continuously streamed, processed, and used for immediate control actions through the Cursor Control Module. To achieve this, careful coordination was required across teams and modules. The subsections that follow outline the overall integration strategy, how the GUI was integrated with the other subsystems and an evaluation of the complete real-time system.

## 6.1 Integration Strategy and Subsystem Coordination

The transition from an offline processing workflow to a real-time BCI system required careful coordination between the three subgroups. Initially, the system operated in a sequential, file-based fashion. EEG data was recorded through the GUI's Training Module and exported as CSV files. These were handed over to the Data Acquisition subgroup for filtering and artifact removal. The cleaned signals were then passed to the Signal Processing subgroup for classification. While this pipeline enabled early validation of system components, it did not support live interaction.

To transition to real-time functionality, a staged integration approach was adopted to manage complexity and isolate potential issues. This involved a bottom-up approach where the Data Acquisition and Signal Processing subgroups first integrated their components. This allowed them to independently test filtering and classification algorithms using previously recorded EEG data, ensuring functional correctness and compatibility before GUI was involved. Once this back-end was stable, the GUI subgroup proceeded to integrate the cursor control module with this shared pipeline.

Subsystem coordination was facilitated through shared interface definitions, collaborative discussions, and incremental testing. The teams also made use of shared repositories and collaborative tools to align on implementation details and troubleshoot integration issues.

## 6.2 Integration with the Back-end Pipeline

To enable real-time BCI functionality, the Graphical User Interface (GUI) was integrated with a shared back-end pipeline developed by the Data Acquisition and Signal Processing subgroups. This pipeline performs real-time EEG data retrieval, pre-processing, and classification, and communicates the results back to the GUI for direct user feedback and control actions.

Once the Cursor Control Module is activated, the GUI connects to the headset's EEG stream using Lab Streaming Layer (LSL) and continuously buffers incoming data. Every second, a 1-second segment of recent EEG samples is retrieved from the internal buffer and passed to the back-end through the `process_live_eeg_segment` function. This function serves as the main classification entry point: it first applies pre-processing (bandpass filtering and notch filtering) to clean the data, then forwards the processed segment to the classification algorithm implemented by the Signal Processing subgroup. Only the most informative occipital and parietal channels—PO7, Oz, and PO8—are used, based on their strong SSVEP response profile as discussed in section 5.1. The classification is triggered by a QTimer (a PyQt component for scheduling repeated tasks) that executes every 1 s, forming a rolling control loop with manageable computational load and consistent responsiveness.

The choice of a 1-second classification window reflects a trade-off between responsiveness and classification reliability. This duration has been widely adopted in SSVEP-based BCIs as it provides sufficient data for accurate frequency detection while maintaining acceptable latency for real-time interaction [25].

Once classification is complete, the function returns three outputs: the predicted frequency, its index relative to the configured stimulus frequencies, and a confidence score. This output is passed back to the GUI and interpreted by the cursor control logic. If the confidence score exceeds a predefined threshold of 0.2— determined based on classification performance insights from the Signal Processing subgroup—the GUI maps the predicted frequency to one of the target commands: 8.57 Hz for left movement, or 12 Hz for right movement. This mapping triggers several types of real-time feedback: the cursor's position is updated, the action ("BCI: Moving Left/Right") is shown as a message on screen, and the prediction details (frequency and confidence) are displayed numerically above the interface (see Figure 6.1). Further details on how the confidence score is computed and evaluated are beyond the scope of this section and can be found in the thesis of the Signal Processing subgroup [24].

This classification-to-feedback translation formed a key integration challenge, as it required careful timing, visual responsiveness, and synchronization between back-end logic and front-end rendering. With these elements successfully integrated, the system was ready for real-time testing and evaluation, as discussed in the next section.

