ROBOTS OR 3D MODELS CONTROL BY BRAIN-COMPUTER INTERFACES

By

TOMOHIRO UCHIMURA
Master of Science (Computer Science)

Submitted to the College of Science and Engineering in partial fulfilment of the requirements for the degree of Master of Science (Computer Science) at Flinders University – Adelaide Australia.

6th December 2021

DECLARATION OF ORIGINALITY

I certify that this work does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Signature: [redacted]

Date: 15 / 10 / 2021

KEYWORD

Brain Computer Interface (BCI), Control system, Conversion Table, Jaw Clenching, Lab Streaming Layer (LSL), Motor Imagery, OpenBCI, OpenViBE, Robot, Sphero RVR, TCP

ABSTRACT

The brain emits electrical signals which is captured to produce electroencephalography (EEG) as representing human activity. These signals have been researched for connecting the brain and computer or machines. The brain-computer interface can collect EEG signals and send them to the computer. The technology is pretty helpful for disabled people to control products without their body movement. Moreover, BCI technology can expand the possibilities of the development of new products and entertainment.

Although many of these BCI devices and biosignal devices support the Lab streaming Layer as a reliable transfer protocol, the protocol is uncommon. Hence, implementing a brain control system on the products requires indepth knowledge of BCI technology and Lab streaming Layer. The project aims to develop convenient and flexible software (it is called the flexible software in the thesis) for connecting BCI devices and other products such as robots or 3D model control systems. The developed software can convert stimulations received from EEG devices into simple string commands and send these commands to control targets via standard transfer protocol, TCP. Combining the software and BCI devices and biosignal devices might reduce the barriers to entry for implementing brain control systems; it leads to boost technologys' evolution in lots of areas.

The system used OpenBCI hardware and software and OpenViBE software for collecting and analysing EEG signals. EEG headband and Ganglion were used as EEG equipment, and Sphero RVR Programmable Robot was chosen for a control target. As a demonstration of bio-signal collection and control, jaw clenching was used as a moving trigger to control Sphero RVR; when relaxing, the Sphero RVR stopped. The programming language was Python 3.5 was used to develop the flexible software. The system was evaluated from the point of view of accuracy and processing speed with two testers. Because having high accuracy and fast processing speed is essential for controlling system to avoid users' confusion, miss control, and accident. The conversion accuracy of jaw clenching marked 100% recognition; however, several relaxing states were missed. The processing speed of conversion and sending commands was pretty high (under 0.150 seconds). There were several limitations on the testing environments and some system delay problems; however, they can be resolved in future work. Hence, the software sufficiently meets the system requirements of the mediation software connecting BCI devices and other products. The project's success will help many manufacturers' brain control system implementation in lots of industries in the world.

ACKNOWLEDGEMENTS

I wish to acknowledge my supervisor Dr Trent Lewis for the continued support, giving suitable advice, and leading on the thesis project. Moreover, I would like to thank Tomoe Uchimura for her contribution effort for the whole testing as a tester. Additionally, I would like to appreciate to my family, friends and school officials for highly supporting.

Table of Contents

1. INTRODUCTION

Human activities occur from the brain, not only physical activities but mental activities such as thought, imagery, or emotions, and they could be detected as an electroencephalogram (EEG). Analysing EEG started a few decades ago for research in neuroscience (Jatoi, 2017). The attempt has been used for extending possibilities of technologies in many aspects, medical, gaming, entertainment.

A brain-computer interface (BCI) can collect brain signals and transport them to the computer. EEG can be categorised into six rhythms, delta wave (1-4 Hz), theta wave (4-7 Hz), alpha wave (8-12 Hz), mu wave (8-13 Hz), beta wave (12-30 Hz), and gamma wave (25-100 Hz), EEG transfer status into these rhythms by changing emotions, taking activities, or imagery. A BCI system can detect the feature of the EEG as stimulations from the pattern of the wave (Amiri, Rabbi, Azinfar, & Fazel-Rezai, 2013). There are three major types of BCI method, P300, Steady-State Visual Evoked Potential (SSVEP), and Motor Imagery (MI). P300 is a BCI method that is based on Event-related potentials (ERPs); ERPs occur from specific cognitive, emotional or activity events as a reaction of the brain (Amiri, Rabbi, Azinfar, & Fazel-Rezai, 2013). Usually, the method uses a display showing the alphabet, specific words or images. These symbols are flashing row by row or column by column; when patients focus on one particular character, phrase or image, ERPs will arise at the back of patients' heads. The method has been used for the virtual keyboard or controlling a wheelchair to help disabled people (Amiri, Rabbi, Azinfar, & Fazel-Rezai, 2013). SSVEP is a visual cortical response caused by constant frequency stimulations; when patients focus on the specific object which is frequently flashing, the same frequency signals as the visual stimulations occur at the back of patients' heads (Amiri, Rabbi, Azinfar, & Fazel-Rezai, 2013). Thus, BCI systems using SSVEP have a display or LED for showing frequent flashing objects to trigger SSVEP. Although MI is a complex BCI method compared with P300 and SSVEP, it has vast possibilities for many products. The MI method can detect stimulations of imaging motor movement without actual movement using machine learning training (Cho, Ahn, Kwon and Jun, 2018). Hence, MI requires collecting some sample data of brain waves before using the brain-control system. Though training for MI might bother patients, the MI method does not require any additional devices such as a display or LED light to control products. The characteristic of the BCI method is not needed movement to detect stimulations; especially, it is pretty helpful for disabled people. For example, a patient with lower-body paralysis because of spine injury could kick the ball on his foot using a robotic suit with the brain-machine interface (BMI) that can connect his brain and machines (Sharif and Ali, 2020).

Additionally, some researchers try to detect patients' emotions using the brain-computer interfaces. Usually, emotion detection uses facial expression; however, the method might not be used for disabled people; hence, brain signal analysis is a valuable approach to detecting emotion (López-Hernández et al., 2019).

Twenty per cent of patients with complete paralysis might turn into incomplete tetraplegia in the first year. However, other eighty per cent of people’s activities will be pretty limited or absent (Burns et al., 2012).

However, there is a case of regaining lost body activities by BCI technology; the man with a spine injury and tetraplegia could grasp hand by muscle stimulation occur from BCI (Bockbrader et al., 2019). Thus, EEG analysis helps disabled people regain their physical abilities and gives hope to them, and the only way to recover a patient's activities might be BCI technology.

According to World Health Organization (2020), over 1 million people has some disability, and almost everyone has a risk of getting some disability (permanently or not). Therefore, BCI technology is a great approach to help humanity in the world.

In the other areas, there are lots of exciting research that using BCI technology. For instance, BCI might represent the unconscious creativities of artists by analysing brain signals; hence, BCI might be a powerful tool for drawing art (Folgieri et al., 2014). Moreover, According to Aricò et al. (2018), the gaming area, driving, and aviation field have been trying to use BCI as a controller or simulation system. In addition, remote working demand has been increasing because of COVID-19 compared past year (McKinsey & Company, de Smet, Langstaff, & Ravid, 2021). Combining the BCI technology or motion capture system and the software might boost remote working technologies' evolution.

However, implementing BCI into products or systems is complex and need deep knowledge, and also requires a lot of developing costs. According to Shih et al. (2012), the initial costs of the BCI system is $5,000 to $10,000. And the cost of high-end models of the BCI system for medical will be 10 times easily.

The project aims to evaluate, develop, and recommend software to connect BCI to other products, such as robots. The software is flexible, easy to change settings, supporting standard interfaces to communicate BCI and other products. Therefore, it can reduce implementation costs and ease barriers to develop brain control products. Hence, the project is helpful in many areas that want to implement brain signal control systems into their products, and it leads to relief humanity worldwide.

The system has three subsystems, and they were developed using Python 3.5. The whole system was tested using OpenBCI EEG headband (OpenBCI, 2021) and Sphero RVR Programmable Robot (Sphero, 2021) as physical devices to evaluate the accuracy and processing speed.

The thesis starts with a discussion of criteria for deciding eligible BCI methods to control robots with the relevant research papers; subsequently, the focus point moves into the potentiality of the brain control system. After that, describe the details of the developed software and evaluation method in section 3. As a result of the evaluation, the system proved to have enough accuracy and high-speed processing to convert a stimulation code to command and send data via TCP. That means that the system gives a lot of benefits to many areas. However, there was a transfer delay problem and several resolvable limitations on the testing environment and software function. Finally, through the project's progress, many essential technologies were found; these methods will be adopted in the future project, and they will enhance the project's comfortability.

2. LITERATURE REVIEW

In the implementation of the BCI system, developers should consider the BCI method first. Amiri, Rabbi, Azinfar and Fazel-Rezai (2013) says hybrid BCI (P300 and SSVEP) effectively enhances the BCI system's accuracy, reliability, and user acceptability. They implemented hybrid BCI as commands and a switch for controlling the virtual smart home. The system showed two BCI methods into a display; a background image flashed under 18 Hz to detect SSVEP, and a commands matrix was used to detect P300. As a result, the system provided an average 88.15% of detection rate and 94.44% classification rate. Additionally, they claimed that P300 is eligible to control discrete command systems, and SSVEP is suitable for sustainable command.

The paper gave an idea of using hybrid BCI in my project. However, control targets of the project are robots or 3D models; 3D software might support P300 or SSVEP; on the other hand, robots do not have a display usually; thus, hybrid BCI (P300 and SSVEP) is not eligible for my project. However, another combination of BCI might be useful to enhance the accuracy and reliability of the system.

On the other hand, Wang et al. (2018) researched a suitable method for controlling robots. They developed a hybrid BCI system that combines SSVEP and MI to playing games. They mentioned that the system could be used for high-performance BCI games with enough accuracy. The project was tested by playing Tetris using MI for supporting a left-direction moving command and a right-direction moving command. The trained trigger of the MI is grabbing a ball by the right hand softly, grabbing a ball by the left hand lightly, and squeezing the towel by both hands, after that choosing two of them because of supporting individual differences. SSVEP was used to rotate bricks. They displayed a flashing brick at the right side to occur SSVEP on users' brains. The testing results were good; the classification accuracy of MI was over 80%, and it provides an average of 90.26% of the classification accuracy of SSVEP. Moreover, they claim that they successfully processed SSVEP commands and moved bricks by MI simultaneously by setting priority. The hybrid BCI (SSVEP and MI) is a remarkable idea to control robots or 3D models because these products require many triggers for controlling each action. For instance, car type robots support a lot of commands such as run, back, turn left, turn right, stop, and also changing speed. However, implementing all commands is difficult only using MI. Hence, considering combining MI and another BCI method is essential to control products.

When using MI requires an effective training method to reduce the load of patients. Alimardani (2018) discussed the effects of visual feedback for MI training. He said that MI training times are reduced by realistic visual feedback using robot arms; most motor imaging training is against their intuition; however, their approach is intuitional. Also, sometimes the training might fail because of a lack of patient's motivation cause by a long time of training. They developed an MI feedback system using a VR device that shows virtual hands on the sight. When the tester imagines grabbing the left hand, the left virtual hand will grab based on detected stimulation by MI.

Furthermore, they compared two types of 3D hand models, metallic virtual hand and natural virtual hand. As a result, the hand model close to humans provides high performance than metallic hands. Their research tells me the importance of considering training methods for MI; however, it requires high implementation costs and time to develop a virtual feedback system. Thus, my project did not adopt it currently, while future work should adopt their technique on the MI training.

Wang and Bezerianos proposed a wheelchair control system using brief MI in 2017. Their system adopted sustained MI as forwarding and back movement commands; it also adopted short MI as turn left command, turn right command, and stop command. They mentioned that imagining both hands movements is a surest stimulate for adopting as stop command without misclassification. The results of the system evaluation, four testers could control the wheelchair without any crashes. The paper's work is in a similar direction as my project, and using sustained MI as a command of going forward and back is an intuitive and easy-to-understand way. However, the system has low scalability; thus, it cannot easily be applied to other products.

Subsequently, finding eligible electrode positioning is vital to get good results of BCI methods. Where stimulations features appear depends on the activities. According to Jurcak et al. (2007), an international 10/20 system has been used as a standard of electrode positioning. The system provides reproducible EEG electrode positions for research. Researchers can discuss electrode positions with common perception by using the standard. Also, there are other standards, the 10/10 system and 10/5 system (Oostenveld and Praamstra, 2001), to extend electrode positioning to the high density.

Boord et al. (2010) provide a way to detect left and right leg movement stimulations as MI method. They claimed that moving the left and the right leg produces a clear alpha wave pattern; the pattern can be classified into several BCI commands by movement differences. They tested with several electrode positioning pairs based on the international 10/20 system to find the best positioning for the left and right leg movement detection. They claimed that the CCPB and CCPF were eligible to detect leg stimulations, and the C3 and C4 were suitable for hand stimulations detection. Moreover, they mentioned that detecting stimulations timing is different caused by conditions or individuals; they implemented an automatically adjust function in the system. The paper gave my project an idea of finding electrode positioning for the MI method and implementing an automatic adjustment function for supporting individual differences as a future goal.

As more recent research, Batula et al. (2017) provided MI method for detecting both sides of hands and both sides of legs separately. They used functional near-infrared spectroscopy (fNIRS) as a BCI device. The fNIRS is an optical brain imaging technology, and it can monitor HbO and HbR levels in the brain. In addition, they claimed that the control system with four stimulation classifications could be enough implemented. The problem of the MI method was a limitation of the detectable stimulations in one system. Several stimulations have a similar EEG pattern as others. For example, left-hand and left-leg movements might be misrecognised (Chae et al., 2012). Hence, an increasing number of simultaneous detectable stimulation on MI boosts the number of supportable products. They trained the system with 50 times moving tasks and 150 times imaging tasks for every five testers. As a testing result, all testers marked over 30% of the accuracy for classifications. It was not an excellent result, while they claimed that the system increased the average of 5.21 accuracy compared with past MI methods. Finally, they concluded that their method might be improved by applying feedback and implementing the neural network as a classification method. Although the article gave another idea of using other types of BCI devices for my project, the fund was limited; thus, the OpenBCI headband was used for the system implementation. Moreover, multiple stimulation detecting is a pretty good approach to controlling products; therefore, future work should consider detecting multiple statuses.

As a trigger of the character moving for the game, Podoprikhin (2015) used eye blinking detected by using EEG. He claimed that eye blinking could be categorised into three types, reflexive, intentional, voluntary; although reflexive eye blinking is regarded as a noise of EEG data, intentional eye blinking could be used as a stimulation. He used machine learning approaches such as Support Vector Machine (SVM) and Random Forest to classify the feature of intentional eye blinking. The results of the classification, the blinks of the left and right eyes were classified with 97% accuracy by Random Forest. However, he said that the system had to do additional short training with each tester to get high accuracy. His approach highly matched with my project, though the method required a long time for classification and many samples of EEG signals. Thus, my project decided to select another method to detecting stimulation.

Costa et al. (2014) proposed a robot arm controlling system using electromyographical signals from jaw clenching as another BCI approach. They claimed that this approach is an easy way to detect stimulations than other BCI methods; furthermore, training time for the jaw clenching is faster than MI and the system's accuracy is high. The system can detect four stimulations, hard clenching right jaw, soft clenching right jaw, relaxing, hard clenching left jaw, and soft clenching left jaw. These features are categorised by threshold; they are used as triggers for deciding moving directions and powers. In addition, they claimed that several artifacts such as blinking eyes and signals that occur from muscles could be used as stimulations. As a testing result, the system is 24 times faster than MI in moving the cursor to a target point. My project adopts the jaw clenching method as a testing trigger because the main point of my project is to develop software that connects BCI and other products, which means my project does not need complex BCI methods. The difference between their system and my project is that they focus on developing the jaw clenching method.

Another paper focuses on the classification method. Ramakuri et al. (2015) researched OpenViBE software that is easy to connect to BCI software and easy to design analysis scenarios. They provided a simple way to detect a relaxing stimulation using several spectrum analyses provided from OpenViBE software as scenario components. My system also adopts OpenViBE software as a classification software to detect jaw clenching stimulation. The paper gave my project a basic knowledge of OpenViBE. However, it only provided a simple way to detect features of EEG; thus, my project implemented a more complex scenario for controlling robots.

