Aisling Moloney, R00260662

Dr. Cárthach Ó Nuanáin

Music & Technology Research (MUSC9004)

Dissertation

15th August 2025

Mapping Cognitive Control: EEG Signals Modulating Real-Time Music Composition

Aisling Moloney

A Thesis in the Field of Music Technology

As part of the Master of Science Degree in Music & Technology,

awarded by Cork School of Music

Supervisor: Cárthach Ó'Nuanáin

Submitted to Cork School of Music (CSM) & Munster Technological University (MTU)

August 2025

Abstract

This thesis explores the use of real-time electroencephalogram (EEG) signals to control Virtual Studio Technology (VST) parameters within a Digital Audio Workstation (DAW), using NeuroBell's portable Luna EEG Amplifier. While many studies have examined the effect of music on the brain, there is growing interest in using brain activity to directly influence musical output instead. In this study, four distinct stimulus environments were designed to engage each of the brain's primary lobes (frontal, parietal, temporal, and occipital), each targeting specific cognitive or sensory processes known to generate characteristic neural oscillatory patterns.

EEG signals from the NeuroBell device were processed in real time via a custom Python script, extracting frequency band features (theta, alpha, beta rhythms) per lobe. These features were mapped to MIDI Control Change messages and routed into Reaper, where they modulated parameters of a wavetable synthesiser playing standardised musical material. The system was first prototyped with sample EEG datasets, then implemented with live data from participant trials.

Results from testing showed that the pipeline successfully achieved stable EEG-to-MIDI translation and real-time VST modulation, with some participants perceiving subtle musical changes during targeted tasks. However, perceptual feedback varied enough to lack full validation of the system.

This research demonstrates a viable, low-cost brain–control music interface, highlighting opportunities for creative expression, performance, and clinical applications. Limitations include small sample size, task duration, and perceptual reliability. This suggests future work in enhanced feature normalisation, expanded mapping structures, and integration with other biometric signals.

Keywords: *EEG, Brain–Computer Music Interface (BCMI), Digital Audio Workstation (DAW), Virtual Studio Technology (VST), MIDI mapping, real-time signal processing, music technology.*

Author's Biographical Sketch

Aisling Moloney was born in Limerick, Ireland. She earned her Level 8 Bachelor BE (Hons) in Electrical and Electronic Engineering from the University of College Cork in 2024. She went on to begin her Master of Science degree in Music & Technology at Munster Technological University in 2024/25.

Dedication

I would like to sincerely thank all the people in my life who helped me through the development of this thesis, along with the master's course over the past year. Thank you to all my fellow peers in Music Technology, as well as to the lecturers of CSM and MTU.

Thank you to Dr. Mark O'Sullivan, a previous alumnus of both my degrees, for seeing the passion and dedication in me and for providing so much guidance and support both technically and emotionally throughout the project.

Finally, a special thank you to my parents, Ann and Con, my two brothers, and partner Isaac for always showing their unwavering support for not only my project but for me on this journey

Acknowledgments

I would like to formally thank the amazing people of NeuroBell, Ballincollig, for allowing me to bring this vision to fruition. A special thank you to the CSO, Dr. Alison O'Shea, and to the CTO, Colm Murphy, for all the help regarding the development and use of the Luna Amplifier, along with all the EEG dataset advice.

Table of Contents

1. Introduction

Many studies have investigated the effect of music on the brain; however, there is an emerging field using brain activity to control music instead. Specific sections of the brain are responsible for various sensory processing functions of the outside world. This has been harnessed in this project and analysed using existing electroencephalogram (EEG) technology through this musical context.

Previous studies have shown that not only are certain parts of the brain responsible for auditory processing of music, but this electrical activity can actually be reprocessed back into the song they are hearing [1]. Certain parts of the brain can be attributed to the processing of specific musical elements; melody, harmony, and rhythm, activating many of the same regions that speech does [2]. As well as this, studies have shown that the emotional state of a person can be used, via EEG data, to trigger and expressively control musical transformations [3].

This study investigates using brain activity to control Virtual Studio Technology (VST) parameters within a Digital Audio Workstation (DAW). This provides an interactive, active music listening experience using real-time signal activity from four brain regions. This utilises EEG technology was used to record brain activity from the scalp and feed it directly into a DAW via a USB connection. There is a possibility for this information to later be live streamed to an internal cloud software and accessed via IP address to then feed directly into a DAW.

A comparison across brain regions was performed using these EEG signals as controllers of VST parameters while the subject experiences various stimuli. Each environment was designed to engage each region, according to the associated lobe function. The variance of control was analysed and compared in both a quantitative and qualitative sense. This type of control over the expressive aspects of the music, rather than a more literal control over playback, etc., would hopefully provide an intriguing insight into our brain's perception of the world and music. The final hypothesis being: can sensory-specific lobes distinctly influence VST control and musical expression?

2. Literature Review & State of the Art

2.1. Background & Theoretical Framework of EEG Signal Processing

Sensory processing refers to how the brain receives, organises, and interprets sensory input from the environment. Each sensory modality (vision, hearing, touch, etc.) primarily engages specific brain regions. Focussing on the cerebrum, each of the brain's hemispheres consists of four true lobes: the Frontal, Parietal, Temporal, and Occipital lobes. Rudimentarily, these can be respectively distilled to movement, spatial reasoning, light/visual, and auditory stimulation [4]. Fig. 2. 1. 1 below shows the anatomy of each brain hemisphere, with the parts of the cerebrum highlighted.

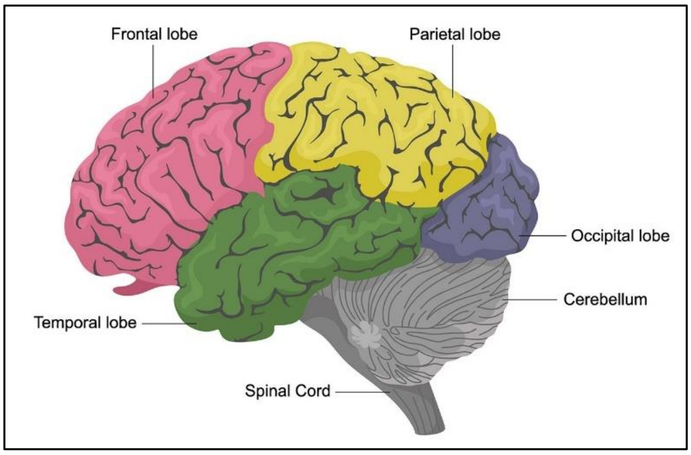

Fig. 2. 1. 1. *Human brain anatomy, highlighting parts of the cerebrum*

The 10-20 system describes the method of exact placement locations of scalp electrodes in the context of an EEG exam [5]. This standardised method allows for the subject's EEG activity to be compiled and compared effectively to others. This system exploits the relationship between the electrode location and the underlying area of the brain. Through these electrodes, the distinct electrical patterns of these areas of the brain can be detected. The "10" "20" placement refers to the distance intervals between adjacent electrodes that are either 10% or 20% of the total front-back (median plane) or right-left (transverse plane) distance of the skull.

In Fig. 2. 1. 2 below, the transverse plane of the skull can be seen along with the standard 10-20 electrode locations. The nasion and inion are used as reference points for the measurement, with the nasion referring to the indent at the top of the nose and the inion referring to the bony bump at the back of the skull. The letters seen at electrode sites correspond to the true lobes (Frontal (F), Parietal (P), Temporal (T), Occipital (O)), with the addition of the Central (C) lobe electrodes. Although there is no "central lobe", depending on the individual, this electrode can represent/exhibit EEG activity more typical of frontal, temporal, and some parietal-occipital activity. The “Z” letters instead refer to electrodes placed along the midline sagittal plane of the skull (plane connecting the nasion to the inion). These sites are not used to reflect either hemisphere’s cortical activity but instead are used as “grounds” or “references”. These are often used in clinical EEG montages to diagnose seizure activity.

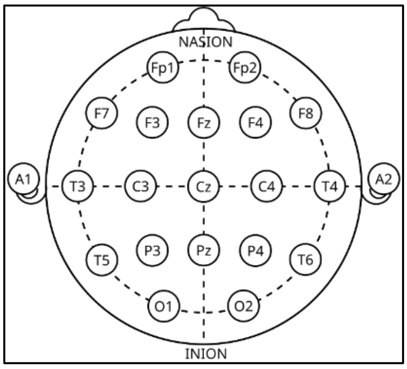

Fig. 2. 1. 2. *Electrode locations of the 10-20 system for EEG recording*

It should be noted that sensory information is not processed in isolation. Instead, the brain integrates sensory signals across modalities, a phenomenon known as multisensory integration, which is important for coherent perception and adaptive behaviour.

Primary sensory cortices, such as the primary visual cortex (V1), somatosensory cortex (S1), and primary auditory cortex (A1), initially decode raw sensory input. These areas are located within each of the specific lobes listed above, with the somatosensory integration located in the parietal lobe. The linear regression model in this study [6] allows for the association of the sensory parameters (β_V, β_S, β_A) with that of the primary sensory signals across the cortical surface of the brain. In Fig. 2. 1. 3 below, this can be seen on the brain surface mapping of group-averaged sensory parameters.

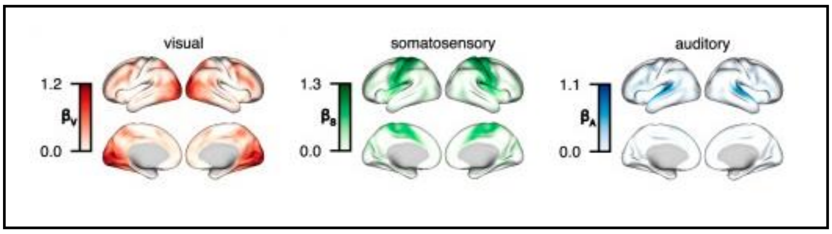

Fig. 2. 1. 3. *Surface mapping of group-averaged sensory parameters*

Beyond primary sensory areas, association cortices, particularly in the parietal, temporal, and frontal lobes, instead combine multiple streams of sensory input. However, EEG studies show that sensory processing is reflected in specific neural modulations. This distinction can allow for the discretion of activity across cortices. For example, Alpha waves (8 - 13 Hz) are associated with visual and attentional processing. Theta waves (4 - 8 Hz) have been linked to memory encoding and retrieval. Gamma waves (>30 Hz) are thought to underlie the "binding" of different sensory elements into unified precepts, key for cognition. Fig. 2. 1. 4 shows some examples of these various types of EEG signals.

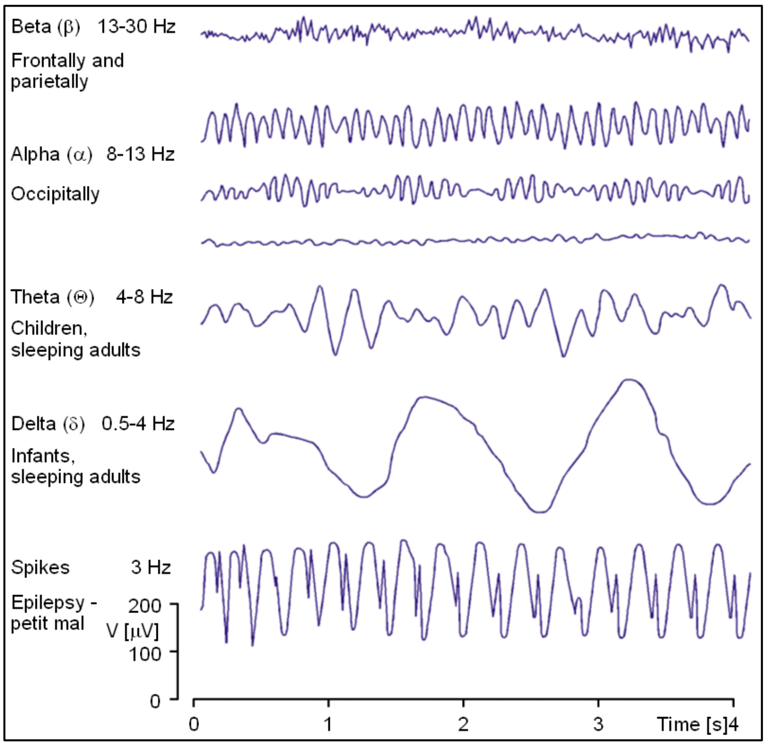

Fig. 2. 1. 4. *Examples of EEG Waves*

The core concepts can be illustrated as follows:

Sensory input ➔ Primary sensory cortex ➔ Association cortex ➔ Cognition

Predictive coding theories suggest that the brain is constantly generating hypotheses about incoming sensory information, with cognitive functions like attention and learning emerging from updating these internal models. Sensory signals are processed dynamically, shaped by expectation, emotion, and experience. EEG signals reflect these processes and serve as real-time markers of cognitive engagement with different sensory stimuli.

2.2. History and Current Use of EEG in Music Tech

Musical perception and performance involve complex computational processes across distributed networks within the brain. Numerous studies have explored the neural basis of these processes, distinguishing specific brain regions responsible for perceiving and producing various musical elements, including rhythmic patterns, pitch, melody, timbre, and tonality [7].

For example, neuropsychological studies have implicated regions in the right hemisphere involved in time-dependent attention and working memory functions, which collectively contribute to the perception of auditory stimulus duration [8], [9]. While another study that used magnetoencephalography (MEG) signals found that the amplitude of M100 auditory field potentials directly correlated to changes in the duration of rhythmic intervals [10]. With increases in durations resulting in M100 auditory amplitudes increases, and decreases showing the inverse.

This is all to predicate the evolution of the integration of EEG signals into the control of musical parameters, performance and composition. This dedicated, in-depth research has provided the basis for many creative endeavours. Integrating EEG control into music technology allows for a conjoined insight into both fields, offering a new avenue of creative expression and technological development.

One example of this emerging field includes the development of "The Encephalophone" [11]. This is a musical biofeedback device using the conscious control of EEG signals. This device uses the alpha wave band activity located in the posterior region of the brain, especially in the visual cortex of the occipital lobes. One oscillatory pattern being used is known as the posterior dominant rhythm (PDR) or alpha rhythm. This occurs in healthy, awake individuals with their eyes closed. This device also uses the mu (μ) rhythm (8 – 13 Hz) EEG pattern observed in the central regions of the brain, in the motor cortex of the brain. Together, this frequency signal power controls a synthesised piano, known as the Encephalophone. This data was used to create a power scale, then converted into a musical scale which the individual could control in real time. A flow diagram of this experimental setup can be seen below in Fig. 2. 2. 1. This device displays the functionality of EEG-controlled musical synthesis.

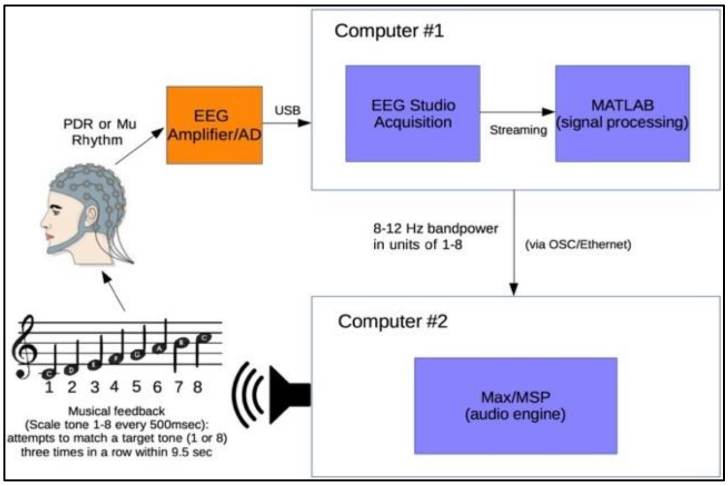

Fig. 2. 2. 1. Block flow diagram of the Encephalophone functionality

Brain-computer musical interfaces (BCMI) have been developed for musical applications aiming to interface brain waves directly with compositional tools, instruments, and algorithmic composers [12]. Eduardo Miranda, professor at the Interdisciplinary Centre for Computer Music Research (ICCMR) at Plymouth University, UK, has contributed greatly to this field. One project used the changing patterns of alpha and beta (13 - 30 Hz) frequency rhythms to switch between different musical styles [13]. This project initially employed a 32-channel EEG placement along with three computers and hacked software to produce this EEG-controlled music mixer. However, this went on to be adapted and developed continually. For example, using a more affordable and portable 4-channel EEG amplifier and introducing new techniques to produce music from the topological behaviour of EEG signals [14]. As well as this, another development involved using the subject's visual gaze direction to allow steady-state visually evoked potentials (SSVEP) of EEG to control musical parameters [15].

Another aspect of this development is providing musical composition for those with severe motor impairments. One project produced a bespoke BCMI for such a person, who had previously been a violinist [16]. Similar to [15], this is an SSVEP-based BCMI. Crucially, this is a dry, wireless, portable EEG headset that provides brain wave activity using just four electrodes (Cz, Pz, O1, O2) for this novel application.

