BCI controlled robotic arm as assistance to the rehabilitation of neurologically disabled patients

Anthony Casey[a] **Hannan Azhar**[b] **Marek Grzes**[a] **Mohamed Sakel**[c]

[a] *Department of Computing, University of Kent, Canterbury, UK;*
[b] *School of Engineering, Technology and Design, Canterbury Christ Church University, Kent, UK;*
[c] *East Kent Hospitals University, NHS Foundation Trust, Canterbury, UK*

CONTACT Hannan Azhar hannan.azhar@canterbury.ac.uk School of Engineering, Technology and Design, Canterbury Christ Church University, Kent, UK

ABSTRACT

Purpose: Brain computer interface (BCI) controlled assistive robotic systems have been developed with increasing success with the aim to rehabilitation of patients after brain injury to increase independence and quality of life. While such systems may use surgically implanted invasive sensors, non-invasive alternatives can be better suited due to ease of use, reduced cost, improvements in accuracy and reliability with the advancement of the technology and practicality of use. The consumer-grade BCI devices often capable of integrating multiple types of signals, including Electroencephalogram (EEG) and Electromyogram (EMG) signals.

Materials and Methods: This paper summarizes the development of a portable and cost-efficient BCI controlled assistive technology using a non-invasive BCI headset "OpenBCI" and an open source robotic arm, U-Arm, to accomplish tasks related to rehabilitation, such as access to resources, adaptability or home use. The resulting system used a combination of EEG and EMG sensor readings to control the arm. To avoid risks of injury while the device is being used in clinical settings, appropriate measures were incorporated into the software control of the arm. A short survey was used following the system usability scale (SUS), to measure the usability of the technology to be trialed in clinical settings.

Results: From the experimental results, it was found that EMG is a very reliable method for assistive technology control, provided that the user specific EMG calibration is done. With the EEG, even though the results were promising, due to insufficient detection of the signal, the controller was not adequate to be used within a neurorehabilitation environment. The survey indicated that the usability of the system is not a barrier for moving the system into clinical trials.

Keywords: BCI ; assistive technology ; EEG ; EMG ; disability

Introduction

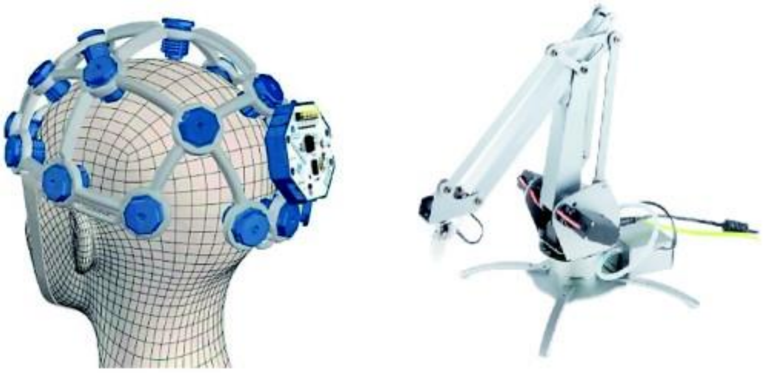
For the rehabilitation of patients suffering from neurological disabilities (particularly those suffering from varying degrees of paralysis), it is necessary to develop technology that bypasses the limitations of their condition. For example, if a patient is unable to walk due to the unresponsiveness in their motor neurons, technology can be developed that used an alternate input to move an exoskeleton, which enables the patient to walk again with the assistance of the exoskeleton.

This research focuses on neuro-rehabilitation within the framework of the NHS at the Kent and Canterbury Hospital in the UK. The hospital currently does not have any system in place for self-driven rehabilitation and instead relies on traditional rehabilitation methods through assistance from physicians and exercise regimens to maintain muscle movement. While this may be somewhat effective for patients suffering from mild forms of paralysis, it is of little-tono use for patients suffering from more severe motor impairments, such as CLIS (complete locked-in syndrome). The use of BCI devices is invaluable in this regard, since it allows for the bypassing of motor commands and instead gets inputs directly from the brain via EEG data. These EEG signals can then be processed and used to control assistive technology such as a robotic arm, or an entire exoskeleton.

The use of BCI devices has been proven to aid in rehabilitation more than traditional methods, such as in cases of paraplegia [1], post-stroke stroke disorders [2] and spinal cord injury [3]. There are several approaches to data processing in the field of neuro-rehabilitation using BCI devices. As an example of this, King et al. [1] used a combination of a BCI, a virtual reality environment (VRE), an EEG-decoding model and a Bayesian classifier to stimulate muscles into walking via electrical pulses. Irimia et al. [4] used a BCI-driven exoskeleton rehabilitation system to assist in moving with the use of electrical pulses for muscle stimulation.

This paper investigates the use of EEG/EMG data gathered from an OpenBCI headset (Figure 1, left) to control an UFactory UArm Metal (Figure 1, right) to develop a portable and cost-effective neurorehabilitation system. This process requires the acquisition of relevant EEG data from the BCI device, the classification of this data and the control of the robotic arm based on the result of the classification. While there are many other papers that focus on BCIdriven neurorehabilitation systems, the one detailed in this paper will be designed with portability and low-cost in mind. The remainder of this paper is organized as follows: Section “Literature review” details a brief literature review into rehabilitation methodologies and EEG classification algorithms. Section “Proposed robotic system” reports the data acquisition methods, further data processing and classification algorithms used in the project. Section “Robotic control protocols and experimental scenarios” gives details of the system components and experimental scenarios. Section “Experimental results” reports the results of the experiments and discuss the performances of the system. Finally, Section “Conclusion and future work” concludes the paper and gives directions to future work.

Figure 1. OpenBCI and UFactory UArm Metal.



Literature review

Given the wide range of issues that neuro-rehabilitation cover, as well as the success in the implementation of these BCI-driven systems, it is apparent that BCI is a viable alternative to traditional neuro-rehabilitation methods. There are several benefits to using a BCI neuro-rehabilitation system, such as; less need for staff to monitor patients during the rehabilitation process (reduces the money spent on staff), re-usable equipment (which again, lowers the cost) and giving confidence and independence to patients during the rehabilitation process. This literature review will focus on two main parts: existing BCI rehabilitation systems and algorithms used for EEG data analysis.

BCI based rehabilitation

Functional electrical stimulation FES is relatively old rehabilitation technique, with the first use of the technique being mentioned in Liberson’s 1961 paper [5] under the name of functional electrotherapy. The name FES was later coined by Moe and Post in their 1962 paper [6]. Pfurtscheller et al. [7] used a BCI-FES to restore the grasping function of a hand in a patient with tetraplegia. This allowed users to perform a “switch-triggered lateral hand grasp”. Do et al. [8] reported on BCI-FES system that was used to control ankle movement. The paper concluded that the integration of a BCI device within a FES system for the lower extremities is “feasible”, based on the results showing that “idling and dorsiflexion epochs could be predicted from the underlying EEG data with an accuracy as high as 97.6%”.

Use of virtual-reality based 3 D environments (VRE) to train users for rehabilitation tasks has been a popular choice amongst researchers [1,9,10]. King et al. [1] used FES to move a user’s leg muscles in accordance with signals obtained from the brain. The user was then required to walk around in a VRE for the purposes of data collection and as an objective way of measuring system success. The results of this paper show that the user was able to obtain “perfect BCI-VRE control after only 11 h of BCI training”. Badia et al. [9] used a VRE as a method of dynamically adjusting the difficulty level of rehabilitation tasks depending on the capabilities of the user through a series of different virtual environments. The results indicate that users responded positively to this system.

Another approach to rehabilitate patients is the use of exoskeletons [4,11] and other assistive technologies during rehabilitation. Irimia et al. [4] used a robotic FES-exoskeleton for the purposes of upper-limb rehabilitation of people with neuromotor disabilities. By using motor imagery (imagining the left or right arm moving), the exoskeleton then proceeds to move in accordance to that signal, while also using FES to stimulate the actual arm muscles. The paper reports that the user was able to control their arm “very well” and there was no discomfort.

EEG data analysis

This section serves as a useful discussion of the different approaches which are the most common feature extraction techniques applied to EEG analysis. Feature extraction is the process of obtaining important data attributes from a piece of data. The main benefit of feature extraction is reducing the amount of data (also known as dimensionality reduction), thereby making classification training easier and faster. Data manipulation (or pre-processing) is often a part of feature extraction, which may not reduce the number of dimensions in the data but may make the data more relevant.

