Control of a Real-Time Neuro-robotic System

Hans W Tschohl[1]
School of Innovation, Design and Engineering, M.Sc.Eng Robotics
Mälardalens University, Västerås, Sweden
Email:[1]htl21001@student.mdu.se

Abstract-This Real-Time Neuro-Robotic system is used to control a prosthetic hand. This is done by obtaining brain signals, using Machine learning algorithms to process the signals and stream the processed signals to a prosthetic hand. The work for this system was dividend into three sections: signal Acquisition, Real-Time Processing, and Application.The Acquisition of the real-time brain signals is done using a headpiece which contains 16 EEG electrodes and streaming the signals from openBCI to matlab for processing. The Real-Time Processing makes use of machine learning algorithms to process the data. The prosthetic hand is connected to a roboRio which receives the real-time processed signal.

***Index Terms*—BCI, ICE, EEG, LSL, Robot Hand, SVM, KNN**

I. Introduction

PATIENTS who have suffered from severe neurological and cognitive disorders such as strokes, paralysing injuries or amyotrophic lateral sclerosis (ALS) can be helped to regain movement through Brain Computer Interfaces (BCI). A BCI is a potential alternative to the neurological pathways destroid by the severe neurological and cognitive disorders suffered by the patients through non-muscular communication and control channels connecting the brain signals to mechanical prosthetic limbs. The prosthetic limb is not connected to the subjects through their muscular or nervous system. (1)

The process of developing a prosthetic limb that can be used as any biological limb starts with understanding how brain signals can be measured and used to implement the physical action of the prosthetic limb. The work done in this paper was to read, process and implement brain signals in real time to simulate the opening/relaxing and the closing of a hand. This neuro-robotic system was thus divided into three sections Acquisition, Real-Time Processing, and Application.

A. Hypothesis

The data will go through some package loss in the aquisition stage which will cause the accuracy of the project to decrease. The ICA processing will yield an 80% accuracy in prediction. The hand will open and close 80% of the time that it is intended to.

B. Acquisition

The methods of acquiring brain signals can be divided into two ways: invasive and non-invasive. In this project, the method used for acquiring brain signals was the non-invasive Electroencephalogram (EEG). An 16 electrode EEG-Helmet, which consists of a frame holding EEG-electrodes and a cyton board, was used to acquire the data from the subject. The data would be streamed using a dongle or a wifi connection to the openBCI program. The openBCI program would then send data packages to matlab using an Internet Protocol (IP) for processing.

The main focus of this section is create a system that can stream the brain signals to a computing device and save the signals in a text file when the real-time processing program is offline and stream to the program when the program is online.

C. Real-Time Processing

The real-time processing is all about predicting if the subject needs the hand to close or stay relaxed. The process will make use of data processing techniques to predict what the subject needs. The processing program consists of pre-processing, feature extraction, sent through a data processing algorithm to be classified as closed or relaxed (1 and 0 respectively).

The pre-processing will make use of a band pass filter remove all unnecessary noise. The chosen feature extraction method is a Principle Component Analysis (PCA) and Independent Component Analysis (ICA) pair. Once the processing was complete, the data would then be streamed to the next component.

D. Application

The final step in this project is to use the data that has been streamed and processed in real-time to actuate a mechanical system. The mechanical system in this case is a 3D printed hand and forearm containing five motors (one for each finger) controlled by a roboRio wich is controlled by a LabVIEW program. An IP is used to stream the processed data from matlab to LabVIEW for actuation.

When a signal representing a closed hand is recieved, all five motors are actuated to simulate a closed hand. Respectively, when a signal is recieved indicating that the subject needs a relaxed or opened hand, motors will then return to their original position to simulate a relaxed or open hand.

II. Background

BCIS look for activity patterns in the brain in real-time and try to make sense of the data it receives through data processing techniques.

A. Acquisition

1) Brain Signal Measurement Methods

There are two main methods used when acquiring brain signals: Invasive methods and Non-invasive methods.

Invasive methods require surgical implants underneath the skull. For example Electrocorticography (ECoG) requires implants beneath the skull directly on the surface of the brain to receive information and record the electrical activity from the brain(2) and the method Intracortical Sensors used in Intracortical Brain Computer interfaces (iBCIs), require the electrodes to be implanted inside of the brain tissue.(3)

Non-invasive methods, the measurement methods that are used in this project, do not require any surgery to measure brain signals. Some non-invasive methods include Electroencephalogram (EEG) and Magnetoencephalography (MEG).

