A General-Purpose Software Platform for Closed-Loop Neuromodulation

William Engelhardt
A General-Purpose Software Platform for Closed-Loop Neuromodulation

by

William Engelhardt

A thesis presented to the McKelvey School of Engineering of Washington University in partial fulfillment of the requirements for the degree of Master of Science

May 2023

St. Louis, Missouri
# Table of Contents

List of Figures .............................................................................................................................. iii
Acknowledgments ........................................................................................................................ iv
Abstract ....................................................................................................................................... v
Chapter 1: Introduction ................................................................................................................ 1
Chapter 2: Methods ....................................................................................................................... 6
  2.1 Configuration ......................................................................................................................... 6
  2.2 Implementation ..................................................................................................................... 8
  2.3 Visualization ......................................................................................................................... 10
Chapter 3: Results ....................................................................................................................... 12
  3.1 Validation ............................................................................................................................. 12
    3.1.1 Configuration ................................................................................................................ 12
    3.1.2 Implementation ............................................................................................................. 15
    3.1.3 Visualization ............................................................................................................... 17
  3.2 Specific use cases ................................................................................................................. 18
    3.2.1 PegasusAstro ............................................................................................................... 18
    3.2.2 EyeLink ........................................................................................................................ 19
    3.2.3 XSens .......................................................................................................................... 21
    3.2.4 NeuroOmega ................................................................................................................ 22
    3.2.5 CereStim ....................................................................................................................... 23
    3.2.6 Cortec .......................................................................................................................... 25
    3.2.7 Stimulation Configuration Tool ................................................................................... 26
    3.2.8 EGI ............................................................................................................................... 27
Chapter 4: Discussion ................................................................................................................... 29
  4.1 Configuration ......................................................................................................................... 29
  4.2 Implementation ..................................................................................................................... 31
  4.3 Visualization ......................................................................................................................... 32
Chapter 5: Conclusion ................................................................................................................ 34
References ..................................................................................................................................... 36
List of Figures

Figure 1. The architecture of BCI2000 allows for closed-loop feedback through signal acquisition, processing, and feedback (NCAN, 2023). ................................................................. 5

Figure 2. An example definition of a waveform, with a train of pulses. This specific figure is referencing stimulator parameter names defined by the NeuroOmega, however the shape can be applied to many other stimulators ................................................................. 7

Figure 3. The Stimulation Configuration Tool in comparison of each native device’s software... 8

Figure 4. Stimulation configurations for the simulated data versus recorded data with an oscilloscope. From top column to bottom, it is the CereStim, g.Estim, and NeuroOmega. ........ 14

Figure 5. Average round-trip time latency for stimulators. The error bars are standard deviation, which represents the jitter of the latency. ................................................................. 15

Figure 6. Informational wiki pages for various devices that were implemented in BCI2000 ..... 16

Figure 7. A comparison between BCI2000’s real-time visualization of the raw data (top) and the CCEP filter (bottom)........................................................................................................................................................................ 18

Figure 8. A demonstration of the USB ports turning on and off due to an expression in BCI2000. ........................................................................................................................................................................ 19

Figure 9. An example experiment conducted in BCI2000, with both the experimenter and participant’s screens displayed. The experimenter is using GazeMonitorFilter, visualizing State watches, and using the real-time visualization of the source to have full real-time knowledge of the experiment ........................................................................................................................................................................ 20

Figure 10. A visual comparison of stimulation trains between MATLAB and BCI2000. The large jitter occurs with the fifth train in the MATLAB version ........................................................................................................................................................................ 25
Acknowledgments

This work would not have been possible without the guidance of Dr. Peter Brunner and the assistance from the rest of my fellow lab members in the Brunner Lab, especially Jürgen Mellinger and Alexander Belsten. In addition, the collaboration with Drs. Sudhin Shah, Enrico Opri, Jon Willie, Cory Inman, the Cortec team, and the EGI team is much appreciated.

William Engelhardt

Washington University in St. Louis

May 2023
ABSTRACT OF THE THESIS
A General-Purpose Software Platform for Closed-Loop Neuromodulation

by

William Engelhardt

Master of Science in Biomedical Engineering
Washington University in St. Louis, 2023

Associate Professor Peter Brunner, Chair

Implementing closed-loop neuromodulation therapies is a challenging and expensive endeavor. It requires developing software capable of acquiring signals from a bio-signal amplifier, analysis of these signals, and initiation of precisely timed stimulation, all of which need to be accomplished in real-time with very low latency. Developing this software is difficult, as it requires a wide range of expertise ranging from interfacing with hardware to real-time signal processing. Even when successfully implementing such a system for one set of hardware, it often then only works within the laboratory that conceived it. This is because of the inherent heterogeneity in the devices that realize interactions with the nervous system, and the lack of standardized interfaces to access and control them. Collaboration thus often necessitates acquiring the same hardware (i.e., amplifier and stimulator) across all sites, which can sometimes be cost-prohibitive. Implementing software that uses these amplifiers and stimulators within a real-time acquisition and feedback software platform, such as BCI2000, would eradicate these obstacles. Multiple stimulators, amplifiers, and software devices have been implemented to provide a maximum amount of real-time feedback and to make the configuration effortless. With these improvements, closed-loop neuromodulation experiments have begun, and are only the beginning of what is possible.
Chapter 1: Introduction