Fast Fourier Transform (FFT) [12–14] is one of the most common techniques for EEG data manipulation. FFT works by computing either the Discrete Fourier Transform (DFT) or Continuous Fourier Transform (CFT) and then Fourier analysis converts the signal from the time domain to the frequency domain. DFT uses the assumption that a signal repeats over a certain period of time, whereas CFT assumes that the signal extends to infinity (i.e., does not have a period of repetition). A Fourier Transformation can also be performed on frequency domain data to get data in the time domain, which is known as an Inverse Fourier Transform (IFT). To obtain useful features from the EEG signal, characteristics of the signal are computed by power spectral density (PSD) [14]. PSD is calculated by applying an FFT operation on the estimated autocorrelation sequence which is found through nonparametric methods, such as Welch’s method [15]. The use of FFT is well-suited to narrowband and stationary signals.

Wavelet transformation (WT) is another feature extraction method [14,16–18] for EEG analysis. WT is a spectral estimation technique which can express any general function as an infinite series of wavelets [14,16–18]. By using variable-sized windows, the time-frequency signal representation is significantly more flexible, being able to show more data over a longer period of time by increasing the window size and likewise reducing the window size decreases the amount of data. There are two different types of wavelet transformation, one is continuous, and the other is discrete. Continuous wavelet transformation (CWT) is the simplest method of wavelet transformation, however since the scaling and translation parameters of the CWT change continuously, meaning that this method uses a lot of computational power and yields a lot of unused information [14]. This approximates to the continuous Fourier transform. Discrete wavelet transformation (DWT) is designed to overcome the shortcomings of CWT and roughly corresponds to the discrete Fourier transform. DWT greatly improves on the amount of required computational power and the yield of useful data, by explicitly describing the features of the EEG signal segment [14]. Wavelet transformations are suitable for both transient and stationary signals. Transient signals change over time while stationary signals repeat over a given period of time or stay the same.

Eigenvectors consist of a variety of functions, designed to handle different issues with the signal. Pisarenko harmonic decomposition [14,19] is one type of eigenvector that is used to estimate the power spectral density (PSD) of a signal. Using the Pisarenko method may result in false zeros, so the MUSIC method [14] was developed to eradicate this issue. The MUSIC method [20] works by using the spectra’s average equivalent to artefact subspace of whole vectors. The minimum norm method [21,22] is another way of addressing the issue of false zeros in Pisarenko’s method, by separating them from real zeros. Eigenvectors are used for signals that are buried with noise [14], however, they have poor statistical accuracy, due to often yielding false zeros [22].

Classification investigation

Support Vector Machines (SVMs) are a machine learning classification algorithm. SVMs have been used for EEG classification in several studies [22,23], with a high level of predictive accuracy, particularly when used in conjunction with eigenvectors. Support vector machines are typically two-class classifiers; however, a variety of methods exist for multiclass classification using SVM. Alternatively, a set of two-class classifier SVMs can be built and the class that is chosen by the most SVMs is the predicted class.

Long Short-Term Memory networks (LSTMs) are a type of recurrent neural network, designed to handle sequential data. LSTMs have been used for EEG classification [24,25] and has yielded very high predictive accuracy, outperforming many of the top classifiers for certain EEG datasets. For example, a study [24] used LSTM to address the issue of other classifiers (SVMs, LDAs etc), by taking advantage of the fact that LSTM makes full use of time sequence information in time-frequency features. The study also used a discrete wavelet transform (DWT) to extract the time-frequency features for the classifier.

Visual evoking paradigms

P300 [26] is a fairly simple visually evoked potential (VEP) which is observable in EEG signals and works on the basis that a subject’s brain will respond in a different way when an infrequent target visual event occurs as opposed to a more frequent non-target visual event. There are a number of different popular paradigms [27–29]; the single character paradigm (SCP), the row-column paradigm (RCP), the chequerboard paradigm (CBP) and the region-based paradigm

(RBP). In simple terms, when a brain observes an infrequent event, it responds with a more noticeable voltage spike than for frequent events.

RCP is the most common of the paradigms and consists of a 2-dimensional grid of images (usually alphanumeric values). The user focuses on one of these images, then a random row or column will flash or highlight quickly and after a brief pause, another random row or column will do the same. This process repeats a number of times until there is a sufficient amount of data for the classifier to determine which character the user was focussing on. It is expected that there will be an observable increase in brain activity when the character the user is focussing on flashes; due to it being an infrequent event. Other paradigms address issues within this paradigm; for example, CBP eliminates the double target flash issue and also reduces the interference by not having adjacent cells highlight sequentially. SCP also solves these issues with RCP, however, it also has a significantly longer trial time, due to needing more flashes to hit the desired cell.

Steady state visually evoked potential (SSVEP) [30] is another popular visually evoked potential. It works by having several objects on the screen flashing at different frequencies and when the subject focusses on one of these objects, there should be an oscillation in their brain waves that matches the frequency (or a multiple of that frequency) of the object flashing. Since the retina has a limited range of frequencies that it can be excited by, the frequencies of the flashes should be within 3.5–75 Hz [30], most commonly using the golden ratio to increment the frequency values for each of the objects.

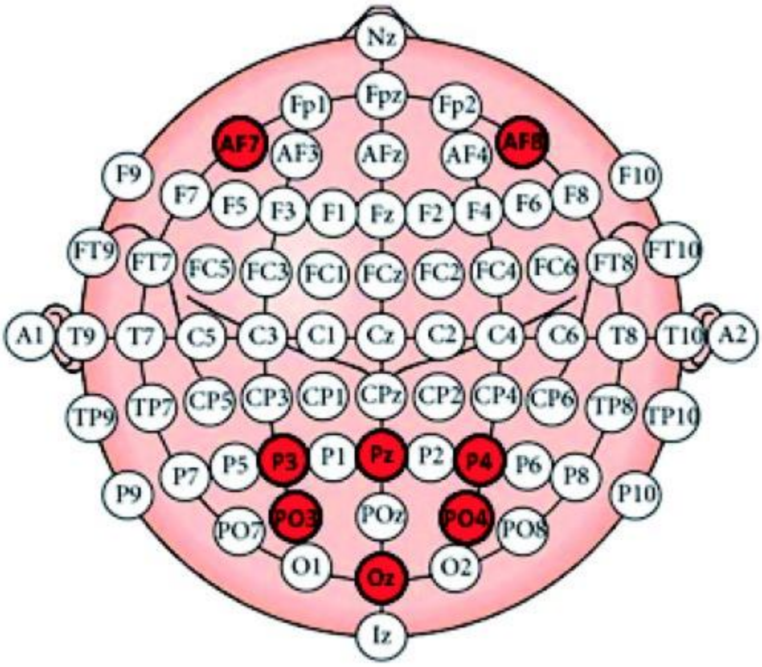
Proposed robotic system

This section details the steps taken in the proposed system to acquire brain signals, extract important “features” from the data and how semantics are applied to those features.

Data acquisition

The first step is to obtain data from the OpenBCI device, which is achieved via the hosting of a Python server leveraging the benefits of a Bluetooth connection direct to the OpenBCI device. Several sets of data are supported by this server, but the most notable are the 8 EEG electrodes; AF7, AF8, Pz, P3, P4, PO3, PO4 and Oz (using the 10–20 international system seen in Figure 2) at a sample rate of 250 Hz.

Figure 2. OpenBCI electrode placement (using the 10–20 International System). Image sourced and modified from [31].

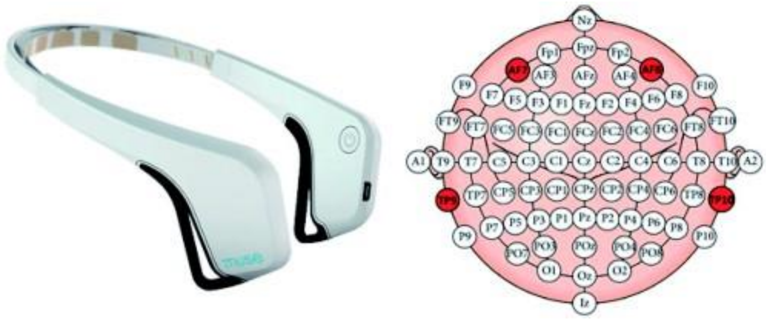


Front electrodes (AF7 and AF8) are used for the detection of three EMG signals; blink, left wink and right wink. After extensive testing, it was found that these two electrodes gave the most noticeable and distinctly individual signals, with minimal noise interference. The remaining six electrodes (Pz, P3, P4, PO3, PO4 and Oz) are located equidistantly around the visual cortex of the brain, so as to be able to pick up visually evoked potentials triggered by the P300 GUI.