2.3. Technology Overview: NeuroBell Luna and DAW Integration

One accessible, low-cost, and mobile EEG device is that of NeuroBell's Luna EEG Amplifier technology. This device was developed for Neonatal Intensive Care Unit (NICU) staff to deliver automated machine-learning seizure detection in neonates [17]. This system allows for EEG recordings to be streamed via WIFI to a hospital server. Theoretically, this would provide the non-invasive recording of the EEG data required for the study. Then, using the NeuroBell cloud software, this would allow for a real-time stream of raw brain activity data to be routed, via IP address and fed directly into a DAW, allowing for the implementation of VST parameter control.

The NeuroBell device uses an 8-channel EEG electrode placement within the previously discussed 10-20 EEG placement system. The electrodes are placed as follows; Pz, Cz, F3, F4, C3, C4, T3, T4, O1, O2. Compared to the previously discussed SSVEP-based BCMI device, the larger electrode placement array and information could provide a more widespread application of this type of device.

Using alpha/beta frequencies as an example of parameter control within a DAW, the following setup could be achieved. Mirroring the flow diagram used in Fig. 2. 2. 1, the wirelessly uploaded EEG signals could be sent to a computer where, using MATLAB signal processing, 8-13 Hz power is converted into 50 values. This could be sent to a DAW like Ableton, where the values are mapped to FX parameters like reverb, delay, etc. The 50 values would correspond to and be mapped in real time to 25-75% of the plugin effect application. This range provides a gradient of change over the musical playback, rather than jarring transitions. This is just one example of the many ways the different EEG frequency patterns could be mapped to musical control within the audio engine.

2.4. Design of Experimental Framework

To further develop the concept of comparing EEG VST control across brain lobes, the design of four environments can be developed based on past research. Although the lobe function and the corresponding frequency patterns for this task can overlap with other lobes, this can be addressed on a case-by-case basis, incorporating the necessary band frequencies when crucial to the function. In each case, a 50 Hz notch filter should be applied to eliminate the power supply frequency present in the EEG data. The EEG data is filtered in real-time for each case and mapped to MIDI control signals.

2.4.1. Occipital Lobe Stimulus Environment

Based on previous experimental methods from the posterior region of the occipital lobe, one environment design could involve subjects experiencing rhythmic visual stimuli to induce alpha-band oscillations. This comes from research into the resonant response in visual areas, where the application of visual stimulation at alpha-frequency (10.6 Hz) leads to a prominent rhythm of the visual system (alpha-rhythm) [18]. This offers a sub-comparison opportunity within this lobe of the brain, by providing different repetitive stimuli and comparing the subsequent effect of the alpha frequencies on VST control.

In this test environment, it is important to negate any unwanted artefacts within the EEG data. The participant should be placed in a quiet, dimly lit, sound-attenuated room to minimise distractions [19]. They should be relaxed with an upright posture with minimised movement, as jaw and facial muscle activity can introduce broadband EEG artefacts [20]. As well as facing the subject away from any unwanted stimuli, such as the DAW environment, that could cause visual feedback or interference.

2.4.2. Temporal Lobe Stimulus Environment

Similarly, to engage the temporal lobe, responsible for auditory processing, rhythm, and memory, this involves a relaxed subject sitting with a headrest to minimise any muscle-movement induced frequencies. Auditory stimuli are delivered via headphones and include rhythmic patterns, spoken phrases, and memory-based recall tasks. Rhythmic sequences elicit gamma-band activity (30 - 50 Hz) in the superior temporal gyrus, linked to auditory discrimination and attention [21]. While recall memory tasks typically increase theta-band power (4 - 7 Hz), especially in hippocampal and temporal areas [22]. One study compared the EEG activity in response to music liked and disliked by the patient [23]. According to this, music that is personally liked by the subjects seems to enhance the EEG power spectra globally and across bandwidths. One parameter mapping example here could involve gamma activity to modulate a filter cutoff, while theta power during memory tasks could adjust delay speed.

2.4.3. Frontal Lobe Stimulus Environment

Here, to engage the frontal lobe, responsible for executive function, motor planning, and emotion, the subject is placed in the same physical environment. EEG electrodes focusing on frontal areas, F3 and F4, are used to capture beta and gamma rhythms linked to cognitive and emotional activity [24]. Because frontal sites are prone to eye movement artefacts, the subject should minimise blinking or eyebrow movements during tests. One type of task could include complex working memory challenges, which increase frontal theta and high-frequency activity during cognitive load [25]. Another task could lean into the emotional stimulation of the frontal lobe. Emotional stimuli, such as video clips or memory recall, are used to evoke affective states. Emotional arousal increases frontal beta coherence and gamma power, especially under anxiety or excitement [26]. Finally, for motor planning, subjects could perform motor imagery, like imagining hand movement, producing event-related desynchronization (beta decrease), followed by a beta rise when it is ended [25].

These changes could be used as EEG-controlled parameter triggers, for example, beta suppression during motor imagery could mute/unmute a track, while gamma increases from emotional or cognitive effort could drive effect intensity.

2.4.4. Parietal Lobe Stimulus Environment

Finally, to engage the parietal lobe, associated with somatosensory integration and spatial cognition, this relaxed setup is used again and involves the subject receiving tactile and spatial stimuli designed to activate P3, P4, and Cz electrodes [27]. Tactile input could be delivered through skin-mounted buzzers or vibrators in patterned pulses; this could be passively experienced by participants or provide the ability to discriminate between the stimuli. These stimuli should provoke mu and beta rhythm suppression over sensorimotor areas, reflecting somatosensory engagement [28] [29]. Spatial tasks like mental rotation engage the superior parietal lobule and reduce parietal alpha rhythms as the task complexity increases. Processed EEG signals, like the mu or beta power at P3/P4, could be appropriately mapped to MIDI or OSC messages, controlling VST parameters in a DAW. For instance, mu suppression during touch could increase an FX intensity, with the subsequent rise reducing it. Variants such as passive vs. active touch or 2D vs. 3D spatial tasks allow for further exploration and comparison of EEG control and tuning across parietal processes.

3. Research & Methodology

3.1. Software Methodology

3.1.1. Overall Software Approach

The objective of this thesis allowed for a multitude of possible outcomes, which will be discussed further in the Future Work section of this paper. Throughout the development of the project, the efficacy of applying EEG control onto musical parameters was researched, for example, whether this data was chosen to be used in a sound synthesis stream or to vary the timbre of the music. It was important during this research stage to distil these possibilities down to a tangible outcome, with regard to both the physical hardware and software.

Ultimately, the overall workflow of the software design was as follows: EEG data is acquired (from an EDF file or device), then processed via a Python script to extract the features, which are then transmitted to Reaper to modulate music in real-time. A block flow diagram of this can be seen below in Fig. 3. 1. *1*.

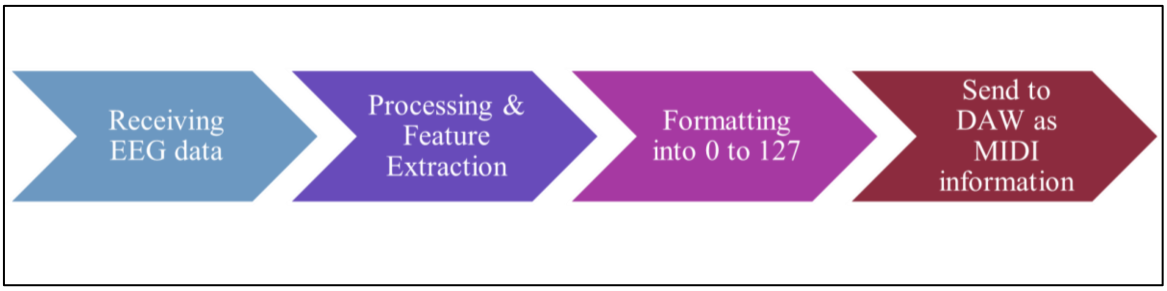

Fig. 3. 1. 1. *Overall Software Block Flow Diagram*

3.1.2. Initial Prototype with Sample Data

Before getting access to the NeuroBell EEG Luna Amplifier device, an initial script was developed to emulate the stream of EEG data that this device would later provide. This step enabled crucial research and development in script formatting and device specifications, while also providing insight into interfacing an IDE with a DAW of choice.

In this stage, initial signal processing could begin by loading in pre-existing event-related EEG datasets that use a larger array of electrodes/channels. By investigating these other studies, where subjects had undergone specific stimuli, along with the timestamps at which they occurred, this could help show areas of the scalp where rhythms are strongest and where they are strongly associated with particular stimuli. As well as this, choosing open-source datasets that used a similar sampling frequency to that of the NeuroBell Amplifier (250 Hz) would allow for a more straightforward implementation later in the project.

The initial code was developed using the MNE library, an open-source Python package for MEG and EEG analysis and visualisation. Using this library along with EDF files acquired from OpenNeuro, a basic EEG processing pipeline for event-related analysis could be achieved. The dataset used was a resting state EEG, where participants had to open and close their eyes [30]. The aim of using this was to experiment with loading data, preprocessing, plotting, and analysing where the event occurred. This code led to exploratory work into power spectral density plots of channels, the rereferencing of EEG channels using "Z" electrode sites and beginning to emulate a real-time feed of EEG data. This code was developed using the Google Colab IDE and can be found in 0. However, after this point, Visual Studio Code IDE was used instead to provide more stability and low-latency processing, and to avoid the timeouts that occur in Colab.

3.1.3. Digital Signal Processing Methods

As mentioned in Section 2.4, digital signal processing is crucial to this project, as filtering and artefact removal should provide a stronger correlation of stimulus experienced to VST parameter control. For the sample EDF file, any preprocessing of the data could be found using the MNE "info" data structure, which displays channel locations and even any applied filters. Following this, a bandpass filter structure was observed with a highpass filter being applied at 0.5 Hz, and a lowpass filter being used at 60 Hz. The initial power spectral density plots of this sample data showed that the average power (μV^2) of the signals occurred within around 0 Hz and 5 Hz. As this dataset was recorded in Mexico, the lowpass threshold of 60 Hz was used to remove the power supply frequency artefact, compared to 50 Hz in Europe.

Applying an appropriate bandpass filter to this specific frequency band activity allows for the feature extraction to begin. A discrete fast Fourier transform (FFT) was used to convert from the time domain EEG data into the frequency domain. This then allows for the calculation of the amplitude value in a specific frequency band correlating to alpha, beta, and theta waves. This band power, as discussed previously, would go on to act as the features during stimulus environments. This process was initially developed in the code of Appendix C), however, a working version of this can be found in the VS Code Appendix D.1). For ease of development, initially, all the channel groups were averaged together into a combined signal before this process. That way, emulating a live EEG stream and then interfacing with various DAWs could be explored with a smaller group of values to send and manage.

To emulate this live stream of data processing using the offline EDF data, the python script splits the continuous EEG signal into 1-second epochs based on the recording's sampling frequency (sfreq), generating fixed-size frames or chunks (chunk_samples = sfreq * 1). A for loop iterates through all samples in the dataset, processing and performing feature extraction across each frame. For each frame, the DC offset is calculated and removed. This step subtracts the mean of the chunk to remove any very-low-frequency drift present. This is important as any artefacts at these low frequencies could mask the desired activity in the theta/alpha/beta ranges. The FFT is then applied using the NumPy library (np.fft.rfft), calculating the frequency spectrum and providing the positive frequency bins up to the Nyquist limit (sfreq/2). These frequency bins (freqs) are found (np.fft.rfftfreq) using sampling frequency and frame length, allowing for the band selection defined earlier for theta (4–7 Hz), alpha (8–12 Hz), and beta (13–30 Hz). Within each of these bands, the mean magnitude of the complex FFT (np.abs(fft_vals)) is calculated as a simple representation for band power in that epoch. A visualisation of these frequency bins in spectral analysis can be seen below in Fig. 3. 1. 2, however, instead of the mean amplitudes for each bin being displayed on the y-axis, the power values from the FFT within each bin are summed and then expressed as a percentage of that sum for each frequency bin/band [31].

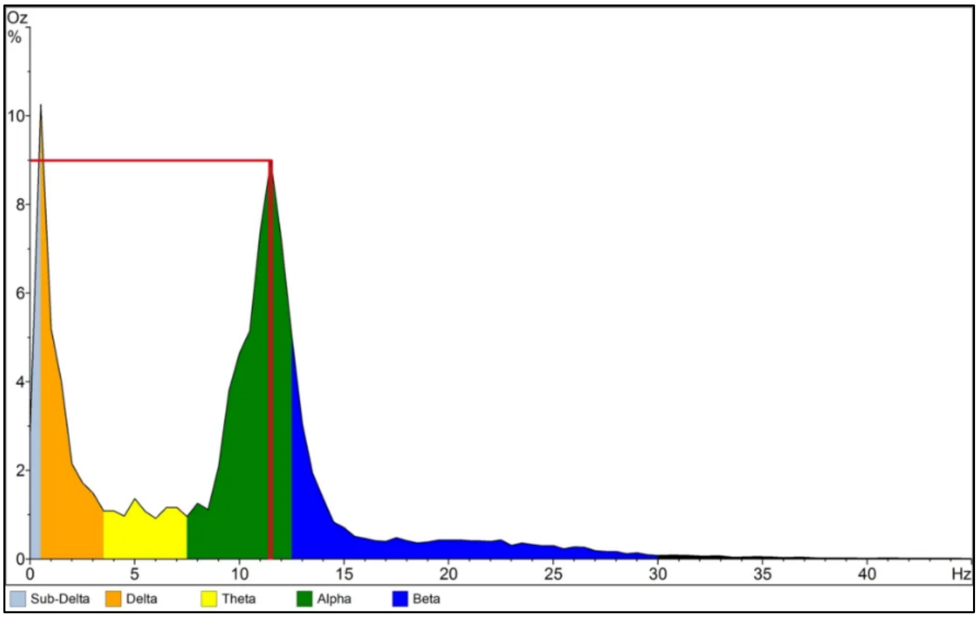

Fig. 3. 1. 2. Normalised Power Spectrum (as %) of EEG Frequency Bands

3.1.4. Feature Mapping & DAW Integration

To begin translating the input EEG signal into information that a DAW could use, feature mapping was integrated. The value of the average magnitude calculated in each frequency band per epoch was scaled according to the MIDI protocol. The code uses the MIDI Control Change (CC) system via the Python mido library. A virtual MIDI output port is opened once at the start (outport = mido.open_output(midi_port_name)), specifically using the loopback device loopMIDI, which any DAW could then recognise as an input. For each 1s analysis window, the feature extraction returns a scalar value per band (theta/alpha/beta). These values are linearly scaled into the MIDI range and dispatched as CC messages.

Instead of the percentage of band power being used, the form of normalisation used in this code was a fixed scaling factor. Through experimentation, an initial value of 1×10^5 was used, since the standard unit of EEG signals from the sample datasets (and would be device) is in μV or 1×10^{-6} V. This allowed for the values to fit into the 7-bit MIDI range of 0-127, clipping any values outside. This method keeps absolute changes in the signal strength and is not dependent on the power present in the other bands. Once again, this is a simple method of mapping, but it would prove to be reliable for real-time mapping. However, this meant that in some instances, such as varying subjects or the lobe engaged, the scaling factor would have to be changed to more effectively map to the MIDI range.

As part of the CC message (mido.Message), each EEG feature is accompanied by a specific CC number (e.g., alpha power to CC22), so the DAW can map that control stream to a specific plugin parameter using standard MIDI learn/automation. This one-to-one mapping should ensure clarity of effect where the changes of each band lead to a direct musical response due to stimulus: increases in band amplitude result in corresponding increases in the controlled parameter.

Further interfacing experimentation was done using various DAWs, such as Ableton and Reaper. The benefit of using Ableton is that it offers more creative control with the many built-in plug-ins, sounds, and effects available. Additionally, mapping from MIDI inputs is intuitive, with MIDI effects that generate chords, scales, and variations. Using Max for Live would also provide the ability to edit code directly in Ableton. However, the CPU usage of Ableton was significantly higher than that of Reaper. Another issue arose when attempting to split the one MIDI input channel containing all of the CC addresses within Ableton. Comparatively, the Reaper VST plug-in Helgobox: ReaLearn & Playtime could easily process each address, learn the source and target of each message, and even allow for presets to be saved [32]. This would later be useful for splitting the multiple streams of data coming from each lobe in the Python script. Reaper's minimal GUI and CPU usage made it the obvious choice for this project.

3.2. Hardware Methodology

3.2.1. Hardware Selection

Through discussions with the team at NeuroBell, they kindly offered to supply the biometric device used in this project. Their EEG device would provide 8 channels of EEG data, as mentioned in Section 2.3. At the time of research in this project, there were multiple iterations of the Luna Amplifier device that could be viable. One option would allow for the storage of prerecorded data from stimulus experiments, which could later be analysed via the script. This would have removed the real-time variation of the VST parameters. Another version would provide a live stream of data via an IP address. Due to manufacturing delays, this wouldn't be feasible for the project deadline and would also involve research into compatibility with DAWs, for example, which would provide the least buffer size. Thankfully, the fantastic team offered a Goldilocks solution, a serial port USB-compatible version of the device. This iteration is battery-powered and chargeable via the USB-C port. The play button starts the streaming over USB. The device can be seen below in Fig. 3. 2. 1.