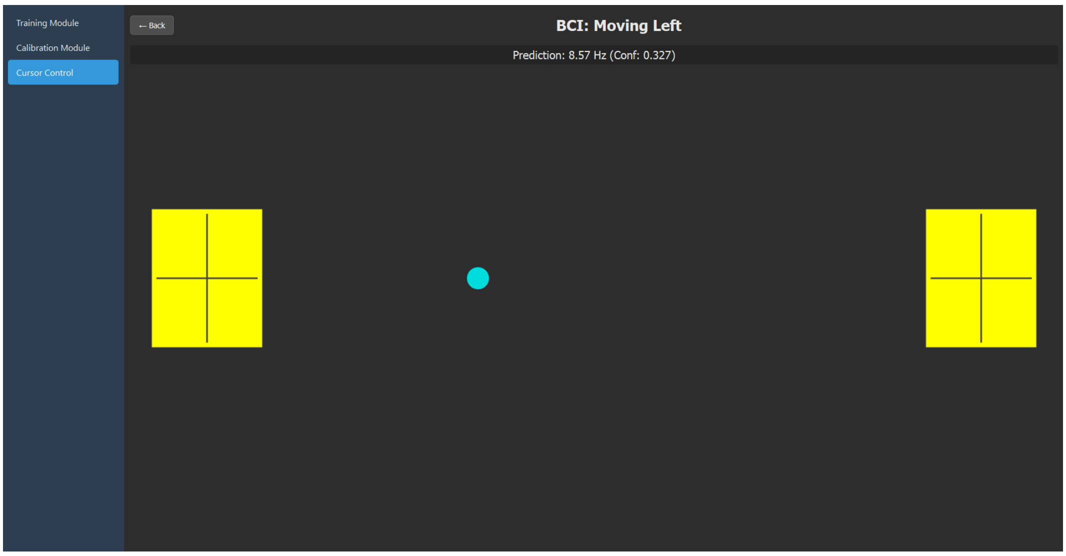



**Figure 6.1:** Cursor Control interface during real-time BCI operation

## 6.3 Evaluation of Real-Time Cursor Control

To assess the system as a whole, a series of evaluations were performed under both controlled and real-world conditions. Initial tests were conducted using artificial signals from a function generator to isolate the classifier's performance from real-world EEG noise and brain artifacts. This allowed for precise tuning and verification of the end-to-end pipeline under known signal conditions before validating performance on human subjects. To support this evaluation, the Cursor Control Module was further developed to include a performance session feature, enabling users to review classification results and corresponding confidence scores at each one-second interval after each session. Upon completion of a session, users are presented with a summary plot and the option to export session data in CSV format for further analysis.

### 6.3.1 Measurements with function generator

The performance of the EEG-based BCI for real-time cursor control was rigorously assessed through a controlled measurement setup designed to simulate typical EEG voltages without interference from noise and artifacts commonly found in actual EEG recordings.

A voltage divider circuit was constructed using a function generator to produce signals at frequencies of 8.57 Hz and 12 Hz (see Figure 6.2). The EEG headset analog within the circuit was represented by a voltage meter, with reference nodes connected to ground. Specifically, channel 7 of the EEG headset was connected between two resistors. One resistor was fixed at 280Ω, while the other was a tunable resistor ranging from 2.8kΩ to 25kΩ. This configuration enabled the adjustment of voltages recorded across the EEG headset from approximately 0.1mV up to 10mV. However during testing, the lowest achieved voltage was a few hundred microvolts.

This tunable voltage divider was necessary because the minimum amplitude achievable by the function generator in question ($AFG3021C$) was limited to a minimum of 10mV, exceeding typical EEG signal amplitudes observed in human brain recordings.