There is another similar article as the project of Ramakuri et al. Markovinović et al. (2020) also mentioned an eye blinking stimulation remove system from EEG signals. Eye blinking is an unconscious human activity;

thus, removing these signals is helpful to detect other stimulation from EEG signals. The paper focused on an algorithm to classify the eye blinking artifact from EEG signals. They use the international 10/10 system as a standard to decide electrode positioning, and the system uses 16 positions to collect EEG signals. They claimed that the ADJUST algorithm could remove eye blinking signals from EEG signals; it is helpful to getting clean EEG data. The paper's approach might help the future work of my project because getting clean EEG data is essential to detect stimulations with high accuracy.

Many articles mentioned controlling robots through BCI devices. Chae et al. (2012) developed the controlling system for a humanoid type robot using BCI. The control target of the system was the humanoid robot NAO (SoftBank Robotics, 2018), and moving triggers were left hand, right hand, and leg movement. These stimulations were detected by the MI and classified by hierarchical classifiers. The main system communicated with NAO via TCP, and they monitored testers' stimulations every 250 milliseconds. Detected stimulations were sent to NAO from the main system, and visual feedback was returned back to the main system from the front camera of NAO. As a result of their project, they successfully controlled three actions of NAO, go forward, turning left and turning right. However, they claimed that there were several misclassifications between the same side of hands and legs. Moreover, they also mentioned that the delay of the feedback affected controlling accuracy. Their system was separated into three subsystems, BCI system, interface, and control system, and it is similar to my project well. On the other hand, their system was adjusted to detect specific stimulations, and supporting other BCI methods required software improvement. The control target NAO might be controlled using my software as well.

Stawicki et al. (2016) developed the mobile robotic car controlling system using SSVEP as a BCI method. They claimed that the system was providing a high recognition accuracy, and it could provide the user with a comfortable system that gave only a small load to the user. Control target was created by themselves and connected to a controlling system via UDP. Also, they created a control system with a graphical user interface. The user could control the directions of the robotic car and showing any camera view from five positions. Moreover, they developed a calibration system for SSVEP to collect samples for system training. The testing was done with 61 testers, and testers were instructed to control the robotic car following a specific route. As a result, their system provided 97.14 % of mean accuracy. They claimed that using SSVEP to detect four commands was a robust system to control the robotic car. However, they also mentioned that sometimes the tester focused on camera view feedback rather than command buttons, which temporarily lost control. Although their system was excellent in controlling robots with a display, the method is not always available because they also mentioned that several testers required sight correction to testing. Their evaluation method adopted users' comfortability. My current project had only two testers; however, future work should adopt comfortability as an evaluation point to enhance the system useability. Moreover, they said that the system has a limitation on transfer reliability because of using UDP. The control system should be reliable to avoid accidents; thus, they should choose TCP as a transfer protocol as the same as my project.

There is also another proposal of the BCI system, which supports four commands. Bousseta et al. (2018) proposed a robot arm controlling system with the MI method. They detected the left and right-hand movement, both-hands movement and leg movement as stimulations using an SVM with data from a combination of the Fast Fourier transform (FFT) and the Principal Component Analysis (PCA). They used an EMOTIV EPOC headset (EMOTIVE, 2021) as EEG equipment, and a control target was ARM 2.0 (GearWurx, n.d.). They tested the system with four testers; the tester took training for 5 minutes; after that, they were instructed to tried to move the robot in four directions ten times each. As a result, their system provided over the average 85% of the accuracy of stimulation detection. The accuracy of detecting features of human activity depends strongly on the pre-processing of EEG signals and classifier choice. Thus, their approach was important to find a better combination of analysis methods for MI engineers. As future work, providing powerful pre-processing function collection as a part of the software might enhance the value of my project. There is the latest research of a pre-processing method to reduce the complexity of EEG signals and increase the accuracy of the feature detection. In 2020, Baig et al. proposed the filtering technology for channel selection for EEG signals. They claimed that the eligible channel selection could reduce 80% of the processing channel's number and boost the BCI system's performance because EEG signals contain several unsuitable channels with noise. They said that there are several channel selection approaches, filter method, wrapper method, and hybrid method; the filter method is based on statistical properties and have good computational efficiency. On the other hand, the wrapper method depends on the classifier; thus, it requires more calculation than the filter method. However, the wrapper method might provide high classification accuracy. The hybrid method is flexible and generalised because of having both characteristics of the filter method and the wrapper method. They concluded that analysing a huge EEG data set needs a lot of processing time; hence, reducing the EEG data is essential; thus, the pre-processing technique is helpful to efficient feature detection. Hence, their proposal is remarkable for MI engineers, and also, brain control system developers should know channel selection methods.

Chen et al. (2017) suggested using the Fuzzy Feature Threshold Algorithm (FFTA) to detect suitable thresholds for each stimulation. The approach could adjust thresholds for automatically detecting stimulations from EEG signals. Firstly, they filtered EEG signals to removing unconscious eye blinking noise. After that, the signals were transformed by FFT; then, these signals were used as input data of FFTA. They showed the shape of transferred EEG signals in the figure, and it represented that the success of detecting stimulations well. They tested the system with a car type robot and fifteen testers. The result of their project marked the average of 88.02% of the accuracy for robot control by SSVEP. Their FFTA technique was pretty notable for my system. Using Jaw clenching as a moving trigger requires finding specific criteria that depend on the person; adjusting thresholds was a big issue to the project's success. However, the current project could not implement their method; hence, it should be adopted in future the project. Furthermore, there is another approach for detecting features from EEG efficiently.

Moreover, Xu et al. (2015) suggested combining the discrete wavelet transform and autoregressive method to detect stimulations. The classifier was linear discriminant, and the control target was arm type robot. The EEG equipment was connected to a computer by USB cable, and the arm type robot was connected to the computer via a robot control board. Received EEG signals were analysed on the computer and transformed by the wavelet transform. They claimed that the wavelet transform method provided good results than FFT and autoregressive model to separate signals. The evaluation was done with six testers, and they claimed that the result was pretty high accuracy (90% recognition rate). Moreover, they mentioned that the system's processing time for hand motor imagery recognition was only 7.4 milliseconds. Their results were excellent and very fast; hence, the MI method system should consider adopting their suggestion.

In addition, there is the latest comparison research for filtering methods for MI. In 2020, Miladinović et al. published a paper that compared three BCI filtering technology, Source Power Co-Modulation (SPoC), Spectrally Weighted Common Spatial Patterns (SpecCSP), and Filter-Bank Common Spatial Patterns (FBCSP). They mentioned that the SPoC has a high resistance to noise while tend to misrecognise the muscle contraction as a stimulation. The SpecCSP is an expansion of the SPoC, and it provides good results than SPoC. Their testing targets five stroke patients and is retried 15 times each. The FBCSP can resolve the weakness of the SPoC by frequency filtering. As a result of the comparison, the FBCSP marked the best score (mean average *85.1%). On the other hand, the result of the* SPoC and SpecCSP was almost the same value. They conclude that the FBCSP method is a good approach to helping brain control system using MI for stroke people. In this way, there are many filtering methods, classification methods and data transfer techniques; thus, researching these approaches is essential to implementing the MI method on the brain control system. That means the system needs depth knowledge of brain science to developers, and they might have to do many types of work. Thus, reducing the cost of developing connecting software reduces work amount and collection time for knowledge.

Subsequently, move on to the point of view of possibilities on the brain control system. From the research by Holz et al. (2015), they claimed that brain control systems have good effects on disabled people. They researched a case of a 73 years old man who could only move his eye. They gave him the drawing application using the P300 method for controlling the system. Then, they observed BCI usage for 14 months; these data were evaluated from the point of view of effectiveness, efficiency, satisfaction. As a result of the evaluation, their drawing system positively affected his happiness, pride, quality of life, confidence, and more. Finally, the man finished drawing excellent arts using BCI. Hence, remote control supporting systems that use BCI technology are remarkable to increase the quality of disabled people's lives and give many opportunities to challenge new activities.

According to Kothe et al. (2019), the LSL is used not only in BCI devices but in many types of biosignal devices such as motion capture systems, eye-tracking systems, and more. Therefore, investigating the usefulness of these biosignal devices might give an idea for the effective utilisation of my software. Black et al. (2017) researched facial emotion recognition methods on people who have autism spectrum disorders (ASD) by using a combination of EEG signals and eye-tracking technology. Recognise facial emotion is important for living in human society; however, ASD people might be difficult to recognise emotion well.

Combined EEG analysis and eye-tracking can detect their gaze behaviour and brain condition during facial recognition. Although they researched many past pieces of research about facial emotion detection by EEG and eye-Tracking, they claim that they could not find relevance between facial emotion recognition and EEG signals and between facial emotion recognition and eye-tracking. However, they also claimed that the integration approach of EEG analysis and eye-tracking did not investigate well. If the facial motion recognition biomarker is found, the method can help correct ASD people's facial recognition by giving feedback. The eye-tracking device and EEG equipment they used could be replaced with LSL supporting devices; if they do, their system will be suitable to implement my software. Therefore, the research gives my project the possibility that the software will be used for helping ASD persons in the future.

According to Aristidou and Lasenby (2010), the demand for hand motion capture systems has increased in many areas, such as gaming or developing 3D models. They mentioned that the cost of the globe type capture system is high, and it has a limitation on moving recognition and a low accuracy problem. On the other hand, they claimed that the developed hand motion capture system, which used joint angles with Inverse Kinematics (IK) to estimate the hand motion, was inexpensive and had enough recognition accuracy. Several marker-based capture systems support LSL; hence, my project also helps motion capture systems development. However, current software only analysis the first data field of one sample; therefore, the flexible software should slightly expand for supporting marker-based capture systems as future work.

Furthermore, a remote maintenance and assembly machine for the ITER vacuum vessel (VV) using TCP command control was constructed (Li et al., 2013). the ITER is a huge fusion device for observing high-temperature fusion reactions; the ITER requires safety and a reliable remote maintenance machine for executing daily maintenance and repairing (Friconneau et al., 2017). According to the paper of Li et al. (2013), their maintenance machine converted GUI operations into command data and sent them to the server system that was implemented on the maintenance machine. The design concept of the software was similar to my project in the point of view of converting UI operations into commands and send them via TCP. Moreover, the system was tested from the point of view of system delay as the same as my project; the acceptable system delay was almost the same as the result of the delay measurement for my project. According to their software architecture, although their project was not modularised, the user interfaces might be able to be replaced with BCI or other biosignal devices. Hence, my project also might help to provide a reliable conversion system and the possibility of implementing brain signal control to the ITER sector.

Moreover, BCI technology is used not only for control systems but used for observing user statuses. According to Dhole et al. (2019), the BCI technology was used for detecting workers' drowsiness to avoid work accidents. Their project suggested implementing BCI on a safety helmet; the device kept monitoring workers' brain signals during working. They claimed that the drowsiness detection using EEG gives us high accuracy than other methods such as facial recognition or heartbeat. The classification method was the random forest algorithm. The results of the evaluation, their system provided 98% of accuracy to detect workers' drowsiness. Their project was impressive to alert the sign of overwork; it leads to making safe working environments.

Their work used several sensors to improve detecting accuracy. My current work could not support getting combination data, while implementing a function for supporting combination data might spread the usefulness of my work.

3. METHODOLOGY

As shown in Fig. 1, the system had three subsystems, a BCI analysis system, flexible software, and a control products system. These systems were written by Python 3.5 and tested on Surface Laptop 3.

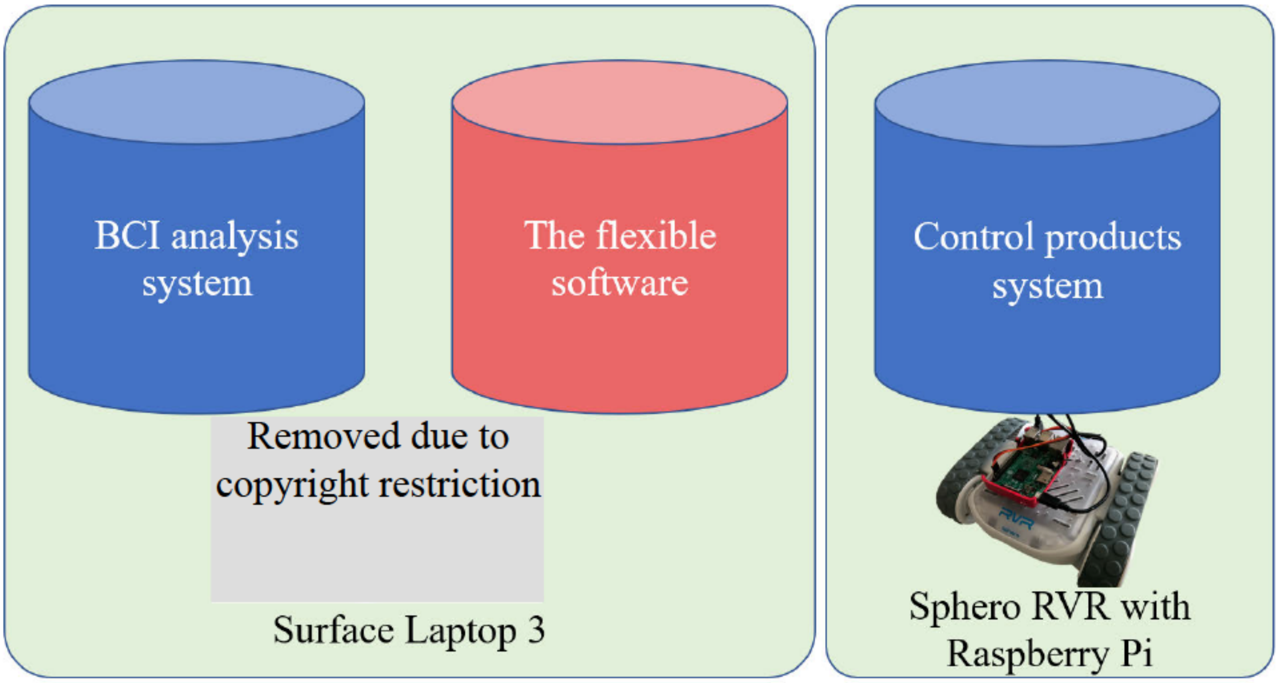

Fig. 1. The software system diagram on the project

The BCI analysis system included EEG equipment, OpenBCI software, and OpenViBE software, the system connection diagram is shown in Fig. 2. The main focus point of the project was the flexible software that contained Lab Streaming Layer (LSL) receiver, a data analysis and conversion function, and a TCP client function, as shown in Fig. 3. The control products system consisted of a TCP server function and a calling outside software function; the system diagram is shown in Fig. 4. The control target was Sphero RVR Programmable Robot, a car type robot that supported Raspberry Pi 4 (Raspberry Pi Foundation, 2021) as an expansion board. They are connected as shown in Fig. 5.