Fig. 3. 2. 1 NeuroBell Luna Amplifier, USB Streaming Iteration

3.2.2. Signal Acquisition

Access and experimentation with this version of the amplifier were possible from mid-July. The device shows up as a COM port on Windows computers, and using the Serial Port Plotter v1.3.0 demo software, the microvolt (µV) values from the electrodes could be plotted [33]. The protocol for displaying information on this software was the basis for transmitting data from this USB EEG device. The data is sent out in 1 packet per second, where each packet has 250 lines and each line has 8 channels. The format of the message over USB is as follows: $Ch1 Ch2 Ch3 Ch4 Ch5 Ch6 Ch7 Ch8; where channels are assigned right to left as seen on the device in **Error! Reference source not found.**. I will discuss the full implementation of the device into this project in the following chapter, Chapter 4.

As previously mentioned, the device contains two reference channels, Pz (Ground) and Cz (Reference or Bias), which are essential to the operation of the amplifier and to EEG acquisition. Pz stabilises the amplifier and rejects common-mode noise, while Cz serves as the reference point for all active channels. Because EEG records the voltage difference between each active electrode and the reference, any neural activity at Cz is subtracted from all other channels, acting as the bias. This can be seen in the formula below.

$$V_{channel} = V_{active\ electrode} - V_{reference\ electrode}$$

Through discussions with the NeuroBell team, placing these reference electrodes at quieter sites, such as the mastoids (behind the base of the ear), minimises this bias. This site helps to highlight the desired patterns in feature extraction by avoiding introducing unnecessary neural activity at the reference and reducing the power subtracted from active sites.

3.2.3. Recording Reliability & Set Up

The use of this EEG hardware in a non-clinical, music-production environment requires a reusable and reliable method. Before each session, the preparation of the electrode contact quality will be checked to ensure low impedance for stable signal acquisition; minimising artefacts and maximising feature extraction accuracy. The use of exfoliant skin prep gel to remove dead skin cells should reduce impedances. As these are non-invasive electrodes being placed onto the scalp, any hair could also cause impedances, so ideally participants with shaved or bald heads would be beneficial. This, along with the use of conductive electrode paste with tape, should provide longevity to the sessions and accurate recording. Procedurally, the participants will be instructed on correct posture and avoidance of sudden head movements during recording to maintain signal stability. During any stimulus experiments, a calibration window will be integrated before and after to allow for a comparison control.

3.2.4. Transition from Prototype to Device

The methodology for this project involves two phases: first, the design and testing of a signal processing and mapping pipeline using offline EEG datasets, and second, the implementation of a refined pipeline with live data from the NeuroBell Luna Amplifier device. The insights gained during the prototype stage, such as filter parameters, appropriate scaling ranges for MIDI mapping, and DAW integration preferences, will directly inform the live implementation.

In transitioning to the device phase, the same software modules will be modified appropriately to accommodate streaming input rather than offline files. The band-power extraction, MIDI mapping logic, and DAW structure will follow the prototype discussed. Now that the operational concept for both the software and hardware has been defined, the subsequent chapter will focus on the final process of implementation and problem-solving undertaken.

4. Development & Device Implementation

4.1. Software Implementation

4.1.1. Code Adaptations

The initial EDF-based prototype described in Chapter 3 was reformatted for the live stream of data from the NeuroBell Luna EEG Amplifier. In this updated Python script, found in Appendix D.2), the MNE read_raw_edf input stage was removed and replaced with a serial COM port configuration. The exact COM port being used by the device is found in the Device Manager of the computer and set in this section. This also involved setting a 115200 baud rate and 250 Hz sampling frequency, matching the specifications of the device's output. This version required a parser function (parse_serial_line) to recognise the formatted packets that begin with "$", end in ";". These packets contain eight space-separated integer values per sample, including negative microvolt values, correlating to the data of eight channels. Unlike in the prototype, the channels were split, stored, and processed individually. New features were added including a CSV logging step timestamp and record these raw incoming values for offline analysis and debugging. Additionally, a feature to enable or disable each lobe could be set at runtime (in the Mapping>Lobe Toggle section), supporting stimulus-specific experiments.

The filtering system was updated to account for the unprocessed device output. A 4th-order Butterworth bandpass (1.6 Hz – 40 Hz) was applied to each lobe-averaged signal, attenuating any low-frequency drift or power supply frequency artefacts before spectral analysis. This was achieved using the SciPy (scipy) Python package for scientific computing.

This final code still operates on 1-second (250-sample) epochs but is now fed from a continuously filling live buffer (eeg_buffer). It also includes multiple validation systems. For example, before processing, the code verifies the size of the buffer, printing an error if it is not. For each lobe, only channels that are not all zeros are considered "valid". This means that a lobe is skipped entirely when its mapped electrodes deliver no signal, preventing insignificant FFT outputs and then incorrect MIDI messages. For lobes that pass validation and are set to active, the code outputs a lobe representative by averaging valid channels (lobe_signal = np.mean(valid_channels, axis=0)). It then applies the Butterworth band-pass filter (apply_bandpass(lobe_signal, 1.6, 40.0, sfreq)), followed by per-epoch mean removal like before (filtered -= np.mean(filtered)) to remove any very-low-frequency drift present and stabilise the spectral estimates. The features are then extracted per lobe, with the same FFT logic from the prototype, where frequency band mean amplitudes are linearly mapped to 7-bit MIDI values using the fixed scale_factor.

4.1.2. Updated Interfacing with Reaper

Now the MIDI mapping has been expanded from one combined channel signal to four independent lobe streams, each outputting three CC values (Theta, Alpha, Beta) per epoch, for a total of 12 MIDI messages per second. The updated code structure maintains stability in feature processing and for DAW interfacing. The discrete per-lobe CC mapping, found in the initialisation of the code, ensures features are correctly assigned and that the transfer of MIDI data to Reaper/Helgobox ReaLearn is straightforward, also enabling mapping presets to be saved directly within the plugin.

For a validation method, a custom Reaper JSFW plugin (found in Appendix E) was developed to plot the incoming MIDI CC messages on four separate graphs correlating to each lobe (Frontal, Central, Temporal, Occipital). Ideally, each lobe has a scrolling plot showing 30 seconds of data at 1Hz, with labelled traces for Theta, Alpha, and Beta. This required a lot of development; however, due to the time constraints of the project, it never fully reached a workable demo. The GUI of this plugin and a small trace of Theta values for the Frontal lobe can be seen below in Fig. 4. 1. 1.

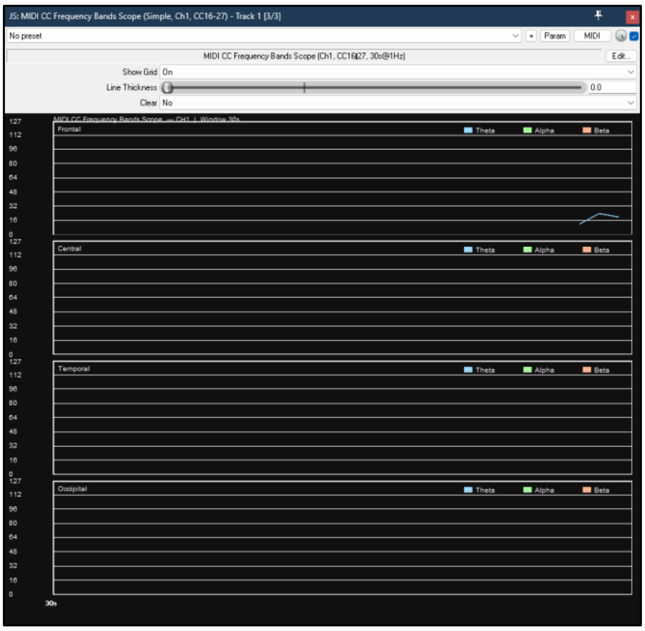

Fig. 4. 1. 1 JSFX Plugin of MIDI CC Frequency Bands Scope

4.1.3. Iterating the Mapping Implementation

The additional CC assignments allowed for experimentation with the musical aspect of the project. This included investigations into sound synthesis methods using the MIDI information, for example, using PureData patches to map to frequency/amplitude modulation ratios. Finally, a routing to the Vital Spectral Warping Wavetable Synth plugin was used for effectively applying the EEG-driven modulation to its synth parameters. For comparison across stimulus experiments, this plugin provided playback of four different MIDI blocks of Bach preludes [34].The use of these standards would provide a musical control across participants and also allow them to experience semi-real-time feedback of parameter modulation from tasks performed. The CC values were mapped to three parameters of four LFOs, each controlling the wave frame of an oscillator playing the MIDI notes. A sine wave oscillator was also used for stability across the four preludes. Across all four mapping presets, the theta values were mapped to the smoothing effect of the LFO, the alpha values to the delay effect, and the beta values to the stereo width. The shape of each of the four LFOs was as follows: triangle, sine, saw up, and saw down to provide variance for the participants.

An overview of the final Reaper session with Helgobox ReaLearn, Vital Wavetable, and the four preludes can be seen below in Fig. 4. 1. 2.

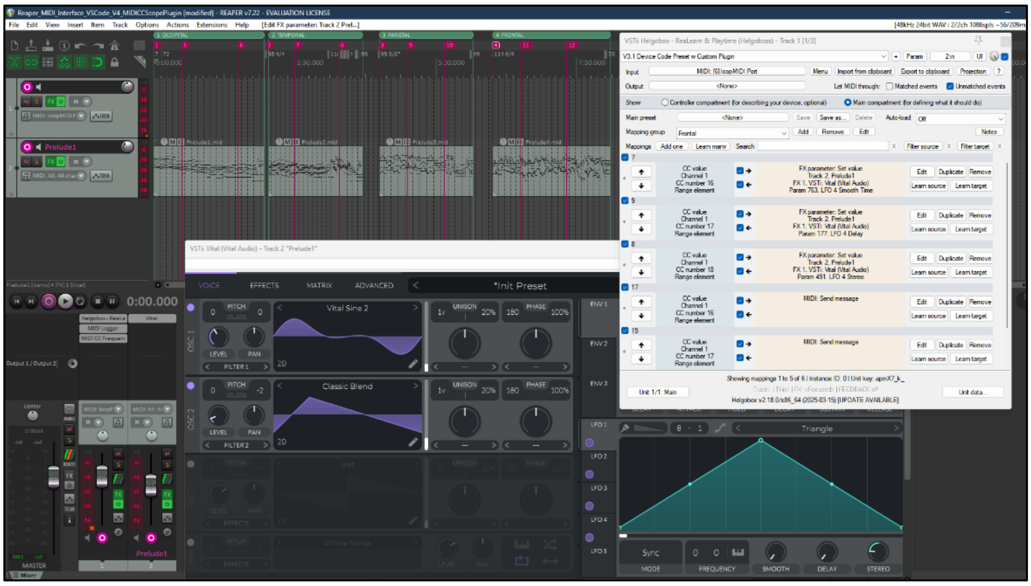

Fig. 4. 1. 2 Reaper Session with ReaLearn, Vital, and Preludes

4.1.4. Testing During Development

Internal tests of this final software structure were run using self-recorded sessions. Some of these can be seen in the video demo found at Appendix G). In this, motor imagery and movement tasks were performed to specifically evoke a decrease in beta activity. While general modulation behaviour was recorded, the 1-second buffer meant that instant correlations between stimulus and audio change were harder to perceive. As mentioned in the literature review, sensory information is not processed in isolation, which means that cross-lobe activity could be invoked in these self-recorded sessions due to necessary interfacing with the DAW. Essentially, visual stimulation in the occipital lobe could affect the desired features in the frontal lobe. Cross-validation against known stimulus events was currently limited by the device's fixed 1 Hz output rate; however, I will discuss other synchronisation possibilities in the next section. With this setting, this led to the development of calibration periods before and after events, to help more broadly estimate this stimulus-modulation. The pink markers seen in Fig. 4. 1. 2, indicate these calibration periods with stop-start-stop positions for each task within each prelude.

Additionally, the CSV files of the raw data generated could be used to further validate the correlation of stimulation to modulation. CSV logs containing timestamp information were fed into a new Python script (found in Appendix I.1) that concatenated lobe-specific trials and plotted these together to visualise activity over time, marking approximate calibration and stimulus intervals. However, for this to effectively check for pattern correspondence, these CSV files were later reprocessed, showing the changes in features over time. This script development was performed after the external stimulus experiments were done in Chapter 5.

4.2. Hardware Implementation

4.2.1. Hardware-Software Synchronisation

As mentioned, with the device's fixed 1-second packet rate, the software loop was aligned to only process this rate of data. The NeuroBell team offered an option to return the device and increase this frequency, which, in theory, would allow for a more detailed tracking of brainwave changes and help validate the extracted features against the stimulus in almost real-time. However, increasing the packet rate would reduce the number of samples per window, lowering the dynamic resolution, making it harder to capture stable frequency-band amplitudes. A higher update speed could also introduce additional hardware-software synchronisation challenges, needing more accurate buffering and timing alignment to avoid misaligned data to MIDI values. In the end, the system's structure had already been developed for 1 Hz processing, and this new change could be explored as future work for the project. The

1-second window preserved band-power estimates and ensured MIDI updates remained consistent and in sync across all lobes.

4.2.2. Challenges and Adjustments

Occasional readings were observed from electrodes close to the reference ports, producing false activity in inactive lobes. In self-testing, O1/O2 proved the most reliable channels. The device offered a soft reset using the power button, and a hard reset was achieved by unplugging the battery within it. Battery checks were performed manually using a multimeter via PCB test points. It was important to avoid external interference with the device during recording sessions, caused by other electrical devices on the desk holding the amplifier. Using the original Serial Port Plotter, this noise was evident on the plots, so it was ensured that the device was placed away from as many of these interferences as possible.

4.2.3. Final Integrated System

By the end of this development, the system consisted of the Luna amplifier streaming eight channels of EEG into the Python script, which processed each lobe independently and output MIDI CC values in real-time to Reaper. The ReaLearn preset routed these to the Vital plugin, modulating effects while the Bach preludes were played. This implementation matched the plan set out in Chapter 3 while adapting to the new constraints of the live hardware environment. This final system can be seen below in Fig. 4. 2. 1, and will provide the basis for the next chapter, Testing and Surveying.

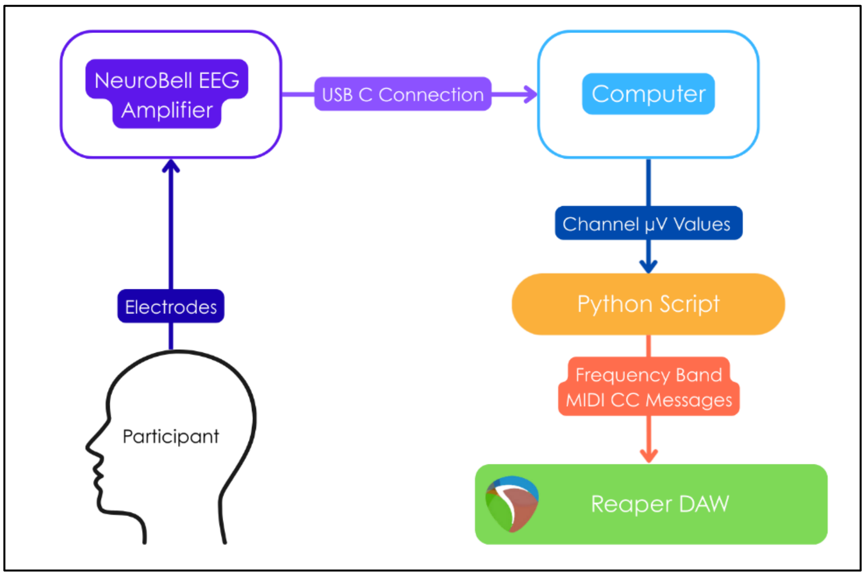

Fig. 4. 2. 1 Block Flow Diagram of Entire System

5. Testing and Surveying

5.1. Introduction to Real-World Testing

The experimental implementation previously discussed was transitioned from personal demos to external participant trials. The internal tests allowed for the evaluation of the EEG-to-MIDI mapping in a controlled environment. These trials allowed for debugging of the hardware interface, refining filter parameters, and confirming the mapping in Reaper. The external participant trials were more structured, following an updated experimental framework method developed from the theoretical stimulus environments in Section 2.4. While the original literature review described a multitude of musical controls and experiments, including additional frequency bands, the real-world tests simplified these into four discrete tasks, one per lobe, each lasting for approximately one minute within each prelude. This provided consistency across trials and made the instructions more straightforward for participants. The participants were asked to listen to each of the four preludes, complete the relevant task during, and fill out the questionnaire following each of them. This questionnaire can be found in Appendix H), and a video demo from the sessions can be seen in Appendix G).

5.2. Updated Stimulus Framework

5.2.1. Rationale Behind Each Task & EEG Target Bands

The following table discusses the lobe and region being targeted for each task, the stimulus carried out, the target EEG frequency band invoked, and a recall of the rationale behind the choice. Following this, a recount of the actual sessions will be discussed.

Table 5. 1 Finalised Stimulus Tasks

Brain Region	Stimulus Task	Target Band	Scientific Basis
Occipital Lobe (Visual Cortex)	9 Hz flicker stimulus	Alpha (8–12 Hz)	SSVEP show alpha-band resonance in the visual cortex when the flicker frequency aligns with the occipital alpha rhythm [18].