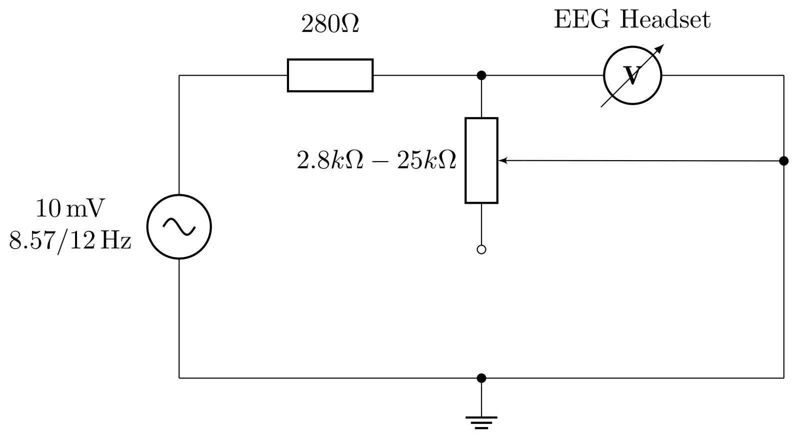
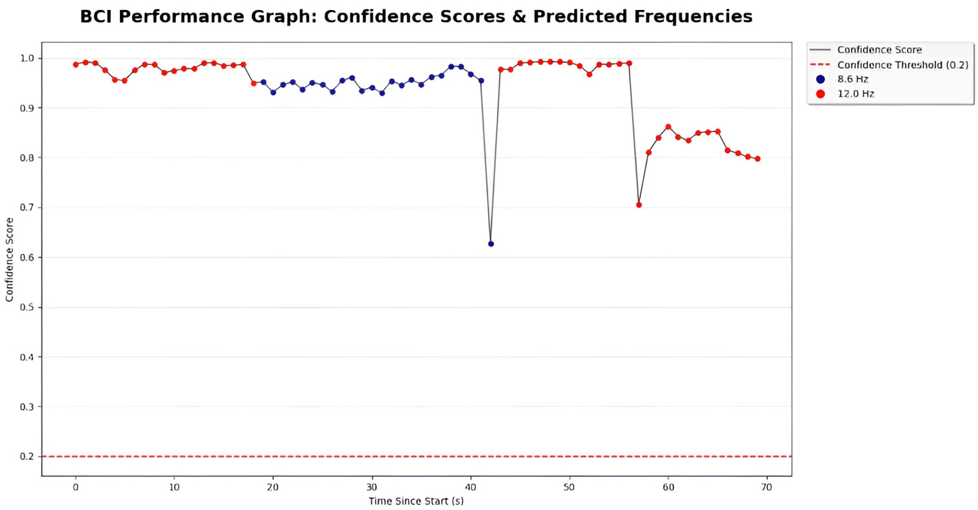