The field of neuro modulation is currently one of the most tantalizing subjects in scientific forums. This is mainly due to its promising future and relatively short history. Throughout this short history, much progress has been made. One of the first modern-day occurrences of electrical stimulation of the brain was done by Dr. Roberts Bartholow in 1874, where a patient with a purulent ulcer of the scalp with skull osteomyelitis had an exposed area of the brain after debridement, and stimulation was applied (with a homemade stimulation device), resulting in the response of a muscle spasm. Similarly in 1909, Dr. Harvey Cushing stimulated an awake patient in the post-central gyrus, which showed contralateral muscle movement (Gildenberg, 2009). These beginnings demonstrated the ability of stimulation to affect nerves throughout the body, which has led to the development of neuroprosthetics and brain-computer interfaces as scientific fields of investigation.

Throughout the 1900s, stimulation was used in various brain regions to conduct functional brain mapping (Penfield and Jasper, 1954) and to better understand the underlying physiology (Spiegel, 1947). This was the beginning of the now widespread practice of using localized stimulation to pinpoint cortical onset zones of seizures, which assists in determining the resection area in the surgical treatment of intractable epilepsy.

Many of these stimulation devices were made by the operating physicians, for there were no companies to mass produce stimulators. This freedom allowed for new discoveries, such as noticing stimulation’s improvement of symptoms of Parkinson’s patients (Gildenberg, 2009). However, this freedom came at a cost. Without keeping physicians accountable for causing patients pain, experiments could be tried that had a high chance of failing. For example, Dr. Riechert caused an accidental seizure in a patient by stimulating their brain at 50 Hz,
demonstrating that this frequency is harmful (Riechert, 1980). This balance of having the freedom to try new things yet maintaining patient safety is a difficult line to follow, and our modern widespread solution has been to limit freedom in exchange for keeping a patient’s safety a top priority.

This solution is enforced by the FDA, which approves all new medical devices and prohibits misuse of existing ones. Although the FDA has been very effective in ensuring device safety and minimizing harm to patients, it inherently limits the medical device industry to remain within the designs of what is FDA-approved. The principal approach for this approval process is implemented through the 510k approval process. If any change beyond the scope of the original predicate device is desired to be made, the applying company must go through a laborious process of proving the safety and efficacy of the new device. Unless this modification is going to vastly change the device, it is usually easier to remain within the envelope of the predicate device, even if this means that the new device might be less effective than desired.

As the neurotechnology field has undergone substantial growth over the past two decades, companies have jumped at the opportunity to be involved in the medical device industry. This has led to the standardized production of robust amplifiers and stimulators and has empowered physicians to simply buy the device they need instead of manufacturing it on their own. Companies usually focus on certain areas relevant to device development as their core competency, as it is highly profitable. To distinguish themselves from competitors, companies tend to vary from one another in both their hardware and software.

As physicians become more accustomed to specific devices, they are inclined to remain with the same company for future devices and become more and more specialized in the company’s conventions. However, this specialization limits the breadth of scientific
experimentation, as the devices that have new applications would feel alien and difficult to integrate into the physician’s workflow.

Not only is it hard for a physician or research laboratory to switch companies, but it also makes it very difficult to collaborate with other institutions that use different systems. The data files will be different, the configurations will vary, and the experimental findings will be hard to compare.

The specialization of companies also tends to be directed towards the hardware side, which is proper given the application. This focus and expertise in the hardware side create robust products from an electrical engineering point of view. However, it can also leave out development in other areas. Software, where the ease of use for the user should be prioritized, often remains an afterthought rather than an essential component of the development process. For example, error messages are often not helpful enough to assist the user in solving the problem on their own. Instead, they are so convoluted that one would need help from the company itself. This is often seen in the operating room, where the moment a device doesn’t work properly, the company representative is called.

Lastly, another problem in human subject research involving neurotechnology is that the experiments can be very complex and demanding, making them highly prone to user error. For most data analysis of neurological signals, precise timing is key to allow synchronization of various signals. If many devices are used for an experiment, such as an eye-tracker, a camera, and an electroencephalogram (EEG) recording, synchronizing these is a challenging task. Especially with different sampling rates (e.g., eye-trackers with 150 Hz, camera with 24 Hz, and EEG with 2000 Hz), aligning the data samples is a cumbersome task, and requires forethought as to how this will be accomplished. However, when the problems are realized only after the
experiments when the data is being analyzed, it is impossible to properly align the signals from the different devices. Also, the neural activity is relatively small in amplitude, and the recording of such data can be easily overwhelmed with noise. Such noise is often only noticed during the analysis of the data post-experiment, which results in discarding the entire recording for this patient.

