

Product Developments

Brain-Computer Interface technology: a challenging research field

by Maria Schatt

A Brain-Computer Interface is a system for controlling a computer or other device by human volition without using muscle or peripheral nerve activity.

A significant amount of BCI research relates to the task of developing BCI applications based on non-invasive brain signal recordings such as EEG. Unlike invasive or partially-invasive methods which require electrodes to be implanted inside the brain or the cortical surface, non-invasive EEG recordings pose no risk for the person being examined. A lot of effort has therefore been devoted to this field despite the problems resulting from the poor signal-to-noise ratio found in scalp EEG recordings.

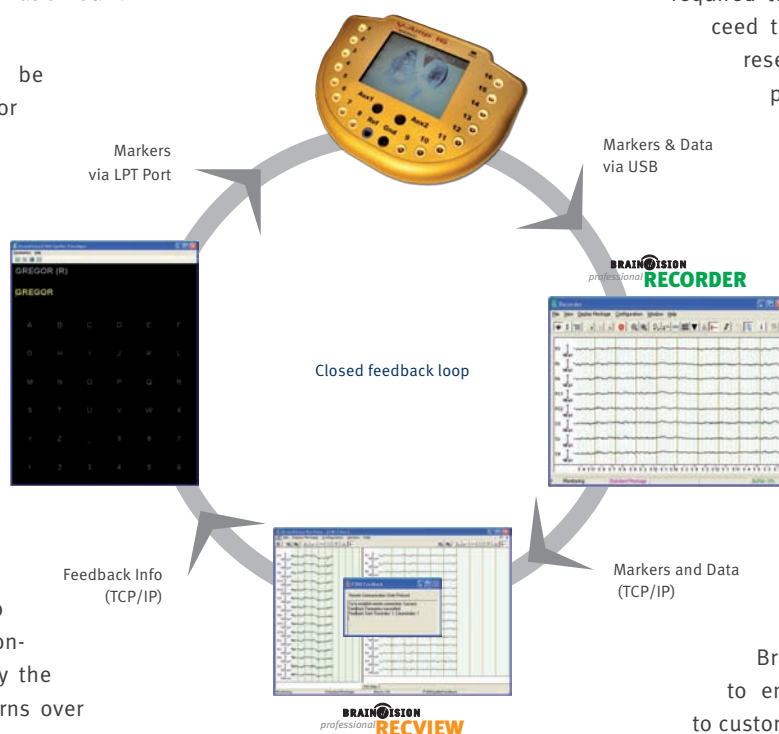
Other challenges have to be faced in addition to the poor signal quality. Even though some characteristic brain activity patterns are well known to be correlated with certain tasks (for example, modulations of the μ -rhythm in response to real or imagined movements), the way these patterns occur in different individuals can differ considerably. In addition to this phenomenon of “inter-subject-variability”, BCI developers and users also have to cope with “inter-session-variability” which is caused by the non-stationarity of brain patterns over time.

One way of dealing with these problems is to provide intensive training to subjects: A subject who is to use a BCI system tries to learn to produce a certain pattern of brain activity reliably and thus to “adapt” to a given BCI system. Another – and more promising – strategy is to automatically adapt the BCI system to the individual who is using it in each new session by applying machine learning techniques.

To make this adaptation possible, a BCI experiment usually consists of three phases: First a supervised training session is conducted and recorded. The recorded data is then analyzed offline. This involves the recognition of the relevant subject- and task-specific features and the training of a classifier on the extracted features. Feature extraction often consists in finding optimal spatial and temporal filters, for example by means of blind source separation techniques such as Independent Component Analysis or Common Spatial Patterns algorithms. Classifiers can be derived using methods such as linear discriminant analysis, Support Vector Machines or

neural networks, to name but a few. During the feedback phase, the relevant features are extracted and the classifier output assigns a control signal to the extracted features.

The above-mentioned problems are only those resulting from neurophysiology and signal processing. However, anyone developing a BCI system “from scratch” faces a large number of technical challenges in terms of informatics and real-time processing. The time and effort required to perform such a task often exceed the resources available in a BCI research project dedicated to a precisely defined research field.



Recent years have seen the emergence of a number of open-source BCI frameworks which can relieve BCI-researchers of the majority of this development effort and allow them to concentrate on finding answers to their individual research questions.

Unfortunately, not all frameworks interface with all types of recording hardware. We are currently trying to improve the integration of Brain Products hardware in order to enrich the possibilities available to customers interested in the BCI field.