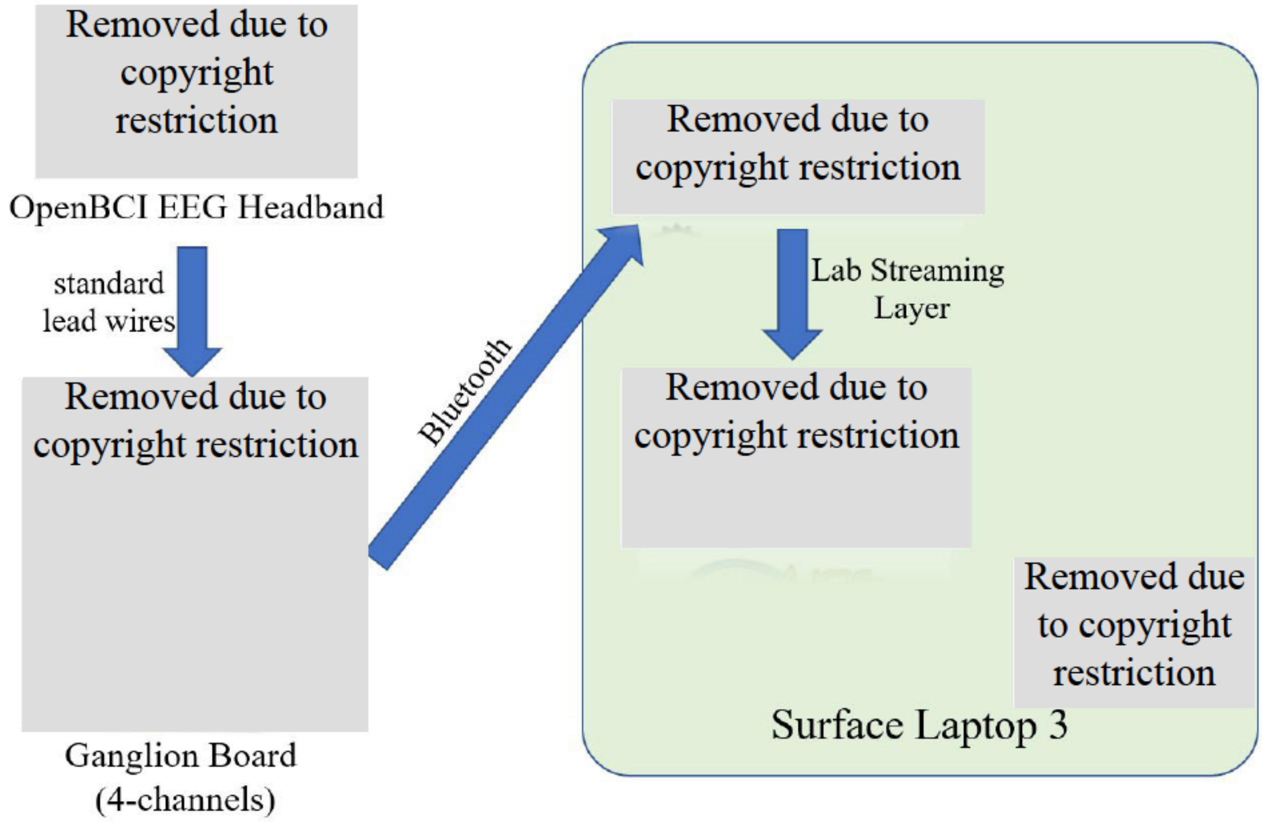

Fig. 2. The system connection diagram of the BCI analysis system

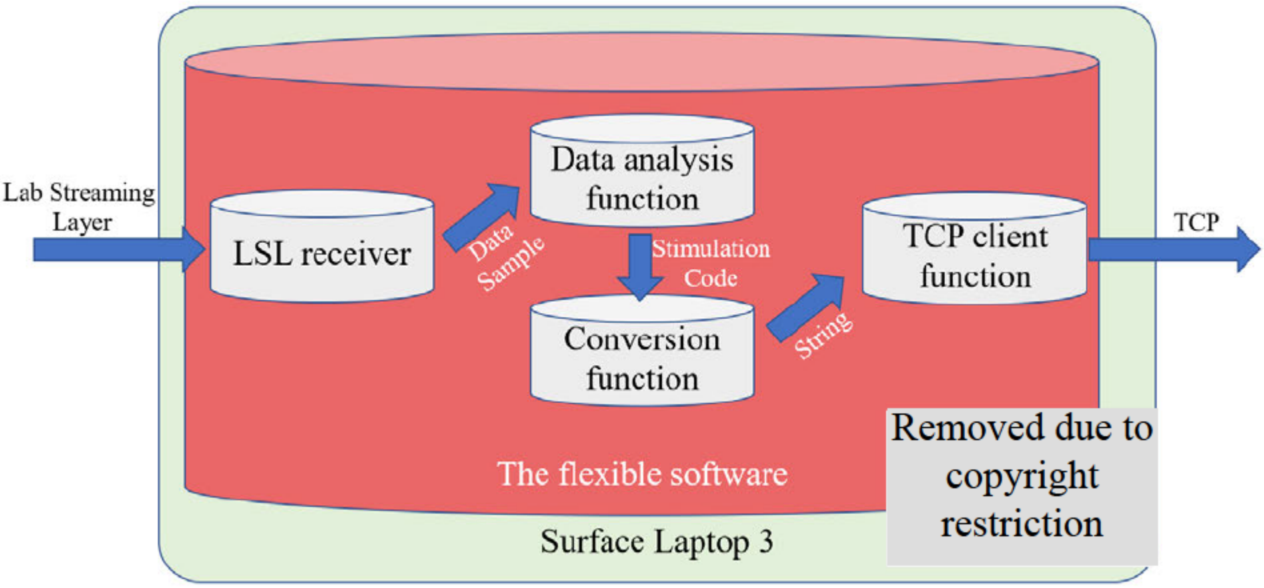

Fig. 3. The system diagram of the flexible software

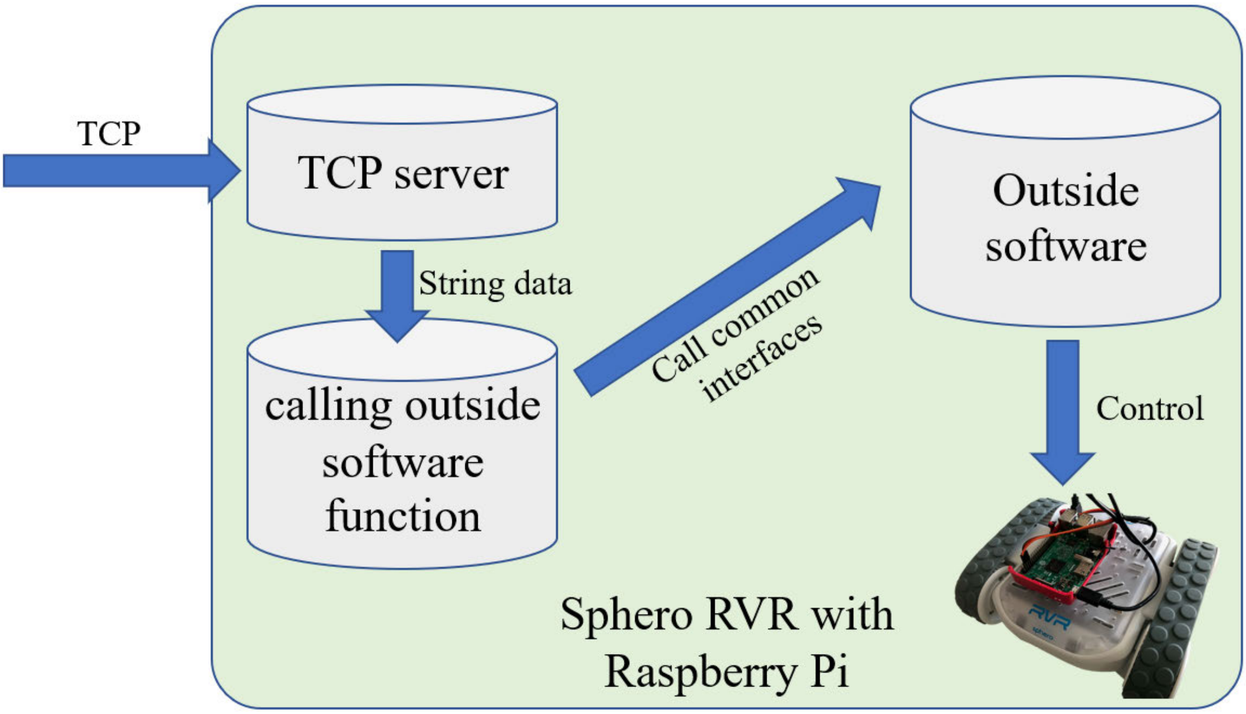

Fig. 4. The system diagram of control products system

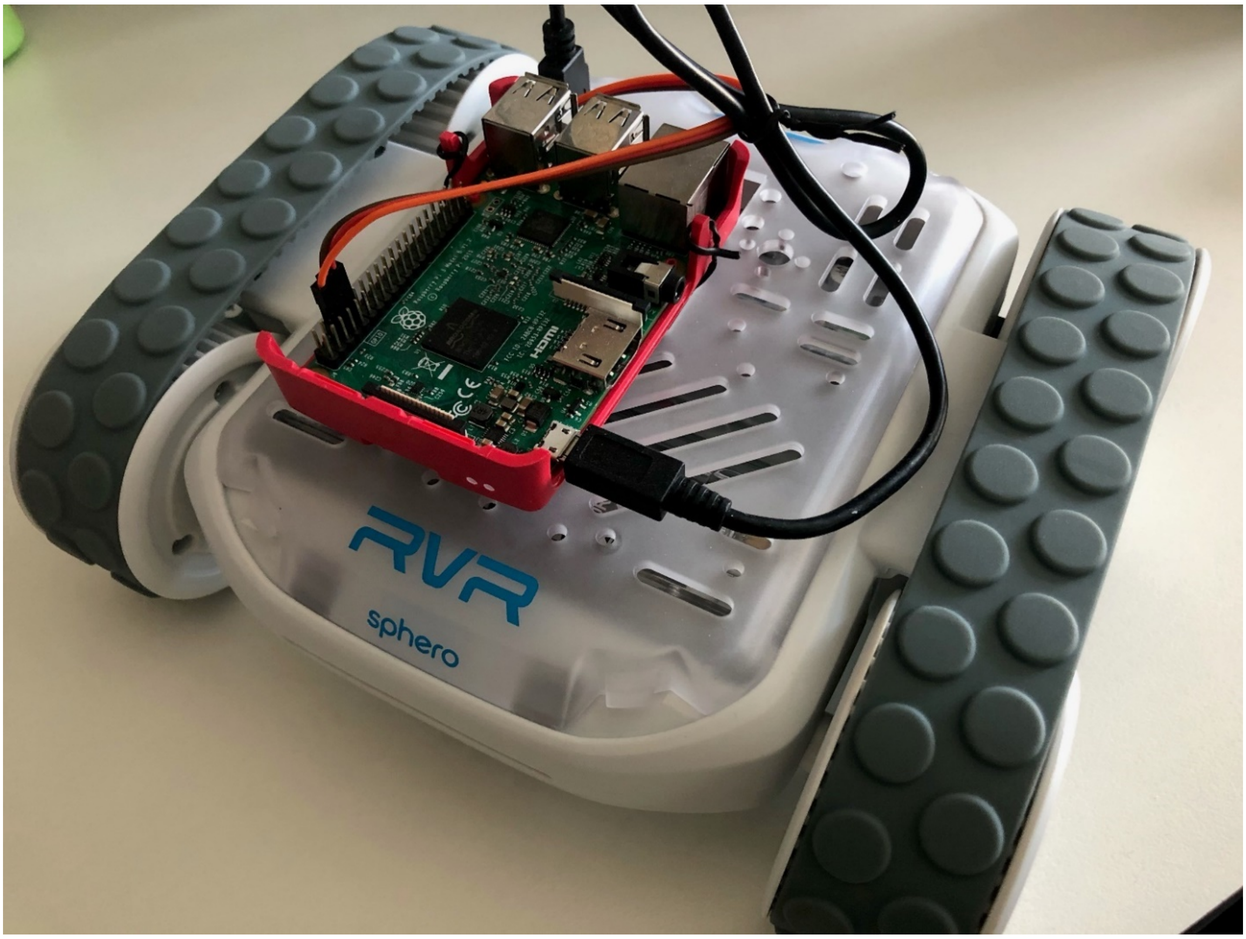

Fig. 5. The control target: Sphero RVR with Raspberry Pi

3.1 BCI analysis system

The BCI analysis system aimed to detect the jaw clenching stimulation and relaxing state, and it was constructed by combining a BCI device and two third-party software.

3.1.1 Lab Streaming Layer (LSL)

Many EEG devices support Lab Streaming Layer (Kothe et al., 2019), for instance, eego sports (ANT Neuro, 2021), Bittium NeurOne (Bittium, 2021), EmotivPRO (EMOTIV, 2021), OpenBCI (OpenBCI, 2021), and many other products. LSL has been used for research experiments because of its high reliability and valuable function for streaming data. LSL has an automatic failure recovery function and a time-synchronized function that supports sub-millisecond accuracy. The system put some sampling data together into a chunk, and it can stream data that include chunk and metadata. Moreover, developers can set a sampling rate and a number of the streaming channel. Additionally, LSL provides wrapper classes for several programming languages such as C, C++, Python, Java, C#, and MATLAB; hence, it can be easily implemented on EEG products. However, it is not familiar to general developers, and many robots or medical products only supported TCP or UDP as a method for sending and receiving data.

3.1.2 System flow and system environment of the BCI analysis system

The system used Ganglion Board (4-channels) with OpenBCI EEG Headband as the EEG equipment as shown in Fig. 6; the Headband connected two flat EEG snap electrodes and two dry comb electrodes via standard lead wires and two ear clips with replacement electrodes. Two dry combs possessed on the forehead and two flat EEG snap electrodes set on the back of the head. Two ear clips hold both tester's ears. EEG signals getting from the EEG equipment are sent to OpenBCI software version 5.0.5 via Bluetooth 4.0 interfaces.

As shown in Fig. 7, OpenBCI software was working on Surface Laptop 3 version 10.0.19042 build 19042 (Microsoft, 2021); installed Microsoft Windows 10 Home (64-bit) and equipped with Intel(R) Core (TM) i5-1035G7 CPU 1.2 GHz, 8 GB RAM, and 256 GB hard drive.

EEG signals were filtered under 50 Hz by a notch filter to remove environmental noise using an OpenBCI function. After that, the signals were transported to OpenViBE software version 3.1.0 through LSL protocol. LSL can stream wave data efficiently and is usually used for research experiments. It has high reliability and proper function, such as a time-synchronized function and automatic failure recovery function; moreover, it supports major programming languages. In addition, most EEG equipment support LSL as a transfer method.

OpenViBE software was working on Surface Laptop 3, although version 3.1.0 was not the latest version of OpenViBE because the newest version had a problem connecting Ganglion Board. OpenViBE software was composed by OpenViBE Acquisition Server and OpenViBE Designer; OpenViBE Acquisition Server received LSL stream from OpenBCI and sent them to the OpenViBE Designer. OpenViBE Designer was used for design scenarios and testing. The software has a lot of functions to analyze EEG signals and detect stimulations efficiently; therefore, developers can easily extract specific stimulation from EEG signals by combined function blocks. When OpenViBE received EEG signals from OpenBCI, it processed them into specific stimulations followed a created scenario. After that, detected stimulations were sent to the flexible software via LSL protocol.