The EEG method makes use of a headpiece which contains several electrodes (16 in the case of this project) that will be placed directly on the scalp of the subject to measure the electrical potential difference between an electrode and one reference electrode which is ideally placed on the earlobe of the subject.

MEG measures the magnetic fields produced by your brains electrical currents by using a super conducting quantum interface device (SQUID) and a computer. The magnetic field distribution is generally overlapped with an image of the brain to visualise the active brain areas.(4)

2) Data Streaming

There are a number of methods to stream data in real-time. In this project the main three considered methods were User Data-gram Protocol (UDP), Transmission Control Protocol (TCP) and Lab Streaming Layer (LSL).

UDP is an unordered data delivery protocol with higher end to end delay and packet loss rate than TCP but it has a better throughput than TCP, according to test done by Madhuri and Chenna (5). The most apealing feature of UDP is its low latency which allows for high volume of data to be processed with minimal delay (Higher speeds)

TCP is an ordered data delivery protocol which is allows for flow and congestion control. TCP has a low packet loss rate which is desirable for sensitive data such as in the case of this project(5). The main disadvantage of TCP is that it is slower than UDP.

LSL provides precise timing and synchronization when handling data that has been sampled at differing rates and makes it possible to receive data from multiple sources by making use of ping. LSL can be easily integrated to many programming languages such as matlab. (6)

B. Real-Time Processing

1) Pre-Processing

Filtering is important because EEG are subject to a lot of noise and artifacts that can come from the potential difference caused by muscle movements or a rapid eye activity. The neuro-information that holds any inportance lies under 100Hz and in most cases under 30Hz, acording to Sanei and Chambers (7). This projects makes use of the change in frequencies over time, and this is observed well with the use of a frequency analysis technique that will preserve the time information, such as the Hilbert transform or the wavelet analysis. Although no filtering is needed before using the Hilbert transform, it is recomended to use a bandpass filter on the data to get a more evenly weighted power throughout the frequencies(8).

Principle component Analysis (PCA) is used to construct weights based on the co-variance of given variables (in the case of this project, the variables are the EEG electrodes) in such a way that the components generated can explain the variance of the data while being uncorrelated to all other components. PCA allows for specific features that may be difficult to identify in unfiltered data to be highlighted.(8)

2) Feature Extraction

Independent Component Analysis (ICA) is a feature extraction method used to separate lineally mixed signals. Before ICA can be applied to a data set, the data needs to go through pre-processing. The ICA pre-processing includes centering and whitening. Centering is to ensure that the data has a zero mean and the whitening is used to ensure that the data is uncorrelated. (9) These techniques are done by applying PCA before applying ICA to the data. This gives the maximum effectiveness of ICA.

Common spatial pattern (CSP) is a mathematical procedure used to maximise the variance between two windows in a multivariate signal. CSP is commonly used in EEG data processing.(10)

3) Classification

Support Vector Machines (SVM) is a supervised learning model in machine learning used to analyse data for classification analysis of new data presented to the model. Classification, regression and other tasks preformed by an SVM is done through the construction of a hyperplane or a set of hyperplanes, depending on the data. The ideal data seperation is done by the hyperplane or set of hyperplanes wich have the largest reach to the closest training-data point of any class. (11)

K-nearest neighbor (KNN) algorithm is a non-parametric supervised learning method used in classification and regressing. Although it is commonly used for statistics, it can also be applied in machine learning algorithms. The function approximation, in KNN, is only local. Due to distance being the classification factor, the pre-processing can be vital for accurate results in the case of KNN. (12)

C. Application

1) Mechanical Hand

A similar mechanical hand was developed by Park, Bae, Park, and Baeg (13), although their robot hand had three fingers and a thumb, along with 16 degrees of freedom. The hand developed in this project has four fingers and a thumb and 5 degrees of freedom. Their hand had much more control and could move in ways that the hand developed in this project can not.

