An Overview of Brain Computer Interfaces

Stephen Adams
University of Minnesota Morris
600 E 4th St.
Morris, Minnesota USA
adams601@morris.umn.edu

ABSTRACT

Brain computer interfaces (BCIs) are systems that allow a user to control a computer program using only the power of thought. The study of Brain Computer Interfaces combines several different subjects including: Neurology, machine learning, electrical engineering, and human computer interaction. This paper provides an overview of how neural activity is measured, classified, and used to control systems. It will also discuss how BCI systems are being applied in the world today.

Categories and Subject Descriptors

K.4.2 [**Computers and Society**]: Social Issues—*Assistive technologies for persons with disabilities*; H.5.1 [**Information Interfaces and Presentation (e.g., HCI)**]: Multimedia Information Systems—*Artificial, augmented, and virtual realities*

1. INTRODUCTION

Since the invention of the mouse over fifty years ago [10], and the introduction of graphical user interfaces almost forty years ago [10], the way in which we could interact with computers has not substantially changed. The windows, icons, menus, and pointers (WIMP) design paradigm has been dominant since the 1980s [10]. BCIs offer a wholly new way of interacting with computers. BCI systems have been applied to the WIMP design paradigms, providing a new and novel way to control a pointer. Something not fully explored are the completely new ways of interacting with computers that BCIs could be used for. Though the earliest BCIs adapted existing design paradigms, recently wholly new interactions are being developed for BCI systems. BCIs can react to thoughts that no other device can leverage. These possibilities have not been fully explored. This paper will cover the work that has been done in this developing field.

This paper will provide an overview of many different sciences come together to make brain computer interfaces possible. Section 2 covers the neurological concepts that a BCI system is based on. Section 3 will discuss how a system learns to classify brain activity so a BCI system can accurately respond to a user's thoughts. Finally Section 4 will discuss current uses of BCIs.

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/us/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

2. EVENT-RELATED POTENTIALS, ELECTROENCEPHALOGRAPHY, AND P300

The study of the brain has developed ways to classify and record brain activity. BCIs have adapted these methods so to understand BCIs a little must be known about these concepts. This section will cover how brain activity is recorded with EEG, then it will cover how brain waves are broken into more easily understandable parts.

2.1 Electroencephalography (EEG)

This section will discuss how brain activity is retrieved and recorded as described in [13]. The brain is made of billions of neurons. These neurons are electrically charged from ions being pumped across their membranes by enzymes. Neurons are signaled by their neighbor neurons, releasing ions into the surrounding space. The neurons' response to this signal is to release more ions. Ions of like charge repel each other, this causes ions to be forced away from the neuron that originally fired and signal other neurons. As neurons in an area are all activated ions move out of an area in a wave. This effect is known as *volume conduction*. When this wave reaches the scalp the electron charges are now in an easily accessible location.

Scalp electrodes record the voltage of neural activity. The electrodes are attached to the area of the scalp with a conductive gel or paste. When the volume conduction wave reaches the scalp, ions push on electrons in the metal of the electrodes. This pushing force is the voltage of the signal. The differences in voltage between any two electrodes is measured by a voltmeter. The recording of this electrical activity along the scalp is known as electroencephalography, more commonly referred to as EEG

Electrodes are either attached to individual wires or into nets or caps (see Figure 1). The caps or nets are used for systems that require a lot of electrodes. Studies suggest that only about half of the contribution to each electrode come from a source within 3 centimeters of the electrode [1]. This means that very little extra information can be gained by having a lot of electrodes. As a result BCI systems normally use a low number of electrodes.