This plethora of issues is addressed by BCI2000, a general-purpose software platform for the implementation of brain-computer-interface experiments. It allows the user to abstract their experiment from the device they use, which allows for a variation of devices to be used. The data output is the still the same, regardless of the devices used. Not only does this make it simpler for data analysis, but it also allows for collaboration between institutions. BCI2000 has real-time visualization, which allows for the data quality to be checked before the experiment has started and allows the researcher to make the necessary adjustments to ensure the quality of the recordings. It also has an intuitive interface and adaptable configurations, allowing for real-time changes. Additionally, BCI2000 is set up to enable closed-loop experiments, with real-time analysis and feedback. Figure 1 displays the system pipeline of BCI2000.
However, BCI2000 is only as powerful as its implemented features. It has existed for over 20 years, with each year’s development building on the past. The most recent implementations involve stimulators, amplifiers, peripheral devices, and software tools that greatly improve BCI2000’s capabilities. Each improvement solves a previously mentioned issue, and the way that was done will be discussed. Overall, the configuration, implementation, and visualization of neuromodulation devices and peripherals in BCI2000 have benefitted physicians, scientists, and patients by allowing for rapid prototyping, lowering the barrier to entry, and improving data collection.
Chapter 2: Methods

2.1 Configuration

A main dogma of the scientific method is the attempt to standardize everything to allow for clear communication, objective results, and easy comparison. However, when dealing with real-world companies, science is not the main driving factor. Instead, companies create devices that stand out, are novel, and have their own specific value proposition. This results in a vast number of devices that each have their own specific application and stand out where other devices fall short. The academic solution to all the variation of these devices is to pick one and go all-in. This works well enough, however limits collaboration between the institutions that have chosen the different companies. Ideally, one could have the standardization of these devices, while keeping the variation that each device has. This is what BCI2000 is solving, one stimulation device at a time.

The idea behind the interface of the implementation of the devices in BCI2000 is to keep them as close to the device’s native software as possible. This way, when collaborators want to switch from the native software to BCI2000, the parameters are intuitive, and the switch is easy. For example, the frequency of pulses and the time in between pulses give the same information. If the native software prompts the user for the frequency, then the implementation in BCI2000 should do the same.

Then behind the interface, with the inner workings of BCI2000, the implementation should be as similar as possible between the devices. Even though the devices have different application programming interfaces (APIs), the differences can be minimized. Figure 2 shows an example waveform on the left and the stimulation train on the right. A typical stimulation pattern looks similar to the shown waveform and usually incorporates a stimulation train.
Figure 2. An example definition of a waveform, with a train of pulses. This specific figure is referencing stimulator parameter names defined by the NeuroOmega, however the shape can be applied to many other stimulators.

Although stimulation parameters need to produce a waveform exactly as intended, the nomenclature and the complexity of it can be confusing. This is why Stimulation Configuration Tool was created, to visualize the stimulation that will be produced, and then directly exported to BCI2000 (Figure 3). This further standardizes the stimulators, by using inputs similar to their native software, then visualizing them and exporting them in a standard format.
2.2 Implementation

The main advantage of implementing devices in BCI2000 is that they can be fully integrated into the BCI2000 signal acquisition and processing pipeline. This includes real-time feedback and visualization, the adaptable configuration of parameters, programmatic stimulation, normalized data output, and integration with peripheral devices. As discussed earlier, these features solve many problems that are present in many experiments involving neurotechnology. However, for this process to be harnessed fully, the implementation of the amplifiers and stimulators needs to be precise and robust.
When using stimulation for the brain, trains of pulses are commonly desired. A pulse is a single waveform designed to excite neurons. Multiple pulses, with varying shapes and frequency, can then elicit a unique response, based on their shape, frequency, and duration. And lastly, you can turn on and off these sections of pulses at some lower frequency, to systematically continue the stimulation (as shown in Figure 2). These trains of pulses must be precisely triggered at the specified frequency, and usually, the device does not keep track of stimulation trains. Effectively, this means that when this functionality is desired, it must be done manually in BCI2000 to allow for the automatic triggering of the stimulation train.

BCI2000 has a unique system architecture, where all incoming signals are processed in blocks. The duration of a block can be configured, depending on parameters set by the user, but usually range from 20 to 100 ms. Each block processes a certain number of samples for that duration, based on the device’s sampling rate. For example, for a block size of 20 ms and an acquisition system that has a sampling rate of 5 kHz, there will be 100 samples processed per block (per 20 ms) because 20 ms * 5 kHz = 100. This architecture of relying on a block size for processing data allows for real-time feedback, to the exact resolution that one would need it for.

However, when stimulation needs to be triggered outside of this block-timing pipeline within a precision of a millisecond, a different, more asynchronous solution needs to be developed.

The ideal solution for this problem is to create a parallel thread, where the timing can be fully controlled without blocking BCI2000’s pipeline. This thread is necessary for stimulation trains, which, once triggered, must occur at a specific frequency and duration. BCI2000 block sizes are in the magnitude of tens of milliseconds, while stimulation trains are in the scope of a few seconds. The train can be triggered by some expression in BCI2000 (that the user sets), and
then the thread waits until the next train needs to be triggered and does so accordingly. This parallel thread is run alongside the main BCI2000 thread, where they can interact but not get in one another’s way.

Another rationale for the necessity of a parallel thread is the occasional latency and blocking of an API call. When sending a call to the device, it can sometimes take a while for the device to register the request and send a response back. This could potentially block BCI2000 while the request is waiting to get a response. However, if this is done in a separate thread, the main BCI2000 timing is not perturbed.

2.3 Visualization

BCI2000 shows the real-time data in a visualization window, where it is updated every process block. This generates a stream of raw data, where one can add high-pass or low-pass filters to best visualize the signal. This solves the problem of not knowing about the quality of data, for noise in the data would immediately be seen in the visualization signal. However, for stimulation, the signal can be hard to see.