In addition to the OpenBCI device, to further demonstrate the adaptability of this system, an Interaxon Muse BCI device (Figure 3) also was tested with the EMG component of these experiments, due to the Muse also having AF7 and

AF8 front electrodes present (TP9 and TP10 are not used). This allows for a comparison of how BCI device can influence the results of experiments.

Figure 3. Interaxon Muse and its electrode placement based on the 10–20 International System [31].



Pre-processing

There is no pre-processing for the filtration of EMG signals active in the AF7 and AF8 electrodes. The remaining 6 electrodes (Pz, P3, P4, PO3, PO4 and Oz) are filtered to remove signal noise (e.g., electromagnetic interference). First, a Kaiser windowing function [32] is applied to the time-series data to ensure the start and end voltages are the same (which mimics a continuous signal). Next, the data is run through a 50 Hz notch filter (a type of narrow bandstop filter) to mitigate the 50 Hz frequency spike due to power line noise. After this, the data is manipulated by a 12 Hz low-pass filter (a Butterworth filter), which narrows the region of interest to frequencies where P300 features are found. Thus, only data between 0.5 Hz and 12 Hz is used to extract features within the frequency domain for P300 events. A Fast Fourier Transform (FFT) function is used to get the data in the frequency domain. The unfiltered time-series data are also stored in memory, which is used to extract an additional set of features.

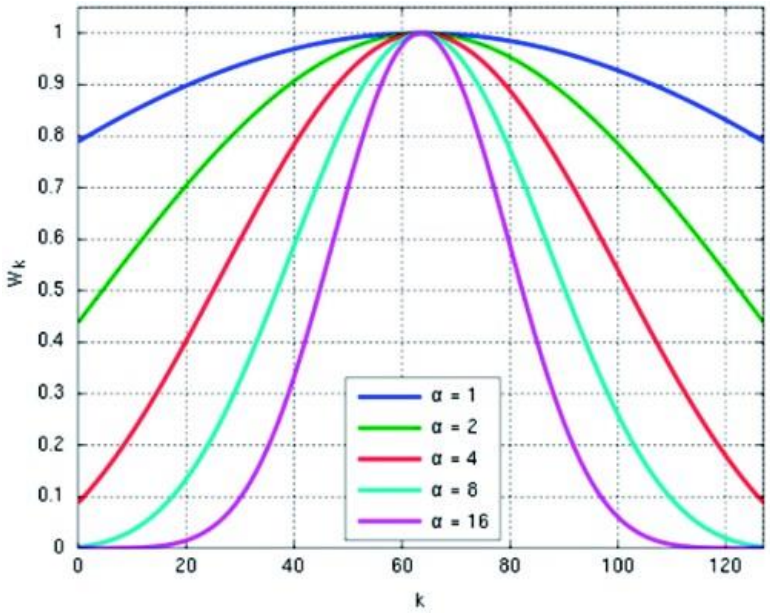
Kaiser window

The Kaiser window, (also known as a Kaiser-Bessel window) is a type of windowing function used in finite impulse response (FIR) filter design and spectral analysis [32]. Windowing is important when looking at discrete snapshots of EEG data, in order to mimic a continuous signal (this has the effect of flattening or tapering the signal at either end). To apply this filter in the time domain, simply sample the Kaiser window (Equation (1)) to the same length as the data and multiply each point in the data, with the corresponding point in the window. Alternatively, a Fourier transform can be applied to the window, allowing for manipulation of data in the frequency domain.

$$W[n] = \begin{cases} \dfrac{I_0\left[\alpha\sqrt{1-\left(\dfrac{4n^2}{(N-1)^2}\right)}\right]}{I_0[\alpha]}, & 0 \le n \le N-1, \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Only the α parameter in a Kaiser window is mutable. By changing the α value of the Kaiser window, the shape of the window changes (Figure 4). For example, an α value of 0 would give a rectangular filter (also known as a Boxcar filter), a value of 5 would be similar to a Hamming window and a value of 6 would be closer to a Hann window. I_0 represents the zeroth-order Bessel function used in the Kaiser function and N is simply the length of the window. This experiment used a Kaiser filter with an α value of 4.

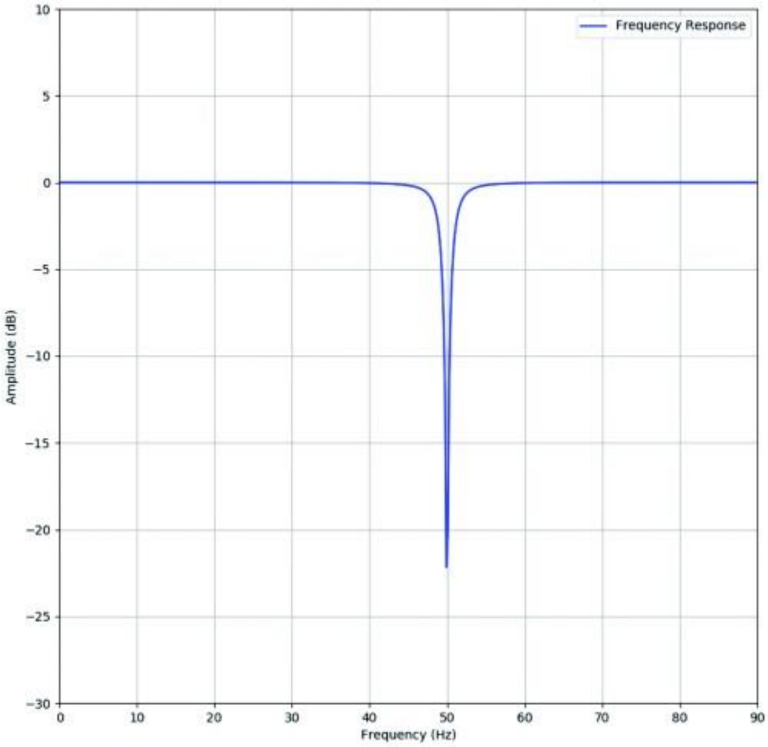
Figure 4. Changing α for a Kaiser window.



Notch filter

Band-stop filters are a type of filter that attenuates (changes the value of) frequencies within a certain range to very low levels. A notch filter is a type of band-stop filter with a very narrow range of frequencies and a sharp decrease in amplitude. Figure 5 shows the notch filter used in the pre-processing stage of this experiment.

Figure 5. 50 Hz notch filter.



Butterworth filter

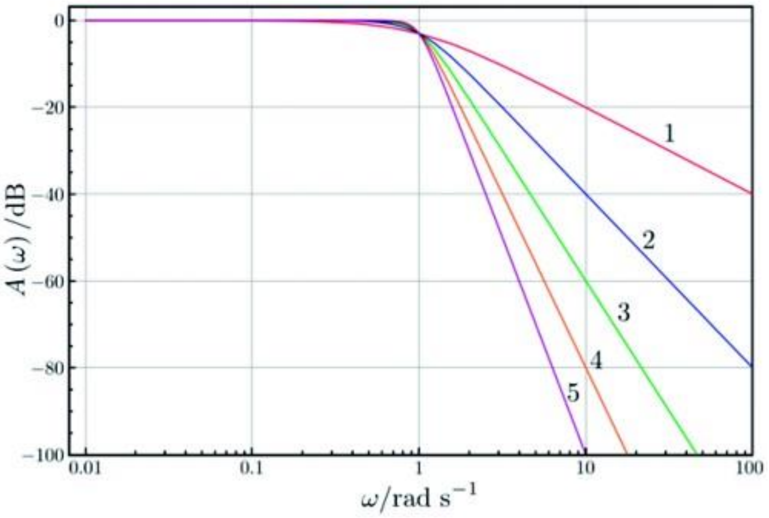
Butterworth filters are a type of band-pass filter, typically used as a low-pass filter, but can be modified to act as a high-pass filter, a band-pass filter or a band-stop filter. Band-pass filters allow any signals within a certain minimum and maximum range to be boosted, thereby amplifying frequencies in that range. Low-pass filters boost low frequencies while not affecting higher frequencies and likewise, a high-pass filter boosts higher frequencies and does not affect lower frequencies. Finally, band-stop filters cut off frequencies above a certain point or below a certain point.