2) Motor

Paralax servo motors provide a 180° of movement and position control. These servos are easily interfaced with any Pulse width modulation (PWM) device and makes use of high precision gear made of polyacetal resin to ensure smooth operation without backlash. (14)

An alternative may be a Hitec servo. A Hitec servo has good durability and the ability to be easily modified. Due to the output shaft being supported on the both the top and the bottom and that the potentiometer is indirect drive, this motor can be modified for wheeled applications if necessary. This servo is great for inexpensive robot applications. (15)

3) roboRio

The roboRIO is a portable reconfigurable I/O device that can be used with ease for Robotic, Automation or Mechatronic systems. (16). It is created by the company National Instruments wich gives it some advantages such as being compatible with LabView, one of National Instruments visual programming languages (VPL). (17)

III. Method

A. Acquisition

1) Hardware

THE complete data acquisition kit from openBCI included a Ultracortex mark IV 3D-printed headset designed by openBCI, a cyton board with a daisy module equip ed, USB dongle for streaming data to a computing device and TDE-200 dry EEG electrodes. (18) (19)

The Ultracortex mark IV 3D-printed headset allows for a maximum of 35 electrodes, however only 16 electrodes will be used in the standardised 10-20 configuration. The openBCI program used on the laptop did not label the electrodes as to the 10-20 standard. The electrode labels needed to be entered and arranged on the openBCI graphical user interface (GUI) manually.

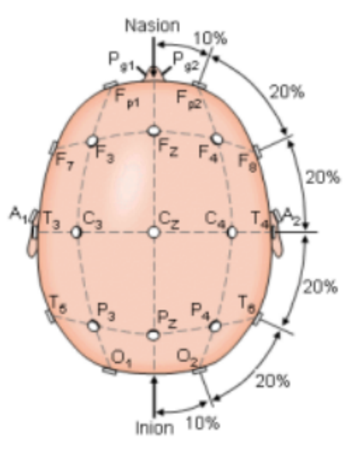

Figure 1. Standardized 10-20 configuration system electrode Placement(20)

Electrode labels and corresponding 10-20 labeling					
Project	10-20	Project	10-20	Project	10-20
1	F_{p1}	7	O_1	13	P_4
2	F_{p2}	8	C_Z	14	P_Z
3	C_3	9	F_3	15	T_5
4	C_4	10	F_4	16	T_6
5	T_3	11	O_2	Left ear	A_1
6	T_4	12	P_3	Right ear	A_2

Table I
The project electrode placement names and the equivalent standardized 10-20 electrode placements names

Once the configuration of the electrode layout on the openBCI GUI was done according to the table above and placed on the subjects head, it was ready to stream the brain signals. This was done with the use of the provided dongle mentioned above. The dongle is plugged into the laptop and received the data being streamed via a Bluetooth 4.0 wireless communication protocol.

2) Subject Task

The subject was tasked with imagining closing their hand into a fist 6 second and then relaxing their hand for 6 seconds. This was repeated for a total time of 120 seconds or two minutes. This gave us a total of 10 open/relaxed hand trials and 10 closed hand trials. To Prevent any external stimuli/distractions, black felt pin boards were placed on the subjects left, right, and in front of them with a white dot pinned to the board in front of them to keep their eyes from wondering. The subject was given audible notice when they should change their imagined state of their hand.

3) Real-Time Streaming and Configuration

$$T = 1/f => 1/125 = 0.008 Seconds \tag{1}$$

.

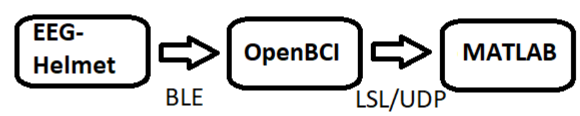

Figure 2. (1) The helmet uses a bluetooth communication to connect with the OpenBCI GUI. (2) The data is represented in the OpenBCI GUI, impedance and connectivity are checked in this block before streaming the data further to the next block. (3) The data collected by the 16-channels is streamed via the OpenBCI GUI using the LSL/UDP protocol over to MATLAB.

To be able to use the LSL protocol, the LSL library needs to be loaded and configured. When the connection is recognised between the LSL protocol and the openBCI GUI, a message stating that the data is now streaming will be displayed.

Matlab receives the data and saves each sample data into a matrix. This readies the data for the prediction and training.

B. Real-Time Processing

1) Pre-Processing

To prepare the data for processing, usable data needs to be extracted from the given data. This was done using a band-pass filter and centering all outliers. The outliers are the mostly coming from the muscular stimuli produced by the subject's eyes blinking.