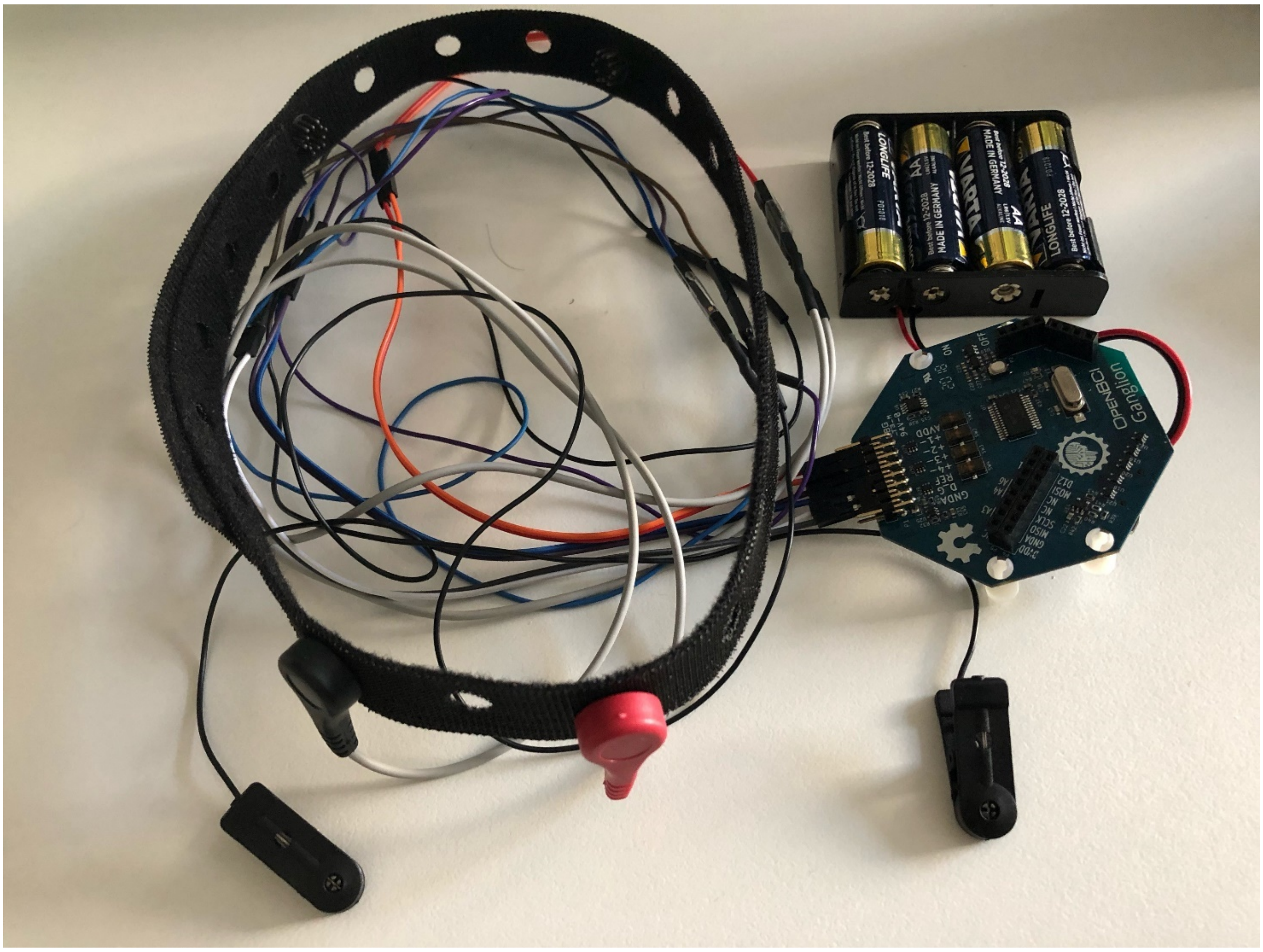

Fig. 6. Connection between OpenBCI EEG Headband and Ganglion Board (4-channels)

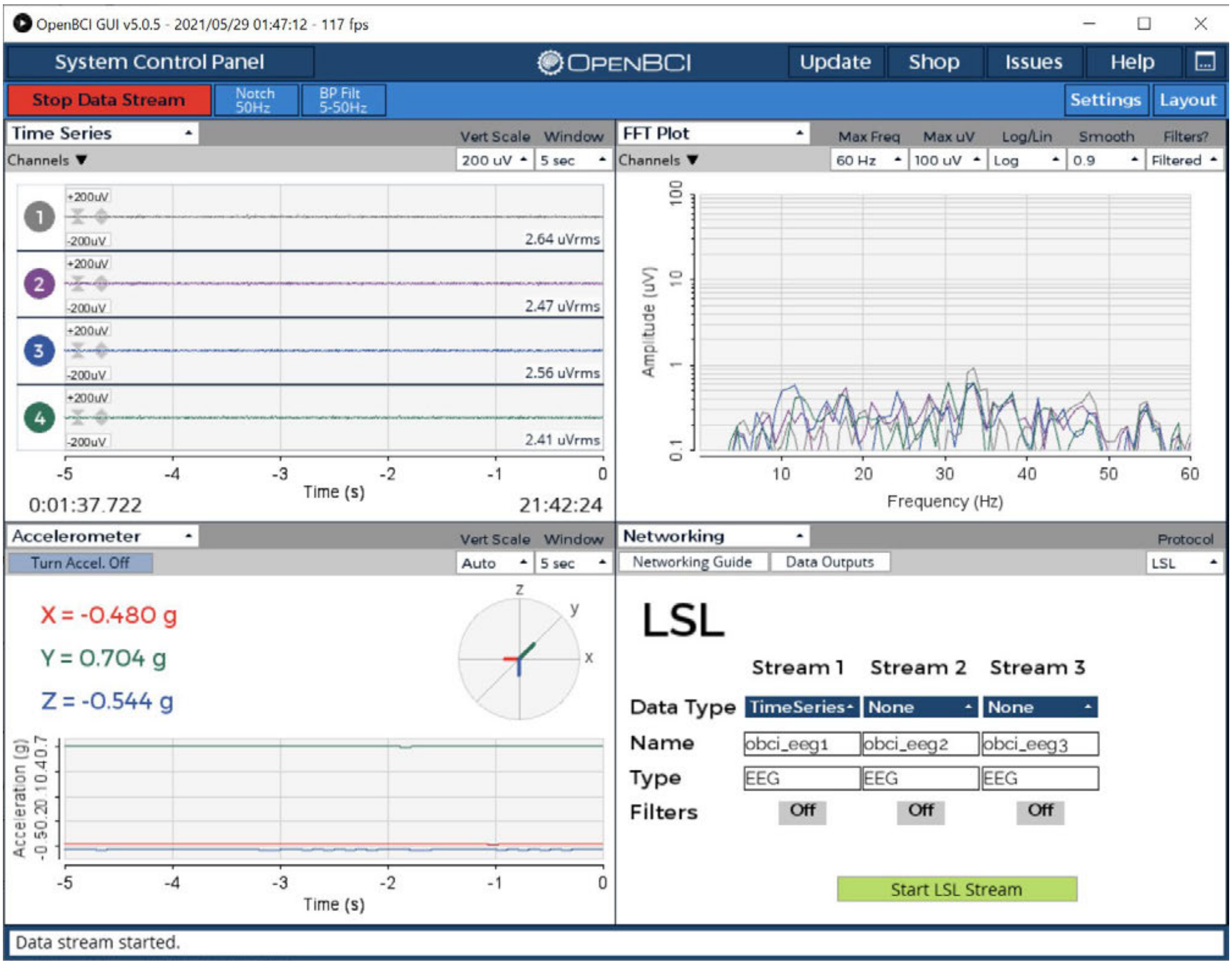

Fig. 7. Screen capture of OpenBCI that working on Surface Laptop 3

3.1.3 OpenViBE scenario for detecting jaw clenching

The scenario was created using OpenViBE Designer. As shown in Fig. 8, the scenario starts from the Acquisition client box. The box received EEG signals and could separate signals to transport them to other boxes. Then, the Channel Selector box extracted specific a channel that wanted to use because the single-channel contained enough EEG data to detect the jaw clenching. Subsequently, the signals were filtered into the alpha wave band (8-12Hz) by the Temporal Filter box to remove unnecessary waves. However, muscle activations tend to strongly occur in more high-power bands; thus, the alpha power band might not be suitable for detecting jaw clenching. The waveform was taking a too complex shape to detect specific conditions because signals were moving between the positive and negative values at this point. Hence, the system applied the root mean square (RMS) to the waveform by the Simple DSP box. Then the following two Simple DSP boxes shifted the signals based on specific thresholds. The thresholds represented switching status jaw clenching or relaxing; when the signal goes over point A, the tester took the jaw clenching; if the signal downed under the threshold B, it means the tester took the relaxing status. The signal Change Detector box could pass these statuses to the following boxes as stimulations; these detected stimulations were combined into one LSL data by the Stimulation multiplexer boxes. The Clock stimulator box kept creating defined stimulation to avoid time out of LSL and was combined with detected stimulations by the Stimulation multiplexer box. Finally, the LSL export box sent stimulations to the flexible software via LSL protocol. The LSL export box had two settings, type and name, and they should be set the same values of the flexible software. As shown in Fig. 9, The Signal display box was used to verify the status of converted signals step by step; the example showed that converted EEG signals took clearly different shapes in jaw clenching or not.

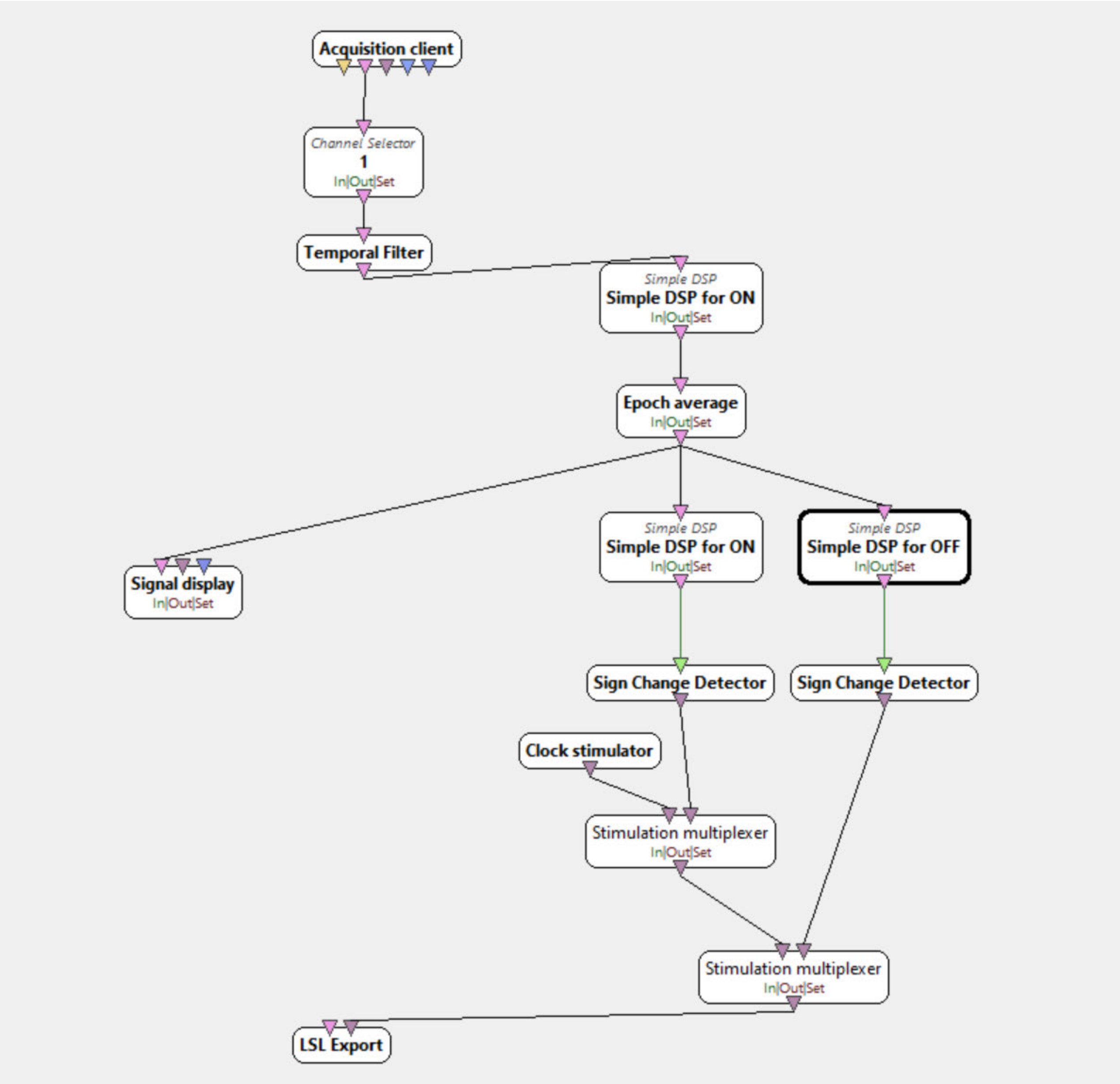

Fig. 8. The OpenViBE scenario for detecting jaw clenching

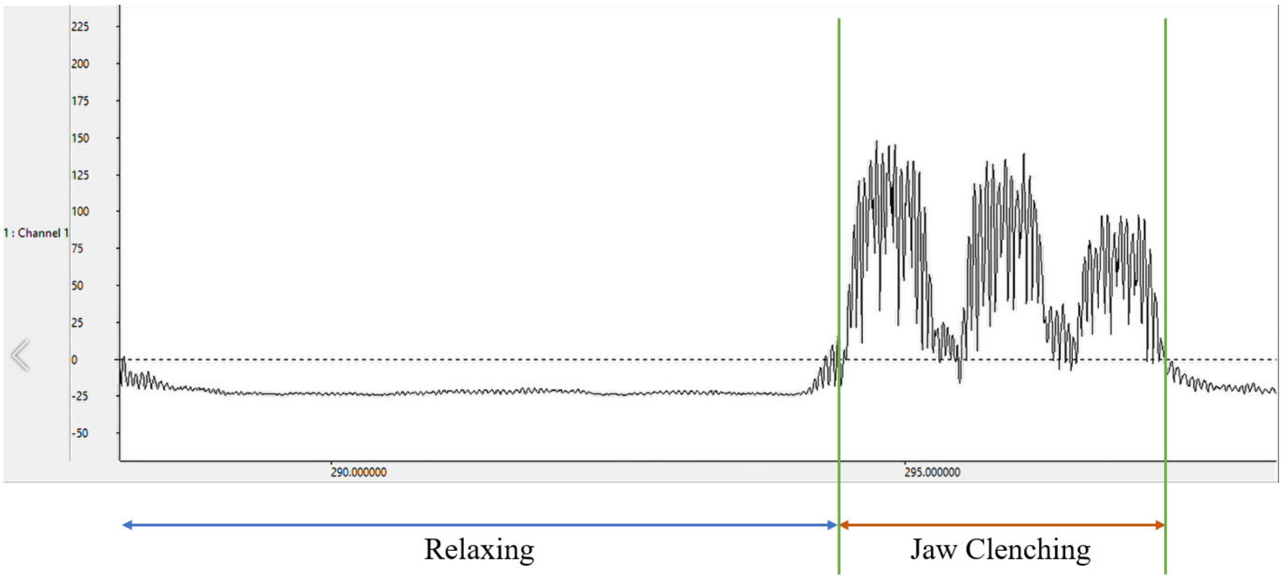

Fig. 9. Example of outputs of the OpenViBE scenario

3.2 Flexible software

The purpose of the flexible software was to provide a convenient connection environment between BCI and other products by a changeable config file, including a conversion table. The system converted the received LSL stream into simple string commands such as "Start", "Stop", and "Turn" using a defined conversion table, then sent converted commands to the control products system via TCP.

3.2.1 Transmission Control Protocol (TCP)

Transmission Control Protocol is a standard protocol for sending and receiving data in the world. The beginning of TCP was in 1974; Cerf and Kahn published "A Protocol for Packet Network Intercommunication" (1974); TCP is based on the technology provided by the paper. TCP has been used with many software and products to communicate to other systems; LSL is also using TCP internally. In addition, most programming languages support TCP by default; therefore, supporting TCP offers flexibility to communicate with other systems. Also, there is another standard protocol, UDP. As shown in the Table. 1, TCP is low speed compare with UDP, while TCP has advantages on the reliability and loss guarantee. Therefore, the project chose TCP as a transfer method.

	TCP	UDP	LSL - Communication	LSL - Streaming
Speed	Low	High	Same as TCP	Same as UDP
Reliability	High	Mid	Same as TCP	Same as UDP
Guarantee of loss	Yes	No	Same as TCP	Same as UDP
CPU cost	High	Mid	Same as TCP	Same as UDP
Easy to use	Most programming languages support TCP	Most programming languages support UDP	Useful to connect BCI and OpenVibe. however, does not have a wrapper class for several programming languages.	