Butterworth filters are designed to have a maximally flat frequency response in the passband, meaning no ripples unlike some other filters e.g., Chebyshev and Elliptic filters. The reason for this behaviour, as Butterworth stated in his paper is “An ideal electrical filter should not only completely reject the unwanted frequencies but should also have uniform sensitivity for the wanted frequencies” [33]. The roll-off of a Butterworth filters frequency response is determined by its order, i.e., a first-order filter would have a roll-off a -6dB, a second-order at -12dB, a third-order at -18dB, etc.

$$G^2(\omega) = \left|H(j\omega)^2\right| = \frac{{G_0}^2}{1+\left(\frac{j\omega}{j\omega_c}\right)^{2n}} \quad (2)$$

Equation (2) shows the formula for the transfer function of a Butterworth filter, where n is the order of the filter, ω_C is the cut-off frequency and G_0 is the DC gain. By increasing the order of the filter (increasing n), the function gets closer to approximating a rectangular filter (Figure 6). Frequencies below the cut-off frequency (ω_C) will be passed with the DC gain (G_0) and all other frequencies will be supressed, which means that frequencies above the cut-off frequency are reduced in accordance to the frequency response of the Butterworth filter.

Figure 6. Changing the order of a Butterworth filter.



Fast fourier transform

A Fast Fourier Transform (FFT) takes a signal in the time domain and outputs the frequency components of that signal. An Inverse Fast Fourier Transform (IFFT) takes a set of frequency components and reconstructs them into the original time-series signal. The FFT algorithm calculates the Discrete Fourier Transform (Equation 3) of the data, albeit faster. There are several different types of FFT algorithm, however the most common is the Cooley-Tukey algorithm [12] which is the implementation used in this experiment.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N},\ k = 0, \ldots, N-1 \quad (3)$$

It is important to note that when sampling a signal, it is vital that Nyquist’s theorem is used in order to ensure that the original signal can be reconstructed. In other words, the highest frequency that can be observed in the frequency domain data are equal to half the sampling rate of the original signal. In this experiment, the OpenBCI has a sampling rate of 250 Hz, meaning that the maximum frequency that can be accurately observed is 125 Hz. This is not an issue since the P300 frequency range that this experiment uses as features fall within the 0.5–12 Hz range.

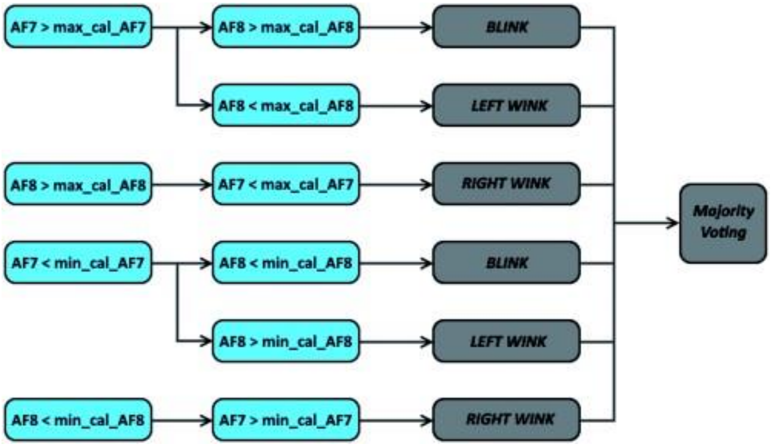
Feature extraction

EMG features in this experiment simply consists of the minimum and maximum voltages over a 1 s sample of data (2 features per electrode; 4 features in total). These features are gathered by the front 2 electrodes in the OpenBCI (AF7 and AF8). EEG features are a little more complicated, where only data from 250 ms after the P300 speller flash to 600 ms after the flash is considered to be relevant data. The features consist of the minimum and maximum voltages of the relevant time-series data, the minimum and maximum amplitudes in the frequency domain data (filtered FFT of the relevant data) and also the corresponding frequencies at the minimum and maximum amplitudes in the frequency domain data (6 features per electrode; 36 features in total). The EEG features are gathered by the rear 6 electrodes in the OpenBCI device (Pz, P3, P4, PO3, PO4 and Oz).

Classification

Determining the class of the EMG signal is a simple task of statistical classification (Figure 7), by comparing the data stored in the EMG user profile with the data given by the system. First the classifier cheques for the presence of a blink, which is validated when both the left and right electrode voltages exceed the maximum voltages for the respective electrodes stored in the user profile or the left and right voltages are less than the minimum values stored in the calibration file. If the left electrode voltage is greater than the maximum value in the calibration file but the right electrode is lower than the maximum value in the calibration, the data is classified as a left wink and if the left electrode is less than the minimum value in the calibration file and the right electrode is greater than the minimum value in the calibration file, the data is also classed as a left wink. Likewise, the data is classified as a right wink if the right electrode voltage is greater than the maximum right electrode calibration file value and the left electrode is less. If the right electrode value is less than the data in the calibration file and the left is greater, then that data are also classified as a right wink.

Figure 7. EMG classification process.

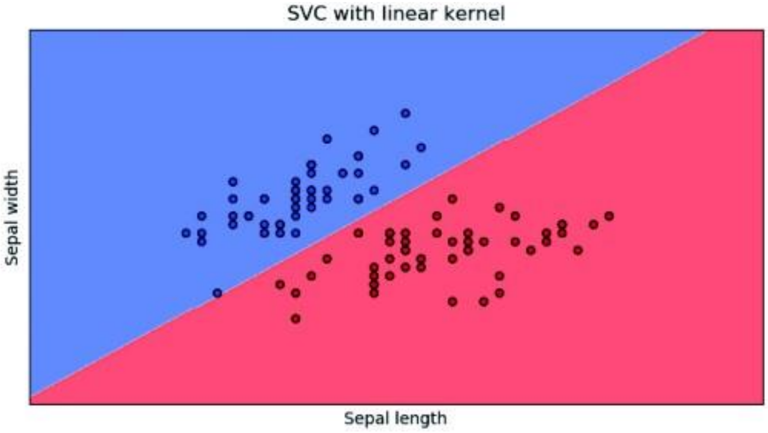


In the event that more than one of set of criteria is fulfilled, a majority voting algorithm is used to determine the more likely class. In the event of a matching number of different classes, the system will return an inconclusive decision and no further action is taken. EEG calibration involves getting a set of training data with which to train the classifier. The features of the P300 training set are described in the feature extraction section (under data acquisition) and the ground truths created from these features are fed into SVM. The ground truth of a data point in P300 classification is whether or not it is a P300 signal (two-class classification problem). Once the SVM is trained, it can be used during the testing phase.

Support vector machines

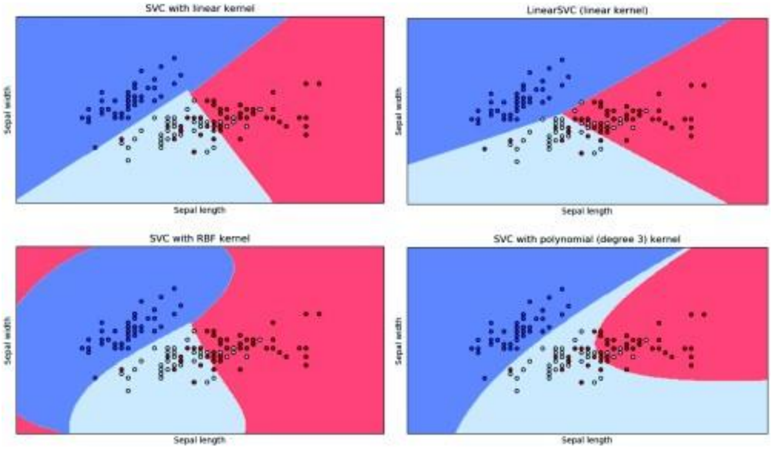
SVMs are a type of algorithm [34] that was originally designed for two-class classification problems (Figure 8), however multiclass SVM classifiers also exist. SVMs work by creating a hyperplane between a number of classes. Hyperplanes are essentially an X-dimensional vector of best fit, where X is the number of features. The calculation of the hyperplane occurs during the training stage, where a set of data, with known ground truths are used to calculate the best hyperplane where the maximum number of data points in the training set would be correctly classified, by adjusting hyperplane parameters to get the minimum value from the cost function. By observing which side of the hyperplane a data point falls on, the class of that data point can be determined.

Figure 8. Linear SVM for two-class classification.



There are many equations for the calculation of hyperplanes, some of which are linear while others are non-linear (Figure 9). Standard linear kernels are the most common type of SVM however this can be very limiting for complex relationships, for example a non-liner (e.g., quadratic, cubic) relationship could not be reliably classified via linear SVM. This issue is addressed by non-linear SVM equations, consisting of more parameters than linear SVMs that are updated during training to get a better reflection of data relationships.