Brain Region	Stimulus Task	Target Band	Scientific Basis
Temporal Lobe (Auditory/Medial Temporal)	Mental recall of a pre-learned number list	Theta (4–7 Hz)	Theta rhythms emerge during memory retrieval; increased theta power is linked to hippocampal and temporal processing [22][35].
Parietal Lobe (Sensorimotor Cortex)	Mental rotation of a 3D object	Alpha (8–12 Hz)	Spatial tasks like mental rotation engage the superior parietal lobule and reduce parietal alpha rhythms [36].
Frontal Lobe (Prefrontal Cortex)	Imagined hand movement	Theta (4–7 Hz), Beta (13–30 Hz)	Frontal-midline theta increases with working memory and attention load; Motor imagery should induce beta desynchronisation (beta decrease), signalling motor engagement followed by a beta rise when it is ended[25][37].

5.2.2. Occipital Lobe Task

Firstly, the participants would experience a visual flicker stimulus. A MATLAB script was developed to generate this visual flicker at a frequency of 8 Hz. This should cause alpha-band resonance in the lobe, leading to a strong presence of alpha band power compared to calibration levels. For calibration levels, participants were instructed to keep their eyes open, as removing the visual stimulus leads to a rise in alpha activity.

However, given that the script was being run via the Matlab web-based version of the software, unfortunately, on the day that sessions were carried out, this script repeatedly failed to produce the flashing plot. This wasn’t an issue with the script as it is quite a limited process, but rather it was an issue with the online software. Two sessions were carried out, collecting data from the following three tasks.

5.2.3. Temporal Lobe Task

Next, participants performed a memory recall task to stimulate theta activity. This involved mentally recalling their phone numbers along with four numbers from their credit card. This was a reliable choice of numbers as it's assumed people in general already have a stored memory of these.

5.2.4. Parietal Lobe Task

The parietal lobe data was processed using the "Central" lobe notation in the code (as this was based on the device labelling). For this lobe, participants engaged in a spatial imagery task. This task combined elements from the literature's motor imagery and spatial rotation concepts, but it was implemented purely as mental 3D object rotation to simplify instructions and avoid muscle artefacts.

5.2.5. Frontal Lobe

Finally, participants undertook a motor imagery task. Users were instructed to imagine hand movement, targeting frontal beta rhythms associated with motor planning. This should theoretically invoke a beta decrease before returning to the beta calibration level.

5.3. Procedure & Environment

Testing was conducted in a quiet environment, a reserved lecture theatre in CSM, to minimise external distractions. The NeuroBell Luna device was placed on a table away from the laptop being used or any other equipment in the room. The electrode leads were applied with conductive gel and placed on the participants according to the pre-established methods. The electrode contact quality was checked before starting, ensuring as full contact with skin as possible. Each participant first completed the ethics agreement of the questionnaire. They then completed all four tasks in the order specified by the survey. After each task, they completed the Likert-scale and open-ended questions from the questionnaire, focusing on perceived musical changes and task difficulty.

During the trials, the Python script checked that the lobe being tested was enabled (active_lobes dictionary). To aid in the isolation of musical control, when a lobe wasn't being used in the task, it was disabled in this dictionary before beginning. This also allowed for all incoming EEG data per lobe to be saved to a CSV file per task for later analysis.

5.4. Data Collected

The outputs of the offline CSV file validation scripts previously discussed can be seen here.

5.4.1. Objective Data

In Fig. 5. 4. 1, the first script (found in Appendix I.1) used to concatenate the CSV files and plot the raw data is visible, where frontal, parietal and temporal sessions were performed.

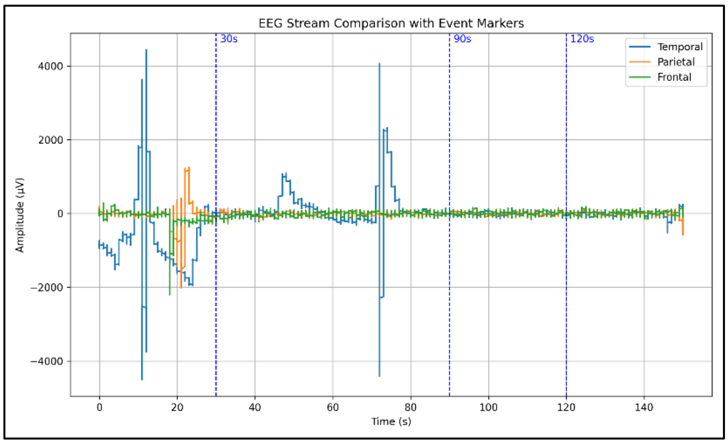

Fig. 5. 4. 1 EEG CSV Comparison of Lobes with Event Markers

However, for actual feature verification, the second script developed (found in I.2) was used instead to produce the plots in Fig. 5. 4. 2. These scripts can provide the basis for further comparative studies; currently, there are not enough sample datasets where participants underwent the exact trials previously outlined.

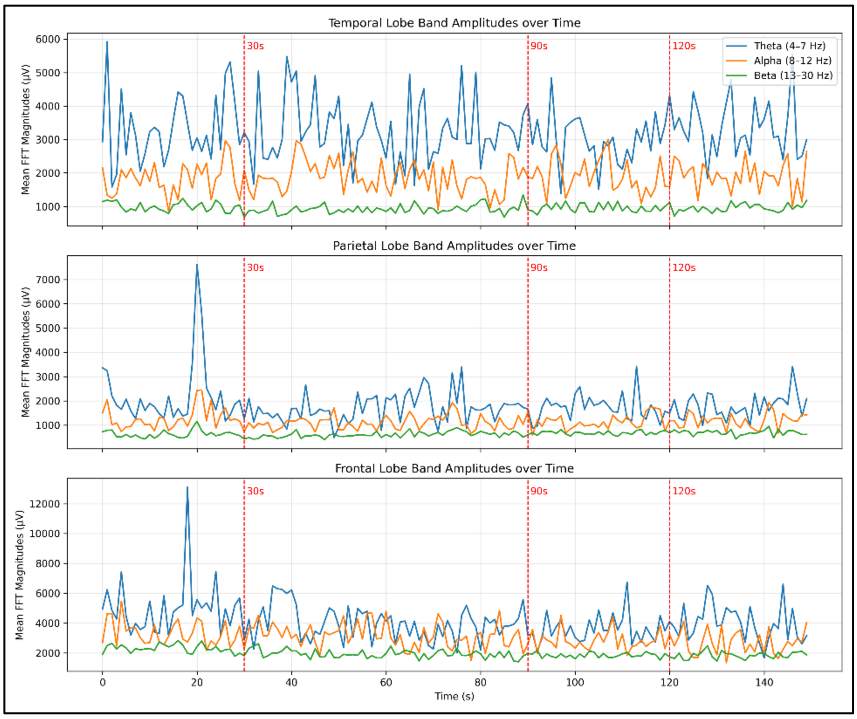

Fig. 5. 4. 2 Feature Comparison across Lobes with Event Markers

5.4.2. Subjective Data

The participant feedback from the post-task questionnaires indicated that perceived changes in the music were inconsistent across the four tasks. This is of course a minimal sample size, however, it did provide information on how the project should be developed further in the future. Given that the participants didn't undergo the occipital task, this data was excluded when reviewing it.

While some subtle alterations in timbre and pitch were reported during the memory recall task, there was also frequent feedback of no noticeable difference, especially during spatial imagery and motor imagery tasks. In certain cases, participants noted that the cognitive effort required for the task drew attention away from the music, which may have reduced their ability to perceive modulation. Overall, participants' awareness of modulation seemed to depend on both the nature of the task and their ability to maintain focus on the auditory output while completing it.

5.5. Summary of Findings

This testing and survey phase demonstrated that the EEG-to-MIDI mapping system can elicit subjective perceptions of musical change during targeted cognitive tasks, though the effect was not consistently perceived by all participants. Tasks with more direct sensory input, could lead to more consistent reports of change. The system's 1 Hz update rate was sufficient for slow, evolving timbral modulation, but limited its potential for fast, rhythm-synchronised effects. These findings suggest that while the system shows promise as an interactive brain–music interface, further refinements in stimulus design, feedback precision, and system responsiveness would improve user perception and control in future experiments.

6. Discussion

6.1. Review of Objectives

The aim of this research was to develop and evaluate a system that could map EEG signals to musical parameters within a DAW in real time. The first objective was successfully met, achieving stable EEG-to-MIDI translation using the NeuroBell Luna Amplifier. The final system streamed eight EEG channels, processed frequency-band features per lobe, and transmitted up to 12 MIDI Control Change messages per second into Reaper, where they modulated the Vital Wavetable Synth parameters.

The second objective, assessing whether users could meaningfully influence music through cognitive tasks, was partially achieved. While some participants reported clear changes in timbre or effects during certain tasks (particularly memory recall), feedback was inconsistent across lobes and individuals. This meant that the technical pipeline was functional, but the musical modulation was not precise enough for user perception.

6.2. Interpretation of Key Findings

The temporal lobe memory recall task provided the clearest subjective reports of change, with some participants describing perceived increases in tempo or subtle timbral shifts during the task. Ideally, this aligns with research linking hippocampal and medial temporal theta increases to active memory retrieval. Following this task, through discussions, subjects also noted that this perceived change could also have been a false correlation, in that the music was changing regardless of the stimulus. Through analysis of Fig. 5. 4. 2, the theta amplitudes on the Frontal Lobe plot were predicted to increase, however, the values in the calibration periods (0s-30s, 90s-120s) are not objectively lower than those during the task.

As mentioned, the parietal lobe spatial imagery (mental rotation) task produced minimal reported differences, despite literature suggesting alpha suppression over the superior parietal lobule during such tasks. Looking at Fig. 5. 4. 2 again, the alpha amplitudes on the Parietal Lobe subplot were predicted to decrease. The values can be seen to have slightly decreased from the initial calibration period (0s-30s) to when the task began.

Finally, the frontal lobe motor imagery task aimed to evoke beta desynchronisation (decrease) followed by a rise in values. In practice, participants reported little or no change, again suggesting that either the 1 Hz data rate lacked the temporal resolution to track beta fluctuations in real time, or that the mapping to stereo width was less perceptually obvious. As well as this, looking at the change of beta (μV) values on Fig. 5. 4. 2, the change in values across periods is hard to decipher.

In total, the mappings from frequency band amplitudes to effect parameters may have been too nuanced compared to, for example, timbral controls, potentially diluting the cause-and-effect relation for participants. In order for a more accurate comparison of values across periods, I will discuss another possible method of processing and analysis in the next chapter, Chapter 5.

6.3. Mapping Efficacy

The one-to-one mapping of band-specific amplitudes to distinct MIDI parameters provided technical stability and conceptual simplicity, consistent with early BCMI design recommendations. However, while simple mappings are easier for participants to grasp, some mappings in this study were not inherently intuitive or noticeable. Literature suggests, and confirmed by this project's testing, that mappings to parameters with significant, immediate timbral impact (e.g., filter cutoff, reverb wet/dry) can improve perception of control.

6.4. Technical Performance

The NeuroBell Luna provided stable data capture at 1 Hz updates, which suited slow, evolving timbral changes but limited the possibility for fast, rhythm-synchronised effects. The fixed 1-second window preserved band-power stability but may have missed transient events. The Butterworth bandpass filtering effectively reduced drift and power supply artefacts.

6.5. Comparison with Literature

This project somewhat reinforces the findings of the studies in [13][14], in showing that even low-channel, low-cost EEG hardware can be used to achieve meaningful musical control, although with reduced precision compared to their advanced and validated systems. The mixed perceptual results are also consistent in a way with early BCMI prototypes of [15][16], where participants with varying knowledge of musical theory produce varying changes due to this background.

6.6. Limitations

The limitations in this study form the interpretation of its results. The small sample size of only two external participants limits comparisons and hence statistical certainty. As mentioned, the absence of the occipital flicker task during these external trials, due to technical issues, resulted in a lack of comparison to any internal observations. The short stimulant times, where each task lasted for roughly 1 minute, could be seen as insufficient in comparison to the literature. Finally, the fixed scaling mapping could be suppressing significant features in the system.

6.7. Significance of Findings

Even with its limitations, the system showed that it's possible to integrate a portable EEG amplifier with a DAW for real-time music control based on specific brain lobes. This opens up opportunities for creating accessible, affordable, and non-invasive BCMIs. It also has potential applications in music production, whether for live performances, composition, or neurofeedback-driven music therapy.

6.8. Creative and Technological Implications

The integration of EEG signals into musical environments allows for both creative expression and technological innovation. This work shifts the role of the brain in music from passive reception to active, expressive agency. For composers and live performers, this allows for a unique form of musical control, where expressive nuance can be influenced by cognitive or emotional states in real time. For users with physical impairments, systems like this offer a non-mechanical instrument, expanding the inclusivity of musical expression. Such systems could redefine performance by introducing these personalised, adaptive elements that respond to the performer's internal state rather than just physical gestures.

Technologically, low-cost EEG hardware coupled with robust DAW integration lowers the barrier for BCMI adoption outside research labs by reducing the financial and technical barriers that have previously limited BCMI use.

Looking forward, the potential for emotion-responsive sound design is significant. Appropriate signal filtering and machine learning have provided adaptive musical systems that can respond to listener arousal, engagement, or emotional shifts. This could inform a broad range of applications such as music therapy, dynamic film scoring, or immersive gaming experiences that respond to user states in real time.

However, as this study suggests, challenges remain. EEG susceptibility to artefacts demands careful setup and filtering, inter-participant variability could complicate generalisation, and ethical considerations around brain-data privacy must be addressed and upheld. As BCMI technology becomes more mainstream, clear consent protocols and user control over data will be essential.

Ultimately, the system presented in this study offers a functional, adaptable platform. A foundation from which stronger, more responsive, and more accessible brain-controlled musical experiences can be built.

7. Future Work

Building on the final system established and the insights gained during development and testing, several technical, methodological, and creative extensions could be proposed to advance the system's performance, usability, and relevance to both artistic and research contexts.

7.1. Technical Enhancements to EEG Processing

While the current implementation calculates the mean FFT magnitudes for each band (theta, alpha, beta) in real time, the outputs were not normalised across bands. This preserved absolute signal strength for MIDI mapping but limited comparability across participants and sessions. Future work should implement relative power normalisation, for example, by dividing each band's power by the total three-band (4–30 Hz) power to produce a percentage, similar to the approach seen in Fig. 3. 1. 2. This approach, commonly used in EEG research, mitigates inter-individual variability caused by factors such as skull thickness, electrode impedance, or amplifier gain. Using ratios such as alpha/beta could also produce more stable and meaningful mappings, potentially revealing stronger correlations between stimulus and musical change.

Additionally, FFT processing could be improved by using power-based estimators (e.g., magnitude squared or log-power in dB) for better spectral stability. The use of a spectral window function could reduce FFT leakage and improve frequency resolution. These refinements would provide more robust and comparable datasets and could strengthen both quantitative analysis and creative reliability in performance settings.

7.2. Cross-Disciplinary Collaboration

During discussions with Dr. Mark O'Sullivan at NeuroBell, the potential for collaboration with medical professionals was introduced, including exploratory conversations with clinical neonatologist Dr. Elisabeth Kooi in Groningen, Norway. Together, they highlighted the potential for music therapy applications of this project for neonates in the NICU. Their clinical expertise, academic backgrounds, and musical insights aligned perfectly with this project, making their suggestions for the future of this system invaluable. With Dr. O'Sullivan's Master of Science degree in Music & Technology, paired with Dr. Kooi's studies into neonatal music therapy and its effect on heart rate and cerebral oxygenation [38], future phase could involve developing the current system into a feedback loop, enabling a baby's brain activity to "control" the music. Although this neonatal music therapy research requires further testing and validation, this avenue of experimentation could provide an important additional dimension to the evaluation.

7.3. Additional Biometric Inputs

In addition to EEG, integrating other biometric signals, such as heart rate variability or respiration, could create richer and more responsive control systems. These signals could be mapped in parallel or combined with EEG features to create multi-dimensional control, opening possibilities for more nuanced modulation of musical parameters or cross-modal artistic expression.

7.4. Validation of Stimulus & Expansion of Musical Control Strategies

A key limitation in the present study was the lack of statistical validation linking stimulus events to perceived or measured musical changes. Future work should implement further cross-validation frameworks, using the timestamped stimulus logs and automated feature extraction scripts developed, to confirm whether observed spectral variance reliably corresponds to the intended task conditions. This would help distinguish genuine brain-driven modulation from accidental variation.

Currently, EEG features modulate pre-set Vital Synth parameters applied to a fixed MIDI sequence. Future studies could explore alternative strategies mentioned, involving sound synthesis techniques such as FM modulation or granular synthesis. Another development could involve replaying a standard recorded piece but altering its playback parameters in real time based on the participant's EEG, enabling direct comparison between an unaltered original and its brain-driven adaptation. Finally, expanding the range of musical parameters under control, from timbral qualities to generative composition processes, allows the EEG signals to instead influence melody, harmony, or rhythmic structure.