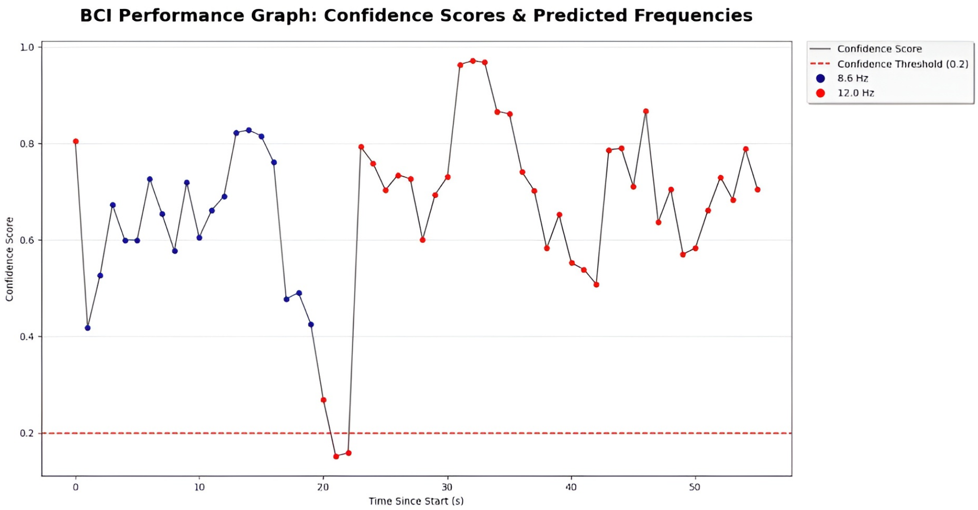

**Figure 6.2:** Measurement setup with function generator and tunable voltage divider

This setup allowed for evaluation of the system's response to EEG-like signals within realistic amplitude ranges, isolated from typical noise and artifact sources. The system demonstrated reliable functionality, maintaining correct frequency classification even at these lower amplitudes.

Two distinct performance graphs depicting confidence scores associated with frequency classifications were generated during the evaluation (Figures 6.3b and 6.3a). When the generated frequency was set within approximately ±0.3Hz of the target classifiers (8.57 Hz or 12 Hz), the system consistently identified the correct frequency with high confidence.




(a) Recording with maximum amplitude at 0.1mV




(b) Recording with maximum amplitude at 1V

**Figure 6.3**

The first graph, characterized by relatively stable and flat curve, represents scenarios where the amplitude measured by the EEG headset was around 1V. Each point in the graph is relatively close to the maximum confindence level. This amplitude resulted in high confidence scores, indicating highly robust system performance. Noticeable dips in the curves correspond precisely to periods of switching from one frequency to the other.

The second graph, exhibiting a more dynamic and fluctuating pattern, corresponds to conditions with amplitudes around 0.1mV. Although not achieving maximum confidence levels, the system still exhibited significantly high confidence scores, demonstrating effective classification even at lower signal strengths.

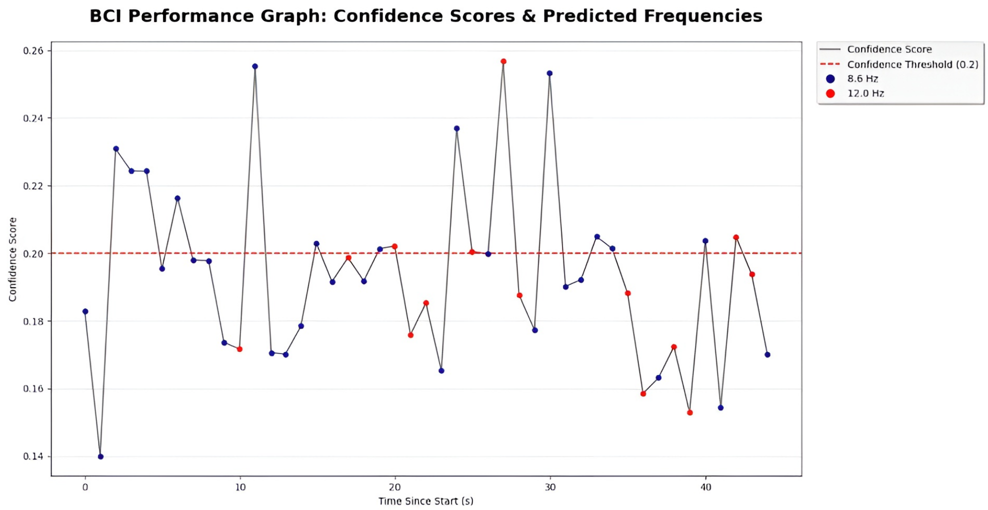
These results affirm the BCI system's effectiveness in accurately identifying EEG signal frequencies under controlled conditions, reinforcing its potential for reliable real-time cursor control in practical applications.