EEG is a non-invasive way to record brain activity [12]. Non-invasive procedures are critical for large scale BCI projects

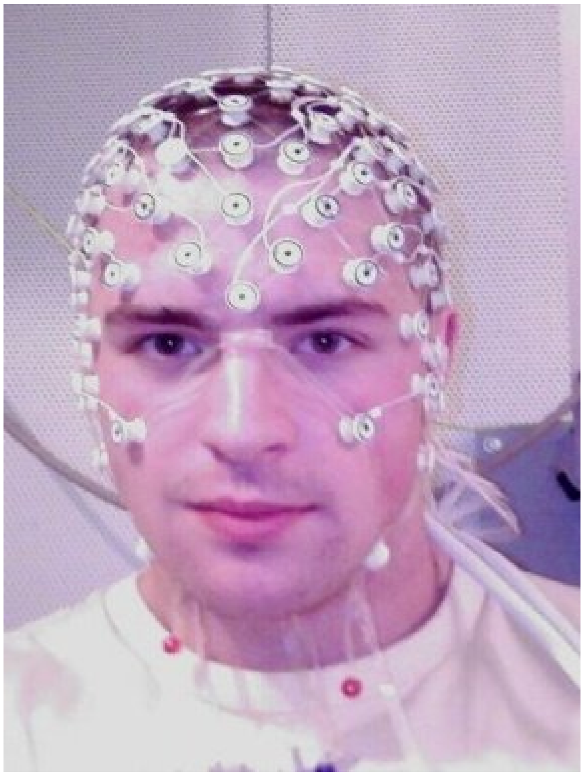

Figure 1: A user with an EEG cap on. Taken from [13]

because of the limited number of people who undergo the brain surgery required for other recording methods [12].

EEG is also very well adapted to investigate brain function as it has excellent temporal resolution [5]. Temporal resolution refers to the amount of data that can be collected per second or in the case of EEG per millisecond. Temporal resolution is important in determining which ERP components have been fired.The spatial resolution of EEG is how easily the source of a signal can be determined. Again the source of brain activity is very important in determining what component has been fired. Spatial resolution is one of EEG's largest limitations. The meninges, cerebrospinal fluid, and skull, which are all not very conductive, "smear" the EEG image so that the source of a signal from the brain is obscured. Some of this spatial resolution can be made up by increasing the number of sensors, which helps pinpoint where a signal originated [5]. Current BCI systems are not able to effectively use more spatial resolution so the number of electrodes that systems use remain small.

2.2 Event-Related Potentials (ERP)

Event-related potentials are any measured response from the brain resulting from a thought or a perception; the brain is constantly firing ERPs [14]. ERPs are integral for a computer to be able to interpret an analog brain wave. The ERPs that have been measured and had their causes discovered are known as *components.*

Components are correlated to various stimuli and their meaning to the user. The majority of components are named using a letter and then a number. The letter indicates whether the voltage deflection is positive or negative. The number indicates the time, in milliseconds, that it takes for the component to occur after stimulus; this is called latency [14]. For example the P300 wave has a positive voltage and occurs about 300 milliseconds after stimuli. One challenge of trying to categorize components is that many of them have a variable latency [14, 5].

2.3 The P300 Component

The P300 component is generated in the parietal lobe when a person is attempting to accomplish some task and sees something related to the task [15, 5]. For example in [4] subjects looked at a screen with 12 "options" on it. Subjects would focus on the option that they were told was relevant, and when that option flashed a P300 event was recorded [15].

P300 events are a strong candidate for BCI design for several reasons. First they are involuntary which means that subjects will need little training on how to activate the P300 to control a BCI, and people with sections of their brain injured will most likely still be able to use the system [15, 12]. The P300 doesn't vary much from users so that means that BCI designs can be simplified and be used by a greater number of people [15]. Up until now subjects using BCIs were required to remain sitting or lying down to avoid muscle movements that pollute the EEG signal [7]. Recent research has suggested, however, that P300 based systems may be usable for users who are standing or even walking around [7]. This means that P300-based BCI systems could be used in settings where other systems would not be reliable.