Further research could explore whether particular genres or musical textures elicit distinctive responses in specific lobes. Short excerpts from varied genres could be played while logging EEG, with analysis focused on frequency-band changes across lobes. This could reveal inter-participant variance and genre-specific neural markers, offering both scientific insight and guidance for optimising mappings for participants; what features are strongly invoked and when, so they can be used effectively as modulators of DAW VST parameters.

7.5. Outlook

By implementing these refinements, the system could evolve from a functional basis into a more scientifically rigorous and musically versatile tool. Improved normalisation and validation would enhance reliability, broader biometric integration could expand expressivity, and diversifying musical applications would increase creative potential. In the long term, these developments could position the platform for use in live performance, therapeutic interventions, and experimental research, contributing to the growing accessibility and creative reach of brain-computer music interfaces.

8. Conclusion

This project set out to investigate whether EEG signals, captured from distinct brain lobes during targeted sensory and cognitive tasks, could be mapped in real time to control VST parameters in a DAW. A fully functioning pipeline was developed, integrating NeuroBell's Luna EEG Amplifier, real-time Python feature extraction, MIDI mapping, and Reaper-based modulation, which successfully translated live neural data into continuous musical control signals. The system met this first objective: stable, lobe-specific EEG-to-MIDI conversion.
The second objective, enabling participants to meaningfully influence music through thought-based tasks, was partially achieved. Memory recall in the temporal lobe currently produces the clearest subjective reports of musical change. However, effects were less consistently perceived in spatial imagery and motor imagery tasks, suggesting that mapping strategies and perceptual strength require refinement.
Key contributions of this work include demonstrating that a portable, low-cost EEG device can be integrated into a music production environment for brain–music interaction, and that this system can be adapted for both artistic and research applications. Limitations, such as the current fixed update rate, small participant pool, and absence of the occipital task in trials, reflect the need for more testing and validation to be carried out.

This research highlights EEG-driven music control as both a creative tool and a potential therapeutic interface. Future work could involve adaptive, machine learning based mapping, multimodal biometric integration, and exploration into genre or emotion-specific mappings.
As the field of brain–computer music interfaces continues to grow, it offers the potential for performance inclusivity and innovation across neuroscience, technology, and art.

References

[1] L. Bellier *et al.*, “Music can be reconstructed from human auditory cortex activity using nonlinear decoding models,” *PLOS Biology*, vol. 21, no. 8, pp. 1–27, Aug. 2023, doi: 10.1371/journal.pbio.3002176.

[2] M. H. Thaut, P. D. Trimarchi, and L. M. Parsons, “Human brain basis of musical rhythm perception: common and distinct neural substrates for meter, tempo, and pattern.,” *Brain Sci*, vol. 4, no. 2, pp. 428–452, June 2014, doi: 10.3390/brainsci4020428.

[3] S. I. Giraldo and R. Ramírez, “Brain-Activity-Driven Real-Time Music Emotive Control,” 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:51762634

[4] D. Fahim, J. Grewal, and R. Ellendula, *A Novel Approach to Image EEG Sleep Data for Improving Quality of Life in Patients Suffering From Brain Injuries Using DreamDiffusion*. 2024. doi: 10.48550/arXiv.2407.02673.

[5] J. Malmivuo and R. Plonsey, “Bioelectromagnetism. 13. Electroencephalography,” 1995, pp. 247–264.

[6] W. Wei *et al.*, “A function-based mapping of sensory integration along the cortical hierarchy.,” *Commun Biol*, vol. 7, no. 1, p. 1593, Nov. 2024, doi: 10.1038/s42003-024-07224-z.

[7] V. Alluri, P. Toiviainen, I. P. Jääskeläinen, E. Glerean, M. Sams, and E. Brattico, “Large-scale brain networks emerge from dynamic processing of musical timbre, key and rhythm.,” *Neuroimage*, vol. 59, no. 4, pp. 3677–3689, Feb. 2012, doi: 10.1016/j.neuroimage.2011.11.019.

[8] D. L. Harrington, K. Y. Haaland, and R. T. Knight, “Cortical networks underlying mechanisms of time perception.,” *J Neurosci*, vol. 18, no. 3, pp. 1085–1095, Feb. 1998, doi: 10.1523/JNEUROSCI.18-03-01085.1998.

[9] E. B. Wencil, P. Radoeva, and A. Chatterjee, “Size Isn’t All that Matters: Noticing Differences in Size and Temporal Order.,” *Front Hum Neurosci*, vol. 4, p. 171, 2010, doi: 10.3389/fnhum.2010.00171.

[10] F. Tecchio, C. Salustri, M. H. Thaut, P. Pasqualetti, and P. M. Rossini, “Conscious and preconscious adaptation to rhythmic auditory stimuli: a magnetoencephalographic study of human brain responses.,” *Exp Brain Res*, vol. 135, no. 2, pp. 222–230, Nov. 2000, doi: 10.1007/s002210000507.

[11] T. A. Deuel, J. Pampin, J. Sundstrom, and F. Darvas, “The Encephalophone: A Novel Musical Biofeedback Device using Conscious Control of Electroencephalogram (EEG).,” *Front Hum Neurosci*, vol. 11, p. 213, 2017, doi: 10.3389/fnhum.2017.00213.

[12] J. Eaton, D. Williams, and E. Miranda, “The Space Between Us : Evaluating a multi-user affective brain-computer music interface,” *Brain-Computer Interfaces*, vol. 2, pp. 103–116, Apr. 2015, doi: 10.1080/2326263X.2015.1101922.

[13] E. Miranda, “Brain-Computer Music Interface for Generative Music”.

[14] E. Miranda and V. Soucaret, “Mix-It-Yourself with a Brain-Computer Music Interface,” Jan. 2008.

[15] E. Miranda, N.-F. Wilson, R. Palaniappan, J. Eaton, and W. Magee, “Brain-Computer Music Interfacing (BCMI) From Basic Research to the Real World of Special Needs,” *Music and Medicine*, vol. 3, pp. 134–140, July 2011, doi: 10.1177/1943862111399290.

[16] S. Venkatesh, E. R. Miranda, and E. Braund, “SSVEP-based brain-computer interface for music using a low-density EEG system.,” *Assist Technol*, vol. 35, no. 5, pp. 378–388, Sept. 2023, doi: 10.1080/10400435.2022.2084182.

[17] "NeuroBell Luna Amplifier," NeuroBell Tech. [Online]. Available: https://www.neurobell.com/tech
[18] T. A. de Graaf, J. Gross, G. Paterson, T. Rusch, A. T. Sack, and G. Thut, "Alpha-band rhythms in visual task performance: phase-locking by rhythmic sensory stimulation.," *PLoS One*, vol. 8, no. 3, p. e60035, 2013, doi: 10.1371/journal.pone.0060035.
[19] T. Sollfrank *et al.*, "The Effects of Dynamic and Static Emotional Facial Expressions of Humans and Their Avatars on the EEG: An ERP and ERD/ERS Study.," *Front Neurosci*, vol. 15, p. 651044, 2021, doi: 10.3389/fnins.2021.651044.
[20] D. Valentine, "Learning EEG - Chewing & Hypoglossal Movement." [Online]. Available: https://www.learningeeg.com/artifacts#:~:text=CHEWING%20%26%20HYPOGLOSSAL%20MOVEMENT
[21] M. C. Cervenka, S. Nagle, and D. Boatman-Reich, "Cortical high-gamma responses in auditory processing.," *Am J Audiol*, vol. 20, no. 2, pp. 171–180, Dec. 2011, doi: 10.1044/1059-0889(2011/10-0036).
[22] E. Tan *et al.*, "Theta activity and cognitive functioning: Integrating evidence from resting-state and task-related developmental electroencephalography (EEG) research.," *Dev Cogn Neurosci*, vol. 67, p. 101404, June 2024, doi: 10.1016/j.dcn.2024.101404.
[23] J. O'Kelly, L. James, R. Palaniappan, J. Fachner, J. Taborin, and W. L. Magee, "Neurophysiological and Behavioral Responses to Music Therapy in Vegetative and Minimally Conscious States," *Frontiers in Human Neuroscience*, vol. Volume 7-2013, 2013, doi: 10.3389/fnhum.2013.00884.
[24] F. Abdullahi *et al.*, "Beta and Gamma EEG Oscillatory Waves of the Frontal Cortex Increase After Wet Cupping Therapy in Healthy Humans," *Journal of Research in Medical and Dental Science*, vol. 7, pp. 123–130, Jan. 2019.
[25] M. Assem *et al.*, "High gamma activity distinguishes frontal cognitive control regions from adjacent cortical networks.," *Cortex*, vol. 159, pp. 286–298, Feb. 2023, doi: 10.1016/j.cortex.2022.12.007.
[26] M. Roshanaei, H. Norouzi, J. Onton, S. Makeig, and A. Mohammadi, "EEG-based functional and effective connectivity patterns during emotional episodes using graph theoretical analysis," *Scientific Reports*, vol. 15, no. 1, p. 2174, Jan. 2025, doi: 10.1038/s41598-025-86040-9.
[27] A. Gallina and D. Mytilinaios, "Kenhub - Cerebrum Parietal Lobe," Kenhub. [Online]. Available: https://www.kenhub.com/en/library/anatomy/parietal-lobe#:~:text=for%20blood%20drainage.-,Functions,-The%20parietal%20lobe
[28] H. M. Hobson and D. V. M. Bishop, "Mu suppression - A good measure of the human mirror neuron system?," *Cortex*, vol. 82, pp. 290–310, Sept. 2016, doi: 10.1016/j.cortex.2016.03.019.
[29] D. Jenson, A. L. Bowers, D. Hudock, and T. Saltuklaroglu, "The Application of EEG Mu Rhythm Measures to Neurophysiological Research in Stuttering," *Frontiers in Human Neuroscience*, vol. Volume 13-2019, 2020, doi: 10.3389/fnhum.2019.00458.
[30] M. de Jesús Sánchez Gama, L. Alberto Barradas Chacón, and L. Chacón Gutiérrez, "OpenNeuro Dataset ds005420.v1.0.0 (Resting state EEG with closed eyes and open eyes in females from 60 to 80 years old)." 2024. doi: 10.18112/openneuro.ds005420.v1.0.0.
[31] Dr. Yin Fen Low, "How to decide on the right EEG spectral analysis method in BrainVision Analyzer," Brain Products. [Online]. Available: pressrelease.brainproducts.com/spectral-analysis-methods/#:~:text=long%20continuous%20data.-,FFT,-One%20crucial%20assumption

[32] B. Klum and A. Valdez, "Helgobox: ReaLearn & Playtime." helgoboss / helgobox, github, July 09, 2025. [Online]. Available: github.com/helgoboss/helgobox?tab=readme-ov-file#installation
[33] S. Benoit, "Serial Port Plotter." CieNTi / serial_port_plotter, github, Sept. 07, 2018. [Online]. Available: github.com/CieNTi/serial_port_plotter
[34] "Complete Bach Midi Index," Feb. 10, 2018. [Online]. Available: https://www.bachcentral.com/midiindexcomplete.html
[35] P. B. Sederberg, M. J. Kahana, M. W. Howard, E. J. Donner, and J. R. Madsen, "Theta and gamma oscillations during encoding predict subsequent recall.," *J Neurosci*, vol. 23, no. 34, pp. 10809–10814, Nov. 2003, doi: 10.1523/JNEUROSCI.23-34-10809.2003.
[36] C. M. Michel, L. Kaufman, and S. J. Williamson, "Duration of EEG and MEG α Suppression Increases with Angle in a Mental Rotation Task.," *J Cogn Neurosci*, vol. 6, no. 2, pp. 139–150, Spring 1994, doi: 10.1162/jocn.1994.6.2.139.
[37] S. Itthipuripat, J. R. Wessel, and A. R. Aron, "Frontal theta is a signature of successful working memory manipulation.," *Exp Brain Res*, vol. 224, no. 2, pp. 255–262, Jan. 2013, doi: 10.1007/s00221-012-3305-3.
[38] {Nienke H.} {van Dokkum} *et al.*, "Neonatal music therapy and cerebral oxygenation in extremely and very preterm infants: A pilot study," *Music and Medicine*, vol. 13, no. 2, pp. 91–98, Apr. 2021, doi: 10.47513/mmd.v13i2.813.

Appendices

Appendix A) Lists

A.1) Abbreviations Used

EEG	–	Electroencephalogram
VST	–	Virtual Studio Technology
DAW	–	Digital Audio Workstation
MEG	–	Magnetoencephalography
PDR	–	Posterior Dominant Rhythm
BCMI	–	Brain-Computer Musical Interfaces
ICCMR	–	Interdisciplinary Centre for Computer Music Research
SSVEP	–	Steady-State Visually Evoked Potentials
NICU	–	Neonatal Intensive Care Unit
MIDI	–	Musical Instrument Digital Interface
OSC	–	Open Sound Control
EDF	–	European Data Format
V	–	Volts
µV	–	microvolts (1×10^{-6} V)
FFT	–	Fast Fourier Transform
VS Code	–	Visual Studio Code

A.2) List of Figures

A.3) List of Tables

Mapping Cognitive Control: EEG Signals Modulating Real-Time Music Composition

Aisling Moloney
MSc. *Music & Technology*
MTU, Cork School of Music
Cork, Ireland
aisling.moloney@mymtu.ie

***Abstract* — This thesis explores the use of real-time electroencephalogram (EEG) signals to control Virtual Studio Technology (VST) parameters within a Digital Audio Workstation (DAW), using NeuroBell's portable Luna EEG Amplifier. Four stimulus environments were designed to engage each of the brain's primary lobes (frontal, parietal, temporal, occipital), targeting neural oscillatory patterns. EEG signals were processed in real time, extracting theta, alpha, and beta features per lobe, mapped to MIDI Control Change messages, and routed into Reaper for modulation of a wavetable synthesiser. Results showed stable EEG-to-MIDI translation and real-time modulation, though perceptual feedback varied. This demonstrates a viable, low-cost brain–music interface with potential creative and technological applications.**

***Keywords* — EEG, Python, Brain–Computer Music Interface (BCMI), DAW, VST, MIDI mapping, real-time signal processing, music technology.**

Abbreviations

SSVEP	–	Steady-State Visually Evoked Potentials
NICU	–	Neonatal Intensive Care Unit
MIDI	–	Musical Instrument Digital Interface
EDF	–	European Data Format
FFT	–	Fast Fourier Transform

I. Introduction

Many studies have investigated the effect of music on the brain; however, there is an emerging field using brain activity to control music instead. Specific sections of the brain are responsible for various sensory processing functions. This could be harnessed and analysed using electroencephalogram (EEG) technology in a musical context.

Previous studies have shown that certain parts of the brain are responsible for auditory processing of music and that this electrical activity can be reprocessed back into the song being heard [1] Certain parts of the brain can be attributed to processing specific musical elements, activating many of the same regions that speech does [2]. Emotional state can also be used via EEG data to trigger and control musical transformations [3]

This study investigates using brain activity to control VST parameters within a DAW, providing an interactive musical experience using real-time activity from four brain regions. EEG technology was used to record brain activity from the scalp and feed it directly into a DAW via a USB connection. The hypothesis is that sensory-specific lobes can distinctly influence VST control and musical expression.

II. Background & Design

A. Background & Theoretical Framework

EEG measures electrical activity across the scalp via standardised 10–20 electrode placement, with each site corresponding to underlying brain regions [4].

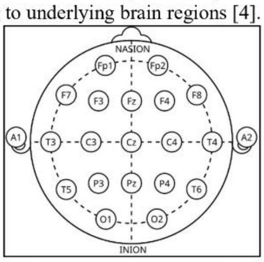

Fig. 1 - Electrode locations of the 10-20 system for EEG recording

For the oscillatory patterns, alpha waves (8–12 Hz) are linked to visual and attentional processing, theta waves (4–8 Hz) to memory, and beta waves (13–30 Hz) are linked to cognitive activity as well as motor planning (especially in the frontal lobe).

B. EEG in Music Technology

Music perception engages distributed brain networks, and EEGs have been used for real-time musical control in systems such as the Encephalophone [5], ICCMR's EEG-driven musical style switching [6], and accessibility-focused brain–computer music interfaces (BCMIs) [7]. These systems demonstrate that low-channel EEG can meaningfully drive synthesis parameters.

C. Technology Overview

The NeuroBell Luna EEG Amplifier, developed for neonatal seizure monitoring, streams 8-channel EEG either wirelessly or via USB [8]. Its portable design and electrode layout (Pz, Cz, F3, F4, C3, C4, T3, T4, O1, O2) allowed for mapping of activity across lobes to DAW parameters.

D. Experimental Framework

This study's framework involves four stimulus environments. The finalised stimulus tasks performed can be seen below in 0This discusses the primary lobe and region being targeted for each task, the stimulus carried out, the target EEG frequency band invoked and mapped to MIDI control in Reaper, and a summary of the rationale behind the choice.