### 6.3.2 Human Subject Measurements

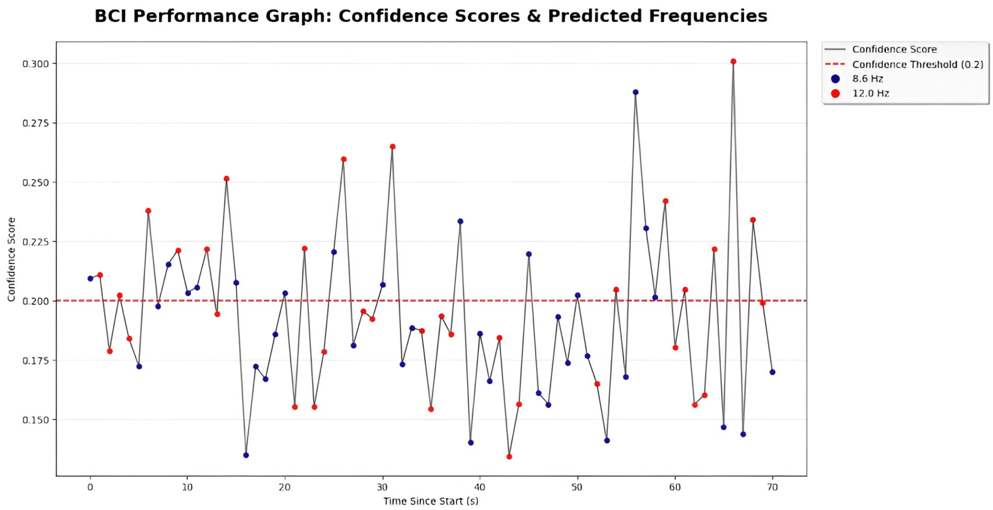
Following the validation using function generator-induced signals, additional measurements were conducted with EEG signals recorded directly from human subjects. These experiments aimed to evaluate the system's real-time performance under natural brain activity, including realistic levels of noise and artifacts.

In these trials, the subject was instructed to focus attention on either the left or the right flickering block in the graphical user interface. In one session, the subject was instructed to look first at the left flicker target and then shift to the right. In another session, the order was reversed. Each of these conditions was repeated to assess consistency and robustness of classification.

The resulting performance graphs of the 2 sessions, presented in Figures 6.4a and 6.4b, show significantly more fluctuation in confidence scores compared to the function generator trials. The curves are notably more fluctuating and more than half of the time do not satisfy the 0.2 confidence threshold. This signifies that the classification algorithm is struggling with decreased the signal quality, mainly in part from the brain artifacts caused by blinking and other processes. Moreover, physiological artifacts such as eye-blinking, subtle head movements, and other involuntary muscle activity contribute additional noise, degrading classification accuracy.




**(a)** Recording with subject looking first to the left




**(b)** Recording with subject looking first to the right

**Figure 6.4**

Unlike the function generator trials, the system showed noticeably more difficulty consistently identifying whether the subject was focusing on 8.57 Hz or 12 Hz targets. This observation highlights the challenge of translating performance from idealized test setups to real-world applications and reinforces the importance of robust artifact filtering and adaptive signal processing for future improvements.

# 7

# Results and Discussion

## 7.1 Results and Requirement Fulfillment

The developed Graphical User Interface (GUI) for the SSVEP-based Brain-Computer Interface (BCI) successfully meets nearly all specified requirements and performs reliably within the overall system architecture. It was designed to support real-time responsiveness, intuitive user interaction, and seamless integration with signal processing and data acquisition subsystems—objectives that were largely achieved.

During testing, the GUI demonstrated full functionality and operational independence. It reliably processed EEG signals from all sources - whether from actual hardware, mock LSL data streams, or artificial function generator signals - displaying them in real time. This confirms the interface itself is not the system's limiting factor, as classification accuracy depends primarily on the back-end processing

All core modules of the interface—Main Menu, Training, Calibration, and Cursor Control—performed according to design specifications. The user interface offers intuitive navigation and does not require prior BCI experience, making it accessible to first-time users. Visual stimuli are rendered with precise timing control aligned to the screen refresh rate, and the Training Module supports configurable EEG data recording trials for offline analysis. Real-time feedback, including cursor movement and prediction indicators, accurately reflects classification output, ensuring a responsive and interactive user experience.