Figure 9. Different SVM kernels.

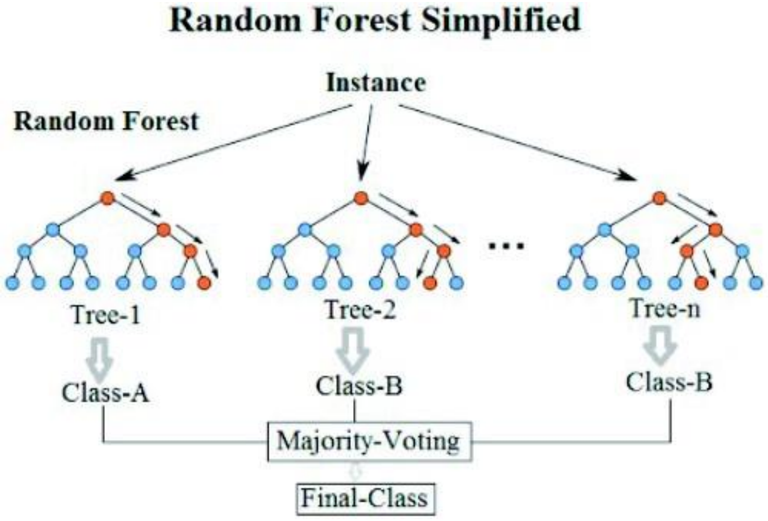


While linear relationships would only use gradient and y-intersect as parameter (y = mx + c), non-linear relationships can be much more flexible. Examples of common non-linear SVM kernels include, radial basis function (RBF) and polynomial kernels.

Decision trees and random forests

Decision trees [35] are algorithms that can be used for classification or regression. There are two distinct types of decision tree; classification trees (which use discrete values as inputs) and regression trees (using continuous values as inputs). While classification assigns a class to a dataset, regression provides a probability that the dataset falls within a certain class (Figure 10).

Figure 10. Example of a decision tree.



Random forests [36] consist of an ensemble of decision trees. Like regular decision trees, there are two types of random forest; classification forests and regressive forests, depending on the type of decision tree used when generating the random forest. In classification forests, the result of classification is the mode (most frequent occurrence) of the decision trees in the model; essentially a majority vote. Regressive random forests calculate the mean predictions of regression trees. The benefit of a random forest is a more robust model over its decision tree counterpart while addressing the issue of overfitting by looking at the output of all trees as a whole.

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x') \qquad (4)$$

Random forests can also make use of bootstrap aggregation (also known as bagging), which reduces the variance of a model without increasing bias, leading to more accurate results. Tree bagging (Equation (4)) consists of repeatedly selecting random samples and fits trees to these samples. The bagging algorithm for random forests has a single variable B which is the number of times to bag, the optimal number of which can be found through observing the change in error as B is changed; usually using cross-validation or by observing the out-of-bag error.

Robotic control protocols and experimental scenarios

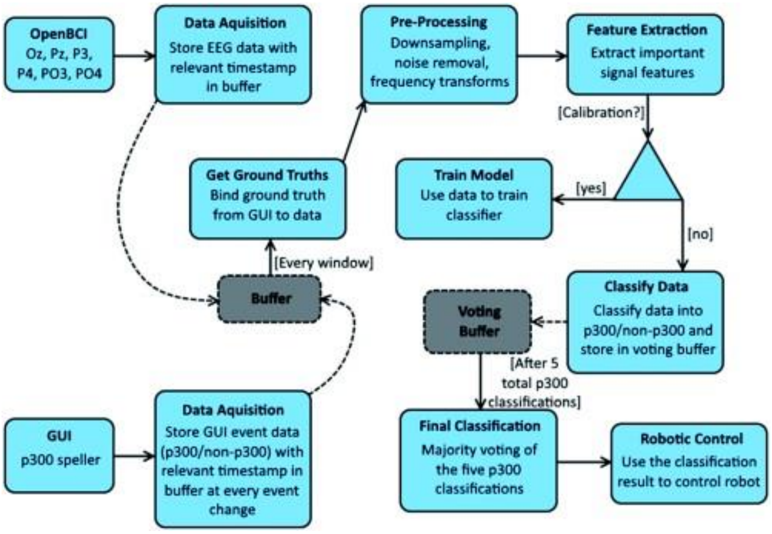
This section details an outline of the system data flow and details around the use of control protocols.

Data flow

The system is developed to convert EEG and EMG data into usable commands that can be executed by the UArm in accordance with the control protocol. The process has two distinct components; the EEG component (focussing on P300 events) and the EMG component (artefacts caused by muscle movement). Since P300 classification is a relatively slow process and therefore not suitable for fine motor controls, the EEG data will be used to carry out predefined procedures, as described in the results section, as part of the currently selected control protocol. EMG data has a much faster classification time, meaning that it is suitable for fine motor controls and is therefore used for controlling the stepper motors in the robotic arm.

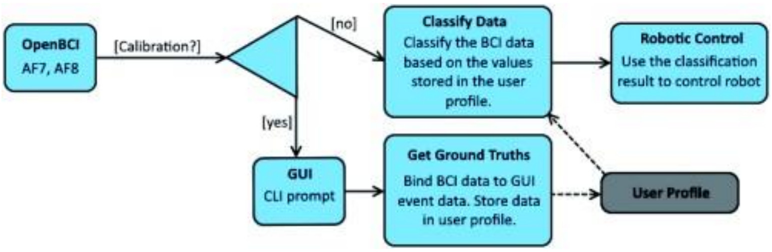
Figure 11 shows the flow of data within the EEG component of the system. A graphical user interface will be present at all times over the course of the experiment (both calibration/training and testing). By checking the state of the GUI (the 6x6 matrix, consisting of the characters A-Z and numbers 0-9) stored in the buffer at time *t* and the EEG data at time *t*, the ground truth of the data can be established. Ground truths are a collection of labelled data (essentially key-value pairs for data, where the key is the label and the value is the data). The purpose of ground truth is for the training of data classifiers. This data is then processed, has features extracted from it and is used to train the classifier if in the training stage, otherwise it is classified by the model and the result is stored in a voting buffer. Once there have been a total of 5 positive P300 classifications, majority voting is carried out to give the prediction for the P300 trigger. The classified result is used to carry out a pre-defined procedure on the robotic arm (mapping from EEG to arm movements is defined in the results section). To summarize, EEG signals are given labels based on training data and these labels are then used to control the robotic arm.

Figure 11. Overview of the P300 data flow.


OpenBCI
Oz, Pz, P3, P4, PO3, PO4
Data Aquisition
Store EEG data with relevant timestamp in buffer
Pre-Processing
Downsampling, noise removal, frequency transforms
Feature Extraction
Extract important signal features
[Calibration?]
Get Ground Truths
Bind ground truth from GUI to data
Train Model
Use data to train classifier
[yes]
[no]
[Every window]
Buffer
Voting Buffer
Classify Data
Classify data into p300/non-p300 and store in voting buffer
[After 5 total p300 classifications]
GUI
p300 speller
Data Aquisition
Store GUI event data (p300/non-p300) with relevant timestamp in buffer at every event change
Final Classification
Majority voting of the five p300 classifications
Robotic Control
Use the classification result to control robot

EMG processing is a much simpler task than its EEG counterpart (Figure 12). If the system is in the training stage, a command line interface (CLI) will be shown, guiding the user through calibration process for the EMG signals: blink, left wink and right wink, which are classified and used to control the robotic arm. Features are then extracted from this calibration session (consisting of maximum and minimum voltages for each EMG event) and saved into a user profile. This data is used for classification during the testing stage. If the real-time EMG data is sufficiently close to the data in the user profile, the classification returns the corresponding value and this result acts upon the robotic arm in accordance with the control protocol, detailed in the results section.