3. ANALYSIS OF BRAIN ACTIVITY

Due to the complexity of ERP components, BCI systems up to this point have focused on recognizing at most only a few components [12]. Machine learning techniques are used to recognize the components of the brain wave activity. These classification algorithms take in various pieces of information, or features, from the EEG signal as a list; this list is called the feature vector [6]. Possible features include the amplitude of the EEG signals and time-frequency information [6]. From this information the classifier will output if the given feature vector represents the component the BCI system is based on or not. Before the machine learning techniques can effectively classify feature vectors it needs to be "trained". For BCI systems, training will involve stimulating the user's brain so that the component that the system is looking for is fired.

A major design consideration for a BCI system is what classification algorithm to use. Features that a successful algorithm must have are [6]:

- Ability to handle noise in the input data
- Ability to handle a large feature vector
- Ability to deal with a wide range of values
- Ability to train quickly on a small training set

Many different techniques have been employed to recognize components. The next few sections will outline a few of these techniques.

3.1 Linear Classifiers

Linear classifiers are one of the most popular algorithms used for BCI systems [6]. In general linear classifiers use linear functions to draw lines between classes of brain waves. Figure 2 shows an example of this happening. The circles are one class and the squares are another, the line separates these two classes which allows the system to classify new data based on which side of the line it is on. The two primary types of linear classifiers used by BCIs are linear discriminant analysis and support vector machines [6].

3.1.1 Linear Discriminant Analysis

Linear discriminant analysis identifies different components by using hyperplanes to separate feature vectors [6]. Essentially linear discriminant analysis draws a line between the points that represent each class. This line is established by giving the system example brain activity where the ERP component it represents is known. During training linear discriminant analysis moves the line so that the line is always between feature vectors of different classes. Once training is complete, the hope is that this line will be able to correctly separate new data. If more than two classes exist linear discriminant analysis can be used to create multiple hyperplanes to separate the different classes [6].

3.1.2 Support Vector Machines

Support vector machines (SVM) are also commonly used as linear classifiers by BCI systems. The goal of a support vector machine is similar to that of linear discriminant analysis, that of trying to place a hyperplane which separates the classes [6]. For any given separation problem there could be multiple hyperplanes that separate the classes. Support vector machines try to find the optimal hyperplane [8]. The optimal hyperplane is the one that maximizes the margins around it [6, 8]. The margins are the distance between the hyperplane and the nearest training points [6]. This margin gives the system some protection from data with a lot of noise or variation when the system is being used. A support vector machine can also be adapted to separate classes in non-linear ways [6, 8].

An SVM has a kernel which defines what class of function it creates the optimal hyperplane from. This is known as the "Kernel trick" [6]. Considering that MLPs are much easier to implement and are very well suited to a BCI classification problem SVMs are less popular [6].

3.2 Neural Networks

Neural networks, along with linear classifiers make up the majority of classifiers used in BCI systems [6]. Two neural network architectures are primarily used in BCI systems: the perceptron and the multilayer perceptron (MLP) [6]. Other architectures have been used for BCIs but not very often so they are not discussed here in any detail. For example the gaussian classifier, which was designed specifically with BCIs in mind, outperforms MLPs but is more difficult to implement and because of this gaussian classifiers are rarely used [6].

Neural networks are made up of a set of artificial neurons known as McCulloch and Pitts neurons [8]. McCulloch and Pitts neurons are composed of three parts, a set of weighted inputs, an adder, and an activation function (see Figure 3). The weighted inputs are just the information the neuron is receiving (electrode voltage data, or values from other