In line with the defined system-level and subgroup-specific requirements, latency has remained well within acceptable limits. EEG data visualization happened almost instantaneously, and interaction with the classification pipeline fits within the broader target of 1 second end-to-end latency. Real-time operation, smooth interaction, and low-latency streaming were maintained across several trials.

The complete BCI system pipeline—from signal acquisition and stimulus presentation to real-time streaming, classification, and user feedback—was successfully validated using artificial input. However, when tested with real EEG signals, the system encountered significant performance limitations, particularly in the classification module. Despite extensive GUI-side optimizations—including adjustments to the flicker stimulus design (e.g., color, size, layout, focus indicators) and even simplifying the interface to a single flickering element—classification accuracy remained below acceptable levels for reliable control.

The GUI modules did nonetheless interact correctly with other system components, consistently transmitting control signals and receiving classification output. This confirms successful integration with the data acquisition and signal processing modules and that the GUI is not the source of the system's shortcomings.

In conclusion, the GUI successfully serves as an intuitive, reliable interface for SSVEP-based cursor control. Its modular and extensible design provides a strong foundation for future enhancements. With improvements to signal quality and classifier performance, this system can evolve into a fully functional, practical BCI system.

## 7.2 Reflections and Future Work

Working on this project provided valuable hands-on experience in translating theoretical BCI concepts into a functional, real-time system. Throughout the development process, the GUI subgroup gained deeper insights into cross-disciplinary collaboration, iterative problem-solving, and user-centered interface design. One key takeaway was the importance of modular implementation and early testing, which allowed the interface to remain functional even before full integration with other subgroups. This flexibility proved essential for debugging and validating GUI components independently. Overall, the project highlighted both the challenges and rewards of engineering a working BCI pipeline in a collaborative setting.

Potential improvements for the overall BCI system include:

- Implementing adaptive algorithms and machine learning to better handle individual variations in EEG signals.
- Increasing robustness against EEG artifacts and signal noise to improve overall reliability.
- Enhancing the user interface for greater intuitiveness and accessibility.

Future development directions for the GUI include:

- Multiclass classification with three or four inputs, expanding the range and complexity of user interactions. Also exploring the possibility of other known BCI paradigms and potentially combining them could help in providing multiple ways to interact with the system.
- Extending functionality with user-specific settings to improve customization and user experience.
- Transitioning from a Python script into a full-fledged Windows application to enhance usability for both individual users and commercial deployment, facilitating broader adoption.

# 8
# Conclusion

This thesis presented the design and implementation of a user-focused graphical interface for a real-time SSVEP-based brain-computer interface (BCI) system. Developed in collaboration with Data Acquisition and Signal Processing subgroups, the GUI enabled users to control a computer cursor using only brain activity, with key functionalities including real-time EEG visualization, visual stimulus presentation, configurable data recording trials , and live feedback based on classification results.

Throughout development, emphasis was placed on creating an intuitive and modular interface that could operate independently while integrating seamlessly into the broader BCI pipeline. The system was successfully validated using both artificial input and human EEG recordings, demonstrating robust real-time performance under controlled conditions. Despite challenges related to signal noise and classification reliability with real EEG data, the GUI subsystem itself fulfilled all functional, performance, and integration requirements.

Importantly, the findings confirmed that the GUI was not a limiting factor in system performance. Instead, the limitations observed in real-world classification accuracy were primarily linked to signal quality and inherent EEG variability, reinforcing the need for more advanced signal processing and adaptive algorithms in future work.

Ultimately, this project has laid a strong foundation for further development in accessible BCI applications. The GUI's architecture allows for future enhancements such as multiclass control, personalization, and broader integration into real-world assistive systems. With continued improvements to signal quality and classification robustness, this system has the potential to evolve into a reliable and practical tool for non-invasive brain-computer interaction.