When recording and stimulating at the same time, the problem is that stimulation drowns out any neural activity because of the difference in magnitudes. The stimulation is so much larger, that it makes the real-time visualization of the signal almost unreadable. The BCI2000 signal visualization is also updated every process block continuously, but the response from a stimulation could be on the order of tens to hundreds of milliseconds. Therefore, it would be ideal to have a time-locked visualization of the source after stimulation has occurred. This time-locked signal is ideal for viewing evoked responses, particularly cortico-cortical evoked potentials (CCEPs).
From an experimenter’s point of view, the important information is which channel is being stimulated (at which area of the brain), and what the response is. If the visualization shows that the signal is saturated, the setup can be changed immediately before the experiment proceeds. To present this information best visually, the CCEP filter separates the channels into a grid with each individual channel. The channel that is being stimulated is clearly shown that is the stimulating one, and then the channels that have some responses are also identified by color. A response from a channel is determined by if the activity goes beyond some threshold after stimulation, which is customizable by the user. However, the stimulation artifact will be larger than any neural response, so there must be some way of ignoring the stimulation artifact. The way this is done in the CCEP filter is by having some latency, specified by the user, in which the magnitude is ignored before, and only considered after.
Chapter 3: Results

3.1 Validation

3.1.1 Configuration

To validate the functionality of the Stimulation Configuration Tool, it was validated against the ground truth of the stimulator output. This is shown in Figure 4, for each simulator that is implemented in the configuration tool. The pulse shapes had to be slightly different between stimulators due to the device's limitations. However, the pulse shape was kept constant across the three views. Each viewpoint validates a different aspect of the stimulation and represents how the stimulation configuration tool is set up. The first shows that the pulse shape is the same, the second validates that the pulses in a train are aligned, and the last displays that the trains are triggered at the proper time.

Each pulse was measured with an oscilloscope, measuring the difference in voltage over a resistor. Since the configuration tool displays the stimulation in current units (mA), it was converted by the resistance to voltage units. Each pulse delivered an amplitude of 5 mA, had 50 pulses in a stimulation burst, and had 2 stimulation trains. Additionally, for the CereStim waveform, the pulse width was 2 ms, the interphase duration was 4 ms, the stimulation bursts were at 50 Hz for 1 s, and the stimulation train was at 0.5 Hz. The g.Estim waveform had the same stimulation bursts and stimulation train frequency and duration, however, the pulse width was 0.2 ms, and the interphase duration was 0.4 ms. The NeuroOmega waveform had a pulse width and interphase duration of 0.5 ms, had pulse bursts at 200 Hz for a duration of 0.25 s, and had a stimulation train of 2 Hz for 1 s.

As for how stimulation configurations are implemented in BCI2000, they are separated into two tables, one for the configuration and one for the triggers. The configuration table
contains all the parameters that define the pulse shape and size, including the train definition. The table for the triggers contains information for which expression starts the stimulation, as well as the information for which electrodes will be stimulated. Organizing the parameters this way isolates the configuration from the more logistical elements. This also allows one to use the same configuration for multiple electrode set-ups, instead of having to rewrite the configuration for each electrode. That would be a cumbersome task, and prone to the error of not having the same exact configuration for each different trigger. Lastly, this setup allows for the loading of multiple configurations, to be able to programmatically vary them during the same run.
Figure 4. Stimulation configurations for the simulated data versus recorded data with an oscilloscope. From top column to bottom, it is the CereStim, g.Estim, and NeuroOmega.
3.1.2 Implementation

The integration of each stimulator in BCI2000 has the goal of abstracting the stimulator from the experiment. However, there are hardware limitations that vary between devices. One such metric that varies is the round-trip latency for stimulation. This is measured as the time between the onset of the trigger in BCI2000, to the measurement of the start of the stimulation in BCI2000. The results are shown in Figure 5, where the error bars represent the standard deviation. The standard deviation can be interpreted as the jitter of the latency, which is also an important metric for closed-loop stimulation.

![Figure 5. Average round-trip time latency for stimulators. The error bars are standard deviation, which represents the jitter of the latency.](image)
As can be seen, the magnitude of the GTEN stimulation is almost 10x higher than for the other stimulators with a latency of close to 1.2 s. In contrast, the CereStim and g.Estim both exhibit a latency in the 10s of milliseconds. This latency is important to consider when choosing a device, as well as the other features of the device.

Since all the stimulators and amplifiers are unique, multiple variables must be considered when choosing one to use in an experiment. In fact, a lot of the devices give specifications that other devices don’t touch on, making comparisons very difficult. As an overarching solution to that, as part of the implementation in BCI2000, there is an emphasis on documentation of the devices. The main documentation is present on the BCI2000 website, in which each device has its own page, which has similar formats to easily compare between devices.

Figure 6. Informational wiki pages for various devices that were implemented in BCI2000.
3.1.3 Visualization

To best analyze stimulation in real-time, the CCEP filter was created. This filter generates a small figure for each channel that is desired and shows a time-locked signal of the response after stimulation. To allow for various types of devices, parameters are customizable. However, to create the user experience, certain devices are implemented as “filters”, which automatically determine the best parameters for that device. These implemented devices also determine which channel is stimulating, and which use the parameters specific to that device.