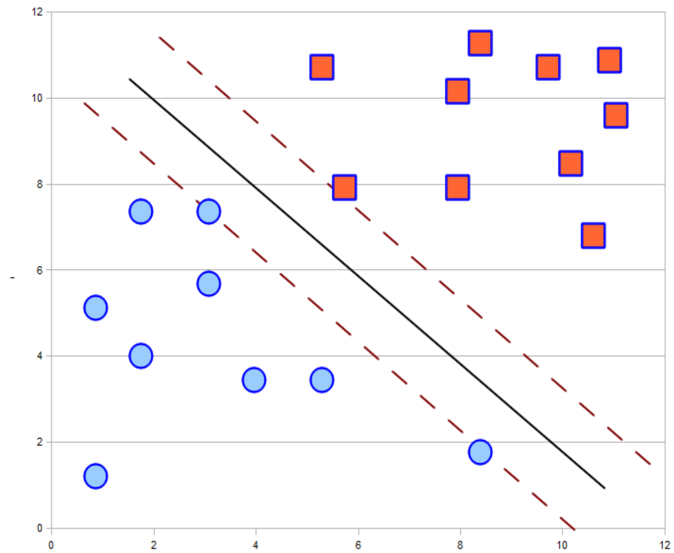

Figure 2: An SVM places the optimal hyperplane (the solid line) between the margins to separate the two classes.

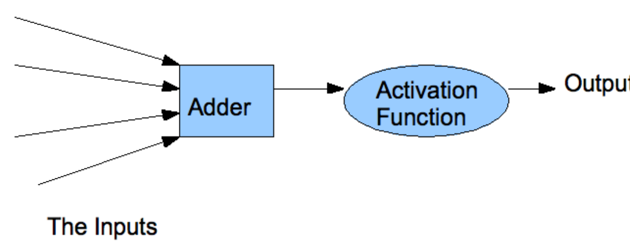

Figure 3: A graphical representation of a McCulloch and Pitts neuron.

neurons) multiplied by a weight. These weights are what are going to be adjusted in the training phase. The adder simply sums up the weighted inputs and sends the sum along to the activation function. The activation function decides whether the neuron fires or not. The activation function is often just a threshold function which means that if the sum from the adder is above some value the neuron will fire. When a neuron fires it sends a value to either other neurons or as output. The most basic threshold function sends out a 1 if the neuron fires and a 0 if it does not. Depending on the application, the activation function can be adapted to send out different types of outputs [8].

3.2.1 Perceptron

The simplest neural network architecture is called a perceptron. A perceptron consists of a single McCulloch and Pitts neuron. Perceptrons are mathematically equivalent to linear discriminant analysis, so it has sometimes been used instead [6].

3.2.2 Multilayer Perceptron

The most popular neural network architecture is the multilayer perceptron [6]. MLPs are composed of several layers: an input layer, one or two "hidden" layers of neurons, and an output layer [6, 8]. The weighted inputs in an MLP lead to the first hidden layer. The first hidden layer's outputs are multiplied by a weight and become the input to the next hidden layer. Once the data has moved through every hidden layer it moves to the output layer. By modifying the number of hidden layers, the number of neurons in each layer, and the activation function, an MLP can classify a feature vector into any number of classes [6]. MLPs can classify problems too complex for linear classification. Such a problem is illustrated in Figure 4. As you can see there is no straight line that can completely separate the two classes. An MLP can produce complex, non-linear functions which can be used to separate the classes.

3.2.3 Training

During the calibration of a BCI using a neural network as its classifier, the neural network's weights are being trained. Every feature vector that is sent to the network is already known to be either the component the system is based on or some other ERP. After the network outputs which class it thinks the feature vector belongs to, the network begins training.

δ_{ok}	Error for an output layer neuron
δ_{hk}	Error for a hidden layer neuron
η	The training rate
w_{jk}	The input weights for output layer neurons
v_{ij}	The input weights for hidden layer neurons
a_j	The output from an output layer neuron
a_j^{hidden}	The output from a hidden layer neuron
x_i	The inputs

The table above shows the variables that are used in the training algorithm.