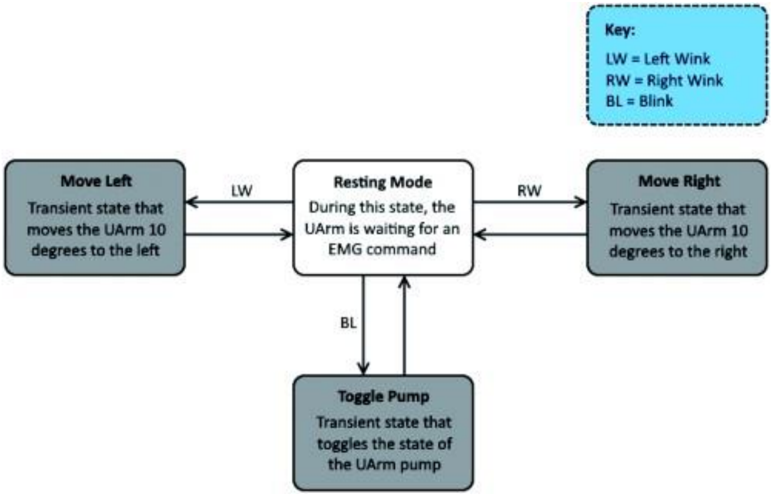
Figure 12. Overview of the EMG data flow.


OpenBCI
AF7, AF8
[Calibration?]
[no]
[yes]
Classify Data
Classify the BCI data based on the values stored in the user profile.
Robotic Control
Use the classification result to control robot
GUI
CLI prompt
Get Ground Truths
Bind BCI data to GUI event data. Store data in user profile.
User Profile

Robotic control in experiments

In the EMG experiment, a left wink will move the UArm 10° to the left, a right wink will move the UArm 10° to the right and a blink will toggle whether the UArm vacuum pump is active or not (Figure 13). In the EEG experiment, the EMG labels are switched out for P300 speller values ("L" for left, "R" for right and "P" for pump).

Figure 13. Robotic control protocol using in system testing.


Key:
LW = Left Wink
RW = Right Wink
BL = Blink
Move Left
Transient state that moves the UArm 10 degrees to the left
LW
Resting Mode
During this state, the UArm is waiting for an EMG command
RW
Move Right
Transient state that moves the UArm 10 degrees to the right
BL
Toggle Pump
Transient state that toggles the state of the UArm pump

EMG experiment

The first experiment consists of using EMG signals (using the protocol in Figure 13) to move the robotic arm from its starting location to a point where a card is located, toggling the UArm pump (thereby picking up the card), then moving the robotic arm back to its starting point and turning the pump off again. This process is carried out a total of five times, each of which is timed (every repetition is its own trial). Additionally, the number of attempted EMG events is recorded and compared to the number of EMG events actually correctly classified.

EEG experiment

This experiment involves using only EEG data gathered from the P300 speller programme to control the UArm. Subjects must first focus on the following characters to control the UArm {"P": toggle pump, "L": move left, "R": move right}. To complete this experiment, the users must first turn the pump on, move the arm to the left, move the arm to the right and finally toggle the pump off again (Sequence: P, L, R, P). This experiment will only be carried out once per subject, since the experiment takes a while to complete and avoiding fatigue is important to obtain consistent results. The time each user takes to complete the experiment is recorded along with the number of incorrect classifications.

Calibration

The EMG experiment starts by asking for a filename linking to the profile of the current user. If no such filename is found, a new EMG calibration session is run. The calibration process begins by asking the user to blink ten times, then to left wink ten times and finally to right wink ten times, leading to a total calibration time of 30 s. The data stored in the profile is the average minimum voltage in the AF7 electrode, the average minimum voltage in the AF8 electrode, the average maximum voltage in the AF7 electrode and the average maximum voltage in the AF8 electrode; each of which is recorded for the three distinct EMG events (12 data records in total).

For the calibration of the EEG experiment, the system also asks for a filename linking to the appropriate user profile at start-up. If that profile does not exist, then a new P300 training session is run (Figure 14). The GUI is a standard 6x6 P300 speller matrix, consisting of characters A-Z and the numbers 0-9 (Figure 15). The duration of each flash is 0.1

ms and the inter-stimulus interval is 0.25 ms. The background colour of the speller is grey (#999999), the colour of the inactive characters is black (#000000) and the colour of the active characters (flashed characters) is white (#FFFFFF).

Figure 14. Example P300 training session.

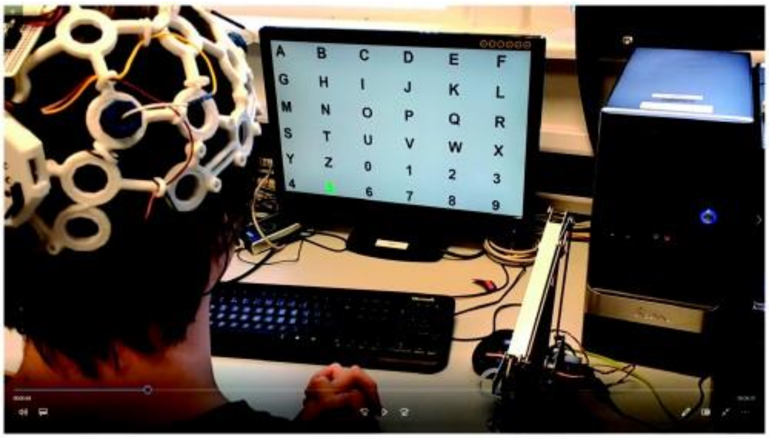
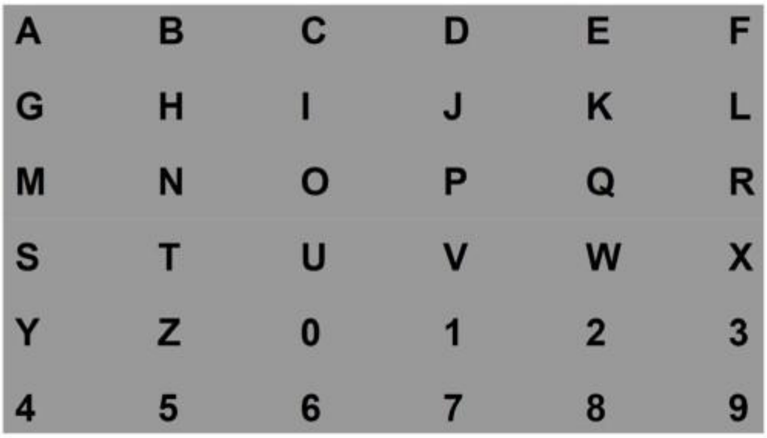


Figure 15. 6 × 6 P300 speller matrix.



A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

When the P300 training session is initiated, a number of trials and the length of each trial is specified (default values are; trial number: 3, trial length: 25). A single character then flashes green (#00FF00), indicating that the user should focus on that character for the duration of that trial. Random rows and columns are then flashed for a total of the trial length. This process is repeated for the number of trials. After all the trials are complete, the event data stored in the global buffer is then merged with the EEG data stored in the buffer to establish the ground truths (based on the EEG features described in the previous section) for the two data classes (target and non-target). This data is then used to train the classifier and the classifier parameters and weights are stored in a user profile. Since different users may have different P300 responses (e.g., brain latency), individual user profiles may serve to increase the classification success rate.

System usability measure

In addition to the objective metrics gathered in the experiments, subjective metrics shall also be taken in the form of a short survey of user experience, following the system usability scale (SUS) [37] comprising of ten questions (Table 1) (on a scale of one to five).

Table 1. SUS survey for system usability.

Q1	I think that I would like to use this system frequently.
Q2	I found the system unnecessarily complex.
Q3	I though the system was easy to use.
Q4	I think that I would need the support of a technical person to be able to use this system.
Q5	I found the various functions in this system were well integrated.

Q6	I thought there was too much inconsistency in this system.
Q7	I would imagine that most people would learn to use this system very quickly.
Q8	I found the system very cumbersome to use.
Q9	I felt very confident using the system.
Q10	I needed to learn a lot of things before I could get going with this system.

These data will be used to gauge whether the system is usable in its current state. The main criteria that the system aims to fulfil are that it is easy to learn, easy to use and that users are comfortable using the system for extended periods of time.

According to the SUS scoring method, each odd item in the questionnaire has one subtracted from it, whereas each even item is subtracted from five. These values are then summed together and multiplied by 2.5, giving a score out of 100. It should be noted that SUS scores are not percentages and in order to convert the scores into a percentile ranking, normalization of the usability scores is necessary. Sauro et al. [38,39] carried out an analysis of over 5,000 user scores over almost 500 studies of varying application types, leading to the conclusion that the average SUS score is 68 with a standard deviation of 12.5. As a result of this study, it can be determined that a SUS score of 68 is in the 50th percentile, so a SUS score of greater than 50 would be above average and a score lower than 50 would be below average. SUS scores can also be translated into several distinct letter grades (Table 2), which can be easier to interpret than a numerical score.

Table 2. Grading scale for SUS scores.