One widely used and well-documented BCI framework is BCI2000 (www.bci2000.org), a general-purpose system for brain-computer interface (BCI) research. It can also be used for data acquisition, stimulus presentation, and brain monitoring applications. Its user contribution area features an RDA client which interfaces with the BrainVision Recorder as well as a source module that makes it possible to access the Brain Products V-Amp directly. Apart from the possibility of recording in the Analyzer-compatible formats EDF and GDF, the integrated BCI2000 export tool allows users to export data and “states” (which correspond to BrainVision markers) in BrainVision format. As a result, Analyzer 2 users can apply ICA transformations during offline analyses. To make it possible to benefit from these capabilities in feedback scenarios, we provide a corresponding macro which is soon to be followed by an export component which exports ICA matrices in a BCI2000 and MATLAB compatible format. Exported ICA matrices can thus be used as spatial filters in BCI2000.

Another platform for developing and testing BCI and real-time neuroscience applications is the C++ based OpenViBE (<http://openvibe.inria.fr>). This platform is remarkable for its high level of modularity and meets the needs of programmers and non-programmers alike thanks to the provision of a user-friendly graphical language which allows non-programmers to design a BCI without writing a single line of code. OpenViBE provides partial support for access to the BrainAmp via RDA client using the BrainVision Recorder. This means that the implementation is not robust and there is currently no associated documentation. Fortunately, work is currently underway on the direct integration of the Brain Products V-Amp and this is expected to be complete by the end of 2009.

The relatively new BCI++ (www.sensibilab.campuspoint.polimi.it) is dedicated to the development and fast prototyping of BCI systems, PC-driven protocols for a variety of bio-signal acquisition paradigms and BCI-based applications. Apart from a module for signal acquisition, storage and display, real-time execution and management of custom algorithms, it features a Graphic User Interface module based on a high-level 2D/3D Graphic Engine (Irrlicht). Work on direct V-Amp integration is currently in progress. Although an RDA client exists, we were not able to operate it successfully in our tests.

The MATLAB-based BCI2VR (www.engineering.vcu.edu/eegbci/bci2vr_index.html) is a software system for neural signal processing and classification. Unfortunately, Brain Products hardware is currently not supported. Customers who want to use this framework will need to procure the BrainAmp/V-Amp SDK and interface with the MATLAB Data acquisition toolbox.

Naturally, we have also considered the question of the extent to which we can use the Brain Products software components to build a BCI application. The simplest way to answer this question was to try. We decided to implement a P300 speller application, one of the first BCI spelling devices developed by Farwell and Donchin [1]. The P300 Speller is based on the elicitation and recognition of an event-related potential referred to as P300. This positive deflection in the EEG is characteristic of a subject's reaction to an unusual target stimulus. One paradigm that is widely used to provoke the P300 response in experimental contexts is the so-called oddball paradigm: Within a series of stimuli, an unexpected,

deviant stimulus is presented in a random location and with a small probability of occurrence. The P300 speller paradigm is a variant of the oddball paradigm which is used in a BCI application for the spelling of characters. The paradigm makes use of a selection of characters (letters, numbers or icons) arranged in a matrix. The character matrix usually consists of six rows and six columns. The subject focuses on the desired character while the rows and columns flash in a random sequence. The target stimulus occurs when the character which the subject is focusing on is highlighted. The ratio of the target to the neutral stimuli is 1:5. By detecting the row-and column flashes that, on average, elicit the strongest P300 responses, the system tries to recognize what character the user has been focusing on.

While using BrainVision Recorder as recording software, BrainVision Analyzer for offline analysis and RecView for online processing, it was still necessary to implement the speller paradigm itself. The point of interest here lies in synchronizing the paradigm activity with the recording, i.e. marking the point at which the different stimuli occur in the recorded data. To do this, we used the amplifier trigger port. The speller paradigm sends a TTL signal which encodes the flash event to the parallel port and this then appears as stimulus marker in the recorded data.

We additionally designed an Analyzer export component which implemented a stepwise feature selection algorithm based on linear discriminant analysis [2]. This export component identifies relevant features and classifier weights and exports the information to an xml file.

This xml file is subsequently read into a RecView P300 feedback filter. This filter performs the online evaluation of the recorded data and returns the feedback information to the speller paradigm via TCP/IP. In this way, a full feedback loop is established.

Although we have not yet scientifically studied the information transfer rate in an experimental setting, we have been encouraged by the spelling results achieved by various BP colleagues. This prototype illustrates the possibilities that Brain Vision RecView offers as a feedback tool and provides us with an incentive to continue extending the capabilities of our software beyond the habitual areas of application. ●

References

- [1] Farwell LA, Donchin E. *Talking off the top of your head: toward a mental prosthesis utilizing event-related brain-potentials.* *Electroenceph Clin Neurophysiol* 1988; 70:510–23.
- [2] Geoffrey J. McLachlan: *Discriminant Analysis and Statistical Pattern Recognition*, Wiley-Interscience; 1 Edition (March 27, 1992), Chapter 12