TABLE I. FINALISED STIMULUS TASKS

Brain Region	Stimulus Task	Target Band	Scientific Basis
Occipital Lobe (Visual Cortex)	9 Hz flicker stimulus	Alpha (8–12 Hz)	SSVEP show alpha-band resonance in the visual cortex when the flicker frequency aligns with the occipital alpha rhythm [9].
Temporal Lobe (Auditory/ Medial Temporal)	Mental recall of a pre-learned number list	Theta (4–7 Hz)	Theta rhythms emerge during memory retrieval; increased theta power is linked to hippocampal and temporal processing [10][11].
Parietal Lobe (Sensorimotor Cortex)	Mental rotation of a 3D object	Alpha (8–12 Hz)	Spatial tasks like mental rotation engage the superior parietal lobule and reduce parietal alpha rhythms [12].
Frontal Lobe (Prefrontal Cortex)	Imagined hand movement	Theta (4–7 Hz), Beta (13–30 Hz)	Frontal-midline theta increases with working memory. Motor imagery should induce beta desynchronisation (beta decrease), signalling motor engagement followed by a beta rise when it is ended[13][14].

E. *Final System*

For clarity of the system structure, Fig. 2 shows a block flow diagram of the final desired setup. The following section will discuss how it was implemented.

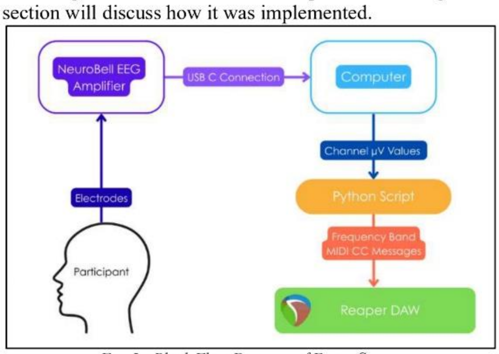

Fig. 2 - Block Flow Diagram of Entire System

III. IMPLEMENTATION

A prototype EEG-to-MIDI system was initially built using sample EDF datasets to simulate a live feed of EEG data, verify the structure and perform initial experiments. This Python script was then adapted for live use with NeuroBell's Luna EEG Amplifier. Live streaming replaced offline file input, with USB serial communication configured at 115200 baud and 250 Hz sampling. Data packets containing eight microvolt channels were parsed, validated, and stored per lobe.

A 4th-order Butterworth bandpass filter (1.6–40 Hz) removed low-frequency drift and electrical power supply artefacts. Each lobe's valid channels were averaged, mean-removed, and transformed via FFT to extract theta (4–7 Hz), alpha (8–12 Hz), and beta (13–30 Hz) amplitudes. These were linearly scaled into 0–127 MIDI Control Change (CC) values and assigned fixed CC numbers for DAW mapping.

Four independent lobe streams (12 CC messages per second) were routed to Reaper's Helgobox ReaLearn plugin, mapped to parameters in the Vital wavetable synthesiser. Bach preludes provided a standardised musical base, with CC values modulating LFO smoothing, delay, and stereo width. Fig. 3 shows this setup.

Fig. 3 - Reaper Session with ReaLearn, Vital, and Preludes

The final system consisted of the Luna device, Python processing, MIDI mapping, and Reaper-based modulation in a fully integrated, lobe-specific control chain, as per Fig. 2.

IV. TESTING & RESULTS

External trials were conducted in a quiet lecture theatre to minimise noise and distractions. Participants wore the Luna EEG device, with electrode contact optimised using conductive gel. Four lobe-specific tasks, each approximately 1 minute long, were completed in sequence, with inactive lobes disabled in software to isolate control.

Objective data was captured as CSV logs of raw EEG and feature-extracted values. Custom scripts plotted feature trends against calibration periods (0s-30s, 90s-120s) and stimulus (30s-90s) markers. An example of this can be seen below in Fig. 4.

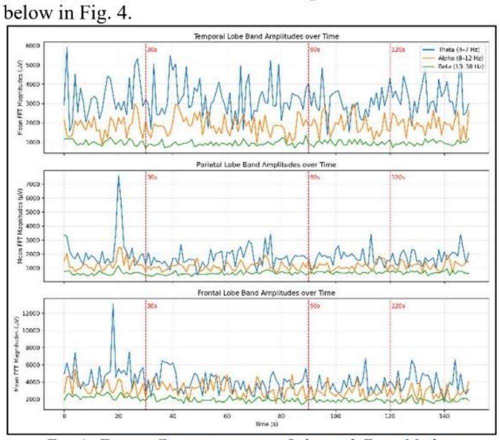

Fig. 4 - Feature Comparison across Lobes with Event Markers

Subjective data was gathered via Likert-scale and open-ended questionnaires after each task.

Results showed an inconsistent perception of modulation. Memory recall tasks (temporal lobe) produced the most frequent reports of subtle timbral change. Parietal and frontal tasks yielded minimal perceived effects, with some participants noting that cognitive effort reduced attention to the music.

The 1 Hz system update rate was sufficient for slow timbral changes but limited for fast, rhythm-linked effects. Findings indicated functional lobe-specific mapping, but with variable perceptual impact.

V. Discussion

The project achieved stable, real-time EEG-to-MIDI translation using low-cost hardware. Technical performance was reliable, with band-specific amplitude mapping preserved across four lobe streams. However, user influence over music was only partially validated.

The temporal lobe memory recall task aligned best with the literature, producing theta increases and subjective reports of change, though statistical correlation was inconclusive. Parietal alpha suppression during spatial tasks and frontal beta changes during motor imagery were less evident both in data and perception. Mapping subtle spectral shifts to nuanced effects (e.g., stereo width) may have reduced perceived control.

VI. Future Work

Several refinements could improve the system's scientific validity and creative scope.

For signal processing, implementing relative power normalisation (e.g., expressing each band as a percentage of total 4–30 Hz power) would enhance comparability across participants and sessions. Ratios such as alpha/beta could yield more stable mappings, while spectral windowing and log-power calculations would improve FFT resolution.

For stimulus validation, increasing the EEG data rate would better synchronise neural changes with musical modulation. Live timestamped event logging and statistical analysis could confirm alignment between EEG patterns and intended tasks, while longer trials and larger participant pools would increase reliability.

In mapping strategies, exploring FM or granular synthesis, assigning EEG features to more perceptually obvious parameters, or modulating pre-recorded playback could create clearer before/after comparisons.

These enhancements could make the system more precise, responsive, and versatile for both research and artistic performance.

VII. CONCLUSION

This study investigated whether EEG signals from distinct brain lobes could be mapped in real time to control VST parameters within a DAW. A portable EEG device (NeuroBell Luna) was successfully integrated with a Python-based processing pipeline and Reaper, achieving stable, lobe-specific MIDI control.

While the system met its technical objective, enabling continuous neural modulation of synthesiser parameters, perceptual results were mixed.

Key contributions include demonstrating that low-channel, consumer-grade EEG hardware can support functional BCMIs. Future work will refine signal processing, expand mapping strategies, validate stimulus–response correlations, and explore cross-disciplinary applications.

The findings support EEG-driven modulation as a creative tool, with promise for inclusive performance, adaptive composition, and responsive sound design.

Acknowledgments

I would like to formally thank the amazing people of NeuroBell, Ballincollig, for allowing me to bring this vision to fruition. A special thank you to the CSO, Dr. Alison O'Shea, and to the CTO, Colm Murphy, for all the help regarding the development and use of the Luna Amplifier, along with all the EEG dataset advice. Thank you to all my peers in Music Technology, as well as to the lecturers of CSM and MTU.

References

[1] L. Bellier *et al.*, "Music can be reconstructed from human auditory cortex activity using nonlinear decoding models," *PLOS Biology*, vol. 21, no. 8, pp. 1–27, Aug. 2023, doi: 10.1371/journal.pbio.3002176.

[2] M. H. Thaut, P. D. Trimarchi, and L. M. Parsons, "Human brain basis of musical rhythm perception: common and distinct neural substrates for meter, tempo, and pattern.," *Brain Sci*, vol. 4, no. 2, pp. 428–452, June 2014, doi: 10.3390/brainsci4020428.

[3] S. I. Giraldo and R. Ramírez, "Brain-Activity-Driven Real-Time Music Emotive Control," 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:51762634

[4] J. Malmivuo and R. Plonsey, "Bioelectromagnetism. 13. Electroencephalography," 1995, pp. 247–264.

[5] T. A. Deuel, J. Pampin, J. Sundstrom, and F. Darvas, "The Encephalophone: A Novel Musical Biofeedback Device using Conscious Control of Electroencephalogram (EEG).," *Front Hum Neurosci*, vol. 11, p. 213, 2017, doi: 10.3389/fnhum.2017.00213.

[6] E. Miranda, "Brain-Computer Music Interface for Generative Music".

[7] E. Miranda, N.-F. Wilson, R. Palaniappan, J. Eaton, and W. Magee, "Brain-Computer Music Interfacing (BCMI) From Basic Research to the Real World of Special Needs," *Music and Medicine*, vol. 3, pp. 134–140, July 2011, doi: 10.1177/1943862111399290.

[8] "NeuroBell Luna Amplifier," NeuroBell Tech. [Online]. Available: https://www.neurobell.com/tech

[9] T. A. de Graaf, J. Gross, G. Paterson, T. Rusch, A. T. Sack, and G. Thut, "Alpha-band rhythms in visual task performance: phase-locking by rhythmic sensory stimulation.," *PLoS One*, vol. 8, no. 3, p. e60035, 2013, doi: 10.1371/journal.pone.0060035.

[10] E. Tan *et al.*, "Theta activity and cognitive functioning: Integrating evidence from resting-state and task-related developmental electroencephalography (EEG) research.," *Dev Cogn Neurosci*, vol. 67, p. 101404, June 2024, doi: 10.1016/j.dcn.2024.101404.

[11] P. B. Sederberg, M. J. Kahana, M. W. Howard, E. J. Donner, and J. R. Madsen, "Theta and gamma oscillations during encoding predict subsequent recall.," *J Neurosci*, vol. 23, no. 34, pp. 10809–10814, Nov. 2003, doi: 10.1523/JNEUROSCI.23-34-10809.2003.

[12] C. M. Michel, L. Kaufman, and S. J. Williamson, "Duration of EEG and MEG α Suppression Increases with Angle in a Mental Rotation Task.," *J Cogn Neurosci*, vol. 6, no. 2, pp. 139–150, Spring 1994, doi: 10.1162/jocn.1994.6.2.139.

[13] M. Assem *et al.*, "High gamma activity distinguishes frontal cognitive control regions from adjacent cortical networks.," *Cortex*, vol. 159, pp. 286–298, Feb. 2023, doi: 10.1016/j.cortex.2022.12.007.

[14] S. Itthipuripat, J. R. Wessel, and A. R. Aron, "Frontal theta is a signature of successful working memory manipulation.," *Exp Brain Res*, vol. 224, no. 2, pp. 255–262, Jan. 2013, doi: 10.1007/s00221-012-3305-3.

Appendix C) Sample EDF Google Colab Code – Python

```
# -*- coding: utf-8 -*-
"""SampleVersionofCode.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1f6Xrm84LfBD03b3eavmXg5R2c-qpU14u

# Install MNE
"""

!pip install mne

import mne
import numpy as np
import matplotlib.pyplot as plt
import time

"""# Load EDF file"""

edf_path = '/content/sub-1_task-oa_eeg.edf'  # Update path as needed
raw = mne.io.read_raw_edf(edf_path, preload=True)

"""# Pick EEG channels only

"""

raw.pick_types(eeg=True)

print(raw)
print(raw.info)

# Compute PSD using Welch's method
psd = raw.compute_psd(method='welch', fmin=0.5, fmax=40, n_fft=2048)
psds = psd.get_data()
freqs = psd.freqs

# Average across channels
psd_mean = psds.mean(axis=0)

# Plot
plt.figure(figsize=(10, 5))
plt.plot(freqs, psd_mean, label='Average PSD')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Power Spectral Density (uV²/Hz)')
plt.title('EEG Power Spectral Density (Welch)')
plt.grid(True)
plt.legend()
plt.show()

"""# Set parameters"""
```

```python
sfreq = int(raw.info['sfreq'])  # Sampling frequency
chunk_size = sfreq  # 1-second chunks
total_samples = raw.n_times
n_channels = len(raw.ch_names)
n_chunks = (total_samples - chunk_size) // chunk_size

# Frequency bands
bands = {
    "delta": (1, 4),
    "theta": (4, 8),
    "alpha": (8, 12),
    "beta": (13, 30),
}

"""Visualise"""

# Get 1-second EEG data (e.g., first 1 sec)
data, times = raw[:, :chunk_size]  # EEG data: shape (n_channels, chunk_size)

# FFT
fft_vals = np.abs(np.fft.rfft(data, axis=1))
freqs = np.fft.rfftfreq(chunk_size, 1 / sfreq)

# Plot FFT of Frontal EEG channels
channel_idx1 = 0
channel_idx2 = 1
channel_idx3 = 2
plt.figure(figsize=(12, 15))
# Create the subplot first
ax = plt.subplot(3, 1, 1)
# Then plot the data on the created axes
ax.plot(freqs, fft_vals[channel_idx1], label=raw.ch_names[channel_idx1])
ax.set_xlabel('Frequency (Hz)')
ax.set_ylabel('Amplitude')
# Use channel_idx1 in the title
ax.set_title(f'Raw FFT of Channel {raw.ch_names[channel_idx1]}')
ax.grid(True)
ax.legend()

# Create the subplot first
ax = plt.subplot(3, 1, 2)
# Then plot the data on the created axes
ax.plot(freqs, fft_vals[channel_idx2], label=raw.ch_names[channel_idx2])
ax.set_xlabel('Frequency (Hz)')
ax.set_ylabel('Amplitude')
# Use channel_idx1 in the title
ax.set_title(f'Raw FFT of Channel {raw.ch_names[channel_idx2]}')
ax.grid(True)
ax.legend()

# Create the subplot first
ax = plt.subplot(3, 1, 3)
# Then plot the data on the created axes
ax.plot(freqs, fft_vals[channel_idx3], label=raw.ch_names[channel_idx3])
ax.set_xlabel('Frequency (Hz)')
ax.set_ylabel('Amplitude')
```

```
# Use channel_idx1 in the title
ax.set_title(f'Raw FFT of Channel {raw.ch_names[channel_idx3]}')
ax.grid(True)
ax.legend()

plt.show()

raw.plot(n_channels=10, duration=20, scalings='auto', title='Raw EEG')

"""ReReference using Fx"""

rereferenced_raw, ref_data = mne.set_eeg_reference(raw, ["EEG Fz-A1A2"],
copy=True)
fig_orig = raw.plot()
fig_reref = rereferenced_raw.plot()

"""Check for Events"""

if raw.annotations:
    events, _ = mne.events_from_annotations(raw)
    mne.viz.plot_events(events, sfreq=sfreq, first_samp=raw.first_samp)

"""# Step 4: Simulate real-time feed"""

# Storage for visualization
band_history = {band: [] for band in bands}
midi_history = {band: [] for band in bands}
time_stamps = []

# Process data in chunks
for i in range(n_chunks):
    start = i * chunk_size
    stop = start + chunk_size
    data, _ = raw[:, start:stop]

    fft_vals = np.abs(np.fft.rfft(data, axis=1))
    freqs = np.fft.rfftfreq(chunk_size, 1 / sfreq)

"""# Step 5: Compute band amplitudes"""

band_amps = {}
for band, (fmin, fmax) in bands.items():
        mask = (freqs >= fmin) & (freqs <= fmax)
        power = fft_vals[:, mask].mean(axis=1).mean()
        band_amps[band] = power
        band_history[band].append(power)

"""# Normalize to MIDI [0-127]"""

max_amp = max(band_amps.values()) or 1e-6  # avoid div by zero
midi_values = {
        band: int(np.clip((amp / max_amp) * 127, 0, 127))
        for band, amp in band_amps.items()
    }

for band in bands:
```

```
        midi_history[band].append(midi_values[band])

"""# Output MIDI String"""

# Output MIDI string
midi_str = ", ".join(f"{band.upper()}={val}" for band, val in
midi_values.items())
print(f"[{i:02d}s] MIDI: {midi_str}")

time_stamps.append(i)
time.sleep(0.1)  # Simulated fast live feed for demo

"""Plot of Band Power Time Series"""

for band in bands:
    if len(band_history[band]) != len(time_stamps):
        print(f"Warning: Mismatch in lengths for band '{band}'")

plt.figure(figsize=(12, 6))
for band in bands:
  plt.plot(time_stamps, band_history[band], label=f"{band} (amp)", alpha=0.6)
plt.title("Raw EEG Band Power Over Time")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.legend()
plt.grid(True)
plt.show()

"""Plot MIDI Values Over Time"""

plt.figure(figsize=(12, 6))
for band in bands:
    plt.plot(time_stamps, midi_history[band], label=f"{band} (MIDI)",
linewidth=2)
plt.title("Normalized MIDI Values Over Time (0-127)")
plt.xlabel("Time (s)")
plt.ylabel("MIDI Value")
plt.legend()
plt.grid(True)
plt.show()
```