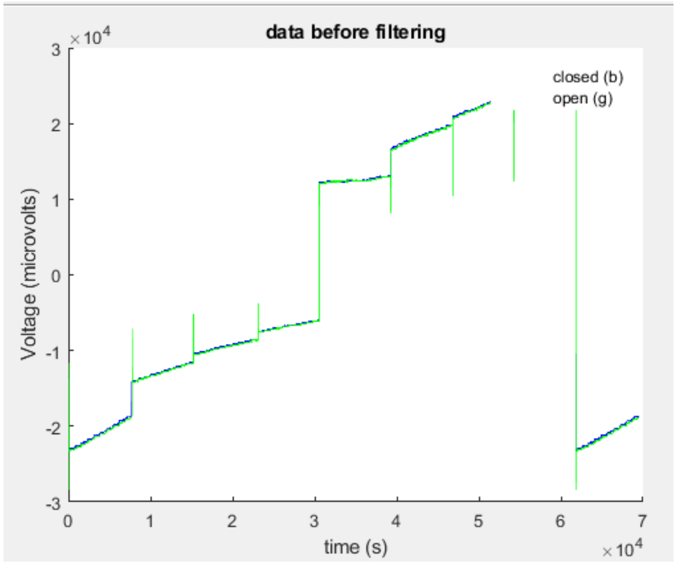

Figure 3. This is the raw, unprocessed signal produced by the brain in one electrode throughout all trials. The y-axis represents the electrical potential difference between the electrode and a basis signal measured from the subjects earlobes. The x-axis represents the time of the recording

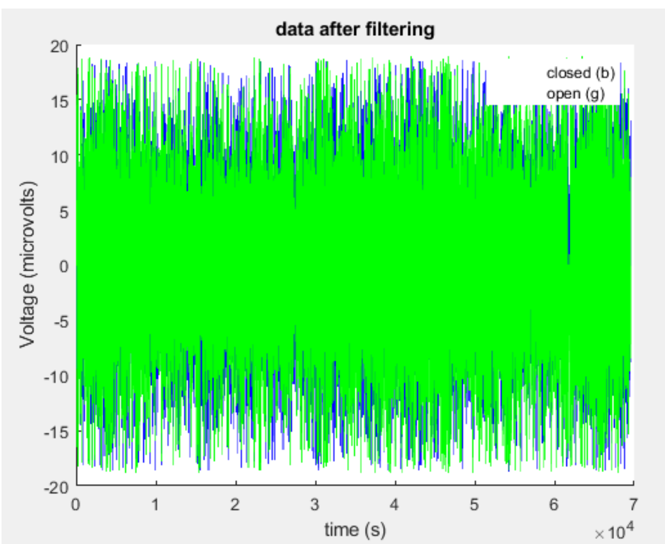

Figure 4. This is the signal produced by the brain in one electrode throughout all trials after passing through the band-pass filter. The y-axis represents the electrical potential difference and the x-axis represents the time of the recording

2) *Feature Extraction*

To apply ICA to the data, the data needs to go through more pre-processing. The data needs to be sent through a PCA process in Matlab to center and whiten the data to ensure that the data is uncorrelated. Once the extra pre-processing is done, it goes through the reconstructive ICA (RICA) function in Matlab.

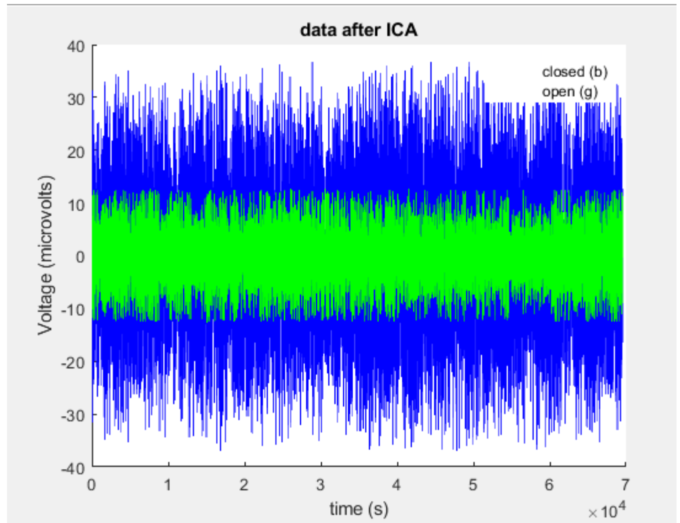

Figure 5. This is the signal produced by the brain in one electrode throughout all trials after being processed by the RICA function in Matlab. The y-axis represents the electrical potential difference and the x-axis represents the time of the recording

The feature used in the classification is the natural logarithmic variance of the channels in a trial. A comparison between the average natural logarithmic variance of the independent components can be seen in the fig. 6 and fig. 7