Letter grade	Numerical score range
A+	84.1–100
A	80.8–84.0
A−	78.9–80.7
B+	77.2–78.8
B	74.1–77.1
B-	72.6–74.0
C+	71.1–72.5
C	65.0–71.0
C−	62.7–64.9
D	51.7–62.6
F	0–51.6

Experimental results

This section reports the results of the experiments detailed in the previous section. All data gathered for each of the experiments is also analyzed in this section.

Offline data analysis

Prior to real-time experiments with a number of subjects, labelled EEG data was used to determine the classification success rate of two different classifiers (SVM using RBF kernel and a random forest). While initial results were not too promising, after some tweaking of classifier parameters the success rate started to increase. SVM gave a true positive rate of 56% and a false positive rate of 41% at its best configuration. The random forest provided considerable improvements over that with a true positive rate of 64% and a false positive rate of 44%. This indicates that a random forest classifier should be adequate for the classification of real-time EEG data, due to a noticeable increase in true positive classification, with a minimal increase in false positive rate.

EMG experiment

The results of the EMG experiment (shown in Table 3), indicate that the success rate of classification and the speed at which users were able to complete the task make EMG a very feasible approach to controlling assistive technology.

Table 3. EMG experiment results.

	ID #1	ID #2	ID #3	ID #4
Average Time (s)	5.610	5.886	8.204	4.905
Average Success Rate (%)	100	100	71.429	100

The average time is simply the mean time that it took each user to complete each trial. Average EMG success rate is calculated using Equation (5), where EMG_c is the number of classifications for the given EMG event and EMG_a is the attempted number of the given EMG event.

$$Success\left(\%\right) = \left(\left(\frac{blinks_c}{blinks_a} + \frac{left_c}{left_a} + \frac{right_c}{right_a}\right)*100\right)/3 \qquad (5)$$

while three of the users had a 100% success rate with EMG detection and classification, one of the users had a lower than average EMG detection rate, which may have been caused by the BCI not being a perfect fit (i.e., the electrodes did not have perfect contact with the skin). In an attempt to counter this low success rate, recalibration of EEG detection was carried out for that user (to rule out faulty calibration), but this had little impact on the success of the system for that user.

EEG experiment

Table 4 shows the results of the EEG experiment. The results show that, compared to the EMG experiment, the same task took significantly longer and had a much lower success rate. One of the subjects was unable to complete the experiment due toinsufficient detection of P300 signals, which was likely due to complications when fitting the BCI device, which would also explain the below average result obtained in the EMG experiment.

Table 4. EEG experiment results.

	ID #1	ID #2	ID #3	ID #4
Time (s)	145.85	235.9	DNF	216.21
Success Rate (%)	50	30.769	DNF	33.333

Since there were 36 possible outcomes for each VEP event (since the system uses a 6x6 P300 matrix) the average classification success rate over all users of 38.023% is far greater than random guessing (which would have a success of 2.778%) demonstrating that the classifier is able to distinguish between p300 and non-p300 EEG signals.

Subjective results

This section details the results of the SUS survey (Table 1) that all users were given at the end of their trials. These results will be cross-referenced with the objective measures of success for the given experiments to determine a conclusion as to the overall system success (Table 5).

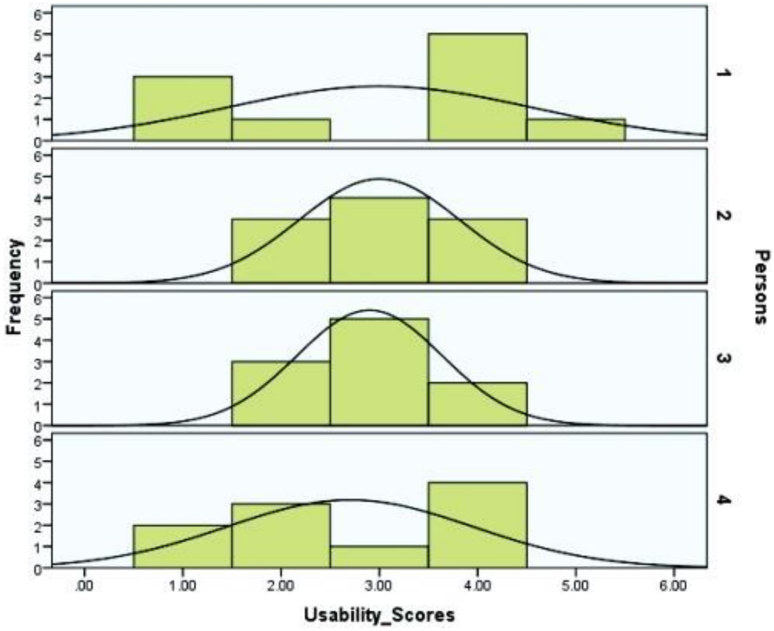
Table 5. SUS survey results.

	ID #1	ID #2	ID #3	ID #4
Q1	1	4	3	1
Q2	2	3	2	2
Q3	4	4	4	4
Q4	1	2	2	2
Q5	4	3	3	4
Q6	4	2	2	3
Q7	5	3	3	4
Q8	4	2	3	2
Q9	4	4	4	4
Q10	1	3	3	1
SUS scores	65	65	62.5	67.5

SUS Stats	Mean, M = 65; Std. Deviation, SD = 2.04; Std. error, SE = 1.02
	95% Confidence interval for mean:
	Lower bound 61.7 and upper bound 68.24;

Using the standard process for analyzing SUS scores [35], the mean SUS score was calculated as 65 with confidence interval for the mean between 61.7 and 68.24. This final score indicates that the system is very close to the average usability score of 68 which is accepted score for more than 50 percent of all tested systems and applications. One-sample t-test was carried out to test if there was a significant difference of the mean SUS score observed from the target value of 68 to achieve a reasonably good system. We found that the observed average user experience of the system measured indicated by the SUS scoring (Mean = 65, SE = 1.02) was less than the reported average SUS score of 68. This difference was not statistically significant t(3)= −2.93, *p*>.05. When translated into letter grades, the observed average SUS score is mapped to a usability rank of "C" (Figure 16), which is acceptable as it is higher than 40 percent of all tested systems and applications [40]. From a usability standpoint, there are no concerning usability issues that would prevent the system from further testing (including clinical trials in the future).

Figure 16. Histogram plot of usability scores.



The majority of user scores are largely congruent with one another, particularly users 2 and 3, as clearly shown in Figure 16. The average usability score seems to be around 3 across all the users surveyed. The significant (*p*<.05) correlation between Q1 and Q10 indicates the fact that as the users didn't need to learn a lot before using the system, they wanted to use the system more often. As for the highly significant (*p*<.001) correlations, Q5 and Q10 are correlated (−1) with the likely cause that the good integration of features in this system resulted in them not needing to learn a lot before getting started. So the better the integration of features within the system, the less need to learn in order to operate the system.

Q6 and Q7 are also highly correlated, with a possible reason that the lack of inconsistency in the system resulted in a consensus where it was deemed that most people would learn how to operate the system in a short period of time. This indicates that the speed at which new users would learn to use the system is directly correlated to the inconsistency in the system. The conclusion that can be drawn from this finding is that if the system's level of inconsistency is reduced, users would be able to learn to use the system in a shorter period of time.

Conclusion and future work

This paper shows that EMG is a very reliable method of assistive technology control provided that the calibration methodology detailed in this paper is followed. It is also possible that such a system could be used in a neurorehabilitation environment for certain patients that are not suffering from severe full-body paralysis, since the current system only needs the use of a few muscles on the forehead.

When it comes to using EEG to control assistive technology, it is certainly possible, however the current system does not provide an adequate level of success to use within a neurorehabilitation environment. The EEG VEP events used are also very slow to classify, given the amount of data required to ensure even somewhat reliable classification, which makes it impractical for small precise movements, though it may have some application for more complicated, pre-scripted routines.

Given the shortage of available EEG data, it is unlikely that LSTM would have been effective for classification in our system, so we opted for the SVM. Additionally, since the experiment is a two-class problem, SVM would be more suitable for fast and reliable classification. However, possible future work could incorporate a LSTM classifier.

While the SUS results indicated that system usability is not a barrier for moving this system into clinical trials, a result of "C" is not as high as desirable. Future work will likely involve improving system usability, potentially with the aid of detailed questionnaires whereby users suggest improvements to the system that would make it easier to use.

Future work would also focus on improving the robustness of the EEG component of the system, by experimenting with other types of VEP events (such as SSVEP and RSVP), improving the pre-processing of the system (using wavelet transforms to get the time-frequency domain data) and improved feature extraction (for example PCA/CPCA could be used).