Training is based on the error of each node. These errors are computed differently depending on whether the node is in the output layer or in a hidden layer. The error for an output layer weight k δ_{ok} is based on the actual output of the node y_k and the expected or target output t_k. The error

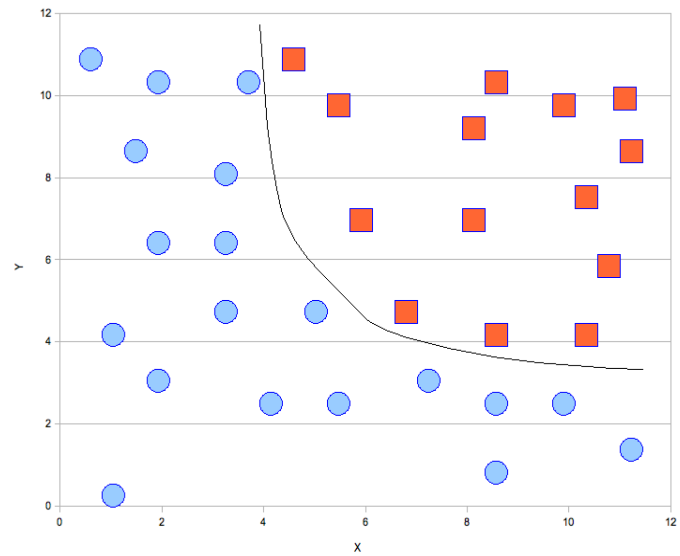

Figure 4: A classification problem that needs a nonlinear classifier. This could have been solved by an MLP or an SVM with a nonlinear kernel.

function is shown below.

$$\delta_{ok} = (t_k - y_k)y_k(1 - y_k)$$

The hidden layer errors δ_{hk} are based on the output of the current hidden layer node a_j and then the sum of all the output layer weights w_{jk} and the output layer error:

$$\delta_{hk} = a_j(1 - a_j)\sum_k w_{jk}\delta_{ok}$$

The weights can now be updated. If you trained a neural network based just on the error alone it would be very unstable, overcompensating each time it misclassified a feature vector. To reduce this effect a training rate η normally between 0.1 and 0.4 is used. The training rate is used when updating both output layer weights w_jk and hidden layer weights v_ij. Output layer weights are updated using the following function:

$$w_{jk} \leftarrow w_{jk} + \eta\delta_{ok}a_j^{hidden}$$

where a_j^{hidden} is the output from the hidden layer. To update a hidden layer weight the following equation is used:

$$v_{ij} \leftarrow v_{ij} + \eta\delta_{hj}x_i$$

where x_i is the input to that particular node.

3.3 Other classifiers

Other classifiers have been used in BCI systems but it is out of the scope of this paper to discuss them in detail.

Brain computer interfaces have used two Bayesian classifiers: Bayes quadratics, and Hidden Markov Models [6]. Bayesian classifiers produce nonlinear decision boundaries and are better suited to classifying vectors which are similar to members of different classes.

Another type of classifier is known as a nearest neighbor classifier. Nearest neighbor classifiers attempt to measure the distance from a feature vector to representative members of different classes; the class that a feature vector is nearest to is the class it is placed in. The two nearest neighbor classifiers most commonly used in BCI systems are the k-nearest neighbor technique, and the Mahalanobis distance [6].

3.4 Applying Combinations of several classifiers

One final idea for classification that has been used recently with some success is applying several classifiers [6]. The three combinations that have been tried so far are called boosting, voting, and stacking.

Boosting uses several classifiers in cascade [6]. Each of the classifiers focuses on the errors committed by the previous classifier [6]. After training this can become a very powerful classifier. It is, however, sensitive to mislabeled training data, which could explain why it was unsuccessful in some experiments [6].

Voting is a very simple technique that is easy to implement. In voting, several classifiers each assign a feature vector to a class. Whichever class was the most popular is the one that the vector will be assigned to [6]. This is the most popular method of combining classifiers in BCI research because of its simplicity and effectiveness [6].