The visualization of the CCEP filter also provides markers that orient the user to the status of the stimulation. The CCEP filter provides three sections of data to the user within the time-locked signal: the baseline data before the stimulation, the area with the stimulation artifact, and the time after the stimulation containing the possible neurological response. There are two markers in-between the three sections, which show the respective areas for the user.

Even though visualizing the time-locked data in real-time is helpful for the user, the CCEP filter also shows the channels that show a response to the stimulation. This is determined by exceeding some threshold determined by the user. The specified channels change color, which makes it easy to pick out from the array of all the figures.
Figure 7. A comparison between BCI2000’s real-time visualization of the raw data (top) and the CCEP filter (bottom).

3.2 Specific use cases

The following paragraphs present how solutions were devised that address the relevant scientific problems, with specific use cases of these solutions.

3.2.1 PegasusAstro

PegasusAstro has a USB control hub that allows for programmatic control of up to 6 USB ports. This feature can be used in areas where frequent unplugging and plugging back in of
a USB device is necessary. This is often used when a device needs to be fully disconnected to ensure patient safety. Not only would manually powering cycling devices be laborious, but it also can’t be precisely time-locked to anything, and prone to error (e.g., which port to unplug, how often the devices should be unplugged).

Implementing this in BCI2000 allows for programmatic switching/powering of USB ports where the USB port turns off according to an expression in BCI2000, as seen in Figure 8. This allows for increased efficiency in experiments and less risk of error.

![Figure 8. A demonstration of the USB ports turning on and off due to an expression in BCI2000.](image)

### 3.2.2 EyeLink

The EyeLink 1000 Plus is a precise and high sampling rate eye-tracker that records many eye states, such as gaze position, pupil size, and eye validity. Eye-tracking can be useful in many experiments by comparing the data with the neural activity and/or the stimulus presented on the screen. Tracking the gaze data allows the researcher to determine when the patient is looking at the screen, when they saccade, and when they blink. However, this can only be collected if properly calibrated and with low movement of the patient. In reality, the patient moves a lot, and the gaze data is unable to be tracked. The problem is that this is often detected during the analysis of the data, rendering that patient as unusable.
The implementation of the eye-tracker in BCI2000 allows for communication with the GazeMonitorFilter, which shows the real-time gaze of the eye. There are also states that can be monitored by the researcher, as well as alerts that show when the fixation is kept or lost. Experiments can even be paused if the fixation is lost and can be continued once the fixation is regained. This setup can be seen in Figure 9, where the experimenter’s monitor and the participant’s monitor are shown. One can see that the participant’s monitor is simple, only showing the necessary parts. The experimenter’s screen can show all the collected data in real-time, which includes the eye-tracker states, the EEG signal, and the gaze location of the eyes.

Figure 9. An example experiment conducted in BCI2000, with both the experimenter and participant’s screens displayed. The experimenter is using GazeMonitorFilter, visualizing State watches, and using the real-time visualization of the source to have full real-time knowledge of the experiment.

Dr. Sudhin Shah’s laboratory at Weil Cornell Medicine was conducting an experiment where the aforementioned problem happened, and the gaze data couldn’t even be analyzed because it was too inconsistently recorded between patients and had large gaps of time where the
eyes weren’t being tracked. The best they could do was calibrate at the beginning of the experiment, and hope the patient didn’t move, but this often wasn’t good enough. This was solved by collecting the gaze data in BCI2000, where it is seen in real-time when the gaze is lost. This can then be immediately fixed, and the experiment continued.

Another feature of the EyeLink system is the format in which it uses to export data. When collected with EEG signals, the two sources are recorded separately, then must be aligned post-experiment. There are two main complications to this procedure, 1) the eye-tracker and the EEG signals have different sampling rates, which means a different number of samples during the same period, and 2) the two sources need to be aligned, which is hard to do, especially when this need is realized after the fact.

All these problems are automatically solved in BCI2000. The eye-tracker data is up-sampled to match the same number of data as the EEG source, and the data is already aligned in the output file of BCI2000.

During the same experiment conducted by Dr. Shah’s lab, the eye-tracker data was also recorded in the Eyelink native application, which recorded data at a separate sampling rate and a different data file. When trying to analyze this data and align it to the main brain signals, it was so difficult that it bordered on impossible, and the analysis was given up. If it had been done in BCI2000, this data collected from patients could have been easily analyzed.

3.2.3 XSens

The Motion Tracking wireless Inertial Monitoring Units (IMUs) from XSens are wireless accelerometers that track motion. Up to 32 can be combined and recorded from at one time, which allows for powerful data analysis. Similar to the eye-tracker, these devices are collecting data at a different sampling rate than the main neurological data. The combination of these two
signals is hard to analyze post-experiment, and unless proper care is taken before and during the experiment, can be impossible to align exactly with the brain signals.

Just like the eye-tracker, this problem is taken care of in BCI2000. These XSens IMUs can be used in Parkinson’s patients during deep brain stimulation (DBS) surgery, to quantify the amount of tremor for the different stimulation amplitudes. The current method is to try different amplitudes and give the patients tests at each amplitude step. This method is time-consuming and only tests approximately three amplitudes. If the quality of the trials was measured by the IMUs in BCI2000, any experiment using these devices could become suitable for closed-loop operation, allowing for the timing of the delivery of stimulation to a particular phase of movement as captured by the IMUs.