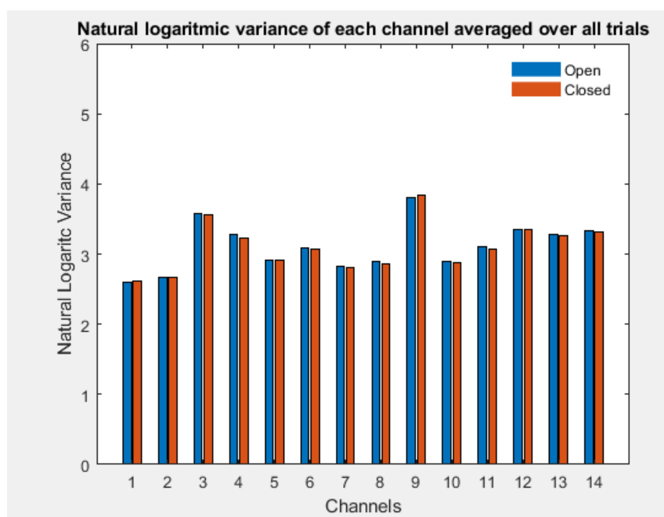

Figure 6. This is a bar-graph representing the difference between the average natural logarithmic variance of all open and closed trials per component before the ICA processing

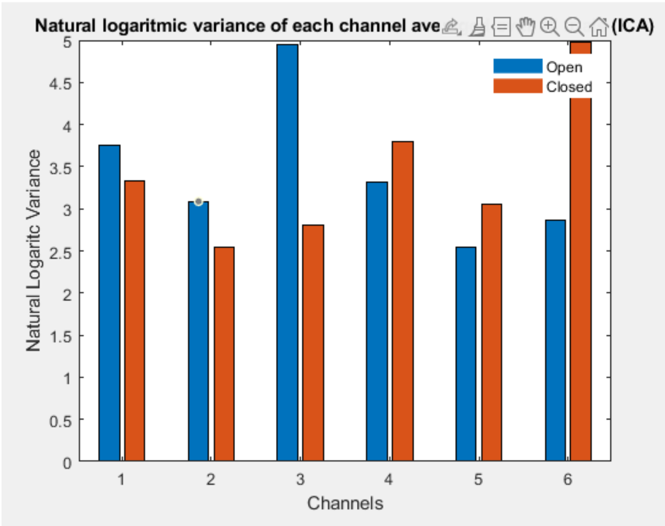

Figure 7. This is a bar-graph representing the difference between the average natural logarithmic variance of all open and closed trials per component after the ICA processing. ICA reduces the number of component for more accurate readings.

3) Classification

To classify the data accurately, an SMV model with a linear kernel was chosen as the classification model. The data was split into testing and training data. 70% of the data was used to train the SVM model and 30% was used to test the SVM model.

C. Application

1) Hardware

The Hardware provided for application of this project consisted of a roboRIO, two arms with hands and some related components, three special servos and two continuous servos. Both arms were damaged and only had allocated space for a single servo, the fishing line used to close the robotic hand was cut short, the tension springs were too long and were cut leaving the whole contraption unpractical and rather aesthetically unappealing. The provided servo motors were also unusable due to them being continuous servo motors, meaning that they could not hold an angle. Thus only the roboRIO was used and all other components were replaced. The Hitec HS-422 (**?**) servos was chosen because they were available, inexpensive, had satisfactory velocity and strength, could be operated using the 5-6V provided by the roboRIO, and uses a PWM signal as input for angle representation.

The forearm designed for this project was designed to hold 5 servo motors and their respective pulleys, one for each finger of the hand, and connect perfectly to the existing hand. To get the fit exactly right, a photo of the connection plane (underneath where the hand is connected to the forearm) hand was taken and opened in SolidWorks in order to trace the shape and scale to the correct size. Once the 3D CAD model was finished in SolidWorks, the model was 3D-printed and the robot hand was constructed. Two servos were mounted with the pulley on the right and three servos were mounted with the pulley on the left.

2) Software

Fig. 8 is the flow diagram of the algorithm used by for the program.