# Bibliography


[1] X.-Y. Liu, W.-L. Wang, M. Liu, *et al.*, "Recent applications of eeg-based brain-computer-interface in the medical field," *Military Medical Research*, vol. 12, Mar. 2025. DOI: `10.1186/s40779-025-00598-z`.

[2] L. F. Nicolas-Alonso and J. Gomez-Gil, "Brain computer interfaces, a review," *Sensors*, vol. 12, no. 2, pp. 1211–1279, 2012, ISSN: 1424-8220. DOI: `10.3390/s120201211`. [Online]. Available: `https://www.mdpi.com/1424-8220/12/2/1211`.

[3] M. Orban, M. Elsamanty, K. Guo, S. Zhang, and H. Yang, "A review of brain activity and eeg-based brain–computer interfaces for rehabilitation application," *Bioengineering*, vol. 9, no. 12, 2022, ISSN: 2306-5354. DOI: `10.3390/bioengineering9120768`. [Online]. Available: `https://www.mdpi.com/2306-5354/9/12/768`.

[4] G. A. M. Vasiljevic and L. C. de Miranda and, "Brain–computer interface games based on consumer-grade eeg devices: A systematic literature review," *International Journal of Human–Computer Interaction*, vol. 36, no. 2, pp. 105–142, 2020. DOI: `10.1080/10447318.2019.1612213`.

[5] F.-B. Vialatte, M. Maurice, J. Dauwels, and A. Cichocki, "Steady-state visually evoked potentials: Focus on essential paradigms and future perspectives," *Progress in Neurobiology*, vol. 90, no. 4, pp. 418–438, 2010, ISSN: 0301-0082. DOI: `https://doi.org/10.1016/j.pneurobio.2009.11.005`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0301008209001853`.

[6] E. Netzer, A. Frid, and D. Feldman, "Real-time eeg classification via coresets for bci applications," *Engineering Applications of Artificial Intelligence*, vol. 89, p. 103 455, Mar. 2020. DOI: `10.1016/j.engappai.2019.103455`.

[7] S. Saha, K. A. Mamun, K. Ahmed, *et al.*, "Progress in brain computer interface: Challenges and opportunities," *Frontiers in Systems Neuroscience*, vol. Volume 15 - 2021, 2021, ISSN: 1662-5137. DOI: `10.3389/fnsys.2021.578875`. [Online]. Available: `https://www.frontiersin.org/journals/systems-neuroscience/articles/10.3389/fnsys.2021.578875`.

[8] I. Choi, I. Rhiu, Y. Lee, M. H. Yun, and C. S. Nam, "A systematic review of hybrid brain-computer interfaces: Taxonomy and usability perspectives," *PLOS ONE*, vol. 12, no. 4, pp. 1–35, Apr. 2017. DOI: `10.1371/journal.pone.0176674`. [Online]. Available: `https://doi.org/10.1371/journal.pone.0176674`.

[9] A. Kübler, E. M. Holz, A. Riccio, *et al.*, "The user-centered design as novel perspective for evaluating the usability of bci-controlled applications," *PLOS ONE*, vol. 9, no. 12, pp. 1–22, Dec. 2014. DOI: `10.1371/journal.pone.0112392`. [Online]. Available: `https://doi.org/10.1371/journal.pone.0112392`.

[10] W. Ishihara, K. Moxon, S. Ehrman, M. Yarborough, T. Panontin, and D. Nathan-Roberts, "Feedback modalities in brain–computer interfaces: A systematic review," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 64, pp. 1186–1190, Dec. 2020. DOI: `10.1177/1071181320641283`.

[11] O. Maslova, Y. Komarova, N. Shusharina, *et al.*, "Non-invasive eeg-based bci spellers from the beginning to today: A mini-review," *Frontiers in Human Neuroscience*, vol. 17, p. 1 216 648, 2023. DOI: `10.3389/fnhum.2023.1216648`.