### 3.2.4 NeuroOmega

The NeuroOmega system is used in DBS surgeries, as it records neural activity from the leads that are being implanted and controls the delivery of stimulation. To visualize the raw signals, the system shows a real-time recording of the data, however, it is hard to interpret anything from this visualization, as it is not adaptable and changes very quickly. If stimulation with the device is desired, it must be done manually, and the settings adjusted on the spot. It is also possible to conduct the stimulation remotely through their API, but this has its own problems. It is difficult to precisely time the stimulation and would take a lot of work to create a system that has real-time feedback and visualization.

These problems are solved by the implementation in BCI2000, where the visualization of the data is flexible and allows you to see the quality of the recorded signal. The stimulation is accurately timed and aligned with the BCI2000 pipeline. The stimulation can be
programmatically initiated with little risk of user error (especially with the configuration tool) and done beforehand, and lastly, the stimulation can be visualized with the CCEP filter.

These issues and software limitations in the native solution as implemented in previous work conducted by Dr. Enrico Opri, were addressed within this BCI2000 implementation. He conducted his stimulation through their API with MATLAB, and only had the native visualization of the data. Unfortunately, the timing through MATLAB is not as precise as through C++, which is done in BCI2000.

Secondly, after implementing the experiment in BCI2000, multiple problems were discovered that would have otherwise been only noticed after analyzing the data post-experiment. The first problem was having a high level of noise in the signal, which drowns out the neurological responses. This was found in the real-time visualization in BCI2000 and was used to resolve noise issues with the setup before initiating the experiment. Without the visualization, the data would have been collected with noise, effectively rendering it unusable.

Having NeuroOmega in BCI2000 also lowers the barrier to entry for experiments, resulting in a greater number of experiments being created with it. Dr. Kai Miller at the Mayo Clinic started using it in a closed-loop experiment with a cursor task, and it is also beginning to be used by Dr. Luis Manssuer at the University of Cambridge.

3.2.5 CereStim

The CereStim is a programmable neural stimulator that can be controlled through its native software or triggered remotely with its API. The native software is designed for open loop operation, where one would press a button to initiate stimulation. The API allows for closed-loop stimulation. However, real-time, precise timing and processing would need to be implemented.
Usually, the API is used to simply trigger stimulation, essentially completing the same task as the native software.

BCI2000 allows for closed-loop neuromodulation by incorporating the CereStim in its real-time processing and feedback system. Specific neurological behavior can trigger stimulation, which is sent to the device with precise timing. The CereStim also has low round-trip latency, which allows for close to immediate stimulation as soon as it is desired.

Dr. Cory Inman of the University of Utah and Dr. Jon Willie, a neurosurgeon at Washington University in St. Louis, were conducting a study to measure the effect of amygdala stimulation on the formation of declarative memory (Inman, 2018). These findings were published, and a continuation of this study is being continued with epilepsy patients at Washington University in St. Louis. In starting to collect data from patients here, the experiment was ported into a BCI2000 experiment. In doing this, some problems were discovered, and the experiment was made more precise.

One problem that was observed was the jitter in delivering the simulation using MATLAB. As seen in Figure 10, there is a large amount of jitter in-between the stimulation trains with the MATLAB version, which has been resolved by implementing the same functionality within BCI2000. This comparison could have been further analyzed, but since the data was already collected in MATLAB and will be continued in BCI2000, it would not have made a difference.
3.2.6 Cortec

The implantable Brain Interchange Communication (BIC) unit from Cortec is used for both stimulating and recording intracranially and wirelessly transmits the data to a receiver attached to the outside of the head. The implantable device is charged inductively and is designed to be implanted long-term.

The hardware on this device is very reliable and meets the challenge of having an implantable device last long-term. However, the software side of the company and the device is not at the same level of expertise. This reveals itself through meaningless error messages, a faulty connection process, and an unintuitive interface with the device.
However, if the hardware is there, BCI2000 can easily take over the software side. The stimulation is implemented as the other stimulation devices, and a lot of the problems are hidden behind the interface, so the user doesn’t have to deal with them.

This implementation in BCI2000 was robust enough that it was used to record data and check the quality of the signals in real time during the pivotal implantation trial of the device in a canine a couple of weeks ago. This implantation, and future ones, are part of a grant protocol that shows the safety and efficacy of the device long-term.

Implementing the BIC unit in BCI2000 also helped identify and resolve problems that existed that the Cortec developers weren’t aware of. By using their API and trying to replicate and improve the features in their software, there were a couple of problems that were identified. For example, the connection process was very unreliable and took a long time, occasionally resulting in a permanent state of not being able to connect that could only be solved by restarting the computer. After days of discussing and troubleshooting with the Cortec team, it turns out the problem was that the device was pulling too much power from the USB port and causing it to turn off. Switching the USB port to the back of the computer, a more robust port resolved the issue. This problem can now be fixed in the next hardware generation and documented in the meantime for the use of current generation devices.