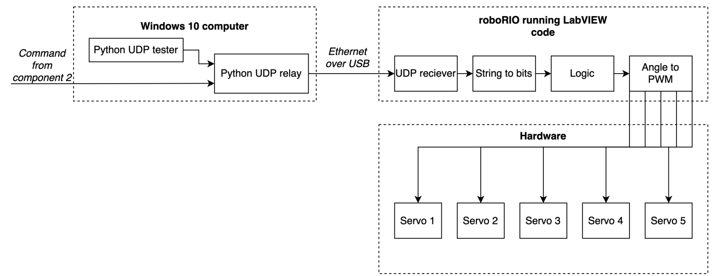

Figure 8. Flow diagram of the logical layout of component 3. Control commands are sent from component 2, or generated by a local Python script. The commands are then relayed from the Windows computer to the roboRIO using LabVIEW to do all processing in order to make the servos turn as commanded.

A python script was written to relay strings to the roboRIO via UDP IP since the roboRIO is on a separate network to the real-time processing. The python relay script transfers the strings to the roboRIO via Ethernet or USB connection. Each string is written in Windows Command Prompt to increase the ease of debugging. The python UDP relay would send a signal to the roboRIO in the form of a string encoded using UTF-8.

LabView on the roboRIO was used to receive the strings from the python relay script. Each string consisted of a 7 digit binary code. The first two characters of the string would decide the operation method.

Robot Hand Operation Method	
Characters	State of Hand
00	Hand Open/Relaxed
10	Hand Closed
01	Hand Half Open
11	Finger Control Protocol

Table II
TABLE REPRESENTING THE DIFFERENT MEANS OF OPERATION DEPENDING ON THE FIRST TWO CHARACTERS OF THE STRING SENT FROM THE PYTHON UDP RELAY TO THE ROBORIO

The finger control protocol separately activates every servo motor, pulling down every finger on the robotic hand. The last 5 digits of the string received by the roboRIO decide which finger servo will be actuated.

Individual finger movement	
1100001	Little finger servo actuated
1100010	Ring finger servo actuated
1100100	Middle finger servo actuated
1101000	Index finger servo actuated
1110000	Thumb servo actuated

Each digit in the received string represents a command. The seven-digit string is split into seven individual digits. The string was first converted into the variable type double, where the formula $\frac{X}{10^n}\%10$ is applied, where X is the input string, n is the position of the digit to extract and $\%$ represents modulus operation.

Due to the above mentioned forearm design, two servos rotated in the opposite direction to the other three. The servos need a PWM signal with varying duty cycle to rotate the desired angles stated in the program. A sub VI converted the angles into duty cycles and generated the 50 Hz PWM signals for each finger.

IV. Results

A. Acquisition

As can be seen in fig. 9 the amount of data points lost using UDP is not enough to be used. The capture rate of packets was 25%.

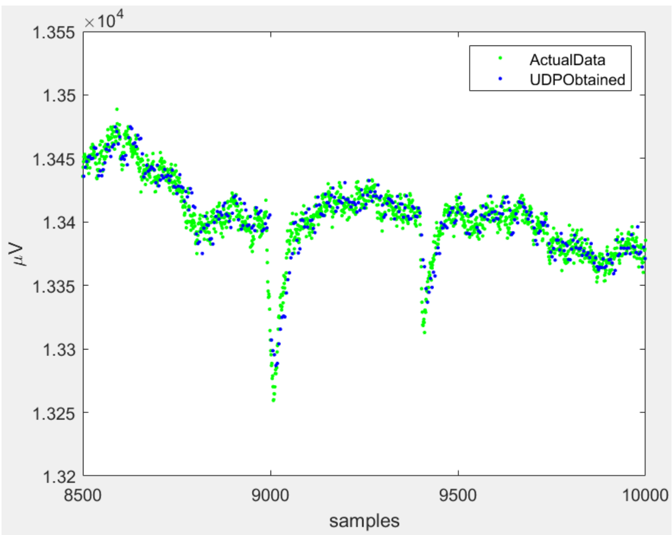

Figure 9. Results produced by using the UDP protocol. The green dots represent the data collected and saved by the OpenBci GUI. Blue dots represent the data collected in MATLAB using the UDP protocol.