References

1. King CE, Wang PT, McRimmon CM, et al. The feasibility of a brain-computer interface functional electrical stim-ulation system for the restoration of overground walking after paraplegia. [updated 2015 Aug; cited 2018 Apr 9]. Available from: https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-015-0068-7.

2. Dokkum LEHV, Ward T, Laffont I. Brain computer interfaces for neurorehabilitation – its current status as a reha-bilitation strategy post-stroke. [updated 2015 Feb; cited 2018 Apr 12]. Available from: https://www.ncbi.nlm.nih.gov/ pubmed/25614021.

3. B.C.A O. Neurorehabilitation of hand functions using Brain-Computer Interface. [updated 2015 Sep; cited 2018 Apr 12]. Available from: http://theses.gla.ac.uk/7245.

4. Irimia DC, Poboroniuc MS, Serea F, et al. Controlling a FES-EXOSKELETON Rehabilitation System by Meansof Brain-Computer Interface. [updated 2016 Oct; cited 2018 Apr 15]. Available from: https://ieeexplore.ieee.org/document/7781361.

5. Liberson WT, Holmquest HJ, Scot D, et al. Functional electrotherapy: stimulation of the peroneal nerve synchron-ized with the swing phase of the gait of hemiplegic patients. [updated 1961 Feb; cited 2018 May 30]. Available from: https://www.ncbi.nlm.nih.gov/pubmed/13761879.

6. Moe JH, Post HW. Functional electrical stimulation for ambulation in hemiplegia. [updated 1962 Jul; cited 2018 May 30]. Available from: https://www.ncbi.nlm.nih.gov/pubmed/14474974.

7. Pfurtscheller G, Muller GR, Pfurtscheller J, et al. "Thought" - control of a functional electrical stimulation to re-store hand grasp in a patient with tetraplegia. [updated 2003 Nov; cited 2018 Apr 24]. Available from: https://www.sciencedirect.com/science/article/pii/S0304394003009479.

8. Do AH, Wang PT, King CE, et al. Brain-computer interface controlled functional electrical stimulation system forankle movement. [cited 2018 Apr 24]. Available from: https://jneuroengrehab.biomedcentral.com/track/pdf/10.1186/1743-0003-8-49.

9. Badia SB, Morgade AG, Samaha H, et al. Using a hybrid brain computer interface and virtual reality system tomonitor and promote cortical reorganization through motor activity and motor imagery training. [updated 2012 Nov; cited 2018 Apr 24]. Available from: https://ieeexplore.ieee.org/abstract/document/6363609.

10. Pietro C, Silvia S, Federica P, et al. NeuroVirtual 3D: a multiplatform 3D simulation system for application inpsychology and neuro-rehabilitation. [updated 2014 Apr; cited 2018 Apr 24]. Available from: https://link.springer.com/chapter/10.1007%2F978-3-642-54816-1_15.

11. Ruiz AF, Arturo FC, Rocon E, et al. Exoskeletons for rehabilitation and motor control. [updated 2006 Feb; cited2018 Apr 24]. Available from: https://ieeexplore.ieee.org/abstract/document/1639155.

12. Cooley JW, Tukey JW. An algorithm for the machine calculation of complex fourier series. Mathematics of com-putation, Vol. 19, No. 90. [updated 1965 Apr; cited 2018 Jun 4]. Available from: https://www.jstor.org/stable/2003354?seq=1#page_scan_tab_contents.

13. Cochran WT, Cooley JW, Favin DL, et al. What is the fast fourier transform. Proceedings of the IEEE, Vol. 55,No. 10, pp. 1664–1674. [updated 1967 Oct; cited 2018 Jun 4]. Available from: https://ieeexplore.ieee.org/document/1447887.

14. Al-Fahoum AS, Al-Fraihat AA. Methods of EEG signal features extraction using linear analysis in frequency andtime-frequency domains. ISRN neuroscience. [updated 2014; cited 2018 Jun 4]. Available from: https://www.hindawi.com/journals/isrn/2014/730218.

15. Solomon. PSD computations using Welch's method. [updated 1991; cited 2018 Jun 4]. Available from: https://www.osti.gov/servlets/purl/5688766.

16. DAUBECHIES. The wavelet transform, time-frequency localization and signal analysis. IEEE Trans InformTheory. 1990;36(5):961–1005.

17. Soltani S. On the use of the wavelet decomposition for time series prediction. Neurocomputing. 2002;48(1–4):267–277.

18. Prochazka , J Kukal, O. Vysata Wavelet transform use for feature extraction and EEG signal segments classifica-tion. Proceedings of the 3rd international symposium on communications, control and signal processing (ISCCSP '08), pp. 719–722. [updated 2008 Mar; cited 2018 Jun 10]. Available from: https://ieeexplore.ieee.org/document/ 4537317.

19. Stoica P, Nehorai A. Study of the statistical performance of the Pisarenko harmonic decomposition method. IEEProc F Commun Radar Signal Process UK. 1988;135(2):161–168.

20. Awang SA, Paulraj MP, Yaacob S. Analysis of EEG signals by Eigenvector methods. Proceedings of the IEEEEMBS Conference on Biomedical Engineering and Sciences (IECBES '12). 2012:778–783. [cited 2018 Jun 10]. Available from: https://ieeexplore.ieee.org/document/6498164.

21. Übeyli ED. Analysis of EEG signals by implementing Eigenvector methods/recurrent neural networks. DigitalSignal Processing. 2009;19(1):134–143.

22. Übeyli ED. Analysis of EEG signals by combining Eigenvector methods and multiclass support vector machines.Comput Biol Med. 2008;38(1):14–22.

23. Jatupaiboon N, Pan-Ngum S, Israsena P. Real-time EEG-based happiness detection system. Sci World J.2013;2013:1.

24. Li M, Zhang M, Luo X, et al. Combined long short-term memory based network employing wavelet coefficientsfor MI-EEG Recognition. Proceedings of the 2016 IEEE international conference on mechatronics and automation
(ICMA '16). [updated 2016 Aug; cited 2018 Jun 112]. Available from: https://ieeexplore.ieee.org/document/7558868.

25. Hefron RG, Borghetti BJ, Christensen JC, et al. Deep long short-term memory structures model temporal dependencies improving cognitive workload estimation. Pattern Recog Lett. 2017;94:96–104.

26. Chapman RM, Bragdon HR. Evoked responses to numerical and non-numerical visual stimuli while problemsolving. Nature. 1964;203(4950):1155–1157.

27. Fazel-Rezai R, Allison BZ, Guger C, et al. P300 brain computer interface: current challenges and emergingtrends. Front Neuroeng. 2012;5:14.

28. Fazel-Rezai R, Gavett S, Ahmad W, et al. A comparison among several P300 brain-computer interface spellerparadigms. Clin EEG Neurosci. 2011;42(4):209.

29. Pan J, Li Y, Gu Z, et al. A comparison study of two P300 speller paradigms for brain-computer interface. CognNeurodyn. 2013;7(6):523–529.

30. David R. Evoked potentials in psychology, sensory physiology and clinical medicine. Netherlands: Springer;1972.

31. Sharbrough F, Chatrian GE, Lesser R, et al. American electroencephalographic society guidelines for standardelectrode position nomenclature. J Clin Neurophysiol. 1991;8(2):200–202.

32. Kuo FF, Kaiser JF. System analysis by digital computer. New York: John Wiley and Sons; 1966.

33. Butterworth S. On the theory of filter amplifiers. Experimental Wireless and Wireless Engineer. 1930;7:536–541.

34. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20(3):273–297.

35. Quinlan JR. Induction of decision tress. Mach Learn. 1986;1(1):81–106.

36. Breiman L. Random forests. Mach Learn. 2001;45(1):5–32.

37. Brooke J. SUS – A quick and dirty usability scale. Usability evaluation in industry. [cited 2018 Aug 31]. Availa-ble from: https://hell.meiert.org/core/pdf/sus.pdf.

38. Sauro J. Does prior experience affect oerceptions of usability. [updated 2011; cited 2019 May 14]. Availablefrom: http://www.measuringu.com/blog/prior-exposure.php.

39. Sauro J. Measuring usability with the System Usability Scale (SUS). [updated 2011; cited 2019 May 14]. Availa-ble from: http://www.measuringu.com/blog/sus.php.

40. Sauro J. SUStisfied? Little-known system usability scale facts. [updated 2011; cited 2019 May 14]. Availablefrom: http://uxpamagazine.org/sustified/.