Table 1. Characteristics of the transfer protocol

3.2.2 Software design

The software design was based on object orientation to maintain modularisation; it had three classes, main class, TCPIPControl class, and LSLControl class. The role of the main class was to control the TCPIPControl class and the LSLControl class. The TCPIPControl class had seven functions: initialize, delete class, observe TCP, TCP connect, TCP disconnect, and send command. Also, the LSLControl class had six classes: initialize, delete class, observe LSL, data analysis, create LSL stream, and close LSL stream. Each function had a simple design and appropriate logging. The system design diagram is shown in Fig. 10.

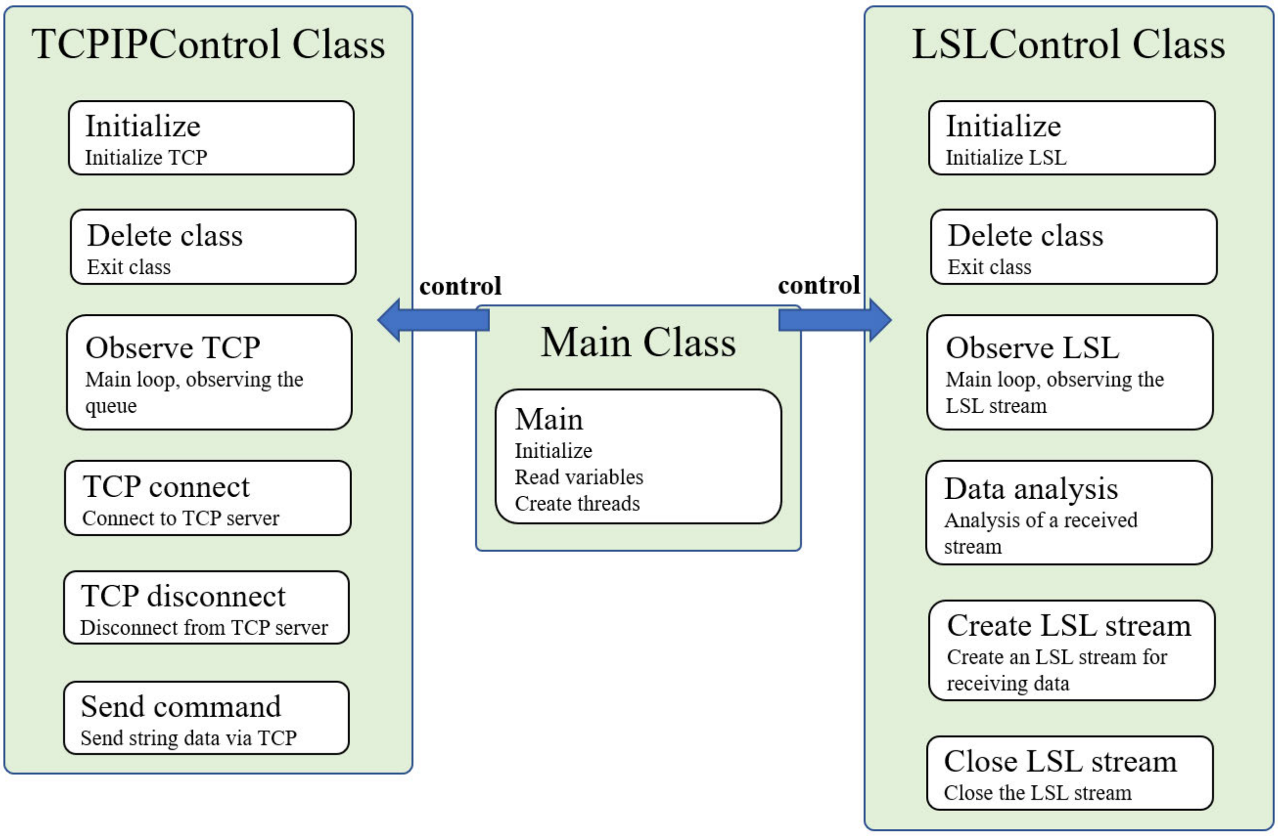

Fig. 10. Software design of the flexible software

3.2.3 System flow of the flexible software

The flexible software flow chart is shown in Fig. 11, and the setting file configuration is shown in Table. 2. The flexible software has two threads for TCPIPControler and LSLControler. Firstly, the system imported essential classes like socket, NumPy and so on. Then, it defined default parameters for common settings, LSL settings and TCP parameters. After that, the system loaded into variables a config.ini that included setting parameters and the conversion table. The main class started the TCPIPControl thread and LSLControl thread and waited to finished these threads. The LSLControl thread waited to found an LSL stream based on the loaded setting until defined waiting time. If the system did not find the LSL stream until a defined time, the system finished processing and informed an error message. After founding the LSL stream, the thread started to receive LSL data and put contained stimulation code into a global queue variable. The TCPIPControl thread observed the global queue variable frequently. If the queue contained defined code by the conversion table, the thread tried to connect to the control products system based on configurations. If the system could not connect the target IP, the system finished processing and informed an error message. After connection, the system sent string data (character code: UTF-8) corresponding to the stimulation code via TCP. Then the system disconnected the TCP connection and backed to the waiting status again.

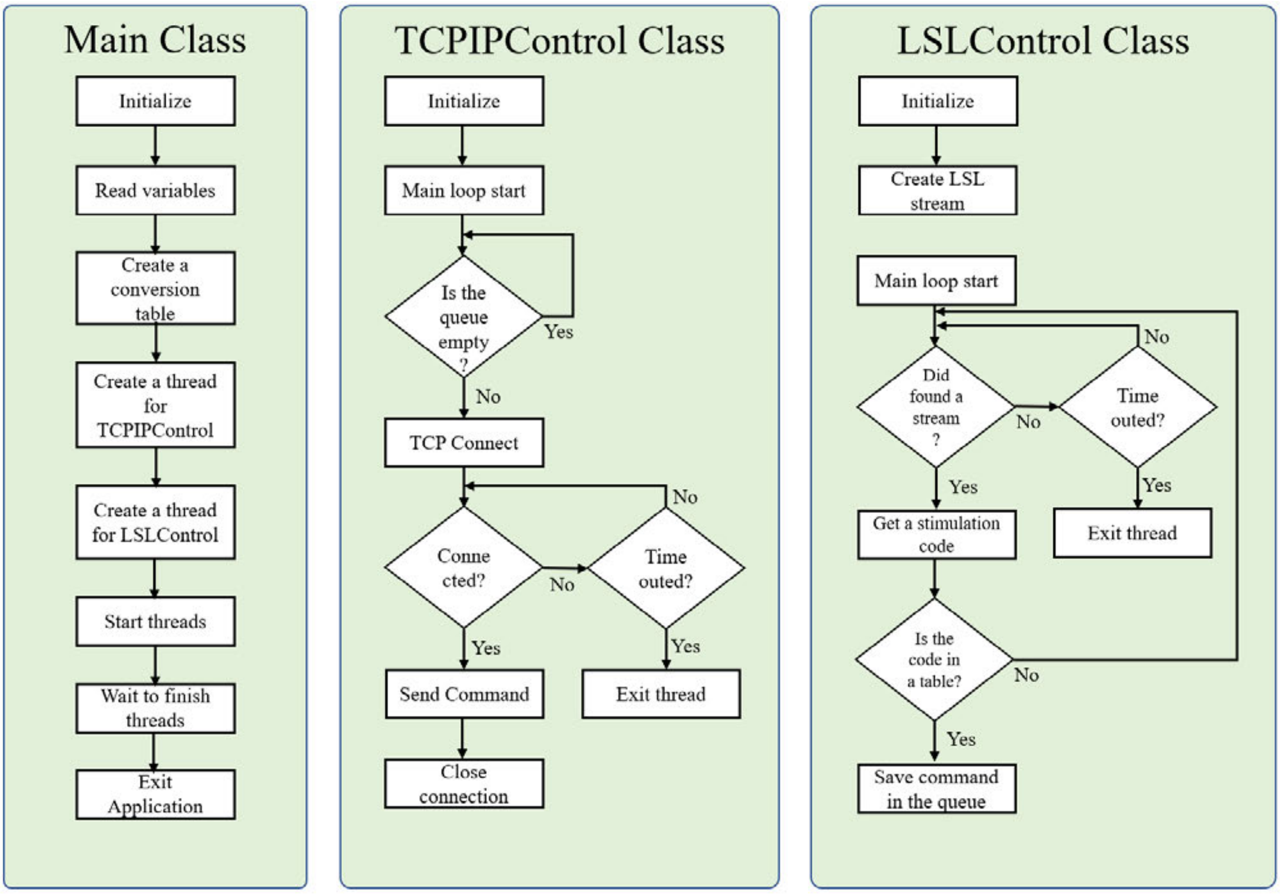

Fig. 11. System flow of the flexible software

Section	Keyword	Default	Detail
COMMON	DefCount	2	Number of the conversion table
	TimeRange	0	The time range for receiving one data in defined time. When the value is 0 or negative, nothing to do.
TCP	TargetIP	N/A	IP address of TCPCommandServer
	TargetPort	4000	Port number for TCP connection
	RetryCount	3	Number of the connection retry
	RetryWaitSec	1	Time of wait on each the connection retry
LSL	ReadType	EEG	LSL stream type
	ReadName	openvibeMarkers	LSL stream name
	RetryCount	3	Number of the connection retry
	RetryWaitSec	0.5	Time of wait on each the connection retry
RECEIVE	Def00	N/A	Numerical code of conversion source (e.g., 31000)
	Def01	N/A	Numerical code of conversion source
	:		Continues for the defined amount
SEND	Def00	N/A	String of conversion destination (e.g., start)
	Def01	N/A	String of conversion destination
	:		Continues for the defined amount

Table 2. configuration setting for the flexible software

3.2.4 Information of the configuration file

The configuration file (config.ini) was put on next to the flexible software. As shown in the Table. 2, it had five sections, "COMMON", "RECEIVE", "SEND", "TCP", and "LSL". The COMMON section included "DefCount" and "TimeRange"; the DefCount defined the number of pairs on the conversion table; the TimeRange was used as a length of time not received LSL data. The TimeRange was an optional setting; when setting the zero or negative value, it was ignored. The RECEIVE and SEND sections had the same number of definitions. The definitions started from "Def00" and continued with "Def01" and "Def02" based on the DefCount on the COMMON section. These definitions were the conversion table, and received LSL data were converted to string data based on the table. In the TCP section, there are "TargetIP", "TargetPort", "RetryCount", and "RetryWaitSec". The TargetIP corresponded with control products' IP address, and the TargetPort was the connection port for TCP connection. The RetryCunt was the number of retries for TCP connection, and the RetryWaitSec corresponded the seconds for the wait between each retry loop. The LSL section had four definitions, "ReadType", "ReadName", "RetryCount", and "RetryWaitSec". The ReadType should be set the same as the LSL stream setting on OpenViBE software; the default value was "EEG". Also, the ReadName should be set the same as the name of the LSL stream on OpenViBE; the default was "openvibeMarkers". The RetryCount and RetryWaitSec had the same meaning as the setting of the TCP section. An example of the configuration file is shown in Fig. 12.

```
[COMMON]
DefCount=2
TimeRange=0

[RECEIVE]
Def00=1094
Def01=1040

[SEND]
Def00=start
Def01=stop

[TCP]
TargetIP=100.92.103.102
TargetPort=4000
RetryCount=3
RetryWaitSec=1

[LSL]
ReadType=EEG
ReadName=openvibeMarkers
RetryCount=3
RetryWaitSec=0.5
```

Fig. 12. Example of the configuration file for the flexible software

3.3 Control products system

The control products system was a simple design. My system should call the familiar interface to keep flexibility because product control software depends on each product. My system implemented an RVR controller as outside software to move the Sphero RVR for evaluation.

3.3.1 System design

A TCPCommandServer software was also based on object orientation as the flexible software; it has two classes, the main class and a TCPCommandServer class. The main class controlled a thread for TCPCommandServer and accepting keyboard interrupt. TCPCommandServer had only four functions: connected, disconnected, received and time waiting. And each included appropriate logging. The system design diagram is shown in Fig. 13.

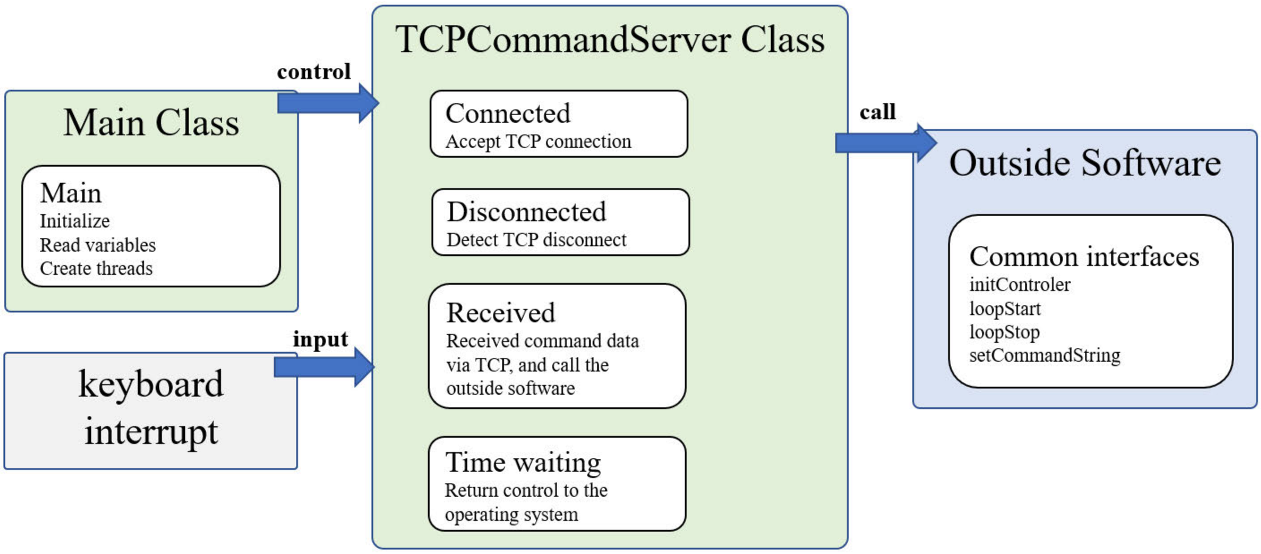

Fig. 13. Software design of the control products system

3.3.2 System flow of the control products system

The flow chart of the control products system is shown in Fig. 14. The control products system could be separated by the TCPCommandServer software (TCPCommandServer.py) and outside software. Firstly, the TCPCommandServer loaded a config file, including IP address, port information, and buffer size, to set TCP server settings. After that, the software called an initControler function implemented on outside software to initialize the outside software. The main class of the TCPCommandServer started a keyboard interrupt thread to stop software by user interruption and created a TCP server to receive TCP connections from the flexible software. After being ready to accept TCP connections, the software informed messages to the command prompt. Subsequently, the TCPCommandServer called a loopStart function implemented on the outside software to start a thread of product control. When the user inputted any character on the command prompt, the software stayed for an additional input with a confirmation message; after getting a response from the user. Then the TCPCommandServer called a loopStop function of the outside software, and the processing was finished. When the TCPCommandServer accepted TCP connection from the flexible software, the software informed the user and waited to receive string data (character code: UTF-8). Next, the TCPCommandServer called a setCommandString function of outside software with received string data as an argument.