Stacking involves several levels of classifiers. The first level of classifiers are called level-0 classifiers. Level-0 classifiers are given the feature vector to classify into a class [6]. The output from the level-0 classifiers goes to the level-1 classifier which is also called a meta-classifier [6]. The meta-classifier combines the results from the level-0 classifiers and makes the final decision as to what class the feature vector is assigned to. BCI research into stacking, so far, has used hidden Markov models as level-0 classifiers and a support vector machine for the meta-classifier [6].

4. APPLICATIONS OF BCI SYSTEMS

Brain computer interfaces have been used for many different purposes. For example [7, 16, 9] all deal with BCIs used as an interface for video games, while [11, 2, 3] use BCI systems to help people with disabilities use computers. This section will discuss these and other uses for brain computer interfaces.

4.1 Assistive and Therapeutic Applications

BCIs were originally used in assistive technology applications [12] so the body of research in this area is much larger than that of entertainment. Early BCI systems were focused on supporting people who suffer from locked-in syndrome. Before BCIs these people had few chances at communication, normally people would read off letters to the patient who would signal when they wanted to select a letter. Character by character people would spell out words. This is obviously quite difficult, and it assumes that the patient has enough motor control to signal the selection of a letter. The first BCIs computerized this process by flashing letters up on a screen and when the patient's P300 component fires the letter on the screen is chosen [12]. This simple method of communication has since been extended by many researchers (e.g. [2, 11]). The method described in [2] attempts to speed up locked-in patient's communication. As opposed to the early BCI designs, where individual letters had to be painstakingly selected. The system in [2] collects commonly used words and phrases and displays them along with the alphabet for the user to select from.

Another way the a BCI has been used to support people with severe physical disabilities is a system that allows people to browse the web using their mind [11]. People with these types of physical disabilities have trouble successfully manipulating a mouse to select something as small as a hyperlink. The level of control that BCIs can currently offer their users is also unsuited to allow successful navigation of traditional browser interfaces. For these reasons [11] developed new interface design paradigms to allow for a BCI system to surf the web. Previous work done by the authors of [11] showed that BCI systems could fairly easily control linear interfaces. Conventional web browsers, and the pages they display, involve navigating two spatial dimensions. The challenge in [11] is to adapt this structure to a single dimension. The design paradigm that was developed includes a toolbar with common internet navigation operations (back, forward, home, etc.). Two unique buttons that are very important to the usability of this system allow a user to move back and forth over the links in a page. These buttons effectively serialize a two dimensional web page.

Physical therapy and rehabilitation are also being assisted by BCI systems. There is an ERP component known as a mu rhythm that has been shown to weaken with the decline of physical ability [3]. Physical ability can increase by using a mu based BCI system [3]. This opens the possibility that training on a mu-based BCI system could help people regain lost physical abilities [3]. For example

4.2 Entertainment

The use of BCIs for gaming and other types of entertainment is a relatively new idea. Most of the work done in this area tends to be about theoretical designs or testing new BCI features [7, 9]. In [7] the authors developed a BCI system that could be used while standing up and walking around. This research was done explicitly so that non-disabled persons could use BCIs for entertainment and virtual reality purposes. [9] is a survey of the accomplishments of BCIs in the entertainment field. Two main design ideas were put forth for BCI researchers to look into: replacing more traditional input methods such as mice and keyboards or a controller with a BCI, and using a BCI to make a traditional game react to a person's state of mind. For example if the BCI could sense a user getting frustrated then the system could automatically make the game less difficult. This type of BCI system could be used to help an activity adapt to better meet the users needs at any given moment [9].

[16] describes a game called NeuroWander that has an integrated BCI. NeuroWander is based around the fairytale of Hansel and Gretel. Players use the keyboard to move their character around the game world. The BCI system evaluates two values, "attention" and "meditation" [16]. As the user meditates (relaxes their mind) bread crumbs appear, and if the user concentrates pebbles appear. The bread crumbs and pebbles lead the character through the woods. When enough of these markers appear, the player is led to the witches cottage where they can push the witch into the oven and thereby win the game. The BCI system here was used to enhance the traditional gameplay mechanics. As opposed to just relying on the users motor movements pushing buttons on the keyboard, the game can also respond to the players state of mind.