The use of the LSL protocol yielded a much higher capture rate than the UDP protocol. This can be seen in fig. 10

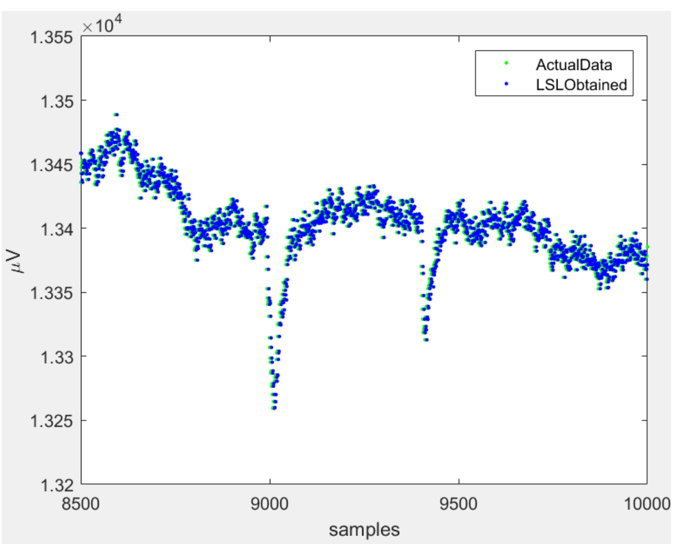

Figure 10. Results produced by using the LSL protocol. The green dots represent the data collected and saved by the OpenBci GUI. Blue dots represent the data collected in MATLAB using the LSL protocol.

B. Real-Time Processing

After processing the data with the use of the ICA feature extraction Method, the SVM model was able to predict the desired outcome with approximately 83% accuracy. This is an approximate 23% increase from the results produced using the CSP feature extraction which only yielded an approximate 60% accuracy (21).

1) Application

The forearm needed to be wider than a human forearm which made the robot hand look very unrealistic. The final constructed product can be seen in fig. 11

Figure 11.

V. Discussion

A. Acquisition

A lot of time was spent on trivial tasks, such as getting the EEG Helmet to connect to any laptop that was available, at the beginning of the project. The rare times that a stable signal was established between the EEG Helmet and a laptop, 20 minutes had already passed.

Another large problem that needed tackling was the data streaming. UDP was attempted at first but many more problems arose. The UDP buffer held onto 5 packets of data and repeatedly sent them to Matlab instead of streaming the rest of the data. This problem was fixed by flushing the UDP buffer. This led to another issue. Every time the buffer got flushed, 20 data packets would be lost.

Due to the small amount of data getting sent through, a different approach had to be found. This led to the LSL protocol for streaming. LSL lost 3 data packages in the beginning of a 2 minute recording which had 15000 data packets in total.

One problem with the hardware was the brittleness of the cables inside the EEG electrodes. It was very easy to break and once broken, the electrode needed to be replaced. Subjects stated that the headset was also very uncomfortable to wear.

An improvement for this project could be to record the separate movements of each finger so that the hand could be made to do more complex tasks than holding an object.

B. Real-Time Processing

It was important for the acquisition to get data from using the same hand movement (left or right) because the different hands light up different sides of the brain. The left hand causes neurons to fire on the right side of the brain and the left hand causes neurons to fire on the right side of the brain.

For the feature extraction CSP was not accurate enough to produce results a lot more accurate than a coin toss (50%). ICA was then used. The choice of ICA algorithm needed to be chosen. There was the possibility writing a function using the algorithm produced by Hyvärinen (**?**), or to download a matlab function that applied this algorithm. These were both attempted and with no positive results. If the algorithm could be programmed to work, it would help with the total understanding of ICA and could be tweaked to improve the prediction accuracy.

When spiltting the data, a 70/30 split was used. This means that 70% of the data was used for training and 30% of the data was used for testing. Spliting the data to favor the training even more could result in a higher prediction accuracy.

C. Application

The idea for the use of the relay was not planned for but was thought up rather quickly when the roboRIO failed to connect to LabVIEW socket through the computer. One of the people working on the project had experience with using sockets in Python. They made use of this knowledge and experience to quickly come up with a solution to the failed roboRIO labview connection.

Lots of time was wasted trying to connect the roboRIO to LabVIEW only to discover that LabVIEW versions other than the 2016 version are uncompatible with the roboRIO interfaces.

There were a few minor issues with regards to streaming the sting to the roborio. These problems were caused by inexperience but were easily resolved.

Separate motors have been added to control each finger to make the improvement easier for the next people who take on this project.

VI. Conclusion

A. Acquisition

The use of LSL protocol for streaming the data from the EEG headset to the openBCI program on the laptop worked very well and yielded satisfactory results. The system validation was done by comparing the online data with the recorded offline data to visualise the no-to-minimal package losses. The LSL protocol only went through minimal to negligible packet loss. This was in accordance with the hypothesis stated above.