3.2.7 Stimulation Configuration Tool

Stimulation parameters can be confusing, and cause experimenters to be uncertain about what waveform they are truly delivering, or to run into unknown errors in their waveform that are only discovered in the analysis of data. The parameters are also varied per stimulator, and it is hard to compare between stimulators. Manually putting the parameter numbers is also prone to
error, as they might be in the wrong units, or be mis-transferred over from the creation of them to the experiment.

The Stimulation Configuration Tool is created for creating and visualizing the stimulation configurations before the experiment. Visualization of the waveform helps immensely with figuring out exactly what shape the stimulation takes and can show comparisons between multiple stimulation configurations. The tool can also load parameter files, change them, and then export them, which can be loaded directly in the experiment. This gets rid of any risk of error of accidentally changing parameters as they are changed between formats.

Before the configuration tool, loading stimulation parameters into BCI2000 was difficult to validate. For example, in the previously mentioned research with Dr. Opri, transferring the parameters from the existing experiment in MATLAB to BCI2000 was too tedious to do by hand, and a complicated script had to be created. This script was very prone to error, and in fact, caused the first experiment to have wrong stimulation waveforms. However, once the tool was created, the parameters were imported, and checked visually, and some redundant configurations were found.

Another example of this tool being useful is regarding the Epilepsy Monitoring Unit (EMU), where a patient was staying longer than expected and could do additional research. A new experiment was desired with a new stimulation configuration. The tool was used at the bedside to create the new waveform and was used in an experiment the following day.

3.2.8 EGI

The company EGI has Net Amps GTEN 200, a transcranial electrical stimulator and an amplifier. This device has similar problems as the previously mentioned ones, most notably how only open-loop stimulation can be done.
When implemented in BCI2000, closed-loop stimulation can be done. This feature is currently being taken advantage of Dr. Shah and Dr. Samuel Louviout, who are conducting an Attention Network Task using the EGI system. They will be using BCI2000 to measure the predominant alpha wave frequency of the subject, then will stimulate with that frequency to attempt to modulate it. This pipeline was easily set up in BCI2000.
Chapter 4: Discussion

4.1 Configuration

Discussing the results from Figure 4, the results appear overall aligned, however, there are some discontinuities between the simulated waveforms and the real data.

For the Cerestim, the pulse shape has the greatest variation. The waveform has the same shape, but when the amplitudes are required to be held for the duration of the pulse, the amplitude of the stimulus drops throughout each pulse. Since the impedance was a single resistor, it shows that the device has some capacitance, or the voltage was close to the compliance voltage. This is useful for knowing exactly what the waveform looks like, but for the purposes of validating the parameters of the configuration tool, it shows they are what one expects. The capacitance element of the waveform is also affected by what is being stimulated, which will cause varying effects. As for the second figure, all the peaks are aligned. And for the last figure, the peaks are all within the “on” duration of the stimulation train.

For the gEstim, the two pulse shapes match up exactly. And like the Cerestim, the pulses are aligned, and they are within the duration specified for the train.

Looking at the NeuroOmega figures, the pulse shape has the same duration, but the amplitude has a slight variation. However, this is variable over varied frequencies and durations and has the same shape as the simulation. The interesting discovery is that the second graph indicates an apparent discrepancy. As the pulses go on over time, the alignment of the pulses becomes mismatched, almost to the point of alternating between the simulated and real pulses. After analyzing the timing between each pulse, it is seen that the simulated pulses stay on the 200 Hz frequency, while the real data is slightly slower. The measured data (yellow) has
approximately a frequency of 196 Hz. This is worrisome for experiments using high number of pulses needing an exact frequency, but for most applications, this resolution is acceptable.

The stimulation configuration tool’s unique abilities, such as loading parameter files, changing them, and then exporting again, allow for validation of the stimulation configuration before the experiment is run. Loading an existing parameter file allows for the user to check that their stimulation configuration is what they desire. Changing the parameters with the tool, while being able to immediately visualize the difference, lets the user fully understand the changes they are making and the effects they have on the final stimulation. Then to limit any user error in transporting the data from the configuration tool to BCI2000, the parameters can automatically be exported from the tool, and then loaded directly into your experiment. This automation removes any potential for user error and streamlines the process of delivering effective stimulation.

Although this configuration tool is helpful for visualizing configurations, there is a potential for uncovering greater capabilities. One such capability that could be implemented is the creation of stimulus sequences in the tool, which are commonly the triggers for the stimulation. With this, the entire run of stimulation could be visualized, so one could be sure that the stimulation is occurring exactly when desired.

Another drawback of the tool is that it is only configured for 3 stimulation devices currently (CereStim, NeuroOmega, and g.Estim). These are currently the three most used stimulators in BCI2000, however two other stimulation devices are also available: the GTEN system from EGI and the BIC unit from Cortec. It would be helpful for the new users that will be using these devices to be able to configure their stimulation with the configuration tool.
As seen in Figure 4, the simulated waveform can sometimes be slightly different when compared to the real data. This could be another feature of the tool, to try to incorporate real device properties in the simulation of the tool. For example, this could be done by making the user input the intended resistance, and confirming it is below the measured compliance voltage of each device.