Appendix D) Visual Studio Code – Python

D.1) Sample EDF & Reaper Interface Code

```
# VSVersionofCode_V1.0.py
# Real-Time EEG Band Power to MIDI CC Conversion Script V1.0
#________________________________________________________________________________
# Import Libraries
#````````````````````````````````````````````````````````````````````````````````
````````````````````````````````````````

import mne
import numpy as np
import mido
import time

Load Data
#``
````````````````````````````````````````

# Path to your EEG EDF file
edf_path = 'C:/Users/Windows 10/OneDrive/Documents/OpenNeuro/sub-1/eeg/sub-
1_task-oa_eeg.edf'

# Load the raw EDF data
raw = mne.io.read_raw_edf(edf_path, preload=True)
print(f"Loaded data with {raw.info['nchan']} channels and duration {raw.times[-
1]:.2f} seconds.")

# Select only EEG channels (drop others like EOG or annotations)
raw_eeg = raw.copy().pick_types(eeg=True)
print(f"EEG channels selected: {raw_eeg.ch_names}")

# Define channel groups by lobe based on name prefixes
channel_groups = {
    "Frontal":   [ch for ch in raw_eeg.ch_names if ch.upper().startswith("EEG
F") or ch.upper().startswith("F")],
    "Central":   [ch for ch in raw_eeg.ch_names if ch.upper().startswith("EEG
C") or ch.upper().startswith("C")],
    "Temporal":  [ch for ch in raw_eeg.ch_names if ch.upper().startswith("EEG
T") or ch.upper().startswith("T")],
    "Parietal":  [ch for ch in raw_eeg.ch_names if ch.upper().startswith("EEG
P") or ch.upper().startswith("P")],
    "Occipital": [ch for ch in raw_eeg.ch_names if ch.upper().startswith("EEG
O") or ch.upper().startswith("O")]
}

print("Channel groups by lobe:", {lobe: len(chs) for lobe, chs in
channel_groups.items()})
```

```
# For future use: could iterate over channel_groups to process each lobe
separately.
# For now, combine all EEG channels for a single analysis.
data = raw_eeg.get_data()                # shape is (n_channels, n_samples)
combined_signal = data.mean(axis=0)      # average across all EEG channels

# Get Info About Dataset
#``````````````````````````````````````````````````````````````````````````````
``````````````````````````````````````

print(f"Sampling Frequency of Set: {raw.info['sfreq']}")

print(f"EEG Sample Data: {raw_eeg.get_data(0, 120)}")

# Set up MIDI Output - Virtual port
#``````````````````````````````````````````````````````````````````````````````
``````````````````````````````````````

midi_port_name = 'loopMIDI Port 1'

# Open available chosen output port
outport = mido.open_output(midi_port_name)
print(f"MIDI output port '{midi_port_name}' opened. Send this port to Reaper as
MIDI input.")

# Define frequency band ranges (Hz)
theta_band = (4.0, 7.0)
alpha_band = (8.0, 12.0)
beta_band  = (13.0, 30.0)

# Assign MIDI CC numbers for each band (adjust as needed)
cc_theta = 21  # e.g., CC21 for Theta
cc_alpha = 22  # e.g., CC22 for Alpha
cc_beta  = 23  # e.g., CC23 for Beta


# Simulate Live EEG Stream (1-second Chunks)
#``````````````````````````````````````````````````````````````````````````````
``````````````````````````````````````

sfreq = int(raw.info['sfreq'])  # sampling frequency (Hz)
chunk_samples = sfreq * 1  # number of samples in 1-second chunk
total_samples = combined_signal.shape[0]

print("Starting real-time simulation...")
for start in range(0, total_samples, chunk_samples):
    end = start + chunk_samples
    if end > total_samples:
        break  # ignore last partial chunk if any

    chunk = combined_signal[start:end]
    # Remove DC offset (mean) from the chunk
    chunk = chunk - np.mean(chunk)

    # Compute FFT of the 1-second chunk
    freqs = np.fft.rfftfreq(len(chunk), d=1.0/sfreq)
```
```

```
    fft_vals = np.fft.rfft(chunk)
    magnitudes = np.abs(fft_vals)

    # Test FFT Values

    #print(f"FFT Freq Values Sample Data: {freqs}")
    #print(f"FFT Values EEG Sample Data: {fft_vals}")

    # Compute average magnitude in each frequency band
    theta_idx = (freqs >= theta_band[0]) & (freqs <= theta_band[1])
    alpha_idx = (freqs >= alpha_band[0]) & (freqs <= alpha_band[1])
    beta_idx  = (freqs >= beta_band[0])  & (freqs <= beta_band[1])
    theta_amp = magnitudes[theta_idx].mean()
    alpha_amp = magnitudes[alpha_idx].mean()
    beta_amp  = magnitudes[beta_idx].mean()

    # Scale amplitudes to MIDI 0-127 range

    # Scaling factor for normalisation as needed
    scale_factor = 1e+05
    theta_val = int(np.clip(theta_amp * scale_factor, 0, 127))
    alpha_val = int(np.clip(alpha_amp * scale_factor, 0, 127))
    beta_val  = int(np.clip(beta_amp  * scale_factor, 0, 127))
    # Send MIDI Control Change messages for each band
    outport.send(mido.Message('control_change', control=cc_theta,
value=theta_val))
    outport.send(mido.Message('control_change', control=cc_alpha,
value=alpha_val))
    outport.send(mido.Message('control_change',
control=cc_beta,  value=beta_val))
    # Print the values for debugging/monitoring
    print(f"t={start/sfreq:.1f}s – Theta:{theta_val}, Alpha:{alpha_val},
Beta:{beta_val}")
    # Wait for 1 second before next chunk (simulating live stream timing)
    time.sleep(1)

print("EEG-to-MIDI streaming finished.")
```

D.2) Final Four Lobe Code for Device & Reaper Interfacing

```
# VSVersionofCode_LobeSpecific_V3.3.py
# Real-Time EEG Band Power to MIDI CC Conversion with Filters & Serial
Streaming
#_______________________________________________________________________________
__

# Adjust date of CSV file name ; Line 77

# Import Libraries
#`````````````````````````````````````````````````````````````````````````````````
``````````````````````````````````````

import numpy as np
import mido
import time
import serial
import csv
from scipy.signal import butter, filtfilt

# Serial Port Configuration
#`````````````````````````````````````````````````````````````````````````````````
``````````````````````````````````````

serial_port = 'COM8'
baud_rate = 115200
sfreq = 250
chunk_samples = sfreq
num_channels = 8
midi_port_name = 'loopMIDI Port 1'


# Mapping & Scaling
#`````````````````````````````````````````````````````````````````````````````````
``````````````````````````````````````


# Channel to lobe mapping
channel_lobe_map = {
    "Occipital": [0, 1],
    "Temporal": [2, 3],
    "Central": [4, 5],
    "Frontal": [6, 7]
}

# Active lobes toggle
active_lobes = {
    "Frontal": True,
    "Central": False,
    "Temporal": False,
```
```

```
    "Occipital": False
}

# MIDI CC numbers per lobe-band
cc_mapping = {
    "Frontal":   {"Theta": 16, "Alpha": 17, "Beta": 18},
    "Central":   {"Theta": 19, "Alpha": 20, "Beta": 21},
    "Temporal":  {"Theta": 22, "Alpha": 23, "Beta": 24},
    "Occipital": {"Theta": 25, "Alpha": 26, "Beta": 27}
}

# EEG bands
bands = {
    "Theta": (4.0, 7.0),
    "Alpha": (8.0, 12.0),
    "Beta": (13.0, 30.0)
}

scale_factor = 0.001


# Filtering & Setups
#`````````````````````````````````````````````````````````````````````````````
``````````````````````````````````````


# Filter design
def butter_bandpass(lowcut, highcut, fs, order=4):
    nyq = 0.5 * fs  # Nyquist limit set
    b, a = butter(order, [lowcut/nyq, highcut/nyq], btype='band')
    return b, a

def apply_bandpass(data, lowcut=1.6, highcut=40.0, fs=250):
    b, a = butter_bandpass(lowcut, highcut, fs)
    return filtfilt(b, a, data)

# Serial port setup
try:
    ser = serial.Serial(serial_port, baud_rate, timeout=1)
    print(f"Listening on {serial_port} at {baud_rate} baud...\n")
except Exception as e:
    raise SystemExit(f"Could not open serial port: {e}")

# MIDI setup
outport = mido.open_output(midi_port_name)
print(f"MIDI output port '{midi_port_name}' opened.\n")

# CSV logging setup
csv_file = open("eeg_stream_log_rename_here.csv", "w", newline='')
csv_writer = csv.writer(csv_file)
```
```

~~~python
csv_writer.writerow(["Timestamp"] + [f"Ch{i}" for i in range(num_channels)])

# Parse function
def parse_serial_line(line):
    try:
        # Decode and strip control characters
        line = line.decode(errors='ignore').strip()

        # Check format: starts with $, ends with ;
        if not (line.startswith('$') and line.endswith(';')):
            return None

        # Remove '$' and ';', then split by space
        body = line[1:-1].strip()
        parts = body.split()

        # Convert all to integers, allow negative numbers and zeros
        if len(parts) != num_channels:
            return None

        return [int(p) for p in parts]

    except Exception as e:
        print(f"Parse error: {e}")
        return None


# Main Loop
#``````````````````````````````````````````````````````````````````````````````
``````````````````````````````````````


print("Streaming EEG data...\n")
while True:
    eeg_buffer = [[] for _ in range(num_channels)]
    start_time = time.time()

    while all(len(ch_data) < chunk_samples for ch_data in eeg_buffer):
        try:
            line = ser.read_until(b';')
            values = parse_serial_line(line)

            # Ensure valid and complete sample
            if values is not None:
                for ch in range(num_channels):
                    eeg_buffer[ch].append(values[ch])
                timestamp = time.strftime("%H:%M:%S", time.localtime())
                csv_writer.writerow([timestamp] + values)
            else:
                print(f"Incomplete or invalid line skipped: {line}")
~~~

```
        except Exception as e:
            print(f"Serial read error: {e}")

    for i, ch in enumerate(eeg_buffer):
        if len(ch) != chunk_samples:
            print(f"Channel {i} only has {len(ch)} samples!")

    eeg_data = np.array(eeg_buffer)  # shape (8, 250)

    for lobe, ch_indices in channel_lobe_map.items():
        if not active_lobes.get(lobe, False):
            continue

        valid_channels = [eeg_data[i] for i in ch_indices if not
np.all(eeg_data[i] == 0)]
        if not valid_channels:
            print(f"{lobe} skipped: all channels zero.")
            continue

        lobe_signal = np.mean(valid_channels, axis=0)
        raw_mean = int(np.mean(lobe_signal))
        print(f"{lobe} µV mean: {raw_mean}")

        filtered = apply_bandpass(lobe_signal, 1.6, 40.0, sfreq)
        filtered -= np.mean(filtered)

        freqs = np.fft.rfftfreq(chunk_samples, d=1.0 / sfreq)
        fft_vals = np.fft.rfft(filtered)
        magnitudes = np.abs(fft_vals)

        for band_name, (f_low, f_high) in bands.items():
            idx = (freqs >= f_low) & (freqs <= f_high)
            amp = magnitudes[idx].mean() if np.any(idx) else 0.0
            midi_val = int(np.clip(amp * scale_factor, 0, 127))
            cc_num = cc_mapping[lobe][band_name]
            outport.send(mido.Message('control_change', control=cc_num,
value=midi_val))
            print(f"Sent MIDI CC{cc_num} ({lobe}-{band_name}): {midi_val}")

    elapsed = time.time() - start_time
    if elapsed < 1.0:
        time.sleep(1.0 - elapsed)
```

## Appendix E) Custom Reaper JSFW Plugin Code

```
desc:MIDI CC Frequency Bands Scope (Ch1, CC16–27, 30s@1Hz)

// Hard MIDI CC Mapping (Ch 1):
// Frontal:   Theta=CC16, Alpha=CC17, Beta=CC18
// Central:   Theta=CC19, Alpha=CC20, Beta=CC21
// Temporal:  Theta=CC22, Alpha=CC23, Beta=CC24
// Occipital: Theta=CC25, Alpha=CC26, Beta=CC27
// - 30s fixed window, 1Hz sample-and-hold, normalized 0–1
// - Only draws traces after receiving valid data
// - Passes through MIDI unchanged

slider1:1<0,1,1{Off,On}>Show Grid
slider2:2<1,4,1>Line Thickness
slider3:0<0,1,1{No,Yes}>Clear

@init
WIN_S = 30;
SPS = 1;
NSAMP = WIN_S * SPS + 1;
N_LOBES = 4;
N_BANDS = 3;
N_TRACES = N_LOBES * N_BANDS;

function map_cc(idx)(
  idx == 0 ? 16 :
  idx == 1 ? 17 :
  idx == 2 ? 18 :
  idx == 3 ? 19 :
  idx == 4 ? 20 :
  idx == 5 ? 21 :
  idx == 6 ? 22 :
  idx == 7 ? 23 :
  idx == 8 ? 24 :
  idx == 9 ? 25 :
  idx == 10 ? 26 :
  27;
);

lobe_name_0 = "Frontal";
lobe_name_1 = "Central";
lobe_name_2 = "Temporal";
lobe_name_3 = "Occipital";

function TIDX(lobe, band)(lobe*3 + band);
function times_ptr(idx)(idx * NSAMP);
function values_ptr(idx)(N_TRACES * NSAMP + idx * NSAMP);
```

```
times_base = 0;
values_base = N_TRACES * NSAMP;
wr_base = values_base + N_TRACES * NSAMP;
seen_base = wr_base + N_TRACES;
total_mem = N_TRACES * NSAMP * 2 + N_TRACES * 2;
memset(0, 0, total_mem);

i=0; loop(N_TRACES,
  memory[wr_base+i] = 0;
  memory[seen_base+i] = 0;
);

i=0; loop(N_TRACES,
  held_val[i] = -1;
);

t = 0;
next_sample_t = 1;
font_sz = 13;

function set_band_color(b)(
  b==0 ? (gfx_r=0.6; gfx_g=0.85; gfx_b=1.0) :
  b==1 ? (gfx_r=0.65; gfx_g=1.0; gfx_b=0.6) :
         (gfx_r=1.0; gfx_g=0.7; gfx_b=0.55);
  gfx_a = 1;
);

function set_text()(
  gfx_r=gfx_g=gfx_b=0.95;
  gfx_a = 1;
);

function draw_line_thick(xa,ya, xb,yb, thk)(
  thk <= 1 ? gfx_line(xa,ya, xb,yb)
  : (
    span = thk - 1;
    i = 0;
    loop(thk,
      dy = (i - span*0.5);
      gfx_line(xa, ya+dy, xb, yb+dy);
      i += 1;
    );
  );
);

@slider
show_grid = slider1|0;
thick = max(1, slider2|0);
do_clear = slider3|0;
```

```
do_clear ? (
  memset(0, 0, total_mem);
  i=0; loop(N_TRACES, held_val[i] = -1;);
  t = 0; next_sample_t = 1;
  slider3 = 0;
);

@block
t += samplesblock / srate;

while (midirecv(ofs, m1, m23)) (
  midisend(ofs, m1, m23);

  status = m1 & 0xF0;
  ch = (m1 & 0x0F) + 1;
  d1 = m23 & 0xFF;
  d2 = (m23 >> 8) & 0xFF;

  (status == 0xB0) && (ch == 1) ? (
    idx = -1;
    j=0; loop(N_TRACES,
      (map_cc(j) == d1) ? idx = j;
    );
    idx >= 0 ? (
      d2 = min(max(d2, 0), 127);
      held_val[idx] = d2 / 127;
      memory[seen_base + idx] = 1;
    );
  );
);

while (t >= next_sample_t)(
  tr=0; loop(N_TRACES,
    wr = memory[wr_base + tr];
    tp = times_ptr(tr);
    vp = values_ptr(tr);
    memory[tp + wr] = next_sample_t;
    v = held_val[tr];
    memory[vp + wr] = (v >= 0) ? v : -1;
    memory[wr_base + tr] = (wr + 1) % NSAMP;
  );
  next_sample_t += 1;
);

@gfx 900 560
gfx_clear = 0x101010;
W = gfx_w; H = gfx_h;

padL=72; padR=12; padT=12; padB=44;
```

```
innerW = max(20, W - padL - padR);
innerH = max(60, H - padT - padB);
rows = 4;
rowH = innerH / rows;

gfx_setfont(1, "Arial", font_sz);
set_text();
gfx_x = padL; gfx_y = 2;
gfx_drawstr("MIDI CC Frequency Bands Scope  — CH1  |  Window 30s");

t_now = t;
t_l = t_now - WIN_S;

l=0; loop(rows,
  x0 = padL; x1 = padL + innerW;
  y0 = padT + l*rowH;
  y1 = y0 + rowH - 10;

  gfx_a=1; gfx_r=gfx_g=gfx_b=0.85;
  gfx_rect(x0-1,y0-1, innerW+2, (y1-y0)+2, 0);
  gfx_line(x0,y0, x0,y1);
  gfx_line(x0,y1, x1,y1);

  show_grid ? (
    gfx_r=gfx_g=gfx_b=0.35;
    gi=0; loop(9,
      val = min(gi*16, 127);
      gy = y1 - (val/127)*(y1-y0);
      gfx_line(x0, gy, x1, gy);
      set_text(); gfx_x=8; gfx_y=gy - font_sz*0.5;
      sprintf(#v,"%d",val); gfx_drawstr(#v);
      gi+=1;
    );
    divs=10; gj=0; loop(divs+1,
      gx = x0 + (gj/divs)*innerW;
      gfx_line(gx, y0, gx, y1);
      l == rows-1 ? (
        set_text(); gfx_x=gx-12; gfx_y=y1+6;
        sprintf(#tl,"%.0fs",(divs-gj)/divs*WIN_S); gfx_drawstr(#tl);
      );
    );
  );

  set_text();
  l==0 ? lstr=lobe_name_0;
  l==1 ? lstr=lobe_name_1;
  l==2 ? lstr=lobe_name_2;
  l==3 ? lstr=lobe_name_3;
  gfx_x = x0 + 6; gfx_y = y0 + 4;
  gfx_drawstr(lstr);
```