B. Real-Time Processing

The prepossessing of the data had uncorrelated the data sufficiently and prepared the data well for the feature extraction. The ICA used as feature extraction separated the data from open/relaxed data and the closed data. This helped the SVM model predict the state the robotic hand should be with satisfactory accuracy. Ideally the accuracy would be 100% for anyone using a prosthetic arm, but it was not be achieved in this project. The accuracy was at roughly 80%, as predicted in the hypothesis stated above.

C. Application

The hand was designed in a way to provide separate control for each finger and the program was written in a way to activate all servos together to simulate a closed hand. The program was also written to test that each servo motor could operate separately. The Program written in LabVIEW could open / close the robotic hand on each accurate command recieved from the matlab script written for the real-time processing. This is in accordence with the hypothesis stated above.

Acknowledgment

The author would like to thank Mälardalen University for the access given to use its resources.

References

[1] A. Bonci, S. Fiori, H. Higashi, T. Tanaka, and F. Verdini, “An introductory tutorial on brain–computer interfaces and their applications,” *Electronics*, vol. 10, no. 5, p. 560, 2021.

[2] “Ecog performance status - ecog-acrin,” ECOG-ACRIN, 2012. [Online]. Available: https://ecog-acrin.org/resources/ecog-performance-status

[3] M. L. Homer, A. V. Nurmikko, J. P. Donoghue, and L. R. Hochberg, “Sensors and decoding for intracortical brain computer interfaces,” *Annual Review of Biomedical Engineering*, vol. 15, no. 1, p. 383–405, 2013.

[4] R. a. ACR, “Magnetoencephalography,” Jul 2021. [Online]. Available: https://www.radiologyinfo.org/en/info/meg

[5] D. Madhuri and P. C. Reddy, “Performance comparison of tcp, udp and sctp in a wired network,” in *2016 International Conference on Communication and Electronics Systems (ICCES)*, 2016, pp. 1–6.

[6] “What is lsl?” accessed: 05-03-2022. [Online]. Available: https://labstreaminglayer.readthedocs.io/info/intro.html

[7] S. Sanei and J.A.Chambers, *EEG signal processing*. John Wiley amp; Sons, 2011.
[8] M. X. Cohen, *Analyzing neural time series data: theory and practice*. MIT press, 2014.
[9] “Ica for dummies,” Jul 2020. [Online]. Available: https://arnauddelorme.com/ica_for_dummies/
[10] Z. J. Koles, M. S. Lazar, and S. Z. Zhou, “Spatial patterns underlying population differences in the background eeg,” *Brain Topography*, vol. 2, no. 4, p. 275–284, 1990.
[11] V. VAPNIK and C. CORTES, “Support-vector networks - image.diku.dk.” [Online]. Available: http://image.diku.dk/imagecanon/material/cortes_vapnik95.pdf
[12] N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *The American Statistician*, vol. 46, no. 3, p. 175, 1992.
[13] S.-W. Park, J.-H. Bae, J.-H. Park, and M.-H. Baeg, “Development of an anthropomorphic robot hand aimed at practical use for wide service robot application,” in *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, 2012, pp. 431–435.
[14] “Parallax Standard Servo (#900-00005),” 2017, accessed: 04-03-2022. [Online]. Available: https://docs.rs-online.com/0e85/0900766b8123f8d7.pdf
[15] “Hitec HS422 Servo,” accessed: 05-03-2022. [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SER0002_Web.pdf
[16] “NI roboRIO User Manual (FRC) - National Instruments,” 2014, accessed: 05-03-2022. [Online]. Available: https://www.ni.com/pdf/manuals/374474a.pdf
[17] “Labview roborio software suite download.” [Online]. Available: https://www.ni.com/sv-se/support/downloads/software-products/download.labview-roborio-software-suite.html#326406
[18] “Ultracortex "mark iv" eeg headset,” accessed: 05-03-2022. [Online]. Available: https://docs.openbci.com/AddOns/Headwear/MarkIV/
[19] “Dry eeg comb electrodes,” accessed: 05-03-2022. [Online]. Available: https://shop.openbci.com/collections/frontpage/products/5-mm-spike-electrode-pack-of-30?variant=8120433606670
[20] A. Morley and L. Hill, “10-20 system eeg placement,” accessed: 06-03-2022. [Online]. Available: https://www.ers-education.org/lrmedia/2016/pdf/298830.pdf
[21] J. N. N. K. S. I. H. S. B. J. H. P. G. J. F. F. L. W. O. L. V. A. H. T. Emanuel Bjurhager, Simon Friberg, “Controlling a robotic hand using a real-time neurorobotic system,” p. 18, 2022.