5. CONCLUSION AND FUTURE WORK

In conclusion a BCI system takes a user's brain activity classifies that activity into a category which a computer can then respond to. The classification has been done by multiple methods. This classification is done by machine learning algorithms. The most common being multilayer perceptrons. Initially BCIs were used to give people with severe physical disabilities ways to communicate and interact with computers. Accessibility research continues to progress today, getting more and more complex. Recently BCI research has become more focused on able-bodied users. Specifically the possibilities for BCIs to enhance entertainment are being investigated.

Brain computer interfaces have many possible questions that haven't been answered yet. As was mentioned previously, many of the papers that have been published about entertainment uses of BCIs were more theoretical rather than implementing an actual system, so much more work can be done in that area to test the ideas that have already been suggested. Many of the classification techniques have not been fully explored either. There are also, of course, many environments where BCIs could be incorporated but have not been. Overall, there are still many things that have not been explored about BCIs.

6. REFERENCES

[1] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Muller. Optimizing spatial filters for robust eeg single-trial analysis. *Signal Processing Magazine, IEEE*, 25(1):41 –56, 2008.

[2] K. Carroll, C. Schlag, O. Kirikci, and M. Moore. Communication by neural control. In *CHI '02 extended abstracts on Human factors in computing systems*, CHI '02, pages 590–591, New York, NY, USA, 2002. ACM.

[3] A. B. Davis. Investigating the mutual effects of physical training and mu-based brain-computer interface systems. *SIGACCESS Access. Comput.*, pages 30–32, June 2005.

[4] L. Farwell and E. Donchin. Talking off the top of your head: Towards a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology*, 70:510–523.

[5] T. C. Handy, editor. *Event-Related Potentials A Methods Handbook*. The MIT Press, Massachusetts Institute of Technology Cambridge, Massachusetts 02142, 2005.

[6] F. Lotte, M. Cengedo, A. Lecuyer, F. Lamarche, and B. Arnaldi. A review of classification algorithms for eeg-based brain computer interfaces. *Journal of Neural Engineering*, 4, 2007.

[7] F. Lotte, J. Fujisawa, H. Touyama, R. Ito, M. Hirose, and A. Lécuyer. Towards ambulatory brain-computer interfaces: a pilot study with p300 signals. In *ACE '09: Proceedings of the International Conference on Advances in Computer Enterntainment Technology*, pages 336–339, New York, NY, USA, 2009. ACM.

[8] S. Marsland. *Machine Learning An Algorithmic Perspective*. CRC Press, 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487-2742, 2009.

[9] A. Nijholt, D. Tan, B. Allison, J. del R. Milan, and B. Graimann. Brain-computer interfaces for hci and games. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI '08, pages 3925–3928, New York, NY, USA, 2008. ACM.

[10] H. Sharp, Y. Rogers, and J. Preece. *Interaction Design: Beyond human-computer interaction*. John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA, 2007.

[11] O. Tomori and M. Moore. The neurally controllable internet browser (brainbrowser). In *CHI '03 extended abstracts on Human factors in computing systems*, CHI '03, pages 796–797, New York, NY, USA, 2003. ACM.

[12] Wikipedia. Brain-computer interface, Oct. 2010.

[13] Wikipedia. Electroencephalography, Oct. 2010.

[14] Wikipedia. Event-related potential, Oct. 2010.

[15] Wikipedia. P300 (neuroscience), Oct. 2010.

[16] M.-S. Yoh, J. Kwon, and S. Kim. Neurowander: a bci game in the form of interactive fairy tale. In *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing*, Ubicomp '10, pages 389–390, New York, NY, USA, 2010. ACM.