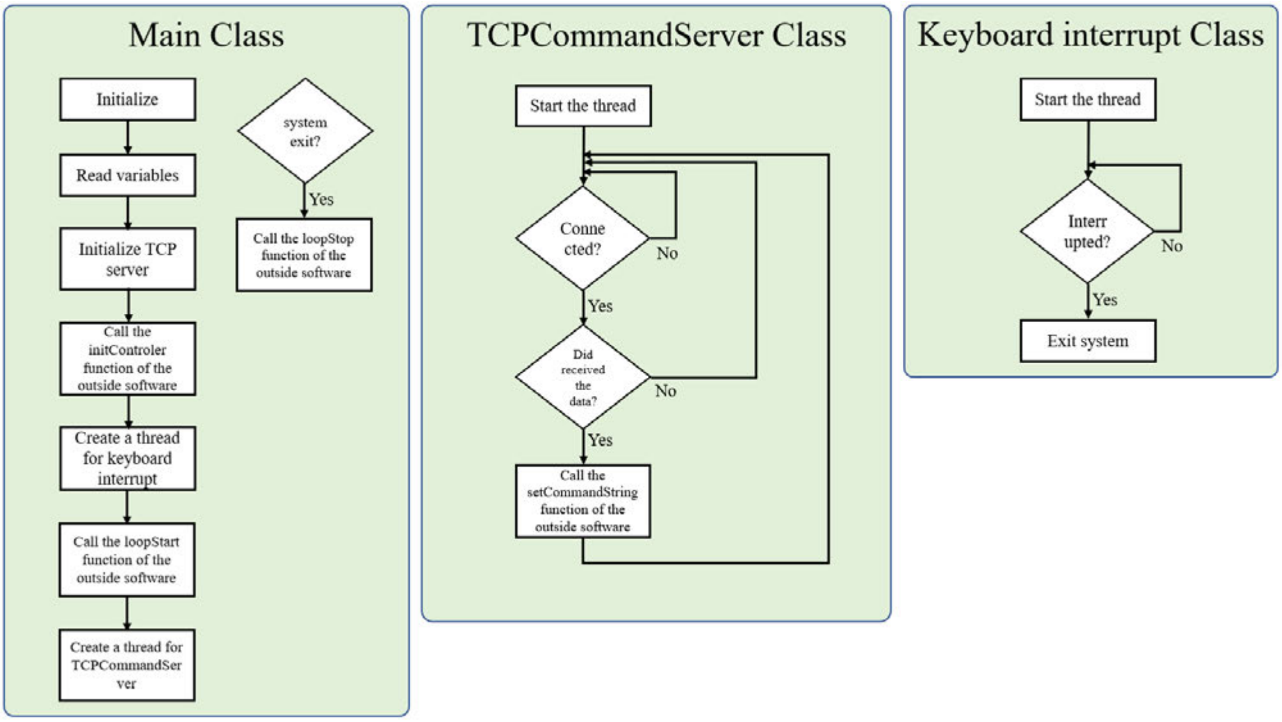

Fig. 14. System flow of the TCPCommandServer

3.3.3 Information of the configuration file

The configuration file (TCPCommandServer.ini) for the TCPCommandServer software was placed in the same directory as TCPCommandServer.py. As shown in Table. 3, the file only included a TCP section because the software was just a mediator for connecting the flexible software and outside software. The TCP section had three settings, "ServerIP", "TargetPort", and "BufferSize". The ServerIP corresponded with the device's IP address, and the TargetPort was a TCP connection port. The BufferSize was a memory size of the receive function.

Section	Keyword	Default	Detail
TCP	ServerIP	N/A	IP address of TCPCommandServer
	TargetPort	4000	Port number for TCP connection
	BufferSize	1024	Buffer size for receiving TCP data

Table 3. configuration setting for the TCPCommandServer

3.4 Control target

Sphero RVR and Raspberry Pi 4 were selected as control targets to evaluate the whole system. Sphero RVR supported javascript as a standard programming language; however, Sphero RVR only could be controlled via dedicated software developed by the manufacturer. Thus, my project chose Raspberry Pi 4 to extend the functions of Sphero RVR. Raspberry Pi 4 board supported Python version 2.7.x and 3.5.x, and it has several helpful interfaces such as USB ports, a Micro HDMI port, and a Wi-Fi device. As shown in Fig. 15, the board could easily connect Sphero RVR with two wires and a USB cable; the USB connection was only used for the electricity supply. Developers can connect a display and keyboard or Virtual Network Computing (VNC) protocol to control the device. Windows computers could easily connect with The Raspberry Pi 4 using the Virtual Network Computing (VNC) protocol; my project used the VNC Viewer as a connecting software. Therefore, Raspberry Pi was a pretty helpful device to control and test the whole system.

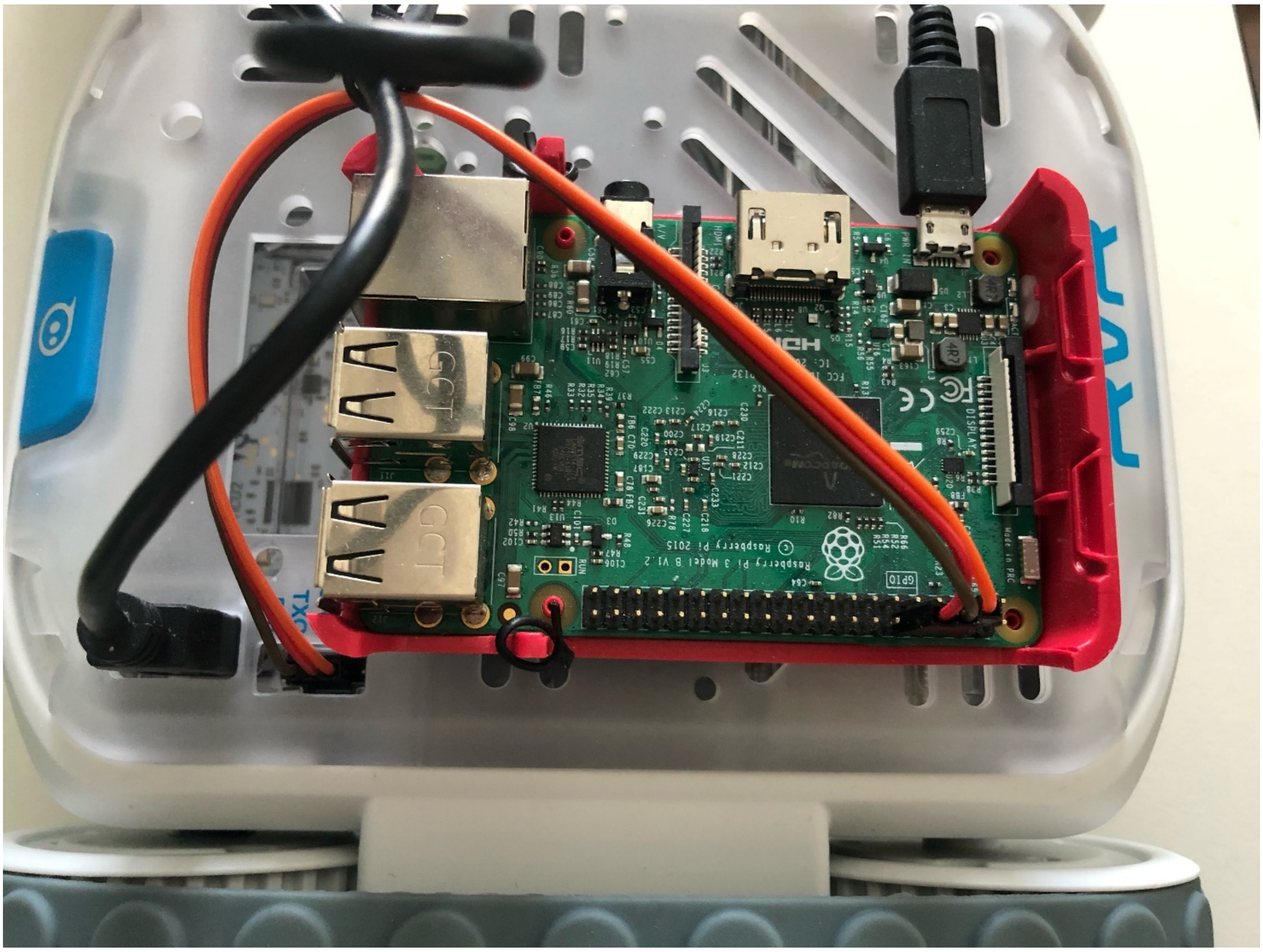

Fig 15. Connection between Sphero RVR and Raspberry Pi

3.4.1 System flow of the RVR controller (outside software)

The RVE controller had four global variables, speed, heading, flags, and loopStatus. The speed variable was used to store the speed parameter of Sphero RVR. The heading corresponded to the taking direction of Sphero RVR, and the flags was the direction of travel of Sphero RVR (0: go forward, 1: back). The loopStatus represented the status of an observation thread; if the loopStatus was True, the observation thread ran; otherwise, the thread stopped. The observation thread kept watching global variables and applied variables to the status of Sphero RVR. Sphero RVR could only get one order in 0.5 seconds; thus, the observation thread includes a wait function after applied variables. When the setCommandString function was called from the control products system, the function changed variables to values corresponding to an argument. The flow chart of the RVR controller is shown in Fig. 16.

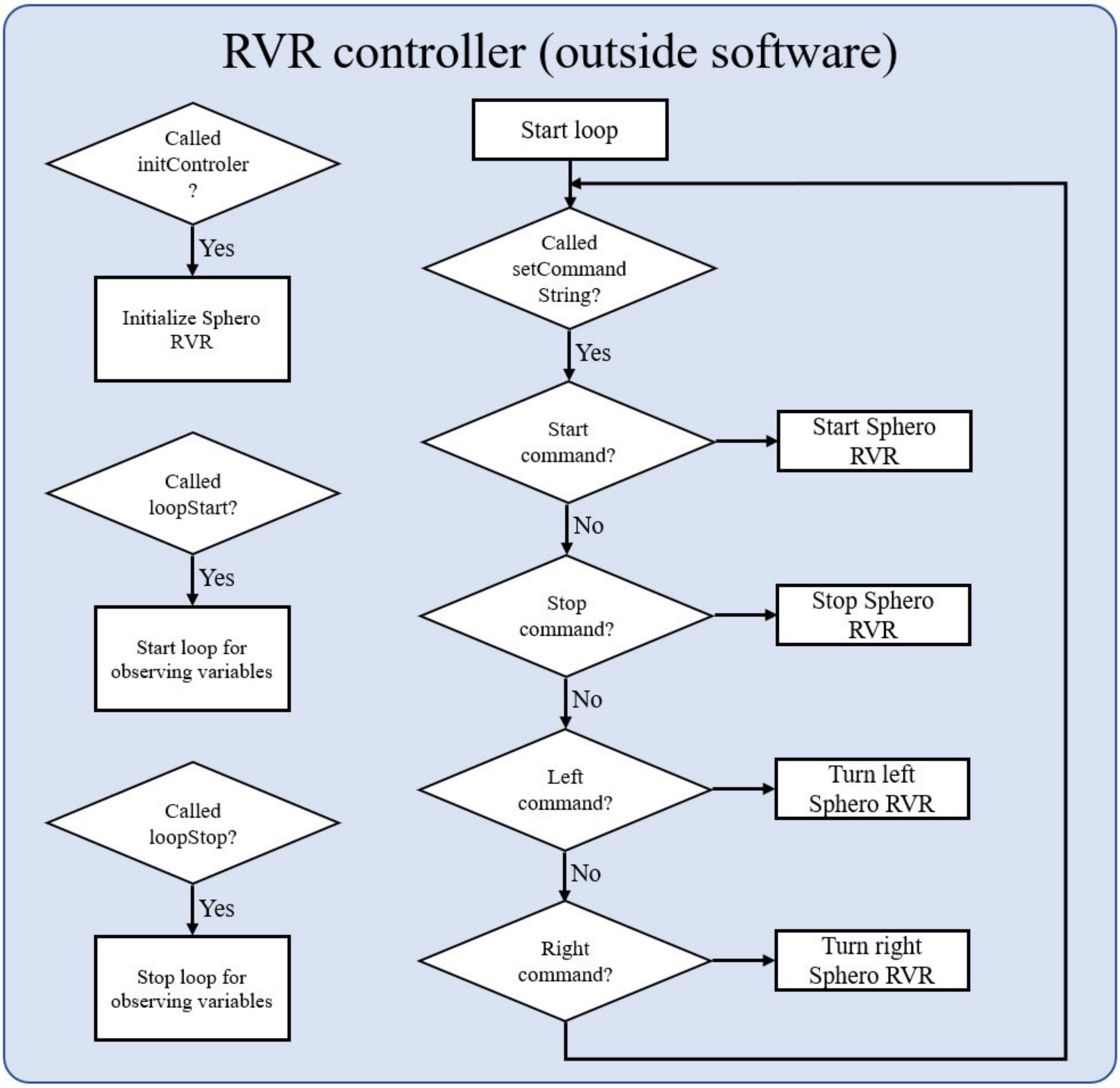

Fig 16. System flow of the RVR controller (outside software)

3.5 System evaluation method

The system was evaluated by doing a whole system test and, most importantly, a delay measurement test.

3.5.1 Whole system test

The whole system test aimed to ensure that the total system was moving well. The system included the BCI analysis system, flexible software, and the control products system.

3.5.1.1 Testing environment

The comprehensive testing was held on between the 13th to 16th of September in a small room (2 meters width and 5 meters depth). There was one male and one female tester, and each tester took jaw clenching or relaxing alternately; testing was recorded by video and evaluated after testing. Tester equipped OpenBCI headband and put both ear clips to their ear. After that, the tester sat on a chair far from the control PC to avoid electrical noise from the power tap and computer. The distance between the control PC and the tester was about two meters; when the space was less than two meters, the EEG device got many noises, leading to detection error.

3.5.1.2 Testing methodology

After the tester was ready, an electrical switch of the Ganglion board was turned on; then OpenBCI software was waked up on the Surface laptop 3. Subsequently, OpenViBE Acquisition Server and OpenViBE Designer were run on the computer. After that, data sending of OpenBCI was started via the LSL stream, and OpenViBE Acquisition Server began to receive them. Then finally, the OpenViBE scenario was executed on OpenViBE Designer. Testers were instructed to alternately took jaw clenching and relaxing at their favourite timing. Also, they suggested that open the lip widely to recognize the status easily when taking jaw clenching status. When the system received jaw clenching stimulation, it converted them to "start" commands, and when the system got relaxing statuses, they were converted to "stop" commands. After getting the start command, Sphero RVR kept moving to forward until receiving the stop command. Each tester took 50 times jaw clenching or relaxing. After testing, behaviours of Sphero RVR and testers were compared by recorded video, calculated accuracies for each status and collected delay time.

3.5.2 Delay measurement

3.5.2.1 Testing environment

After collected the delay time from whole system testing, the delay measurement was done in a room with a width of 2 meters and a depth of 5 meters. As the same as the whole system test, two testers cooperated in the measurement on the 17th and 18th of September. Each tester equipped the EEG headband on the head and also equipped both ear clips on their ear. The testing was executed one person by one person, as shown in Fig. 17, the display out of the testing computer was recorded to the video.

