

**CEBL₃: A Modular Platform for EEG Signal Analysis and Real-Time Brain-Computer Interfaces**

*Elliott Forney and Charles Anderson*

---

**Non-Invasive Brain-Computer Interfaces**

Brain-Computer Interfaces (BCI) are systems for establishing a direct channel of communication between the human brain and a computerized device. BCI utilize a communication protocol along with signal processing and machine learning algorithms to convey the user's intent to a computer system. In our lab, we use Electroencephalography (EEG) to monitor brain activity because it is non-invasive, portable and relatively affordable. An important application for BCI is in the development of assistive technologies for people with severe motor impairments. For those who find it difficult to communicate or perform everyday tasks, even a somewhat slow BCI may prove to be an invaluable tool.

**CEBL₃:** A complete re-write of our BCI software platform. Currently available BCI software systems fall short in one of two areas. Either:
- They are not well-suited for real-time experiments
- Or they are not easily extended to new paradigms and methods.

CEBL₃ addresses these issues through the use of high-level programming languages and a modular design approach that promotes code reuse throughout the entire process of developing BCI prototypes. Additionally, CEBL₃ includes a number of state-of-the-art signal processing and machine learning algorithms as well as various modules for graphics, visualization and user interface development.

**Python, et al.**

CEBL₃ is written almost entirely in Python. This allows for the rapid-development of clearly-written modules and APIs. Python also has many available libraries and packages that allow developers to leverage existing functionality. CEBL₃ relies heavily on the NumPy and SciPy libraries which utilize highly optimized C and Fortran libraries for numerical computations.

Matplotlib is also used to generate both static and real-time visualizations. The wxPython library is used for graphics and user interfaces. Custom modules or routines can also be written in C in order to maintain the levels of performance required for real-time processing.

**Modular Design**

CEBL₃ follows a modular, two-tiered design approach. The first tier contains modules that are generally useful for EEG analysis. This includes:
- Signal processing routines.
- Machine learning algorithms.
- Classes for managing EEG segments and files.
The second tier contains modules that are specifically designed for real-time functionality. This includes:
- Drivers for EEG hardware.
- Filters for processing EEG data streams.
- Widgets and graphics components for developing user interfaces.
- Base-classes for developing experiments, called "pages."

This two-tiered design is intended to follow the development process from beginning to end. A researcher can first test new ideas in an offline fashion in the first tier. This can be done from an interactive environments like IPython. Experiments that prove to be fruitful, can then be extended into real-time "pages" using the functionality of the second tier. This saves time by allowing the reuse of existing code.

**A New Software Platform for BCI**

A New Software Platform for BCI

**Visualization and Graphics**

New ways of visualizing EEG can easily be created in CEBL₃ simply by using the wxPython backend to the Matplotlib plotting library. For fine-grained control or performance-intensive animated graphics, the standard wxPython graphics routines work well. A number of pre-built widgets make it relatively simple for a researcher using CEBL₃ to reuse or extend various existing graphical components.

**Interfaces**

Construction of several prototype user-interface modules for CEBL₃ are currently underway. The first interface implements a P300 based BCI known as the Rapid Serial Visual Presentation (RSVP) speller. In this interface, the user is presented with a series of rapidly flashing characters. The user looks for the character they wish to spell while a machine learning algorithm attempts to identify changes in EEG when the correct character flashes.

Another user-interface utilizes Motor Imagery (MI) to select one of two items from a circular "pie" menu. In this paradigm, the user simply imagines moving their left or right arm while a machine learning algorithm looks for changes in power in the mu band in the contralateral motor region of the brain.

The MI paradigm may also be combined with an Event-Related Potential known as Error-Related Negativity to identify when the BCI has made an incorrect selection.

A third interface is similar to the MI paradigm except that it allows a variety of imagined mental tasks that may elicit changes in activity over various regions of the brain. With practice, this paradigm may yield multiple degrees of freedom and fluid, asynchronous control.

**Controlling Real-World Devices**

The ultimate goal for BCI researchers is to construct systems that are practically useful. To this end, we have implemented a module for driving an ER1 robot over a wireless connection. We have also begun to implement several games that utilize box2d to create realistic simulations of physical environments.