[12] L. J. Trejo *et al.*, "Bcis for 1-d and 2-d cursor control: Spectral vs. ssvep," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 2, 2006. DOI: `10.1109/TNSRE.2006.875578`.

[13] X. Chen *et al.*, "Online adaptation boosts ssvep-based bci performance," *IEEE Transactions on Biomedical Engineering*, 2021. DOI: `10.1109/TBME.2021.3133594`.

[14] Y. Liu *et al.*, "A high-speed ssvep-based bci using dry eeg electrodes," *Scientific Reports*, vol. 8, 2018. DOI: 10.1038/s41598-018-29187-4.

[15] Y.-L. Chang *et al.*, "Ssvep-based bci using novel qr-code stimulus patterns," *Journal of Neural Engineering*, vol. 20, 2023. DOI: 10.1088/1741-2552/acbee0.

[16] M. Reitelbach *et al.*, "Optimal stimulus properties for steady-state visually evoked potential brain–computer interfaces: A scoping review," *Multimodal Technologies and Interaction*, vol. 8, no. 2, 2024. DOI: 10.3390/mti8020006.

[17] R. Kus, A. Duszyk, P. Milanowski, M. Łabecki, M. Bierzynska, *et al.*, "On the quantification of ssvep frequency responses in human eeg in realistic bci conditions," *PLoS ONE*, vol. 8, no. 10, e77536, 2013. DOI: 10.1371/journal.pone.0077536.

[18] C. Farmaki *et al.*, "Applicability of ssvep-based bcis for robot navigation," in *Proceedings of the IEEE EMBC*, 2016. DOI: 10.1109/EMBC.2016.7591304.

[19] M. Benda and I. Volosyak, "Comparison of different visual feedback methods for ssvep-based bcis," *Brain Sciences*, vol. 10, no. 4, 2020. DOI: 10.3390/brainsci10040240.

[20] K. Sumi, K. Yabuki, T. J. Tiam-Lee, *et al.*, "A cooperative game using the p300 eeg-based brain-computer interface," in *Assistive and Rehabilitation Engineering*, Y. Rybarczyk, Ed., Rijeka: IntechOpen, 2019, ch. 10. DOI: 10.5772/intechopen.84621. [Online]. Available: https://doi.org/10.5772/intechopen.84621.

[21] C. Hindi Attar and M. Müller, "Selective attention to task-irrelevant emotional distractors is unaffected by the perceptual load associated with a foreground task," *PloS one*, vol. 7, e37186, May 2012. DOI: 10.1371/journal.pone.0037186.

[22] D. A. Subgroup, "Ssvep-based brain-computer interface for cursor control," https://drive.google.com/file/d/1j3WbSNagE_gPMLHrHj0y460ToweYtR-d/view?usp=sharing, Bachelor's thesis, Delft University of Technology, 2025.

[23] g.tec, *Unicorn hybrid black*, https://www.gtec.at/product/unicorn-hybrid-black/, Accessed: 2025-06-12, 2019.

[24] S. P. Subgroup, "Ssvep-based brain-computer interface for cursor control," https://drive.google.com/file/d/1Q0gbB2TdpTL2zMEjlEz7QBbACA7RSOK_/view?usp=sharing, Bachelor's thesis, Delft University of Technology, 2025.

[25] X. Chen, Y. Wang, S. Gao, T.-P. Jung, and X. Gao, "Filter bank canonical correlation analysis for implementing a high-speed SSVEP-based brain-computer interface," *Journal of Neural Engineering*, vol. 12, no. 4, 046008, p. 046 008, Aug. 2015. DOI: 10.1088/1741-2560/12/4/046008.

# A

# Source Code

Click here to view the source code

## A.1 Statement use of AI

AI tools were used to assist in writing this report. They helped summarize certain papers, clarify complex methodologies, refine vocabulary, and improve the conciseness and readability of some self-written sections.