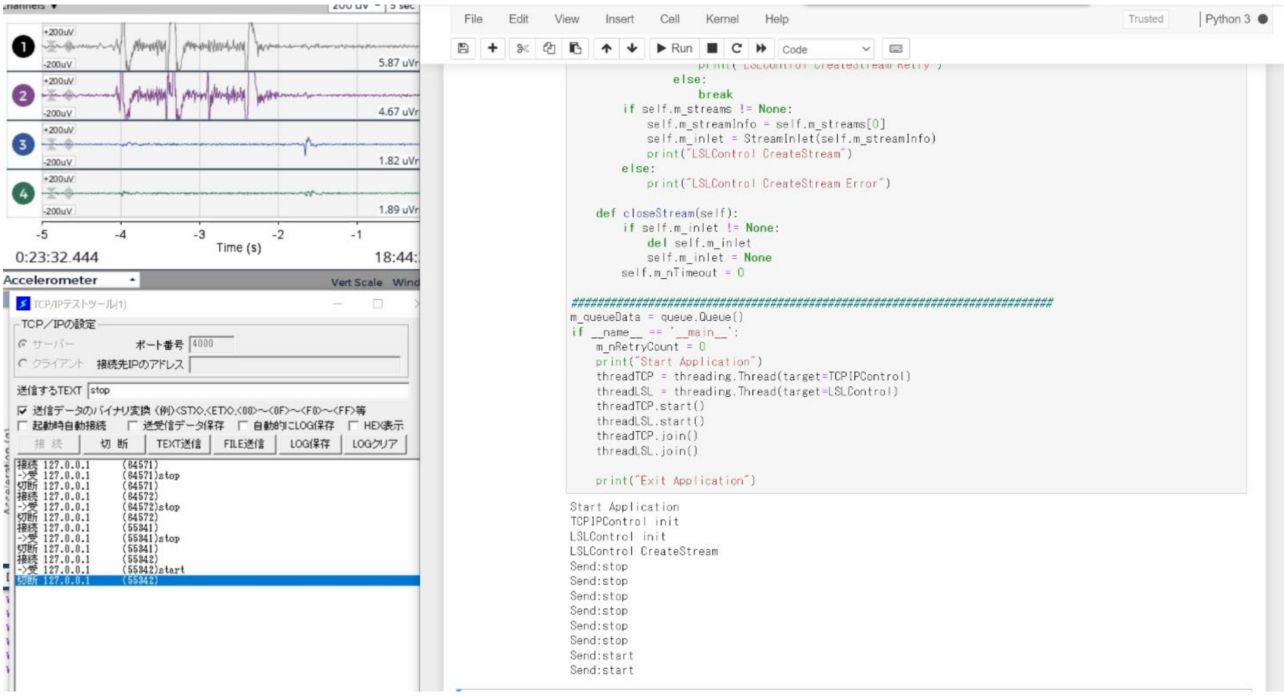

Fig 17. Screenshot of the video file for the delay measurement

3.5.2.2 Testing methodology

The testers were requested to do jaw clenching at the specified timing 25 times each. Although each system showed log data on display except OpenViBE software, the timing of the behaviour of the OpenViBE could be ensured from the showing waveform. Therefore, the measure was executed by comparing each system log's timing and the waveform on OpenViBE. Moreover, function speed testing for the flexible software was done to collect processing speed. After function testing, the flexible software was tested with dummy LSL software that is simply sent numerical data every 0.5 seconds to detect the cause of the transfer delay. The send and receive test was done with 100 data sending and collected.

4. RESULTS

The purpose of the system is to provide flexible software to connect BCI and other products; Hence, the most critical point is proof of the flexibility. To ensure that, the system was tested on different BCI methods without software editing. Moreover, also evaluation for the delay time is necessary because delays on the system might cause controlling mistakes or accidents. Additionally, not only delay time but finding the cause of delay is essential. Accuracy is also important, while the accuracy might be based on the OpenViBE scenario design; Thus, the priority of evaluating accuracy is low than others.

4.1 Whole system test

One hundred behaviour samples were collected from two testers to evaluate the whole system. The results of the testing are shown in Table. 4. Jaw clenching was 100% correctly detected as the start command; in all testing cases, testers could start to move Sphero RVR forward. On the other hand, the relaxing status was 77.8% correctly recognised as the stop command, and there was no miss classification as jaw clenching; however, 22.2% of the relaxing behaviour was missed. After the testing, both testers claimed bit gums pain, which continued for about one day. Sometimes, the status of the Sphero RVR had to reset because of the stop command lost. Moreover, there were several delays between taking activities and reflecting command to Sphero RVR. The length of the delay time was identified in the Delay measurement testing. The system test image getting from a recorded video is shown in Fig. 18.

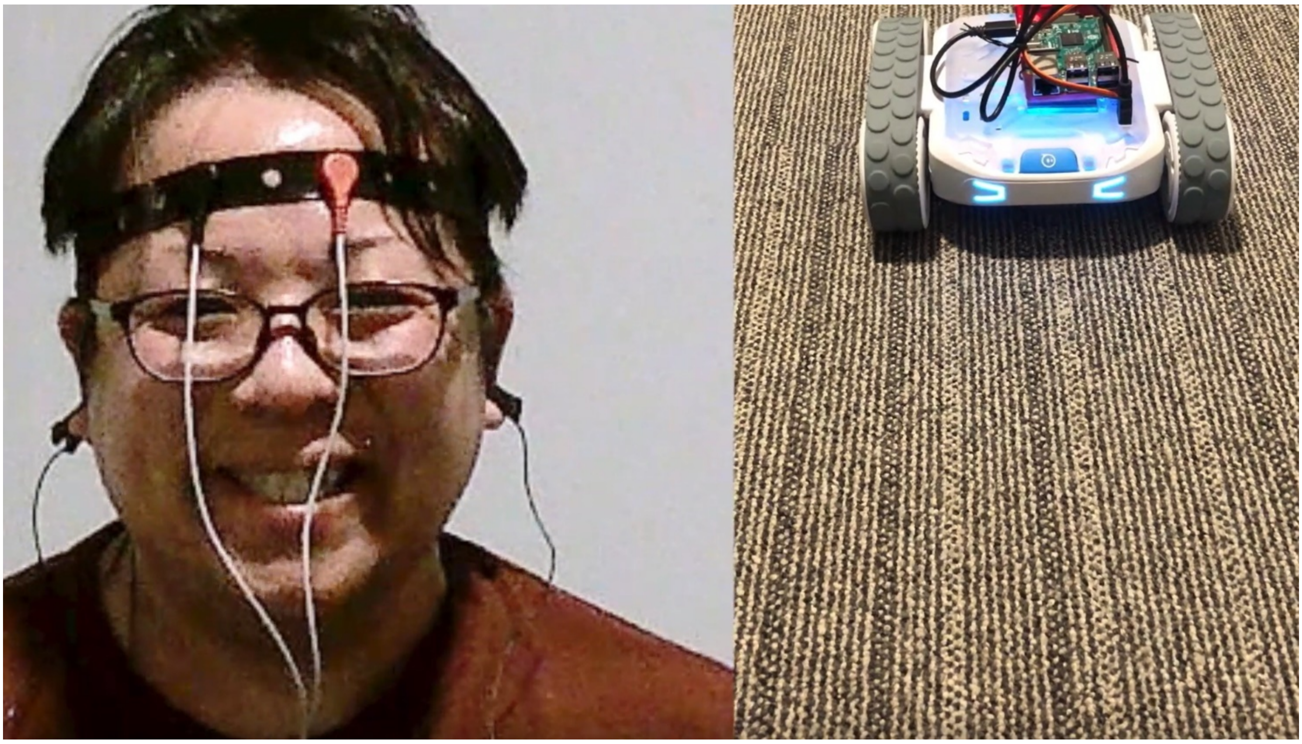

Fig 18. The Captured image of the video file for the whole system test

	Moving	Stopping	Missed	Average delay (Second)
Jaw clenching	100%	0%	0%	2.48
Relaxing	0%	77.8%	22.2%	3.24

Table 4. The results of the whole system test

4.2 Delay measurement

The delay measurement was executed to identify the system's bottleneck because, in the whole system test, there was some delay between testers' activities and applying movement to Sphero RVR. As a result, the average of taking time between OpenBCI to OpenViBE was 0.56 seconds. Sixteen out of twenty cases took the range of 0.2 seconds to 0.4 seconds; however, three were over 1.4 seconds. Subsequently, between OpenViBE and the flexible software took an average of 0.50 seconds; it also had two cases of long delay (1.89 and 2.0 seconds) as between OpenBCI to OpenViBE. On the other hand, between the flexible software to the control products system took under 0.33 seconds. The minimum observation time was 0.33 seconds because measurement values were calculated by the captured video frame (30 frames per second). Thus, an additional delay measurement was done to ensure the processing time of conversion and TCP transfer. As a result, command conversion took an average of 0.048 seconds, and TCP transfer used an average of 0.101 seconds. After that, the system was connected with dummy LSL software to find the system delay's bottleneck. As a result, the flexible software received an LSL stream once in each second.

5. ANALYSIS

5.1 Whole system test and Delay measurement analysis

As a result of the testing, the system could provide high accuracy of the stimulation conversion. Occurred stimulations were correctly captured and converted to string based on the conversion table. Several detections missing were caused by the design of the OpenViBE scenario; it requires professional knowledge of EEG analysis; hence, the missing stimulation problem will be resolved by improvement of the OpenViBE scenario. From the testers condition after the test, which was feeling a bit of gums pain, using jaw clenching as the stimulation of the moving trigger was not suitable for repeat the test. In addition, the delay problems were occurring in the LSL stream; these delays caused the delay between testers' activity and applying movement to Sphero RVR. Between the flexible software and the control products system had no delay. Although the cause of the delay on the LSL stream has not been identified yet, the reason might be the system setting for the LSL library or the system environment because they did not always occur. The LSL library has a buffer size setting; however, users cannot change the value from the UI of OpenBCI and OpenViBE. Therefore, there was no way to ensure that the buffer size was the cause or not. Also, there were three communications (two LSL streams and TCP) in the same laptop computer and worked much software to execute the whole system. Hence, using a high spec testing computer or separating works to multiple computers might resolve the delay problem.

6. DISCUSSION

The project aims to provide software that easily connects BCI devices and other products without editing software to reduce implementation costs. The flexible software was modularised well to connect many systems that using Lab Streaming Layer and TCP.

Lab Streaming Layer, which uses EEG devices, is not a standard protocol, and most products do not support it as the control protocol. Therefore, the mediation software between BCI devices and other products is helpful for manufactures who want to implement the brain control system in many areas.

There are lots of biosignal hardware; according to Kothe et al. (2019), they claim that the major EEG systems use LSL on the market. For example, twelve systems that target biosignal hardware use LSL; furthermore, seventeen devices that include ANT Neuro eego sports, Brain Products LiveAmp, Starcat HackEEG Shield for Arduino, and so on, also support LSL as a transfer protocol of vendor-provided software. Additionally, three biosignal hardware support LSL natively, and several functional near-infrared spectroscopy (fNIRS) hardware include LSL. Also, much eye-tracking hardware support LSL, such as Eye Tribe Tracker Pro, SMI iViewX, Tobii Eye trackers, and more. Besides them, over twenty products, which include human interface hardware, motion capture hardware, multimedia hardware, stimulation hardware, and stimulus presentation software, support LSL. That means that the flexible software can receive LSL data from over fifty products in many areas as a moving trigger. Therefore, the flexible software can connect lots of sectors and controllable products. For example, there is an approach to detect methods for facial emotion recognition from Autism Spectrum Disorder (ASD) person using eye-tracking and EEG (Black et al., 2017); the flexible software might be used for the eye-tracking device and the EEG device to get stimulations. Moreover, Aristidou and Lasenby (2010) proposed a real-time hand gesture system with PhaseSpace, a motion capture device supporting LSL. The hand gesture system can be used in the gaming sector, medical, art, and so on; therefore, the flexible software also helps implement and develop these areas. Hence, it can be said that the software has high extensibility.

On the other hand, TCP is a more general protocol; the standard protocol has been used in many devices and systems to implement transfer functions. For instance, there is an approach to control the assembly machine for the ITER vacuum vessel via TCP command (Li et al., 2013). Moreover, several famous game development software and 3D design/control software, including Unity (Unity Technologies, 2021) and Maya (Autodesk Inc, 2021), can be controlled through TCP. In this way, supporting TCP as a transfer protocol to connect the controlling target is essential for software generalization. Hence, the flexible software has the possibility to connect many types of sensor devices and systems to any product in the world.

Additionally, my software must be flexible for conversion. Because there are many BCI methods such as P300, SSVEP, MI, and hybrid BCI, what kind of stimulations will be output from EEG systems depends on the developer's scenario. As mention above, there are also a lot of devices and systems supporting LSL; thus, the software should provide a wide range of flexible entrances. My flexible software has a conversion table to convert numerical code to string; it can be edited by standard editor software. In addition, the software has a function to limit the number of received stimulation to one in the defined time range; it provides avoiding excessive data receiving to fit output to control devices. Thus, anyone can easy to implement BCI devices to their product using the flexible software. As a result of the system test, the flexible software could correctly connect the BCI system and the robot.

However, current software has a limitation on the receiving LSL stream. Although the software only analysis the first data of the sample, the LSL stream can contain additional channel data in the sample. Depending on the products, the channel data area might have vital information to distinguish a stimulation into multiple stimulations. It is too difficult to predict what the additional channel data contains because the meaning of the data and the number of channel data also depends on the system. Thus, the project only focused on the first data of the sample in the LSL stream. As the future work, containing the channel numbers in the conversion table and supporting a combination of the stimulation as a trigger will be pretty helpful to expand the system flexibility.

Evaluating the point of the view of processing speed is also essential on the control system because system delay directly affects the working of the control targets. Moreover, users might do double operations because of miss understand caused by system delay. These risks might lead to users' confusion, accidents or injury. For example, when a patient controls a wheelchair via a BCI control system, if there are system delays, the patient cannot move the wheelchair at their timing, which might cause falling from the upper floor or protrude to the roadway. Or, when controlling drones with BCI devices, it requires precise movement to avoid crashing obstacles. If the control target moves at 6 kilometres per hour, which is the maximum speed for the wheelchair, and the control system has a 1-second delay, when the user tries to stop the target, the control target will stop after moving to 1.7 meters forward. However, if the system delay is under 100 milliseconds, the overrun distance will be under 17 centimetres. The delay measurement result shows that the flexible software only requires an average of 0.048 seconds for LSL stream conversion, and sending a command via TCP took an average of 0.101 seconds. Thus, the software only needs about 0.150 seconds for each processing; it means it has enough performance to control products. While the results also mean that the system only can receive LSL data once in 0.150 seconds; hence, if the devices send multiple samples in 0.150 seconds, the system will get delayed. Moreover, the system has a bottleneck on the LSL connection.

Although the cause of the delay in LSL communication is unknown yet, the delay never happened every time. Therefore, the cause of the bottleneck might be a buffer size setting of OpenBCI and OpenViBE software or a lack of processing power of the testing computer. Thus, the delay caused by LSL might resolve by manufactures' system design. The system developer should consider the specification of LSL and should adjust the buffer size setting to avoid system delay.

Hence, the flexible software has enough function and processing speed to connect lots of biosignal hardware to many products, which have TCP as a transfer protocol. From the fact of that, my project can expand the LSL contained system's viability because reducing the implementation's difficulty leads to developing new products. For example, according to Unity technologies (2021), Unity can build software for VR and support Python as a programming language. That means that the flexible software might connect BCI devices and VR devices. Not only that, Unity provides development environments for many platforms, such as smartphones, Windows, Mac, Linux, gaming consoles, and the web (Unity technologies, 2021). Therefore, the flexible software can connect biosignal devices to these platforms. That leads to help people with tetraplegia or disability because patients might be difficult to control entertainment software by their hands; thus, developing new products using biosignal devices as a controller is helpful for them.

Additionally, many gaming software companies might develop BCI control software in the future. The new gaming system has vast excellent entertainment possibilities, and many people have imagined playing games via BCI since a few decades ago. The project might accelerate the gaming industry to the future by providing a shortcut for connecting products.

Furthermore, developing new biosignal control systems might increase the employment rate because the control systems, which have high accuracy and low delay, might provide opportunities for employment for disabled people and make remote working possible. And in the COVID-19 situation, remote working demand is increasing; according to McKinsey＆Company, de Smet, Langstaff, & Ravid (2021), fifty-two per cent of workers want hybrid work, eleven per cent of workers want to work online in the world. The number of workers who prefer hybrid working increased 22 % than before COVID-19; the number of people who preferred online working increased 3% after facing COVID-19. Combining motion capture devices and my software might control many industrial machines or robot arms; it provides convenient remote working environments. Thus, the project is also helpful to workers worldwide.

Current flexible software contained the LSL class and the TCP class as one package. However, the LSL class and TCP class should be separated into other script files in view of the modularisation. It leads to given further flexibility in my project. If the software can easily change the input and output transfer protocol, the software will be able to support any product in the world.

However, there was a limitation on the testing environment. The testers were only two-person, and the number of attempts was less than 50 times on each testing. In the future project, the researcher should test with many subjects and repeat each test more.