```
  legend_x = x1 - 240; legend_y = y0 + 6;
  set_band_color(0); gfx_rect(legend_x, legend_y+2, 12, 8, 1);
  set_text(); gfx_x=legend_x+16; gfx_y=legend_y; gfx_drawstr("Theta");
  set_band_color(1); gfx_rect(legend_x+85, legend_y+2, 12, 8, 1);
  set_text(); gfx_x=legend_x+101; gfx_y=legend_y; gfx_drawstr("Alpha");
  set_band_color(2); gfx_rect(legend_x+170, legend_y+2, 12, 8, 1);
  set_text(); gfx_x=legend_x+186; gfx_y=legend_y; gfx_drawstr("Beta");

  b=0; loop(3,
    idx = TIDX(l,b);
    tp = times_ptr(idx);
    vp = values_ptr(idx);
    wr = memory[wr_base + idx];
    set_band_color(b);

    start = wr % NSAMP;
    have_prev = 0;

    k=0; loop(NSAMP,
      p = (start + k) % NSAMP;
      ts = memory[tp + p];
      val = memory[vp + p];

      (val >= 0) && (ts >= t_l) && (ts <= t_now) ? (
        px = x0 + ((ts - t_l)/WIN_S)*innerW;
        py = y1 - val*(y1 - y0);
        have_prev ? draw_line_thick(px_prev, py_prev, px, py, thick)
                  : (gfx_x=px; gfx_y=py);
        px_prev=px; py_prev=py; have_prev=1;
      );
      k+=1;
    );
  );

  l+=1;
);
```

## Appendix F) Hardware Set-Up

NeuroBell Luna Amplifier, USB Streaming Iteration:


Stills from Testing and Surveying Video Demos:

## Appendix G) Video Demo Link

https://youtu.be/jQLDnM-_FBs

# Appendix H) User Questionnaire



## Questionnaire for Mapping Cognitive Control

EEG Signals Modulating Real-Time Music Composition
This survey is in relation to an in-person trial where each participant performs four tasks designed to engage different brain regions while listening to a Bach prelude (approximately 2 minutes long). Each task follows a **stop–start–stop** structure: a brief rest period, an active task period (around 1 minute), and a final rest. After each task, participants answer a set of questions.
Thank you for your participation.

* Required

* This form will record your name, please fill your name.

### Ethics Agreement

1. I consent that I voluntarily agree to participate in this questionnaire. *

   ◯ Yes

   ◯ No

2. I agree that all data I provide as part of the questionnaire may be used as part of this research study, the nature of which has been fully explained to me in detail. *

   ◯ Yes

   ◯ No

3. I understand that I can withdraw from partaking in the study and may retract my questionnaire at any time after the original session (in which case all data will be deleted). *

   ◯ Yes

   ◯ No

4. I also understand that I may decline to answer any part of the questionnaire if I wish. *

   ◯ Yes

   ◯ No

5. I accept that there will be no direct benefits from my participating in this questionnaire.
*

- Yes
- No

6. I understand that all data I provide will be treated confidentially and anonymously, and only be used for the purposes of the agreed research study.
*

- Yes
- No

7. I recognise that I can reach out at any time if I wish to access the information I have provided, either for review or withdrawal.
*

- Yes
- No

## Occipital Lobe Stimulus Test

Visual Flicker Task (9 Hz Light Stimulus) **Task:** You will look at a visual stimulus that flashes at 9 Hz (9 times per second) to induce activity in the visual cortex. The Bach prelude will play while the light alternately flashes (active period) and stays off (rest periods). This task lasts about 2 minutes in total (rest – recall – rest).

8. During the flashing-light task, did you notice any change in the music's **timbre or sound** when the light was **flashing** compared to when it was **not**? *

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 = No difference noticed, 5 = Major difference noticed | ○ | ○ | ○ | ○ | ○ |

9. How **easy or difficult** was it to keep your eyes on the flashing stimulus and remain still for the duration of this task *while listening to the music*? *

| | Extremely difficult | Neutral | Somewhat Easy | Somewhat difficult | Very Easy |
|---|---|---|---|---|---|
| Statement | ○ | ○ | ○ | ○ | ○ |

10. "Focusing on the visual flashing **distracted** me from the music." – To what extent do you **agree** or **disagree** with this statement? *

| | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
|---|---|---|---|---|---|
| Statement | ○ | ○ | ○ | ○ | ○ |

11. "I felt that my **mind/brain activity** (looking at the flashing light) had a **direct effect** on the way the music sounded." – How much do you agree? *

| | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
|---|---|---|---|---|---|
| Statement | ○ | ○ | ○ | ○ | ○ |

12. Please **describe any differences** in the music you heard while the light was flashing versus when it was off. (If you didn't notice a difference, you can note that as well.) *

## Temporal Lobe Stimulus Test

Auditory Memory Task (Mental Recall) **Task:** You will hear the Bach prelude, and at the cue, you will perform a **mental recall** task to engage auditory and memory regions. You will be asked to **recall a series of numbers** you learned beforehand for about 1 minute, then stop. The music continues playing throughout, with periods of just listening (rest) before and after the mental recall. Total task time is about 2 minutes (rest – recall – rest).

13. During the memory recall task, did you notice any **changes in the music's timbre or effects** when you were actively **recalling the numbers** compared to when you were just listening normally? *

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 = No difference noticed, 5 = Major difference noticed | ○ | ○ | ○ | ○ | ○ |

14. How **difficult** was it to perform the mental recall task *while listening to the music*? *

| | Extremely difficult | Neutral | Somewhat difficult | Somewhat easy | Very easy |
|---|---|---|---|---|---|
| Statement | ○ | ○ | ○ | ○ | ○ |

15. "Concentrating on remembering the information **made it hard for me to notice** changes in the music." – How much do you agree with this statement? *

| | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
|---|---|---|---|---|---|
| Statement | ○ | ○ | ○ | ○ | ○ |

16. "I felt that my **mental state** (memory exercise) **influenced** the way the music sounded in real time." – Please rate your agreement. *

| | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
|---|---|---|---|---|---|
| Statement | ○ | ○ | ○ | ○ | ○ |

17. Describe any **differences in the sound** of the music that you noticed while you were focusing on recalling the information. (If you didn't notice any change, you can state that.) *

## Parietal Lobe Stimulus Test

Spatial Imagery Task (Mental Rotation) **Task:** In this task, you will be performing a **spatial imagination** exercise while the music plays. At a cue, you should begin to **imagine a 3D object rotating in your mind's eye** (for example, a cube turning in space) for about 1 minute, then stop. Alternatively, you might be asked to visualise moving through a familiar space or do a quick mental math problem. The task includes a rest period before and after the imagery. Total task time is about 2 minutes (rest – spatial task – rest).

18. While you were doing the mental rotation/visualisation, did you notice any **difference in the music's sound or effects** compared to when you were not doing the task? *

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 = No difference noticed, 5 = Major difference noticed | ◯ | ◯ | ◯ | ◯ | ◯ |

19. How **difficult** was it to carry out the mental rotation or visualisation task *while listening to the music*? *

| | Extremely difficult | Somewhat easy | Neutral | Somewhat difficult | Very easy |
|---|---|---|---|---|---|
| Statement | ◯ | ◯ | ◯ | ◯ | ◯ |

20. "Visualising the object or space **took my attention away** from the music." – How would you rate your agreement with this? *

| | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
|---|---|---|---|---|---|
| Statement | ◯ | ◯ | ◯ | ◯ | ◯ |

21. "I felt that my **thoughts/imagery** during this task **modulated the music** I was hearing." – Please indicate your level of agreement. *

| | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
|---|---|---|---|---|---|
| Statement | ◯ | ◯ | ◯ | ◯ | ◯ |

22. Describe any **changes in the music** that you perceived when you were performing the spatial imagery task. (If you did not perceive any change, please note that.) *

## Frontal Lobe Stimulus Test

Motor Imagery Task (Imagined Movement) **Task:** While listening to the prelude, you will be prompted to perform a **motor imagery** exercise. When instructed, you should **imagine moving your right hand** (opening and closing your fist) without actually moving it. You will imagine the movement for about 1 minute, then relax again. The music plays continuously, with a rest period before and after the imagery. Total task time is about 2 minutes (rest – imagery – rest).

23. Did you notice any **change in the music's timbre/character** when you were **imagining the hand movement** versus when you were at rest (not imagining anything)? *

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 = No difference noticed, 5 = Major difference noticed | ○ | ○ | ○ | ○ | ○ |

24. How **easy or difficult** was it to concentrate on the imagined movement for the whole duration *while still listening to the music*? *

| | Extremely difficult | Somewhat difficult | Neutral | Somewhat easy | Very easy |
|---|---|---|---|---|---|
| Statement | ○ | ○ | ○ | ○ | ○ |

25. "Performing the imagined movement **distracted me from hearing** the music's details." – How do you rate your agreement with this statement? *

| | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree |
|---|---|---|---|---|---|
| Statement | ○ | ○ | ○ | ○ | ○ |

26. If you noticed differences, **what changes in the sound** did you perceive while performing the motor imagery? (If you did not notice any difference, please indicate that.) *

Thank you for participating in this trial !

This content is neither created nor endorsed by Microsoft. The data you submit will be sent to the form owner.

Microsoft Forms

## Appendix I) Data Processing Scripts

### I.1) Raw EEG Comparison Plotting

```
# EEG Plots from trials of Mapping Cognitive Control
# CSV Files containing microvolt information from each lobe under each testing
senario
#______________________________________________________________________________
____________
import pandas as pd
import matplotlib.pyplot as plt
from pathlib import Path

# File paths
file_paths = [
    "eeg_stream_log_08_08_TEMPROB.csv",
    "eeg_stream_log_08_08_PARIETALROB.csv",
    "eeg_stream_log_08_08_FRONTALROB.csv",
]
labels = ["Temporal", "Parietal", "Frontal"]


def process_file(path):
    df = pd.read_csv(path)
    # Convert Timestamp to datetime
    df['Timestamp'] = pd.to_datetime(df['Timestamp'], format='%H:%M:%S')
    # Calculate relative time in seconds
    df['Time_s'] = (df['Timestamp'] -
df['Timestamp'].iloc[0]).dt.total_seconds()
    # Cap to 150 seconds
    df = df[df['Time_s'] <= 150]
    # Compute mean amplitude across EEG channels
    df['Mean_uV'] = df.drop(columns=['Timestamp', 'Time_s']).mean(axis=1)
    return df['Time_s'], df['Mean_uV']

# Process each file
plot_data = [process_file(file_paths[i]) for i in range(len(file_paths))]

# Build comparison DataFrame
comparison_df = pd.DataFrame({
    "Time_s": plot_data[0][0],  # Use the first file's time as reference
    labels[0]: plot_data[0][1],
    labels[1]: plot_data[1][1],
    labels[2]: plot_data[2][1]
})
```

```python
# Save trimmed version as CSV
comparison_df.to_csv("eeg_stream_log_comparison.csv", index=False)
print("Capped comparison CSV saved as raw_eeg_stream_log_comparison.csv")

# ________________________________________________________

# Plot EEG with vertical stimulus markers
plt.figure(figsize=(10, 6))
for i, label in enumerate(labels):
    plt.plot(plot_data[i][0], plot_data[i][1], label=label)

# Add event markers
markers = [30, 90, 120]
for mark in markers:
    plt.axvline(x=mark, color='blue', linestyle='--', linewidth=1)
    plt.text(mark + 1, plt.ylim()[1]*0.95, f"{mark}s", color='blue')

# Styling
plt.xlabel("Time (s)")
plt.ylabel("Amplitude (μV)")
plt.title("EEG Stream Comparison with Event Markers")
plt.legend()
plt.grid(True)
plt.tight_layout()

# Save and show
plt.savefig("eeg_stream_comparison_plot.png", dpi=300)
plt.show()
```

### I.2) Feature Plotting

~~~python
# EEG per-lobe band features plotter (Theta/Alpha/Beta) with 150 s cap ===
# Reads per-lobe CSV logs, computes per-second band amplitudes, and plots 3
subplots.

# Import Libraries
#`````````````````````````````````````````````````````````````````````````````
```````

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.signal import butter, filtfilt


# CSV INPUTS
# ------------------------------------------------------------
file_paths = [
    "eeg_stream_log_08_08_TEMPROB.csv",
    "eeg_stream_log_08_08_PARIETALROB.csv",
    "eeg_stream_log_08_08_FRONTALROB.csv",
]
labels = ["Temporal", "Parietal", "Frontal"]


fs = 250                    # Hz
max_time_s = 150            # Cap analysis to 150 seconds
samp_per_sec = fs

channel_lobe_map = {
    "Occipital": [0, 1],
    "Temporal":  [2, 3],
    "Parietal":  [4, 5],
    "Frontal":   [6, 7],
}

bands = {
    "Theta": (4.0, 7.0),
    "Alpha": (8.0, 12.0),
    "Beta":  (13.0, 30.0)
}

bp_low, bp_high = 1.6, 40.0
event_marks_s = [30, 90, 120]

# DSP
# ------------------------------------------------------------
def butter_bandpass(low, high, fs, order=4):
~~~

```
    nyq = 0.5 * fs
    b, a = butter(order, [low/nyq, high/nyq], btype='band')
    return b, a

def apply_bandpass(x, low=bp_low, high=bp_high, fs=fs):
    if len(x) < 5:
        return x
    b, a = butter_bandpass(low, high, fs)
    return filtfilt(b, a, x)

def compute_band_amplitudes(signal_1d, fs=fs, bands=bands):
    seg = signal_1d - np.mean(signal_1d)
    freqs = np.fft.rfftfreq(len(seg), d=1.0/fs)
    mags  = np.abs(np.fft.rfft(seg))
    return {name: float(mags[(freqs >= f_lo) & (freqs <= f_hi)].mean()) if
np.any((freqs >= f_lo) & (freqs <= f_hi)) else 0.0
            for name, (f_lo, f_hi) in bands.items()}

def load_and_features(csv_path, lobe_name, lobe_channel_map, fs=fs):
    df = pd.read_csv(csv_path)
    ch_cols = [c for c in df.columns if c.startswith("Ch")]
    if len(ch_cols) < 8:
        raise ValueError(f"{csv_path}: expected 8 channel columns named
Ch0..Ch7.")

    ch_idx = lobe_channel_map.get(lobe_name, [])
    if not ch_idx:
        raise ValueError(f"No channel mapping for lobe '{lobe_name}'.")

    for i in ch_idx:
        if f"Ch{i}" not in df.columns:
            raise ValueError(f"{csv_path}: missing Ch{i} for {lobe_name}.")

    # Limit to first MAX_TIME_S seconds of data
    max_samples = max_time_s * fs
    if len(df) > max_samples:
        df = df.iloc[:max_samples]

    lobe_matrix = df[[f"Ch{i}" for i in ch_idx]].to_numpy(dtype=float)
    lobe_signal = np.mean(lobe_matrix, axis=1)
    lobe_filt = apply_bandpass(lobe_signal, bp_low, bp_high, fs)

    n_win = len(lobe_filt) // fs
    times = np.arange(n_win, dtype=float)
    feats = {name: [] for name in bands.keys()}

    for w in range(n_win):
        seg = lobe_filt[w*fs:(w+1)*fs]
```

```python
        band_vals = compute_band_amplitudes(seg, fs, bands)
        for name in bands.keys():
            feats[name].append(band_vals[name])

    return times, feats

# Run & plot
# ------------------------------------------------------------
results = []
for path, lobe in zip(file_paths, labels):
    t_s, band_feats = load_and_features(path, lobe, channel_lobe_map, fs)
    results.append((lobe, t_s, band_feats))

fig, axes = plt.subplots(nrows=3, ncols=1, figsize=(12, 10), sharex=True)

for ax, (lobe, t_s, feats) in zip(axes, results):
    ax.plot(t_s, feats["Theta"], label="Theta (4–7 Hz)")
    ax.plot(t_s, feats["Alpha"], label="Alpha (8–12 Hz)")
    ax.plot(t_s, feats["Beta"],  label="Beta (13–30 Hz)")
    ax.set_title(f"{lobe} Lobe Band Amplitudes over Time")
    ax.set_ylabel("Mean FFT Magnitudes (µV)")
    ax.grid(True, alpha=0.3)
    for m in event_marks_s:
        ax.axvline(x=m, linestyle="--", linewidth=1, color="red")
        ax.text(m + 0.5, ax.get_ylim()[1] * 0.92, f"{m}s", color="red")

axes[-1].set_xlabel("Time (s)")
axes[0].legend(loc="upper right")
fig.tight_layout()
plt.savefig("per_lobe_band_features_capped.png", dpi=300)
plt.show()
```