4.2 Implementation

The use of stimulators in BCI2000 allows for the programmatic use of stimulation, as well as closed-loop modulation. Due to the low latency for most devices shown in Figure 5 and the closed-loop nature of BCI2000, experiments can be configured to modulate any of the stimulation parameters, such as frequency or amplitude, based on the recorded data from the amplifier. Without implementation in BCI2000, this would take too large of an overhead that most people would not go through the trouble.

Other than the closed-loop nature, BCI2000 also solves the problem of having various sample rates, with multiple devices that aren’t aligned. BCI2000 aligns the data from various devices in real-time to the amplifier as the ground truth. This allows for easy analysis post-experiment, where the alignment of that data doesn’t have to be worried about. Specifically for the stimulators, there are markers that signify when the stimulation occurred.

The implementation of multiple stimulators allows for exchanging devices without requiring any changes to one’s experiment. If two institutions are desire to run the same experiment, but one has a CereStim, and the other has a g.Estim, the major change they would have to make is changing “--EnableCereStim=1” to “--ActivateEstim=1” in the batch file. Additionally, some of the parameters would have to fit the format of the g.Estim, but the only difference in the specification of the parameters between the two is that the CereStim is
configured with a specific train duration, however, the g.Estim has the number of trains specified. After these changes are made, the experiment can run as the original institution intended, without fear of a difference in the implementation of the experimental paradigms.

The implementation of stimulators in BCI2000 is a recent development, and the number of stimulators that are supported will continue to increase. As more and more are implemented, the availability of the closed-loop nature of BCI2000 can become a reality for more and more users. The customizability of the application allows for a large variety of experiments to be incorporated, and if the stimulator is implemented, then a wide range of opportunities becomes available for the user.

4.3 Visualization

BCI2000 already has an intuitive visualization system, where one can make changes to the visualization of the raw data to best view it. However, these changes do not affect the saved raw data. This real-time visualization is helpful for viewing the signal quality, and the response on various electrodes, and for confirmation that a signal is being collected. This is already better than most applications collecting data, however, for stimulation, it is not as helpful as it could be.

The CCEP filter can allow for the real-time adjustment of stimulation parameters, which can lead to the customization of closed-loop neuromodulation. The way this could work is by using the CCEP filter to determine which channels have a response to certain electrodes and then using that information to isolate the channels that seem to be functionally connected to conduct brain-computer interface experiments. Obtaining this feedback in real-time can lead to adaptable experiments which focus on individual treatment instead of just basing the experiment on a one-size-fits-all model. This leads to improved treatment options, happier patients, and increased scientific discovery.
Although the CCEP filter greatly improves the visualization and real-time understanding of the response to the stimulation, some future changes could be made to improve where it is currently at. One such improvement could be the classification of single neuron response from a local field potential recording, based on the CCEP response magnitude. A single neuron will always have the same magnitude for a given electrode placement, so it would be possible to define how the system behaves even further. Another area of improvement would be a visualization of the overall CCEP response at the end of the run, where one could see a mapping of which electrode affected which other channels. Some basic processing of the cause-and-effect of the channels could create an initial functional mapping, where it could be seen how all the channels are related.
Chapter 5: Conclusion

BCI2000 creates a working framework for brain-computer interface experiments. However, its horizons are greatly expanded by the new additions of stimulation devices and updated visualization. The application was once mainly used for the control of interfaces with trained brain signals, but the new implementations can seamlessly allow for closed-loop neuromodulation. This allows for improved treatment, such as improved therapy for DBS of patients with depression (Scangos, 2021), and for many more applications that have yet to be discovered.

For the purpose of this thesis, I will briefly explicitly declare the implementations that I developed. I finished the implementation of the PegasusAstro USB hub and the XSens IMUs which were partially implemented, and fully made their wiki pages. Regarding the NeuroOmega, I fully implemented the stimulation feature (including obtaining stimulation markers from the device) and increased the robustness of data recording. I added the capability for stimulation trains in the CereStim, which helped develop the format for the rest of the stimulators. I fixed the alternating current stimulation for the EGI implementation. For the Cortec device, I implemented stimulation trains, updated the implementation to their new API, and improved the robustness of data collection. I fully designed and developed the Stimulation Configuration Tool with MATLAB’s app designer. I developed the CCEP filter with the foundation of another filter that Jürgen Mellinger made. I made various small changes to many devices, some of which haven’t been mentioned insofar. These include the g.Estim, GazeMonitorFilter, g.EstimProSwitchingUnit, NatusServer, EyetrackerLogger, and TuckerDavis. For all these changes, I updated the documentation accordingly, including tutorial videos for the XSens,
EyeLink, and NeuroOmega. I also conducted the testing of each of these changes to ensure validity, acceptable block timing, and compatibility in the BCI2000 pipeline.

With capabilities for closed-loop neuromodulation, as well as rapid prototyping through the stimulation configuration tool and real-time visualization of neurological responses, the future possibilities of experiments are endless. More complicated paradigms can be constructed and made clear through the available tools and visualizations. These progressions advance the field of science by helping researchers further understand the connectivity of the brain and the responses it produces. Just as new insights were originally discovered from having complete freedom of stimulation, these rapid adjustments can once again be made with BCI2000. Through the simple configuration of stimulation parameters, implementation of these devices in BCI2000, and the real-time visualization of the neural activity, revolutionary discoveries of the brain can once again be found.
References