Moreover, the project only tested with Sphero RVR as a control target. The control target should extend more to prove the flexibility of the software. Also, the system only tested with OpenBCI Headband; additional biosignal devices will make suitable testing environments and give more concrete results to the researcher. Moreover, the testing computer was not good for evaluation; the system was constructed in one laptop PC, and many software might be moving in the background. Thus, the testing computer should be separate each system for evaluation correctly.

In addition, the moving trigger should be changed from jaw clenching to other stimulation because the testers claimed small pain on their gums after repeating the testing. A suggestion for alternative stimulation is eye blinking. According to Podoprikhin (2015), detecting eye blinking requires a machine learning approach and training while testers' load will be reduced. Moreover, the filtering method for detecting stimulation should be considered by new knowledge obtained from the literature review.

7. CONCLUSION

The project aims to provide flexible software that connects BCI devices and many controllable products such as robots to helping manufacturers who want to implement brain signal control systems into their products. The software was designed based on object orientation and modularised each subsystem to realise the flexible system. The system supports Lab Streaming Layer as a receive data protocol and supports TCP as a transfer protocol for connecting to control targets. Although the LSL is an uncommon transfer protocol, it is used by many biosignal devices such as EEG equipment, eye-tracking hardware, and motion capture system to stream data with high stability. Thus, supporting LSL is essential for connecting BCI devices efficiently. On the other hand, TCP is the standard transfer protocol in the world; many operating systems, devices, products use TCP to communicate with each other. Moreover, most programming languages provide convenient classes to use TCP easily. Hence, control software should support TCP to take communicate with other systems and products smoothly. Furthermore, the system should provide the changeable conversion table without compiling and editing the software to maintain manageability and avoid system bugs. The flexible software implemented the conversion table into the config file. The table contains numerical codes for receiving data and string data to convert numerical codes to defined words; it can be edited by standard editor software easily. Therefore, anyone can make a conversion rule without knowledge of the programming language. In addition, excellent mediation software should provide reliable accuracy and high processing speed to avoid users' miss recognition, miss operation, confusion, and accidents. These essential points were evaluated by whole system test and delay measurement; it was held in a small room with the cooperation of two testers. As the evaluation result, the system provided 100% correct conversion; the processing speed between starting conversion and finishing transporting command via TCP was the average under 0.15 seconds. That means the project is quite notable for providing the system to mediate biosignal devices and many products; developing a new brain control system using flexible software might lead to expanding the possibilities of disabled people.

Moreover, connecting BCI to industrial machines might lead to increasing the employment rate by spreading possibilities of remote working. Also, disabled people might get opportunities for getting new jobs. Additionally, the project can provide an easy way to implement connecting biosignal devices and gaming systems.

Hence, many industries' possibilities will be enhanced by using flexible software as a mediation. However, the system had a problem with the transfer delay. Between OpenBCI to OpenViBE and OpenViBE to the flexible software had an over 1-second delay several times. The problem might be caused by buffer size settings for LSL or testing environments. However, the buffer size cannot be changed by a user interface on the OpenBCI and OpenViBE, and cost limitations cannot change the testing environment. Thus, an actual cause of the problem has not been identified yet. In future work, software developers should bring the LSL settings to the user interface and consider the effects of the buffer size on the delay.

Subsequently, there were several limitations; the testing and evaluation were done in a limited space with only two testers. Moreover, the number of the attempt for the testing was executed around only 50 times each testing because Jaw clenching, which was a stimulation trigger for the testing, damaged their gums and teeth slightly. Therefore, the detect target of the stimulation should be changed to eye blinking in additional research.

Additionally, testing equipment was limited; the testing computer was a middle spec laptop (Surface laptop 3), and it was not optimal to evaluation the whole system in one computer. Moreover, the testing only used the EEG headband as a BCI device and Sphero RVR as a control target.

Therefore, testing environments should be reviewed in future experiments. Stimulation trigger should be changed to easier one like eye blinking or open/close eyes to increase the number of testing attempts. Furthermore, testing computers should be separated by each subsystem for accurate evaluation results; also, EEG equipment and testing targets should be expanded to prove the usefulness of the system in many areas.

As future works, the system should be expanded its functions to support many types of LSL streams. The LSL analysis function only focused on the first data of an LSL sample; however, the number of the data in one sample depends on each system. Thus, to enhance the flexibility of the software, the software should expand the conversion table and analyse all data in the LSL sample. Additionally, supporting a combination of multiple stimulations convert into one command is also helpful to communicate with many devices. Furthermore, separating the flexible software into the main class, the LSLControl class, and the TCPIPControl class will be useful because these modules can easily be replaced according to the product.

Finally, finding from the literature review study, there are helpful many techniques and methods for the BCI system. Therefore, implementing several specific techniques to the future project as a function collection might highly enhance the usefulness of the project.

REFERENCES

Alimardani, M. 2018. Brain-Computer interface and Motor Imagery Training : The Role of Visual Feedback and Embodiment. IntechOpen.

Amiri, S., Rabbi, A., Azinfar, L., & Fazel-Rezai, R. 2013. A Review of P300, SSVEP, and Hybrid P300/SSVEP Brain- Computer Interface Systems. Brain-Computer Interface Systems - Recent Progress and Future Prospects. Published. https://doi.org/10.5772/56135

ANT Neuro. 2021. viewed September 2021, https://www.ant-neuro.com/

Aricò, P., Borghini, G., Di Flumeri, G., Sciaraffa, N. & Babiloni, F. 2018. Passive BCI beyond the lab: current trends and future directions. Physiological Measurement, 39, 08TR02.

Aristidou, A. & Lasenby, J. 2010. Motion capture with constrained inverse kinematics for real-time hand tracking. IEEE.

Aricò, P., Borghini, G., Di Flumeri, G., Sciaraffa, N. & Babiloni, F. 2018. Passive BCI beyond the lab: current trends and future directions. Physiological Measurement, 39, 08TR02.

Aristidou, A. & Lasenby, J. 2010. Motion capture with constrained inverse kinematics for real-time hand tracking. IEEE.

Autodesk Inc. 2021. viewed September 2021, https://www.autodesk.com/products/maya/

Baig, M. Z., Aslam, N. & Shum, H. P. H. 2020. Filtering techniques for channel selection in motor imagery EEG applications: a survey. The Artificial intelligence review, 53, 1207-1232.

Batula, A. M., Kim, Y. E. & Ayaz, H. 2017. Virtual and Actual Humanoid Robot Control with Four-Class Motor-Imagery-Based Optical Brain-Computer Interface. Biomed Res Int, 2017, 1463512-13.

Bittium. 2021. viewed September 2021, https://www.bittium.com/medical/bittium-neurone

Black, M. H., Chen, N. T. M., Iyer, K. K., Lipp, O. V., Bölte, S., Falkmer, M., Tan, T. & Girdler, S. 2017. Mechanisms of facial emotion recognition in autism spectrum disorders: Insights from eye tracking and electroencephalography. Neurosci Biobehav Rev, 80, 488-515.

Bockbrader, M., Annetta, N., Friedenberg, D., schwemmer, M., Skomrock, N., Colachis, S., Zhang, M., Bouton, C., Rezai, A., Sharma, G. & Mysiw, W. J. 2019. Clinically Significant Gains in Skillful Grasp Coordination by an Individual With Tetraplegia Using an Implanted Brain-Computer Interface With Forearm Transcutaneous Muscle Stimulation. Arch Phys Med Rehabil, 100, 1201-1217.

Boord, P., Craig, A., Tran, Y. & Nguyen, H. 2010. Discrimination of left and right leg motor imagery for brain–computer interfaces. Med Biol Eng Comput, 48, 343-350.

Bousseta, R., El Ouakouak, I., Gharbi, M. & Regragui, F. 2018. EEG Based Brain Computer Interface for Controlling a Robot Arm Movement Through Thought. Ingénierie et recherche biomédicale, 39, 129-135.

Burns, A. S., Marino, R. J., Flanders, A. E. & Flett, H. 2012. Chapter 3 - Clinical diagnosis and prognosis following spinal cord injury. In: VERHAAGEN, J. & MCDONALD, J. W. (eds.) Handbook of Clinical Neurology. Elsevier.

CERF, V. & KAHN, R. 1974. A PROTOCOL FOR PACKET NETWORK INTERCOMMUNICATION. IEEE TRANSACTIONS ON COMMUNICATIONS, 22, 637-648.

CHAE, Y., JEONG, J. & JO, S. 2012. TOWARD BRAIN-ACTUATED HUMANOID ROBOTS: ASYNCHRONOUS DIRECT CONTROL USING AN EEG-BASED BCI. IEEE TRANSACTIONS ON ROBOTICS, 28, 1131-1144.

CHEN, S.-C., CHEN, Y.-J., ZAENI, I. A. E. & WU, C.-M. 2017. A SINGLE-CHANNEL SSVEP-BASED BCI WITH A FUZZY FEATURE THRESHOLD ALGORITHM IN A MAZE GAME. INTERNATIONAL JOURNAL OF FUZZY SYSTEMS, 19, 553-565.

CHO, H., AHN, M., KWON, M. AND JUN, S. 2018. A STEP-BY-STEP TUTORIAL FOR A MOTOR IMAGERY–BASED BCI. BRAIN–COMPUTER INTERFACES HANDBOOK, PP.445-460.

COSTA, Á., HORTAL, E., IÁÑEZ, E. & AZOŔN, J. M. 2014. A SUPPLEMENTARY SYSTEM FOR A BRAIN-MACHINE INTERFACE BASED ON JAW ARTIFACTS FOR THE BIDIMENSIONAL CONTROL OF A ROBOTIC ARM. PLOS ONE, 9, E112352-E112352.

DHOLE, S. R., KASHYAP, A., DANGWAL, A. N. & MOHAN, R. 2019. A NOVEL HELMET DESIGN AND IMPLEMENTATION FOR DROWSINESS AND FALL DETECTION OF WORKERS ON-SITE USING EEG AND RANDOM-FOREST CLASSIFIER. PROCEDIA COMPUTER SCIENCE, 151, 947-952.

EMOTIV. 2021. VIEWED SEPTEMBER 2021, HTTPS://WWW.EMOTIV.COM/EMOTIVPRO/

FOLGIERI, R., LUCCHIARI, C., GRANATO, M. & GRECHI, D. 2014. BRAIN, TECHNOLOGY AND CREATIVITY. BRAINART: A BCI-BASED ENTERTAINMENT TOOL TO ENACT CREATIVITY AND CREATE DRAWING FROM CEREBRAL RHYTHMS. IN: LEE, N. (ED.) DIGITAL DA VINCI: COMPUTERS IN THE ARTS AND SCIENCES. NEW YORK, NY: SPRINGER NEW YORK.

FRICONNEAU, J. P., BEAUDOIN, V., DAMMANN, A., DREMEL, C., MARTINS, J. P. & PITCHER, C. S. 2017. ITER HOT CELL—REMOTE HANDLING SYSTEM MAINTENANCE OVERVIEW. FUSION ENGINEERING AND DESIGN, 124, 673-676.

GEARWURX. N.D. VIEWED SEPTEMBER 2021, HTTPS://GEARWURX.COM/

HOLZ, E. M. M., BOTREL, L. M. S., KAUFMANN, T. P. & KÜBLER, A. P. 2015. LONG-TERM INDEPENDENT BRAIN-COMPUTER INTERFACE HOME USE IMPROVES QUALITY OF LIFE OF A PATIENT IN THE LOCKED-IN STATE: A CASE STUDY. ARCH PHYS MED REHABIL, 96, S16-S26.

JATOI, M. A. 2017. BRAIN SOURCE LOCALIZATION USING EEG SIGNAL ANALYSIS, MILTON : TAYLOR & FRANCIS GROUP.

JURCAK, V., TSUZUKI, D. & DAN, I. 2007. 10/20, 10/10, AND 10/5 SYSTEMS REVISITED: THEIR VALIDITY AS RELATIVE HEAD-SURFACE-BASED POSITIONING SYSTEMS. NEUROIMAGE, 34, 1600-1611.

KOTHE, C., MEDINE, D., BOULAY, C., GRIVICH, M., AND STENNER, T. 2019. LAB STREAMING LAYER (LSL). VIEWED SEPTEMBER 2021, HTTPS://LABSTREAMINGLAYER.READTHEDOCS.IO/

LI, M., WU, H., HANDROOS, H. & YANG, G. 2013. SOFTWARE DESIGN OF THE HYBRID ROBOT MACHINE FOR ITER VACUUM VESSEL ASSEMBLY AND MAINTENANCE. FUSION ENGINEERING AND DESIGN, 88, 1872-1876.

LÓPEZ-HERNÁNDEZ, J. L., GONZÁLEZ-CARRASCO, I., LÓPEZ-CUADRADO, J. L. & RUIZ-MEZCUA, B. 2019. TOWARDS THE RECOGNITION OF THE EMOTIONS OF PEOPLE WITH VISUAL DISABILITIES THROUGH BRAIN–COMPUTER INTERFACES. SENSORS (BASEL), 19, 2620.

Markovinović, I., Vrankić, M. & Vlahinic, S. 2020. Removal of eye-blink artifacts from EEG signal. Engineering review, 40, 101-111.

Microsoft. 2021. viewed September 2021, https://www.microsoft.com/

Miladinović, A., Ajčević, M., Jarmolowska, J., Marusic, U., Silveri, G., Battaglini, P. P. & Accardo, A. 2020. Performance of EEG Motor-Imagery based spatial filtering methods: A BCI study on Stroke patients. Procedia computer science, 176, 2840-2848.

Oostenveld, R. & praamstra, P. 2001. The five percent electrode system for high-resolution EEG and ERP measurements. Clin Neurophysiol, 112, 713-719.

OpenBCI. 2021. viewed April 2021, https://openbci.com/

Podoprikhin, D. A. 2015. Implementation of the "Labyrinth" Game by Brain –Computer Interface Tools. Computational mathematics and modeling, 26, 555-565.

Ramakuri, S., Kumar, V. & Gupta, B. 2015. Feature extraction from EEG signal through one electrode device for medical application.

Raspberry Pi Foundation. 2021. viewed September 2021, https://www.raspberrypi.org/

Sharif, S. & Ali, S. M. 2020. "I Felt the Ball"–The Future of Spine Injury Recovery. World Neurosurg, 140, 602-613.

Shih, J. J. M. D., Krusienski, D. J. P. & Wolpaw, J. R. M. D. 2012. Brain-Computer Interfaces in Medicine. Mayo Clin Proc, 87, 268-279. SoftBank Robotics. 2018. viewed September 2021, https://www.softbankrobotics.com/emea/en/nao

Sphero. 2021. viewed May 2021, https://sphero.com/

Stawicki, P., Gembler, F. & Volosyak, I. 2016. Driving a Semiautonomous Mobile Robotic Car Controlled by an SSVEP-Based BCI. Comput Intell Neurosci, 2016, 4909685-14.

Tong, F. & Pratte, M. S. 2012. Decoding patterns of human brain activity. Annu Rev Psychol, 63, 483-509.

Unity Technologies. 2021. viewed September 2021, https://unity.com/

Wang, H. & Bezerianos, A. 2017. Brain-controlled wheelchair controlled by sustained and brief motor imagery BCIs. Electronics letters, 53, 1178-1180.

Wang, Z., Yu, Y., Xu, M., Liu, Y., Yin, E. & Zhou, Z. 2019. Towards a Hybrid BCI Gaming Paradigm Based on Motor Imagery and SSVEP. International journal of human-computer interaction, 35, 197-205.

World Health Organization. 2020. Disability and health. Retrieved September 29, 2021, from https://www.who.int/news-room/fact-sheets/detail/disability-and-health

Xu, B., Song, A., Zhao, G., Xu, G., Pan, L., Yang, R., Li, H. & Cui, J. 2015. Design and evaluation of a motor imagery electroencephalogram-controlled robot system. Advances in mechanical engineering, 7, 168781401557360.